On model-order reduction in neutronic systems via POD-galerkin projection

by

Rabab Elzohery

AN ABSTRACT OF A DISSERTATION

submitted in partial fulfillment of the
requirements for the degree

DOCTOR OF PHILOSOPHY

Alan Levin Department of Mechanical & Nuclear Engineering
Carl R. Ice College of Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

2022

Abstract

Numerical simulation has become an indispensable tool in a wide range of research areas and industries. In nuclear engineering, such simulations are important to understand the behavior of nuclear reactors under different conditions and, accordingly, to develop optimized designs with established safe operational limits. Estimating quantities of interest like neutron core power, and temperature distributions requires the solution of a set of partial differential equations that model the nuclear reactor physics. These equations can be solved deterministically using a variety of phase-space discretization techniques and, generally, a sufficiently "fine" phase-space grid is needed to obtain a desired accuracy. However, a brute-force pursuit of accuracy using ever finer grids results in ever larger algebraic systems that can quickly become too expensive to solve. This expense is multiplied for applications that require repeated simulations, such as design optimization and uncertainty quantification.

Reduced-Order Models (ROMs) were developed to overcome the high computational cost of numerical simulation for neutronic systems by providing a rapid approximation of the simulated output. The work presented here was primarily exploratory in nature, and the primary goal was to understand the applicability of various model reduction techniques to specific problems of interest in reactor physics. In particular, steady-state and transient neutronic of varying levels of complexity have been analyzed. Efforts began by exploring a 1-D model of a reactor subject to a ramp insertion of reactivity, for which POD-Galerkin projection was tested and selected for more complex problems. These problems included the classic 2-D TWIGL and LRA transient diffusion benchmarks, for which a fine grid solution of the diffusion equations served as the full-order model (FOM) and a source of data for construction of ROMs. To reduce the cost of constructing and applying ROMs for nonlinear problems, the hyper-reduction technique known as Discrete Empirical Interpolation Method (DEIM) and its matrix version were used with POD-Galerkin projection.

Although most of the work presented was based on the diffusion equation, a preliminary application to the transport equation was also conducted. A critical difference of the application of the ROM to the transport equation is that the FOM relies on matrix-free operators, which can complicate the use of POD-Galerkin projection. However, the methods developed proved to be applicable to the implementation. Importantly, although the POD-Galerkin method is an intrusive ROM technique, the implemented ROM did not require any major changes to the existing code, making it a *trivially intrusive* technique. Rather, access only to the system operator, which represents all of the underlying system discretization, was needed.

Although many individual cases were considered, the primary conclusions to make are that (1) POD-Galerkin projection of diffusion- or transport-based models yield ROMs that approximate core powers with errors less than the 1% and with computational speedups that range from approximately 3 to 50 depending on the type and numerical fidelity of the underlying FOM and (2) such ROMs can be implemented in the many modern diffusion or transport codes (e.g., the various MOOSE-enabled tools based on finite-element methods[1], or the highly-scalable Denovo discrete-ordinates code[2]).

On model-order reduction in neutronic systems via POD-galerkin projection

by

Rabab Elzohery

B.S., Alexandria University, 2006

M.S., Alexandria University, 2012

A DISSERTATION

submitted in partial fulfillment of the
requirements for the degree

DOCTOR OF PHILOSOPHY

Alan Levin Department of Mechanical & Nuclear Engineering
Carl R. Ice College of Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

2022

Approved by:

Major Professor
Jeremy Roberts

Copyright

Abstract

Numerical simulation has become an indispensable tool in a wide range of research areas and industries. In nuclear engineering, such simulations are important to understand the behavior of nuclear reactors under different conditions and, accordingly, to develop optimized designs with established safe operational limits. Estimating quantities of interest like neutron core power, and temperature distributions requires the solution of a set of partial differential equations that model the nuclear reactor physics. These equations can be solved deterministically using a variety of phase-space discretization techniques and, generally, a sufficiently "fine" phase-space grid is needed to obtain a desired accuracy. However, a brute-force pursuit of accuracy using ever finer grids results in ever larger algebraic systems that can quickly become too expensive to solve. This expense is multiplied for applications that require repeated simulations, such as design optimization and uncertainty quantification.

Reduced-Order Models (ROMs) were developed to overcome the high computational cost of numerical simulation for neutronic systems by providing a rapid approximation of the simulated output. The work presented here was primarily exploratory in nature, and the primary goal was to understand the applicability of various model reduction techniques to specific problems of interest in reactor physics. In particular, steady-state and transient neutronic of varying levels of complexity have been analyzed. Efforts began by exploring a 1-D model of a reactor subject to a ramp insertion of reactivity, for which POD-Galerkin projection was tested and selected for more complex problems. These problems included the classic 2-D TWIGL and LRA transient diffusion benchmarks, for which a fine grid solution of the diffusion equations served as the full-order model (FOM) and a source of data for construction of ROMs. To reduce the cost of constructing and applying ROMs for nonlinear problems, the hyper-reduction technique known as Discrete Empirical Interpolation Method (DEIM) and its matrix version were used with POD-Galerkin projection.

Although most of the work presented was based on the diffusion equation, a preliminary application to the transport equation was also conducted. A critical difference of the application of the ROM to the transport equation is that the FOM relies on matrix-free operators, which can complicate the use of POD-Galerkin projection. However, the methods developed proved to be applicable to the implementation. Importantly, although the POD-Galerkin method is an intrusive ROM technique, the implemented ROM did not require any major changes to the existing code, making it a *trivially intrusive* technique. Rather, access only to the system operator, which represents all of the underlying system discretization, was needed.

Although many individual cases were considered, the primary conclusions to make are that (1) POD-Galerkin projection of diffusion- or transport-based models yield ROMs that approximate core powers with errors less than the 1% and with computational speedups that range from approximately 3 to 50 depending on the type and numerical fidelity of the underlying FOM and (2) such ROMs can be implemented in the many modern diffusion or transport codes (e.g., the various MOOSE-enabled tools based on finite-element methods[1], or the highly-scalable Denovo discrete-ordinates code[2]).

Table of Contents

List of Figures

List of Tables

List of Symbols

Symbol	Description	Units
Σ	macroscopic cross section	cm^{-1}
ψ	angular flux	$\mathrm{cm}^{-1}\mathrm{s}^{-1}\mathrm{sr}^{-1}$
ϕ	scalar flux	$\mathrm{cm}^{-1}\mathrm{s}^{-1}$
$\mathbf{\Omega}$	solid angle vector	sr
E	energy	MeV
ν	neutrons produced per fission	
χ	PDF for fission emission energy	
μ	cosine of the discrete scattering angle	
∇	del operator	

Acknowledgments

I would like first to express my deep gratitude to my parents, who started this journey with me and left our world before it ended. I owe them everything for their devotion and unconditional love that kept empowering me even after they left. I would like to thank Dr. Jeremy Roberts for his guidance and encouragement as my advisor and for being a caring friend in all the hard times I went through during the five years I spent here. A very special thanks to my kids, Saif and Salma, who shared this journey with me. I am thankful for their love and patience, for cheering me up, and for their independence that gave me time to work on my thesis. I would like to thank Dr. Mohamed Abdo for being very generous in teaching me various topics to continue my research. I also would like to thank the committee members for their helpful suggestions that improved the work and proof-reading the thesis. Thank you to my colleagues, Richard, Kevin, Ye, John, and Leidong, for the insightful discussion we had and the fun times we spent together. Last but not least, I would like to thank my siblings and the friends I met in Manhattan. They made this journey enjoyable and worth it.

Dedication

To my mother, Sohair Shoair, who was addicted to achievement, and my father, Reda Elzohery, who was fond of knowledge.

Chapter 1

Introduction

1.1 Motivation and Goal

Modeling and simulation of complex systems has become an indispensable tool in many different science and engineering disciplines. Commonly, these complex systems are described by a set of equations that is challenging to solve analytically. Alternatively, these systems are solved by employing numerical discretization schemes such as the finite-difference, finite-volume and finite-element methods.

These methods rely on discretizing the system along each independent variable, and the solution is provided only at these discrete points. To improve the method accuracy, the system needs to be solved over a very fine mesh. Although this results in a more reliable solution, it leads to a growing space dimension, which is known as the *curse of dimensionality*, and an accordingly large computational cost. This problem of the computational cost becomes inevitably prohibitive with applications that require repetitive execution of the simulation. Examples of these applications include uncertainty quantification, parametric studies and design optimization. In such studies, the high-fidelity model has to be solved at multiple parameter points. Hence, surrogate models that are cheap, yet accurate, are sought to provide a rapid and reliable approximation of the solution of these systems. Reduced-order models (ROMs) are a broad class of surrogate models that exploit the correlation that exists in the

input/output space to represent the system in terms of significantly fewer degrees of freedom. The premise is that this lower-order model will capture the essential features of the system and preserve the input-output mapping as much as possible.

Many ROM techniques have been developed and used in a wide range of applications and they can be classified into two categories: intrusive and non-intrusive. Intrusive methods are ones that demand access to the source code of the simulation and modifying/adding to the existing code. This requires knowledge of the underling equations that govern the system being studied and hence they are considered physics-based models. Projection-based methods[3] are examples of such methods. On the other hand, non-intrusive methods use the simulation as a black box to generate some outputs that correspond to some inputs and then build a mapping function between these inputs and outputs. In general, these methods are purely data-driven and not physics-based. Examples include response surfaces and Dynamic-Mode Decomposition (DMD)[4;5]. Each broad categories has advantages and drawbacks. Intrusive methods are often challenging to implement because access to source code can be limited or prohibited. Furthermore, intrusive methods require a solid understanding of the physics and the governing equations. However, once they are implemented, the developed ROM can be used for many different scenarios under various conditions. Data-driven methods, on the other hand, are more suitable to predict outputs for the specific conditions under which it was derived. In the same time, they offer a way to readily develop approximate models, especially with the recent availability of open source libraries that include a variety of modern methods and solvers such as TensorFlow[6]and Scikit-learn[7].

In this thesis, we are interested in developing ROMs for nuclear engineering applications, particularly neutronics, in which the main physics is neutron transport. These applications could be computationally intractable in situations that require repetitive core calculations with varying various model parameters. For example, in recent years, safety analysis within the nuclear industry and regulatory bodies have shifted to best-estimate plus uncertainties (BEPU) calculations rather than to rely on conservative models that lead to overestimated safety margins.

In these calculations, confidence bounds are computed by means of uncertainty analysis

that entails repetitive executions of the numerical simulations, where increasing the number of simulation runs leads to more reliable confidence intervals and, accordingly, reliable safety and operational limits. For modern, light-water reactor systems, methods developed over the past four decades have been highly tuned for these systems to provide credible results quickly. These methods include a variety of nodal-diffusion and cross-section homogenization techniques, but for advanced nuclear reactors (e.g., fast-spectrum, metal-cooled reactors or high-temperature, gas-cooled reactors with multiple levels of fuel heterogeneity), these existing methods are insufficient. Advanced nuclear reactors are subject to optimization and parametric studies require running the simulation at multiple input parameters to explore these parameters space and aid in developing optimal and economical designs. Because there exists much less experimental data to support these analyses, the availability of accurate but efficient methods is especially important.

Since the solutions of high-fidelity models evolve in a lower-dimensional space due to correlation induced by the spatial discretization, governing physics, and geometry constraints, they can be performed using reduced-order models.

Here, an intrusive ROM based on the POD-Galerkin projection has been developed to approximate neutronic steady-state and transients with substantially reduced simulation run times so that multi-query studies can be completed affordably.

The original models to be approximated solve for the neutron population in a nuclear reactor core, which can be used to derive essential quantities to the reactor safety and operation such as the core power, core temperature, and the energy deposited. Our goal is to develop and assess a ROM framework that can be used for different purposes, such as design optimization and propagating input parameter uncertainties like nuclear data. Moreover, we aim to identify practical limitations and computational challenges, if any, that could make method implementation in production level codes problematic. The choice of this method is based on a scoping study that has been made using a simple 1-D problem to explore the potential of different methods to approximate neutronic transients[8]. The methods included DMD and DMD/POD Galerkin projection. The POD-Galerkin projection gave the best results, and hence, it was used to explore and analyze more problems with increasing level of

complexity.

In particular, a preliminary study of multi-physics problems, in which coupling of other physics induces non-linearity, was performed. Non-linearity naturally leads an expensive full-order model and the same for the corresponding reduced-order model. To mitigate the impact of non-linearities, a method to improve the ROM computational efficiency by splitting it into two distinct, offline and online stages was developed. The method is based on the Discrete Empirical Interpolation (DEIM)[9] and its matrix version which belong to the broad class of hyper-reduction techniques. The full-order model to be approximated is mostly based on the multi-group diffusion equation, however, a preliminary application to the transport equation is presented, as well. The implementation has been performed in Detran[10], an open-source code for deterministic transport and algorithm development.

1.2 Reduced-order models for neutronics applications

For neutronics applications, deterministic methods to characterize the neutron population are based mainly on the neutron transport equation and the neutron diffusion equation[11–13]. These equations can be coupled with other models to account for other physics, such as thermal hydraulic and fuel performance. An exact solution for these equations can be found only for simple problems. In practice, various assumptions and simplifications are made in order to be able to solve them for realistic problems. These simplifications involve discretizing the system in space, energy, and angle, which leads to a large number of algebraic equations that are very expensive to solve. Over the last few decades, some methods were employed in order to reduce this computational burden. In fact, the diffusion equation is an approximation of the transport equation, by assuming a linear angular dependence. Spatial homogenization of cross sections is another simplification that has been introduced to reduce the level of heterogeneity inside the reactor core and hence the spatial space[14;15]. Methods to accelerate the convergence of the iterations in the transport calculations, and hence speed up the computational time, have been introduced[16;17].

For time-dependent problems, quasi-static methods[18–20] are used to approximate the

time-dependent flux by a two-level process in which the flux shape is update at coarse time steps between which its magnitude is calculated over fine time mesh by solving an inexpensive system of equations. Although these strategies have alleviated the computational burden of reactor core calculations, they remain relatively expensive because they still solve the problem in the original high dimensional space. Recently, there has been a growing interest in reduced-order models (ROMs) to provide a rapid approximation of the response of the simulation by performing most of the computation in a lower dimensional space. In the nuclear community, various ROM methods have been used for different problems in the neutronics field. However, the focus here is on the work that has been done to approximate the steady-state and the time-dependent neutron flux coming from solving the diffusion and the transport equations.

1.2.1 Steady-State Problems

In the framework of projection based methods, Wang[21] proposed a POD-like approach to find a reduced basis for the transport and diffusion eigenvalue problems. The algorithm builds this subspace in an iterative manner by adding a new basis vector to the subspace in each iteration until a user-set accuracy is satisfied. One basis set was generated for all energy groups and the developed ROM was used for a parametric study in which the cross sections were the parameters of interest. Subsequent work[22] used POD-Galerkin projection for a similar problem, but a basis set was generated for each energy group and a greedy POD sampling[23;24] was used to construct a global basis in order to capture the parameter domain variance. Moreover, for computational efficiency, the developed ROM was split into an offline and online stage by taking advantage of the affine parameter dependence resulting from the finite element descretization. Another approach to build the reduced subspace was presented by Chunyu[25]. In this approach, a certified reduced basis was constructed by using greedy sampling in which, at each iteration, a full-order solution, based on the finite element approximation, is added to the subspace after which it is orthonormalized. Steady-state multi-physics ROMs have been also addressed. For example, parametrized ROM

based on the reduced basis (RB) method was developed for Lead Fast Reactor (LFR) single channel[26], where the parameter of interest were the inner radius of the fuel pellet and the inlet lead velocity. In this work, the non-linearity induced by the thermal feedback coupling was simplified by linearization the non-linear term. Another application of multi-physics ROM for steady-state problems was presented for a Molten-salt reactor (MSR)[27]. The ROM was also based on the POD-Galerkin projection, and the non-linearity was treated using the discrete empirical interpolation method (DEIM). Recent work[28] combined POD-Galerkin projection with domain decomposition. The goal was to improve the ROM computational efficiency by decomposing the problem into non-overlapping domains for which parallel computing can take place. Results showed that there was no loss of accuracy due to the domain decomposition compared to the POD-Galerkin ROM without applying domain decomposition.

For steady state application, we developed a parametric ROM for the eigenvalue diffusion equation using the matrix version of the DEIM to obtain an offline-online decomposition of the ROM that is described in chapter 3.

1.2.2 Time-dependent problems

For a time-dependent problem, Sartori[29] compared modal methods and POD modes to predict the reactivity induced by a perturbation in the reactor core. The snapshots used were the steady-state solutions generated with varying specific parameters in the perturbed regions. The results showed that the POD performed better for localized perturbations for which the modal method needed a considerable number of eigenfunctions to correctly predict the reactivity. In different work[30] modal methods with adjoint and forward eigenfunctions as the test basis were compared with POD basis of both the forward and adjoint steady state flux. The ROM was sought to reproduce the reactivity induced following either thermal feedback effects or control rod movement in the ALFRED reactor. Again, the POD basis outperformed the modal basis where the adjoint test basis was the best choice in reproducing the reactivity effects in terms of both accuracy and computational time. For control applications, a parameterized ROM was developed in which the control rod height was the parameter of

interest[31]. The movement of the control rod was modeled as a geometric parametrization of the spatial domain. Tsujita[32] used POD-Galerkin projection for the time-dependent diffusion in which two different snapshots sets were used to generate the POD bases, i.e, steady-state snapshots and time-dependent snapshots. Results showed that the POD basis derived from time-dependent snapshots is slightly better with maximum errors in the power of less than 0.6%. Another method based on the proper generalized decomposition (PGD) was proposed by Alberti[33]. Although the method provided acceptable accuracy for homogeneous problems, it did not perform as well for problems with stronger spatial heterogeneity of the same problems and the results were not promising in terms of the accuracy and the computational efficiency. For the transport equation as a full-order model, the POD-Galerkin projection was employed to models based on the discrete-ordinate method where snapshots of the angular flux were used to generate the basis[34]. The ROM was used to predict the flux outside the time range of snapshots for which the error was about 1%.

For kinetic application, we develop POD-Galerkin ROMs for different problems where greedy sampling was used to parameterize the ROM. To improve its efficiency the matrix version of the DEIM, i.e, MDEIM was used in conjunction with the POD projection for the diffusion models, where interpolation of the reduced operator was used for the transport model.

1.3 Outline of the Thesis

This thesis is organized as follows. Chapter 2, an overview of the different methods used to develop the ROM and improve its efficiency is presented and the algorithm of each method is described. In chapter 3, the mathematical formulations of the steady state diffusion equation and the corresponding ROM are given. Also, an application with its results are presented. The time-dependent diffusion equation and the developed ROM are presented in chapter 4 with an application to the TWIGL benchmark. Non-linear problems resulting from multi-physics coupling are addressed in chapter 5 with an application to the LRA problem. In chapter 6, a preliminary application to the transport equation based on a discrete-ordinate method is presented. Discussion, general remarks and ideas for future works are given in

chapter 7.

Chapter 2

Methods

This chapter aims to give an overview of the methods and tools used throughout the thesis. The overview starts with the *Proper Orthogonal Decomposition* (POD) which is a key component in the developed framework that is used to explore different problems potential of reduction. After generating a problem-specific POD space, the state variable is approximated in terms of its basis, where the expansion coefficients are computed by means of a *POD-Galerkin projection*. Next, *a POD Greedy sampling* is presented aiming to take the POD-Galerkin model one step forward towards treating parametric problems by generating a global subspace that can be used for a potentially wide range of parameters. In order to improve the ROM efficiency for nonlinear and parametric problems, splitting the ROM into an offline and online stages is desirable. For some problems, this splitting can be easily obtained due to some affine properties[1] that enable ROM decomposition into an expensive one-time offline stage, and an inexpensive online stage that needs to be performed for each new parameter or time instance. Here, we are dealing with problems for which such decomposition is not readily available. Thus, to approximate such decomposition, a method called the *Discrete Empirical Interpolation Method* is employed within the framework of the POD-Galerkin projection. Finally, a brief overview on the *Dynamic Mode Decomposition* (DMD), which is a different

[1] Affinity here means that time/parameter function/operator can be represented by a linear combination of time/parameter independent functions/operators, each weighted by time/parameter dependent scalar function.

reduction method is introduced. In this work, DMD was used as part of a scoping study performed to assess the performance of different reduced models to approximate transient problems.

2.1 Proper Orthogonal Decomposition

Proper Orthogonal Decomposition (POD) is one of the most popular reduction techniques. The method is used to construct a reduced basis that approximates a high-dimensional quantity. The main assumption is that the intrinsic dimension of the system is significantly small and hence can be captured by few basis vectors. First, we illustrate the method using the continuous formulation and then make the connection to the discrete form and the *Singular value Decomposition* (SVD).

Let $y(x,t)$ be a solution of some PDE where $t \in [0,T]$ and $x \in [-L,L]$. The POD aims to approximate a given data set using a basis $\mathbf{\Phi}$ of rank l, such that the mean square of the reconstruction error is minimized in the interval $[0,T]$. Mathematically this can be expressed as

$$\min_{\mathbf{\Phi}} \frac{1}{T} \int_0^T \|y(x,t) - \langle y(x,t), \mathbf{\Phi} \rangle \, \mathbf{\Phi}\|^2 dt \qquad \text{subject to } \|\Phi\|^2 = 1\,, \tag{2.1}$$

where $\langle \cdot, \cdot \rangle$ denotes the inner product. Note that Eq. 2.1 is equivalent to the following maximization problem

$$\max_{\mathbf{\Phi}} \frac{1}{T} \int_0^T \| \langle y(x,t), \mathbf{\Phi}\mathbf{\Phi}^T \rangle \|^2 dt \qquad \text{subject to } \|\Phi\|^2 = 1 \tag{2.2}$$

One can show that upon solving this maximization problem, it reduces to an eigenvalue problem as follows

$$\langle \mathcal{R}(x',x), \mathbf{\Phi} \rangle = \lambda \mathbf{\Phi}\,, \tag{2.3}$$

where $\mathcal{R}(x',x)) = \frac{1}{T}\int_0^T y(x',t) y^*(x,t) dt$. Note that in a discrete form this corresponds to the correlation matrix between a set of trajectories at discrete time instances. The eigenvectors are the POD modes and the corresponding eigenvalue λ of each mode denotes the weight or

the energy of each mode. For real-world problems, a closed form of the $y(t,x)$ is not easily obtained; rather, most models are solved using numerical methods, which yield solutions of discrete points in time and/or space. Hence, a common way to construct the subspace is the method of snapshots[35]. Let $\mathbf{Y} = [\mathbf{y}_1, ..., \mathbf{y}_m]$ be a real $n \times m$ matrix where each column $\mathbf{y}_i \in \mathbb{R}^n$ represents a snapshot of some observable extracted from either an experiment or a simulation. A subspace is sought to be constructed such that the reconstruction error is minimized across all snapshots. If the basis set is limited to l basis vectors $\mathbf{u}_i$, $0 < i < l$, the discrete variant of Eq. 2.1 can be written as

$$\min_{\{\mathbf{u}_i\}_{k=1}^{l}} \sum_{i=1}^{i=m} \|\mathbf{y}_i - \sum_{k}^{l} < \mathbf{u}_k, \mathbf{y}_i > \mathbf{u}_k\|_2 \tag{2.4}$$

$$\text{subject to } (\mathbf{u}_i, \mathbf{u}_j) = \delta_{ij} \text{ for } 1 \leq i \leq l, 1 \leq j \leq l\,,$$

where δ is the Kronecker delta [2]. The correlation matrix $\mathbf{Y}\mathbf{Y}^T$ can be constructed for which the eigenvectors can be computed. Equivalently, and more computationally efficient, the SVD of the data matrix is computed

$$\mathbf{Y} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T\,, \tag{2.5}$$

where $\mathbf{U} \in \mathbb{R}^{n\times n}$ and $\mathbf{V} \in \mathbb{R}^{m\times m}$ are orthogonal matrices representing the left and right singular matrices respectively, and $\mathbf{\Sigma} \in \mathbb{R}^{n\times m}$ is a diagonal matrix of singular values. Thus, the subspace comprised from the first r vectors of $\mathbf{U}$ is the best rank r approximation of $\mathbf{Y}$, also called POD basis of rank r. There is no general *a priori* rule based on which the rank is chosen; however one can set a threshold ϵ such that

$$\frac{\sum_{i=0}^{r} S_i}{\sum_{i=0}^{m} S_i} \geq \epsilon\,, \tag{2.6}$$

where Si is the ith singular value. The steps of the POD procedure is described in algorithm 1.

[2] $\delta_{ij} = \begin{cases} 0 & \text{if } i \neq j \\ 1 & \text{if } i = j. \end{cases}$

Algorithm 1 POD; Proper Orthogonal Decomposition

Input: Snapshots matrix: $\mathbf{Y} \in \mathbb{R}^{n \times m}$, ϵ

Output: Reduced basis matrix $\mathbf{U} \in \mathbb{R}^{n \times r}$.

1: Compute the SVD: $\mathbf{Y} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$.

2: Set $\mathbf{U} = \mathbf{U}[:, :r]$ where $\frac{\sum_{i=0}^{r} S_i}{\sum_{i=0}^{m} S_i} \geq \epsilon$.

2.2 POD-Galerkin Projection ROM

Consider the following dynamic system that results from discretizing a partial differential equation using a numerical scheme such as the finite difference method:

$$\frac{d\mathbf{y}(t)}{dt} = \mathbf{A}\mathbf{y}(t) + \mathbf{f}(\mathbf{y}(t)), \qquad \mathbf{y}(\mathbf{0}) = \mathbf{y_0}\,, \tag{2.7}$$

where $\mathbf{y} \in \mathbb{R}^n$ is the state variable and $\mathbf{A} \in \mathbb{R}^{n \times n}$ is the operator resulting from the system discretization. In general, this method involves two main steps. First, the state variable $\mathbf{y}$ is expanded in terms of a trial subspace basis, which is the POD space in this case, where each basis vector is weighted by a temporal coefficient

$$\mathbf{y} \approx \mathbf{\Phi}\hat{\mathbf{y}}(t)\,, \tag{2.8}$$

where $\hat{\mathbf{y}}(\mathbf{t}) \in \mathbb{R}^r$ is a vector of the temporal coefficients. Second, the system is projected onto a test subspace, which is the same as the trial subspace in the Galerkin projection procedures. By multiplying the system on the left by $\mathbf{\Phi}^T$ and making use of the orthogonality of the POD basis (i.e, $\mathbf{\Phi}^T\mathbf{\Phi} = \mathbf{I}$), we end up with

$$\frac{d\hat{\mathbf{y}}}{dt} = \mathbf{A}_r\hat{\mathbf{y}} + \mathbf{\Phi}^T\mathbf{f}(\mathbf{\Phi}\hat{\mathbf{y}}) \qquad \hat{\mathbf{y}}(0) = \mathbf{\Phi}^T\mathbf{y}_0\,, \tag{2.9}$$

where $\mathbf{A}_r \in \mathbb{R}^{r \times r} = \mathbf{\Phi}^T\mathbf{A}\mathbf{\Phi}$. Thus, we are left with r ODE's instead of n ODE's, which makes the resulting ROM more computationally efficien, and, once it is solved for $\hat{\mathbf{y}}$, the full

order solution can be computed by Eq. 2.8. Note that the POD basis is chosen here for its optimality property. However, different methods have been used in the literature, such as the eigenmodes analysis and the reduced-basis method[36–38]. Moreover, the test basis and the trial basis could be different, in which case the method is called Petrov-Galerkin projection. In such cases, unless the test and the trial basis are bi-orthogonal, there will be an additional computational cost associated with inverting the matrix resulting from multiplying the left hand side of Eq. 2.7 by the test basis.

2.3 Greedy POD Sampling

A greedy sampling in the parameter space[39] can be combined with POD to construct a global subspace that can be used in the framework of parametric ROMs. The primary goal is to ensure that the parameter domain of interest (i.e, $\mathbb{P}$) is efficiently sampled and includes enough information such that it can be used or wide range of parameters including those of higher dimensions. To construct this subspace, a training parameters dataset, $P_{train} \subset \mathbb{P}$, is prepared. The greedy subspace is initialized with the first r_1 POD basis computed from a randomly selected sample of the training data. Next, this subspace is used to compute the ROM prediction error of the state variable for the rest of the training samples. The sample at which the error is worst is selected, and the subspace is enriched by adding the first r_1 POD basis computed from the selected sample snapshots. Selecting the samples with the worst errors ensures efficient sampling of the parameter space. Intuitively, samples that are close to the previously chosen parameter points will have a smaller ROM error. On the other hand, one expects that the ROM will fail to give an accurate prediction at samples that are far from the included points in the current subspace. Thus, we keep adding new basis vectors until a maximum size of the subspace is reached or a user-specified error tolerance at a reference parameter point $\boldsymbol{\mu}_{\text{ref}}$ is met. Note that each time a new basis vector is added, the subspace is orthonormalized. Regarding the error criteria based on which a new sample is selected, there are two approaches. Ideally, the true error can be used leading to "strong POD-greedy" [40]. Alternatively, an error indicator can be used to alleviate the computational burden associated

with evaluating the full-order-model. Here, we used the strong-POD greedy approach. The algorithm steps are shown in Algorithm 2. The term $\Delta(\mathbf{U}, \boldsymbol{\mu}_{\text{ref}})$ is the error measure using the current reduced subspace at $\boldsymbol{\mu}_{\text{ref}}$, and r is the maximum size of the greedy basis.

Algorithm 2 POD-greedy sampling procedures

Input: $\epsilon_{tol}, r, r_1, P_{train} \subset \mathbb{P}$

Output: A reduced space $\mathbf{U}$

set n=0

while $\epsilon := \Delta(\mathbf{U}, \boldsymbol{\mu}_{\text{ref}}) > \epsilon_{tol}$ or $n < r$ **do**

$\boldsymbol{\mu} := \text{argmax}_{\boldsymbol{\mu} \in P_{train}} \Delta(\mathbf{U}, \boldsymbol{\mu})$

evaluate the FOM at $\boldsymbol{\mu}$, $\mathbf{Y}(\boldsymbol{\mu}) := \{\mathbf{y}^1, \mathbf{y}^2, ...\mathbf{y}^k\}$

compute the POD modes of the FOM snapshots, $\{\zeta_1, \zeta_2, ..., \zeta_{r1}\} := \text{POD}(\mathbf{Y}(\boldsymbol{\mu}), r_1)$

Compute the SVD of $\mathbf{Y}(\mu)$ and keep the first r_1 POD modes:$\{\zeta_1, \zeta_2, ..., \zeta_{r1}\} = \text{POD}(\mathbf{y}^1, \mathbf{y}^2, ...\mathbf{y}^k\}, r_1)$

$Z = \{Z, \{\zeta_1, \zeta_2, ..., \zeta_{r1}\}\}$ and set $n = n + r_1$

update the reduced space, $\mathbf{U} := \mathbf{U} \cup \{\zeta_1, \zeta_2, .., \zeta_{r1}\}$ and orthonormalize

$n := n + r_1$

compute $\epsilon = \Delta(\mathbf{U}, \mu_0)$

end while

2.4 Offline-online

Solving the ROM in Eq. 2.9 involves two main steps. The first step is generating the projected terms, i.e, $\mathbf{U}^T\mathbf{A}\mathbf{U}$ and $\mathbf{U}^T\mathbf{f}(\mathbf{U}\hat{\mathbf{y}}(\mathbf{t}))$. The second step is solving the system of ODEs using a suitable time scheme. In terms of the computational cost, the first step is significantly more expensive since the projected terms have to be computed at each time step. This projection requires performing operations with computational complexity that scales with the original dimensions of the system (i.e., n). In particular, to project the nonlinear function $\mathbf{f}(\mathbf{U}\hat{\mathbf{y}}(\mathbf{t}))$, the full-order solution needs to be reconstructed at each time step to evaluate the function

before performing the inner products (i.e, $\mathbf{U}^T\mathbf{f}$), which requires $\mathcal{O}(2nr)$ *flops*. Moreover, in some cases, the operator $\mathbf{A}$ is time-dependent in which case $\mathbf{U}^T\mathbf{A}(t)\mathbf{U}$ has to be computed at each time step. In general, this term requires $\mathcal{O}(n^2r + r^2n)$ *flops*. On the other hand, the second step involves solving a linear system, at each time step, that has a size equal to the number of reduced dimensions (i.e, r). Hence, it is favorable to split the ROM into two stages, an offline stage in which a one-time projection operation takes place, and an online stage in which the reduced system is solved over the time domain of interest. To achieve this decomposition, the operator/function can be expanded into number of time-independent operators/functions each weighted by a temporal coefficient. To illustrate more, the nonlinear term $\mathbf{f}(\mathbf{y}(t))$ can be approximated as

$$\mathbf{f}(\mathbf{y}(t)) = \boldsymbol{\Psi}\boldsymbol{\theta}(t)\,, \tag{2.10}$$

where $\boldsymbol{\Psi}$ is a time-independent basis matrix and $\boldsymbol{\theta}(t)$ is a vector of the associated coefficients. With this decomposition, the projection can be written as

$$\mathbf{U}^T\mathbf{f}(\boldsymbol{\Phi}\hat{\mathbf{y}}) = \mathbf{U}^T\boldsymbol{\Psi}\boldsymbol{\theta}(t)\,, \tag{2.11}$$

Note that the term $\mathbf{U}^T\boldsymbol{\Psi}$ does not depend on time and hence it can be precomputed and stored, while the coefficient $\boldsymbol{\theta}(t)$ is required to be computed at each time step. In a similar fashion, the operator $\mathbf{A}(t)$ can be decomposed as

$$\mathbf{A}(t) = \sum_{i=0}^{k} \theta_i(t)\mathbf{A}_i\,, \tag{2.12}$$

where the matrices $\mathbf{A}_i$ are time independent and $\theta_i(t)$ are the corresponding coefficients. The projection is performed as follows

$$\mathbf{U}^T\mathbf{A}\mathbf{U} = \sum_{i=0}^{k} \theta_i(t)\mathbf{U}^T\mathbf{A}_i\mathbf{U}\,, \tag{2.13}$$

Again, the matrices $\mathbf{U}^T\mathbf{A}_i\mathbf{U}$ are precomputed in an offline stage and the coefficients are computed in the online stage for each new time step. Note that the above discussion applies to parameter-dependent functions and operators, where the parameter variable is treated in the same exact manner as the time. An example of this will be shown in the next chapter. Realistically, some problems either do not admit such decomposition or in the case of time/parameter dependent operators, they require fundamental intrusive changes to the existing high-fidelity model operator to enable this decomposition before performing the reduction. Methods have been proposed to be used with the POD-Galerkin projection to overcome this computational complexity. For example, DMD was used to approximate non-linear terms in the framework of Galerkin projection[41]. Also, methods such as gappy POD, Empirical Interpolation Methods (EIM)[42],[43] and trajectory piecewise-linear (TPWL) approximation[44],[45] were used in the literature to address this issue. Here, we focus on the Discrete Empirical Interpolation Methods (DEIM) and its matrix version to treat non-linearity and time/parameter operators without, intrusively, modifying the high-fidelity model. These methods are known as hyperreduction techniques since they represent another level of reduction within the reduced order model.

