

Acceleration of the power and related methods with dynamic mode
decomposition

by

Leidong Xu

B.S., Kansas State University, 2017

A THESIS

submitted in partial fulfillment of the
requirements for the degree

MASTER OF SCIENCE

Department of Mechanical and Nuclear Engineering
Carl R. Ice College of Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

2019

Approved by:

Major Professor
Jeremy Roberts

Copyright

© Leidong Xu 2019.

Abstract

An algorithm based on dynamic mode decomposition (DMD) is presented for acceleration of the power method (PM) and flattened power method (FPM) that takes advantage of prediction from a restarted DMD process to correct an unconverged solution. The power method is a simple iterative scheme for determining the dominant eigenmode, and its variants, such as flattened power method, have long been used to solve the k-eigenvalue problem in reactor analysis. DMD is a data driven technique that extracts dynamics information from time-series data with which a reduced-order surrogate model can be constructed. DMD-accelerated PM (DMD-PM) and DMD-accelerated FPM (DMD-FPM) generate “snapshots” from a few iterations and extrapolate space in “fictitious time” to produce a more accurate estimate of the dominant mode. This process is repeated until the solution is converged to within a suitable tolerance. To illustrate the performance of both two schemes, a 1-D test problem designed to resemble a boiling water reactor (BWR) and the well-studied 2-D C5G7 benchmark were analyzed. Compared to the PM without acceleration, these tests have demonstrated that DMD-PM and DMD-FPM method can reduce the number of iterations significantly.

Table of Contents

| | |
|---|------|
| List of Figures | vi |
| List of Tables | viii |
| Acknowledgements | ix |
| Dedication | x |
| 1 Introduction | 1 |
| 1.1 Motivation | 1 |
| 1.2 Summary of previous work | 3 |
| 1.2.1 DMD Accelerated Iterative Methods | 5 |
| 1.3 Objective | 7 |
| 2 The MultiGroup Transport and Diffusion Equation | 9 |
| 2.1 Transport Theory | 9 |
| 2.2 Operator Notation | 11 |
| 2.3 Diffusion Theory | 12 |
| 3 The Power and Flattened-Power Methods | 14 |
| 3.1 The Power Method | 15 |
| 3.2 Convergence | 16 |
| 3.3 Application to the K-eigenvalue | 17 |
| 3.4 Flattened Power iteration | 18 |
| 4 Dynamic Mode Decomposition | 20 |

| | | |
|-------|--|----|
| 5 | The DMD-PM(n) Method and The DMD-FPM(n) Method | 24 |
| 5.1 | An Accelerated Power Method using DMD | 24 |
| 5.2 | The DMD-FPM(n) Method | 26 |
| 5.2.1 | Aitken Extrapolation | 26 |
| 6 | Results and Analysis | 29 |
| 6.1 | Test problem | 29 |
| 6.1.1 | Test problem for DMD-PM(n) | 29 |
| 6.1.2 | Test problem for DMD-FPM(n) | 31 |
| 6.2 | Results for DMD-PM(n) | 33 |
| 6.2.1 | Skipping Ahead with DMD-PM(n) | 33 |
| 6.2.2 | Application of Restarted DMD-PM(n) | 35 |
| 6.2.3 | Restarted DMD-PM(n) for Higher Modes | 37 |
| 6.3 | Results for DMD-FPM(n) | 40 |
| 6.3.1 | DMD-FPM(n) for 1D BWR Test Problem | 40 |
| 6.3.2 | DMD-FPM(n) for 2D C5G7 Test Problem | 40 |
| 7 | Conclusions and Future Work | 43 |
| 7.1 | Summary | 43 |
| 7.2 | Future Research | 44 |
| | Bibliography | 46 |

List of Figures

| | | |
|-----|---|----|
| 6.1 | Geometry as modeled for the IAEA 2-D diffusion benchmark. Material properties can be found in the benchmark documentation. Materials 0 and 1 are fuel, material 2 represents control, while material 3 represents the outer reflector. | 30 |
| 6.2 | Configuration for the BWR Test Problem | 31 |
| 6.3 | Configuration for the C5G7 benchmark. Each square represents the area of a 17×17 pin assembly | 32 |
| 6.4 | Configuration for a UO_2 fuel bundle. The green represents a UO_2 pincell, while the blue represents a guide tube modeled as a pincell filled with moderator . | 33 |
| 6.5 | Configuration for a MOX bundle. The light red represents 4.3% MOX fuel, the medium red represents 7.0 % MOX fuel, and the dark red represents 8.7% MOX fuel. The blue represents moderator (i.e., light water) | 34 |
| 6.6 | Configuration for pincell. The circular fuel element had a radius of 0.54 cm and was homogenized with cladding for this model. | 35 |
| 6.7 | Error in the DMD-predicted, dominant eigenmode as a function of iteration. The legend shows the number of power iterations n used to construct the surrogate, and in parentheses is the effective number of power iterations the DMD surrogate can produce. | 36 |
| 6.8 | The error in the predicted eigenmode for DMD-PM(n), where n is the number of power iterations performed. Errors are also included for the power method (PM) and Arnoldi's method. | 37 |
| 6.9 | First three reference eigenmodes; the corresponding eigenvalue ratios are shown above. | 38 |

| | | |
|------|---|----|
| 6.10 | Shown in the top row of each subfigure are the first three modes as computed from several applications of DMD-PM(30). The second row shows the error $\mathbf{e}_i = \mathbf{x}_i^{\text{reference}} - \mathbf{x}_i^{\text{approximate}}$. The corresponding $e_i \approx \lambda_i/\lambda_0$ (top row) and norm of the error $\ \mathbf{e}_i\ _2$ (bottom row) are also shown. | 39 |
| 6.11 | The absolute error in the predicted eigenmode for DMD-PM-aitken(n) for the 1D BWR problem, where n is the number of transport sweeps performed. . . | 41 |
| 6.12 | The absolute error in the predicted eigenmode for DMD-PM-Aitken(n) for the 2D C5G7 problem, where n is the number of transport sweeps performed. . . | 42 |

List of Tables

| | | |
|-----|-------------------------------------|----|
| 6.1 | Number of transport sweep | 41 |
|-----|-------------------------------------|----|

Acknowledgments

I would take some time to thank the numerous people who has helped me.

First, I thank my parents, Xu Fajian and Lv Wei, who works as two distinguished mechanical engineers pushed me to this great field.

I would like to thank to Professor Jeremy Roberts for offering me this great opportunity working on this project. He has served as my both undergraduate and graduate advisor with considerable patience since 2014.

I would like to thank Professor James Chen for leading me to numerical mathematics and encourage in me to challenge myself.

I would like to thank Professors Mingjun Wei and Hitesh Bindra for serving on my thesis committee and for answering any of my questions.

I would also like to thank two mentors and great friends, Dr. Mohammad Abdo and Richard Reed, for teaching me a great amount of expertise in nuclear engineering and data science.

I sincerely appreciate the help from my friends and colleagues at K-State - Cheikh, Dr. Wenkai Fu, Khalid, Dr. Luis Wonnell, Mohsen, Rabab, John, and Ye.

For the funding for this effort, I acknowledge the Alan Levin Department of Mechanical and Nuclear Engineering department for sponsoring me in the past two years.

Dedication

“I learned very early the difference between knowing the name of something and knowing something.”

Richard P. Feynman

Chapter 1

Introduction

Nuclear power was initially studied in the 1940s and has become an important source of energy worldwide. The nuclear reactor core containing the fuel is where the chain reactions take place. The amount of heat generation is proportional to the neutron population in the core, which itself is proportional to the number of fission events. Without somehow modeling the neutron population, it is impossible to understand the control of the neutronic systems and to generate suitable designs. In nuclear reactor physics, computational simulations play a critical role in the optimization of these designs and the prediction of the dynamics. However, explicit modeling a full reactor core remains too computationally expensive even with the most advanced, large-scale, supercomputers. Therefore, accurate and efficient approaches are needed to solve the various problems of reactor physics, including determining criticality and the neutron distribution.

1.1 Motivation

The generalized eigenvalue form of the neutron transport equation and its use for criticality analysis is central to computational reactor physics. The dominant eigenvector and corresponding eigenvalue of this equation can represent a system in steady-state conditions.

Though the criticality problem is a simplification of the time-dependent processes, it is often sufficient to reduce the transport equation into the steady-state form. The steady-state transport equation can be written in the generic form

$$\mathbf{A}\mathbf{x} = \frac{1}{k}\mathbf{B}\mathbf{x}, \quad (1.1)$$

where \mathbf{x} describes the scalar flux, \mathbf{A} represents neutron losses, \mathbf{B} represents the fission neutron gains from neutron, and the eigenvalue k represents the ratio of gains to losses. The so-called criticality of a reactor is determined by examining the dominant eigenvalue k , where $k = 1$ is “critical” (the neutron population holds steady), $k < 1$ is “subcritical” (the neutron density decreases over time), and $k > 1$ is “supercritical” (the neutron density increases over time). The eigenvector corresponding to the largest eigenvalue k is often called the fundamental (or dominant) eigenmode and corresponds to the scalar flux distribution when the system reaches a steady state. A more generalized multigroup neutron transport equation will be presented along with its diffusion approximation in Chapter 2.

The numerical solution of the transport equation requires the use of iterative techniques. A classical method for solving Eq. (1.1) is the power method, which requires the repeated application of $\mathbf{A}^{-1}\mathbf{B}$. The operator \mathbf{A}^{-1} represents the solution of inhomogeneous transport (or diffusion) equation which also often requires the use of iterative techniques, e.g., Richardson (or “source”) iteration and Gauss-Seidel iteration. Iterative techniques have traditionally dominated the transport community because the explicit construction of \mathbf{A} is prohibitively expensive with respect to both memory and processing costs. In other words, various iterative methods provide a way to solve the eigenvalue equation without forming a matrix inverse. The actual application of \mathbf{A} depends on the specific numerical method used (e.g., discrete ordinates or finite-volume diffusion), some details of which are described later. However, the cost of any methods is proportional to the number of applications of \mathbf{A} , and various techniques have been explored that greatly outperform the traditional methods. Of these, several have emerged from the broader numerical linear algebra community and include the

family of Krylov subspace methods, e.g., Generalized minimal residual method (GMRES)¹ for linear systems and generalized Davidson (GD)² for the generalized eigenvalue problem. Other methods are “physics driven” and include diffusion synthetic acceleration (DSA) for inhomogenous problems^{3;4}, and nonlinear diffusion acceleration methods like CMFD⁵.

These advanced methods work well but require specialized treatments of the transport (or diffusion) equations. As an alternative, data-driven techniques may be able to take as input a series of unconverged iterates from a classical iterative scheme (like the power method) and produce as output a greatly improved estimate for the converged solution. Dynamic mode decomposition (DMD) is one such approach that has emerged from the computational fluid dynamics community. This data-driven method can produce reduced-dimensional “surrogate” models by gathering modes directly from a sequence of states from some time-dependent process.

1.2 Summary of previous work

Before continuing, it is useful to review in brief past applications that demonstrated the utility of DMD. This technique was initially proposed by Schmid^{6,7} to extract dynamics information from time-series data of fluids observations. He applied DMD to both numerical Navier-Stokes code results and experimentally measured data and illustrated how DMD can identify coherent structure in the fluid dynamic system. As opposed to proper orthogonal decomposition (POD)⁸, DMD is a purely data-based procedure and does not project a higher-order system and equations on a reduced space. DMD was shown to be related to Koopman analysis^{9;10} of nonlinear dynamical systems and can be used to extract dynamic modes to describe the global spatiotemporal behavior. In Schmid’s original work, the time-dependent dynamic systems are described in the form of snapshots of an observable \mathbf{x}_i in time, or

$$\mathbf{X}_1^N = \{\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_N\}. \quad (1.2)$$

We assume that there exists a linear mapping operator \mathbf{A} that produces the snapshot sequence Eq. (1.2) in the form of the Krylov subspace when applied to the initial snapshot \mathbf{x}_0 repeatedly, i.e.,

$$\mathbf{X}_1^N = \{\mathbf{x}_0, A\mathbf{x}_1, A^2\mathbf{x}_1, \dots, A^{N-1}\mathbf{x}_1\}. \quad (1.3)$$

Schmid applied the singular value decomposition (SVD) to obtain the robust approximation

$$\tilde{\mathbf{A}} = \mathbf{U}^H \mathbf{X}_1^{N-1} \mathbf{V} \Sigma^{-1}, \quad (1.4)$$

where $\mathbf{X}_1^{N-1} = \mathbf{U} \Sigma \mathbf{V}^H$. The eigenvector \mathbf{x}_i of $\tilde{\mathbf{A}}$ and left eigenvector matrix \mathbf{U} are used to define the DMD modes,

$$\Psi_i = \mathbf{U} \mathbf{x}_i, \quad (1.5)$$

and, therefore, recover the reduced the mapping operator \mathbf{A} . This scheme is often referred to as the “standard” DMD approach in later research¹¹, and a more detailed discussion of the standard DMD scheme is presented in Chapter 4.

