PARAMETRIC DYNAMIC MODE DECOMPOSITION FOR REDUCED ORDER MODELING

A Thesis

by

QUINCY AARON HUHN

Submitted to the Graduate and Professional School of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Chair of Committee,	Jean Ragusa
Co-Chair of Committee,	Mauricio Tano
Committee Member,	Alan Demlow
Head of Department,	Micheal Nastasi

August 2022

Major Subject: Nuclear Engineering

ABSTRACT

Dynamic Mode Decomposition (DMD) is a method by which spatial modes can be extracted from experimental or numerical data sets. DMD can be used as well to obtain a low rank or reduced operator for a physical system by singular value decomposition of temporal data sets. This work focuses on an extension of the general DMD algorithm to take advantage of parametric data sets, parametric DMD. Parametric DMD has application in multi-query problems such as design optimization and uncertainty quantification. The work comprises of two novel methods developed reduced eigen-pair interpolation and reduced Koopman operator interpolation and compares these methods to the current state of the art approach for parametric DMD stacked DMD. The methods are compared on a set of problems including a multiphysics radiative transfer example. The reduced Koopman operator interpolation method is shown to have improvement in computational efficiency as well as reconstruction error in select parametric problems.

CONTRIBUTORS AND FUNDING SOURCES

Contributors

This work was supported by a dissertation committee consisting of Dr. Jean Ragusa (chair) of the Department of Nuclear Engineering and Dr. Mauricio Tano (co-chair) of the Department of Nuclear Engineering and Dr. Alan Demlow of the Department of Mathematics. The work also benefitted from the guidance of Dr. Youngsoo Choi from Lawrence Livermore National Lab.

Funding Sources

Graduate study was supported by a Nuclear Regulatory Commission fellowship and a research grant from Lawrence Livermore National Lab.

NOMENCLATURE

DMD	Dynamic Mode Decomposition
ROM	Reduced Order Model
SVD	Singular Value Decomposition
rEPI	Reduced Eigen-Pair Interpolation
rKOI	Reduced Koopman Operator Interpolation

TABLE OF CONTENTS

Page

LIST OF FIGURES

FIGURE Page

LIST OF TABLES

TABLE Page

1. INTRODUCTION AND LITERATURE REVIEW

Physics simulations can be incredibly computationally expensive especially in cases with coupled physics, in cases where there is some form of parametric dependence in the problem, or where one or more of the physical quantities in the problem are uncertain. This computational burden is sometimes prohibitive enough where it would be preferable to have a lower order approximation in much less time than it would take to solve the higher order system with a finite element solver.

There is a large set of methods that aim to take the full order model and create a reduced order model (ROM) that can be used to find such a lower order approximate solution. These ROMs can be used in multi-query problems such as uncertainty quantification, optimization problems, and parametric sensitivity studies. There are two main types of ROM, Intrusive ROM and Non-Intrusive ROM.

Intrusive ROM requires direct manipulation of the governing equations of the system within whatever physics solver is used. This can be a disadvantage because there are some scenarios in which there will be no access to the source code of a physics solver or even where only the output data is available. One of the main methods that is an example of Intrusive ROM is Galerkin projection based Proper Orthogonal Decomposition.

Non-Intrusive ROM on the other hand requires no information about the governing equations of a physics system. Because it requires no information on the governing equations a Non-Intrusive ROM can be applied to any physical system very easily. The only thing that is required for Non-Intrusive ROMs is the parametric snapshots of the solutions of the underlying dynamic model making Non-Intrusive ROMs entirely data-driven. Machine learning systems such as neural networks are an example of Non-Intrusive ROMs as well as the method that will be presented in this thesis Parametric Dynamic Mode Decomposition (DMD).

DMD was first developed in fluid dynamics by Schmid et. al. in [1] to study the modal behavior of a given physics problem. In general DMD aims to use a time series of snapshots to develop an operator that can be decomposed and reduced to approximate the system with a reduced order.

Tu et. al. in [2] firmly establish the link between this operator and the Koopman Operator from Koopman spectral analysis which is crucial in both the general DMD algorithm and ones to be presented in this thesis.

There are a lot of extensions to the general DMD algorithm such as multiresolution DMD [3] that was developed for systems with dynamics on multiple scales, compressed DMD [4] which combines Fourier transforms and SVD to create an algorithm optimized for for large data sets such as high resolution video, higher order DMD [5] which extends DMD by adding time lagged snapshots in order to better treat systems with some periodic dynamics, sparsity-promoting DMD [6] which aims to obtain a DMD reconstruction with fewer modes while still maintaining a reasonable approximation, as well as many other extensions. Most extensions to DMD are for non-parametric problems the main exception being in [7] where an extension to DMD was proposed for a parametric problems which was applied to a flame-bifurcation analysis this is the current state of the art that the proposed methods will be compared against.

DMD has been used in nuclear engineering before to speed up source iterations [8], in subcritical metal systems [9], and in calculating time eigenvalues of the neutron transport equation [10]. The research contained in this thesis is novel in two ways; firstly in the application to nuclear engineering where parametric DMD has not been applied and secondly in that two novel methods for performing parametric DMD are presented to be analyzed.

The remainder of the thesis will be divided as follows; Methods will give an overview of the general DMD algorithm as well the two novel parametric DMD methods and the current state of the art, Results will show results on a set of selected problems including a radiative diffusion problem, and Summary and Conclusion will give a summary of the work completed as well as conclusions on the methods presented.

2. METHODS

2.1 DMD

In general we will consider a non-linear governing law of the form

$$\frac{\partial u}{\partial t} = \mathcal{F}(u, t) \tag{2.1}$$

where u is a solution of the governing law and $\mathcal{F}(u, t)$ is a non-linear operator that represents the governing law. In general u can be a set of coupled physics in a multi-physics problem in which case $\mathcal{F}(u, t)$ represents the multiple governing laws. One crucial aspect of this general form is the dependence on time as the Koopman operator works on a time series of data. From this governing law we can also further add dependence on a parameter μ

$$\frac{\partial u}{\partial t} = \mathcal{F}(u, \mu, t) \tag{2.2}$$

this parameter μ can represent any aspect of the problem that can vary or is uncertain which will lead to the multi-query problem which motivated parametric DMD. DMD can be performed with experimental data in which case the time series of snapshots can be gathered from experimental apparatus. However, we propose DMD based off of data gathered from a finite element physical simulation. We perform discretization in space such that $\mathbf{u} \in \mathcal{R}^m$ where m is the number of discretization points where we observe the value of $\mathbf{u}$ and obtain ODE(s) for the governing law(s)

$$\frac{d\mathbf{u}}{dt} = F(\mathbf{u}, \mu, t) \tag{2.3}$$

where F represents the non-linear dynamics in the discrete setting and $\mathbf{u}$ is at a given time t. To get sufficient fidelity $\mathbf{u}$ can be quite large. The goal of DMD is to find a linear operator that

approximates F such that

$$\frac{d\mathbf{u}}{dt} = A\mathbf{u} \tag{2.4}$$

where A is the Koopman operator we will attempt to approximate in DMD.