2.4.1 Discrete Empirical Interpolation Method

Recently, the DEIM[9] was proposed to approximate the non-linearity in model order reduction. The method is a discrete variant of the Empirical Interpolation Method (EIM)[42] that was originally proposed to approximate non-affine coefficient functions using a reduced-basis expansion to facilitate the online/offline decomposition in finite element methods. The DEIM is proposed for the same goal for different discretizations of the full-order model. The method relies on two main ingredients: (1) a reduced space that approximates the nonlinear function, and (2) a set of spatial interpolation indices. To illustrate the method, let τ denote time or a model parameter. The nonlinear function $\mathbf{f}(\mathbf{y}(\tau))$ is approximated by projecting it onto a

subspace that approximates the function space

$$\mathbf{f}(\mathbf{y}(\tau)) = \boldsymbol{\Psi}\boldsymbol{\theta}(\tau)\,, \tag{2.14}$$

where $\boldsymbol{\Psi} \in \mathbb{R}^{n\times k}$ is a low-dimensional subspace with $k \ll n$, and $\boldsymbol{\theta}(\tau) \in \mathbb{R}^k$ is a vector of the corresponding coefficients. POD is used to find the subspace $\boldsymbol{\Psi}$ using snapshots of the function (i.e, $[\mathbf{y}(\tau_1), ..., \mathbf{y}(\tau_m)]$). Note that these snapshots are computed at the same points used for the state variable snapshots required for the reduced model; hence this step will not introduce any additional cost other than computing the SVD. In order to compute $\theta(\tau)$, a few spatial indices are selected iteratively using a greedy search following Algorithm 3. The basic idea is to select k rows of the basis matrix $\boldsymbol{\Psi}$. The algorithm takes as an input the basis vectors of the subspace $\boldsymbol{\Psi}$. The first index is selected where the entry of the first input basis has the largest magnitude. Each other index $\mathcal{I}_i$ is selected where the approximation of the corresponding input basis vector $\boldsymbol{\Psi}_\mathbf{i}$ in the subspace constructed by interpolating the basis $[\boldsymbol{\Psi}_1, ...\boldsymbol{\Psi}_{i-1}]$ at the previously selected indices is worst. Once the indices are selected, the matrix $\boldsymbol{\Psi}_k \in \mathbb{R}^{k\times k}$ is formed such that

$$\boldsymbol{\Psi}_k = \mathbf{P}^T\boldsymbol{\Psi}\,, \tag{2.15}$$

where $\mathbf{P} = [e_{z1}, e_{z2}, ..., e_{zk}] \in \mathbb{R}^{n\times k}$ and $e_{zi} = [0, .., 0, \underbrace{1}_{zi}, 0..., 0]^{\mathrm{T}} \in \mathbb{R}^n$ is the zith column of the identity matrix. The coefficients can be computed by solving

$$\mathbf{P}^T\mathbf{f}(\tau) = \boldsymbol{\Psi}_k\boldsymbol{\theta}(\tau)\,, \tag{2.16}$$

and the final function approximation is given by

$$\hat{\mathbf{f}}(\tau) \approx \boldsymbol{\Psi}(\boldsymbol{\Psi}_k)^{-1}\mathbf{P}^T\mathbf{f}(\tau)\,. \tag{2.17}$$

Note that the matrix $\mathbf{P}$ does not need to be constructed explicitly, since the term $\mathbf{P}^T\mathbf{f}$ extracts only the elements of $\mathbf{f}$ that correspond to the interpolation indices. Thus, at each time step, only k elements $\mathbf{f}$ need to be evaluated. The approximation error can be bounded as

$$\|\mathbf{f} - \hat{\mathbf{f}}\| \leq \|(\mathbf{P}\boldsymbol{\Psi})^{-1}\|\epsilon\,, \tag{2.18}$$

where

$$\epsilon \approx \sigma_{k+1}\,.$$

Note that σ_{k+1} is the $(k+1)th$ singular value coming from the SVD of the function snapshots matrix. Full derivation of the error bound is given in Ref.[9].

As explained in the previous section, the projection can be performed by computing the term $\mathbf{U}^T\boldsymbol{\Psi}$ once in an offline stage, while in the online stage, the vector $\boldsymbol{\theta}$ is computed by solving the linear system of dimension k in Eq. 2.16. The key point is that the cost of the online computation depends on the reduced dimension k, not the original, high dimension n.

Algorithm 3 DEIM

Input: Snapshots matrix: $\mathbf{X} = [\mathbf{f}_1, \mathbf{f}_2, ..., \mathbf{f}_m] \in \mathbb{R}^{n\times m}$

Output: A reduced space $\boldsymbol{\Psi}$, set of indices $\mathcal{I}$

1: $[\boldsymbol{\Psi}_1, \boldsymbol{\Psi}_2, ..., \boldsymbol{\Psi}_k] = \text{POD}(\mathbf{X}, \epsilon)$[a]

2: $\mathcal{I}_1 = \text{argmax}(|\boldsymbol{\Psi}_1|)$, $\mathcal{I} = \{\mathcal{I}_1\}$

3: $\boldsymbol{\Psi} = [\boldsymbol{\Psi}_1]$, $\mathbf{P} = [e_{I_1}]$

4: **for** i=2 **to** i=k **do**

5: Solve $(\mathbf{P}^T\boldsymbol{\Psi})\mathbf{c} = \mathbf{P}^T\boldsymbol{\Psi}_i$;

6: $\mathcal{I}_i = \text{argmax}\,|\boldsymbol{\Psi}_i - \boldsymbol{\Psi}\mathbf{c}|$

7: $\mathcal{I} = \mathcal{I} \cup \mathcal{I}_i$, $\mathbf{P} = \mathbf{P} \cup e_{I_i}$, $\Psi = \Psi \cup \Psi_i$

8: **end for**

[a]POD refers to Algorithm 1

Numerical example

Here, the method is illustrated by a simple example. However one should keep in mind that the method gains its importance in the context of model reduction. Consider the general non-linear 1-D function

$$f(x, \mu) = (1 - x^2)\cos(\pi\mu x)e^{-x\mu^2}\,, \tag{2.19}$$

where $\mu \in [1, \pi]$ and $x \in [-1, 1]$. To approximate the function using DEIM, first, 50 snapshots of the functions were collected by evaluating the function at different values of μ in the prescribed interval. The POD modes were computed using this snapshot matrix, and the interpolation indices were determined using algorithm 3. Shown in Fig. 2.1 are the first 8 POD modes along with the selected interpolation points. Fig. 2.2 shows the singular values. Their rapid decay implies the potential of this function to be reduced and accordingly good approximation using a low-order DEIM.

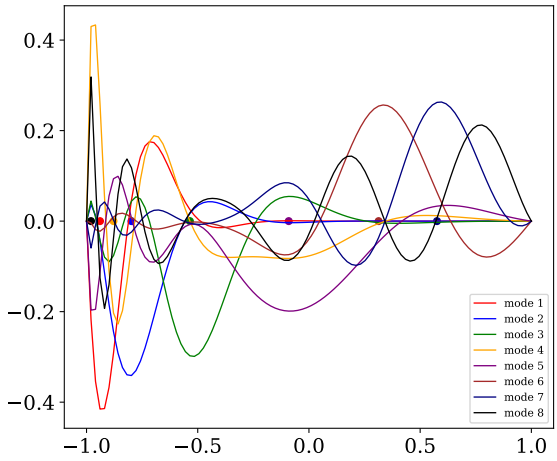


Figure 2.1: POD modes and the interpolation points

The DEIM procedure was performed for different ranks of 2, 4, 6 and 8 and the approximation error was computed. Fig. 2.3 shows the exact solution at $\mu = 2.12$ which is a parameter point not used in generating the snapshots. Also shown is the DEIM approximation using different ranks, with the approximation error (measured in L_2) norm is given for each case. The points used for interpolation are shown in green circles with a corresponding number i, which denotes the index selected in iteration i.

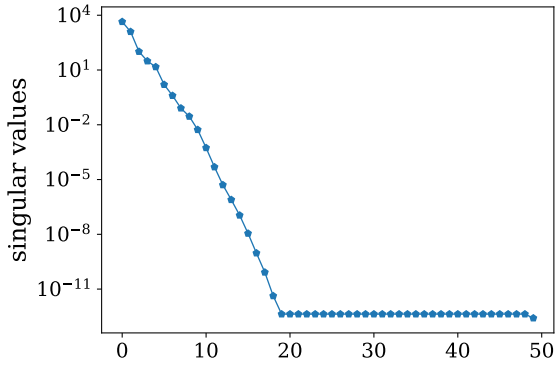


Figure 2.2: POD modes and the interpolation points

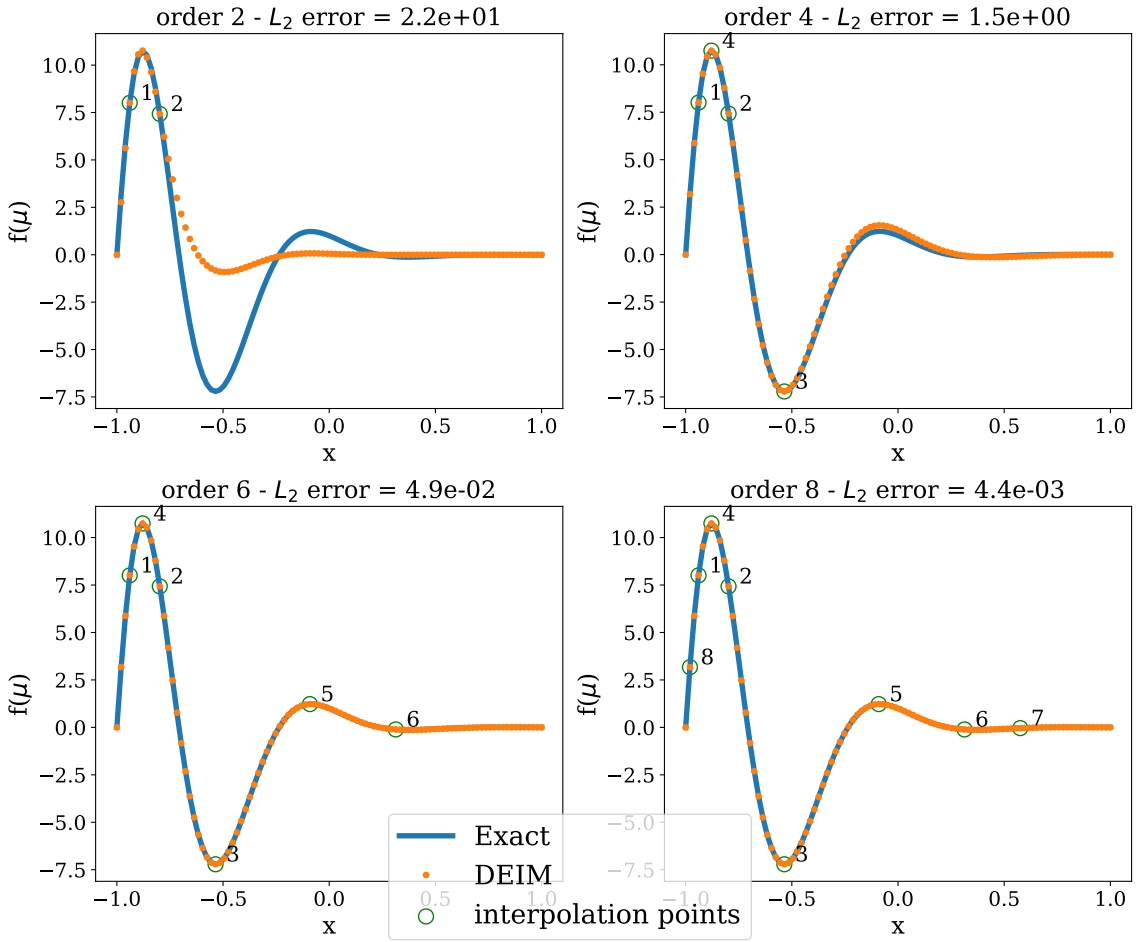


Figure 2.3: The DEIM approximation and the exact function at $\mu = 2.22$

2.4.2 The matrix version of DEIM (MDEIM)

The matrix version of the DEIM (MDEIM) is suggested in[3;46] to approximate parameter/time-dependent operators. Consider a parameterized matrix $\mathbf{A}(\boldsymbol{\mu}) \in \mathbb{R}^{n\times n}$, the goal is to find a decomposition similar to Eq. 2.12 so that an offline-online ROM splitting is attainable. The DEIM procedures can be exploited to find such decomposition. First, the matrix can be cast in a vector form by stacking all of its columns in one vector $\mathbf{a}(\mu) \in \mathbb{R}^{n^2}$. This vector can be treated in the same exact way as the function $\mathbf{f}(t)$ in the previous section, i.e, $\mathbf{a}(\mu) = \boldsymbol{\Psi}\boldsymbol{\theta}(\mu)$. Snapshots of the serialized matrix are collected at different parameter values and the reduced subspace $\boldsymbol{\Psi}$ and the interpolation indices are computed following algorithm 3. Again, the corresponding coefficients are obtained by solving Eq. 2.16. Next, each vector in $\boldsymbol{\Psi}$ can be mapped back to an $n \times n$ matrix, i.e, $\mathbf{A}_i$. Hence, the matrix is approximated as $\hat{\mathbf{A}}(t) = \sum_{i=0}^{k} \theta_i(t)\mathbf{A}_i$ and the projection term is computed as in Eq. 2.13. As pointed out in Ref.[47], the Weyl–Mirsky theorem[48] ensures that the singular values of the approximated matrix approach that of the original as k increases

$$|\sigma_i - \hat{\sigma}_i| \ \leq \|\mathbf{A} - \hat{\mathbf{A}}\|_{\mathrm{F}} \tag{2.20}$$

where σ_i and $\hat{\sigma}_i$ denote the ith singular value of $\mathbf{A}$ and $\hat{\mathbf{A}}$ respectively. Moreover, some properties of the original matrix such as positive definiteness can be preserved[49], if necessary, by imposing some constrains to the problem in Eq. 2.16.

In the practical implementations, the method efficiency can be improved, as suggested in Ref.[47]. First, the matrix $\mathbf{A}$ is, generally, largely sparse, which leads to a snapshots matrix that is sparse, too. Consequently, it is more convenient to store only the nonzero elements (nz) in $\mathbf{a}$, which should be appreciably more efficient in terms of memory and computational expense. In addition, to obtain the expansion coefficients $(\boldsymbol{\theta})$, the term $\mathbf{P}^T\mathbf{a}(\tau)$ needs to be computed at each new parameter value. In fact, this term only extracts the elements of $\mathbf{a}(\tau)$ that corresponds to the selected interpolation indices. In the offline stage, a mapping can be done between these interpolation indices and the corresponding row-column pairs in $\mathbf{A}$.

When the matrix is formed in the online stage, only these entries are evaluated by restricting a *for loop* over these row-column pairs, leading to more reduction in the computational cost.

2.5 Dynamic Mode decomposition

Recently, DMD has been an emerging method for developing reduced models of dynamic systems. It was developed originally by Schmid[4,5] in the fluid dynamics community to identify spatio-temporal coherent structures from high-dimensional data. The method provides a date decomposition in terms of spatial modes, associated with each of them is an eigenvalue that describes both the frequency and the growth/decay rate of corresponding mode. Moreover, it exploits the POD discussed in Sec.2.1 to reveal the low rank structure in the data. There are variations of DMD algorithms that have been proposed in the literature. Here, we focus on the standard DMD algorithm. To illustrate the method, consider the dynamic system in Eq. 2.7 but with excluding the additive non-linear term, for simplicity. An approximate solution to this system is defined using DMD based on the assumption that, over sufficiently small steps in time, the evolution of $\boldsymbol{y}$ can be well approximated by a relationship of the form

$$\frac{d\boldsymbol{y}(t)}{dt} = \mathcal{A}\boldsymbol{y}\,, \tag{2.21}$$

where this mapping operator $\mathcal{A}$ may not be known explicitly. Suppose the system is sampled at different times to generate sequential data that is arranged in two matrices as

$$\mathbf{Y_0} = \begin{bmatrix} | & | & & | \\ \boldsymbol{y_0} & \boldsymbol{y_1} & \cdots & \boldsymbol{y_{m-1}} \\ | & | & & | \end{bmatrix}, \tag{2.22}$$

and

$$\mathbf{Y_1} = \begin{bmatrix} | & | & & | \\ \boldsymbol{y_1} & \boldsymbol{y_2} & \cdots & \boldsymbol{y_m} \\ | & | & & | \end{bmatrix}, \tag{2.23}$$

where m is the number of samples. DMD computes the leading eigendecomposition of the best-fit linear operator $\mathbf{A}$ relating the data

$$\mathbf{Y}_1 \approx \mathbf{A}\mathbf{Y}_0\,, \tag{2.24}$$

where $\mathbf{A}$ satisfies

$$\mathbf{A} = \underset{\mathbf{A}}{\operatorname{argmin}} \|\mathbf{Y_1} - \mathbf{A}\mathbf{Y_0}\|_F\,, \tag{2.25}$$

for which the resulting matrix is the Moore-Penrose pseudoinverse $\mathbf{A} = \mathbf{Y_1}\mathbf{Y_0}^{\dagger}$. The key point is that the problem is a high-dimensional one, and hence $\mathbf{A}$ is a large matrix and computing it explicitly is not efficient. Rather, a low-rank approximation $\tilde{\mathbf{A}}$ is formed as $\tilde{\mathbf{A}} = \mathbf{U_r^H}\mathbf{A}\mathbf{U_r}$. Here, $\mathbf{U}_r$ is the POD basis of rank r computed from the SVD $\mathbf{Y}_1 = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$. The leading r eigenvalues of $\mathcal{A}$ are inferred from the eigenvalues of $\tilde{\mathbf{A}}$. Once computed, the eigenvalues λ_i of the discrete operator $\mathbf{A}$ can be transformed to the continuous frequencies w_i's via $w_i = \log(\lambda_i)/\Delta t$. The procedure is summarized in Algorithm 4, in which $\boldsymbol{b}$ is the amplitude vector computed to satisfy the initial condition in the DMD basis (i.e., $\boldsymbol{b} = \mathbf{\Phi}^{DMD\dagger}\boldsymbol{y_1}$).

Algorithm 4 DMD (Adapted from Ref.[50])]

Input: Snapshots $\{\mathbf{y}_i\}_{i=1}^m$
Output: $\mathbf{\Phi}^{DMD}, \mathbf{\Lambda}$

1: Stack the first $m-1$ columns in the past snapshots matrix $\mathbf{Y_0}$
   Stack the last $m-1$ columns in the future snapshots matrix $\mathbf{Y_1}$
2: Compute the compact SVD of $\mathbf{Y_1}$: $\mathbf{Y_1} = \mathbf{U_r}\mathbf{\Sigma_r}\mathbf{V_r^H}$
3: Define $\tilde{\mathbf{A}} := \mathbf{U_r^H}\mathbf{A}\mathbf{U_r} = \mathbf{U_r^H}\mathbf{N_2}\mathbf{V_r}\mathbf{\Sigma_r^{-1}}$
4: Compute the eigendecomposition for $\tilde{\mathbf{A}}$: $\tilde{\mathbf{A}}\mathbf{W} = \mathbf{W}\mathbf{\Lambda}$; $\quad \mathbf{\Lambda} = \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \lambda_r \end{bmatrix}$
5: DMD Modes $\mathbf{\Phi}^{DMD} = \mathbf{U_r}\mathbf{W}$ (projected modes)
6: Predicted response $\boldsymbol{y}^{DMD}(t) = \sum_{i=1}^{r} b_i \boldsymbol{\phi}_i^{DMD} e^{w_i t} = \mathbf{\Phi}^{DMD}\mathbf{diag}(e^{\boldsymbol{w}t})\boldsymbol{b}; \quad w_i = log(\lambda_i)/\Delta t$

Although the method is proposed, originally, as an equation-free method, it can be used in the framework of a projection based method. For example, DMD-Galerkin projection procedures can be developed in a similar manner to the POD-Galekin projection discussed in Sec. 2.2 but the DMD modes are used instead of the POD modes.

Chapter 3

Parameterized Diffusion k-eigenvalue Problems

3.1 The Diffusion Equation

The characterization of the distribution of neutrons is a central problem in the design and analysis of nuclear reactors. Using the flux distribution, one can determine quantities like the reaction rates and the reactor power. An exact treatment of the neutron distribution is described by the transport equation, which is a linear version of the *Boltzmann equation* that was developed originally to treat gas dynamics. The equation is integro-differential and casts the neutron distribution in terms of seven independent variables: three spatial variables, two angular dimensions, energy, and time. The main drawback is that it is very difficult to solve for realistic problems and an exact solution can be only obtained for simplified problems. An approximate solution can be obtained by employing numerical discretization techniques that yield large system of equations that are very expensive to solve for large problems.

The diffusion equation to the approximation equation to the transport equation significantly reduces this computational complexity and has long been the workhorse method of reactor analysis. One of the main assumption underlying the diffusion equation is the treatment of the angular dependence, which is assumed to be linear anisotropic, equivalent to

the P_1 approximation. Due to the complex variation of the cross sections within the energy range of interest for nuclear reactors, the equation is split into number of equations. Each equation solves for the neutron flux within an energy range along which the cross sections are assumed to be constant. This representation is known as the *multigroup* equation.

The steady-state, multigroup diffusion equation for G energy groups can be written as[12]

$$-\nabla.D_g\nabla\phi_g(r)+\Sigma_{Rg}\phi_g(r)=\sum_{g'\neq g}^{G}\Sigma_{sg'g}\phi_{g'}(r)+\frac{1}{k_{\text{eff}}}\chi_g\sum_{g'\neq g}^{G}\nu_{g'}\Sigma_{fg'}\phi_{g'}(r)\,, \tag{3.1}$$

where ϕ_g, D_g, Σ_{Rg} and $\nu\Sigma_{fg}$ are the neutron flux, diffusion coefficient, the removal cross section and the production cross section, respectively and $\Sigma_{sg'g}$ is the scattering cross section from group g' to group g, χ_g is the probability that a neutron will be born with an energy group in g and k_{eff} is the effective multiplication factor. In a matrix form, this can be written as

$$\begin{pmatrix} -\nabla\cdot D_1\nabla\phi_1+\Sigma_{R1} & -\Sigma_{s21} & \dots & -\Sigma_{sG1} \\ -\Sigma_{s12} & -\nabla.D_2\nabla\phi_2+\Sigma_{R2} & \dots & -\Sigma_{sG2} \\ \vdots & \vdots & \ddots & \vdots \\ -\Sigma_{s_{1G}} & -\Sigma_{s_{2G}} & \dots & -\nabla.D_G\nabla\phi_G+\Sigma_{RG} \end{pmatrix}\begin{pmatrix}\phi_1\\ \phi_2\\ \vdots\\ \phi_G\end{pmatrix} = \frac{1}{k_{\text{eff}}}\begin{pmatrix} \chi_1\nu_1\Sigma_{f1} & \dots & \chi_1\nu_G\Sigma_{f_G} \\ \chi_2\nu_1\Sigma_{f1} & \dots & \chi_2\nu_G\Sigma_{f_G} \\ \vdots & \ddots & \vdots \\ \chi_G\nu_1\Sigma_{f1} & \dots & \chi_G\nu_G\Sigma_{f_G} \end{pmatrix}\begin{pmatrix}\phi_1\\ \phi_2\\ \vdots\\ \phi_G\end{pmatrix}. \tag{3.2}$$

Typically, this system is solved by employing a numerical descretization scheme such as the finite-difference, finite-volume and the finite-element methods, which leads to the linear system

$$\mathbf{L}(\boldsymbol{\mu})\boldsymbol{\Phi}=\frac{1}{k_{\text{eff}}}\mathbf{F}(\boldsymbol{\mu})\boldsymbol{\Phi}\,, \tag{3.3}$$

where $\boldsymbol{\mu}$ is a vector of the model parameters such as the group constants and the boundary condition, the operator $\mathbf{L} \in \mathbb{R}^{N\times N}$ is the loss operator, $\mathbf{F} \in \mathbb{R}^{N\times N}$ is the fission source operator, $N = G \times n$ with n is the number of spatial cells resulting from the system discretization. Thus, the system in Eq. 3.3 represents a generalized eigenvalue problem which is used in order to predict the criticality and flux distribution of a reactor core at a given configuration, i.e, material and geometry specifications. Some applications such as core optimization, control and uncertainty analysis require solving this system multiple times with varying the core configuration or the model parameters. For large problems, this could be computationally demanding and a ROM that can provide a cheaper approximation of the quantities of interest is desired.

3.2 POD-Galerkin Projection ROM for the eigenvalue Diffusion Equation

The POD-Galerkin projection procedures discussed in Sec. 2.1 are used in order to construct a ROM for the system in Eq. 3.3. A POD subspace is constructed for each group flux by collecting snapshots of the fundamental eigenvectors computed at different reactor states (i.e, with varying the input parameters). Then, the group flux is approximated as $\boldsymbol{\Phi}_g = \mathbf{U}_g\mathbf{a}_g$, where $\mathbf{U}_g \in \mathbb{R}^{n\times r}$ is the POD basis of rank r approximating group g and $\mathbf{a}_g$ is a vector of the associated coefficients. The total flux is approximated as

$$\boldsymbol{\Phi} = \mathbf{U}\mathbf{a}\,, \tag{3.4}$$

where

$$\mathbf{U} = \begin{pmatrix} \mathbf{U}_1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \mathbf{U}_G \end{pmatrix},$$

and

$$\mathbf{a} = \begin{pmatrix} \mathbf{a}_1 \\ \vdots \\ \mathbf{a}_G \end{pmatrix} .$$

Multiplying the system in Eq. 3.3 on the left by $\mathbf{U}^T$, yields

$$\mathbf{U}^T\mathbf{L}(\boldsymbol{\mu})\mathbf{U}\mathbf{a} = \frac{1}{k}\mathbf{U}^T\mathbf{F}(\boldsymbol{\mu})\mathbf{U}\mathbf{a}\,, \tag{3.5}$$

or, more compactly,

$$\tilde{\mathbf{L}}(\boldsymbol{\mu})\mathbf{a} = \frac{1}{k}\tilde{\mathbf{F}}(\boldsymbol{\mu})\mathbf{a}\,, \tag{3.6}$$

where $\tilde{\mathbf{L}}(\boldsymbol{\mu}) \in \mathbb{R}^{r_G \times r_G} = \mathbf{U}^T\mathbf{L}(\boldsymbol{\mu})\mathbf{U}$, $\tilde{\mathbf{F}}(\boldsymbol{\mu}) \in \mathbb{R}^{r_G \times r_G} = \mathbf{U}^T\mathbf{F}(\boldsymbol{\mu})\mathbf{U}$ and $r_G = r \times G$. Note that the matrices $\mathbf{L}$ and $\mathbf{F}$ are large and sparse, where $\tilde{\mathbf{L}}$ and $\tilde{\mathbf{F}}$ are dense and small. The key point is that this system has a size of r_G that is, in general, significantly smaller than the original dimension of the system n. Hence, this reduced system should be cheaper to solve, and once the eigenvector $\mathbf{a}$ is computed, the full-order flux can be reconstructed using Eq. 3.4.

3.3 Parameterized ROM for the eigenvalue Diffusion Equation

Although the system in Eq. 3.6 has significantly reduced dimensions compared to the original model in Eq. 3.3, there remains a computational challenge in solving the ROM for studies that require varying the model parameters. In particular, since the matrices $\mathbf{L}$ and $\mathbf{F}$ are parameter-dependent, computing the projected matrices $\tilde{\mathbf{L}}$ and $\tilde{\mathbf{F}}$ needs to be performed at each new parameter value.

The projection operations (i.e, $\mathbf{U}^T\mathbf{L}(\boldsymbol{\mu})\mathbf{U}$ and $\mathbf{U}^T\mathbf{F}(\boldsymbol{\mu})\mathbf{U}$) have a computational cost that is proportional to the original dimension of the system n. Each projection requires a number of flops with $\mathcal{O}(N^2 \times r + r^2 \times N)$, which makes the ROM inefficient for large problems. Therefore, it is desirable to constitute an offline/online decomposition of the ROM. The

projection is performed in the offline stage while, in the online stage, an inexpensive system is solved for each parameter point. The matrix version of the DEIM discussed in Sec. 2.4.1, can be exploited to improve the ROM efficiency for multi-query problems. First, the matrices $\mathbf{L}$ and $\mathbf{F}$ are serialized as $\mathbf{l} \in \mathbb{R}^{N1}$ and $\mathbf{f} \in \mathbb{R}^{N2}$ respectively, where $N1$ and $N2$ are the number of nonzero elements in $\mathbf{L}$ and $\mathbf{F}$, respectively. By applying the DEIM procedures discussed in algorithm 3, $\mathbf{l}$ and $\mathbf{f}$ are approximated as

$$\mathbf{l}(\boldsymbol{\mu}) = \mathbf{U}_l \mathbf{b}(\boldsymbol{\mu}) \,, \tag{3.7a}$$

$$\mathbf{f}(\boldsymbol{\mu}) = \mathbf{U}_f \mathbf{c}(\boldsymbol{\mu}) \,, \tag{3.7b}$$

where $\mathbf{U}_l \in \mathbb{R}^{N1 \times r_l}$ and $\mathbf{U}_f \in \mathbb{R}^{N2 \times r_f}$ are the POD subspaces approximating $\mathbf{l}$ and $\mathbf{f}$ with ranks r_l and r_f, respectively. The POD basis are generated using snapshots of the serialized operators at different reactor states. The coefficient vectors $\mathbf{b} \in \mathbb{R}^{r_l}$ and $\mathbf{c} \in \mathbb{R}^{r_f}$ are computed in an online stage for each new parameter value ($\boldsymbol{\mu}$) by solving

$$\mathbf{P}_l^T \mathbf{U}_l \mathbf{b}(\boldsymbol{\mu}) = \mathbf{l}(\boldsymbol{\mu}) \tag{3.8a}$$

$$\mathbf{P}_f^T \mathbf{U}_f \mathbf{c}(\boldsymbol{\mu}) = \mathbf{f}(\boldsymbol{\mu}) \,, \tag{3.8b}$$

where $\mathbf{P}_l \in \mathbb{R}^{N1 \times r_l}$ and $\mathbf{P}_f \in \mathbf{R}^{N2 \times r_f}$ are the interpolation matrices computed following algorithm 3. By remapping each vector in $\mathbf{U}_l$ and $\mathbf{U}_f$ into a matrix, $\mathbf{L}$ and $\mathbf{F}$ can be expressed as

$$\mathbf{L}(\mu) = \sum_{i=0}^{r_l} \mathbf{L}_i b_i(\boldsymbol{\mu}) \,, \tag{3.9a}$$

$$\mathbf{F}(\mu) = \sum_{i=0}^{r_f} \mathbf{F}_i c_i(\boldsymbol{\mu}) \,, \tag{3.9b}$$

where $\mathbf{L}_i$ and $\mathbf{F}_i$ are $N \times N$ matrices. Hence, the projection can be performed as

$$\sum_{i=0}^{r_l} \underbrace{\mathbf{U}^T \mathbf{L}_i \mathbf{U}}_{\tilde{\mathbf{L}}_i} b_i(\boldsymbol{\mu}) = \frac{1}{k} \sum_{i=0}^{r_f} \underbrace{\mathbf{U}^T \mathbf{F}_i \mathbf{U}}_{\tilde{\mathbf{F}}_i} c_i(\boldsymbol{\mu}) \tag{3.10}$$

The small matrices $\tilde{\mathbf{L}}_i \in \mathbb{R}^{r_G \times r_G}$ and $\tilde{\mathbf{F}}_i \in \mathbb{R}^{r_G \times r_G}$ are computed once and stored in an offline stage, and for each new parameter value the coefficients $\mathbf{b}$ and $\mathbf{c}$ are obtained by solving Eq. 3.8.

For the rest of this chapter, we will refer to this model as the ROM-DEIM, while we refer to the model in Sec. 3.2 as the ROM.

3.4 Application

To explore the potential of the method described above, the IAEA PWR 2D benchmark[51] was used as an illustrative model with the main goal being to develop a parametric ROM that approximates both the flux distribution and the eigenvalue k_{eff} at different given model parameters.

The benchmark features two energy groups for two different fuel assemblies and reflector regions. The core contains 177 fuel assemblies and octant symmetry. The control rods are fully inserted in nine assemblies and partially inserted in four. The fuel assembly pitch is 20 cm. A schematic layout of the core is shown in Fig. 3.1. The corresponding cross sections are given in Table 3.1.

Region	D_1	D_2	$\Sigma_{1\to2}$	Σ_{a1}	Σ_{a2}	$\nu\Sigma_{f2}$	Material
1	1.5	0.4	0.02	0.01	0.08	0.135	Fuel 1
2	1.5	0.4	0.02	0.01	0.085	0.135	Fuel 2
3	1.5	0.4	0.02	0.01	0.13	0.135	Fuel 2 + Rod
4	2	0.3	0.04	0	0.01	0	Reflector

Table 3.1: Two group constants of the IAEA-2D benchmark

The code Detran was used to model the benchmark. For spatially discretizing the problem, the mesh-centered finite difference method was used, and the resultant system was solved using the *Arnoldi* method in *SLEPc* library, with a tolerance of 1×10^{-14} and a maximum number of iterations of 1000. For a spatial grid of 90×90, the eigenvalue using the nominal values of the cross-section in table 3.1 was 1.02948. The normalized flux distributions of the fast and the thermal groups are shown in Figs. 3.2 and 3.3, respectively.