Many practical theories have been developed based on this original work of Schmid. Following Tu et al.^{11 12}, let

$$\mathbf{X}_0 \triangleq \{\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_{N-1}\}, \quad (1.6)$$

$$\mathbf{X}_1 \triangleq \{\mathbf{x}_1, \mathbf{x}_1, \dots, \mathbf{x}_N\}. \quad (1.7)$$

To proposed an “exact” DMD with which the operator \mathbf{A} can be acquired alternatively as

$$\mathbf{A} \triangleq \mathbf{X}_1 \mathbf{X}_0^\dagger, \quad (1.8)$$

where \mathbf{X}_0^\dagger is the pseudoinverse of \mathbf{X}_0 .

The DMD modes and eigenvalues can be found by a direct eigendecomposition; however, it may be too expensive in practice to construct the eigendecomposition of \mathbf{A} . Again, by use of

SVD, one can compute DMD modes or

$$\mathbf{x}_i = \frac{1}{\lambda} \mathbf{X}_1 \mathbf{V} \Sigma^{-1} \Phi. \quad (1.9)$$

Kutz et al.¹³ summarized many variations on the DMD algorithm and illustrated the applicability of each to several complex systems. Fundamental theoretical foundations of DMD and the Koopman operator were also developed in their monograph.

Recent efforts applied DMD to nuclear reactor simulations. For example, Abdo et al.¹⁴ used DMD as a direct, explicit-in-time surrogate for black-box models, e.g., to model the evolution of nuclear reactor isotopics over long time periods as well as the nonlinear response of reactor power during short transients.^{15 16}

1.2.1 DMD Accelerated Iterative Methods

As mentioned briefly, a way to achieve acceleration of iterative methods is to correct the solution estimated at each iteration to reduce the total number of iterations. One way to do this is to use the results from a lower-dimensional, projected system to predict the solution to the higher-dimensional system that we wish to solve.

Andersson and Eriksson¹⁷ first used DMD to accelerate the convergence of a time-dependent finite-volume solver for compressible flow to steady-state conditions. Their time-stepping method can be written in the form

$$\mathbf{x}_{n+1} = \mathbf{A} \mathbf{x}_n + \mathbf{b}, \quad (1.10)$$

where n denoted time. Here, difference of consecutive samples are used to define the snapshots matrix

$$\mathbf{V}_- = \{\mathbf{x}_2 - \mathbf{x}_1, \dots, \mathbf{x}_{n+1} - \mathbf{x}_n\}, \quad (1.11)$$

$$\mathbf{V}_+ = \{\mathbf{x}_3 - \mathbf{x}_2, \dots, \mathbf{x}_{n+2} - \mathbf{x}_{n+1}\}, \quad (1.12)$$

They defined the QR decomposition

$$\mathbf{V}_- = \mathbf{QR}, \quad (1.13)$$

with which

$$\mathbf{V}_+ = \mathbf{AV}_- = \mathbf{AQR}, \quad (1.14)$$

and, hence

$$\mathbf{Q}^T \mathbf{V}_+ = \mathbf{Q}^T \mathbf{AQR} = \tilde{\mathbf{A}}\mathbf{R}, \quad (1.15)$$

or

$$\tilde{\mathbf{A}} = \mathbf{Q}^T \mathbf{V}_+ \mathbf{R}^{-1}. \quad (1.16)$$

Because steady state implies $\mathbf{x}_{n+1} = \mathbf{x}_n$, one actually seeks \mathbf{x} such that $\mathbf{Ax} = \mathbf{B}$. Given an unconverged iterate \mathbf{x}_{n+1} define $\mathbf{V}_n = \mathbf{x}_{n+1} - \mathbf{x}_n$ and solve

$$(\mathbf{I} - \mathbf{A})\mathbf{V}_n + (\mathbf{I} - \mathbf{A})\mathbf{x}_{n+1} = \mathbf{b}, \quad (1.17)$$

or

$$(\mathbf{I} - \mathbf{A})\mathbf{V}_n = (\mathbf{x}_{n+2} - \mathbf{x}_{n+1}). \quad (1.18)$$

Now let $\mathbf{x}_n = \mathbf{Q}\mathbf{y}_n$. Then

$$\begin{aligned} \mathbf{Q}^T(\mathbf{I} - \mathbf{A})\mathbf{Q}\mathbf{y}_n &= (\mathbf{I} - \tilde{\mathbf{A}})\mathbf{y}_n \\ &= \mathbf{Q}^T(\mathbf{x}_{n+2} - \mathbf{x}_{n+1}). \end{aligned} \quad (1.19)$$

By noting $\mathbf{x}_{n+2}^{improved} \approx \mathbf{x}_{n+1} + \mathbf{Q}(\mathbf{I} - \tilde{\mathbf{A}})^{-1}\mathbf{Q}^T(\mathbf{x}_{n+2} - \mathbf{x}_{n+1})$. In other words, the solution is updated by solving a lower-dimensional problem. By performing this process several times to correct the solution, both the number of iterations and the average fluctuation were reduced almost 30% for compressible flow problems.

Then, McClarren and Haut¹⁸ presented an equivalent acceleration technique for improving the

convergence of Richardson(source) iteration for source-driven neutronics problems. Richardson iteration can be expressed as

$$\mathbf{x}^{(n+1)} = (\mathbf{I} - \mathbf{A})\mathbf{x}^{(n)} + \mathbf{b}. \quad (1.20)$$

Their algorithm also employed a set of successive differences $\mathbf{x}^{(n)} - \mathbf{x}^{(n-1)}$ to produce data matrices \mathbf{V}_+ and \mathbf{V}_- , while the standard DMD with SVD decomposition was applied to form the approximation $\tilde{\mathbf{A}}$. The algorithm is as follows

1. Perform R source iterations: $\mathbf{x}^l = \mathbf{A}\mathbf{x}^{l-1} + b$
2. Compute K source iterations to form \mathbf{V}_+ and \mathbf{V}_- . The last column of Y_- we call \mathbf{x}^{K-1}
3. Compute $\mathbf{x} = \mathbf{x}^{K-1} + \mathbf{U}\Delta y$.

where U is the left unitary matrix from SVD decomposition of \mathbf{V}_- and Δy is computed by

$$(\mathbf{I} - \tilde{\mathbf{A}})\Delta y = \mathbf{U}^T(\mathbf{x}^K - \mathbf{x}^{K-1}). \quad (1.21)$$

McClarren's results of a homogeneous slab problem and a multi-dimensional pipe problem suggest that a sequence of Richardson iterations followed by corrections reduces the number of iterations required by about one order of magnitude.

These past applications show that DMD has the potential to accelerated a wide variety of simple iterative methods, including the power method.

1.3 Objective

The primary focus of this thesis is to estimate accurate fundamental eigenmodes using DMD to accelerate the power method and other, related methods. Roberts et al.¹⁹ proposed a restarted, DMD-accelerated power method scheme (DMD-PM(n)), and, Xu et al.²⁰ then extended this theory to the flattened power method (or fixed-point iteration (FPI)). To

achieve these goals, we will expand this theory in certain ways in the following chapters. Chapter 2 will introduce the multigroup neutron transport and diffusion equations, and Chapter 3 discusses the mathematical background of the power and flatten power methods. Following that, DMD-PM(n) is presented in Chapter 4, while a DMD-based, accelerated, flattened power method DMD-FPM(n) is presented in Chapter 5. Then, the results using either of acceleration schemes on compute analyzing a 1-D boiling water reactor(BWR) model and the famous 2-D C5G7 test problem, are discussed in Chapter 6. Chapter 7 will include conclusions and future work including the possibility of combining this theory with other methods.

Chapter 2

The MultiGroup Transport and Diffusion Equation

This chapter contains a complete description of the multigroup transport and diffusion equations used to describe steady-state neutron systems, and, thus, provides a more detailed representation of the eigenvalue problem defined by Eq. (1.1).

2.1 Transport Theory

Neutron transport can be well modeled by linearization of the Boltzmann transport equation, which was initially used to describe the statistical behavior of particles in the dynamic thermal systems. This equation was developed and applied to determine the neutron distributions within the development of nuclear reactors as early as the 1940s. It is impossible to solve the full neutron transport equation analytically for any realistic, three-dimensional problems. Instead, approximations are made to simplify the often, intractable dependence of neutron cross sections on energy leading to the multigroup transport equation

$$\hat{\Omega} \cdot \nabla \psi_g(\mathbf{r}, \hat{\Omega}) + \Sigma_{tg}(\mathbf{r}) \psi_g(\mathbf{r}, \hat{\Omega}) = \frac{1}{4\pi} \sum_{g'=1}^{N_g} \Sigma_{sgg'}(\mathbf{r}) \phi_{g'}(\mathbf{r}) + \frac{\chi_g}{4\pi k} \sum_{g'=1}^{N_g} \nu \Sigma_{fg'}(\mathbf{r}) \phi_{g'}(\mathbf{r}) + s(\mathbf{r}, \hat{\Omega}), \quad (2.1)$$

where ϕ_g represents the scalar flux and ψ_g is the angular flux in the discretized energy interval (or “group”). Here, \mathbf{r} and $\hat{\Omega}$ indicate the position vector and angle of travel. In addition, $\Sigma_{t,g}$, $\Sigma_{s,g'g}$, and $\Sigma_{f,g}$ represent the group dependent cross sections for total, scattering, and fission reactions, respectively. Moreover, χ_g is the fission spectrum, ν is the average number of neutrons emitted per fission, and the k -eigenvalue (or “multiplication factor”) represents the balance of neutron gains (by fission) to losses (by absorption and leakage). As mentioned in the last chapter, the value of k indicates whether the reactor is critical, subcritical or supercritical.

There are two basic types of neutron transport problems: fixed-source problems and eigenvalue (criticality) problems. The fixed-source problems are solved to determine the neutron population distribution given a known, external neutron source and are common for shielding and detector applications’. On the other hand, the eigenvalue problem is used to describe the neutron population from a fission chain reaction, and, hence, are important for analysis of criticality in nuclear reactors and related systems. The k -eigenvalue problem mentioned in Chapter 1 is the most common type of eigenvalue problem. Both fixed-source and criticality problems can be solved using deterministic methods or stochastic methods. Based on our research attempts, we think the DMD-PM(n) method and DMD-FPM(n) should also be able to accelerate the Monte Carlo method. However, we only focus on using DMD to accelerate deterministic approaches to solving eigenvalue problems.

2.2 Operator Notation

The multigroup neutron transport Equation (2.1) can be defined in operator form, which is more convenient during numerical implementation. First, let a discrete-to-moment operator \mathbf{D} satisfy

$$\phi_g = \mathbf{D}\psi_g, \quad (2.2)$$

where the spatial dependence (continuous or discretized) is implicit. Also a moment-to-discrete operator \mathbf{M} satisfies

$$\psi_g = \mathbf{M}\phi_g. \quad (2.3)$$

Then, we can define the operator

$$\mathbf{L}_g(\cdot) = (\hat{\boldsymbol{\Omega}} \cdot \nabla + \Sigma_{tg}(\mathbf{r}))(\cdot), \quad (2.4)$$

With this the notation, the multigroup transport equation generalizes to²¹

$$\mathbf{L}_g\psi_g = \mathbf{M} \sum_{g'=1}^{N_g} (\mathbf{S}_{gg'} + \frac{1}{k} \mathbf{X}_g \mathbf{F}_{g'}) \phi_{g'} + q_g, \quad (2.5)$$

where $\mathbf{S} = \Sigma_s(\mathbf{r})$, \mathbf{F} represent the fission operator, \mathbf{X} represent the fission spectrum in operator form χ_g .