Continuing in this section the standard DMD algorithm will be presented. We start with a time series of snapshots $\mathbf{S} = [\mathbf{u}_0, \mathbf{u}_1, \cdots, \mathbf{u}_n] = [\mathbf{u}_i]_{i=0}^{n} \in \mathbb{R}^{m\times n}$ obtained from experiments or physical simulations. Where n is the number of snapshots and m is the full order dimension of the system, either the number of physical measurements from an experiment or the degrees of freedom from a physical simulation. Then $\mathbf{S}$ is separated into a lagged matrix$\mathbf{S}^{-} = [\mathbf{u}_i]_{i=0}^{n-1} \in \mathbb{R}^{m\times(n-1)}$ and a forward matrix of snapshots $\mathbf{S}^{+} = [\mathbf{u}_i]_{i=1}^{n} \in \mathbb{R}^{m\times(n-1)}$. And we then define the Koopman operator to be $\mathbf{A} \in \mathbb{R}^{m\times m}$ such that

$$\mathbf{S}^{+} = \mathbf{A}\mathbf{S}^{-}\,. \tag{2.5}$$

We then derive a way of approximating this Koopman operator at a reduced order starting by taking the Singular Value Decomposition (SVD) of the lagged matrix $\mathbf{S}^{-}$.

$$\mathbf{S}^{-} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^{T}\,, \tag{2.6}$$

where $\mathbf{U} \in \mathbb{R}^{m\times m}$ and $\mathbf{V}^{T} \in \mathbb{R}^{n\times n}$ are two orthogonal matrices that respectively contain the ordered modes of the space and time evolution of the physical system and $\mathbf{\Sigma} \in \mathbb{R}^{m\times n}$ is a diagonal matrix that stores the ordered singular values of the system. The orthogonality of these two matrices will be important in ensuring that the DMD modes of the system are the true modes of the physical system as will be shown later in the derivation. We then reduce order of the system to rank r by truncating $\mathbf{U}$ and $\mathbf{V}$ to the first r columns by defining $\mathbf{U}_r = [\mathbf{u}]_{i=1}^{r} \in \mathbb{R}^{m\times r}$ and $\mathbf{V}_r = [\mathbf{v}]_{i=1}^{r} \in \mathbb{R}^{n\times r}$, respectively. We also truncate $\mathbf{\Sigma}$ to the first r singular values making the diagonal matrix $\mathbf{\Sigma}_r = \mathrm{diag}([\Sigma]_{i=1}^{r}) \in \mathbb{R}^{r\times r}$. In this thesis the performance of DMD as a function of rank r will be evaluated, but frequently r will be chosen such that all of the singular values retained are above some tolerance.

$$\mathbf{S}^{-} \approx \mathbf{U}_r\mathbf{\Sigma}_r\mathbf{V}_r^{t}\,. \tag{2.7}$$

Now using the orthogonality of $\mathbf{U}$ and $\mathbf{V}$ and plugging Eq. 2.7 into Eq. 2.5 we find an approximation of the true Koopman operator as

$$\mathbf{A}_r \equiv \mathbf{U}_r^T \mathbf{A} \mathbf{U}_r = \mathbf{U}_r^T \mathbf{S}^+ \mathbf{V}_r \mathbf{\Sigma}_r^{-1} \tag{2.8}$$

Then we simply perform the eigen-decomposition of $\mathbf{A}_r \in \mathbb{R}^{r \times r}$

$$\mathbf{A}_r \mathbf{W} = \mathbf{\Lambda} \mathbf{W} \tag{2.9}$$

and the projected DMD modes are then $\mathbf{\Phi} = \mathbf{U}\mathbf{W}$ as defined in [2]. The reconstructed system then can be evaluated at any time with the formula

$$\mathbf{u}(t) = \mathbf{\Phi} \mathbf{\Lambda}^{t/\Delta t} \mathbf{b}_0 = \sum_{i=1}^{r} \mathbf{b}_{0i} \phi_i \lambda^{t/\Delta t}\,, \tag{2.10}$$

where $b_0 = \mathbf{\Phi}^\dagger \mathbf{u}_0 \in \mathbb{R}^m$ is the vector of initial conditions, $\mathbf{\Phi}^\dagger$ is the pseudo-inverse of the DMD modes, and $\mathbf{u}_0 \in \mathbb{R}^m$ is the initial snapshot at $t = 0$.

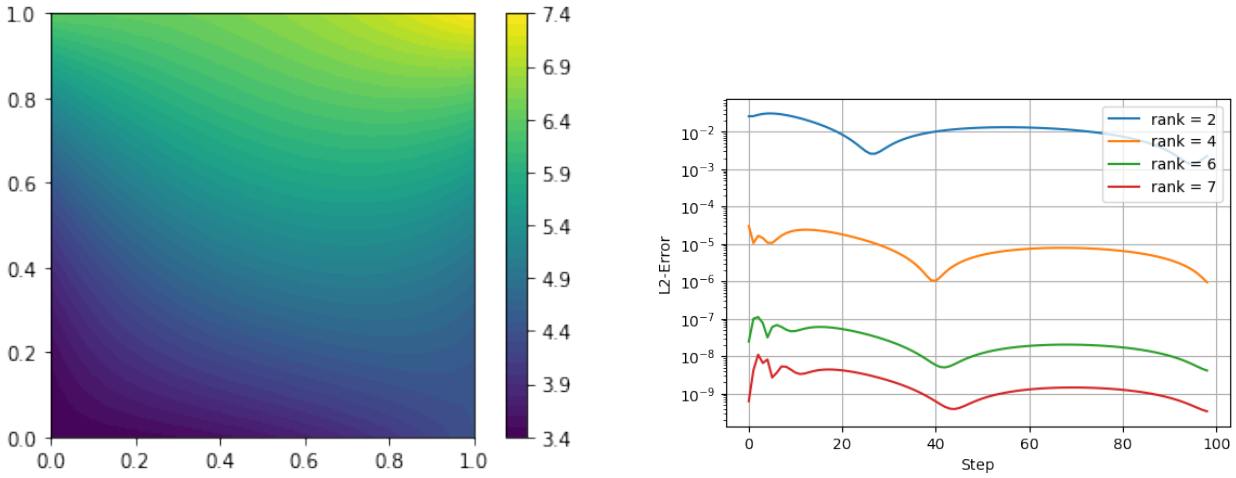
2.2 DMD Demonstration

In this section a demonstration of the standard DMD algorithm will be presented. For this section the implementation in the python package pyDMD [11] will be used. Appendix 1 shows a basic implementation of DMD using pyDMD. The algorithm will be presented on a transient heat conduction problem with the governing law given by Eq. (2.11):

$$\frac{\partial T}{\partial t} = \nabla^2 T + f \qquad \forall \vec{r} \in \Omega \tag{2.11}$$

where $\Omega = \{(x, y) \in [0, 1] \otimes [0, 1]\}$ is the problem domain, with a Dirichlet boundary condition $T(x, y, t) = 1 + x^2 + 3y^2 + 1.2t$ on the entire boundary. A finite element solution to the problem at $t = 2$ is shown in Fig. 2.1a

In 2.1b the error of DMD reconstruction for various ranks is shown. In the results section a



(a) Solution at T=2

(b) L_2 Error of DMD Reconstruction

Figure 2.1: Reconstructive $\langle E(t_i) \rangle$ Error of the General DMD Algorithm on a Transient Heat Conduction Problem

definition for the L_2 error used is given in Eq. 3.1. As the rank increases the L_2 error of the DMD reconstruction decreases. Another feature of DMD is the speed at which a reconstruction can be calculated. The simple finite element calculation takes on the order of one second to calculate while the DMD reconstructions all take on the order of 0.01 seconds. This speed compared to full order solutions is one of the reasons that DMD is attractive to perform reduced order solutions with.