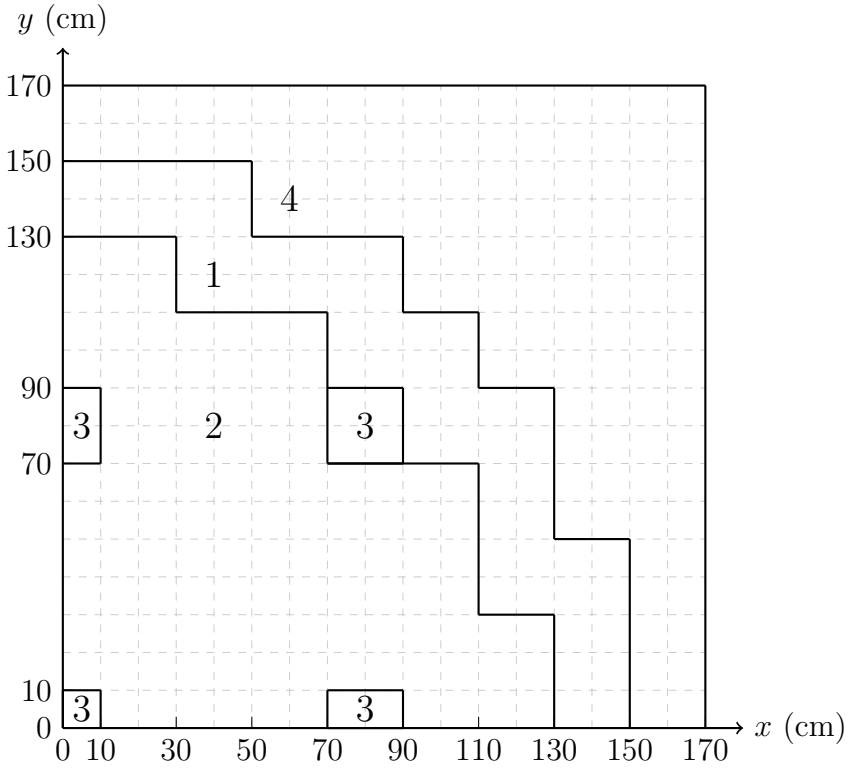


Figure 3.1: Geometry as modeled for the IAEA 2D diffusion benchmark.

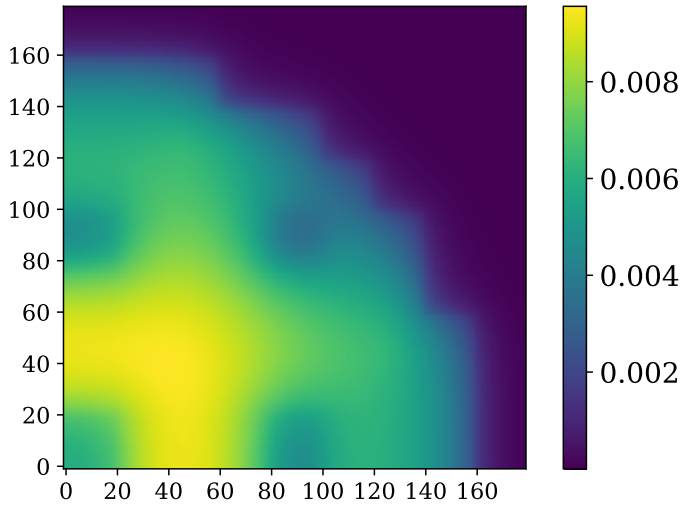


Figure 3.2: Fast flux distribution

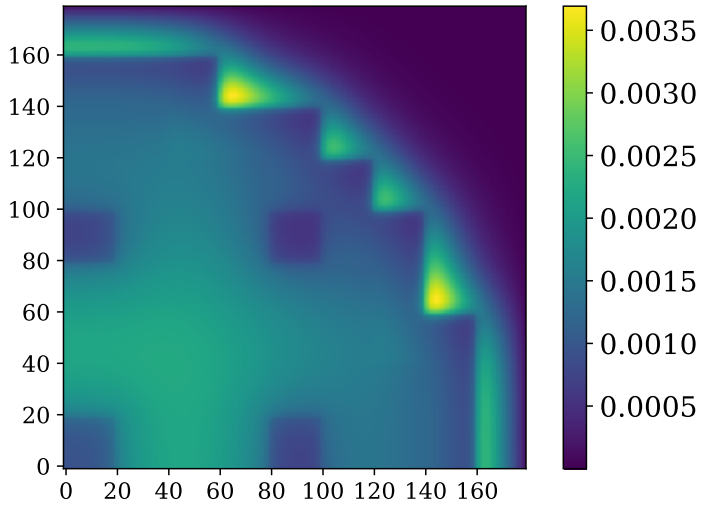


Figure 3.3: Thermal flux distribution

A parametric ROM was developed to approximate the eigenvalue and the group flux at different parameter points. The parametric ROM was developed with the aid of the MDEIM to obtain an offline/online decomposition of the ROM and hence improve its efficiency. Here, the parameters of interest were the cross sections which were assumed to be uniformly distributed about 2% of the nominal value given in Table 3.1.

3.4.1 ROM and ranks selection

Before constructing the parametric ROM, appropriate values for the POD rank and the DEIM orders of the problem operators need to be determined. The ranks are selected based on the flux relative error which is computed as

$$\text{flux relative error}(\%) = \frac{\|\mathbf{\Phi}_{\text{approx}} - \mathbf{\Phi}_{\text{fom}}\|}{\|\mathbf{\Phi}_{\text{fom}}\|} \times 100\,, \tag{3.11}$$

where $\|.\|$ denotes the L_2 norm, $\mathbf{\Phi}_{\text{fom}}$ is the solution of Eq. 3.3 and $\mathbf{\Phi}_{\text{approx}}$ is either the ROM or the ROM-DEIM approximation of the flux. First, snapshots of the group flux, at different cross section values, were generated using the FOM. For the four assemblies materials, 13 parameters were considered, where cross sections of the same assembly were perturbed with the same amount. These perturbed cross sections were generated randomly

using the specified distribution and are considered as the ROM training data. The SVD was computed for each energy group snapshots. The first five POD modes of both groups are shown in Fig. 3.4 where σ_i denotes the singular value of the ith mode where all the singular values are shown in Fig. 3.5. As might be expected, the first POD mode has the same shape as the flux distribution, which is the fundamental eigenvector.

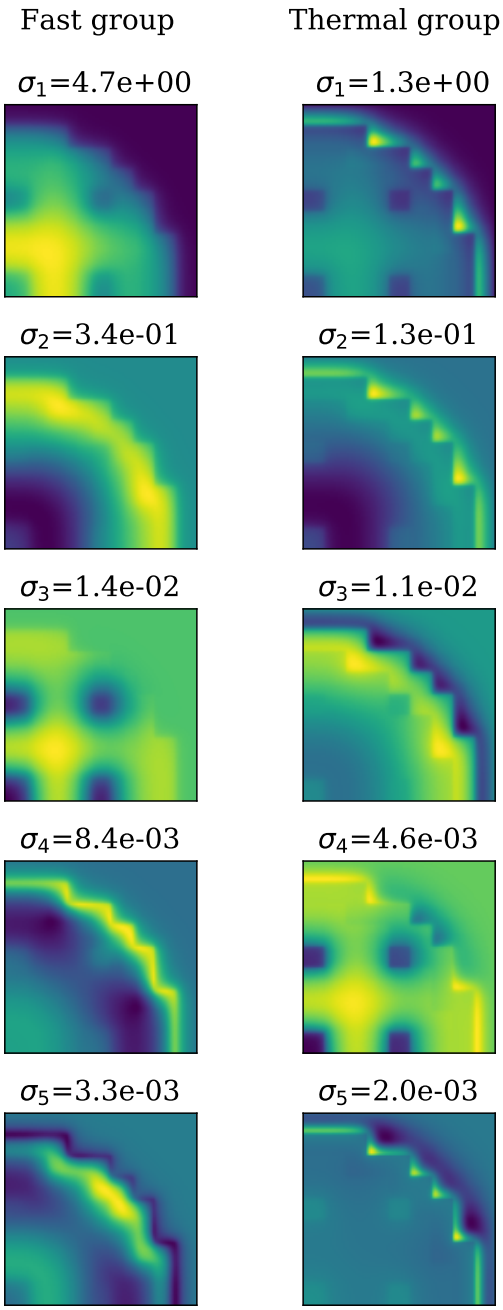


Figure 3.4: POD modes of the fast and thermal groups

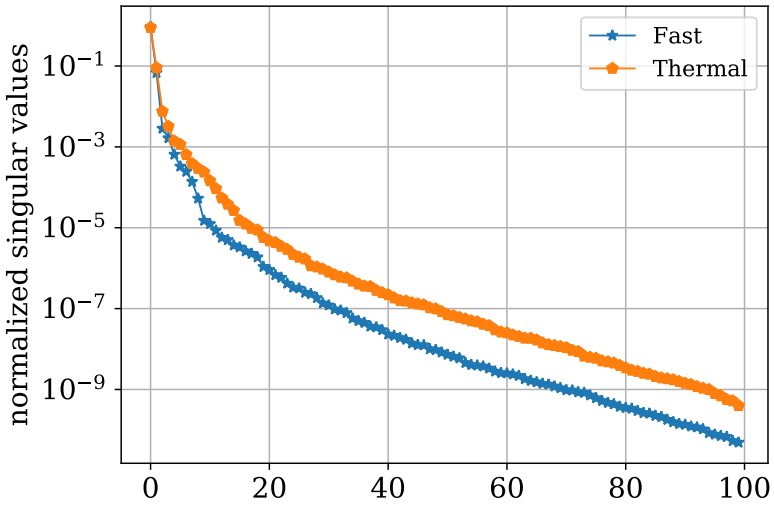


Figure 3.5: Singular values of the fast and thermal groups

In order to select the POD rank, Eq. 3.6 was solved with different POD ranks and the errors between the resultant flux and the FOM flux were computed using Eq. 3.11 and are shown in Fig. 3.6. Different mesh sizes were used which gives different problem sizes and increasing model fidelity. The legend in the figure denotes the spatial grids corresponding to different mesh sizes. Note that the 9×9 grid flux can not have rank higher than 81, (i.e, the number of spatial cells).

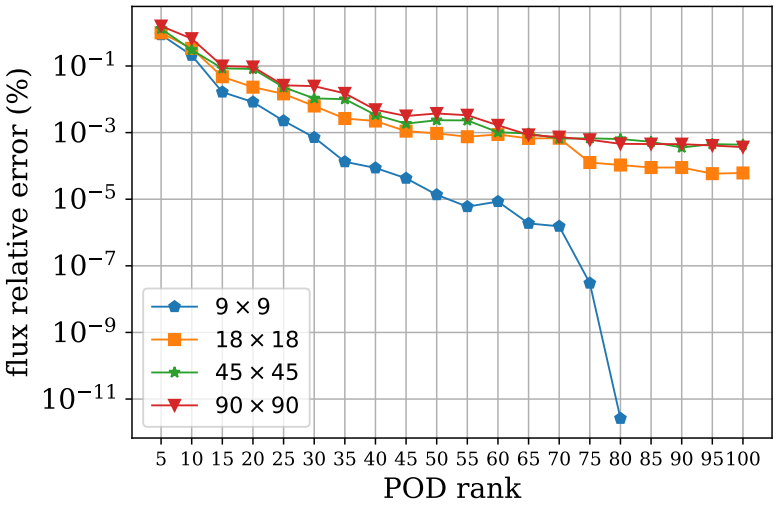


Figure 3.6: Relative L_2 norm of the flux error (%)

Based on the shown results, a rank of 30 was selected because it leads to an error that

is less than 0.01% for all cases. However, a higher rank can be chosen if a better accuracy is desired. Furthermore, since the parametric ROM is aimed to be constructed using the MDEIM for the **L** and **F** operators, ranks for both need to be selected. Hence, using the same training data, snapshots of the serialized matrices were generated, and the SVD was computed for each of them. The singular values are shown in Fig. 3.7.

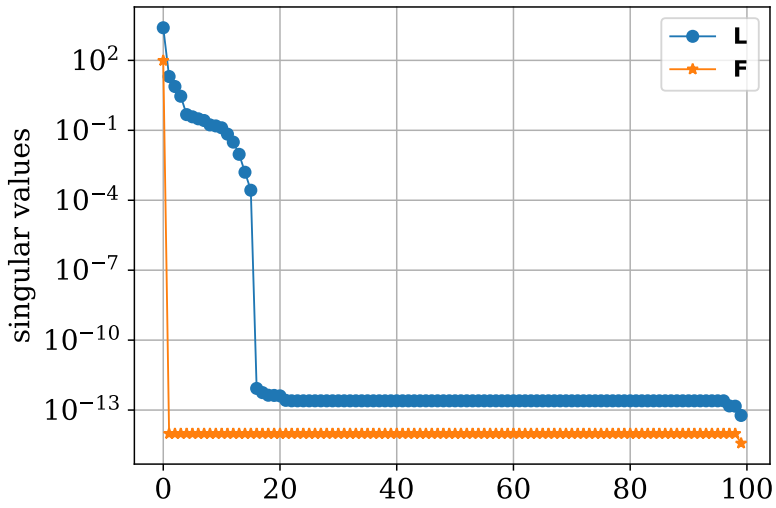


Figure 3.7: Singular-values of the serialized **L** and **F** operators

As can be seen, the singular values of **F** decreases instantaneously and based on this, an order of 1 was selected. Note that this selected low-order is not surprising since this is a very low-rank operator that contains only the fission cross section data; hence, its serialized form can be approximated by few basis vectors. On the other hand, the operator **L** contains more information including the boundary condition, leakage and other cross section data, hence it needs to be approximated with higher DEIM order. It was observed that there is a drastic decrease of the singular values after a rank of 16, and, hence a DEIM of order 16 was selected for this operator. The error resulting from approximating the operators using MDEIM of the selected orders was measured by computing the Frobenious norms of the errors for both operators (i.e, $\|\mathbf{U}^T\mathbf{L}\mathbf{U} - \sum_{i=0}^{r_l} \mathbf{U}^T\mathbf{L}_i\mathbf{U}\|_F$) and they were found to be in the order of 1×10^{-15} and 1×10^{-12} for **F** and **L**, respectively.

3.4.2 Parametric ROM

The MDEIM procedures discussed in Sec. 3.3 were used to build the parametric ROM. Thus, a ROM-DEIM was built with a POD of rank 30 and orders of 16 and *one* for the **L** and **F** operators, respectively. Moreover, to assess its performance and accuracy, a ROM was constructed without using the DEIM, i.e, by performing the projection at each new parameter point. For testing, 100 different cross section samples were generated and the corresponding group flux and k_{eff} resulting from the three models, i.e, FOM, ROM and ROM-DEIM, were obtained. The mean error of all samples is computed for each case as

$$\text{mean error}(\%) = \frac{1}{m}\sum_{i=0}^{m}\frac{\|\mathbf{\Phi}_{g_{\text{approx.}}} - \mathbf{\Phi}_{g_{\text{fom}}}\|}{\|\mathbf{\Phi}_{g_{\text{fom}}}\|} \times 100\,, \tag{3.12}$$

where m is the size of the testing sample. The computed mean errors of the thermal and fast groups of the ROM-DEIM are shown in Fig. 3.8, where there was no significant difference between these errors and the errors of the ROM.

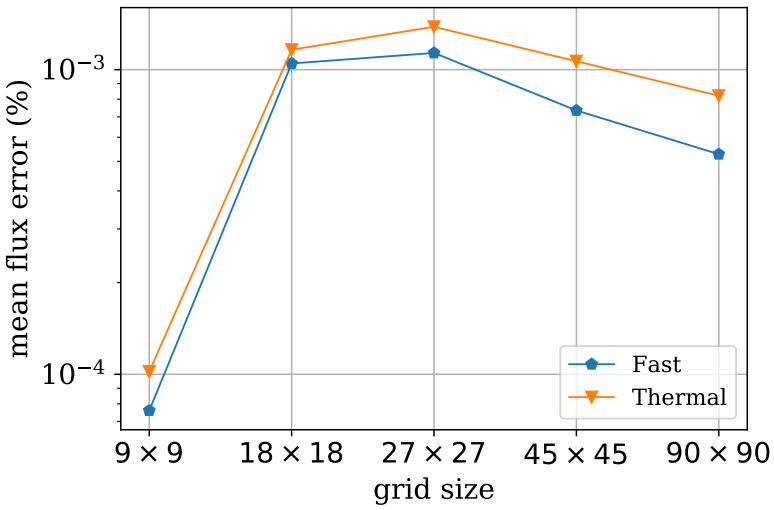


Figure 3.8: Mean relative error of the thermal flux

Furthermore, the error in the predicted k_{eff} was computed and the errors distribution is shown in Fig. 3.9.

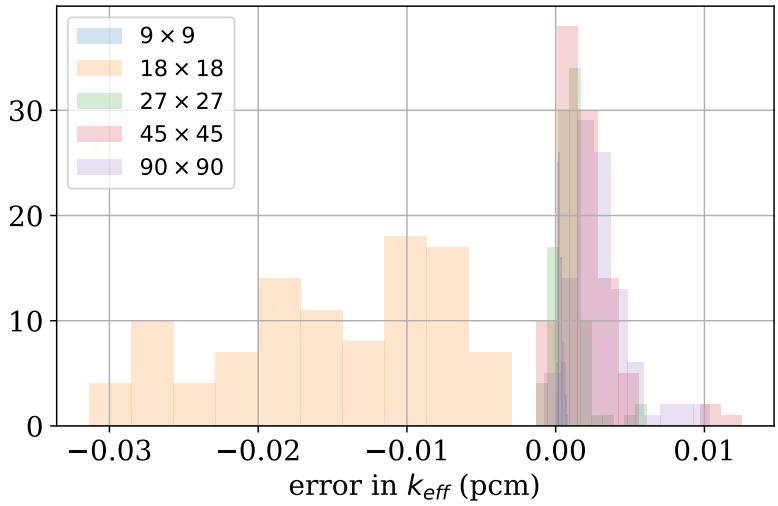


Figure 3.9: Error distribution of the predicted k_{eff} using the ROM-DEIM

Also, the maximum errors across the sample are shown in Table 3.2.

grid size	fast flux (%)	thermal flux (%)	k_{eff} (pcm)
9×9	0.0001	0.0002	0.0008
18×18	0.0024	0.0025	0.0290
27×27	0.0017	0.0020	0.0023
45×45	0.0012	0.0017	0.0054
90×90	0.0011	0.0018	0.0068

Table 3.2: Maximum errors of the flux and the k_{eff}

As for the computational efficiency, the CPU time of the three models was obtained and is shown in Fig. 3.10. Clearly, the ROM cost increases with increasing the number of spatial cells since its cost is proportional to the problem dimension. On the other hand, the ROM-DEIM cost is almost constant and is independent of the problem dimension. Of course, there is an upfront offline cost. However, the key point is that once this offline stage is done, a large sample size can be used without any computational burden.

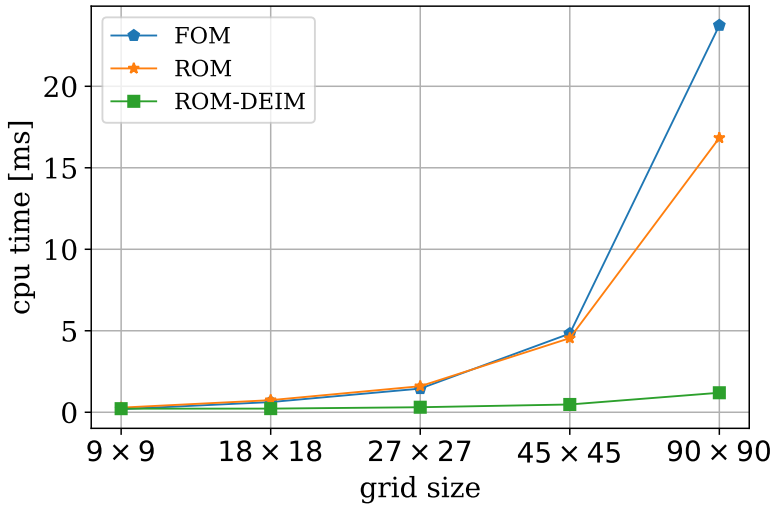


Figure 3.10: CPU time of the models

It was interesting to see the FOM error resulting from using different mesh sizes. As explained in the introductory chapter, a FOM with a coarse mesh can be considered as a surrogate model that is used to obtain a cheaper approximation of a FOM with a finer mesh. The error in k_{eff} was computed between FOMs with grids 45×45 and 90×90 for the testing sample and its distribution is shown in Fig. 3.11. As can be seen, the resulting distribution indicates that the coarser mesh model results in an error in k_{keff} that is higher than the ROM wih 90×90 grid. This suggests that coarse mesh models might not be the best way to perform efficient multi-query studies if good accuracy is desired. It also suggests that ROMs could provide better accuracy while retaining the model fidelity.

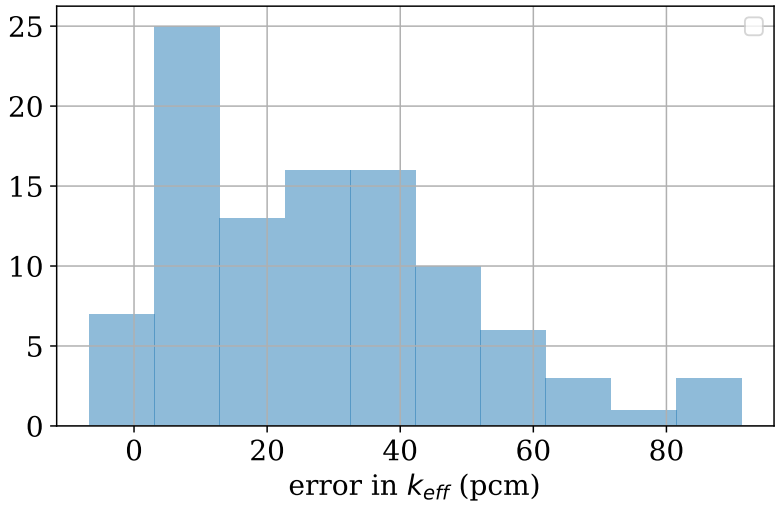


Figure 3.11: k_{eff} error distribution of the FOM with different mesh sizes

Another case was tested in which all assemblies materials were treated as independent materials even if they belonged to the same assembly type, leading to a highly heterogeneous problem with 314 different parameters. For this case, only a grid of 90×90 was used with 200 samples generated for obtaining the POD and DEIM basis. The singular values of the group fluxes and the problem operators are shown in Figs. 3.12 and 3.13 respectively.

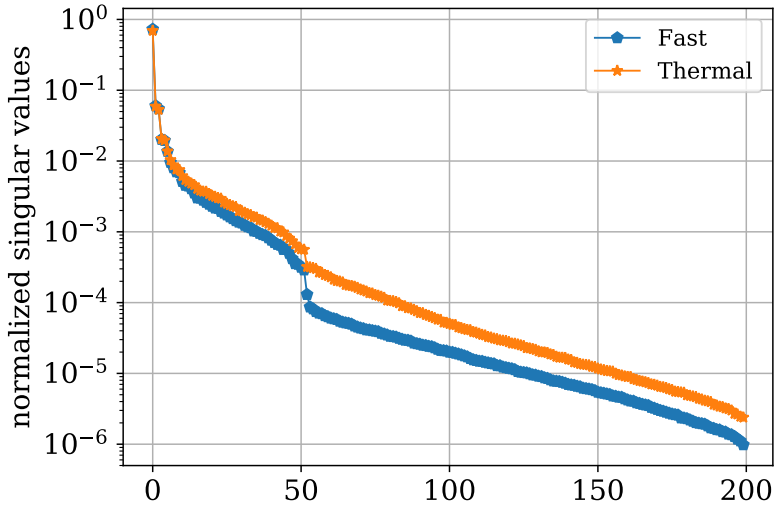


Figure 3.12: singular values of the group fluxes

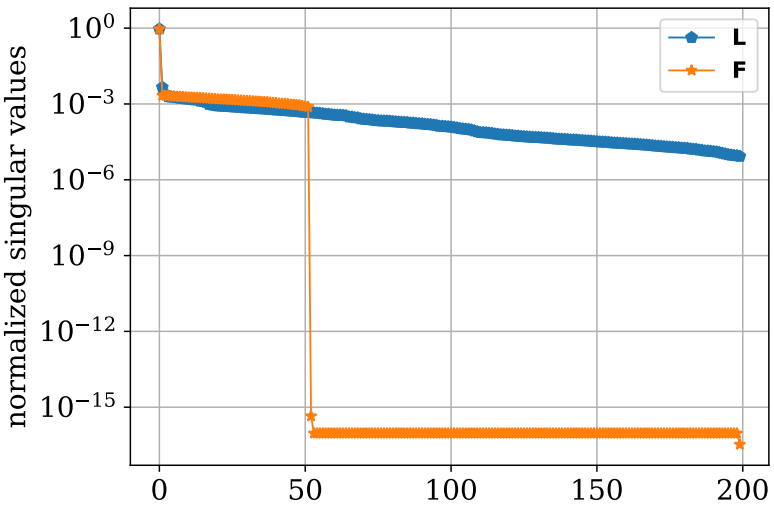


Figure 3.13: singular values of the problem operators

Unlike the previous case, the singular values do not show a rapid decay because of the increasing level of spatial heterogeneity. Note that for the **F** operator, the singular values drop to a value equivalent to zero after 52, which is the number of the assemblies in the core excluding the reflector. A ROM-DEIM was built with a POD basis of rank 50 and order 52 for both **F** and **L** operators. For testing, 200 samples were used and the distribution of the resulting prediction error of k_{eff} is shown in Fig. 3.14. The maximum errors across all samples of the fast and thermal flux were 29% and 25% respectively. Moreover, a ROM was built with the same POD basis and the resulting error of the predicted k_{eff} is shown in Fig. 3.15. The significant difference between the shown errors of the two models suggests a limitation of using the DEIM for problems with operators of large degrees of freedom. Note that increasing the DEIM order should reduce this error, however it poses another challenge with respect to memory requirement since it will result in storing a large number of matrices in the offline stage. It is important to note that the correlation between different cross sections has been ignored in this study. Taking the correlation into account should result in operators with lower degrees of freedom and accordingly increase the potential of obtaining a good approximation of the operator using DEIM of low order.

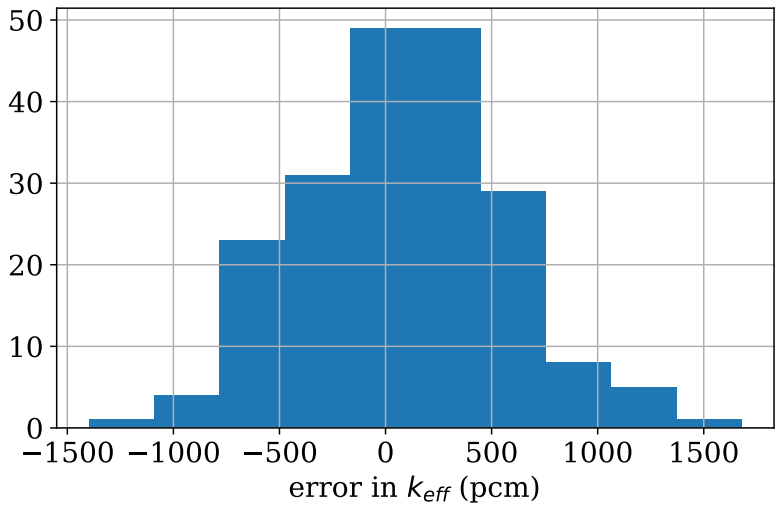


Figure 3.14: Error distribution of the predicted k_{eff} using the ROM-DEIM (case 2)

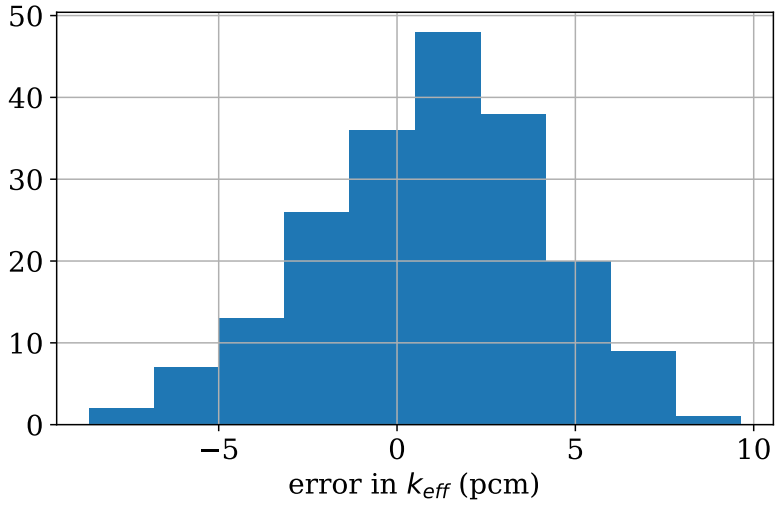


Figure 3.15: Error distribution of the predicted k_{eff} using the ROM (case 2)

Chapter 4

Time Dependent Problems

4.1 Time-Dependent Diffusion Equation

The steady-state diffusion equation discussed in the previous chapter assumes that the neutron population is constant with respect to time. In other words, there exists a balance between the neutrons loss due to absorption and leakage, and the neutrons gain from fission. However, there are cases in which this balance is no longer maintained and accordingly, the neutron population exhibits a time dependence. These cases occur for various reasons with different time scales. Examples include fuel burnup, control rod movement, reactor start up and shutdown, and various accidental conditions. The time-dependent diffusion equation is used to predict the time behavior of the neutron population following a perturbation in the reactor core. In such calculations, the delayed neutrons must be accounted for, in contrary to the steady-state version, where all the fission neutrons are assumed to be promptly emitted at the time of fission. For this purpose, the delayed precursors concentration balance equation is coupled with the flux equation. Thus, the multi-group time-dependent diffusion equation, for G energy groups and I precursors groups, can be written in the following form:

$$\frac{1}{v_g}\frac{\partial \phi_g}{\partial t} = \nabla . D_g \nabla \phi_g - \Sigma_{tg}\phi_g + \sum_{g'=1}^{G} \Sigma_{sg'g}\phi_{g'} + \chi_g(1-\beta)\sum_{g'=1}^{G} \nu\Sigma_{fg'}\phi_{g'} + \chi_g \sum_{i=1}^{P} \lambda_i C_i \,, \tag{4.1}$$

where

$$\frac{\partial C_i}{\partial t} = -\lambda_i C_i + \beta_i \sum_{g'=1}^{G} \nu\Sigma_{fg'}\phi_{g'} \,,$$

and ϕ_g, D_g, Σ_{ag}, $\nu\Sigma_{fg}$ and v_g are the neutron flux, diffusion coefficient, the absorption cross section, the production cross section and the average neutron velocity in group g, respectively. $\Sigma_{sg'g}$ is the scattering cross section from group g' to group g and χ_g is the probability that a fission neutron will be born in group g. C_i is the ith precursor concentration, β_i is the fraction of delayed neutrons in group i, λ_i is the ith group decay constant. The system can be represented compactly in a matrix form as

$$\frac{\partial y(r,t)}{\partial t} = \mathcal{A} y(r,t) \,, \tag{4.2}$$

where the stacked unknowns are

$$y = \begin{pmatrix} \phi_1(r,t) \\ \vdots \\ \phi_G(r,t) \\ C_1(r,t) \\ \vdots \\ C_I(r,t) \end{pmatrix} .$$

and the system matrix is

$$\mathcal{A} = \left(\begin{array}{cccc|ccc} v_1(\nabla\cdot D_1\nabla+\Sigma_{11}) & v_1\Sigma_{12} & \dots & v_1\Sigma_{1G} & v_1 f_{11} & \dots & v_1 f_{1I} \\ v_2\Sigma_{21} & v_2(\nabla\cdot D_2\nabla+\Sigma_{22}) & \dots & v_2\Sigma_{2G} & \vdots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \vdots & v_G f_{G1} & \dots & v_G f_{GI} \\ v_G\Sigma_{G1} & v_G\Sigma_{G2} & \dots & v_G(\nabla\cdot D_G\nabla+\Sigma_{GG}) & & & \\ \hline p_{11} & p_{12} & \dots & p_{1G} & -\lambda_1 & \dots & 0 \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ p_{I1} & p_{I2} & \dots & p_{IG} & 0 & \dots & -\lambda_I \end{array}\right)$$

where

$$\Sigma g g' = (1-\beta)\chi_g \nu\Sigma f g' + \Sigma_{s_{gg'}}\,,$$

$$f_{ig'} = \chi_g \lambda_i\,,$$

and

$$p_{ig} = \beta_i \nu\Sigma_{fg}\,.$$

In a block matrix form, the operator $\mathcal{A}$ can be written as

$$\mathcal{A} = \left(\begin{array}{c|c} \mathcal{L} & \mathcal{D} \\ \hline \mathcal{F} & \mathcal{P} \end{array}\right)\,. \tag{4.4}$$

Discretizing the system in space yields a system of ODEs, or

$$\frac{d\mathbf{y}(t)}{dt} = \mathbf{A}\mathbf{y}(t)\,, \tag{4.5}$$

where $\mathbf{y} \in \mathbb{R}^N$, $\mathbf{A} \in \mathbb{R}^{N \times N}$ and $N = (I + G) \times n$ with n is the number of spatial unknowns arising from discretization. The discretized block matrix $\mathbf{A}$ is written as

$$\mathbf{A} = \left(\begin{array}{c|c} \mathbf{L} & \mathbf{D} \\ \hline \mathbf{F} & \mathbf{P} \end{array}\right) . \tag{4.6}$$

where $\mathbf{L} \in \mathbb{R}^{(n \times G) \times (n \times G)}$, $\mathbf{D} \in \mathbb{R}^{(n \times G) \times (n \times I)}$, $\mathbf{P} \in \mathbb{R}^{(n \times I) \times ((n \times G)}$ and $\mathbf{F} \in \mathbb{R}^{(n \times I) \times ((n \times I)}$. Thus, the system in Eq. 4.5 is the full-order model and it can be solved using a suitable time-scheme. However, it is expensive to solve such systems for fine spatial and time grids, since a linear system of a size proportional to the spatial discretization has to be solved for each time step. Hence, a reduced-order model to approximate it is justified. In the following section, the POD-Galerkin ROM procedures to approximate this problem are illustrated.