To simplify, we can define the space-angle transport sweep operator \mathbf{DL}^{-1} and multiply it on both side of Eq. (2.5), which leads to

$$\mathbf{D}\psi = \mathbf{DL}^{-1}\mathbf{MS}\phi + \frac{1}{k}\mathbf{DL}^{-1}\mathbf{MF}\phi, \quad (2.6)$$

In practice, we are only interested in the scalar flux, and the angular flux is rarely stored explicitly. Therefore, the transport equation can be represented using only the scalar flux by

substitution of Eq. (2.2) into Eq. (2.6), which yields

$$(\mathbf{I} - \mathbf{DL}^{-1}\mathbf{MS})\phi = \frac{1}{k}\mathbf{DL}^{-1}\mathbf{MF}\phi, \quad (2.7)$$

which is equivalent to Equation (1.1) with

$$\mathbf{T} = (\mathbf{I} - \mathbf{DL}^{-1}\mathbf{MS}), \quad (2.8)$$

and

$$\mathbf{B} = \mathbf{DL}^{-1}\mathbf{MF}. \quad (2.9)$$

2.3 Diffusion Theory

Neutron diffusion theory is sufficiently accurate for many reactor problems. This theory is simplified from neutron transport theory and can be formally derived by assuming the angular flux is at most linearly anisotropic and that the source, including external sources and fission sources, is isotropic.²²

We also assume that the source, including external source and fission source, is isotropic, and scattering is at most linearly anisotropic. However, we illustrate a more heuristic derivation. First, integrate both sides of Eq. (2.1) to obtain

$$\begin{aligned} \int_{\hat{\Omega}} [(\hat{\Omega} \cdot \nabla \psi_g(\mathbf{r}, \hat{\Omega}) + \Sigma_{tg}(\mathbf{r})\psi_g(\mathbf{r}, \hat{\Omega})] d\hat{\Omega} = \\ \int_{\hat{\Omega}} \left[\frac{1}{4\pi} \sum_{g'=1}^{N_g} \Sigma_{sgg'}(\mathbf{r})\phi_{g'}(\mathbf{r}) + \frac{\chi_g}{4\pi k} \sum_{g'=1}^{N_g} \nu\Sigma_{fg'}(\mathbf{r})\phi_{g'}(\mathbf{r}) \right] d\hat{\Omega} + s(\mathbf{r}, \hat{\Omega}). \end{aligned} \quad (2.10)$$

The neutron current is defined as

$$\mathbf{J} = \int_{4\pi} \hat{\Omega}\psi d\hat{\Omega}. \quad (2.11)$$

The right hand side can be treated as a single source term Q, which leads to the continuity

equation

$$\nabla \cdot \mathbf{J}_g + \Sigma_{tg}\phi_g = Q. \quad (2.12)$$

Substitution of Fick's law, i.e.,

$$\mathbf{J} = -D\nabla\phi, \quad (2.13)$$

into Eq. (2.12) leads to

$$\nabla \cdot -D\nabla\phi + \Sigma_t\phi = Q. \quad (2.14)$$

or

$$-\nabla \cdot D_g(r)\nabla\phi_g(\mathbf{r}) + \Sigma_{rg}(\mathbf{r})\phi_g(\mathbf{r}) = \sum_{g'=1}^{N_g} \Sigma_{sgg'}(\mathbf{r})\phi_{g'}(\mathbf{r}) + \frac{\chi_g}{k} \sum_{g'=1}^{N_g} \nu\Sigma_{fg'}(\mathbf{r})\phi_{g'}(\mathbf{r}), \quad (2.15)$$

where the group diffusion coefficient can be defined using

$$D_g(r) = \frac{1}{3\Sigma_t}. \quad (2.16)$$

or more accurate definitions. A two-group neutron diffusion Equation (6.1) is used in one of our numerical tests and discussed in Chapter 6.

Chapter 3

The Power and Flattened-Power Methods

This chapter contains a general description of the power method and the flattened-power method and describes their application to the k-eigenvalue neutron transport problem. All the eigenvalues and eigenvectors of a system matrix can be calculated by solving the characteristic equation,

$$\det(\mathbf{A} - \lambda\mathbf{I}) = 0, \quad (3.1)$$

where \mathbf{I} represents the identity matrix. However, the computational cost of solving this equation directly can be extremely high for large systems.

Iterative algorithms solve the eigenvalue problem by producing sequences that converge to the eigenvalues or the eigenvectors. In common applications, the eigenvalue sequences and eigenvector sequences are expressed as sequences of similar matrices. Those sequences will converge to a triangular or diagonal form, which reveal the eigenvalues directly.

Such iterative algorithms have been used for a variety of applications. Throughout scientific computing, some algorithms might not be applicable to general systems, but might be applied to hermitian or symmetric systems. On the other hand, the cost per iteration and the

convergence rates can also depend greatly on the problem, which could be more than an order of magnitude sometimes.

Some of those iterative algorithms can produce all the eigenpairs or eigenvalues. However, as mentioned in Chapter 1, only the largest eigenvalue and corresponding eigenvector are relevant in the k-eigenvalue transport problem. Although there are many iterative methods used for this type of eigenvalue problem, such as the QR algorithm, the Bisection method, and the Jacobi eigenvalue algorithm, the traditional power iteration is the simplest method to determine the fundamental mode and corresponding eigenvalue. This method goes by many names such as power iteration and Von Mises iteration²³. Moreover, some of the more advanced eigenvalue algorithms are in some sense variations of the power iteration; for example, Arnoldi iteration²⁴, like the power method, requires the repeated application of \mathbf{A} , but takes advantage of the whole Krylov subspace.

3.1 The Power Method

The power method is a simple algorithm for identifying the largest real eigenvalue of a matrix \mathbf{A} and its corresponding eigenvector. The basic algorithm is summarized in the following steps:

1. Let $\lambda_0^{(0)} = 1$ and $\mathbf{x}_0^{(0)}$ be a random, real vector normalized such the $\|\mathbf{x}_0^{(0)}\| = 1$.
2. Set $\mathbf{x}_0^{(i)} = \mathbf{A}\mathbf{x}_0^{(i-1)}$, where (i) represents the i th iteration.
3. Update $\lambda_i = k\|\mathbf{x}_0^{(i)}\|$ and set $\mathbf{x}_{(i)} = \frac{\mathbf{x}_i}{\|\mathbf{x}_i\|}$.
4. Repeat steps 2 and 3 for $i = 1, 2, \dots$ until $\|\mathbf{x}_{(i)} - \mathbf{x}_{(i-1)}\| < \tau$ for some tolerance τ .

Here the subscripts (i) indicates the iteration.

3.2 Convergence

The convergence rate is an important evaluation criteria on for comparing iterative method. In practice, the rate of convergence of the power method depends on the relative magnitudes of the leading eigenvalues. In other words, the usefulness of the power method depends upon the ratio $|\lambda_1|/|\lambda_0|$.

The initial guess $\mathbf{x}_0^{(0)}$ can be expressed as a sum of the weighted eigenvectors of \mathbf{A} , i.e.,

$$\begin{aligned}\mathbf{x}^{(0)} &= c'_0 \mathbf{x}_0 + c'_1 \mathbf{x}_1 + c'_2 \mathbf{x}_2 \dots \\ &= c'_0 \left(\mathbf{x}_0 + \frac{c'_1}{c'_0} \mathbf{x}_1 + \frac{c'_2}{c'_0} \mathbf{x}_2 \dots \right) \\ &= c'_0 (\mathbf{x}_0 + c_1 \mathbf{x}_1 + c_2 \mathbf{x}_2 \dots) .\end{aligned}\tag{3.2}$$

Because normalization of an eigenvector is arbitrary, let $c'_0 = 1$. Then, application of the operator \mathbf{A} to this initial guess leads to

$$\begin{aligned}\mathbf{A}\mathbf{x}^{(0)} &= \mathbf{A}\mathbf{x}_0 + c_1 \mathbf{A}\mathbf{x}_1 + c_2 \mathbf{A}\mathbf{x}_2 + \dots \\ &= \lambda_0 \mathbf{x}_0 + c_1 \lambda_1 \mathbf{x}_1 + c_2 \lambda_2 \mathbf{x}_2 + \dots \\ &= \lambda_0 \left(\mathbf{x}_0 + c_1 \frac{\lambda_1}{\lambda_0} \mathbf{x}_1 + c_2 \frac{\lambda_2}{\lambda_0} \mathbf{x}_2 + \dots \right) .\end{aligned}\tag{3.3}$$

Consequently, the repeated application of \mathbf{A} yields

$$\mathbf{A}^n \mathbf{x}^{(0)} = \lambda_0^n \left(\mathbf{x}_0 + c_1 \left(\frac{\lambda_1}{\lambda_0} \right)^n \mathbf{x}_1 + c_2 \left(\frac{\lambda_2}{\lambda_0} \right)^n \mathbf{x}_2 + \dots \right) ,\tag{3.4}$$

which shows that if $|\lambda_0| > |\lambda_1|$, then $\mathbf{A}^n \mathbf{x}_0^{(0)}$ will tend toward the direction \mathbf{x}_0 at a rate governed by the “dominance ratio” $|\lambda_1|/|\lambda_0|$. Because λ_0^n may grow without bound (or vanish), normalization is required during the iteration, as is included in the algorithm above.

As long as the fundamental mode and its corresponding eigenvalue are real and the initial

guess $\mathbf{x}^{(0)}$ is not perpendicular to the fundamental mode \mathbf{x}_0 (i.e., $\mathbf{x}_0^T \mathbf{x}^{(0)} \neq 0$), the power method will converge to the dominant eigenpair $(\mathbf{x}_0, \lambda_0)$.

3.3 Application to the K-eigenvalue

Consider again Eq. (1.1), the k-eigenvalue problem. Such a problem is a generalized eigenvalue problem. In order to apply the power method to this problem, one must set $\mathbf{A} = \mathbf{T}^{-1}\mathbf{F}$ and recognize $k_0 = \lambda_0$, the dominant eigenvalue shown in Chapter 2, the operator \mathbf{T} is the discrete-ordinates equation, while $\mathbf{T} = (\mathbf{I} - \mathbf{DL}^{-1}\mathbf{MS})$ when diffusion is applied. Because the matrix \mathbf{T} is readily constructed in the diffusion approximation, the application of \mathbf{T}^{-1} implies a straightforward converged solution of a linear system. However, for transport, the process is much more complicated, and the operator \mathbf{T} is rarely constructed in practice. Each application of \mathbf{T}^{-1} requires a complete solution of the inhomogeneous multigroup equation $\mathbf{T}\mathbf{x}^i = \mathbf{F}\mathbf{x}^{i-1}$. Because \mathbf{T} is not formed explicitly, iterative techniques based on the application of \mathbf{T} are needed. The number of transport sweeps (over space and angle) is a good measure for the total computational cost of a method because a sweep is the single most computationally expensive operation. To distinguish it from power iterations, the iterations required to solve the inhomogeneous equation needed to invert \mathbf{T} are called inner iterations. As a result, the power method leads to two, nested iteration levels: the outer iteration(eigeniteration) and

inner(source iteration). This algorithm is summarized as follows.

Algorithm 1: Power Method for K-eigenvalue Problem

Result: dominant eigenvalue and steady-state neutron flux

initialize scalar flux ϕ^0 and eigenvalue k ;

while *RHS not converged* **do**

 compute fission source ($b = \frac{1}{k}\mathbf{DL}^{-1}\mathbf{MF}\phi^{i-1}$);

while *LHS not converged* **do**

 compute scattering source ($\mathbf{T}\phi^i = b$);

end

 update eigenvalue k ;

 normalize eigenvector ϕ^i ;

end

3.4 Flattened Power iteration

The power method can be modified in many ways to improve the overall efficiency, and one possible way is achieved by varying the level of precision of the inner iterations. For example, by setting the inner tolerance to be proportional to the current outer residual or some other measure of the current level of error in the outer iteration, one reduces time spent solving the multigroup equation with unconverged fission sources. Another would be to fix the number of inner iterations for each outer iteration. Note that the scattering and fission sources are not completely converged in every outer iteration while either of the inner iteration strategies is used.²⁵

The outer iteration in the full power method often requires fewer and fewer inner iterations to converge when it gets close to the final solution, and, therefore, eventually just one inner iteration may be required the solution. This observation has lead to an important alternative often used in practice, called *flattened* power iteration (FPI), in which a single (space-angle-energy) transport sweep is performed for every power iteration. The flattened

power method converges the scattering and fission sources simultaneously.

The scattering matrix is moved to the right side of Equation 2.7, and, thus, step 2 from the full power iteration becomes

$$\mathbf{x}^{(i)} = \mathbf{DL}^{-1}\mathbf{M}(\mathbf{S} + \frac{1}{k}\mathbf{F})\mathbf{x}^{(i-1)}. \quad (3.5)$$

In this formulation, the inner iteration level is eliminated, by using a single sweep over all the phase-space variable when validating on the neutron flux from the previous iteration, Compared to the traditional power iteration, the computational cost of a single iteration in FPI in this form is obviously cheaper. Although this scheme may require more outer-most iterations (i.e., updates of k), the total cost of solving the k-eigenvalue problem is often reduced significantly.²⁵ Another great feature of this approach is easy to incorporate in existing power method implementation by limiting the number of inner iterations to one.