2.3 Parametric DMD

In this section the state of the art algorithm for parametric DMD will be presented as well as two new methods that were developed during this research. Each of these methods expands upon the standard DMD algorithm to take advantage of information from a large set of snapshots that are evaluated at values of a parameter or set of parameter values μ in order to generate a solution at a new parameter value(s) μ_x. The goal of each of these methods is to find the evolution of the solution to the physical solution at μ_x with minimal error as efficiently as possible using DMD.

2.3.1 Stacked Parametric DMD

First the current state of the art in parametric DMD, stacked DMD, will be shown as presented in [7]. This method is very similar to the standard non-parametric DMD algorithm except the time series of snapshots for the different parameter values will be "stacked" to form one matrix to perform the standard DMD algorithm on. Taking $S(\mu_i) = S_i$ we form $S \in R^{(pm)xn}$ where (pm) is the number of parameters multiplied by the number of data points in each snapshot and n is the number of time points in the set of snapshots. S will be the input to the non-parametric DMD algorithm in the form:

$$S = \begin{bmatrix} - - S_1 - - \\ - - S_2 - - \\ - - S_3 - - \\ \vdots \\ - - S_N - - \end{bmatrix} \tag{2.12}$$

After performing the standard non-parametric DMD algorithm as outlined in the previous section we can obtain projected DMD modes ϕ. The modes will be in the form shown in Eq. 2.13 where $\phi \in R^{(pm)xr}$ is the stacked mode matrix where (pm) is the number of parameters multiplied by the number of data points in each snapshot and r is the truncation rank of the reduced order reconstruction used in DMD. The stacked mode matrix ϕ can then be easily separated into the modes at each of the parameter values.

$$\phi = \begin{bmatrix} - - \phi_1 - - \\ - - \phi_2 - - \\ - - \phi_3 - - \\ \vdots \\ - - \phi_N - - \end{bmatrix} \tag{2.13}$$

With the projected DMD modes calculated for each parameter value it is easy to obtain the initial conditions b_0 for each parameter value. One of the main limitations of stacked DMD comes from

the snapshots being stacked. Because there is only one DMD performed the time eigenvalues λ will be shared at each parameter value. With the shared time eigenvalues all we need to do in order to evaluate DMD at a new parameter value is interpolate the modes and b_0 to find approximate values for both quantities at the new parameter. We can then use the formula in Eq. 2.14 to evaluate the system at a new parameter value μ_x.

$$u(\mu_x, t) = \phi_x \lambda^{t/\Delta t} b_{0,u} \tag{2.14}$$

Another limitation besides the shared time eigenvalue is the large computational cost of performing SVD on a large matrix like the stacked snapshots form. The computational cost of performing SVD scales with the square of the matrix size making performing SVD on the stacked matrix potentially computationally difficult especially in problems with large numbers of parameter values. The new methods that will be presented will address these two limitations of stacked DMD.

2.3.2 Reduced Eigen-pair Interpolation

The first of the two new methods to be introduced is reduced eigen-pair interpolation or rEPI. In this method we start by performing the standard DMD algorithm at each known parameter value. After this we have a set of eigenvalues and eigenvectors that we can simply interpolate to the new parameter value. We do this and also interpolate the initial values b_0 that are calculated at each parameter value. With these three values calculated we can simply use Eq. 2.14 to evaluate the system at a new parameter value.

This method sets out to solve the two limitations of stacked DMD in the most straightforward way. The computational cost is reduced and will now scale in $O(n)$ with n representing the number of parameters where stacked DMD will scale with $O(n^2)$. The time eigenvalues are also no longer shared between all parameter values as they are calculated as a part of the DMD performed at each parameter value.

2.3.3 Reduced Koopman Operator Interpolation

The second method that to be introduced in this this is reduced Koopman Operator Interpolation or rKOI. This method solves the two issues detailed of stacked DMD in a more clever way than the rEPI method. The first step in this method is constructing the reduced Koopman Operator at each of the parameter values as shown in Eq. 2.8. Then we interpolate the reduced Koopman operator to the new parameter value $A_{r,x}$. We then continue with DMD as normal by performing the eigen-decomposition:

$$A_{r,x}w_x = \lambda_x w_x \tag{2.15}$$

Then we can interpolate the modes to obtain U_x at the new parameter value. We then calculate $\phi_x = U_x w_x$ and after interpolating $u_{0,x}$ we then calculate the evaluation given in Eq. 2.14.

This method also solves both limitations of stacked DMD. The time eigenvalues are no longer shared between all parameters as they are calculated as part of the eigen-decomposition of the interpolated reduced Koopman operator, and this method will scale with $O(n)$ where n is the number of parameters instead of the $O(n^2)$ scaling of stacked DMD.

3. RESULTS

In this section a demonstration of the generic DMD algorithm as well as parametric results on three problems will be presented. For the parametric results the finite element physical simulations will be separated into a training set that will be used to interpolate to the new parameter values and a testing set that will be used to evaluate each of the different methods. To quantify the reconstruction errors from the various methods the relative L_2 error norm will be used, given in Eq. (3.1) where u^{ref} is the reference snapshot from the testing set and u^{DMD} is the DMD reconstructed solution at the parameter value. The relative L_2 error norm is calculated for each time step. The time-averaged relative L_2 error norm given in Eq. (3.2) will also be presented where E is the relative L_2 error norm and N_T is the number of time steps.

$$E(t_i, \mu_j) = \frac{\|u^{\text{DMD}}(t_i, \mu_j) - u^{\text{ref}}(t_i, \mu_j)\|_2}{\|u^{\text{ref}}(t_i)\|_2}\,, \tag{3.1}$$

$$\bar{E}(\mu_j) = \frac{\sum_i^{N_T} E(t_i, \mu_j)}{N_T}\,. \tag{3.2}$$

The error norm will also be averaged over parameter values sampled from the testing set as shown in Eq. (3.3)

$$\langle E(t_i)\rangle = \frac{\sum_j^{N_S} E(t_i, \mu_j)}{N_S}\,, \tag{3.3}$$

$$\langle \bar{E}\rangle = \frac{\sum_j^{N_S} \tilde{E}(\mu_j)}{N_S}\,. \tag{3.4}$$

3.1 Transient Nonlinear Diffusion Problem

3.1.1 Problem Description

The first set of parametric results deals with a transient nonlinear diffusion problem. The governing law is given in Eq. (3.5):