4.2 POD-Galerkin Model for the time dependent diffusion equation

In order to build a POD-Galerkin ROM for the system in Eq. 4.5, POD basis sets need to be generated for the group flux and the precursor concentrations. The total flux can be represented as

$$\mathbf{\Phi} \approx \mathbf{U}_\phi \mathbf{a}_\phi(t) \,, \tag{4.7}$$

where $\mathbf{U}_\phi \in \mathbb{R}^{(n \times G) \times r}$ is a POD basis of rank r and $\mathbf{a}_\phi$ is the associated temporal coefficients vector. Similarly, the precursors concentration is approximated as

$$\mathbf{C} \approx \mathbf{U}_c \mathbf{a}_c(t) \,, \tag{4.8}$$

where $\mathbf{U}_c \in \mathbb{R}^{(n \times I) \times r_c}$ is a POD basis of rank r_c and $\mathbf{a}_c$ is a vector of the associated temporal coefficients. The basis can be generated such that all flux/precursors groups are collapsed together using one basis set in which case snapshots of all groups are stacked in one matrix for

which the SVD is computed. Alternatively, each flux/precursor group can be approximated by a separate POD basis set. For example, the gth group flux is approximated as $\mathbf{\Phi}_g = \mathbf{U}_{\phi_g}\mathbf{a}_g(t)$, where $\mathbf{U}_{\phi_g} \in \mathbb{R}^{n\times r_\phi}$ is the POD basis of rank r_ϕ of the gth group and $\mathbf{a}_g \in \mathbb{R}^{r_\phi}$ is the associated temporal coefficients vector. For simplicity, the same POD rank is used for all groups. In such case the flux basis is constructed as

$$\mathbf{U}_\phi = \begin{pmatrix} \mathbf{U}_{\phi_1} & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \mathbf{U}_{\phi_G} \end{pmatrix},$$

and the coefficients vector

$$\mathbf{a}_\phi = \begin{pmatrix} \mathbf{a}_1 \\ \vdots \\ \mathbf{a}_G \end{pmatrix}.$$

The precursor groups can be approximated in a similar manner. The next step is to substitute Eq. 4.7 and Eq. 4.8 into Eq. 4.5 and project the system onto the POD subspace by multiplying it on the left by the transpose of the subspace matrix. To compute the projected matrix, the operator $\mathbf{A}$ does not have to be constructed explicitly. Rather, each sub-matrix is projected onto the appropriate basis, and the reduced operator $\tilde{\mathbf{A}}$ is then constructed by stacking these projected sub-matrices together. The upper sub-matrices $\mathbf{L}$ and $\mathbf{F}$ arise from discretizing the flux equation, hence they are multiplied on the left by $\mathbf{U}_\phi$. On the other hand, $\mathbf{F}$ and $\mathbf{P}$ result from discretizing the precursor concentration equation, and are multiplied on the left by $\mathbf{U}_c$. The projected submatrices are computed as follows

$$\tilde{\mathbf{L}} = \mathbf{U}_\phi^T \mathbf{L} \mathbf{U}_\phi \,,$$

$$\tilde{\mathbf{D}} = \mathbf{U}_\phi^T \mathbf{F} \mathbf{U}_c \,,$$

$$\tilde{\mathbf{P}} = \mathbf{U}_c^T \mathbf{P} \mathbf{U}_C \,,$$

$$\tilde{\mathbf{F}} = \mathbf{U}_c^T \mathbf{F} \mathbf{U}_\phi \,,$$

where $\tilde{\mathbf{L}} \in \mathbb{R}^{r_\phi \times r_\phi}$, $\tilde{\mathbf{D}} \in \mathbb{R}^{r_\phi \times r_c}$, $\tilde{\mathbf{F}} \in \mathbb{R}^{r_c \times r_\phi}$ and $\tilde{\mathbf{P}} \in \mathbb{R}^{r_c \times r_c}$. The projected operator $\tilde{\mathbf{A}} \in \mathbb{R}^{(r_\phi + r_c) \times (r_\phi + r_c)}$ is constructed as

$$\tilde{\mathbf{A}} = \left(\begin{array}{c|c} \tilde{\mathbf{L}} & \tilde{\mathbf{D}} \\ \hline \tilde{\mathbf{F}} & \tilde{\mathbf{P}} \end{array}\right) \tag{4.10}$$

Thus, the POD-Galerkin ROM approximating the system is represented as

$$\frac{d\mathbf{a}(t)}{dt} = \tilde{\mathbf{A}}\mathbf{a}(t)\,, \tag{4.11}$$

where,

$$\mathbf{a}(t) = \begin{pmatrix} \mathbf{a}_\phi(t) \\ \mathbf{a}_c(t) \end{pmatrix}\,.$$

The initial condition of the temporal coefficients $\mathbf{a}(t)$ is naturally defined as

$$\mathbf{a}_{\phi_0} = \mathbf{U}_\phi^T \mathbf{\Phi}_0\,,$$

$$\mathbf{a}_{c_0} = \mathbf{U}_c^T \mathbf{C}_0\,,$$

where $\mathbf{\Phi}_0$ and $\mathbf{C}_0$ are the initial conditions of the flux and the precursors concentration, respectively. Similar to the FOM, the system in Eq. 4.11 can be solved using any time-scheme. However, compared to the FOM in Eq. 4.5, significantly fewer equations need to be solved at each time step which makes it computationally cheaper. Once the reduced solution is obtained, the approximate full-order solutions of the flux and the precursors can be computed using Eqs. 4.7 and 4.8, respectively.

Computational complexity can be encountered in solving the ROM in Eq. 2.9 when the operator $\mathbf{A}$ is time-dependent, since the corresponding reduced operator $\tilde{\mathbf{A}}$ has to be updated at each time step. Because computing this reduced operator has a cost proportional to the dimension of the system, it contributes significantly to the overall expense of the ROM. Two approaches can be employed to avoid the direct projection of the operator at each time

step; the first is an approximate-then-project approach, and an example of a method that belongs to this category is the matrix version of DEIM, presented in Sec.2.4.1. Alternatively, a project-then-approximate approach can be used. An example of such methods is to use interpolation to approximate the elements of the reduced operators. In problems that do not include feedback, this can be done by using pre-computed matrices at specific time instances. However, this is challenging in problems with feedback since the original operator elements, i.e., cross sections, depend on the solution itself. Instead, on-the-fly interpolation can be employed based on the assumption that cross sections change with time considerably slower than the flux evolution. This method can be done by using two time steps, i.e., macro time step and micro time step. The micro time step is the one that is used to integrate the system, and the macro step is a coarser one, where the macro-to-micro ratio is problem-dependent. The projection is restricted to be performed at each macro time step. The projected operator elements are approximated at the micro time steps by interpolating the matrices generated at the macro step mesh, which are updated to account for the feedback coupling.

4.3 Applications

Two time-dependent applications were studied. The first is a 1-D problem, developed specifically to explore the potential of different ROM techniques to approximate neutronic transients. Moreover, the problem is used to understand the impact of the POD group flux basis generation on the ROM prediction accuracy. In particular, two approaches were adopted to generate the flux basis. In the first approach, a single basis set was obtained by collapsing all energy groups together. In the second approach, separate POD basis sets are generated for each energy group. In both cases, the snapshots were the time-dependent group flux corresponding to the transient being studied. The result of this study led us to focus on the POD-Galerkin model and extend it to study higher dimensional problems and also to parameterize the ROM so that it can be used efficiently for multi-query applications. The two applications are given in the next two subsections with their numerical results.

4.3.1 1-D Time Dependent Diffusion

The model problem considered is a slab reactor, as shown in Fig. 4.1, where R indicates a reflector and A_i specifies the ith assembly. All of the assemblies are of the same fuel type, but A_2 and A_4 are control locations. Listed in Table 4.1 are the two-group constants for the reflector, fuel with no control inserted, and fuel with control fully inserted. Linear interpolation is used to compute Σ_{a2} value at intermediate control positions. Precursor data are shown in Table 4.2.

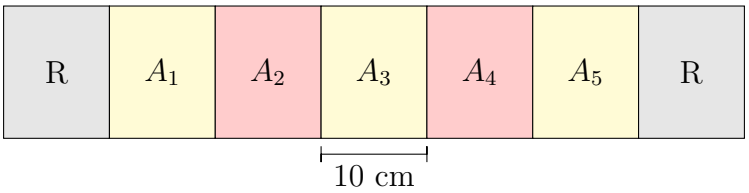


Figure 4.1: 1-D reactor model

The transient begins in steady state with both control elements 25% withdrawn. After 2 seconds, the right control (in A_4) is removed to 30% withdrawn over a period of 2 seconds at a constant rate. This configuration is maintained for 6 seconds. At 10 seconds, the right control is inserted to the 25% position over a period of 2 seconds at a constant rate. The reactor power is followed for an additional 50 seconds, making a total of 62 seconds. To solve this problem numerically, a mesh-centered, finite-difference discretization was used in space. For the time discretization, the backward-Euler method was used, leading to

$$\mathbf{y}^{k+1}(\mathbf{I} - \mathbf{A}\Delta t) = \mathbf{y}^{k}\,, \tag{4.13}$$

where k denotes the time step and Δt is the time step size. A spatial step of 0.2 cm and temporal step of 0.1 s were used for all calculations. Shown in Fig. 4.2 is the FOM power computed using the resulting two group flux.

material	D_1 (cm)	D_2 (cm)	Σ_{a1} (cm^{-1})	Σ_{a2} (cm^{-1})	$\Sigma_{s2\leftarrow1}$ (cm^{-1})	$\nu\Sigma_{f1}$ (cm^{-1})	$\nu\Sigma_{f2}$ (cm^{-1})
reflector	1.5000	0.5000	0.0002	0.0100	0.0320	0.0000	0.0000
fuel (all out)	1.3000	0.5000	0.0105	0.1140	0.0220	0.0030	0.1900
fuel (all in)	1.3000	0.5000	0.0105	0.1640	0.0220	0.0030	0.1900

Table 4.1: Two group constants of the 1-D reactor problem.

group i	β_i	λ_i (s^{-1})
1	2.18×10^{-4}	1.2467
2	1.02×10^{-3}	2.8292×10^{-2}
3	6.05×10^{-4}	4.2524×10^{-2}
4	1.31×10^{-3}	1.33042×10^{-1}
5	2.20×10^{-3}	2.92467×10^{-1}
6	6.00×10^{-4}	6.66488×10^{-1}
7	5.40×10^{-4}	1.634781
8	1.52×10^{-4}	3.554601

Table 4.2: Eight-group, delayed-neutron precursor constants of the 1-D reactor problem.

.

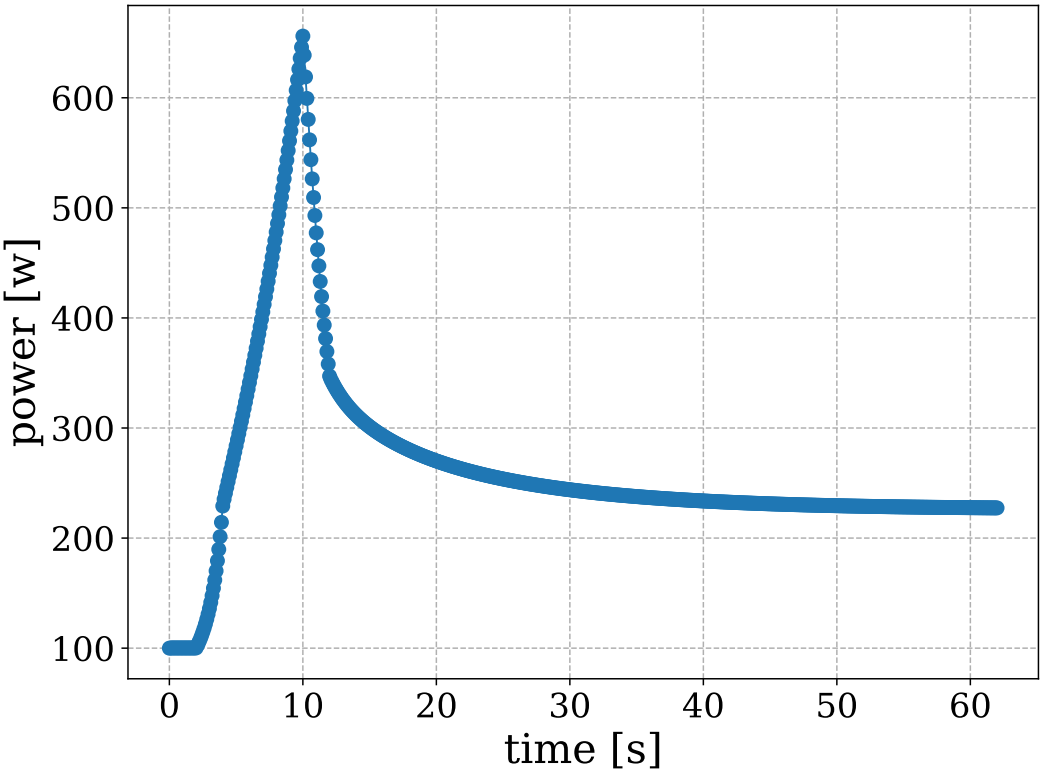


Figure 4.2: 1-D reactor core power

Numerical Results

First, the above model was used to carefully select the basis generation method that results in a more accurate ROM. Hence, two POD-Galerkin ROMs were developed, one using a single basis set of rank 10 for the two energy groups and the second using a group-wise basis sets, each of rank 5, computed from the corresponding group flux snapshots. Eq. 4.11 was solved with the same time scheme used for the full-order model. Shown in Fig. 4.3 are the relative errors in the predicted power using the two ROMs. It is observed that the ROM

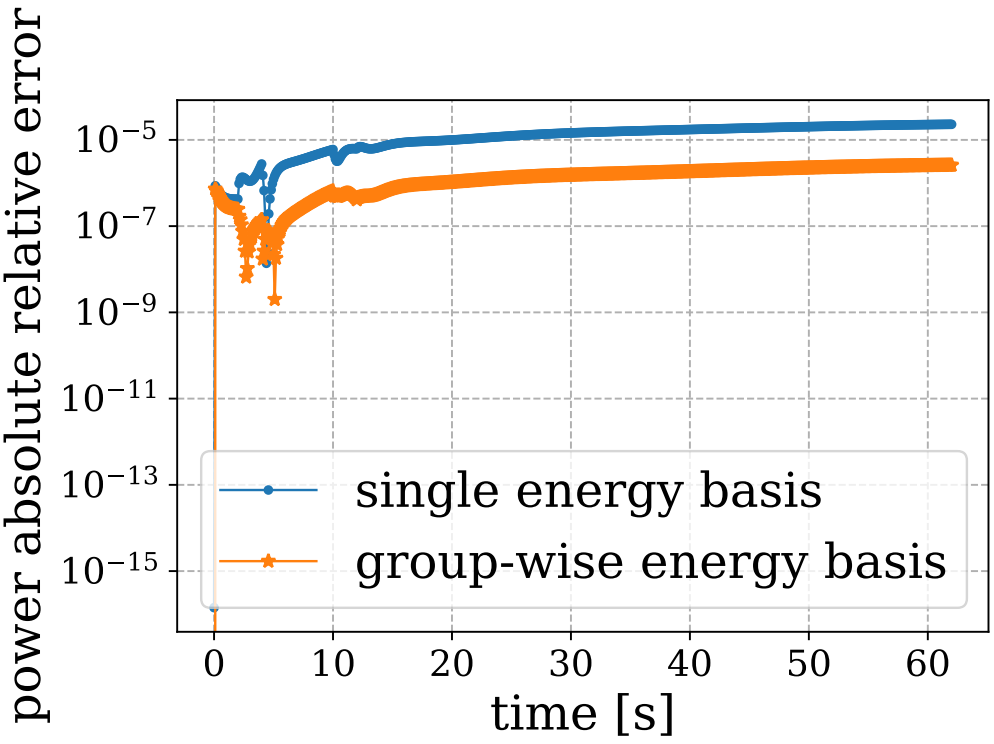


Figure 4.3: Relative errors in power predictions with different group basis

with a group-wise energy basis gives a better accuracy, implying that the variation between the group fluxes is best captured by separate basis sets. Hence, for the rest of applications discussed in this thesis, a POD-Galerkin ROM was developed with group-wise POD basis sets. Moreover, we will show results of two other ROM methods developed for the same problem. The two methods are the DMD-Galerkin projection and the data-driven DMD. For the Galerkin-projection models, a basis set was generated for each group flux while one basis set was generated for all precursor groups. For the POD and DMD bases, 20 modes were

used. For data-driven DMD, snapshots of the Quantities of interest (QoIs) were stacked into one matrix to build a DMD surrogate with 45 DMD modes. While the error for POD always decreases when more basis functions are included, this is not the case for DMD, since the basis vectors are not orthogonal. Hence, different ranks were tested, and the one producing the smallest error was used. Shown in Table 4.3 is the relative L_2 error in the predictions of the group flux and the precursors concentration, averaged over space and time. The group fluxes were used to compute the time-dependent core power, and the resulting prediction errors are shown in Figure 4.4.

Table 4.3: Relative L_2 error in the Quantities of Interest (%)

Model	Fast flux	Thermal flux	Precursor concentration
POD-Galerkin	3.75×10^{-5}	3.8×10^{-5}	0.00026
DMD-Galerkin	0.025	0.025	0.024
DMD	1.5	1.49	31.4

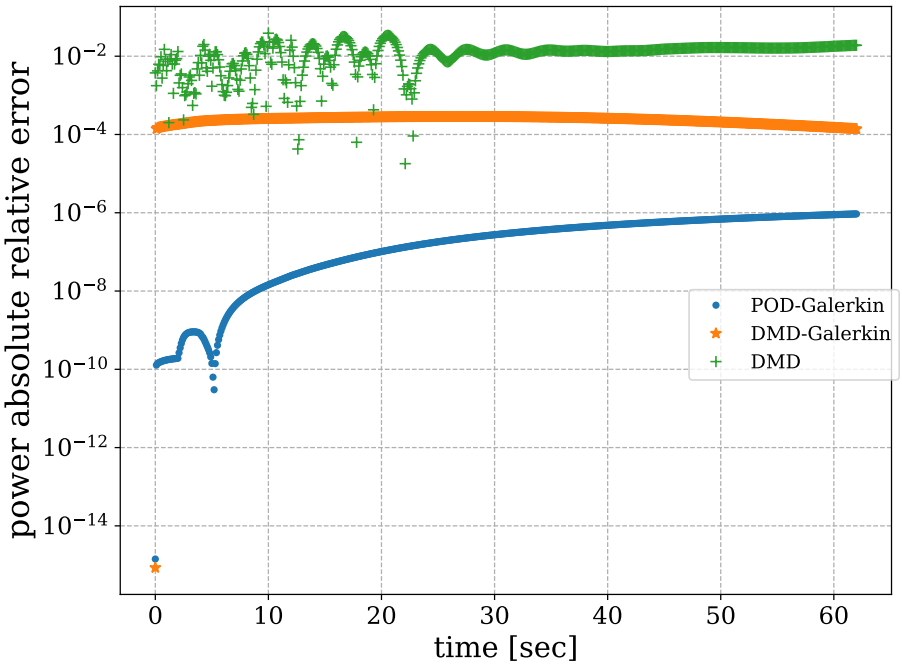


Figure 4.4: Relative errors in power predictions by the surrogates

As shown, the POD-Galerkin projection always performs better than the other two methods. These results were the main reason that we proceeded with the POD-Galerkin

model for more complex problem and also for exploring the potential of approximating problems that exhibit non-linearity resulting from other physics feedback as shown in the next chapter.

4.3.2 2-D Transient

Here, the TWIGL benchmark was used to illustrate the POD-Galerkin projection model on a more complex problem. The code Detran[10] served as the simulation that has the full-order model implemented. Some intrusive modifications were introduced to the existing code in order to perform the projection of the problem operator. The benchmark characterizes a two-dimensional reactor core as shown in Fig. 4.5. The reactor core consists of two different fuel regions; the blanket zone, and the seed. The seed is located in two regions (i.e, seed 1 and seed 2). The two-group cross sections are given in Table 4.4, while there is only one precursor group with $\lambda = 0.914193$ s^{-1}.

Table 4.4: Two group constants of the TWIGL Benchmark.

Region	Group	D (cm)	Σ_a (cm^{-1})	$\Sigma_{sg\leftarrow g'}$ (cm^{-1})	$\nu\Sigma_f$ (cm^{-1})
seed 1 & 2	1	1.4	0.01	0.01	0.007
	2	0.4	0.015	0.0	0.2
blanket	1	1.3	0.008	0.01	0.003
	2	0.5	0.05	0.0	0.06

The transient is induced by positive reactivity insertion in the seed 2 region, which is done by decreasing the absorption cross section. Two different transients are studied, the first is a ramp in which the absorption cross section Σ_a changes in time according to

$$\Sigma_a(t) = \begin{cases} \Sigma_a(0)[1 - 0.11667 \times (t - 0.1)] & \text{for } 0.1 < t < 0.3 \text{ s.} \\ \Sigma_a(0) \times 0.97666, & \text{for } t > 0.3 \text{ s.} \end{cases} \tag{4.14}$$

In addition, a step reactivity insertion is produced by reducing the absorption cross section as follows

$$\Sigma_a(t) = \Sigma_a(0) \times 0.97666 \text{ for } t > 0.1 \text{ s.} \tag{4.15}$$

For more details about the benchmark, the reader may consult Ref.[52]. As in the previous

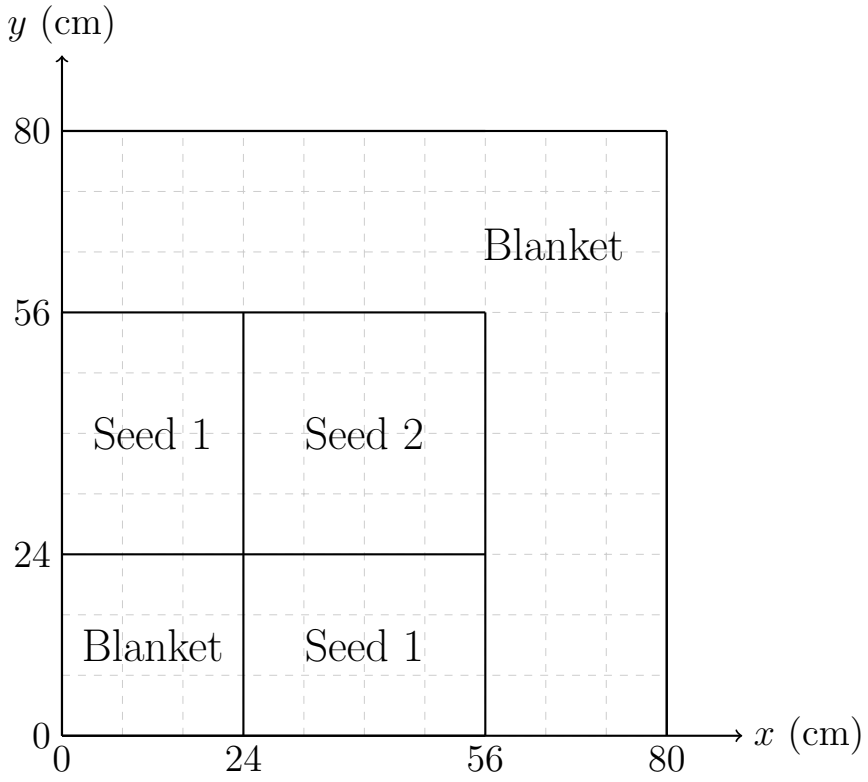


Figure 4.5: Schematic of the TWIGL benchmark.

example, Backward-Euler was used to solve Eq. 4.5 with a time step of 0.001 second. The FOM core power computed of both perturbations is shown in Fig. 4.6.

ROM Results

In order to generate the POD basis, Detran was used to obtain snapshots of the two-group fluxes and the precursor concentrations. With these snapshots, the POD procedure, discussed in section 2.1, was applied to each snapshots set to generate a separate basis for each group flux and one basis for the precursor group. Shown in Fig. 4.7 are the first four POD modes along with the associated temporal coefficients of the fast and thermal flux, where σ_i denotes the singular value associated with the ith POD mode. The first mode is very similar to the flux shape of both groups, with its associated temporal coefficient of the highest magnitude. Different spatial resolutions were used to see how the ROM will behave with a growing

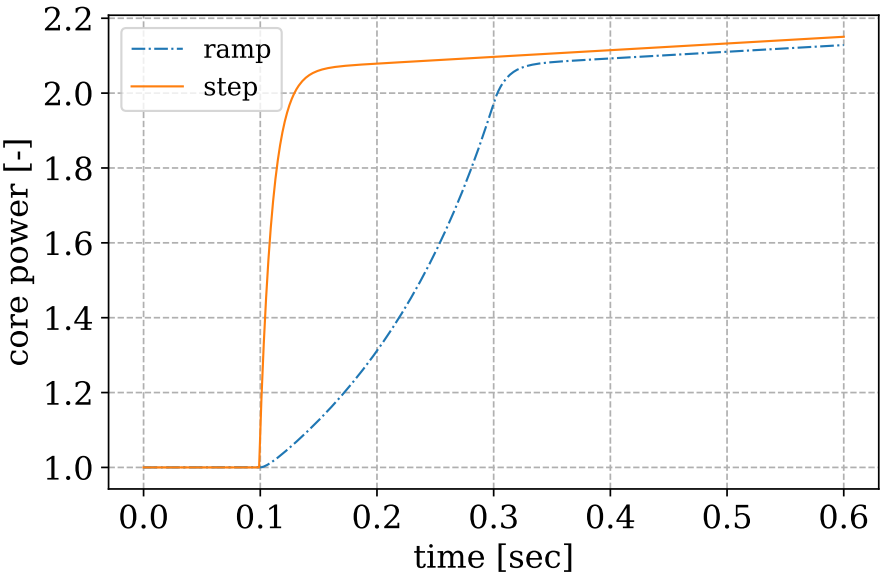
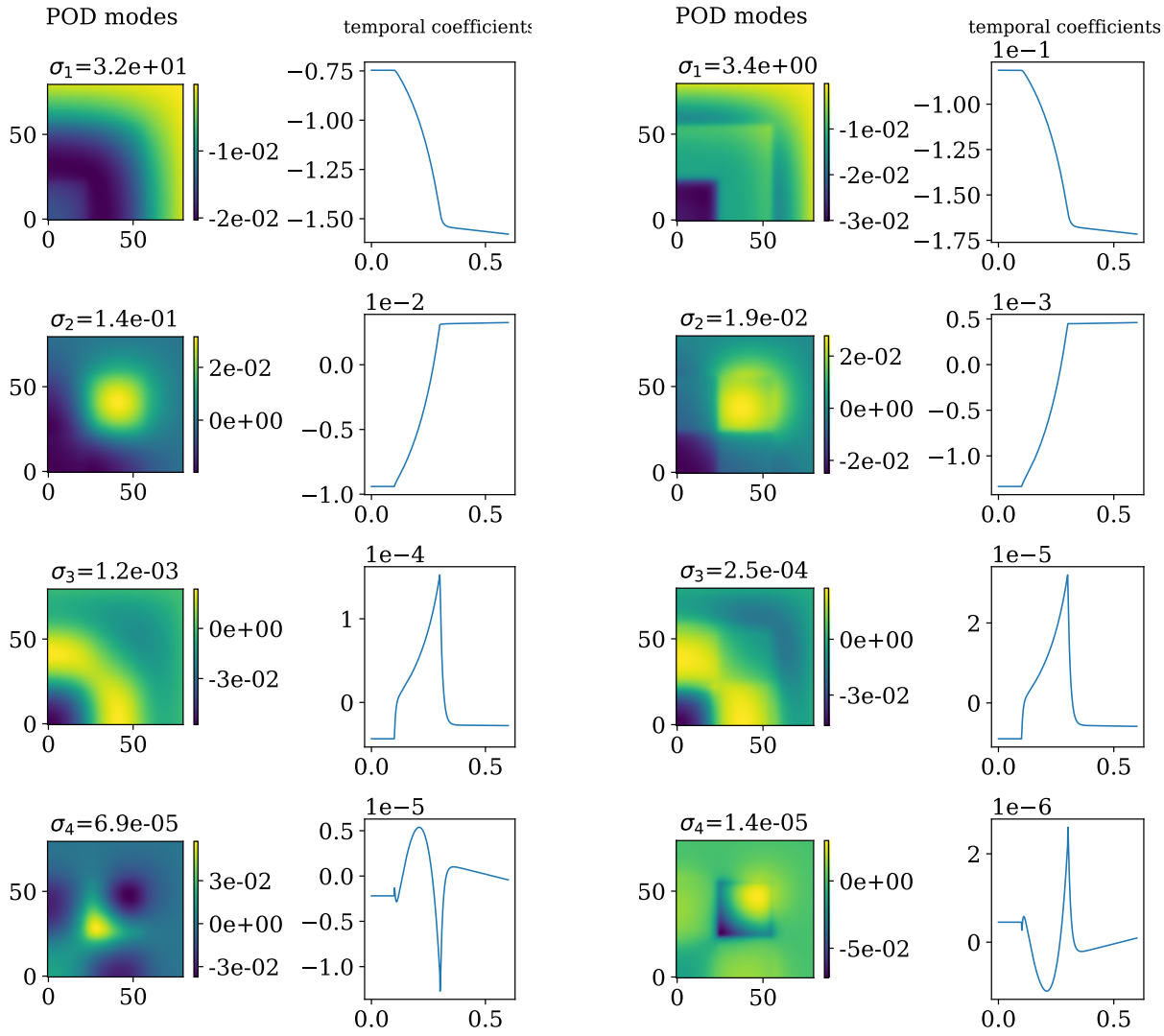


Figure 4.6: Full-order model Core Power.

number of dimensions. Several POD bases of different ranks were generated and used to build a ROM by varying the problem size, i.e, using different spatial discretizations. Fig. 4.8 shows how the mean of the predicted flux relative error decreases as more basis vectors are included in the subspace; the legend in the figure refers to the spatial grid dimension. The accuracy of these ROMs for a given POD rank is nearly identical. In other words, the ROMs (and underlying POD bases) capture the intrinsic dimensions (or features) of the problem, and these intrinsic dimensions depend weakly on the discretization used for that problem.

Also, it can be seen in Fig. 4.8 that the step perturbation error decays faster than that of the ramp, which indicates a faster decay of the singular values. Recall that the step experiment includes a sudden change in the reactor from one state to another through an instantaneous change in the absorption cross section. On the other hand, the ramp is driven by a continuous change in this cross section, and accordingly, the core state undergoes a richer transition that is best captured by a larger basis. In order to study the impact of the snapshot temporal resolution on the ROM performance, POD subspaces were generated using time steps of 0.01 and 0.001 s. For brevity, a fixed 80×80 spatial grid and POD basis of rank 10 were used. The spatio-temporal averaged relative errors of the predicted group flux



(a) fast group.

(b) thermal group.

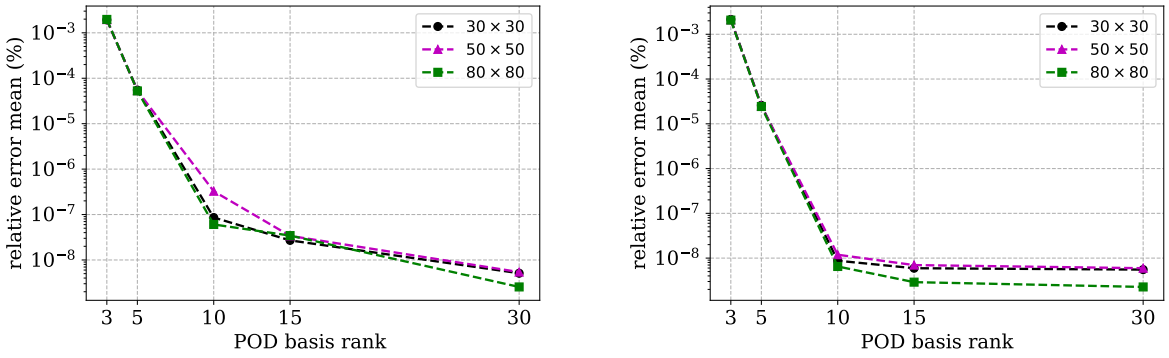
Figure 4.7: POD modes and temporal coefficients of the two-group flux.

and precursors are shown in Table 4.5. It was also interesting to explore the impact of the

Table 4.5: Mean Relative Error (%) between FOM and ROM solutions

Perturbation	snapshot Δt (s.)	Fast flux	Thermal flux	Precursor concentration
Ramp	0.01	2.05×10^{-9}	2.59×10^{-9}	2.54×10^{-9}
	0.001	2.14×10^{-10}	2.16×10^{-10}	2.53×10^{-9}
Step	0.01	2.25×10^{-9}	2.32×10^{-9}	2.49×10^{-9}
	0.001	3.65×10^{-10}	3.65×10^{-10}	2.49×10^{-9}

snapshots resolution on the ROM prediction accuracy. Hence, snapshots were generated with



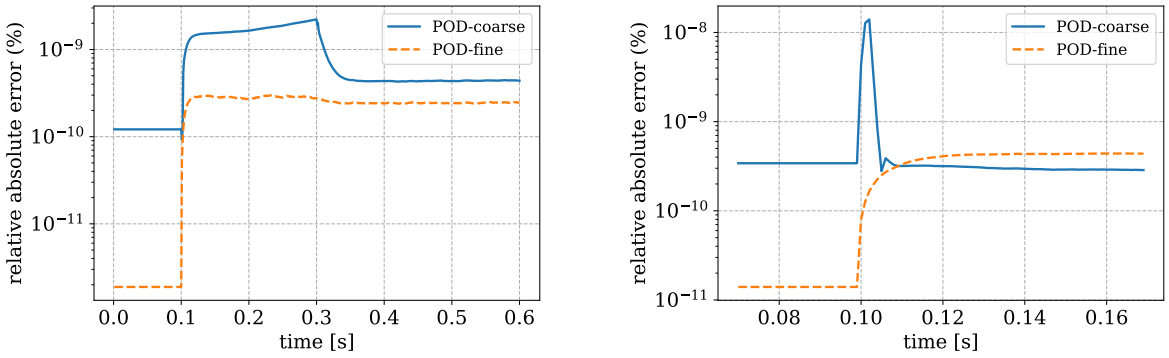
(a) ramp perturbation

(b) step perturbation

Figure 4.8: Spatio-temporal averaged relative error between FOM and ROM fluxes.

two different time mesh sizes (i.e; $\Delta t = 0.001$ s and $\Delta t = 0.01$ s) and two approaches were adopted in constructing the ROM. The first is to generate the basis using the coarse mesh snapshots and perform the testing on a fine mesh. The second is to compute the POD basis and perform the testing using the same fine mesh size. The former approach is favorable from a computational point of view, since using a coarser time mesh is cheaper. Moreover, since the snapshots are expected to exhibit high linear dependence, using a coarse mesh should not impact the quality of the POD modes. Once the reduced space is produced, the ROM can be used to obtain the solution with any desired time resolution. The predicted group fluxes were used to compute the integral core power and the prediction error is shown in Fig.(4.9). The POD-coarse and POD-fine refer to the POD basis generated using 0.01 and 0.001 s spaced snapshots respectively. Note that all the predictions are made on the fine time grid. In the case of the POD-coarse this introduces an error in the initial condition of the temporal coefficients. Recall that the initial condition is computed using $\mathbf{a}3(0) = \mathbf{U}^T\boldsymbol{\phi}(0)$, where T denotes the transpose. However, as can be concluded from the results, the ROM is still able to reproduce the solution with sufficient accuracy. For the step perturbation, the error shown is limited to the time interval 0.05–0.015 s, outside of which the error is nearly constant. The sudden increase in the error corresponds to the sudden insertion of the reactivity, which is represented mathematically by a change in the operator $\mathbf{A}$.