Chapter 4

Dynamic Mode Decomposition

Before introducing how to use dynamic mode decomposition to accelerate the power and flattened power methods, we will first review the details of DMD. As mentioned in Chapter 1, the details of DMD are different based on the application. However, most of these varieties share a familiar, straightforward frame. Here, the most widely-used variant (called “standard DMD” here) will be shown as an example to represent the algorithm.

To start, first consider the generic, dynamic problem defined by

$$\frac{d\mathbf{x}}{dt} = \mathbf{f}(\mathbf{x}, t), \quad (4.1)$$

where $\mathbf{x} \in \mathbb{R}^n$ is the n -dimensional state vector at time t . With sufficiently small steps in time, the evolution of \mathbf{x} can be well approximated by a relationship of the form

$$\frac{d\mathbf{x}(t)}{dt} = \mathcal{A}\mathbf{x}, \quad (4.2)$$

where the evolution operator \mathcal{A} may be unknown and can be considered a “black-box” system. However, one can obtain the system state \mathbf{x}_n at different times, which are then stacked as

the past and future snapshot matrices \mathbf{X}_0 and \mathbf{X}_1 , i.e.,

$$\mathbf{X}_0 = [\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_{m-1}] , \quad (4.3)$$

and

$$\mathbf{X}_1 = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_m] . \quad (4.4)$$

Suppose \mathbf{A} is the discrete-time approximation of the mapping operator \mathcal{A} :

$$\mathbf{A} = e^{A\Delta t} . \quad (4.5)$$

Then,

$$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k, \quad k = 0, 1, \dots, . \quad (4.6)$$

In general, the approximate operator \mathbf{A} not reproduce the \mathbf{x}_i exactly, but a “best” approximation can be formed in a least-squares or minimum-norm sense by solving

$$\mathbf{A} = \underset{\mathbf{A}}{\operatorname{argmin}} \|\mathbf{X}_1 - \mathbf{A}\mathbf{X}_0\|_F . \quad (4.7)$$

Thus, the best-fit operator \mathbf{A} is formally given by

$$A = \mathbf{X}_1 \mathbf{X}_0^\dagger , \quad (4.8)$$

where \mathbf{X}_0^\dagger is the Moore-Penrose generalized inverse of \mathbf{X}_0 . It is possible (and typical) to use SVD factorization to find the inverse of \mathbf{X}_0 by

$$\mathbf{X}_0 = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^* \rightarrow \mathbf{X}_0^\dagger = \mathbf{V}\mathbf{\Sigma}^{-1}\mathbf{U}^* , \quad (4.9)$$

where $\mathbf{U} \in \mathbb{C}^{m \times n}$, $\mathbf{V} \in \mathbb{C}^{n \times n}$, $\mathbf{\Sigma} \in \mathbb{C}^{n \times n}$, and $*$ indicate the conjugate transposes.

However, considering that the number of unknowns in this matrix is often large during

numerical simulations, the matrix \mathbf{A} is not computed explicitly. A low-rank approximation of the original dynamic system $\tilde{\mathbf{A}}$ is formed, i.e.,

$$\tilde{\mathbf{A}} = \mathbf{U}_r^* \mathbf{A} \mathbf{U}_r. \quad (4.10)$$

Then using Eq. (4.8), the reduced order $\tilde{\mathbf{A}}$ is defined by

$$\tilde{\mathbf{A}} = \mathbf{U}_r^* \mathbf{X}_1 \mathbf{V} \Sigma^{-1}. \quad (4.11)$$

Now extract the r largest eigenvalue and corresponding eigenvectors from $\tilde{\mathbf{A}}$ as the DMD modes Φ , which can be treated as the leading r eigenvectors of \mathbf{A} . Note that the solution of Eq. (4.2) is

$$\mathbf{x}(t) = e^{\mathbf{A}t} \mathbf{x}(0), \quad (4.12)$$

and \mathbf{A} is a discrete-time approximation of $e^{\mathbf{A}\Delta}$, which can be applied to the initial condition using the matrix exponential to compute the solution at a particular time. Moreover, the discrete eigenvalues λ_i of \mathbf{A} can be used to compute the continuous eigenvalues $\omega_i = \log(\lambda_i)/\Delta t$. Subsequently, the state can be reconstructed at any time t by a given initial condition

$$\mathbf{x}^{DMD}(t) \triangleq \sum_{i=1}^r \phi_i e^{\omega_i t} b_i, \quad (4.13)$$

where $\mathbf{b} = \Phi^\dagger \mathbf{x}_0$.

The general DMD scheme is summarized as

1. Compute SVD decomposition of the forward snapshots matrix, i.e., $\mathbf{X}_0 = \mathbf{U}_r \Sigma_r \mathbf{V}_r^*$, where r indicates the rank of matrix.
2. Compute $\tilde{\mathbf{A}} = \mathbf{U}_r^* \mathbf{A} \mathbf{U}_r = \mathbf{U}_r^* \mathbf{X}_1 \mathbf{V}_r \Sigma_r^{-1}$, where \mathbf{A} and $\tilde{\mathbf{A}}$ are similar matrices.
3. Compute the eigendecomposition $\tilde{\mathbf{A}} \tilde{\mathbf{W}} = \Lambda \tilde{\mathbf{W}}$.
4. Calculate the DMD modes as $\Phi = \mathbf{X}_1 \mathbf{V}_r \Sigma_r^{-1} \tilde{\mathbf{W}}$.

5. Predict the response by $\mathbf{x}^{DMD}(t) \approx \sum_{i=1}^r \phi_i e^{\omega_i t} \mathbf{b}_i = \Phi \mathbf{diag}(e^{\omega t}) \mathbf{b}$, where $\mathbf{b} = \Phi^\dagger \mathbf{x}_0$.

Chapter 5

The DMD-PM(n) Method and The DMD-FPM(n) Method

5.1 An Accelerated Power Method using DMD

The ultimate goal of this section is to develop a method that uses DMD to extrapolate eigenvectors and eigenvalues from a reasonably small number size of snapshots to produce an estimated eigenvector close to the final, steady-state solution. The difficult part of this strategy is that DMD itself needs to extract information from a time-dependent, dynamic system due to extrapolating the prediction results in the “future”. However, the power iteration does not recover the realistic physics transient process.

Note that because $\omega_i = \log(\lambda_i)$ while $\Delta = 1$, $x^{DMD}(t)$ is a “fictitious” time step corresponding to a single power iteration. First, suppose that m power iterations have been performed to produce the snapshot matrices \mathbf{X}_0 and \mathbf{X}_1 , where the series of snapshots are not ordered by the sequence of time but rather by the number of power iterations. We follow the standard DMD approach as discussed in Chapter 4 and generate leading r DMD modes and eigenvalues. Next, we can set t to a value much larger than the current number of snapshots employed

and predict a new \mathbf{x} , which can be normalized and applied as an initial guess for a new application of the power method.

Here, we will explore a modification from the original recovery scheme. The eigendecomposition of \mathbf{A} or the reduced-order approximation of $\tilde{\mathbf{A}}$ leads to a set of approximate eigenvectors and eigenvalues $e_j \approx \lambda_j/\lambda_0$. As previously discussed in Section 3.2, the convergence of the power method requires that λ_0 is larger than any other eigenvalues, thus

$$e_0 = 1 \quad \text{and} \quad e_j < 1, j = 1, 2, 3, 4 \dots \quad (5.1)$$

There are many strategies to select the optimal rank of DMD modes r used to fit the original system \mathbf{A} . Alternatively, because the power method can only reveal a single, dominant mode if the assumptions described in Chapter 3 are satisfied, only one mode recovered by DMD should remain at $t = \infty$, which is the fundamental eigenmode. This phenomenon can also be proven in a numerical perspective, as mentioned at Eq. (5.1), all other modes with eigenvalue smaller than one will vanish eventually when the diagonal matrix $e^{\omega t}$ is applied at $t = \infty$. Therefore, instead of computing all the DMD modes and predicting the response by Eq. (4.13), we can simply compute only the dominant DMD mode and predict the steady-state solution

$$\mathbf{x}^{DMD}(\infty) \approx \phi_0 \mathbf{b}_0, \quad (5.2)$$

where $b_0 = \phi_0^T \mathbf{x}_0$.

This simplification eliminates the noise from higher modes, which decreases the cost of reconstructing the future solution. In order to accelerate the power method with DMD, the following DMD-PM(n) algorithm is proposed:

1. Guess $\mathbf{x}^{(0)}$ and normalize.
2. Perform n power iterations to produce \mathbf{X}_0 and \mathbf{X}_1
3. Compute the DMD modes and frequencies using a rank- r , truncated SVD (i.e., $r < n$)

4. Apply Eq. (4.13) or Eq. (5.2) to estimate $\mathbf{x}^{(\infty)} = \mathbf{x}(\infty)$, i.e., estimate the steady-state, dominant mode after an equivalent of ∞ power iterations.
5. Set $\mathbf{x}^{(0)} = \Re(\mathbf{x}(\infty))/\|\mathbf{x}(\infty)\|$.
6. Repeat Steps 1 through 5 until converged.

By restarting the process, numerical errors caused by ill-conditioned snapshot matrices can be minimized. Stability analysis using either Eq. (4.13) or Eq. (5.2) and other numerical challenges are represent in Chapter 6. Note that the normalization in step 5 is important for reducing numerical round-off errors introduced by growing (or decaying) iterates.

5.2 The DMD-FPM(n) Method

As mentioned in Chapter 3, the flattened power method is a more efficient approach for solving the multigroup neutron transport equation. We have discussed a restarted, DMD-accelerated power method scheme, which suggests that there is potential to develop this algorithm to fit the flattened power method. This section contains a complete description of an accelerated flattened power method using DMD.

The basic framework of DMD-FPM(n) is similar to DMD-PM(n). The main difference is that the snapshots are now generated by the flattened power method. In short, a set of flattened power iterations are performed, then, DMD uses the snapshots to correct the dominant mode and eigenvalue, which is used to continue power iterations. The process can be repeated until the results converge, which leads to a *restarted* DMD-FPM (or DMD-FPM(n)).

5.2.1 Aitken Extrapolation

Note that not only is the eigenvector required at every restarting point but also the corresponding eigenvalue. The updated eigenvalue in the DMD-PM(n) algorithm is equal to the norm of the DMD predicted eigenmodes. One significant drawback of the flattened power

method is that we cannot compute the corresponding eigenvalue by the DMD response, because the eigenvalue has already been applied to the snapshots of neutron flux. In other words, there is no easy way to find the eigenvalue k associated with the predicted \mathbf{x} , which is projected forward in “time”. Therefore, there would be an inherent mismatch between the accelerated dominant eigenvector and its eigenvalue.

Many mathematical approaches were tested to extrapolate an eigenvalue to match the DMD eigenvector. A first attempt used the last computed eigenvalue, i.e., an eigenvalue that may be the equivalent of tens or hundreds of flattened power iterations in the “past.” However, the error caused by that choice of eigenvalue tended to reduce the improvement of the DMD extrapolation significantly, which erased all the improvement from DMD sometimes. Another attempt was to insert the eigenvalue as the first element in the snapshot, which did not work either. The reason might be that we modified the standard DMD algorithm, and only dominant modes were used. Some other failures include linear and polynomial extrapolation. In order to predict a more appropriate eigenvalue, Aitken extrapolation was employed²⁶ as

$$k_{aitken} = k_{i-2} - \frac{(k_{i-1} - k_{i-2})^2}{(k_i - 2k_{i-1} + k_{i-2})}, \quad (5.3)$$

where k_i is the eigenvalue from the i th iteration of the flattened power iteration. Although Aitken extrapolation does not eliminate the error from eigenvector/eigenvalue mismatch completely, significant improvement in all the numerical tests was observed.

The procedure for applying DMD to the flattened power iteration with Aitken extrapolation is summarized as follows.

1. Assume $k_{(0)}$, $\mathbf{x}_{(0)}$ and normalize.
2. Perform n flattened operator applications (Eq. (3.5)) to produce \mathbf{X}_0 and \mathbf{X}_1 .
3. Compute the DMD modes and frequencies using a rank- r , truncated SVD (i.e., $r < n$).
4. Apply equation $\mathbf{x}_0 = \frac{b_0 \phi_0}{\|b_0 \phi_0\|}$ to estimate $\mathbf{x}^{(\infty)} = \mathbf{x}(\infty)$, i.e., estimate the steady-state, dominant mode after an equivalent of ∞ power iterations.