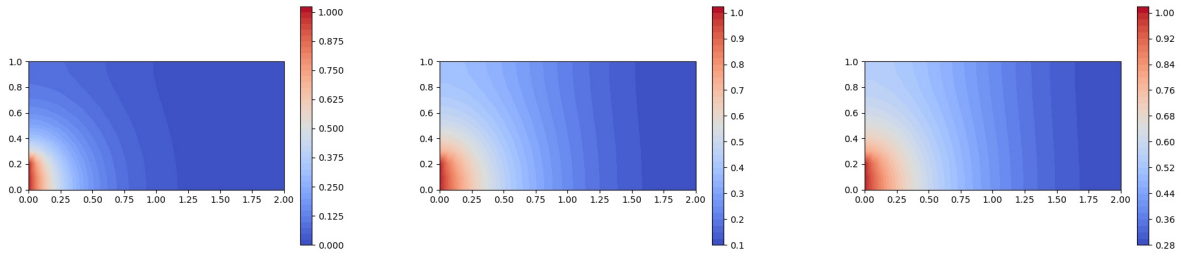
$$\frac{\partial T}{\partial t} + w \cdot \nabla T = \nabla \cdot k(T) \nabla T + f \qquad \forall \vec{r} \in \Omega \tag{3.5}$$

where $\Omega = \{(x, y) \in [0, 2] \otimes [0, 1]\}$ is the problem domain, with a Dirichlet boundary condition $T(x = 0, 0 > y < 0.2, t) = 1$ on the bottom of left wall, and homogeneous Neumann boundary condition elsewhere, $\nabla T \cdot n = 0$. The advection velocity for this problem is $w = 0.1\hat{i} + 0\hat{j}$. The thermal conductivity k is a non-linear function of temperature, given by Eq. (3.6), where a and b are parameters of the conductivity correlation. The unknown parameter chosen vary in this problem is the exponent coefficient b, while a is fixed to a value of 0.01; the parameter range for b is $0 \leq b \leq 4$. Note that, when $b = 0$, the problem is linear. Sample snapshot solutions for $b = 0$ and $b = 4$ are given in Fig. 3.1, at different time instants.

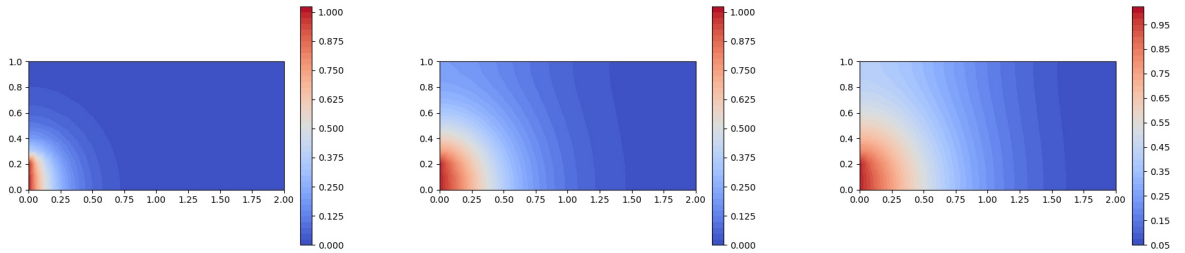
$$k(T) = a + T^b \,. \tag{3.6}$$

3.1.2 Problem Results

In this problem 50 snapshots were generated with a finite element solver with varying thermal conductivity non-linearity exponents b equally spaced in $[0, 4]$. A testing set of 20 snapshots, sampled within the parameter range, was used. The relative L_2 error norm is computed over the testing set, $\langle E(t_i) \rangle$. Data points from the testing set are added until the change in the $\langle \bar{E} \rangle$ error is less than 0.1% or until all of the testing set samples have been utilized. This process is employed in the subsequent results as well. In Fig. 3.2 the $\langle E(t_i) \rangle$ error is shown and in Table 3.1 the $\langle \bar{E} \rangle$ error of each of the three methods as a function of DMD reconstruction rank is shown.



(a) Solution at $b = 0$, $t = 0.1$ (b) Solution at $b = 0$, $t = 0.5$ (c) Solution at $b = 0$, $t = 1.0$

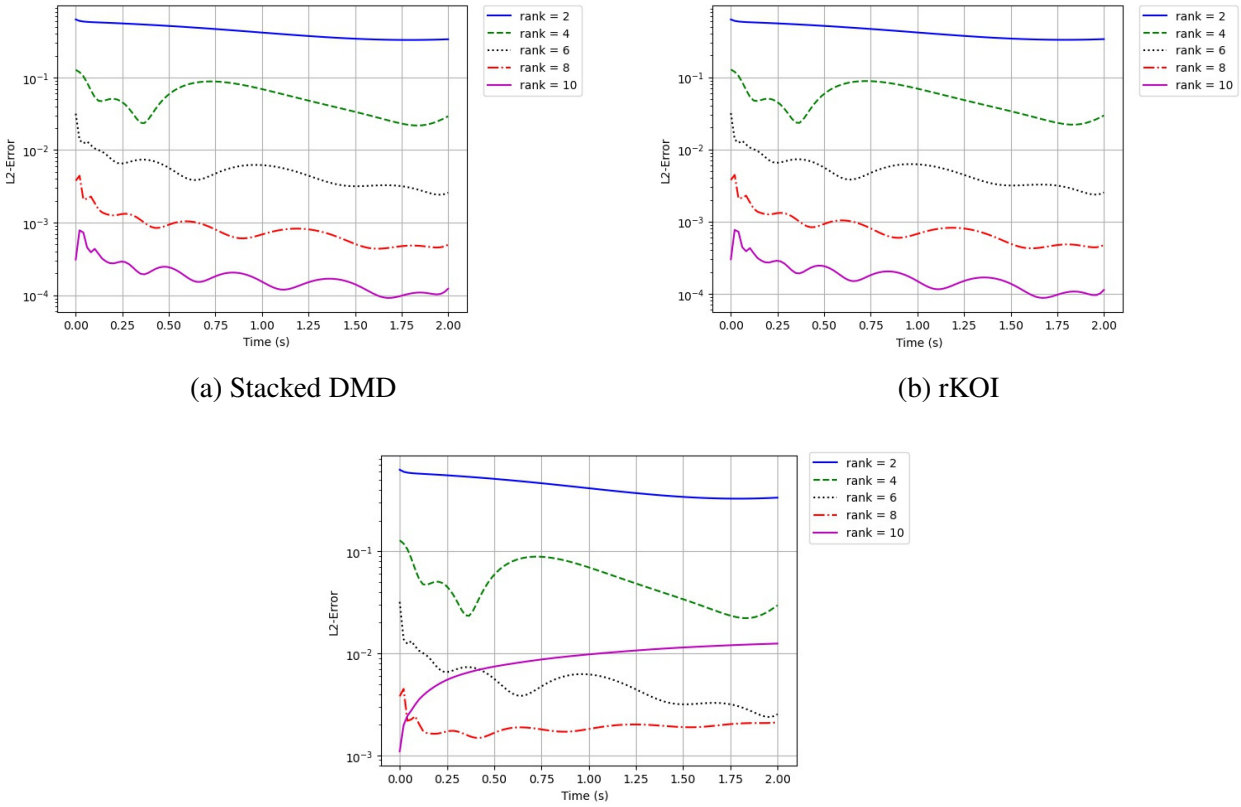


(d) Solution at $b = 4$, $t = 0.1$ (e) Solution at $b = 4$, $t = 0.5$ (f) Solution at $b = 4$, $t = 1.0$

Figure 3.1: Sample Snapshot Solutions of the Nonlinear Diffusion Problem at the Extremes of the Parameter b Range Over Time.