To further assess the accuracy of the ROM, the resulting error was compared to the FOM error of a coarser time mesh. FOMs with coarse discretizations, and lower fidelity



(a) ramp perturbation

(b) step perturbation

Figure 4.9: Relative absolute error between the FOM and the ROM core powers

.

models in general, represents a class of physics-based surrogate models that rely on applying simplification of the original model. Hence, the FOM power was computed twice, once by integrating the system using a fine mesh, i.e, 0.001 s and the other by using coarse time mesh, i.e, 0.01 s. The error between both powers was computed and is shown in Fig. 4.10. As can be seen, the error is significantly higher than that of the POD-coarse Galerkin model shown in Fig. 4.9. Hence, the developed ROM is more reliable and at the same time it preserves the original model fidelity.

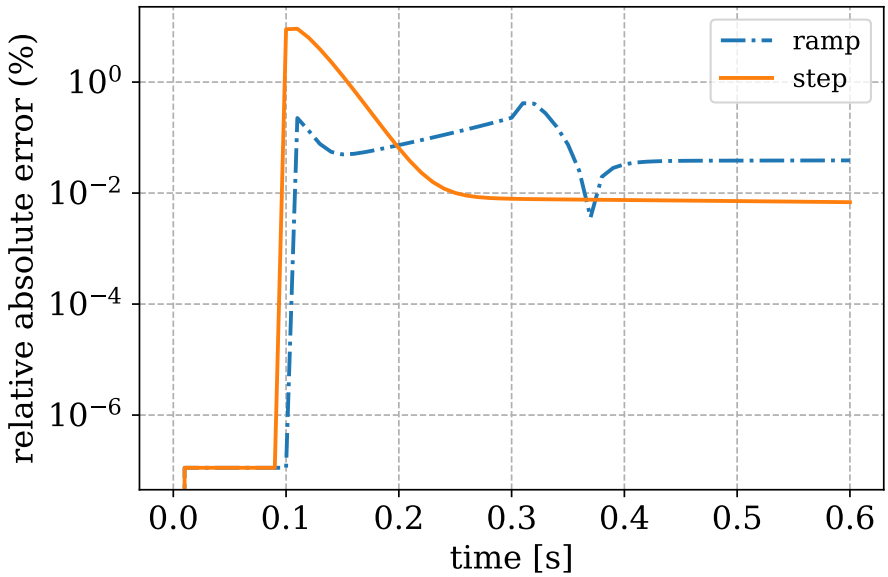


Figure 4.10: Relative absolute error of the FOM core power using coarse and fine time meshes.

To understand the computational cost of the the ROM, recall that the use of the ROM includes two steps, the projection of the full-order operator and the solution of the reduced system. For the problems considered, the cost of the latter step is trivial, with execution times on the order of tens of ms. On the other hand, the projection step is more expensive. Shown in Table 4.6 the core CPU time of the different cases. Although the implementations were not subject to substantial performance tuning, the numbers should represent a realistic and self-contained example for comparing the FOM and ROM performance.

Table 4.6: Computational time of the FOM and the ROM for the TWIGL benchmark.

	Grid	FOM time (s)	ROM time (s)
ramp	30×30	6.3	1.01
	50×50	20.1	2.72
	80×80	61.65	7
step	30×30	6.3	0.6
	50×50	20.5	1.63
	80×80	61.65	4.45

Parametric ROM and POD-greedy results

The greedy sampling algorithm given in Sec. 2.3 was used to parameterize the POD-Galerkin ROM and an uncertainty quantification was carried out to illustrate the method. However, it is important to clarify that the main goal is not to quantify the real problem uncertainties; rather we aim to show the potential of the method to be used for general parameterized problems.

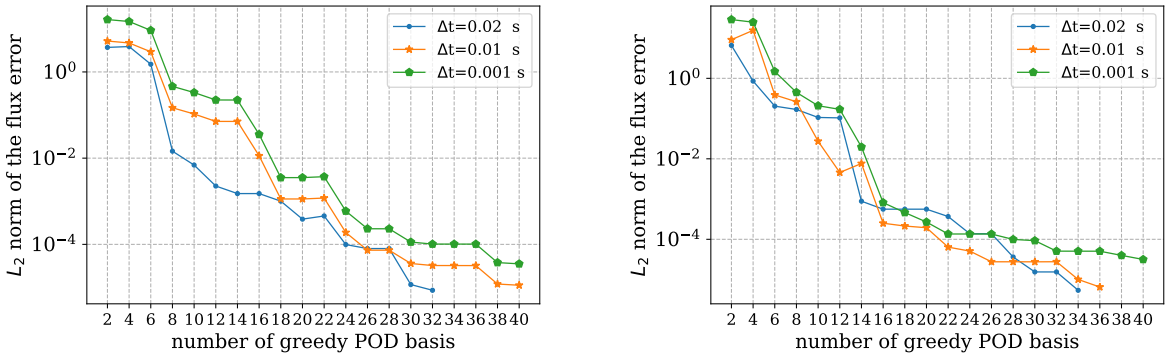
In this analysis, the 30×30 spatial grid was used. The cross sections were assumed to be uniformly distributed within 3% about the nominal values given in Table 4.4. To construct the ROM, 50 independent samples were generated (i.e., training data), while an additional 100 samples were generated for testing (i.e., testing data). Again, different time steps (0.02, 0.01, and 0.001 s) were used to generate the greedy POD subspace, but the ROM was integrated using only a 0.001 s step. Fig. 4.11 shows the error convergence of the greedy algorithm as the subspace is enriched by adding basis vectors derived from different points in the parameter domain. The greedy selection was terminated when a subspace of rank

40 was reached or the reconstruction error fell below the threshold value of 1×10^{-5}. For each selected sample, the FOM was evaluated, and *two* POD basis vectors were retained and added to the greedy subspace. There are no general rules for these stopping criteria or for the number of POD vectors retained at each step. However, for this particular set of problems, ranks beyond 40 (or errors below the threshold) led to negligible improvement. Moreover, including just a single POD vector appeared to leave out the potentially more valuable information contained in the second vector.

For both the ramp and step perturbations, construction of the basis for $\Delta t = 0.02$ s converged rapidly, leading to early termination and subspaces of less than the maximum rank. Contrarily, for $\Delta t = 0.01$ s and $\Delta t = 0.001$ s, the maximum subspace size was reached before the error threshold was satisfied. Each basis was used to construct a ROM with which the sample means and standard deviations of the observables were predicted using the test data. The same quantities were computed using the FOM to estimate the ROM prediction error. Shown in Tables 4.7 and 4.8 are the relative mean error of the sample mean and standard deviation predictions of the group fluxes and the precursor group, respectively. It can be inferred from the numerical results that the ROM is able to capture the solution variation over the parameter domain of interest. The same conclusion can be made from Fig. 4.12, which shows a histogram of the maximum error in the power prediction of all testing samples.

Moreover, the results suggest that the greedy basis could be generated with snapshots that exhibit a coarse time mesh leading to an improved ROM efficiency without sacrificing the accuracy. The selection of optimal resolutions (e.g., time steps) for training depends in part on the corresponding resolution and other characteristics of the target FOM. Such dependence may suggest that a general selection criterion is unlikely to be constructed; nonetheless, effective, problem-dependent heuristics based on experience may be adequate.

The cost of constructing the greedy-POD space was several minutes, which may seem to be relatively expensive, but it is important to note that these procedures are performed only one time. Once the subspace is constructed, the parametric ROM can be evaluated at an arbitrary number of parameter points. For applications that require running hundreds or thousands of cases, the cost of this step diminishes with respect to the cost of the evaluation



(a) ramp perturbation

(b) step perturbation

Figure 4.11: L_2 norm of flux error between the FOM and the ROM with the greedy basis size

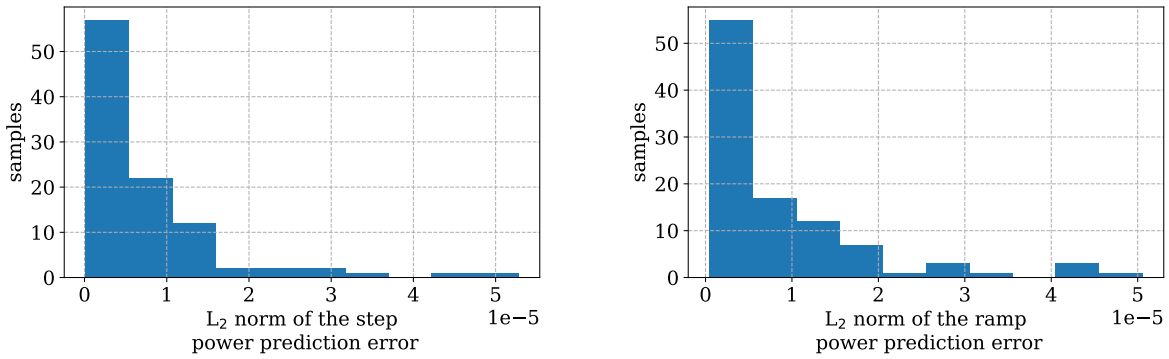
Table 4.7: Mean Relative Error (%) between the FOM and the ROM Sample Means

Perturbation	Δt (s)	Fast flux	Thermal flux	Precursor concentration
Ramp	0.02	1.85×10^{-4}	4.79×10^{-4}	8.13×10^{-4}
	0.01	7.67×10^{-5}	2.75×10^{-4}	3.72×10^{-4}
	0.001	7.66×10^{-5}	2.76×10^{-4}	3.74×10^{-4}
Step	0.02	2.46×10^{-4}	7.52×10^{-4}	3.67×10^{-3}
	0.01	3.52×10^{-5}	1.39×10^{-4}	1.69×10^{-4}
	0.001	5.08×10^{-5}	3.59×10^{-4}	7.43×10^{-4}

Table 4.8: Mean Relative Error (%) between the FOM and the ROM Sample Standard Deviations

Perturbation	Δt (s)	Fast flux	Thermal flux	Precursor concentration
Ramp	0.02	5.93×10^{-3}	5.23×10^{-3}	2.1×10^{-2}
	0.01	2.45×10^{-3}	4.66×10^{-3}	1.6×10^{-2}
	0.001	2.45×10^{-3}	4.66×10^{-3}	1.6×10^{-2}
Step	0.02	4.1×10^{-3}	6.78×10^{-3}	7.12×10^{-2}
	0.01	8.66×10^{-4}	2.5×10^{-3}	2.07×10^{-2}
	0.001	8.62×10^{-4}	4.35×10^{-3}	1.08×10^{-2}

step.



(a) ramp perturbation

(b) step perturbation

Figure 4.12: L_2 norm of the power error between the FOM and the ROM with the greedy basis size

Chapter 5

Nonlinear Diffusion Problems

5.1 Introduction

In the previous chapter, transient problems were treated with single-physics modeling: neutron transport in the diffusion approximation. This treatment is sufficient when the reactor operates at low power where there is no significant feedback from other physics. Realistically, the behavior of nuclear reactors is governed by different physics interactions and requires multi-physics modeling to account for feedback mechanisms. For example, thermal-hydraulics feedback plays an essential role in reactor operation. Changes in the fuel temperature alter the cross sections due to Doppler broadening[13], which in turn impacts the neutron population and hence the fission rate and the rate of heat generation. Also, the coolant density changes with its temperature, impacting the neutron thermalization and, accordingly, the energy spectrum. On the other hand, changes in the flow rate of the coolant cause gradual changes in the reactor temperature, which again impacts the neuron spectrum. Various reactivity coefficients were introduced to quantify these effects on the reactor criticality, such as the Doppler reactivity coefficient, void reactivity coefficient, and moderator temperature coefficient. The magnitude and the direction of these reactivity coefficients are important factors impacting reactor stability and safety. Multi-physics modeling accounts for these different feedbacks mechanisms; however, it poses a challenge in terms of the computational cost, since it entails

solving coupled systems for different fields such as neutron flux, temperature, and flow rate. Moreover, the neutron transport (or diffusion) equation is no longer a linear one due to the indirect dependence of the cross section on the neutron flux.

A simplified, homogeneous multi-physics model that shows the coupling between different physics can be illustrated by the one-group diffusion equation coupled with thermal conduction[53], or

$$\frac{1}{v}\frac{\partial \phi}{\partial t} - \nabla . D\nabla\phi + \Sigma_a(T) = \nu\Sigma_f(T)\phi\,, \tag{5.1}$$

and

$$\rho C_p \frac{\partial T}{\partial t} - \nabla . K(T)\nabla T = w\Sigma_f(T)\phi\,, \tag{5.2}$$

where ρ is the material density, C_p is the heat capacity, K is the thermal conductivity, w is the energy released per fission and T is the material temperature. All the diffusion equation notations are defined in Chapters 3 and 4. Changes in the neutron fission rate result in a temperature change (via Eq.5.2), which impacts the cross section and hence, the solution of Eq. 5.1, i.e., the neutron flux. Note that other systems, e.g., those describing the fluid flow rate or the mechanical response of fuel elements, can be coupled with the above equations. Two approaches are used to solve such systems. The most commonly used one is a serial coupling of two or more separate codes in such a way that the output of one at a given time step, such as the flux distribution, is used as an input to the other, e.g., the temperature (which can be used then to update the cross section and so on). This procedure is commonly referred to as operator splitting or explicit coupling. One drawback of operator splitting is that time integration requires small steps to avoid stability issues, thus leading to high computational time. Alternatively, tight coupling in time between the physics can be maintained by using an implicit scheme that solves a non-linear iteration in each time step, also increasing the computational time. A modern algorithm such as Jacobian free Newton-Krylov (JFNK)[53] can be used to perform this implicit coupling. Another popular approach and the one adopted here is to use fixed-point iteration, with an implicit time scheme to iterate between difference physics within a time step until convergence is satisfied.

In this chapter, a reduced-order model is developed to overcome the high computational expense. The non-linearity was treated using the DEIM algorithm presented in chapter 2, and the LRA benchmark was used as an illustrative example.

5.2 Application to the LRA Benchmark

The LRA benchmark employs two energy groups and two delayed precursor groups and represents a 2-D quarter-core, BWR model subject to a control-rod ejection with adiabatic heating[54]. A schematic layout of the problem is shown in Fig. 5.1, where region numbers indicate different materials and region R is the ejected rod. The transient is initiated by a

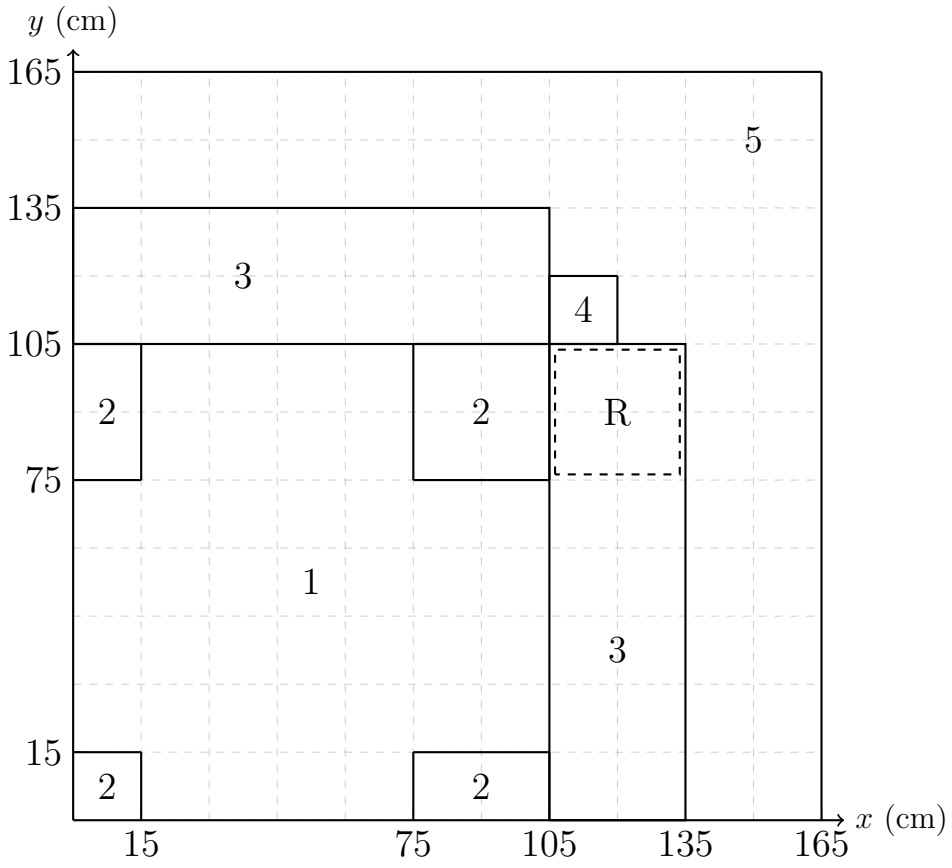


Figure 5.1: Schematic of LRA benchmark.

control rod ejection with constant velocity over a period of 2 seconds leading to a super-critical, prompt transient, in which the power increases by about 10 orders of magnitude in a very

short time. The control rod movement is represented by changing the thermal absorption cross section of the control rod material. The two-group diffusion equation represented in chapter 2, is coupled with thermal feedback through adiabatic fuel heat-up defined by

$$\frac{\partial}{\partial t}T(x,t) = \alpha[\Sigma_{f1}(x,t)\phi_1(x,t) + \Sigma_{f2}(x,t)\phi_2(x,t)]\,, \tag{5.3}$$

and corresponding Doppler feedback defined by

$$\Sigma_{a1}(x,t) = \Sigma_{a1}(x,t=0)[1+\gamma(\sqrt{(T(x,t)} - \sqrt{(T_0)})]\,, \tag{5.4}$$

where Σ_{f_i} is the fission cross section of group i, T is the temperature, Σ_{a1} is the fast absorption cross section and T_0 is the initial temperature which is 300 K, $\alpha = 3.83 \times 10^{-11}$ K cm^3 and $\gamma = 3.034 \times 10^{-3}$ K$^{-0.5}$. The power local is computed as

$$P(x,t) = \kappa[\Sigma_{f1}(x,t)\phi_1(x,t) + \Sigma_{f2}(x,t)\phi_2(x,t)]\,, \tag{5.5}$$

where $\kappa = 3.204 \times 10^{-11}$ W/fission. The model was implemented in the deterministic transport code Detran[10]. A mesh-centered, finite-difference discretization was used in space, while a first-order, backward-difference temporal discretization was used with a fixed 0.001 s step size. The initial condition for the transient was computed by solving the steady-state equation, for which k_{eff} was found to be 0.9975. Note that at each time step the fast absorption cross section is a function of the solution itself. The convergence criterion used to reduce the L_2 norm of the relative flux error between two consecutive iterations to less than 1×10^{-6}, or

$$\|\frac{\mathbf{\Phi}^{i+1} - \mathbf{\Phi}^i}{\mathbf{\Phi}^i}\| \leq 1 \times 10^{-6}\,, \tag{5.6}$$

where the superscript i denotes the iteration number. The transient was calculated for 3.0 seconds. Figure 5.2 shows the reference full-order model power along with the core average temperature, while Figure 5.3 shows the steady-state group flux distribution. This reference solution is not necessarily numerically converged in either space or time but represents the

simulation response that one wishes to approximate.

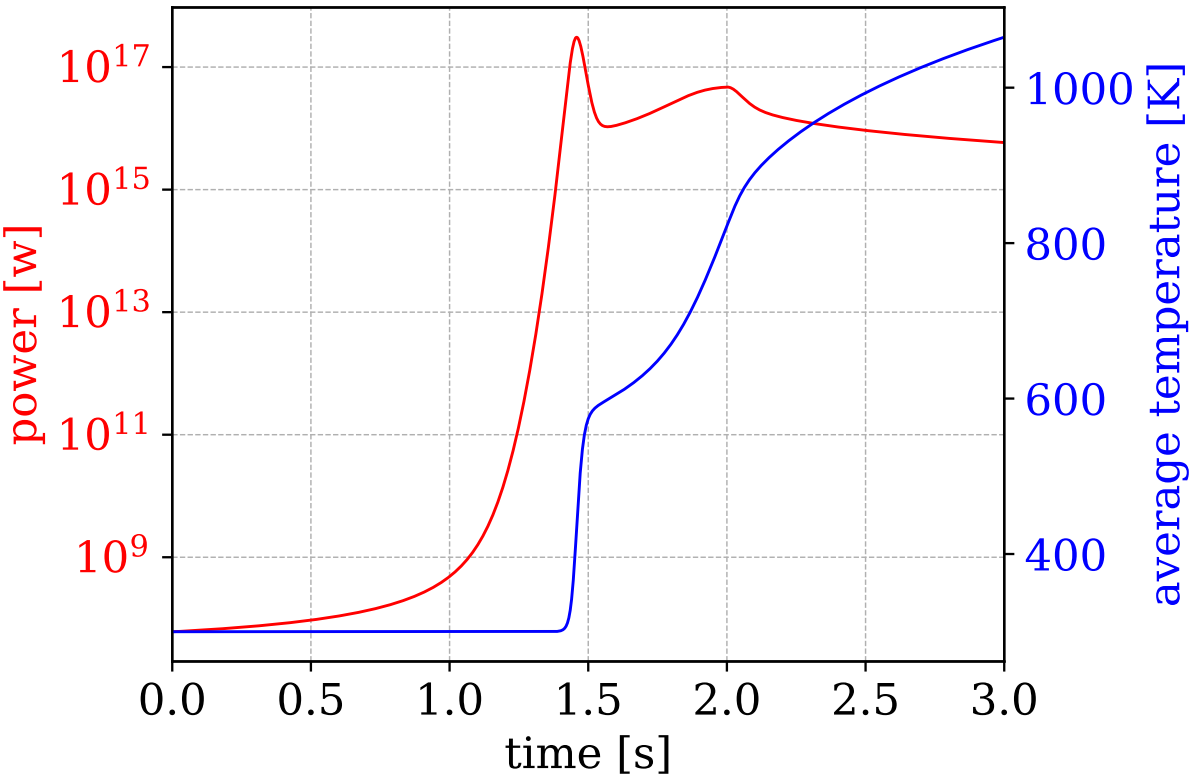
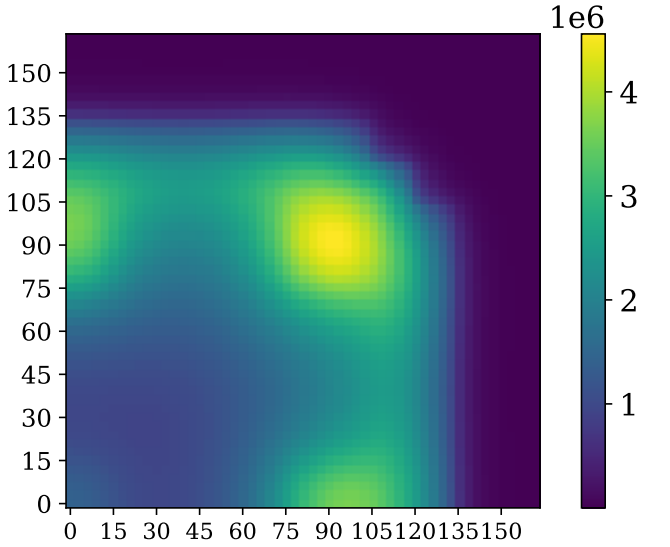
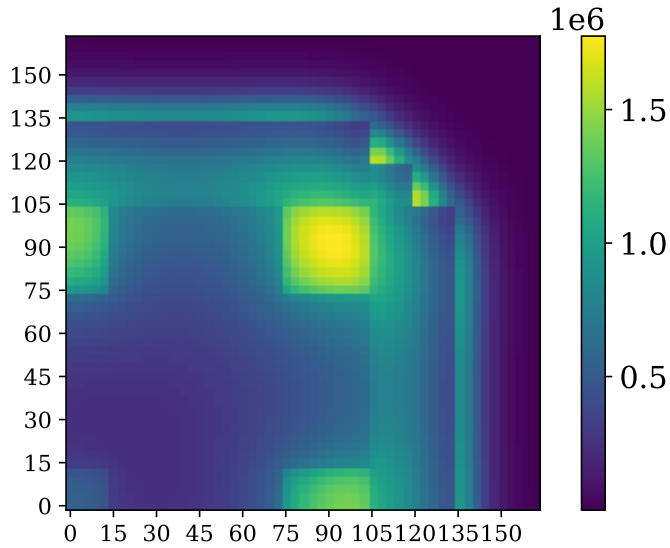


Figure 5.2: LRA core power



(a) Steady state fast flux.



(b) Steady state thermal flux.

Figure 5.3: Steady state group flux (neutrons cm^2/s)

material	group	D (cm)	Σ_a (cm^{-1})	$\Sigma_{s2\leftarrow1}$ (cm^{-1})	$\nu\Sigma_f$ (cm^{-1})
1	1	1.255	0.008252	0.004602	0.02533
	2	0.211	0.1003	0.1091	
2	1	1.268	0.007181	0.004609	0.02767
	2	0.1902	0.07047	0.08675	
3	1	1.259	0.008002	0.004663	0.02617
	2	0.2091	0.08344	0.1021	
4	1	1.259	0.008002	0.004663	0.02617
	2	0.2091	0.073324	0.1021	
5	1	1.257	0.0006034	0.0	0.04754
	2	0.1592	0.01911	0.0	

$v_1 = 3 \times 10^7$ cm/s, $v_2 = \ 3 \times 10^5$cm/s..

Table 5.1: Two group constants of the LRA benchmark.

group	β_i	λ_i (s^{-1})
1	0.0054	0.00654
2	0.001087	1.35

Table 5.2: Delayed neutron data of the LRA benchmark.

5.2.1 ROM Implementation

Using the POD-Galerkin projection, a ROM was constructed to approximate the group flux and the core power of the LRA benchmark. As discussed in the previous chapter, the time-dependent diffusion equation coupled with the precursors group equation can be cast into a compact form as in Eq. 4.5. Similar to the TWIGL benchmark ROM, a POD basis set was generated for each group flux while one basis set was generated for both precursor groups, i.e, $\mathbf{\Phi}_1(t) = \mathbf{U}_1\mathbf{a}_1(t)$, $\mathbf{\Phi}_2(t) = \mathbf{U}_2\mathbf{a}_2(t)$ and $\mathbf{C}(t) = \mathbf{U}_c\mathbf{a}_c(t)$, where the $\mathbf{U}_1$, $\mathbf{U}_2$ and $\mathbf{U}_c$ are the POD bases of the fast group, thermal group and precursor concentration respectively, and $\mathbf{a}_1$,

$\mathbf{a}_2$ and $\mathbf{a}_c$ are their respective temporal coefficients. The first 100 normalized singular values[1] of the group flux are shown in Fig. 5.4. Since the focus of this application is on treating the

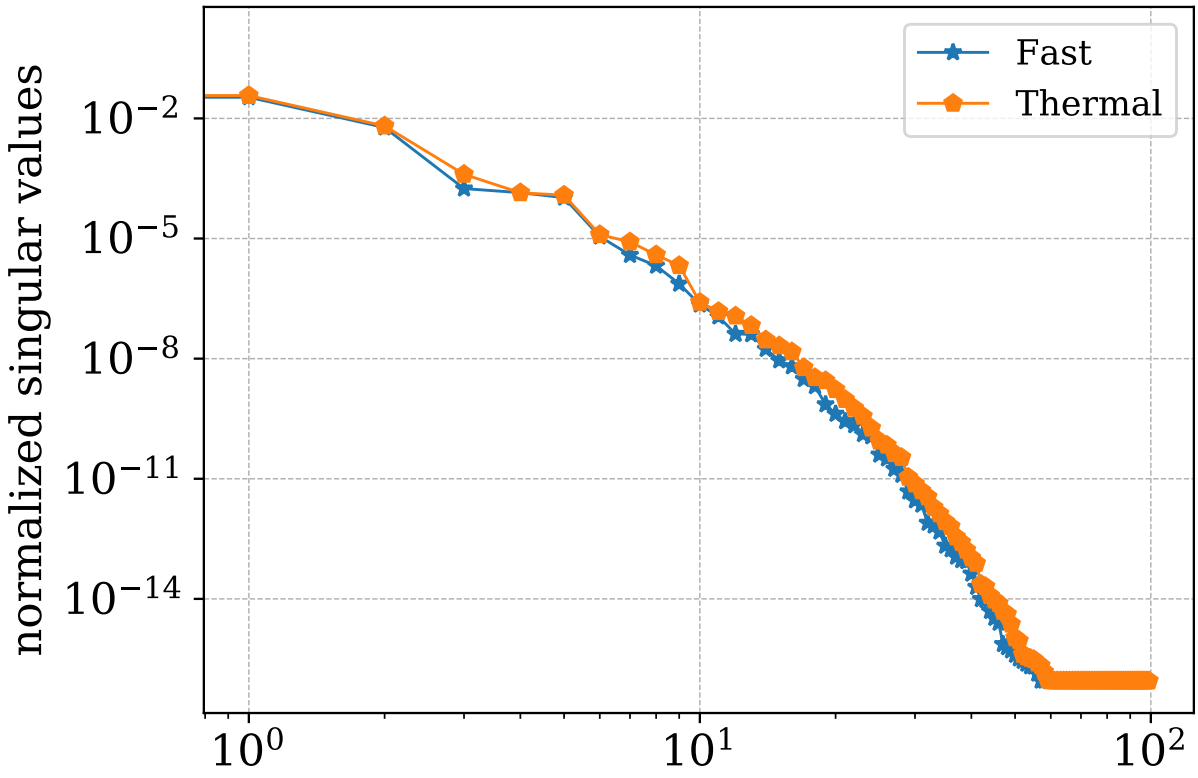


Figure 5.4: Flux normalized singular value.

non-linearity, rather than trying different ranks of the POD basis, a rank of 30 was chosen for each group flux and for the precursor concentration. Also, the temperature field was approximated by projecting it onto a lower dimensional POD space following

$$\mathbf{T}(t) = \mathbf{U}_{\text{temp}}\mathbf{a}_T(t)\,, \tag{5.7}$$

where $\mathbf{U}_{\text{temp}} \in \mathbb{R}^{n\times r_T}$ is the POD subspace of rank r_T and $\mathbf{a}_T \in \mathbb{R}^{r_T}$ is a vector of the corresponding temporal coefficients. To obtain the subspace, snapshots of the temperature were collected at different times for which the SVD was computed. Shown in Fig. 5.5 are the first 100 normalized singular values, based on which a POD of rank 10 was selected for

[1]The normalization is done by dividing each singular value by the sum of all singular values.

approximating the temperature.

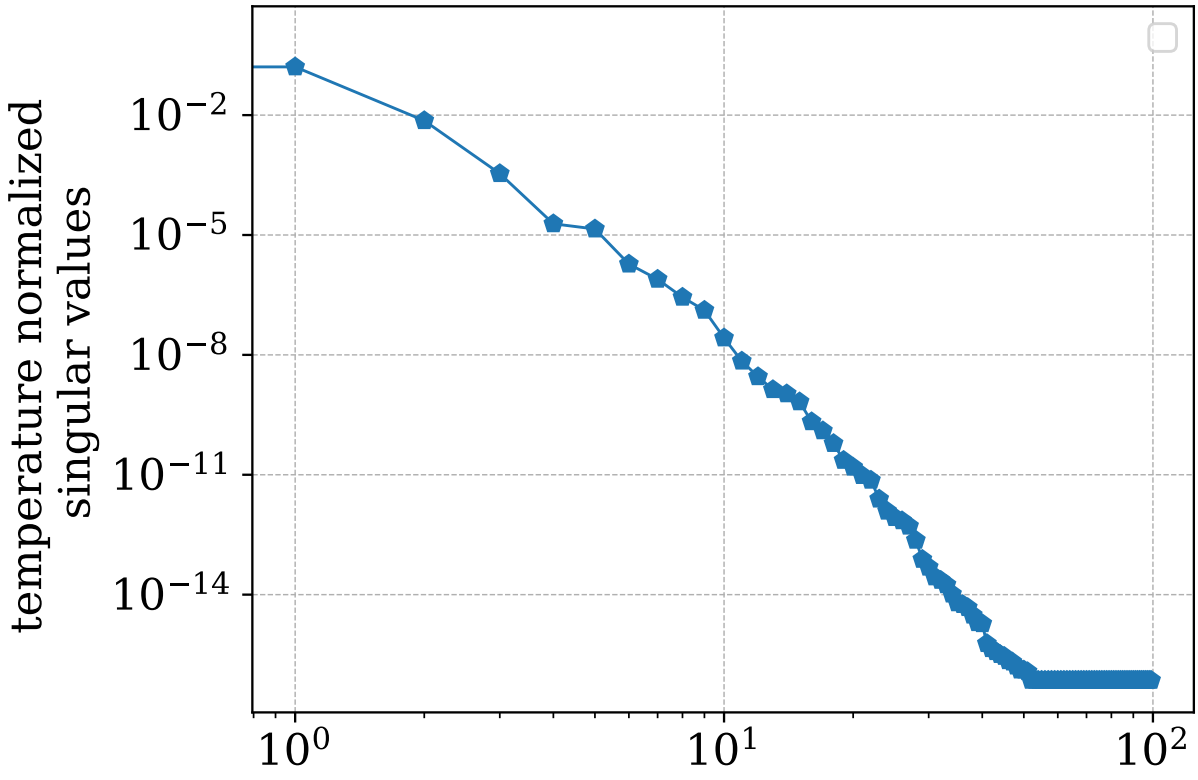


Figure 5.5: Temperature normalized singular value.