5. Update $k_{(0)}$ by Aitken extrapolation.
6. Repeat Steps 1 through 5 until converged.

Both DMD-PM(n) and DMD-FPM(n) are tested to verify the performance to accelerate solving neutron transport/diffusion problem. The testing cases and numerical analysis are presented in the following chapter.

Chapter 6

Results and Analysis

This section contains a complete description of the test problems for modeling as well as results from numerical studies. To illustrate the performance of using the DMD-PM(n) method and the DMD-FPM(n) method, several test problems were analyzed. In all cases, the fundamental eigenmode was computed using a power method implementation as the benchmark and the snapshot generator for use with DMD.

6.1 Test problem

The test problems considered are (1) the 2-D, IAEA diffusion benchmark²⁷, (2) a 1-D, 70-pin BWR core model²⁸, and (3) the 2-D C5G7 benchmark²⁹. Problem(1) is used to test the DMD-PM(n) method, and Problem(2) Problem(3) are used to test the DMD-FPM(n) method.

6.1.1 Test problem for DMD-PM(n)

The well-known, two-dimensional (2-D) International Atomic Energy Agency (IAEA) diffusion benchmark was used to test the DMD acceleration algorithm proposed in Chapter 4. The

governing diffusion equations are

$$\begin{aligned}
 -\nabla D(\mathbf{r})_1 \nabla \phi_1(\mathbf{r}) + \Sigma_{r1}(\mathbf{r}) \phi_1(\mathbf{r}) &= \frac{1}{k} (\nu \Sigma_{f1} \phi_1(\mathbf{r}) + \nu \Sigma_{f2} \phi_2(\mathbf{r})) \\
 -\nabla D(\mathbf{r})_2 \nabla \phi_2(\mathbf{r}) + \Sigma_{a2}(\mathbf{r}) \phi_2(\mathbf{r}) &= \Sigma_{s1 \rightarrow 2} \phi_1(\mathbf{r}),
 \end{aligned}
 \tag{6.1}$$

where the notation defined in Chapter 2. All parameters including the cross sections are defined in the technical report published from Argonne National Laboratory. The basic core layout is shown in Figure 6.1, where the west and south boundary are reflective, and the north and east boundary are subject to vacuum conditions.

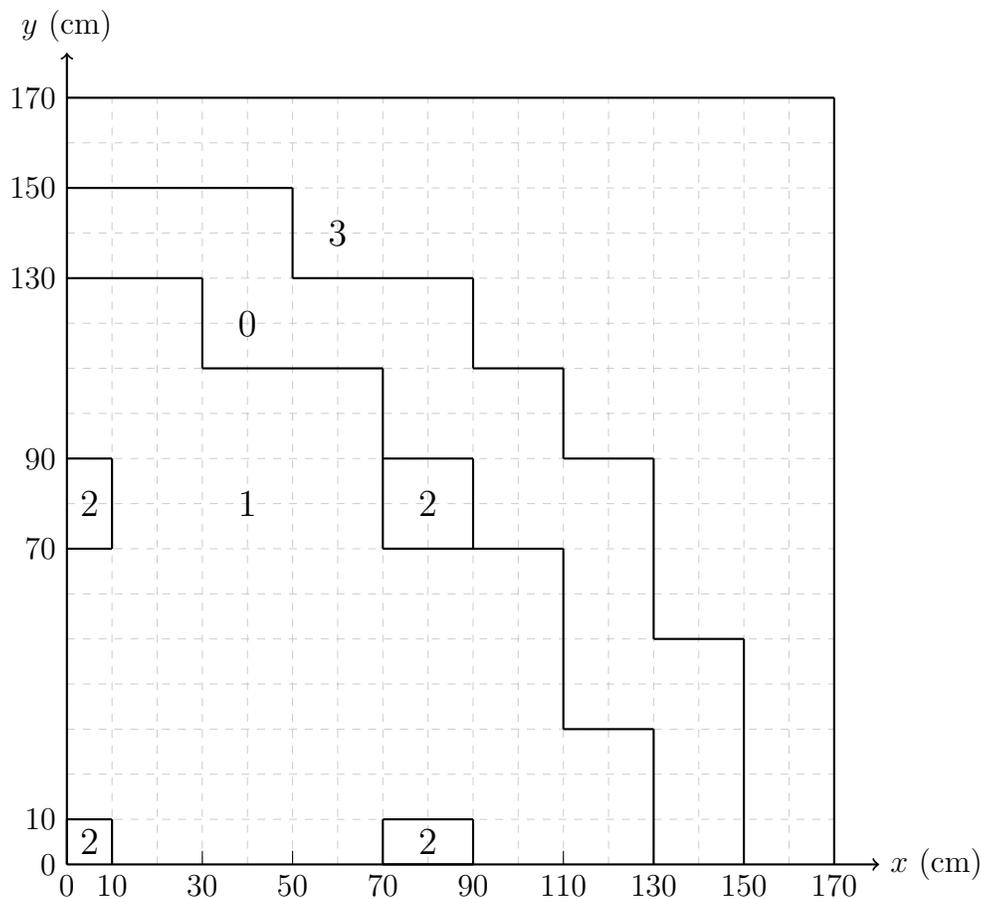


Figure 6.1: Geometry as modeled for the IAEA 2-D diffusion benchmark. Material properties can be found in the benchmark documentation. Materials 0 and 1 are fuel, material 2 represents control, while material 3 represents the outer reflector.

The mesh-center, finite-volume approximation was employed on a uniform, 45×45 spatial mesh. The discrete ordinates transport code DETRAN was used to generate the explicit

system matrix \mathbf{A} . Upon discretization, the entire set of equations was cast in terms of the fission source density, i.e., $\mathbf{f} = \nu\Sigma_{f1}\phi_1(\mathbf{r}) + \nu\Sigma_{f2}\phi_2(\mathbf{r})$, which results in a 2025×2025 operator. Hence, problem(1) is not large, but it proved to be a valuable test case for the method, and, therefore, ensured a reasonable computing time while debugging the code.

All calculations were initialized with a vector in which each element was sampled from the uniform distribution $U[0, 1]$. This randomized starting vector helps to ensure that all eigenmodes can be present. A formal sensitivity study was not performed to understand how this initial guess impacts the algorithm performance, but scoping studies suggest there is little impact on the number of iterations required for any particular algorithm. In this case, a reference solution was computed using the implicitly-restarted Arnoldi method as implemented in SciPy³⁰. All DMD calculations were performed using the Python package PyDMD³¹.

6.1.2 Test problem for DMD-FPM(n)

To illustrate the performance of this method, a simple 1-D test problem was designed to resemble a slab BWR core. This testing case was adapted from previous, transport applications from Rahnema et al.²⁸. The geometry of the 2-group BWR test case is shown in Fig. 6.2.

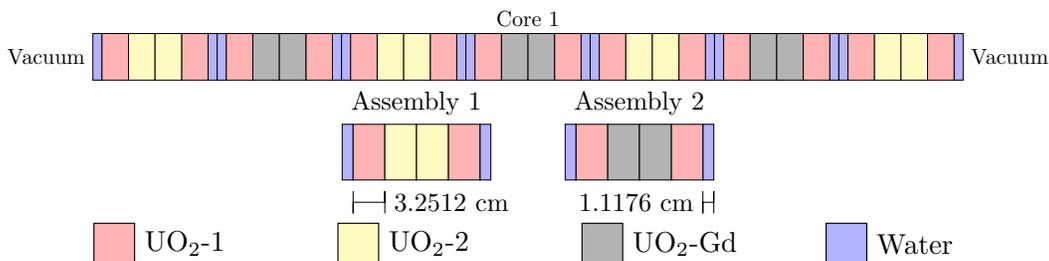


Figure 6.2: Configuration for the BWR Test Problem

This core configuration had two unique assemblies. Three fuel types were used, including 4.5 % enriched UO_2 , 2.5% enriched UO_2 , and 4.5 % enriched UO_2 with 5 wt% Gd_2O_3 . Fuel pins for this problem were 3.2512 cm thick with 1.1176 cm of a moderator on each side. The

baseline pincell discretization consisted of 18 mesh cells of fuel enclosed by six mesh cells of moderator; therefore, each pincell contained 30 mesh points. Boundary conditions on both sides for this case are subject to vacuum conditions.

The final test problem was the well-studied 2-D C5G7 benchmark²⁹, which was used to verify the performance of the algorithm for multi-dimensional problems. The configuration of the benchmark was adapted from Reed³². The configuration of a quarter core contains four fuel-pin assemblies and five moderator assemblies as shown in Figure 6.3. Each fuel assembly used 17×17 individual pincells, and the geometry of a UO_2 assembly is shown in Fig. 6.4, while that of a MOX assembly is shown in Fig. 6.5. Here, each pincell is discretized on a 7×7 Cartesian mesh. The dimensions are shown in Fig. 6.5.

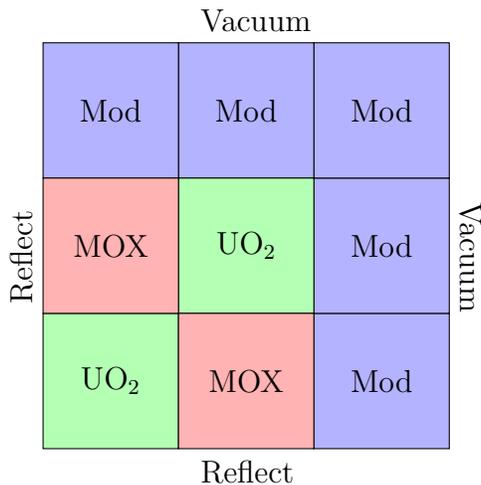


Figure 6.3: Configuration for the C5G7 benchmark. Each square represents the area of a 17×17 pin assembly

Here, the neutron transport equation for both 1-D and 2-D problems are solved by the discrete ordinates method, and all DMD calculations were performed using the PyDMD³¹. An S4 Gauss-Legendre quadrature was used with the diamond difference approximation. The reference eigenvalue and eigenvector were computed using full power iteration (i.e., fully converging the scattering source at each eigen iteration).

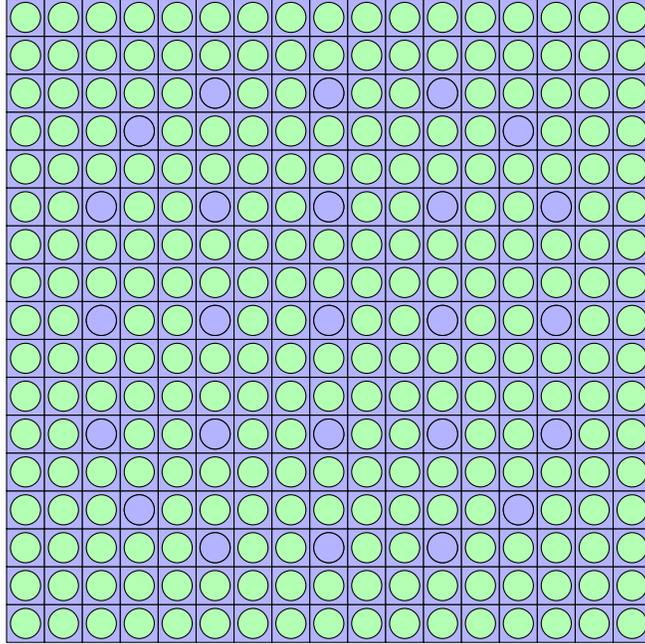


Figure 6.4: Configuration for a UO_2 fuel bundle. The green represents a UO_2 pincell, while the blue represents a guide tube modeled as a pincell filled with moderator

6.2 Results for DMD-PM(n)

As mentioned in the previous chapters, the major cost of the power method is from solving $\mathbf{Ax} = \mathbf{b}$. In this case, the computational time is proportional to the number of iterations. Thus, the number of iterations is used as the indicator of computational cost in the following comparisons.