Rank	2	4	6	8	10
Stacked Error	4.31 E-1	5.21 E-2	5.48 E-3	8.86 E-4	1.89 E-4
rKOI Error	4.37 E-1	5.21 E-2	5.57 E-3	8.82 E-4	1.87 E-4
rEPI Error	4.34 E-1	5.24 E-2	5.51 E-3	7.91 E-3	2.44 E-2

Table 3.1: $\langle \bar{E} \rangle$ Error in Rank for the Three Parametric DMD Methods on the Nonlinear Diffusion Problem



(a) Stacked DMD

(b) rKOI

(c) rEPI

Figure 3.2: $\langle E(t_i) \rangle$ Error as a Function of Rank for the Three Parametric DMD Methods on the Nonlinear Diffusion Problem.

For this problem, the error monotonically decreases as a function of the approximation rank r for stacked DMD and rKOI. Stacked DMD and reduced Koopman operator interpolation perform very similarly while the reduced Eigen-pair interpolation method presents higher errors. The three methods perform the same for rank 2, 4, and 6 reconstructions but rEPI has an increase in error in the rank 8 and 10 reconstruction. This instability in rank is a very undesirable trait because it is impossible to know a priori if increasing rank will decrease the error, making the rEPI method of questionable use for parametric DMD.

3.2 Incident Jet Problem

3.2.1 Problem Introduction

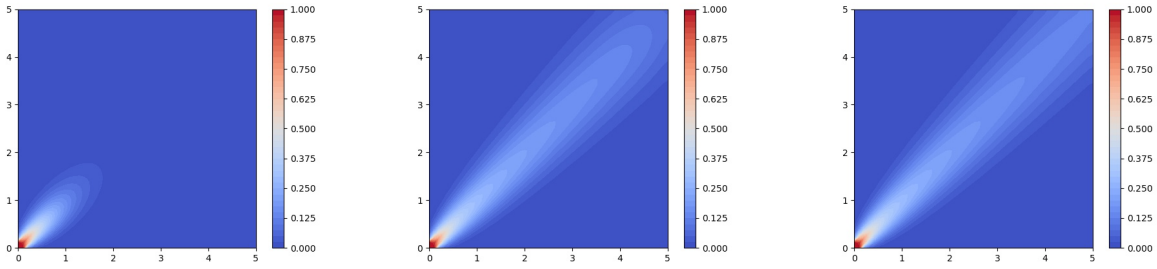
The second parametric problem is an incident-jet problem. Compared to the previous nonlinear diffusion problem, the incident-jet test case is more advection-dominated. The thermal conductivity was chosen to be independent of temperature and varies in $0.2 \leq k \leq 5$. The governing law is given in Eq. (3.7).

$$\frac{\partial T}{\partial t} + w \cdot \nabla T = \nabla \cdot k \nabla T \tag{3.7}$$

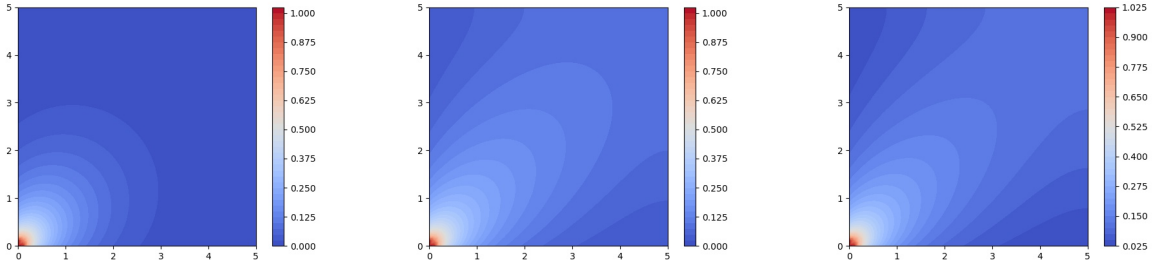
The domain for this problem is $\Omega = \{(x, y) \in [0, 5] \otimes [0, 5]\}$ with a Dirichlet boundary condition of $T(x < 0.1, y < 0.1, t) = sin(10\pi t)$ in the bottom left corner, and a homogeneous Neumann boundary condition of $\nabla T \cdot n = 0$ elsewhere. In this problem, the advection velocity w has x and y components such that $w = 5\hat{i} + 5\hat{j}$. The parameter chosen to vary in this problem is the thermal conductivity, the parameter range for k is $0.2 \leq k \leq 5$. Example solutions at $k = 0.2$ and $k = 5$ over time are given in Fig. 3.3.

3.2.2 Problem Results

In this problem, 50 snapshots were generated at different thermal conductivities k equally spaced in $[0.2, 5]$. A testing set was created by randomly setting aside 20 snapshots, while the training set contained the remainder snapshots. Both the $\langle E(t_i) \rangle$ error and the $\langle \bar{E} \rangle$ error to get a relative L_2 error norm over the testing set. In Fig. 3.4 the ensemble-averaged error $\langle E(t_i) \rangle$ is shown and in Table 3.2 the ensemble- and time-averaged error $\langle \bar{E} \rangle$ of each of the three methods,



(a) $k = 0.2, t = 0.2$ (b) $k = 0.2, t = 1.0$ (c) $k = 0.2, t = 2.0$



(d) $k = 5.0, t = 0.2$ (e) $k = 5.0, t = 1.0$ (f) $k = 5.0, t = 2.0$

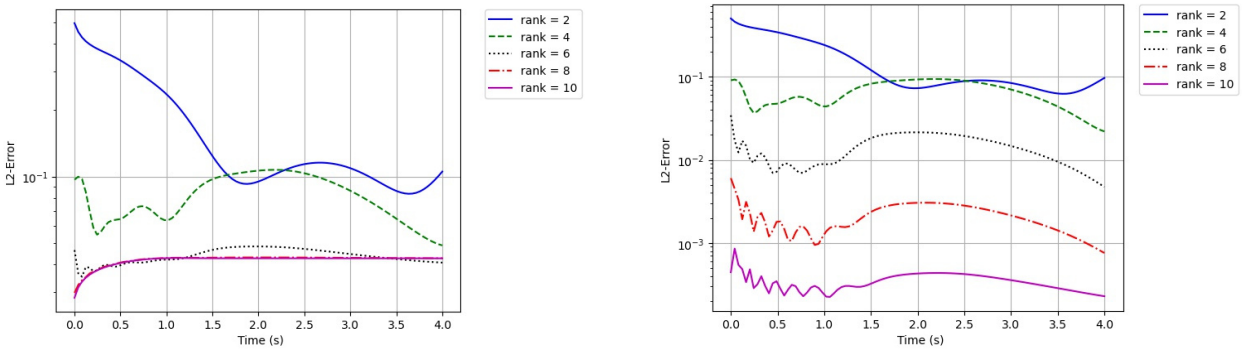
Figure 3.3: Sample Solutions of the Incident Jet Problem at the Extremes of the Parameter Range Over Time