Inserting Eq. 5.7 into Eq. 5.3 and multiplying on the left by $\mathbf{U}_T^T$ [2] yields

$$\frac{d\mathbf{a}_T(t)}{dt} = \alpha[\underbrace{\mathbf{U}_{\text{temp}}^T\boldsymbol{\Sigma}_{f1}\mathbf{U}_1}_{\mathbf{U}_{T1}}\mathbf{a_1}(t) + \underbrace{\mathbf{U}_{\text{temp}}^T\boldsymbol{\Sigma}_{f2}\mathbf{U}_2}_{\mathbf{U}_{T2}}\mathbf{a_2}(t)]\,. \tag{5.8}$$

Note that $\mathbf{U}_{T1}$ and $\mathbf{U}_{T2}$ are time-independent, so they can be precomputed once and stored. Thus, at each time step the flux temporal coefficients (i.e, $\mathbf{a_1}(t)$ and $\mathbf{a_2}(t)$) are computed by solving the reduced system of the diffusion equation and then used to compute the temperature coefficients., i.e, $\mathbf{a}_T$, by employing a first-order, backward-difference scheme to Eq. 5.8. Similar to the full-order model, a fixed point iteration is used to treat the flux non-linearity. However, it was observed that with using the same convergence criteria as in Eq. 5.6, the reduced model needed more iterations to converge which could be due

[2]The superscript T denotes the matrix transpose.

to the relatively small value of some of the flux coefficients; hence, a convergence criteria was employed based on the relative L_2 norm of the error of flux coefficients between two consecutive iterations

$$\frac{\|\mathbf{a}^{i-1} - \mathbf{a}^{i}\|}{\|\mathbf{a}_i\|} \leq 1 \times 10^{-6}\,. \tag{5.9}$$

Note that the fast absorption cross section is a non-linear function of the temperature, so updating it at each time step using Eq. 5.4 requires reconstructing the full-order temperature distribution. Since the cost of this reconstruction is proportional to the full-order dimension, i.e., n, the DEIM was used to approximate the absorption cross section and, hence, to avoid reconstructing the temperature at all the spatial cells in the core. For this purpose, snapshots of the fast absorption cross section were collected in order to generate a POD subspace that is used to approximate it

$$\mathbf{\Sigma}_{a1}(t) = \mathbf{U}_{\sigma}\mathbf{a}_{\sigma}(t)\,, \tag{5.10}$$

where $\mathbf{U}_{\sigma}$ is the generated POD space and $\mathbf{a}_{\sigma}$ is a vector of the associated temporal coefficients. A set of interpolation locations were selected following Algorithm 3 in which an order of 10 was used. Thus, at each time step the temporal coefficients were computed by solving

$$\mathbf{a}_{\sigma} = (\mathbf{P}^T\mathbf{U}_{\sigma})^{-1}\mathbf{P}^T(\mathbf{\Sigma}_{\mathbf{a1}(t=0)}[1 + \gamma(\sqrt{\mathbf{U}_t^T\mathbf{a_t}} - \sqrt{\mathbf{T}_0})])\,, \tag{5.11}$$

where $\mathbf{P}^T$ is the interpolation matrix and $\mathbf{\Sigma}_{\mathbf{a1}(t=0)} \in \mathbf{R}^{n\times n}$ is a diagonal matrix with elements equal to initial fast absorption cross section. As explained in chapter 2, this interpolation matrix extracts elements (rows) from the vector (matrix) it operates on. Hence, the cross section was computed at each time step using temperature reconstructed only at the interpolation locations. Moreover, since the problem operator $\mathbf{A}$ in Eq. 4.5 exhibits a time dependence, the MDEIM was used to decompose the operator and hence avoid performing the expensive projection (i.e., $\mathbf{U}^T\mathbf{A}\mathbf{U}$) at each time step. Note that the $\mathbf{A}$ is composed of four main submatrices of which only the loss matrix, i.e., $\mathbf{L}$, changes with time. Accordingly, the MDEIM was used only to approximate this submatrix of the operator $\mathbf{A}$, while the other three submatrices were projected once at $t = 0$ and stored for use in the subsequent time steps.

Shown in Fig. 5.6 are the singular values resulting from the SVD of the serialized submatrix **L** snapshots. A DEIM of order 15 was employed, meaning that the it was decomposed into 15 time-independent matrices that were computed and stored in an offline stage.

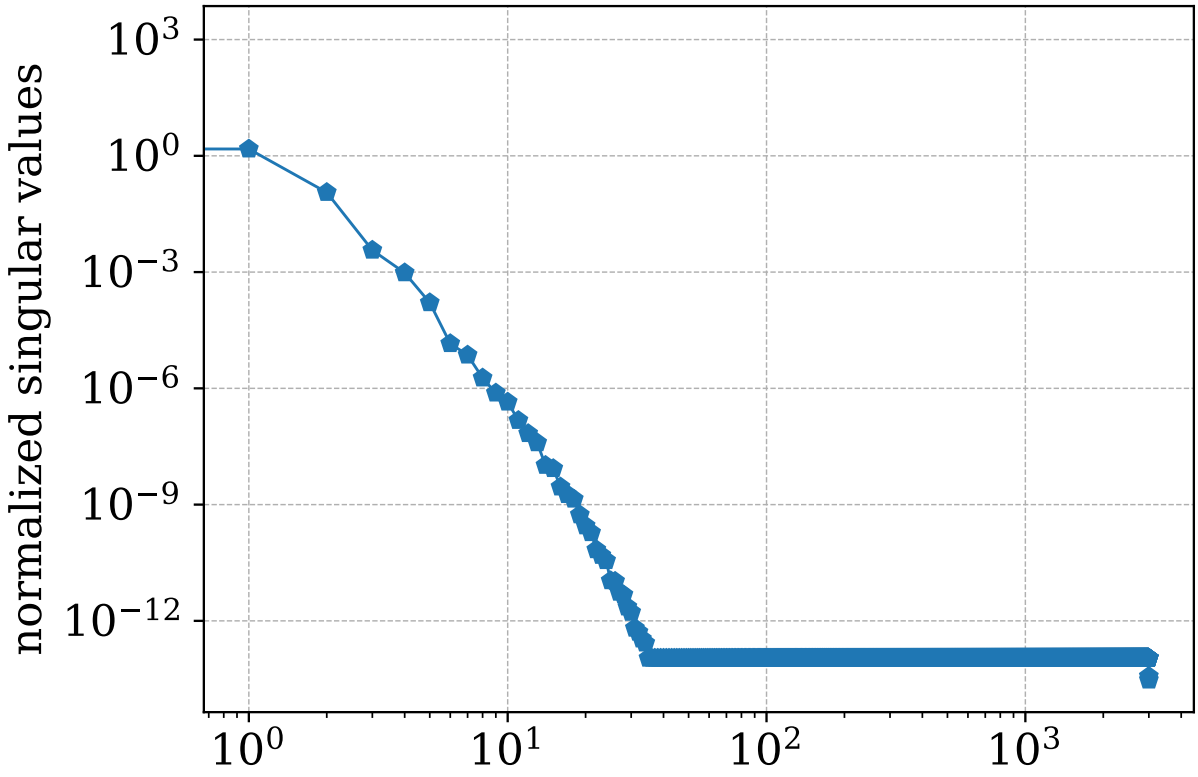
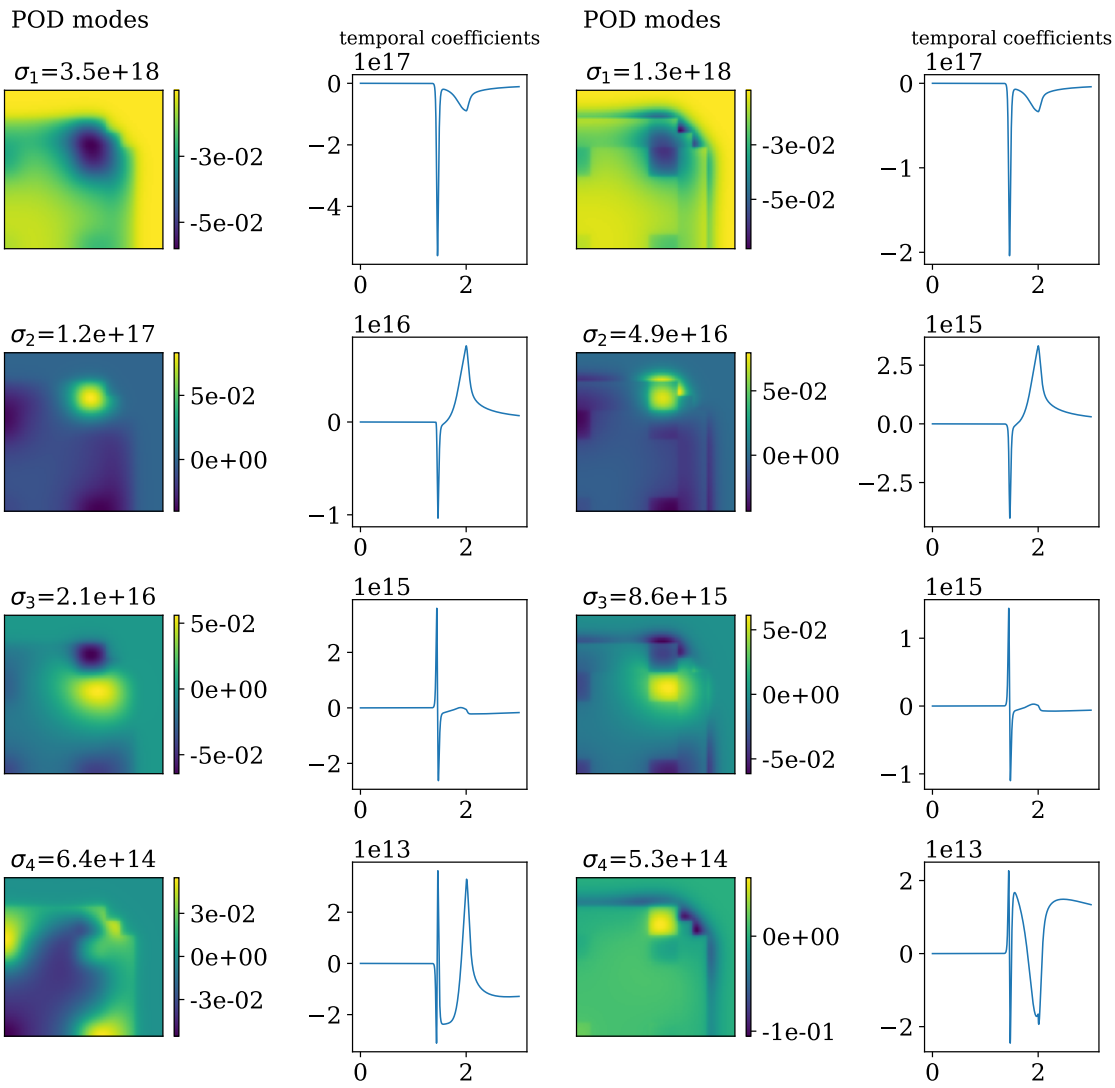


Figure 5.6: Operator normalized singular value.

5.2.2 Results

ROM results

ROMs were constructed for full-order models with increasing the spatial fidelity, i.e, spatial grids of 22×22, 44×44 and 55×55 were used. For brevity, we show the first four POD modes with their temporal coefficients of only the 55×55 grid in Fig. 5.7.



(a) Fast flux modes.

(b) Thermal flux modes.

Figure 5.7: Flux POD modes and their temporal coefficient.

Using a ROM for each spatial grid with the specifications given in the previous section, the absolute relative error in the predicted core powers are computed as shown in Fig. 5.8.

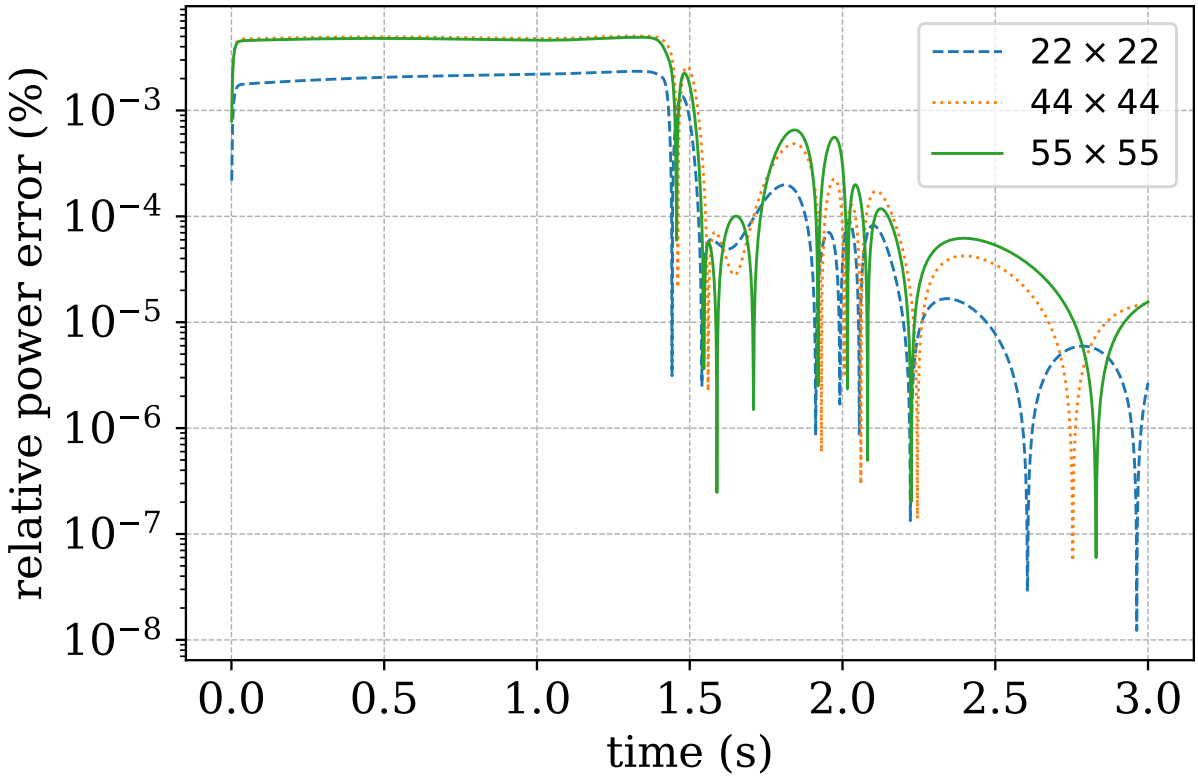


Figure 5.8: Relative error of the power predicted by MDEIM ROM

To understand the impact of of MDEIM on the error;the results are compared this of a ROM without using the MDEIM, i.e., with performing direct operator projection at each time step. The relative difference of power between both ROMs is shown in Fig. 5.9. The shown errors implies that MDEIM accurately approximates the problem operator.

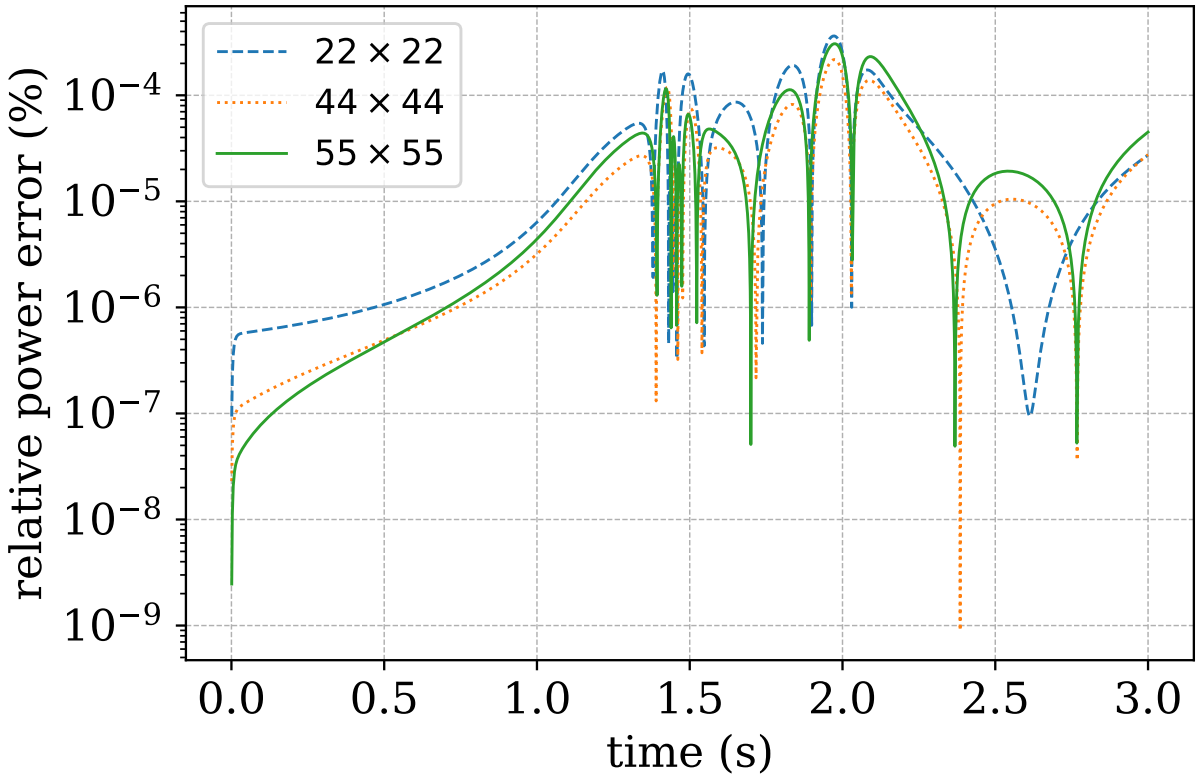


Figure 5.9: Relative error of the power predicted by ROM.

Also, the spatial-averaged relative error in the approximated temperature was computed and is shown in Fig. 5.10.

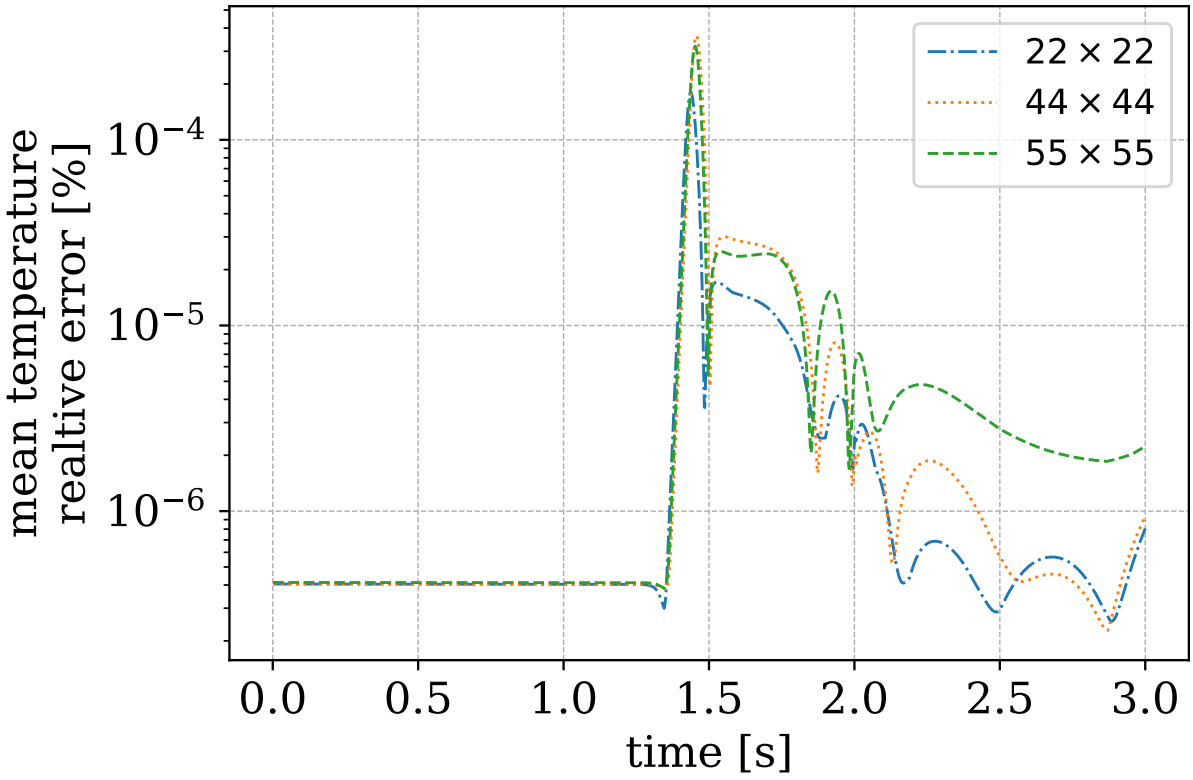


Figure 5.10: Relative error of the temperature predicted by ROM.

As mentioned earlier, the non-linearity of the fast absorption cross section was approximated using DEIM of rank 10. Shown in Fig. 5.11 are the ten selected interpolation points onto the absorption cross section map at the end of transient, i.e., at $t = 3$ s. The temperature was constructed only at these locations from which the cross section is evaluated using Eq. 5.4.

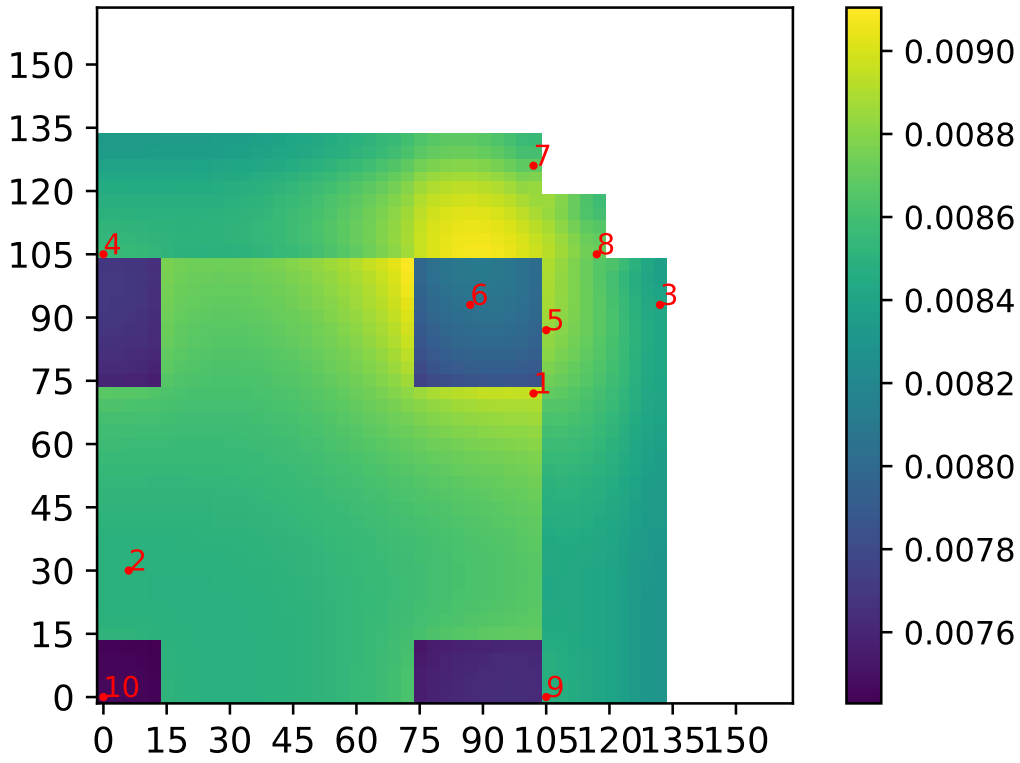


Figure 5.11: The fast cross section at t=3 s with the selected interpolation indices in red color.

The relative error in the assembly power at the peak is computed and is shown in Fig. 5.12 for the 55×55 grid from which it can be seen that the maximum error is in the order of $1 \times 10^{-5}\%$.

Figure 5.12: Assembly-averaged power relative error for the 55×55 grid

Moreover, the computational times of the FOM and the ROMs are given in Table 5.3. The important point here is that the MDEIM-ROM cost is slightly dependent on the problem

Table 5.3: Computational time of the FOM and the ROMs for the LRA benchmark.

Grid	FOM (s)	ROM (s)	MDEIM-ROM (s)
22×22	243	76	38
44×44	1269	226	46
55×55	2195	341	50

dimension. On the other hand, the ROM time increases with the grid size because the operator projection is performed at each time step. The shown MEDIM-ROM time includes the offline cost which involves the operator $\mathbf{L}$ decomposition and the projection of the decomposed matrices. Also, both the ROM and MDEIM-ROM times include the time to reconstruct of the full-order solution.

Parametric ROM Results

In the previous section, the developed ROM was used to reconstruct the solution using a POD subspace obtained from snapshots generated using the same parameter point at which the solution is sought. Here, our purpose is to extend the capability of the developed ROM to perform parametric studies. However, it is important to clarify that it is not our goal in this study to quantify output uncertainties or perform statistical analysis. Instead, we merely aim to measure the prediction error of the ROM at new parameter points using a POD subspace sampled from a parameter domain of interest. A case study was performed in which the parameters of interest are assumed to be the kinetic data, i.e., the delayed neutron fractions and the decay constants. A POD greedy sampling was used to construct a global POD subspace. Although the original benchmark employs two precursor groups, it was more convenient in this study to use six groups to enlarge the parameter space. Shown in Table 5.4 are the parameters normal distributions data adopted from Refs.[55;56]. A POD greedy sampling is used for which a set of 50 parameters points were generated to obtain a global POD space, where at each iteration, three POD basis vectors are added to the space. A grid of 22×22 is used throughout this study. A greedy algorithm stopping criteria was used such that the iteration terminates if the relative norm of the flux error is 1×10^{-4} or the POD space has a size of 6 (i.e., 20 parameters points are selected). As noted earlier, there are no general prior criteria for selecting these parameters. The MDEIM-ROM is sought

group i	$\beta_i \pm \sigma_{\beta_i}$	$\lambda_i \pm \sigma_{\lambda_i}(s)^{-1})$
1	0.038 ± 0.004	0.0127 ± 0.0003
2	0.213 ± 0.007	0.0317 ± 0.0012
3	0.188 ± 0.024	0.1150 ± 0.0040
4	0.407 ± 0.010	0.3110 ± 0.0120
5	0.128 ± 0.012	1.4000 ± 0.1200
6	0.026 ± 0.004	3.8700 ± 0.5500

Table 5.4: 6 groups delayed neutron precursor data.

to predict the power at every 0.001 s during the transient, however, a time step of 0.005 s was used in the greedy procedures. Shown in Fig. 5.13 is the flux error convergence with the greedy iterations. As shown, the maximum size of the POD space, i.e., 60, is reached

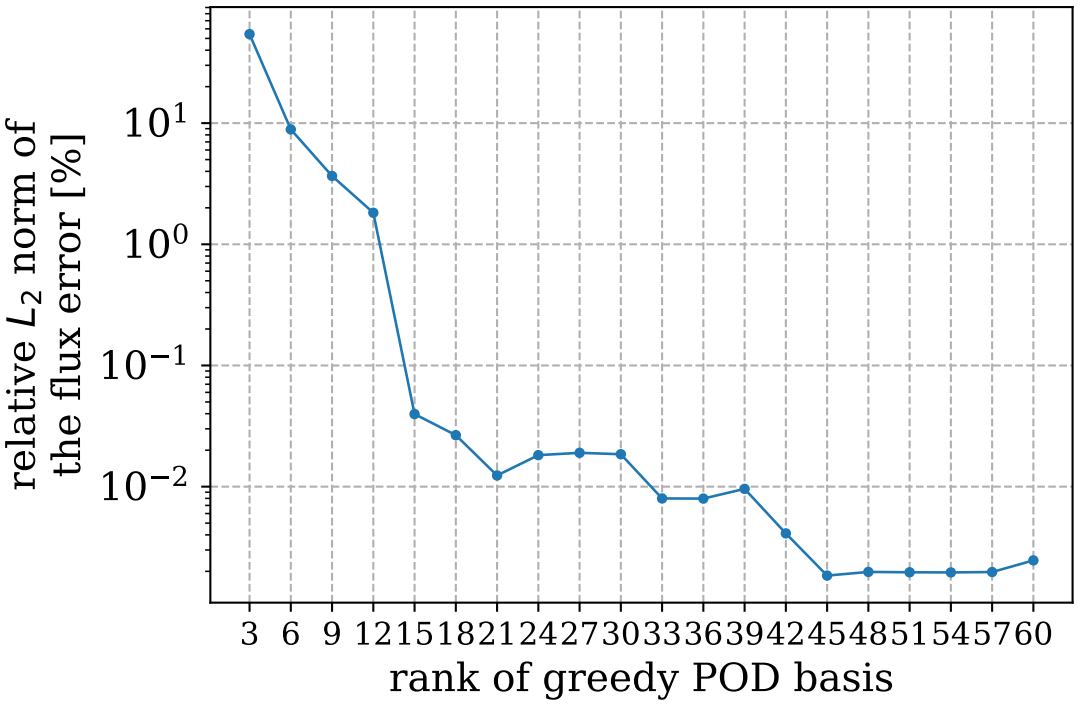


Figure 5.13: Greedy error convergance

before the error approaches the specified tolerance. The resulting POD basis was stored and used for testing the ROM. Thus, 50 new parameter samples were generated, and the ROM was executed with the resultant greedy basis. Moreover, the FOM was run at these testing parameter points, and the error between both models was computed. The quantities of interest were the assembly power at the peak and the maximum core temperature, i.e., the end-of-transient temperature. Note that the time to peak is different from sample to sample. Shown in Fig. 5.14, the sample mean of the assembly power error, at the peak, where the mean core maximum temperature error is shown in Fig. 5.15.

Figure 5.14: Sample mean of the core peak power

Figure 5.15: Sample mean of the core maximum temperature

Among all samples, the maximum assembly power peak was 0.00111%, and the maximum

temperature error was $2.9 \times 10^{-5}\%$. The resulting errors imply that the greedy-sampled POD subspace is rich with information from the whole parameters domain, and accordingly, resulted in a good approximation even for parameters not among the training sample. However, methods to improve efficiency and, possibly, prediction accuracy will be presented in chapter 7 as suggestions for future work..