6.2.1 Skipping Ahead with DMD-PM(n)

As a first test of the method of DMD-PM(n), the goal was to verify that DMD predicted eigenmodes prediction are closer to converged solutions. A series of n power iterations were performed to generate the snapshots, then the dominant eigenmode was reconstructed as a function of iteration using Eq. (4.13). The absolute error with respect to the reference

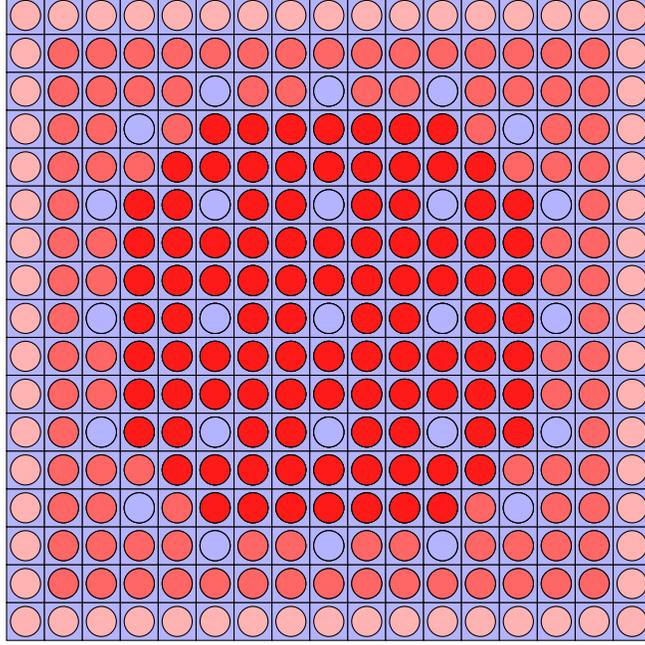


Figure 6.5: Configuration for a MOX bundle. The light red represents 4.3% MOX fuel, the medium red represents 7.0 % MOX fuel, and the dark red represents 8.7% MOX fuel. The blue represents moderator (i.e., light water)

eigenmode was computed as the euclidean norm of the differences, or

$$\|\mathbf{e}\| = \|\mathbf{x}_i^* - \mathbf{x}_{ref}\|_2. \quad (6.2)$$

The results shown in Figure 6.7 are predicted using different number n of snapshots where “time” is increasing. Shown in parentheses is the number of equivalent power iterations to which the final, saturated error in the DMD prediction corresponds. For example, the application of 30 power iterations leads to a DMD surrogate that can predict an eigenmode with an accuracy equal to 149 power iterations, a substantial skip ahead in the number of iterations.

The error is shown in Figure 6.7 approaches an asymptotic, lower bound as predictions are made beyond the number of power iterations used to generate the DMD surrogate. As expected, a larger snapshot matrix can provide more information for learning the function and mapping a more accurate output. Note that the final result reaching the lower bound

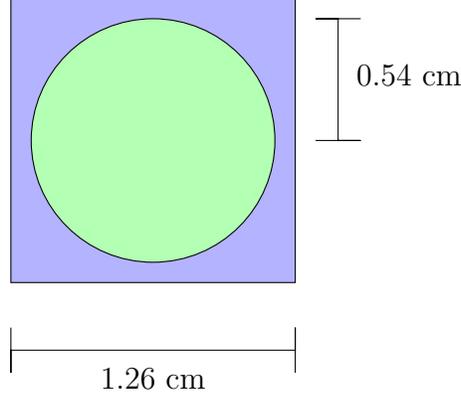


Figure 6.6: Configuration for pincell. The circular fuel element had a radius of 0.54 cm and was homogenized with cladding for this model.

here is very closed to the direct solution predicted by only the dominant DMD modes Φ_0 , which demonstrates that our modified Equation (5.2) can produce results with the same level of accuracy.

6.2.2 Application of Restarted DMD-PM(n)

As mentioned in Chapter 4, the DMD-PM(n) should restart from a set of snapshots multiple times until the solution converges. To test the performance of the iterative application of the DMD-PM(n) scheme, we used a fixed number n at every restart to research the convergence. The results are shown in Figure 6.8, which also includes the error for the unaccelerated PM and the Arnoldi method. Here, the Arnoldi method was used without restarts. The results shown for the Arnoldi method are as a function of the size of the subspace used.

Note that the restart value for Restarted DMD-PM(n) were selected as same as the skip ahead test. Although a larger restart value can often yield a better acceleration, storing a larger number of snapshots uses more memory and requires more operations during the SVD decomposition. Here, we consider each DMD extrapolation process as an iteration on the horizontal, and, therefore, the final points do not match exactly.

In this case, 10^{-14} is used as the tolerance for convergence. For reference, approximately 800 unaccelerated power iterations are required to reach this error. Ignoring the cost from DMD,

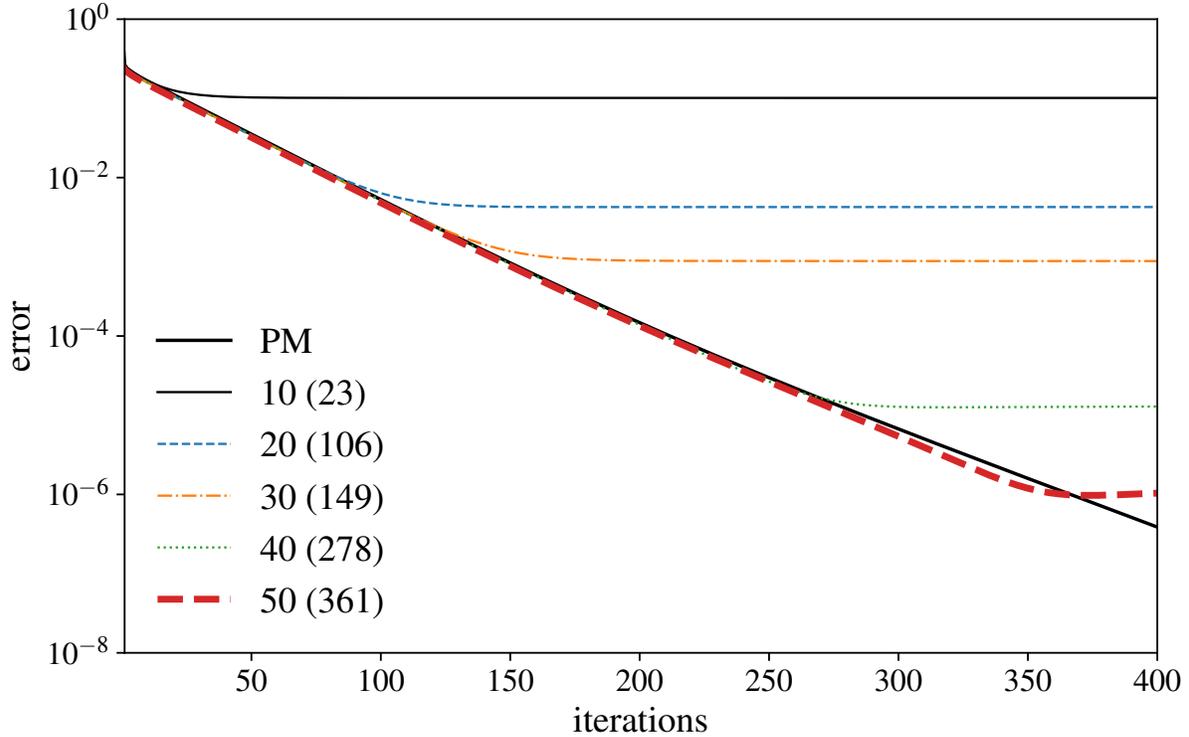


Figure 6.7: Error in the DMD-predicted, dominant eigenmode as a function of iteration. The legend shows the number of power iterations n used to construct the surrogate, and in parentheses is the effective number of power iterations the DMD surrogate can produce.

the best DMD-PM(50) only required around 150 iterations to reach the same error, thus providing ($5\times$) speedup compared to unaccelerated power iterations.

On the other hand, there is a significant performance difference between DMD-PM(n) and the Arnoldi method, which is also expected. The Arnoldi method only requires around 40 iterations to converge. The Arnoldi method is based on a subspace that undergoes continuous orthonormalization, which produces a better-conditioned and, likely, richer basis than can be produced by successive application of \mathbf{A} to a single vector. Consequently, it is inapplicable when the explicit form of \mathbf{A} is not available.

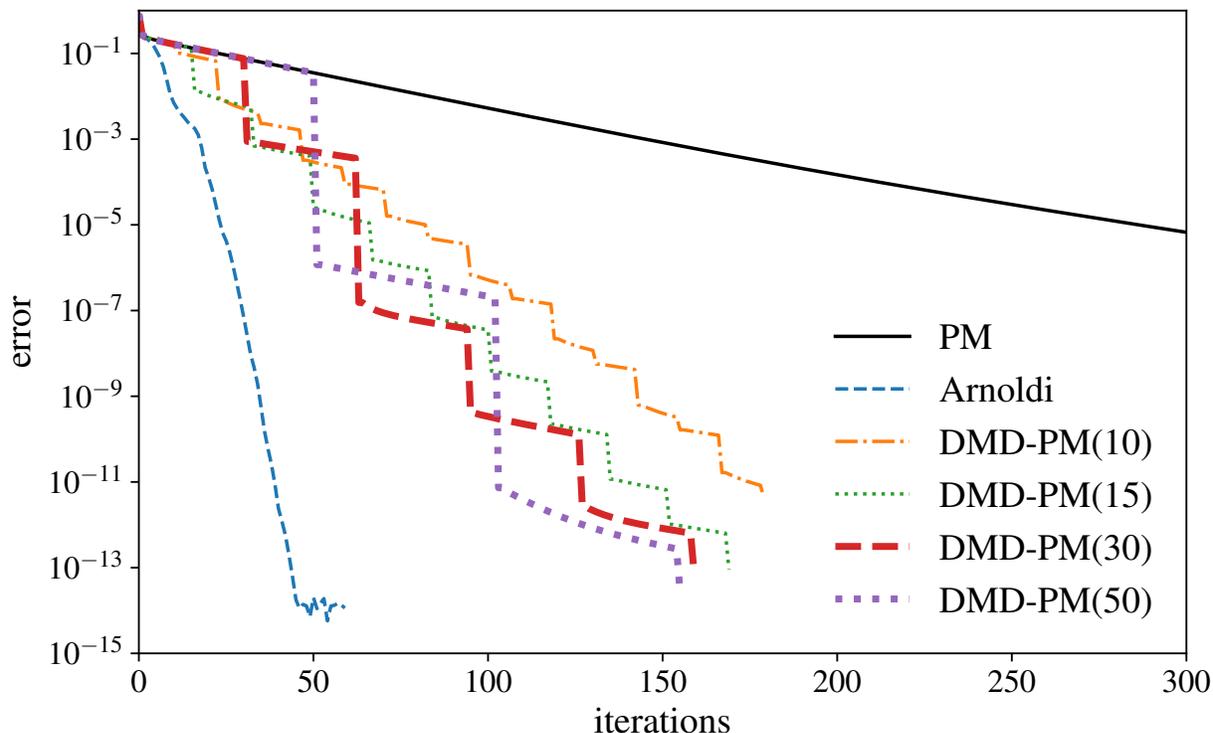


Figure 6.8: The error in the predicted eigenmode for DMD-PM(n), where n is the number of power iterations performed. Errors are also included for the power method (PM) and Arnoldi’s method.

6.2.3 Restarted DMD-PM(n) for Higher Modes

Note that DMD-PM(n) is based on construction of Krylov subspace, therefore, it should be able to recover the higher-order modes approximately. However, an unrestarted DMD-PM approximation produces to an ill-conditioned basis and, hence, cannot produce approximations for higher-order modes with reliable accuracy. Moreover, the iterative DMD-PM(n) using Eq. (5.2) throws away all higher-order modes upon the restart. Instead of using only the dominant DMD modes, the dominant mode was kept with a small contribution from the next two modes in order to capture the three modes with the largest eigenvalues. Specifically, the initial guess at each restarts was computed by $\mathbf{x}_0 + \epsilon(\mathbf{x}_1 + \mathbf{x}_2)$, where ϵ is a small value (here, 10^{-4}). Consequently, the next iteration contains some contribution from the higher-order space.

The reference modes are shown in Figure 6.9. The largest three eigenvalues and their

corresponding modes were recovered using the second, third, or fourth iterations of DMD-PM(30) as approximations for the first three eigenpairs of the original system. Errors in higher-order mode estimates were found to depend on the randomized initial guess for the first power iteration, and the representative values of the absolute errors in the DMD-PM(n) approximations are shown in Figure 6.10, where computed eigenvectors were normalized to unity.

As can be observed, the error in the dominant mode after two iterations (1.77×10^{-7}) is nearly unchanged from the case in which higher modes are not kept (1.54×10^{-7}); see Figure 6.8. However, the performance does degrade somewhat thereafter, with errors after three and four iterations of approximately 4.94×10^{-9} and 1.38×10^{-10} , respectively, compared to 9.90×10^{-11} and 3.00×10^{-12} in Figure 6.8.

The absolute errors in the two higher-order modes (and their eigenvalues) are much larger than the error for the dominant mode, and these errors are also decreasing with each iteration. Similarly, the third and fourth iteration also degrade for the two higher-order modes, especially the third largest eigenpairs. The error of the second order modes reach 1.28×10^{-3} after 3 iterations, and still provide fair improvement in the fourth iteration. Meanwhile, the error of the third order eigenpair only reaches 5.18×10^{-2} after 3 iterations and keep at the same level in the next iteration.