Rank	2	4	6	8	10
Stacked Error	1.67 E-1	8.45 E-2	5.21 E-2	5.00 E-2	4.99 E-2
rKOI Error	1.59 E-1	6.57 E-2	1.41 E-2	2.14 E-3	3.66 E-4
rEPI Error	1.60 E-1	6.52 E-2	1.40 E-2	2.47 E-3	3.64 E-3

Table 3.2: $\langle \bar{E} \rangle$ as a Function of Rank for the Three Parametric DMD Methods on the Incident-Jet Problem

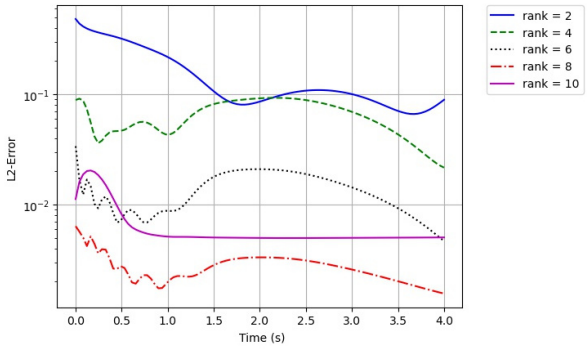
as a function of DMD reconstruction rank is shown.

In this problem rKOI is the clear winner out of the three methods. The stacked DMD error saturates at a relatively large value, even when rank is increased. This may be due to the fact that DMD time eigenvalues are generated and shared among all training snapshots, a drawback that is emphasized more in this advection-dominated problem. The rEPI method exhibits the same behavior as in the previous transient nonlinear diffusion problem, where the error does not monotonically decrease with rank. This behavior is disqualifying for a parametric DMD method due



(a) Stacked DMD

(b) rKOI



(c) rEPI

Figure 3.4: $\langle E(t_i) \rangle$ Error as a Function of Rank for the Three Parametric DMD Methods on the Incident-Jet Problem

to the unpredictability of its behavior in rank. The rKOI method performed better than the rEPI approach for two test problems, thus there seems to be little merit to the rEPI method so for the rest of the thesis only results with rKOI and stacked DMD will be shown.

3.3 Radiative Diffusion Problem

3.3.1 Problem Intro

The main problem which will have the most study is a radiative diffusion problem. The differential equations are given in Eq. 3.9 and Eq. 3.8.

$$\frac{\partial T}{\partial t} - \nabla \cdot (D_T \nabla T) = -\sigma_a (T^4 - E) \tag{3.8}$$

$$\frac{\partial E}{\partial t} - \nabla \cdot (D_r \nabla E) = \sigma_a (T^4 - E) \tag{3.9}$$

Where E is the radiation energy and T is the material temperature. The absorption cross section σ_a is given by Eq. 3.10 where α is an parameter that will be varied and z is the atomic number of the material which will also be a parameter that will be varied in this problem.

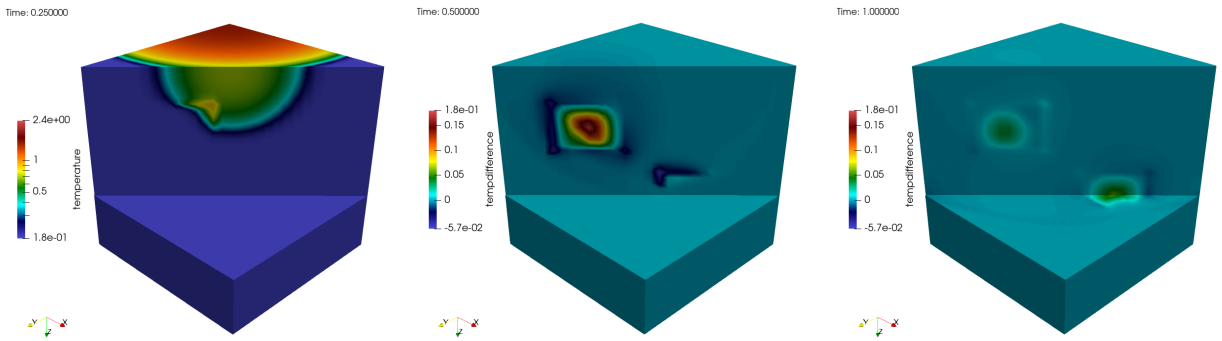
$$\sigma_a = \frac{z^\alpha}{T^3} \tag{3.10}$$

D_t the material conduction coefficient is defined in Eq. 3.11 and D_r the radiation diffusion coefficient is defined in Eq. 3.12.

$$D_T(T) = 1.0 * 10^{-2} T^{2.5} \tag{3.11}$$

$$D_r(E, T) = \frac{1}{3\sigma_a + |\nabla E|/E} \tag{3.12}$$

This problem is a 3-D version of a problem given in (V.A. Mousseau, D.A. Knoll). There are two cubes of high-Z material in a medium of $Z = 1$ material. The two cubes are at $(3/32 \leq x \leq 7/32, 9/32 \leq y \leq 13/32, 3/32 \leq z \leq 7/32)$ and $(9/32 \leq x \leq 13/32, 3/32 \leq y \leq 7/32, 9/32 \leq z \leq 13/32)$. The initial condition for T is given by 3.13 and the initial condition for E is given by



(a) Solution at Z=10, t=0.25 (b) Difference at t=0.5 (c) Difference at t=1.0

Figure 3.5: Sample Solution of the Radiative Diffusion Problem and Plots of the Difference Between the Solutions at Z=5 and Z=15 at Different Times

3.14.

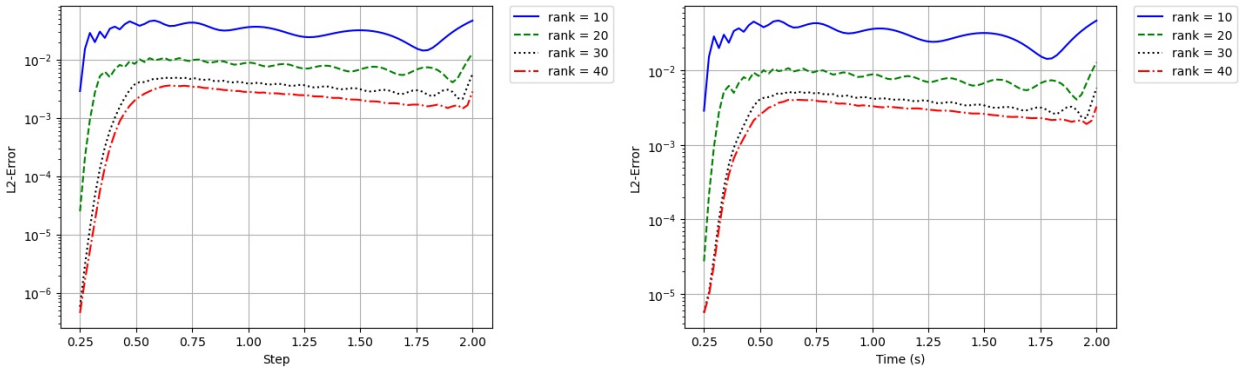
$$T(x,y,z,t=0) = 0.001 + 100 * (e^{-100*(x^2+y^2+z^2)})^{\frac{1}{4}} \tag{3.13}$$

$$E(x,y,z,t=0) = 0.001 + 100 * e^{-100*(x^2+y^2+z^2)} \tag{3.14}$$

The parameters that will be adjusted in this problem are the Z of the two cubes and the α exponent over the Z in the absorption cross section formula. A sample solution as well as difference between two solutions at different parameters is given by Fig. 3.5.