Chapter 6

ROM Application to the Neutron Transport Equation

6.1 Introduction

In this chapter, we study the potential of the Boltzmann neutron transport equation for reduction via the POD-Galerkin projection. As mentioned in chapter 3, the transport equation gives an exact treatment of the neutron flux in a specified system by explicitly considering the angular variable. Because of the inclusion of the direction-of-flight as a dependent variable, solving the equation numerically requires a discretization in angle in addition to time, space, and energy discretization, which makes solving it an expensive task. For this reason, it is considered an attractive application for model order reduction. Here, we present a preliminary study of the POD-Galerkin projection application to the transport equation, where we focus on the actual implementation and the reliability of the ROM. Methods to improve its efficiency are left to future work. In the following section, a formulation of the transport equation is presented and the system is compactly described in a convenient operator notation. The corresponding ROM formulation is presented in Section 6.3, and applications with results are given in Section 6.4

6.2 Neutron Transport Equation

The multi-group time-dependent transport equation is given by

$$\begin{aligned}\frac{1}{v_g}\frac{\partial \psi_g}{\partial t} &= -(\hat{\Omega}\cdot\nabla\psi_g(\boldsymbol{r},\Omega,t)+\Sigma_{tg}(\boldsymbol{r},t)\psi_g(\boldsymbol{r},\Omega,t))\\ &+\frac{1}{4\pi}\int_{4\pi}d\Omega'\sum_{g'=1}^{G}\Sigma_{sg\leftarrow g'}(\boldsymbol{r},\Omega'\cdot\Omega,t)\psi_{g'}(\boldsymbol{r},\Omega',t)\\ &+\frac{1}{4\pi k_{\text{keff}}}\int_{4\pi}d\Omega'\left(\chi_{pg}(\boldsymbol{r})(1-\beta(\boldsymbol{r}))\sum_{g'=1}^{G}\nu\Sigma_{fg'}(\boldsymbol{r},t)\psi_{g'}(\boldsymbol{r},\Omega',t)\right)\\ &+\sum_i \lambda_i C_i(\boldsymbol{r},t)\chi_{ig}(\boldsymbol{r})\\ &+q_g(\boldsymbol{r},\hat{\Omega},t)\,,\quad g=1\ldots G\end{aligned} \tag{6.1}$$

where ψ is the angular flux, Ω is the neutron direction-of-flight and q represents the source term, which could be external or/and the fission source. All other notation has been defined in the previous chapters. The corresponding set of precursor equation is,

$$\frac{\partial C_i}{\partial t}=\frac{\beta_i(\boldsymbol{r})}{k_{\text{eff}}}\int_{4\pi}d\Omega'\sum_{g'=1}^{G}\nu\Sigma_{fg'}(\boldsymbol{r},t)\psi_{g'}(\boldsymbol{r},\Omega',t)-\lambda_i C_i(\boldsymbol{r},t)\,,\quad i=1\ldots P\,. \tag{6.2}$$

In a steady-state condition, the transport equation reduces to

$$\hat{\Omega}\cdot\nabla\psi_g+\Sigma_{tg}\psi(\mathbf{r},\boldsymbol{\Omega})=\frac{1}{4\pi}\int_{4\pi}d\Omega'\sum_{g'=1}^{G}\Sigma_{sg\leftarrow g'}(\boldsymbol{r},\Omega'\cdot\Omega,t)\psi_{g'}(\boldsymbol{r},\Omega')+q_g\,, \tag{6.3}$$

where μ is the cosine of the angle. There exist several numerical methods to solve this system. However, here we focus on the discrete-ordinate method[57–59], regularly called S_N, since the code used to implement the ROM, i.e, Detran, relies on this method for angular discretization. For simplicity, consider a slab geometry with isotropic sources and scattering, for which the

steady-state transport equation simplifies to

$$\begin{aligned}\mu\frac{\partial\psi}{\partial x}+\Sigma_t(x)\psi(x,\mu) &= \frac{\Sigma_s(x)}{2}\int_{-1}^{1}d\mu'\psi(x,\mu')+\frac{S(x)}{2}\\ &= \frac{\Sigma_s(x)}{2}\phi(x)+\frac{S(x)}{2}\,.\end{aligned} \tag{6.4}$$

The discrete ordinates method consists of evaluating Eq. 6.4 only at discrete angles θ_n, where $\mu_n=\cos\theta_n$ and $n\in[1,N]$. In other words, we require

$$\mu_n\frac{\partial\psi}{\partial x}+\Sigma_t(x)\psi(x,\mu_n)=\frac{\Sigma_s(x)}{2}\phi(x)+\frac{S(x)}{2}\,. \tag{6.5}$$

Since ψ is computed only at discrete angles, the scalar flux is approximated using numerical quadrature, or

$$\phi(x)\approx\sum_{n=1}^{N}w_n\psi_n(x)\,, \tag{6.6}$$

where w_n is the weight associated with the nth angle. Different quadrature sets are used, depending on the type of geometry. Gauss-Legendre quadrature is generally used for 1-D and spherical geometries. To represent the angle-discretized system in an operator form, a *discrete-to-moment* operator $\mathcal{D}$, is introduced such that

$$\phi=\mathcal{D}\psi\,. \tag{6.7}$$

Moreover, we introduce a *moment-to-discrete* operator $\mathcal{M}$, defined as

$$\psi=\mathcal{M}\phi\,. \tag{6.8}$$

Finally, we define the operator

$$\mathcal{L}(\,\cdot\,)\equiv\left(\hat{\Omega}\cdot\nabla+\Sigma_t(\boldsymbol{r})\right)(\,\cdot\,)\,, \tag{6.9}$$

Thus, in a multiplying medium, the transport equation becomes

$$\mathcal{L}\boldsymbol{\psi} = \mathcal{M}\mathcal{S}\phi + \frac{1}{k_{\text{eff}}}\mathcal{M}\mathcal{F}\phi\,, \tag{6.10}$$

where $\mathcal{F}$ is the fission source operator. Note that the unknown in Eq. 6.10 is the angular flux ψ. In general, and particularly for reactor physics, our interest is in computing reaction rates for which only the scalar flux is needed. In practice, the angular flux is, therefore rarely stored explicit, but is instead computed on-the-fly during a sweep through the space-angle grid. This can be represented explicitly by manipulating the equations to be functions only of the scalar flux. To illustrate, consider Eq. 6.10. Multiplying through by the space-angle *transport sweep* operator $\mathcal{T} = \mathcal{D}\mathcal{L}^{-1}$, we have

$$\mathcal{D}\psi = \mathcal{D}\mathcal{L}^{-1}\mathcal{M}\mathcal{S}\phi + \frac{1}{k_{\text{eff}}}\mathcal{D}\mathcal{L}^{-1}\mathcal{M}\mathcal{F}\phi\,. \tag{6.11}$$

Rearranging yields

$$(\mathcal{I} - \mathcal{D}\mathcal{L}^{-1}\mathcal{M}\mathcal{S})\phi = \frac{1}{k_{\text{eff}}}\mathcal{D}\mathcal{L}^{-1}\mathcal{M}\mathcal{F}\phi\,. \tag{6.12}$$

Using the same operator notation for the time-dependent, flux and precursors equations and applying backward Euler approximation in time yields

$$\frac{1}{\Delta_t v_g}\left(\psi_g^{k+1} - \psi_g^k\right) + \mathcal{L}\psi_g^{k+1} = \mathcal{M}\sum_{g'=1}^{G}\left(\mathcal{S}_{gg'} + \mathcal{X}_{pg}\mathcal{F}_{g'}\right)\phi_{g'}^{k+1} + \mathcal{M}\sum_{i=1}^{I}\lambda_i\mathcal{X}_{dig}C_i^{k+1}\,, \tag{6.13}$$

and

$$\frac{1}{\Delta_t}\left(C_i^{k+1} - C_i^k\right) = \beta_i\sum_{g=1}^{G}\mathcal{F}_g\phi_g^{k+1} - \lambda_i C_i^{k+1}\,, \tag{6.14}$$

where k denotes the time step. Now, we can cast Eqs. 6.13 and 6.14 as a fixed-source problem. First, we solve Eq. (6.14) for the updated concentrations, which leads to

$$C_i^{k+1} = \frac{\Delta_t\beta_i}{1+\Delta_t\lambda_i}\sum_{g=1}^{G}\mathcal{F}_g\phi_g^{k+1} + \frac{C_i^k}{1+\Delta_t\lambda_i}\,, \tag{6.15}$$

Substituting this result into Eq. (6.13) yields

$$\begin{aligned}\frac{1}{\Delta_t v_g}\left(\psi_g^{k+1}-\psi_g^k\right)+\mathcal{L}_g\psi_g^{k+1}=\mathcal{M}\sum_{g'=1}^{G}\left(\mathcal{S}_{gg'}+\mathcal{X}_{pg}\mathcal{F}_{g'}\right)\phi_{g'}^{k+1}\\ +\mathcal{M}\sum_{i=1}^{I}\lambda_i\mathcal{X}_{dig}\left(\frac{\Delta_t\beta_i}{1+\Delta_t\lambda_i}\sum_{g'=1}^{G}\mathcal{F}_{g'}\phi_{g'}^{k+1}+\frac{C_i^k}{1+\Delta_t\lambda_i}\right).\end{aligned} \tag{6.16}$$

Rearranging again, we find

$$\begin{aligned}\left(\frac{1}{\Delta_t v_g}+\mathcal{L}_g\right)\psi_g^{k+1}=\mathcal{M}\sum_{g'=1}^{G}\left(\mathcal{S}_{gg'}+\left(\mathcal{X}_{pg}+\sum_{i=1}^{I}\lambda_i\mathcal{X}_{dig}\left(\frac{\Delta_t\beta_i}{1+\Delta_t\lambda_i}\right)\right)\mathcal{F}_{g'}\right)\phi_{g'}^{k+1}\\ +\mathcal{M}\sum_{i=1}^{I}\lambda_i\mathcal{X}_{dig}\left(\frac{C_i^k}{1+\Delta_t\lambda_i}\right)+\frac{\psi_g^k}{\Delta_t v_g}.\end{aligned} \tag{6.17}$$

Let

$$\tilde{\mathcal{L}}_g\equiv\frac{1}{\Delta_t v_g}+\mathcal{L}_g\,, \tag{6.18}$$

$$\tilde{\chi}_g\equiv\left(\mathcal{X}_{pg}+\sum_{i=1}^{I}\lambda_i\mathcal{X}_{dig}\left(\frac{\Delta_t\beta_i}{1+\Delta_t\lambda_i}\right)\right), \tag{6.19}$$

and

$$\tilde{q}_g^{k+1}\equiv\mathcal{M}\sum_{i=1}^{I}\lambda_i\mathcal{X}_{dig}\left(\frac{C_i^k}{1+\Delta_t\lambda_i}\right)+\frac{\psi_g^k}{\Delta_t v_g}. \tag{6.20}$$

Then, we write Eq. 6.17 as

$$\tilde{\mathcal{L}}_g\psi_g^{k+1}=\mathcal{M}\sum_{g'=1}^{G}\left(\mathcal{S}_{gg'}+\tilde{\mathcal{X}}_g\mathcal{F}_{g'}\right)\phi_{g'}^k+\tilde{q}_g^k\,. \tag{6.21}$$

Multiplication of this result by $\mathcal{D}\mathcal{L}_g^{-1}$ and rearranging yields

$$\left(\mathcal{I}-\mathcal{D}\mathcal{L}_g^{-1}\mathcal{M}\sum_{g'=1}^{G}\left(\mathcal{S}_{gg'}+\tilde{\mathcal{X}}_g\mathcal{F}_{g'}\right)\right)\phi_g^{k+1}=\mathcal{D}\mathcal{L}_g^{-1}\tilde{q}_g^k\,. \tag{6.22}$$

By dropping group indices, this can be generalized for the multi-group system as

$$\left(\mathcal{I} - \mathcal{D}\mathcal{L}^{-1}\mathcal{M}\left(\mathcal{S}+\mathcal{F}\right)\right)\phi^{k+1} = \mathcal{D}\mathcal{L}^{-1}\tilde{q}^{k}\,. \tag{6.23}$$

Thus, starting from an initial condition of ϕ and C, Eqs. 6.23 and 6.16 are used to march the system forward in time. By applying discretization in space, a compact form, the system can be represented as

$$\left(\begin{array}{c|c} \left(\left(\mathbf{I} - \mathbf{D}\mathbf{L}^{-1}\mathbf{M}\left(\mathbf{S}+\mathbf{F}\right)\right)\right) & \mathbf{0} \\ \hline \mathbf{P}_g & \mathbf{P}_l \end{array}\right)\left(\begin{array}{c} \boldsymbol{\phi}^{k+1} \\ \mathbf{C}^{k+1} \end{array}\right) = \left(\begin{array}{c} \mathbf{D}\mathbf{L}^{-1}\mathbf{q}^{\boldsymbol{k}} \\ \mathbf{C}^{k} \end{array}\right), \tag{6.24}$$

where $\mathbf{M}$, $\mathbf{L}$, $\mathbf{D}$, $\mathbf{F}$ are the corresponding spatially-discrete operators of the operators $\mathcal{M}$, $\mathcal{L}$, $\mathcal{D}$ and $\mathcal{F}$, respectively, $\mathbf{P}_g$ is the precursors gain matrix, and $\mathbf{P}_l$ is the a diagonal matrix whose elements are $1 + \Delta_t\lambda_i$. Note that the system in Eq. 6.24 has the form of $\mathbf{A}\mathbf{x} = \mathbf{b}$ and it can be solved using a linear solver at each time step.

6.3 Reduced-Order Model of the transport equation

Typically, the resulting FOM in Eq. 6.24 is sufficiently large that direct solvers are not efficient and iterative solvers must be used. Developing efficient iterative methods that provide rapid convergence is an active area of research across the computational engineering. Here, we extend the ROM developed in the previous chapters for the diffusion equation to approximate full-order models based on the transport equation. Let $\mathbf{U}_\phi$ and $\mathbf{U}_c$ be the POD basis sets generated for the flux and the precursors concentration, and $\mathbf{a}_\phi$(t) and $\mathbf{a}_c(t)$ be their respective temporal coefficients. The POD-Galerkin ROM approximating the system in Eq. 6.24 is defined as

$$\left(\begin{array}{c|c} \overbrace{\mathbf{U}_\phi^T(\mathbf{I} - \mathbf{D}\mathbf{L}^{-1}\mathbf{M}\left(\mathbf{S}+\mathbf{F}\right)\mathbf{U}_\phi}^{\mathbf{A_1}} & \mathbf{0} \\ \hline \mathbf{U}_c^T\mathbf{P}_g\mathbf{U}_\phi & \mathbf{U}_c^T\mathbf{P}_l\mathbf{U}_c \end{array}\right)\left(\begin{array}{c} \boldsymbol{a}_{\boldsymbol{\phi}}^{k+1} \\ \mathbf{a}_c^{k+1} \end{array}\right) = \left(\begin{array}{c} \mathbf{U}_{\boldsymbol{\phi}}^{\boldsymbol{T}}\mathbf{D}\mathbf{L}^{-1}\mathbf{q}^{\boldsymbol{k}} \\ \mathbf{a}_c^{k} \end{array}\right). \tag{6.25}$$

Note that this developed ROM has a form similar to that of the diffusion-based ROM. However, in practice, it introduces additional challenges with respect to the computational complexity and practical implementation not present in diffusion. First, consider the upper part of the right-hand side of Eq. 6.25, i.e, $\mathbf{U}_{\boldsymbol{\phi}}^{\boldsymbol{T}}\mathbf{D}\mathbf{L}^{-\mathbf{1}}\mathbf{q}^{\boldsymbol{k}}$. Evaluating this term requires computing two additional reduced matrices beside the ones in the left-hand side that are similar to that of diffusion ROM. In particular, this term can be expanded as

$$\mathbf{U}_{\phi}^{T}\mathbf{D}\mathbf{L}^{-1}\mathbf{q}^{k} = \overbrace{\mathbf{U}_{\phi}^{T}\mathbf{D}\mathbf{L}^{-1}\mathbf{U}_{\phi}}^{\mathbf{A}_2}\frac{\mathbf{a}_{\phi}^{k}}{v\Delta t} + \overbrace{\mathbf{U}_{\phi}^{T}\mathbf{D}\mathbf{L}^{-1}\mathbf{P}_{d}\mathbf{U}_{c}}^{\mathbf{A}_3}\mathbf{a}_{\mathbf{c}}{}^{k}\,. \tag{6.26}$$

The matrices $\mathbf{A}_2$ and $\mathbf{A}_3$ are functions of the operator $\mathbf{L}$, which contains the total cross section. In general, this cross section changes with time in most time-dependent problems. Consequently, these matrices must be updated at each time step and at each iteration within a time step for non-linear problems.

As stated previously, the projection has a computational complexity that is proportional to the original dimension of the system; hence this repetitive projection increases the ROM cost. In terms of the practical implementation, a major difference between the transport model and the diffusion model is that the transport operators, i.e, $\mathbf{D}\mathbf{L}^{-1}$ and $((\mathbf{I}-\mathbf{D}\mathbf{L}^{-1}\mathbf{M}\,(\mathbf{S}+\mathbf{F}))$ are often implemented in matrix-free operators. This means that an explicit matrix is not computed for these operators. Recall that the angular flux is computed at each cell by sweeping along the direction of neutron travel for which only information form the preceding cell is required. Moreover, the contribution to the scalar flux is made point-by-point to avoid storing the angular flux, making a matrix-free implementation more convenient for reducing the memory requirements.

Fortunately, operator-vector products can still be performed for such operators. In fact, performing one transport sweep, i.e, $\mathbf{L}^{-1}\boldsymbol{q}$, is equivalent to an operator-vector product. Since the projection is essentially a sequence of vector-matrix products, the reduced matrices needed to solve the ROM can be obtained for these matrix-free operators. To illustrate, let $\mathbf{B}$ be an arbitrary matrix-free operator for which we seek to compute the reduced form, $\mathbf{U}^{T}\mathbf{B}\mathbf{U}$. Using

the matrix-free multiplying function, operator-vector products are carried out independently for each column in $\mathbf{U}$ and the resultant columns are stored in a new explicit operator which is then multiplied on the left by $\mathbf{U}^T$ to produce the required projected operator. To solve the ROM in Eq. 6.25, this process was performed for the operators $\mathbf{U}^T(\mathbf{I} - \mathbf{M}(\mathbf{S} + \mathbf{F})\mathbf{U}$, $\mathbf{U}^T\mathbf{D}\mathbf{L}^{-1}\mathbf{M}\mathbf{U}$, and $\mathbf{U}^T\mathbf{D}\mathbf{L}^{-1}\mathbf{M}\mathbf{P}_d\mathbf{U}_c$.

Another challenge posed by these matrix-free operators in the context of time-dependent ROMs is that it is difficult to obtain an offline-online decomposition of the ROM that avoid performing an expensive projection at each time step. Recall, in the diffusion-based ROM, we used the matrix DEIM to approximate the problem operator before projecting it by decomposition in terms of time-independent matrices each weighted by a temporal coefficient. However, in the case of a matrix-free operator, it might be more convenient to employ a project-then-approximate approach. As proposed in chapter 4, one way to do that is by using interpolation to approximate each of the reduced matrix elements, where the order of interpolation depends on how the matrix elements change with time which is problem dependent. The interpolation can be done by generating pre-computed reduced matrices at different time instances, comprising an offline stage and performing the interpolation online at each time step. Alternatively, this can be done on the fly while solving the ROM by generating the interpolation matrices only needed over a specified coarse time mesh, i.e., macro step, and using them to interpolate and solve the ROM at micro steps within this macro step. These interpolations matrices are then updated for the next macro step and so on. The first method is favored in case the pre-computed matrices can be used for another problem, while the other one is better in terms of memory requirement since only a few matrices are needed to be stored at each macro step.

6.4 Applications

In this section, the ROM described in Eq. 6.25 was applied to two problems. The first represents a linear problem describing a delayed super-critical transient, while the second describes a prompt super-critical transient with feedback, and, hence, that required a non-

linear treatment.

6.4.1 The TWIGL benchmark

The TWIGL benchmark, described in chapter 4, was modeled using the transport equation as the full-order model with eight angles per octant. Here, the quarter core in the original benchmark was extended to a full core with an 80×80 spatial grid. As mentioned previously, the benchmark features two transients, a ramp and a step, in which the control rod movement is simulated by changing the thermal absorption cross section according to equations 4.14 and 4.15, respectively. Here, modified ramp and step transients were considered, and the core power was computed using the full-order model with a time step of 0.001 s. To build the ROM in Eq. 6.25, the POD sub-spaces $\mathbf{U}_\phi$ and $\mathbf{U}_c$ were obtained using snapshots of the scalar flux and the precursors concentration, respectively. For both transients, the same snapshots from the ramp model were used. Shown in Fig. 6.1 are the first 100 singular values of the fast and thermal group flux for the ramp transient.

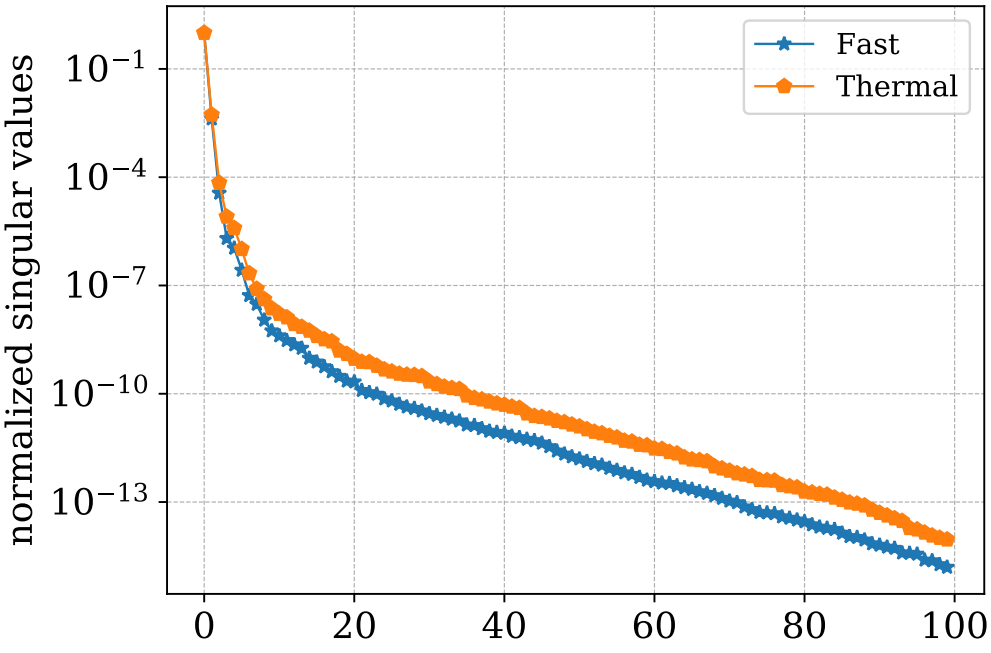


Figure 6.1: Normalized singular values for the ramp transient.

A rank of 20 was selected for each group flux and the precursor concentration. For the

ramp, the total transient time is 0.6 s in which the reactivity is inserted over time from 0.1s to 0.3s. Since this was simulated by changing the absorption cross section, the sweep operator $\mathbf{L}$ depends on time, which requires that the projected matrices have to be computed, at least 200 times if a direct projection is performed. As described in the previous section, this could introduce a cost that exceeds that of the original model, making the ROM of no benefit. To resolve this computational complexity, an interpolation based on pre-computed projected matrices was employed as follows:

1. The full operators were obtained and projected at times 0.0 s, 0.1 s, 0.2 s, and 0.3 s in an offline step.

2. At each time step, the appropriate, pre-computed matrices were selected, and the elements of the projected matrix were computed by linear interpolation.

3. Using the approximated projected matrices, the system in Eq. 6.25 was solved at each time step.

Following these steps, the core power was computed and the error with respect to the full-order model power was estimated. Moreover, a direct projection approach (i.e., projection at every step) was used to assess the approximation made by the interpolation. Shown in Fig. 6.2 is the relative error in the integral core power. Note that the time over which the error increased is the ramp time, where the approximation introduced by interpolation occurs. Figure 6.3 shows the maximum assembly power error using the interpolation approach.

For the step transient, the reduced operators must be updated just once at the time of the step reactivity insertion. This can be done online while solving the ROM, or one can exploit the pre-computed matrices generated for the ramp transient. In both cases, there is no approximation made to compute the projected matrices and hence they produce numerically the same solution. Shown in Fig. 6.4 is the relative error in the integral core power, while Fig. 6.5 shows the maximum assembly power.

Furthermore, to assess the transport ROM performance with respect to other approximate models, the diffusion solution for the ramp (and a modified step insertion) were obtained and

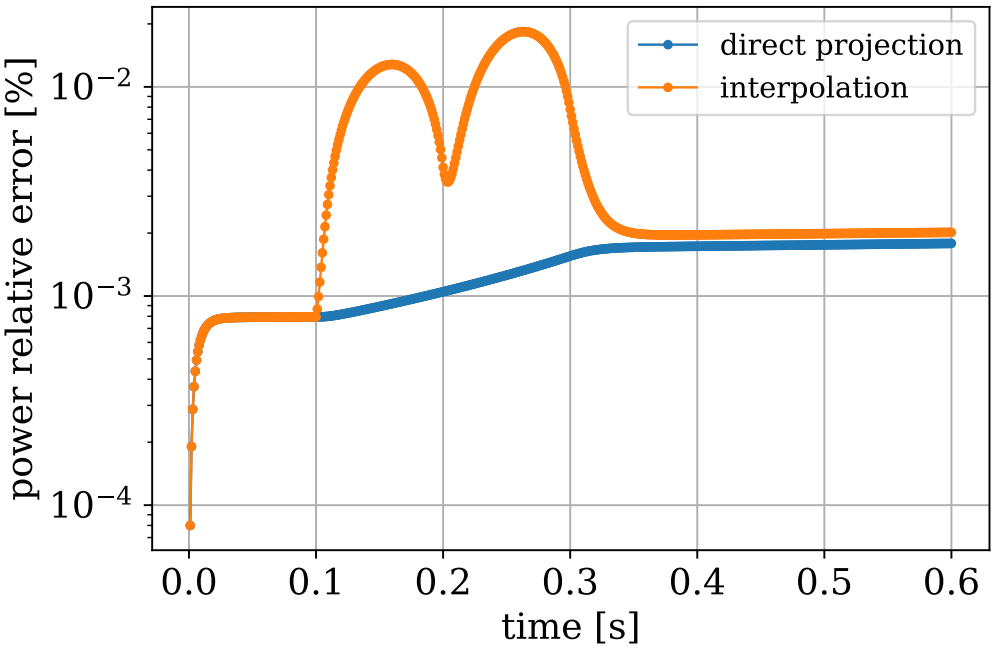


Figure 6.2: Relative error in the ramp transient power

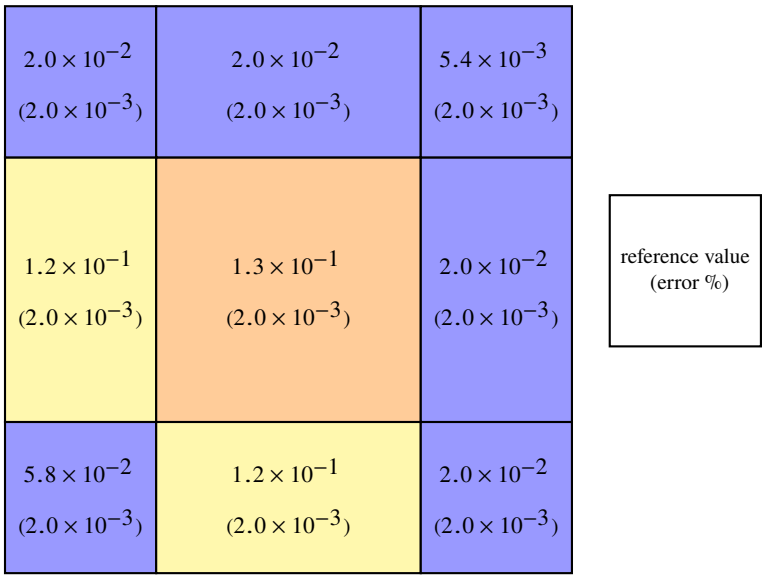


Figure 6.3: Relative error in the ramp transient assembly power.

compared with the transport solution, and the relative error is shown in Fig. 6.6. The primary conclusion to be drawn from Fig. 6.6 and Fig. 6.3 is that a relatively low-order, POD-Galerkin ROM yields substantially lower errors than a fine-mesh, diffusion approximation of the same FOM. A similar conclusion was drawn based on the results shown in Fig. 4.9 and Fig. 4.10,

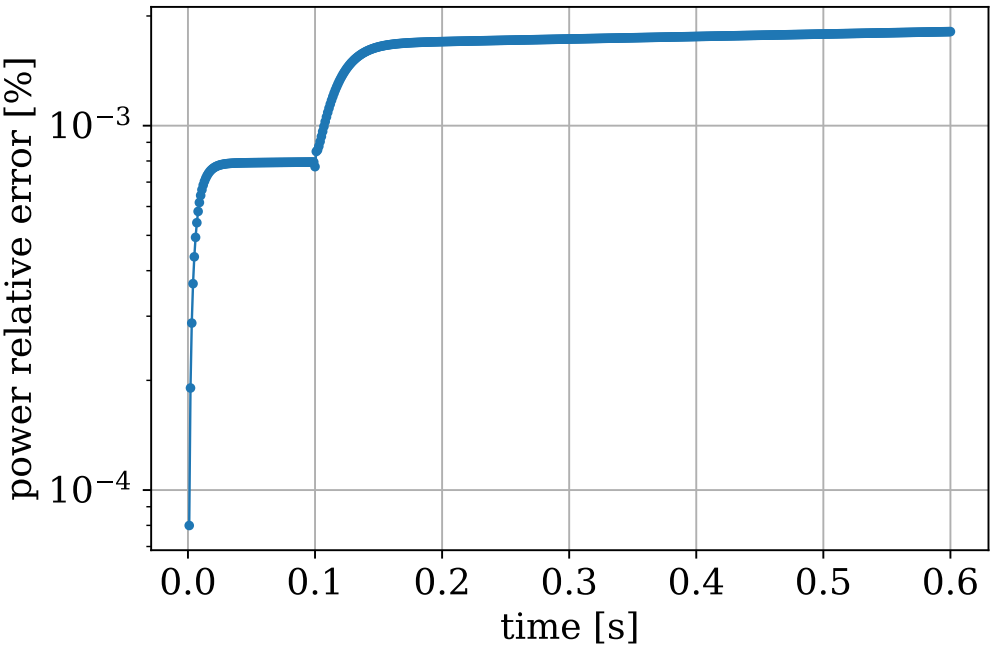


Figure 6.4: Relative error in the step transient power

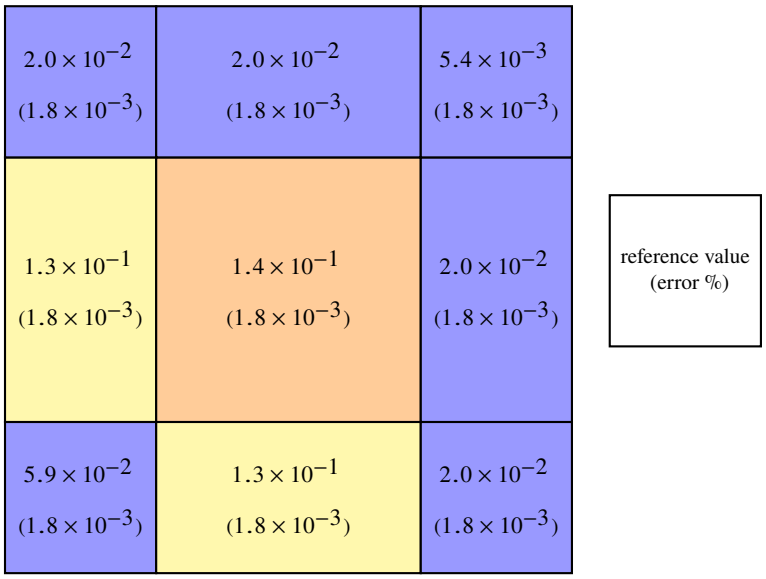


Figure 6.5: Relative error in the step transient assembly power

in which the ROM of a fine-time-step, diffusion-based FOM outperforms a coarse-time-step approximation of the same FOM. In other words, the POD-Galerkin ROM may more readily capture the low-rank dynamics of the higher-order system than do approximations based on direct "coarsening" of the FOM. Based on the times listed in Table 6.1, this better

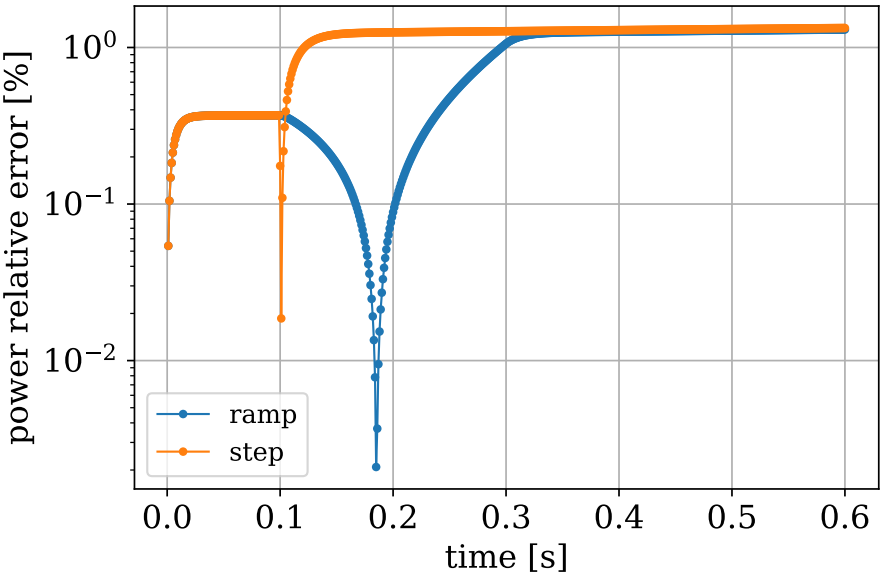


Figure 6.6: Error in diffusion solution relative to transport (FOM) solution.

representation of the low-rank dynamics comes at a *lower* computational cost then the associated diffusion model.

Table 6.1: Computational time of the FOMs and the ROMs for the TWIGL benchmark.

Transient	Transport FOM time (s)	Diffusion FOM time (s)	ROM time (s)
ramp	395	57	18
step	300	53	15

6.4.2 1-D nonlinear application

Here, a simple 1D application is used to demonstrate the transport ROM for non-linear problems, and at the same time to end the chapter with a starting point for future work. The slab reactor shown in Fig. 4.1 is used, where a rapid reactivity insertion was assumed by ejecting the control rod from 25% withdrawn to 50% withdrawn with constant velocity over a period of two seconds. The control rod movement was simulated by adjusting the thermal absorption cross section. The adiabatic heat-up and Doppler feedback models of the LRA benchmark (see 5.3 and 5.4) were used. Shown in Fig. 6.7 is the core power predicted with the full-order model solved using the Backward Euler with a time step of $0.001s$ and the

fixed point iteration.

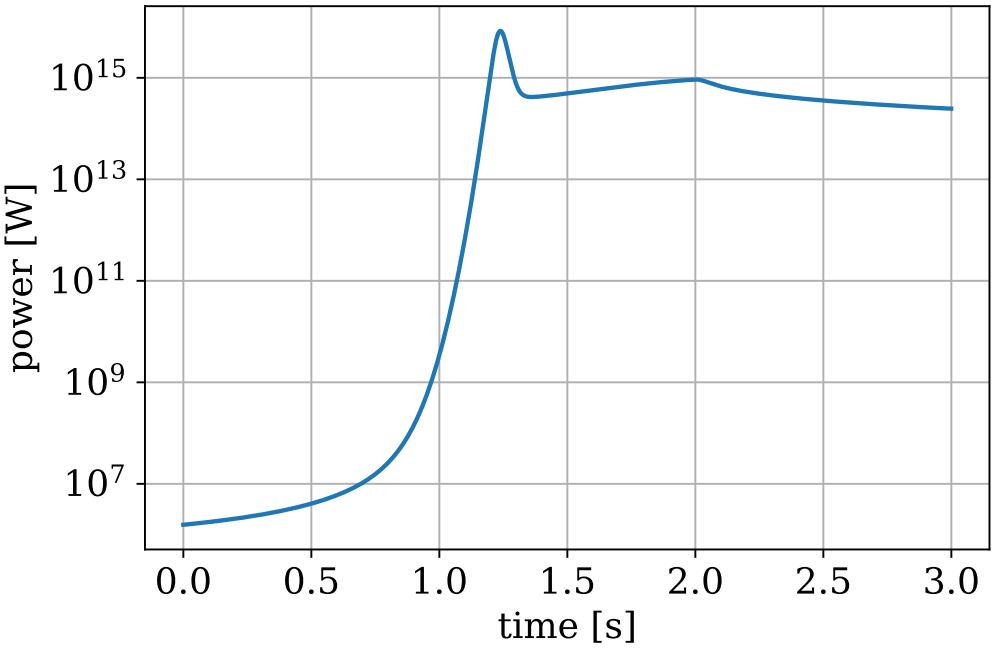


Figure 6.7: Slab reactor core power

The challenge of this problem is that the fast absorption cross section depends on the solution itself, and it needs to be updated while solving the system through non-linear iterations. Thus, interpolation based on pre-computed projected matrices cannot not fully preserve the coupling between the different physics. Alternatively, the interpolation matrices were generated on-the-fly to account for the material dependence on the solution, as explained in Sec.4.2. While solving the ROM, two projected matrices were computed at the beginning and the end of a macro step of 0.01s, where the fast absorption cross section was updated at the end of each step with the temperature obtained by integrating Eq. 5.3 using the macro step. The flux was obtained by integrating the system in Eq. 6.25 using a micro time step of 0.001s. At each micro step within the macro step, the elements of the projected matrices were computed by linearly interpolating the generated matrices. Moreover, we used direct projection as a way to show the validity of the implemented ROM with respect to the accuracy, and also to assess the interpolation error. The ROM was implemented with POD basis sets of rank 15 and 30 for the group flux and the precursor concentrations. Shown

in Fig. 6.8 are the relative errors in the ROM predicted power using the direct projection and the interpolation. The direct projection error supports the potential of the ROM for approximating transport models with good accuracy even with rapid non-linear transients. On the other hand, the interpolation error increases with the power peak with a maximum error of 2.7% due to coarse material discretization. Although the error is significantly higher than that of the direct projection, it is still better than a FOM with a coarse time mesh of $0.01s$ which results in an error at the power peak of 40%. Note that a higher interpolation order can be used to improve the error for which more than two matrices need to be stored at each macro step based of the selected order.