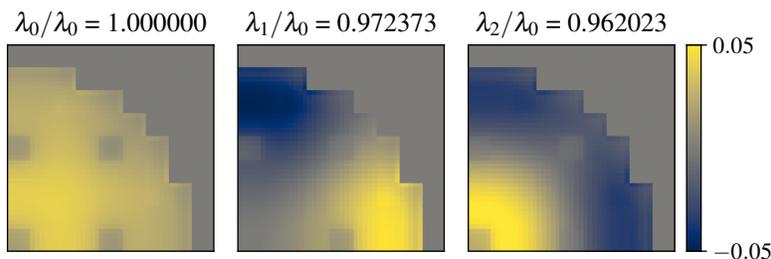
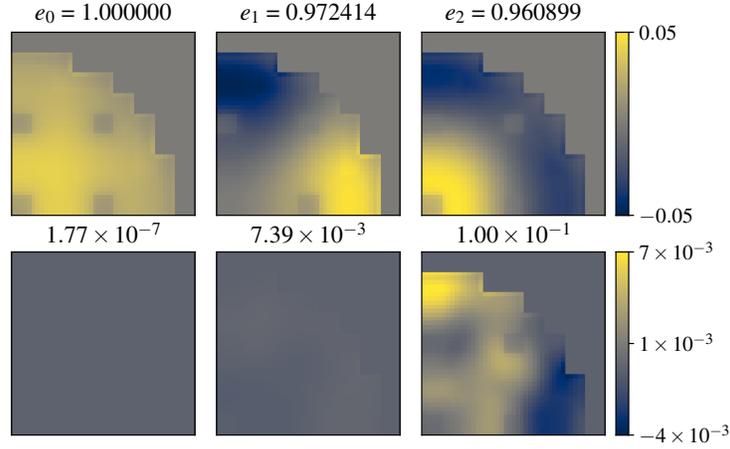
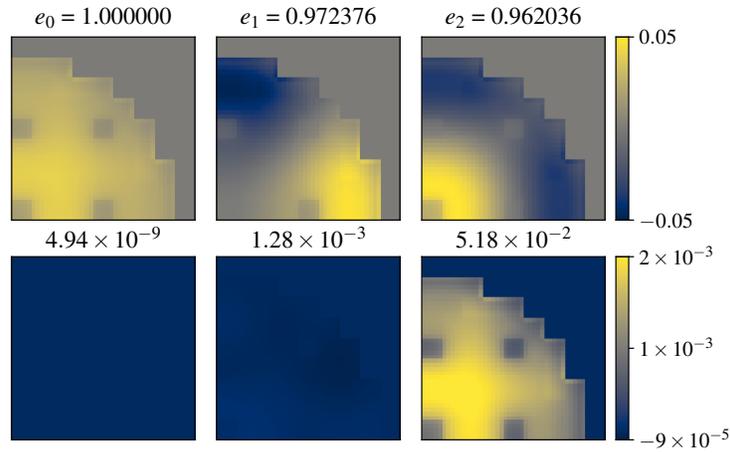


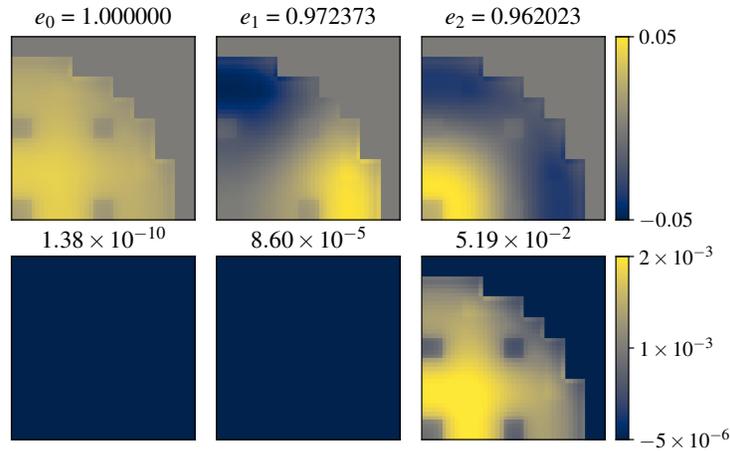
Figure 6.9: First three reference eigenmodes; the corresponding eigenvalue ratios are shown above.



(a) Two iterations of DMD-PI(30) with retention of 3 approximate modes.



(b) Three iterations of DMD-PI(30) with retention of 3 approximate modes.



(c) Four iterations of DMD-PI(30) with retention of 3 approximate modes.

Figure 6.10: Shown in the top row of each subfigure are the first three modes as computed from several applications of DMD-PM(30). The second row shows the error $\mathbf{e}_i = \mathbf{x}_i^{\text{reference}} - \mathbf{x}_i^{\text{approximate}}$. The corresponding $e_i \approx \lambda_i/\lambda_0$ (top row) and norm of the error $\|\mathbf{e}_i\|_2$ (bottom row) are also shown.

6.3 Results for DMD-FPM(n)

As mentioned in the Chapter 3, the flattened PM eliminates the inner iterations and only apply one transport sweep each iteration. Therefore, we used the number of sweeps as the measure for the total computational cost, which is equivalent to the number of iterations. Two similar tests on different geometries are conducted and solved for a variety of restart values n to study the optimum condition. The reference solutions were computed using the power method (i.e., fully converging the scattering source at each eigeniteration). The error is defined as the same as Equation (6.2), and the tolerance of convergence is set as 10^{-8} .

6.3.1 DMD-FPM(n) for 1D BWR Test Problem

To compare performance, the generalized k -eigenvalue problem was solved first by flattened power iteration, which used 3445 transport sweeps for this BWR problem. The best DMD-FPM(n) algorithm used 40 snapshots, and required approximately 270 transport sweeps, thus providing more than an order of magnitude reduction in the computational cost.

This case shows that using a larger number of snapshots cannot promise a faster convergence for DMD-FPM(n). The reason might be the error remaining from mismatched eigenvalues.

6.3.2 DMD-FPM(n) for 2D C5G7 Test Problem

Similar to the 1-D results, the results of C5G7 2-D benchmark also show that the best condition of DMD-FPM(n) algorithm has significant speedup compared to flattened power iteration, which required 351 transport sweep to reduce error to $1e-8$ while using 30 snapshots each restart. For reference, approximately 1570 unaccelerated flattened power sweeps are required for this problem to be fully converged. As expected in this case, a small increase in error may be observed more obviously for each application of DMD-FPM(n) due to the previously discussed eigenvalue error. And the best case is not using the most number of

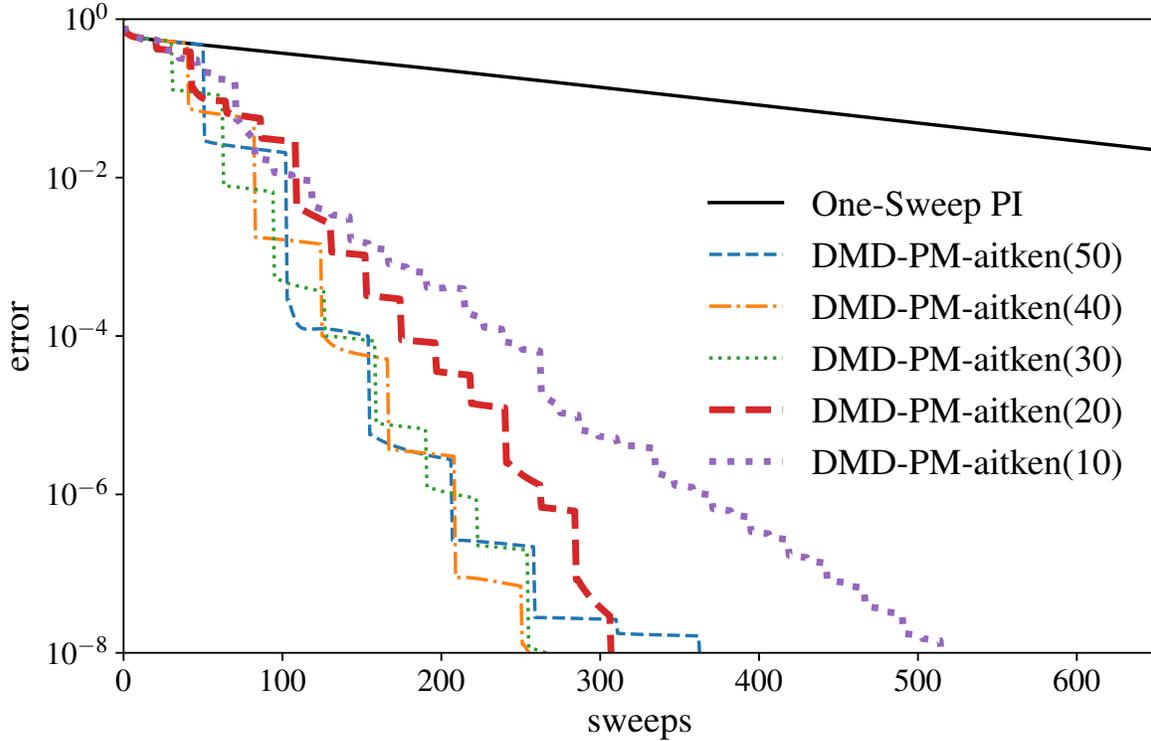


Figure 6.11: The absolute error in the predicted eigenmode for DMD-PM-aitken(n) for the 1D BWR problem, where n is the number of transport sweeps performed.

snapshots, though the difference of sweeps is relatively small.

Using DMD too frequently (i.e., small n) might produce a large numerical error from SVD decomposition, the error increase many levels of magnitude from DMD. This is the reason why DMD-FPM(10) did not reduce the error to within the target range. The total numbers of transport sweeps required to reach a tolerance of 10^{-8} are shown in Table 6.1.

Table 6.1: Number of transport sweep

| | FPM | DMD-FPM(10) | DMD-FPM(20) | DMD-FPM(30) | DMD-FPM(40) | DMD-FPM(50) |
|-------------|------|-------------|-------------|-------------|-------------|-------------|
| <i>BWR</i> | 3444 | 515 | 307 | 267 | 256 | 363 |
| <i>C5G7</i> | 1570 | N/A | 395 | 351 | 377 | 395 |

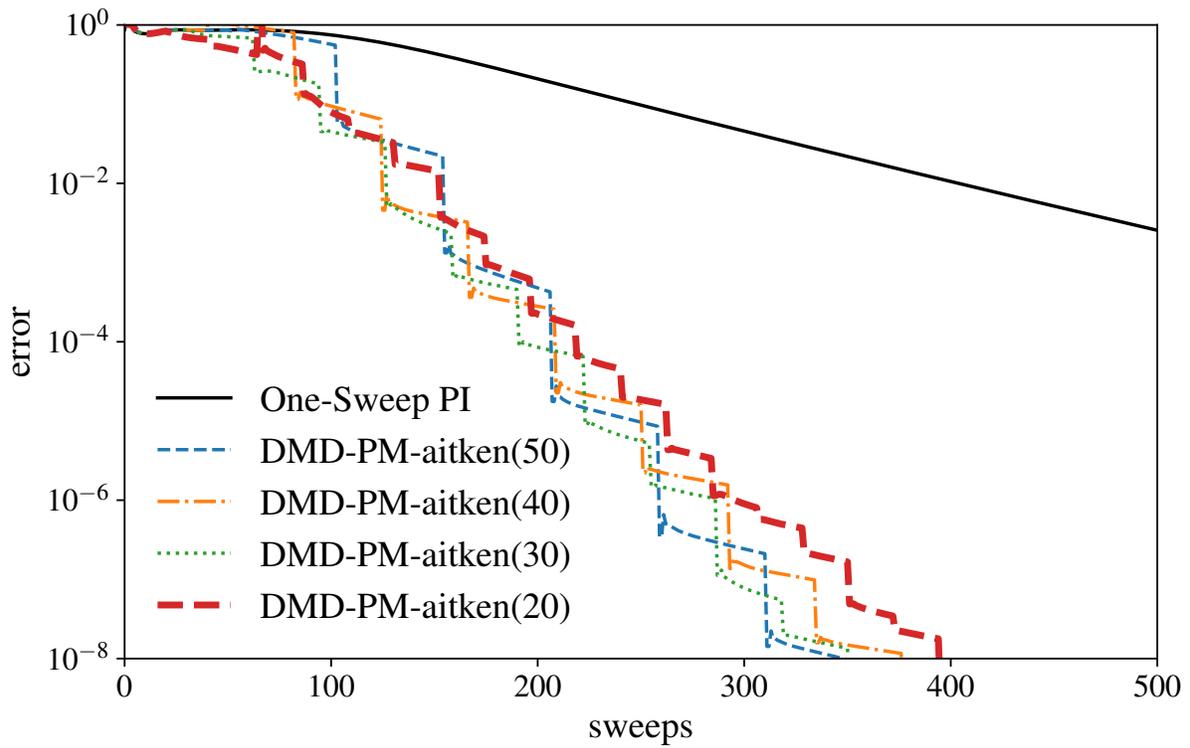


Figure 6.12: The absolute error in the predicted eigenmode for DMD-PM-Aitken(n) for the 2D C5G7 problem, where n is the number of transport sweeps performed.