3.3.2 Problem Results

In this problem first only Z will be varied and $\alpha = 3.0$ will be kept fixed. Varying multiple parameters will be done in the following section. Due to the long run times of these simulations a pre-generated a set of snapshots is used that will be divided into a training and testing set. The data set includes the range $Z \in [1, 15]$ at each integer in that range. The odd values of Z are used for the training snaphots and the even values for the testing values. Like in the previous two problems each error is reported in this section was run for different points in the testing set until the change in average error was less than 0.1% or until all of the testing set was used. In this problem the individual snapshots include both E and T. The error is calculated as an average over the entire snapshot. In Fig. 3.6 the error of stacked DMD and rKOI over rank is shown. The run time of the



(a) Stacked DMD

(b) rKOI

Figure 3.6: $\langle E(t_i) \rangle$ Error as a Function of Rank for Two Parametric DMD Methods on the Radiative Diffusion Problem

Rank	10	20	30	40	50	60
Stacked Run Time (s)	12.6	14.3	16.1	19.1	21.6	24.2
rKOI Run Time (s)	12.6	13.5	15.0	16.1	17.2	18.1
% Speed Up	0.0%	5.6%	6.8%	15.7%	20.3%	25.2%

Table 3.3: Runtime of rKOI and Stacked DMD on the Radiative Diffusion Problem

two methods is also shown in Table 3.3. This timing includes the SVD, eigen-decomposition, and the evaluation of the methods but excludes the interpolations required.

In this problem rKOI and stacked DMD perform similarly in terms of error with stacked DMD slightly beating out rKOI as shown in Table 3.4. However rKOI sees a speed up of up to 25.2% over stacked DMD. The new method therefore is able to perform very close to the state of the art in terms of error while running up to 25% faster.

Rank	10	20	30	40
rKOI Error	3.11 E-2	7.49 E-3	3.49 E-3	2.65 E-3
Stacked Error	3.11 E-2	7.50 E-3	3.24 E-3	2.16 E-3

Table 3.4: $\langle \bar{E} \rangle$ Error in Rank for Two Parametric DMD Methods on the Radiative Diffusion Problem

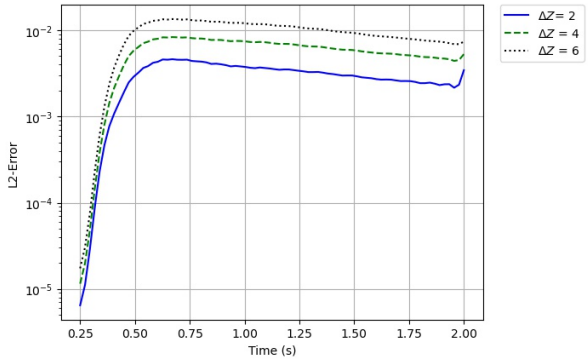


Figure 3.7: $\langle E(t_i) \rangle$ Error as Function of Parameter Spacing on the rKOI Method

3.4 rKOI Performance Study

In this section the Radiative Diffusion problem will be used to study the behavior of the rKOI method in more detail. The error as a function of number of snapshots, performance on a problem with 2 parameters, and how the multi-physics nature of the problem effects the behavior will all be studied.

First the error as a function of the number of snapshots will be investigated. Using the same parameter range for Z different ΔZ are used to generate 3 training sets, one with $\Delta Z = 2$, ($Z = [1, 3, 5, 7, 9, 11, 13, 15]$), one with $\Delta Z = 4$, ($Z = [1, 5, 9, 13]$), and one with $\Delta Z = 6$, ($Z = [1, 7, 13]$). The testing set for this problem is the even integers from 2 to 12. Rank 40 reconstruction was used for each of the runs of this problem, again running for the different points in the testing set until the change in average error was less than 0.1% or until all of the testing set was used. The results are shown in Fig. 3.7. As could be expected the more snapshots that are used the lower the error of the solution is. This property could potentially be used in the future to determine how many full order runs you need to do in order to achieve a certain error in the the reduced solution over the entire parameter domain.

Next the problem will be expanded to a 2-parameter problem where in addition to varying Z in the snapshots the α of the cubes will also be varied. Z will be varied between 5 and 15 including every whole value of Z and 11 values of α equally spaced in $\alpha \in [2.5, 3.5]$. All even valued Z

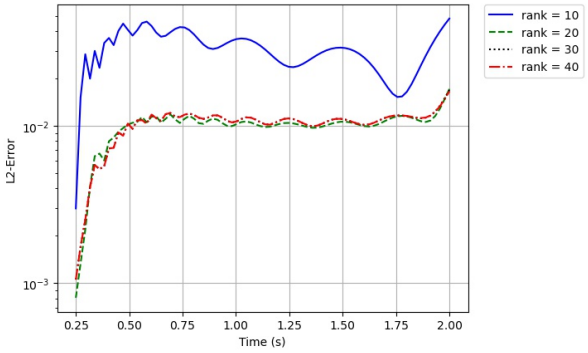
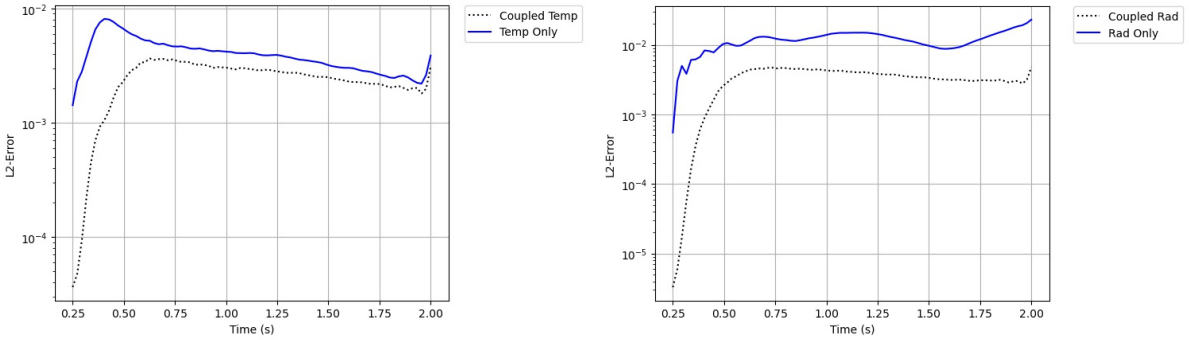


Figure 3.8: $\langle E(t_i) \rangle$ Error on a Multi-Parameter version of the Radiative Diffusion Problem

form the testing set as well as every α with an even tenths place. The set containing each pairing of Z and α testing points was used as the testing set and the set containing each paring of Z and α training points was used as the training set. Each point was evaluated using the same ending criteria as before, but with only using 5 testing points instead of the entire set and the results as a function of rank are shown in Fig. 3.8. Even though this problem has many more snapshots than the single parameter case the error is increased with a much higher lower error bound. The distances in 2-D space are larger than in 1-D parameter space making the distance between parameters for the interpolation required to get the same performance much smaller requiring more snapshots overall. This method is still able to get a reasonable error much faster than the full order model however, meaning that there is still reason to consider DMD even in multi-parameter cases.