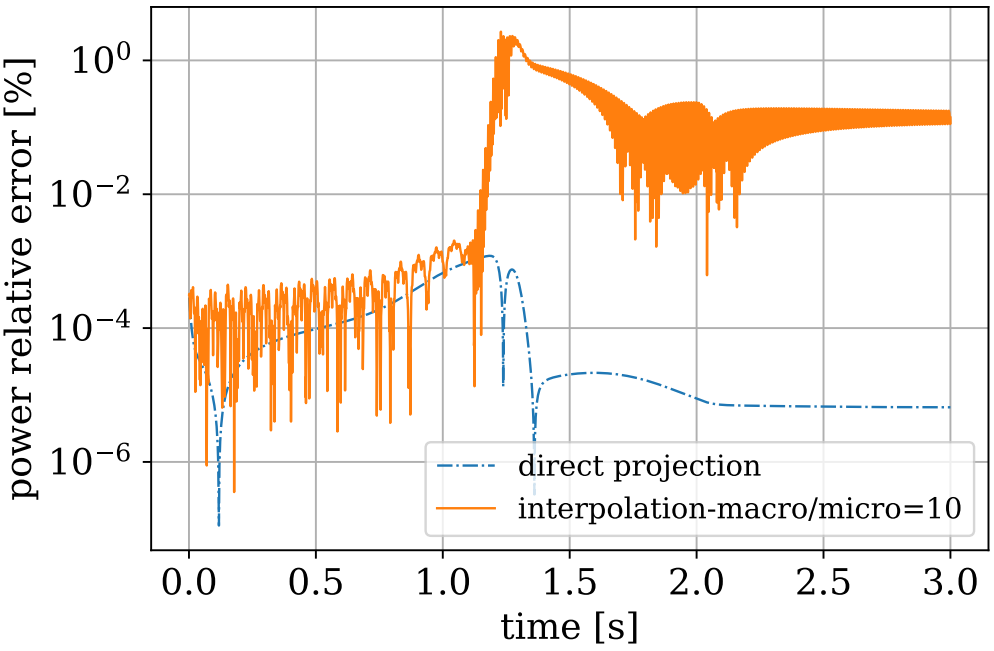


Figure 6.8: Slab reactor ROM relative power error

The discussion here focused on the ROM performance with respect to accuracy, however, more has to be done in the computational performance side since the interpolation ROM did not offer a speed up of the FOM. In particular, efficient methods need to employed to accelerate generating the projection matrices, in case of matrix-free solvers, otherwise the ROM could be impractical approach to reduce the computational cost of high-dimensional systems and this subject will be the focus of future work.

Chapter 7

Conclusion and Future Work

7.1 Conclusion

Such reduced-order modeling directly supports the need for predictive simulation of nuclear systems during transients, an activity that has become increasingly important for the exploration, design, and optimization of advanced reactors whose operating conditions live beyond those of existing designs. High fidelity models have been used to study the behavior of nuclear reactors under different scenarios and postulated events, where at each case the model is run with varying input parameters to account for model uncertainties and optimize the designs. This repetitive run of the simulations results in an enormous computational time, especially if the solutions are required on a fine space/time grid, which is desired for good accuracy. There have been many endeavors to overcome this computational cost by improving the discretization schemes, accelerating the iterative methods, or employing approximate models such as the point-kinetics, which relied upon a strong assumption regarding the spatial shape of the neutron flux in the reactor core.

In this work, we addressed the problem of computational cost via model-order reduction, in a way that preserves the spatial structure of the system. This approach is built on the fact that discretization schemes induce a correlation between model outputs at discrete points in space, besides the correlation due to the physical phoneme itself. Thus, although spatial

discretization of the underlying PDEs yields a solution that lives in a high-dimensional space, it can be efficiently approximated with significantly few degrees of freedom. Throughout this study, we apply a reduction technique based on the Proper-Orthogonal Decomposition to approximate the solution by projection over a lower-dimensional space. Combined with a Galerkin projection of the discretized system, a smaller number of equations is required to be solved. Different problems have been addressed to resolve various complexities. The previously developed deterministic transport code Detran was used to implement the ROM and test it for the different problems.

First, the implemented ROM was tested on the TWIGL benchmark, where the full-order model to be approximated was the neutron diffusion equation. Using this problem, several studies were conducted to understand the performance and the accuracy of the developed ROM. In particular, POD rank study was done for full-order models with different spatial resolutions. The study showed that the rank approximating the solution is weekly dependent on the underlying discretization. Thus, while the cost of the full-order models increases significantly as spatial refinement is done to increase the solution accuracy, the POD-Galerkin ROM can be used to obtain spatially refined solutions without adding similar cost over a ROM with lower spatial fidelity. In multi-query applications, this could lead to meaningful computational time reduction since the cost of obtaining the basis requiring the full-order solution will represent a small fraction of the time required to run the full-order model multiple times to perform such studies. Taking the developed ROM one step forward, a greedy-sampled POD subspace was acquired to run the ROM with different input parameters. The generated subspace required relatively few full-order solutions at selected parameters points in the domain of interest. Moreover, it was shown that this subspace could be trained using a coarser time mesh than that of the desired solution, reducing the upfront cost of the sampling procedures.

In terms of the computation efficiency, performance study showed that computing the projected operator accounts for a significant portion of the total ROM cost, if the projection is performed at each time step. For linear problems that do not include feedback from other physics, this could be resolved by restricting the projection to the time steps at which the

operator changes with time. However, problems with feedback induce non-linearity that requires the operator to be updated at each time step. Hence, it was essential to seek another level of approximation that enables an offline-online decomposition of the ROM such that the expensive projection is made once in an offline step while solving the reduced equation is solved in an online step with a cheaper cost. The matrix version of the Discrete Empirical Interpolation Method (DEIM) was used to achieve this ROM splitting via decomposing the operator into time-independent matrices, each weighted by a temporal coefficient. The LRA benchmark was used to demonstrate this approach with different spatial grids. The maximum relative error in the assembly power was less than 0.001%. Moreover, the offline-online ROM resulted in a speedup over that of a ROM with a direct projection at each time step. More importantly, its cost is weekly dependent on the original dimension of the system, resulting in a speedup that is more appreciable with increasing the spatial resolution, in which case the use of a ROM is of more value.

All the previous full-order models were based on the diffusion equation, and hence, the natural next step was to shift the calculation to the transport equation. The same steps used for the diffusion-based ROM can be used for the transport ROM. In fact, the transport equation can be written in a compact operator notation similar to that of the diffusion, where the operators include information coming from the angle discretization. However, a major difference in the actual implementation of the transport operator is that they exist in Detran as matrix-free operators. Moreover, the use of sweeping algorithms resulted in a ROM formulation requiring more operators to be projected, increasing the computational complexity of the transport ROM. The projection can still be performed for such operators, since vector-matrix operations are applicable to such operators, however, an offline-online decomposition was challenging to obtain.

Instead of using an approximate-then-project approach, as the matrix DEIM provides, a project- then-approximate approach is more convenient. Interpolation of the reduced matrices is one way that belongs to this approach. Currently, the ROM is implemented in a such a way that the reduced matrices elements can be approximated, at each time step, by interpolation of precomputed matrices.

The implementation was tested using the TWIGL benchmark. Results showed a maximum error in the assembly power in the order of $1 \times 10^{-3}\%$. Furthermore, an initial application to non-linear transport problems was presented in which a direct projection was used. Interpolation on-the-fly is more convenient for such problems since the operator elements are functions of the solution itself and, hence, are not known before the system is solved. The ROM provided good approximation of the core power with maximum error of about 5×10^{-4}; however, more work needs to be done to approximate the projected operators and improve the computational efficiency. It is important to mention that the ROM implementation for both the diffusion and the transport was done in a completely non-invasive way to the existing full-order models. The ROM required access to the problem operators, and it was built separately from the original model, which makes it suitable for actual implementation in other codes. Overall, the developed framework can be reliably used in different applications such as propagating systems uncertainties, understanding the behavior of nuclear reactors under different accident scenarios. In practice, the additional inputs required to run the ROM are basis sets for the different state variables, i.e, flux, and precurosrs. These basis sets can be precomputed using greedy sampling for different scenarios and over various time scales based on the target application. Note that the same basis sets can be used to approximate similar transients for the same system if their histories are included in the snapshots used to generate the basis, and, thus, the upfront fractional cost of generating the basis can be greatly reduced. Examples include ramp transients with different insertion or withdrawal times and rates.

Critically, the approach developed and the results provided herein demonstrate that a versatile, POD-Galerkin framework for reduced-order modeling is easily accessible for analysts using a variety of off-the-shelf diffusion and transport tools.

7.2 Future Work

This work has been made to implement the ROM and validate it for different problems. However, with the implementation in place, work can be done for improvements, and some

suggestion are summarized as follows:

1. Although the singular values resulting from the SVD of the full-order solution are a good indication of the error resulting from approximating the solution via projection over a subspace, rigorous error bounds will be more convenient to use in actual application. This will require formal mathematical analysis of the discretized PDEs.

2. Posterior error estimates are more efficient to use in the greedy procedures, such that the full-order model solution is obtained only for selected parameters points.

3. Methods to approximate matrix-free operators for non-linear problems need to be explored.

Bibliography

[1] Derek Gaston, Chris Newman, Glen Hansen, and Damien Lebrun-Grandie. Moose: A parallel computational framework for coupled systems of nonlinear equations. *Nuclear Engineering and Design*, 239(10):1768–1778, 2009.

[2] Thomas M Evans, Alissa S Stafford, Rachel N Slaybaugh, and Kevin T Clarno. Denovo: A new three-dimensional parallel discrete ordinates code in scale. *Nuclear technology*, 171(2):171–200, 2010.

[3] Peter Benner, Serkan Gugercin, and Karen Willcox. A survey of projection-based model reduction methods for parametric dynamical systems. *SIAM review*, 57(4):483–531, 2015.

[4] Peter J Schmid. Dynamic mode decomposition of numerical and experimental data. *Journal of fluid mechanics*, 656:5–28, 2010.

[5] Peter J Schmid, Larry Li, Matthew P Juniper, and O Pust. Applications of the dynamic mode decomposition. *Theoretical and Computational Fluid Dynamics*, 25(1):249–259, 2011.

[6] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: A system for large-scale machine learning. In *12th {USENIX} symposium on operating systems design and implementation ({OSDI} 16)*, pages 265–283, 2016.

[7] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830, 2011.

[8] Rabab Elzohery and Jeremy Roberts. Exploring transient, neutronic, reduced-order models using dmd/pod-galerkin and data-driven dmd. In *EPJ Web of Conferences*, volume 247, page 15019. EDP Sciences, 2021.

[9] Saifon Chaturantabut and Danny C Sorensen. Nonlinear model reduction via discrete empirical interpolation. *SIAM Journal on Scientific Computing*, 32(5):2737–2764, 2010.

[10] Jeremy Alyn Roberts. *Advanced response matrix methods for full core analysis*. PhD thesis, Massachusetts Institute of Technology, 2014.

[11] George I Bell and Samuel Glasstone. Nuclear reactor theory. Technical report, US Atomic Energy Commission, Washington, DC (United States), 1970.

[12] James J. Duderstadt and Louis J. Hamilton. *Nuclear Reactor Analysis*. Wiley, 1976.

[13] Weston M Stacey. *Nuclear reactor physics*. John Wiley & Sons, 2018.

[14] AF Henry, BA Worley, and AA Morshed. Spatial homogenization of diffusion theory parameters. Technical Report IAEA-TECDOC–231, International Atomic Energy Agency (IAEA), 1980.

[15] Kord S Smith. Assembly homogenization techniques for light water reactor analysis. *Progress in Nuclear Energy*, 17(3):303–335, 1986.

[16] Edward W Larsen and Jim E Morel. Advances in discrete-ordinates methodology. *Nuclear Computational Science*, pages 1–84, 2010.

[17] Young Ryong Park and Nam Zin Cho. Coarse-mesh angular dependent rebalance acceleration of the discrete ordinates transport calculations. *Nuclear science and engineering*, 148(3):355–373, 2004.

[18] Sandra Dulla, Ernest H Mund, and Piero Ravetto. The quasi-static method revisited. *Progress in Nuclear Energy*, 50(8):908–920, 2008.

[19] Ang Zhu, Yunlin Xu, and Thomas Downar. A multilevel quasi-static kinetics method for pin-resolved transport transient reactor analysis. *Nuclear Science and Engineering*, 182(4):435–451, 2016.

[20] Allan F Henry. The application of reactor kinetics to the analysis of experiments. *Nuclear Science and Engineering*, 3(1):52–70, 1958.

[21] Congjian Wang and Hany S Abdel-Khalik. Construction of accuracy-preserving surrogate for the eigenvalue radiation diffusion and/or transport problem. In *PHYSOR 2012: Conference on Advances in Reactor Physics - Linking Research, Industry, and Education*, 2012.

[22] Péter German and Jean C Ragusa. Reduced-order modeling of parameterized multi-group diffusion k-eigenvalue problems. *Annals of Nuclear Energy*, 134:144–157, 2019.

[23] Bernard Haasdonk and Mario Ohlberger. Reduced basis method for finite volume approximations of parametrized linear evolution equations. *ESAIM: Mathematical Modelling and Numerical Analysis*, 42(2):277–302, 2008.

[24] Peter Binev, Albert Cohen, Wolfgang Dahmen, Ronald DeVore, Guergana Petrova, and Przemyslaw Wojtaszczyk. Convergence rates for greedy algorithms in reduced basis methods. *SIAM journal on mathematical analysis*, 43(3):1457–1472, 2011.

[25] Zhang Chunyu and Chen Gong. Fast solution of neutron diffusion problem by reduced basis finite element method. *Annals of Nuclear Energy*, 111:702–708, 2018.

[26] Alberto Sartori, Antonio Cammi, Lelio Luzzi, and Gianluigi Rozza. A multi-physics reduced order model for the analysis of lead fast reactor single channel. *Annals of Nuclear Energy*, 87:198–208, 2016.

[27] Peter German, Jean C Ragusa, and Carlo Fiorina. Application of multiphysics model order reduction to doppler/neutronic feedback. *EPJ Nuclear Sciences & Technologies*, 5 (ARTICLE):17, 2019.

[28] Toby RF Phillips, Claire E Heaney, Brendan S Tollit, Paul N Smith, and Christopher C Pain. Reduced-order modelling with domain decomposition applied to multi-group neutron transport. *Energies*, 14(5):1369, 2021.

[29] Alberto Sartori, Davide Baroli, Antonio Cammi, Davide Chiesa, Lelio Luzzi, Roberto Ponciroli, Ezio Previtali, Marco Enrico Ricotti, Gianluigi Rozza, and Monica Sisti. Comparison of a modal method and a proper orthogonal decomposition approach for multi-group time-dependent reactor spatial kinetics. *Annals of Nuclear Energy*, 71: 217–229, 2014.

[30] Stefano Lorenzi. An adjoint proper orthogonal decomposition method for a neutronics reduced order model. *Annals of Nuclear Energy*, 114:245–258, 2018.

[31] Alberto Sartori, Davide Baroli, Antonio Cammi, Lelio Luzzi, and Gianluigi Rozza. A reduced order model for multi-group time-dependent parametrized reactor spatial kinetics. In *International Conference on Nuclear Engineering*, volume 45950, page V005T17A048. American Society of Mechanical Engineers, 2014.

[32] Kosuke Tsujita, Tomohiro Endo, and Akio Yamamoto. Fast reproduction of time-dependent diffusion calculations using the reduced order model based on the proper orthogonal and singular value decompositions. *Journal of Nuclear Science and Technology*, 58(2):173–183, 2021.

[33] Anthony L Alberti. *Reduced Order Modeling of Radiation Diffusion Kinetics via Proper Generalized Decomposition*. PhD thesis, Oregon State University, 2019.

[34] Yue Sun, Junhe Yang, Yahui Wang, Zhuo Li, and Yu Ma. A pod reduced-order model for resolving the neutron transport problems of nuclear reactor. *Annals of Nuclear Energy*, 149:107799, 2020.

[35] Lawrence Sirovich. Turbulence and the dynamics of coherent structure. part i, ii, iii. *Quat. Appl. Math.*, 3:583, 1987.

[36] Gianluigi Rozza, Dinh Bao Phuong Huynh, and Anthony T Patera. Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations. *Archives of Computational Methods in Engineering*, 15(3): 229–275, 2008.

[37] Jan S Hesthaven, Gianluigi Rozza, Benjamin Stamm, et al. *Certified reduced basis methods for parametrized partial differential equations*, volume 590. Springer, 2016.

[38] Karen Veroy, Christophe Prud'Homme, Dimitrios Rovas, and Anthony Patera. A posteriori error bounds for reduced-basis approximation of parametrized noncoercive and nonlinear elliptic partial differential equations. In *16th AIAA Computational Fluid Dynamics Conference*, page 3847, 2003.

[39] Ngoc-Cuong Nguyen, Gianluigi Rozza, and Anthony T Patera. Reduced basis approximation and a posteriori error estimation for the time-dependent viscous burgers equation. *Calcolo*, 46(3):157–185, 2009.

[40] Peter Benner, Mario Ohlberger, Albert Cohen, and Karen Willcox. *Model reduction and approximation: theory and algorithms*. SIAM, 2017.

[41] Alessandro Alla and J Nathan Kutz. Nonlinear model order reduction via dynamic mode decomposition. *SIAM Journal on Scientific Computing*, 39(5):B778–B796, 2017.

[42] Maxime Barrault, Yvon Maday, Ngoc Cuong Nguyen, and Anthony T Patera. An empirical interpolationmethod: application to efficient reduced-basis discretization of partial differential equations. *Comptes Rendus Mathematique*, 339(9):667–672, 2004.

[43] David Galbally, Krzysztof Fidkowski, Karen Willcox, and Omar Ghattas. Non-linear model reduction for uncertainty quantification in large-scale inverse problems. *International journal for numerical methods in engineering*, 81(12):1581–1608, 2010.

[44] Jacob K White et al. *A trajectory piecewise-linear approach to model order reduction of nonlinear dynamical systems*. PhD thesis, Massachusetts Institute of Technology, 2003.

[45] Michał Rewieński and Jacob White. Model order reduction for nonlinear dynamical systems based on trajectory piecewise-linear approximations. *Linear algebra and its applications*, 415(2-3):426–454, 2006.

[46] Kevin Carlberg, Ray Tuminaro, and Paul Boggs. Efficient structure-preserving model reduction for nonlinear mechanical systems with application to structural dynamics. In *53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference 20th AIAA/ASME/AHS Adaptive Structures Conference 14th AIAA*, page 1969, 2012.

[47] Federico Negri, Andrea Manzoni, and David Amsallem. Efficient model reduction of parametrized systems by matrix discrete empirical interpolation. *Journal of Computational Physics*, 303:431–454, 2015.

[48] Hermann Weyl. Das asymptotische verteilungsgesetz der eigenwerte linearer partieller differentialgleichungen (mit einer anwendung auf die theorie der hohlraumstrahlung). *Mathematische Annalen*, 71(4):441–479, 1912.

[49] Kevin Carlberg, Ray Tuminaro, and Paul Boggs. Preserving lagrangian structure in nonlinear model reduction with application to structural dynamics. *SIAM Journal on Scientific Computing*, 37(2):B153–B184, 2015.

[50] Jonathan H Tu, Clarence W Rowley, Dirk M Luchtenburg, Steven L Brunton, and J Nathan Kutz. On dynamic mode decomposition: theory and applications. *arXiv preprint arXiv:1312.0041*, 2013.

[51] Argonne Code Center. Benchmark problem book, report anl-7416 (suppl. 2). Technical report, Argonne National Laboratory, Argonne, IL, 1977.

[52] Jurij Kotchoubey. Polca-t neutron kinetics model benchmarking, 2015.

[53] D Gaston, G Hansen, S Kadioglu, DA Knoll, C Newman, H Park, C Permann, and W Taitano. Parallel multiphysics algorithms and software for computational nuclear

engineering. In *Journal of Physics: Conference Series*, volume 180, page 012012. IOP Publishing, 2009.

[54] Argonne code center: Benchmark problem book. Technical Report ANL-7416, Argonne National Laboratory, 1972.

[55] George Robert Keepin, TF Wimett, and RK Zeigler. Delayed neutrons from fissionable isotopes of uranium, plutonium and thorium. *Journal of Nuclear Energy (1954)*, 6(1-2): IN2–21, 1957.

[56] RJ Tuttle. Delayed-neutron data for reactor-physics analysis. *Nuclear Science and Engineering*, 56(1):37–71, 1975.

[57] Elmer Eugene Lewis and Warren F Miller. *Computational methods of neutron transport*. American Nuclear Society, 1993.

[58] Marvin L Adams and Edward W Larsen. Fast iterative methods for discrete-ordinates particle transport calculations. *Progress in nuclear energy*, 40(1):3–159, 2002.

[59] GL Marchuk and VI Lebedev. *Numerical methods in the theory of neutron transport*. Routledge, 1986.

[60] A. Hébert. *Applied Reactor Physics*. Presses Internationales Polytechnique, 2009. ISBN 2553014368.

Appendix A

Numerical Methods

A.1 Backward Euler Integration

Backward Euler is an implicit time integration scheme that has the advantage of being stable for large time steps in contrary to the Forward Euler method. Consider the ordinary differential equation

$$\frac{d\mathbf{y}(t)}{dt} = \mathbf{A}\mathbf{y}(t)\,, \tag{A.1}$$

with initial condition $\mathbf{y}(t_0) = \mathbf{y}_0$. First, the time domain is discretized into sub-intervals each with a time step Δt. Recall Taylor series

$$\mathbf{y}^k = \mathbf{y}^{k+1} - \Delta t \frac{d\mathbf{y}}{dt}\Big|_{k+1} + \Delta t^2 \frac{d^2\mathbf{y}}{dt^2}\Big|_{k+1} + \ldots..\,, \tag{A.2}$$

where $\mathbf{y}^k = \mathbf{y}(t_0) + k\Delta t$. By keeping the first two terms and rearranging

$$\frac{d\mathbf{y}}{dt}\Big|_{k+1} = \frac{\mathbf{y}^{k+1} - \mathbf{y}^k}{\Delta t}\,. \tag{A.3}$$

Using the right hand side of Eq. A.1 and rearranging, we get

$$\mathbf{y}^k = (\mathbf{I} - \mathbf{A}\Delta t)\mathbf{y}^{k+1}\,. \tag{A.4}$$

Note that this has a form of $\mathbf{Ax} = \mathbf{b}$ and can be solved using a linear solver at each time step, where iteration methods might be required for non-linear problems. The method has a local truncation error of $\mathcal{O}(\Delta t^2)$ and global error at any time t of $\mathcal{O}(\Delta t)$.

A.2 Mesh-Centered Finite Difference

The Mesh-Centered Finite Difference is one of the most widely used methods for spatially discretizing the diffusion equation. For illustration, consider a 2-D, one-group diffusion equation.

$$-\nabla D(\boldsymbol{r})\nabla\phi(\boldsymbol{r}) + \Sigma_r(\boldsymbol{r})\phi(\boldsymbol{r}) = S(\boldsymbol{r})\,, \tag{A.5}$$

where Σ_r is the "removal" cross section. In the one-group approximation, $\Sigma_r = \Sigma_a$. To solve Eq. A.5 numerically, we use a mesh-centered, finite-difference approach, in which cell materials and sources are taken to be constant[60]. By integrating Eq. A.5 over the volume V_{ijk}, one obtains

$$\begin{aligned}\int_{x_{i-1/2}}^{x_{i+1/2}} dx \int_{y_{j-1/2}}^{y_{j+1/2}} dy \Big\{ -\nabla D(\boldsymbol{r})\nabla\phi(\boldsymbol{r}) + \Sigma_r(x,y)\phi(x,y) \Big\} \\ = \int_{x_{i-1/2}}^{x_{i+1/2}} dx \int_{y_{j-1/2}}^{y_{j+1/2}} dy\, S(x,y)\,,\end{aligned} \tag{A.6}$$

or

$$\begin{aligned}-D_{ij}\Big[\Delta_{y_j}\Big(\phi_x(x_{i+1/2},y_j) - \phi_x(x_{i-1/2},y_j)\Big) \\ +\Delta_{x_i}\Big(\phi_y(x_i,y_{j+1/2}) - \phi_y(x_i,y_{j-1/2})\Big)\Big] \\ +\Delta_{x_i}\Delta_{y_j}\Sigma_{r,ij}\phi_{ij} = \Delta_{x_i}\Delta_{y_j}S_{ij}\,,\end{aligned} \tag{A.7}$$

where the cell-centered flux (equal to the average flux) is defined as

$$\phi_{ij} \equiv \frac{1}{\Delta_{x_i}}\frac{1}{\Delta_{y_j}}\int_{x_{i-1/2}}^{x_{i+1/2}} dx \int_{y_{j-1/2}}^{y_{j+1/2}} dy\, \phi(x,y)\,, \tag{A.8}$$

the cell-average source is

$$S_{ij} \equiv \frac{1}{\Delta_{x_i}} \frac{1}{\Delta_{y_j}} \int_{x_{i-1/2}}^{x_{i+1/2}} dx \int_{y_{j-1/2}}^{y_{j+1/2}} dy\, S(x,y)\,, \tag{A.9}$$

and, for example, $\phi_x(x_{i+1/2}, y_j)$ is the derivative of ϕ with respect to x, averaged over y, and evaluated at $x = x_{i+1/2}$. To evaluate the partial derivatives ϕ_x and ϕ_y in Eq. A.7, Taylor-series expansions are employed. For the x-directed terms at the left (or west) boundary, let

$$\phi(x_{i-1}, y_j) \approx \phi(x_{i-1/2}, y_j) - \frac{\Delta x_{i-1}}{2} \phi_x^+(x_{i-1/2}, y_j) \tag{A.10}$$

and

$$\phi(x_i, y_j) \approx \phi(x_{i-1/2}, y_j) + \frac{\Delta x_i}{2} \phi_x^-(x_{i-1/2}, y_j)\,, \tag{A.11}$$

with similar expressions for the right x boundary and the y- term, The $+$ and $-$ superscripts on the partial derivatives indicate forward and backward extrapolation from the cell midpoint, respectively. By continuity of net current, we must have

$$D_{i-1,j} \phi_x^+(x_{i-1/2}, y_j) = D_{ij} \phi_x^-(x_{i-1/2}, y_j)\,. \tag{A.12}$$

Multiplication of Eq. A.10 by $D_{i-1,jk}/\Delta x_{i-1}$ and Eq. A.11 by $D_{ij}/\Delta x_i$, adding the results, and rearranging leads to

$$\phi_{i-1/2,j} = \frac{D_{i-1,j}\phi_{i-1,j}\Delta_{x_i} + D_{ij}\phi_{ij}\Delta x_{i-1}}{D_{i-1,j}\Delta_{x_i} + D_{ij}\Delta x_{i-1}}\,. \tag{A.13}$$

Substituting this into the Eq. A.11 gives

$$\begin{aligned} \phi_x(x_{i-1/2}, y_j) &= \frac{2}{\Delta_{x_i}} \left(\phi_{ij} - \frac{D_{i-1,j}\phi_{i-1,j}\Delta_{x_i} + D_{ij}\phi_{ij}\Delta x_{i-1}}{D_{i-1,j}\Delta_{x_i} + D_{ij}\Delta x_{i-1}} \right) \\ &= \frac{2}{\Delta_{x_i}} \left(\frac{D_{i-1,j}\phi_{ij}\Delta_{x_i} + D_{ij}\phi_{i-1,j}\Delta x_{i-1}}{D_{i-1,j}\Delta_{x_i} + D_{ij}\Delta x_{i-1}} \right) \end{aligned} \tag{A.14}$$

or

$$\phi_x(x_{i-1/2}, y_j) = 2D_{i-1,j}\left(\frac{\phi_{ij} - \phi_{i-1,j}}{\Delta_{x_i} D_{i-1,j} + \Delta x_{i-1} D_{ij}}\right), \tag{A.15}$$

Similarly, one finds

$$\phi_x(x_{i+1/2}, y_j) = 2D_{i+1,j}\left(\frac{\phi_{i+1,j} - \phi_{ij}}{\Delta_{x_i} D_{i+1,j} + \Delta x_{i+1} D_{ij}}\right). \tag{A.16}$$

These equations and the equivalents for y are substituted into Eq. A.7 to obtain a set of *internal equations*.

Internal Equation

Substitution of Eqs. A.15 and A.16 into Eq. A.7 leads to

$$\begin{aligned} -D_{ij}\Bigg\{\Delta_{y_j}\Bigg[2D_{i+1,j}\left(\frac{\phi_{i+1,j} - \phi_{i,j}}{\Delta_{x_i} D_{i+1,j} + \Delta x_{i+1} D_{ij}}\right) + \\ 2D_{i-1,j}\left(\frac{\phi_{i-1,j} - \phi_{i,j}}{\Delta_{x_i} D_{i-1,j} + \Delta x_{i-1} D_{ij}}\right)\Bigg] \ldots + \Bigg\} + \\ \Delta_{x_i}\Delta_{y_j}\Sigma_{r,ij}\phi_{ij} = \Delta_{x_i}\Delta_{y_j} S_{ij}\,. \end{aligned} \tag{A.17}$$

Equation (A.17) represents the balance of neutrons in the cell (i, j). For more simplification, define a coupling coefficient

$$\tilde{D}_{i+1/2,j} \equiv \frac{2D_{i+1,j} D_{ij}}{\Delta_{x_i} D_{i+1,j} + \Delta_{x_{i+1}} D_{ij}}, \tag{A.18}$$

with similar coefficients for each direction. Then we can rewrite Eq. A.17 as

$$\begin{aligned} \frac{\tilde{D}_{i+1/2,j}}{\Delta_{x_i}}\Big(\phi_{ij} - \phi_{i+1,j}\Big) + \frac{\tilde{D}_{i-1/2,j}}{\Delta_{x_i}}\Big(\phi_{ij} - \phi_{i-1,j}\Big) \\ + \frac{\tilde{D}_{i,j+1/2}}{\Delta_{y_j}}\Big(\phi_{ij} - \phi_{i,j+1}\Big) + \frac{\tilde{D}_{i,j-1/2}}{\Delta_{y_j}}\Big(\phi_{ij} - \phi_{i,j-1}\Big) \\ \Big(\phi_{ij} - \phi_{i,j}\Big) + \Sigma_{r,ij}\phi_{ij} = S_{ij}\,. \end{aligned} \tag{A.19}$$

Note that each term on the left with a coupling coefficient represents the net leakage from a surface divided by the area of that surface.

Boundary Equations

At the boundaries, we employ the albedo condition[60]

$$\frac{1}{2}D(x,y)\nabla\phi(x,y)\cdot\hat{n}(x,y)+\frac{1}{4}\frac{1-\alpha(x,y)}{1+\alpha(x,y)}\phi(x,y)=0\,, \tag{A.20}$$

where $\hat{n}$ is the outward normal, α describes the albedo condition ($\alpha=0$ for vacuum, and $\alpha=1$ for reflection). In three dimensions, a mesh cell has six surfaces, some of which may be part of a global surface. As an example, we consider the west global boundary:

$$\text{west boundary}: \quad -\frac{1}{2}D_{1j}\phi_x(x_{1/2},y_j)+\frac{1}{4}\frac{1-\alpha}{1+\alpha}\phi_{1/2,j}=0\,. \tag{A.21}$$

Because this expression contains ϕ at the edge, we again need to employ Taylor expansions. For the west boundary, note that

$$\phi_{1j}\approx\phi_{1/2,j}+\frac{\Delta_{x_i}}{2}\phi_x(x_{1/2},y_j)\,, \tag{A.22}$$

and we rearrange to get

$$\phi_{1/2,j}=\phi_{1j}-\frac{\Delta_{x_i}}{2}\phi_x(x_{1/2},y_j)\,. \tag{A.23}$$

Placing this into the albedo condition yields

$$0=-\frac{1}{2}D_{1j}\phi_x(x_{1/2},y_j)+\frac{1}{4}\frac{1-\alpha}{1+\alpha}\left(\phi_{1j}-\frac{\Delta_{x_i}}{2}\phi_x(x_{1/2},y_j)\right), \tag{A.24}$$

and solving for $\phi_x(x_{1/2},y_j)$ gives

$$\phi_x(x_{1/2},y_j)=\frac{2(1-\alpha)\phi_{1j}}{4(1+\alpha)D_{1jk}+\Delta_{x_1}(1-\alpha)}\,. \tag{A.25}$$

For the east, we similarly find

$$\phi_x(x_{I+1/2}, y_j) = -\frac{2(1-\alpha)\phi_{Ij}}{4(1+\alpha)D_{Ij} + \Delta_{x_I}(1-\alpha)}\,, \tag{A.26}$$

and likewise for the other surfaces. Each of these is placed into the proper partial derivative of Eq. A.7. For example, the leakage contribution on the left hand side of Eq. A.19 due to leakage from the west boundary is transformed as follows

$$\frac{\tilde{D}_{1/2,j}}{\Delta_{x_1}}\Big(\phi_{0j} - \phi_{1j}\Big) \rightarrow \frac{2D_{1j}(1-\alpha)\phi_{1j}}{(4(1+\alpha)D_{1j} + \Delta_{x_1}(1-\alpha))\Delta_{x_1}}\,. \tag{A.27}$$

The derived equations for each cell can be combined to compose a linear system in the form

$$\mathbf{L}\boldsymbol{\phi} = \mathbf{Q} \tag{A.28}$$