Chapter 7

Conclusions and Future Work

7.1 Summary

First, the principal objective of this thesis was to accurately estimate fundamental eigenmodes by using DMD to accelerate the power method and flattened power method for multigroup neutron transport/diffusion problems. Although DMD is a useful tool for extracting information from data, often it can only be applied on the snapshots from the time-dependent dynamic systems. In Chapter 5, we have presented a new scheme for identifying fundamental eigenvector by an improved DMD algorithm using only the dominant DMD modes, which allows us to correct the solution by a more accurate estimation of the steady solution. This restarted version DMD-PM(n) can be applied repeatedly to accelerate the power method.

Flattened iterations update the fission and scattering source at the same time and reduced the total number of transport sweeps in practice, and, thus, are widely used as an more efficient alternative of the power method. Therefore, we also explored a similar scheme DMD-FPM(n) to improve the convergence rate of the flattened power method. Because the eigenvalue cannot be computed by the eigenvector from flattened operator, the Aitken method is employed to extrapolate the eigenvalues corresponding to the DMD prediction.

For the comparison, three test problems were conducted, which include (1) a 2-D, IAEA diffusion benchmark, (2) a 1-D, 70-pin BWR core model, and (3) the 2-D C5G7 benchmark. Through all the numerical examples, we demonstrated that both acceleration schemes provide promising speedup. The choice of the number of snapshots to DMD greatly impacts the effectiveness, the DMD-PM(50) case used only 25% number of power iterations to solve the IAEA diffusion problem. This scheme has also been used to produce an approximation of higher-order modes. Unfortunately, the accuracy of the two higher-order modes is not as good as the dominant mode. Although DMD-PM(n) might not be competitive with other advance acceleration schemes (e.g., the Arnold method), there do exist applications for which access to iterates is only available in a postprocessing sense. As can be expected, DMD-FPM(n) provided approximately a 5x–10x speedup for the two cases studied. However, the failure of DMD-FPM(10) in solving C5G7 case indicate that DMD might produce a large numerical error from SVD when the results are closed to the steady-state solution because the snapshots are linearly dependent. In this case, the DMD should be stopped, then use only the power or flattened power method to reduce the error to within the target range.

7.2 Future Research

In this section, we describe the future direction and substantial value of using these methods in other areas. While these results are promising, the performance of both schemes is not expected to outperform some other popular methods, such as the generalized Davidson method³³ and coarse-mesh finite difference⁵. In reactor analysis, the use of stochastic methods (i.e. Monte Carlo simulations) is widespread. Following this thesis, the application of DMD-PM(n) and DMD-FPM(n) may be able to accelerate Monte Carlo eigenvalue problems for convergence by regressing the distribution tendency of neutron population from DMD modes. In this way, only a small size of neutron populations are sufficient to generate snapshots and extract information, which might be comparable to the other acceleration methods.

Although Aitken extrapolation could estimate the eigenvalues corresponding to DMD re-

sponses, errors still exist at almost every restart point, and, therefore, reducing the desired accuracy. More work can also be done in the future to compute the corresponding eigenvalues, which can improve the performance while a great amount of restarted process is required in the large scale systems.

Bibliography

- [1] Youcef Saad and Martin H Schultz. Gmres: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM Journal on scientific and statistical computing*, 7(3):856–869, 1986. URL <https://epubs.siam.org/doi/abs/10.1137/0907058>.
- [2] Steven Paul Hamilton. *Numerical Solution of the k-Eigenvalue Problem*. PhD thesis, Emory University, 2007. URL <https://etd.library.emory.edu/concern/etds/j6731409k?locale=en>.
- [3] R. E. Alcouffe. Diffusion synthetic acceleration methods for the diamond-differenced discrete-ordinates equations. *Nuclear Science and Engineering*, 64(2):344–355, 1977. doi: 10.13182/NSE77-1. URL <https://doi.org/10.13182/NSE77-1>.
- [4] Jeremy A. Roberts and Benoit Forget. Multigroup diffusion preconditioners for multiplying fixed-source transport problems. *Journal of Computational Physics*, 274: 455–472, October 2014. ISSN 00219991. doi: 10.1016/j.jcp.2014.06.034. URL <https://linkinghub.elsevier.com/retrieve/pii/S0021999114004410>.
- [5] K. S. SMITH. Nodal method storage reduction by nonlinear iteration. *Trans. Am. Nucl. Soc.*, 44:265–266, 1983. URL <https://ci.nii.ac.jp/naid/80002095029/en/>.
- [6] Peter J. Schmid. Dynamic mode decomposition of numerical and experimental data. *Journal of Fluid Mechanics*, 656:5–28, August 2010. ISSN 0022-1120, 1469-7645. doi: 10.1017/S0022112010001217. URL http://www.journals.cambridge.org/abstract_S0022112010001217.
- [7] P. J. Schmid, L. Li, M. P. Juniper, and O. Pust. Applications of the dynamic mode decomposition. *Theoretical and Computational Fluid Dynamics*, 25(1-4):249–259, June

2011. ISSN 0935-4964, 1432-2250. doi: 10.1007/s00162-010-0203-9. URL <http://link.springer.com/10.1007/s00162-010-0203-9>.
- [8] John L Lumley. *Stochastic tools in turbulence*. Courier Corporation, 2007.
- [9] Andrzej Lasota and Michael C Mackey. *Chaos, fractals, and noise: stochastic aspects of dynamics*, volume 97. Springer Science & Business Media, 2013.
- [10] Igor Mezić. Spectral properties of dynamical systems, model reduction and decompositions. *Nonlinear Dynamics*, 41(1-3):309–325, 2005.
- [11] Jonathan H. Tu, Clarence W. Rowley, Dirk M. Luchtenburg, Steven L. Brunton, and J. Nathan Kutz. On Dynamic Mode Decomposition: Theory and Applications. *Journal of Computational Dynamics*, 1(2):391–421, December 2014. ISSN 2158-2491. doi: 10.3934/jcd.2014.1.391. URL <http://arxiv.org/abs/1312.0041>. arXiv: 1312.0041.
- [12] Tu. (*Tu's dissertation*) *Dynamic Mode Decomposition: Theory and Applications*. PhD thesis, Princeton University, 2014. URL <http://crowley.princeton.edu/theses/tu.pdf>.
- [13] J. Kutz, S. Brunton, B. Brunton, and J. Proctor. *Dynamic Mode Decomposition*. Other Titles in Applied Mathematics. Society for Industrial and Applied Mathematics, November 2016. ISBN 978-1-61197-449-2. doi: 10.1137/1.9781611974508. URL <https://epubs.siam.org/doi/book/10.1137/1.9781611974508>.
- [14] Mohammad Abdo, Rabab Elzohery, and Jeremy Roberts. DATA-DRIVEN SURROGATE MODEL TO PREDICT ISOTOPIC COMPOSITION USING DYNAMIC MODE DECOMPOSITION. *PHYSOR 2018: Reactor physics paving the way towards more efficient systems*, page 12, 2018.
- [15] Mohammad Abdo, Rabab Elzohery, and Jeremy A. Roberts. Modeling isotopic evolution with surrogates based on dynamic mode decomposition. *Annals of Nuclear Energy*, 129:280–288, July 2019. ISSN 03064549. doi: 10.1016/j.anucene.2019.01.048. URL <https://linkinghub.elsevier.com/retrieve/pii/S030645491930060X>.

- [16] R Elzohery, M Abdo, and J Roberts. Comparison between gaussian processes and dmd surrogates for isotopic composition prediction. *Transactions of the American Nuclear Society*, 118:459–462, 2018.
- [17] N Andersson and L Eriksson. A Novel Solver Acceleration Technique Based on Dynamic Mode Decomposition. *Conference: 6th European Conference on Computational Fluid Dynamics (ECFD VI)*, page 20, December 2013.
- [18] Ryan G. McClarren and Terry S. Haut. Acceleration of Source Iteration using the Dynamic Mode Decomposition. *arXiv:1812.05241 [physics]*, December 2018. URL <http://arxiv.org/abs/1812.05241>. arXiv: 1812.05241.
- [19] Jeremy A. Roberts, Leidong Xu, Rabab Elzohery, and Mohammad Abdo. Acceleration of the power method with dynamic mode decomposition. *Nuclear Science and Engineering*, 193(12):1371–1378, 2019. doi: 10.1080/00295639.2019.1634928. URL <https://doi.org/10.1080/00295639.2019.1634928>.
- [20] Leidong Xu, Richard L Reed, and Jeremy A Roberts. ACCELERATION OF THE FLATTENED POWER METHOD WITH DYNAMIC MODE DECOMPOSITION. *Transactions of the American Nuclear Society*, 2019.
- [21] R. N. Slaybaugh, T. M. Evans, G. G. Davidson, and P. P. H. Wilson. Multigrid in energy preconditioner for Krylov solvers. *Journal of Computational Physics*, 242:405 – 419, 2013. ISSN 0021-9991. doi: <https://doi.org/10.1016/j.jcp.2013.02.012>. URL <http://www.sciencedirect.com/science/article/pii/S0021999113001228>.
- [22] Weston M Stacey. *Nuclear reactor physics*. John Wiley & Sons, 2018.
- [23] RV Mises and Hilda Pollaczek-Geiringer. Praktische verfahren der gleichungsauflösung. *ZAMM-Journal of Applied Mathematics and Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik*, 9(2):152–164, 1929.
- [24] Walter Edwin Arnoldi. The principle of minimized iterations in the solution of the matrix eigenvalue problem. *Quarterly of applied mathematics*, 9(1):17–29, 1951.

- [25] Daniel F. Gill, Yousry Y. Azmy, James S. Warsa, and Jeffery D. Densmore. Newtons Method for the Computation of k -Eigenvalues in S_n Transport Applications. *Nuclear Science and Engineering*, 168(1):37–58, May 2011. ISSN 0029-5639, 1943-748X. doi: 10.13182/NSE10-01. URL <https://www.tandfonline.com/doi/full/10.13182/NSE10-01>.
- [26] A. C. Aitken. Xxv.—on bernoulli’s numerical solution of algebraic equations. *Proceedings of the Royal Society of Edinburgh*, 46:289–305, 1927. doi: 10.1017/S0370164600022070.
- [27] Argonne Code Center. Benchmark problem book, report anl-7416 (suppl. 2). Technical report, Argonne National Laboratory, Argonne, IL, 1977.
- [28] Farzad Rahnema, Steven Douglass, and Benoit Forget. Generalized Energy Condensation Theory. *Nuclear Science and Engineering*, 160(1):41–58, September 2008. ISSN 0029-5639, 1943-748X. doi: 10.13182/NSE160-41. URL <https://www.tandfonline.com/doi/full/10.13182/NSE160-41>.
- [29] OECD Nuclear Energy Agency. *Benchmark on deterministic transport calculations without spatial homogenisation: a 2-D/3-D MOX fuel assembly benchmark*. Nuclear Energy Agency, Organisation for Economic Cooperation and Development, Issy-les-Moulineaux, France, 2003. ISBN 978-92-64-02139-6. OCLC: 61895507.
- [30] Eric Jones, Travis Oliphant, Pearu Peterson, et al. SciPy: Open source scientific tools for Python, 2001–. URL <http://www.scipy.org/>. [Online; accessed April 20, 2019].
- [31] Nicola Demo, Marco Tezzele, and Gianluigi Rozza. PyDMD: Python Dynamic Mode Decomposition. *The Journal of Open Source Software*, 3(22):530, 2018. doi: <https://doi.org/10.21105/joss.00530>.
- [32] Richard Reed. Applications of the Karhunen-Love Transform for Basis Generation in the Response Matrix Method. Master’s thesis, Kansas State University, 2015. URL <https://krex.k-state.edu/dspace/handle/2097/19081>.

- [33] Steven Paul Hamilton. *Numerical solution of the k -eigenvalue problem*. PhD thesis, Emory University, 2011.