Next the effects of coupling in this problem on parametric DMD will be studied. As a reminder this problem is a set of coupled differential equations in both E and T. In all previous parts the error was reported averaged over both E and T. In this part the error will be presented over only one portion of the solution at a time. In addition DMD will be performed taking the snapshot to include both E and T as in the previous parts and perform DMD independently using only E or only T as the snapshot to predict only E or only T respectively. In this problem only Z is varied as the parameter and $\alpha = 3.0$ is kept constant. The $Z \in [1, 15]$ with odd values will comprise the training set and even values will comprise the testing set. The same ending criteria as before are used and the results are plotted in in Fig. 3.9.



(a) Coupling Effects on Temperature Solution (b) Coupling Effects on Radiation Solution

Figure 3.9: $\langle E(t_i) \rangle$ Error of the rKOI Method in Single Physics Reconstruction Coupled and Uncoupled

The coupled DMD performs better at each time step than the DMD performed with only E or T as the snapshot for both temperature and radiation. The performance being better makes sense because in a coupled problem information about the other portion of the solution will be crucial in predicting behavior. Solving only one portion of the physics could be helpful when there is only information about one portion of a coupled system available.

4. SUMMARY AND CONCLUSIONS

In this thesis two novel methods for parametric DMD were presented and compared to the state of the art. The rEPI method as presented shows some obvious failures and is likely not worth further pursuit. However, the rKOI method performs exceptionally well achieving errors less than 1% on all problems studied. In addition the rKOI method displayed up to a 25% speed up of run times compared to the state of the art method while showing similar performance. In advection dominated problems it was demonstrated that the rKOI method can even out perform the state of the art stacked DMD method in terms of error on problems where the shared time eigen-value in stacked DMD has a larger effect.

The three parametric DMD methods were displayed on a set of problems that included a diffusion dominated problem, an advection dominated problem, and a problem with coupled physics showing the flexibility of the parametric DMD approach. It can easily be applied to a variety of problems and offers a large speed up compared to full order finite element solutions. The rKOI method was also shown to generate better results when parameter spacing was lowered meaning it might be possible to determine how many full order solutions must be generated in order for the rKOI method to ensure an upper bound on the reconstuction error over the parametric domain. It was also shown that the rKOI method can also be used, albeit slightly less effectively, in problems where there is more than one uncertain parameter. In addition it was also shown that the rKOI method can be used to predict a single variable in the solution of a coupled system containing multiple variables. This application of the rKOI method could be beneficial in situations where only one portion of the solution of a coupled system is available making it still possible to predict new parameter values without the entirety of the solution.

Future research could be done looking into improving the interpolation techniques used in the rKOI and rEPI methods potentially utilizing non-linear Grassmann manifold or matrix interpolation methods. Research could also be done applying the rKOI method to more parametric problems to validate its performance on a larger set of problems.

REFERENCES

[1] P. J. SCHMID, "Dynamic mode decomposition of numerical and experimental data," *Journal of Fluid Mechanics*, vol. 656, p. 5–28, 2010.

[2] J. H. Tu, , C. W. Rowley, D. M. Luchtenburg, S. L. Brunton, and J. N. K. and, "On dynamic mode decomposition: Theory and applications," *Journal of Computational Dynamics*, vol. 1, no. 2, pp. 391–421, 2014.

[3] J. N. Kutz, X. Fu, and S. L. Brunton, "Multiresolution dynamic mode decomposition," *SIAM Journal on Applied Dynamical Systems*, vol. 15, no. 2, pp. 713–735, 2016.

[4] N. Erichson, S. Brunton, and J. Kutz, "Compressed dynamic mode decomposition for background modeling," *Journal of Real-Time Image Processing*, vol. 16, 10 2019.

[5] S. Le Clainche and J. M. Vega, "Higher order dynamic mode decomposition," *SIAM Journal on Applied Dynamical Systems*, vol. 16, no. 2, pp. 882–925, 2017.

[6] M. R. Jovanović, P. J. Schmid, and J. W. Nichols, "Sparsity-promoting dynamic mode decomposition," *Physics of Fluids*, vol. 26, no. 2, p. 024103, 2014.

[7] T. Sayadi, P. J. Schmid, F. Richecoeur, and D. Durox, "Parametrized data-driven decomposition for bifurcation analysis, with application to thermo-acoustically unstable systems," *Physics of Fluids*, vol. 27, no. 3, p. 037102, 2015.

[8] R. G. McClarren and T. S. Haut, "Acceleration of source iteration using the dynamic mode decomposition," 2018.

[9] Z. K. Hardy, J. E. Morel, and C. Ahrens, "Dynamic mode decomposition for subcritical metal systems," *Nuclear Science and Engineering*, vol. 193, no. 11, pp. 1173–1185, 2019.

[10] R. G. McClarren, "Calculating time eigenvalues of the neutron transport equation with dynamic mode decomposition," *Nuclear Science and Engineering*, vol. 193, pp. 854–867, feb 2019.

[11] N. Demo, M. Tezzele, and G. Rozza, "PyDMD: Python Dynamic Mode Decomposition," *The Journal of Open Source Software*, vol. 3, no. 22, p. 530, 2018.

APPENDIX A

pyDMD CODE EXAMPLE

This appendix contains a code example for the python package pyDMD. This example assumes only that there exists a value n which will be the reconstruction rank of the DMD solution and that there is a data matrix in the shape $[m, t]$ where m is the resolution of the snapshot and t is the number of time steps in the time series of snapshots. The data provided can be either from experimental results or finite element methods as long as it is in the correct shape. The example code will take these two inputs and perform DMD, calculating the L_2 error of the reconstructed system and plotting the error over time.

```
#Import all modules
from pydmd import DMD
import matplotlib.pyplot as plt
import numpy as np
# Set paramaters and data
rank = n
solutions = data
num_steps = np.shape(data)[1]
# Initialize DMD object with reconstruction rank
dmd = DMD(svd_rank = r)
# .fit performs DMD and calculates reconstructed solution
dmd.fit(solutions)
DMD_solution = dmd.reconstructed_data
# Compare DMD solution to data
error = []
for j in range(1,num_steps):
  error.append(np.linalg.norm(solutions[:,j] - DMD_solution[:,j])\
              /np.linalg.norm(solutions[:,j]))
# Basic Plotting
plt.semilogy(error, label = 'rank = ' + str(rank))
plt.legend()
plt.grid()
plt.ylabel('L2-Error')
plt.xlabel('Step')
```