MULTIPHYSICS MODEL ORDER REDUCTION FOR MOLTEN SALT REACTORS

A Dissertation

by

PETER GERMAN

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

DOCTOR OF PHILOSOPHY

Committee Chair,	Jean C. Ragusa
Committe Co-chair,	Mauricio E. Tano Retamales
Committee Members,	Marvin L. Adams
	Andrea Bonito
	Jim E. Morel
Head of Department,	Michael Nastasi

August 2021

Major Subject: Nuclear Engineering

ABSTRACT

This work presents a model-order reduction approach for parametric multiphysics problems. The developed method utilizes the intrusive Proper Orthogonal Decomposition aided Reduced Basis technique (POD-RB) which builds suitable, only space-dependent, reduced subspaces for the solution field variables using snapshots generated by higher fidelity Full-Order Models (FOMs). The basis functions of the generated reduced subspaces together with the original operators of the FOM are then used to create parametric Reduced-Order Models (ROMs). The evaluation of these ROMs is computationally inexpensive compared to the FOM, therefore, they can be used as emulators to speed up multi-query applications which require the solution of the same model with different parameter configurations. Typical examples of such tasks are design optimization and uncertainty quantification. The applicability of the developed method is demonstrated on Molten Salt Reactors (MSRs) whose simulation can be computationally expensive, since is requires the solution of coupled fluid dynamics, neutronics and heat transfer problems. In this work, we use the porous-medium incompressible Navier-Stokes and energy equations coupled with a multi-group neutron diffusion equation accounting for the drift of the delayed neutron precursors. Models involving turbulent flows employ the Reynolds-Averaged Navier-Stokes (RANS) approach with a Boussinesq eddy viscosity approximation, while buoyancy effects are modeled using the Boussinesq buoyancy approximation. The FOM is created and solved using GeN-Foam, an OpenFOAM finite volume library based open source multiphysics solver. To assess the applicability of the derived POD-RB method, GeN-ROM, an OpenFOAM-based ROM framework has been created. The ROMs generated by this framework are tested on multiple parametric steady-state and transient scenarios of two MSRs: the Molten Salt Fast Reactor (MSFR) concept and the Molten Salt Reactor Experiment (MSRE). The results obtained throughout the validation tests indicate that the developed method is able to yield accurate multiphysics ROMs which can be 1-5 orders of magnitude faster compared to the FOM. To show potential applications of the ROMs, we carry out the uncertainty quantification of numerous quantities of interest (QoIs), e.g., effective multiplication

factor, effective delayed neutron fraction or maximum temperature, using the ROMs as the emulators. The results show that computationally demanding multi-query tasks, such as the generation of Sobol Indices or accurate statistical moments of the QoIs, become 1-3 orders of magnitude faster using the ROMs.

DEDICATION

To my mother, Erzsébet, my father, István, my sister, Borbála, my grandmother, Mária and my girlfriend, Linnea.

ACKNOWLEDGMENTS

First of all, I would like to thank my supervisor, Dr. Jean C. Ragusa for all the guidance, help, advice and kind encouragement from the past 4 years. Without his caring mentorship, this dissertation would have never seen daylight. Moreover, I am grateful to Dr. Mauricio E. Tano for his guidance in the field of computational fluid dynamics together with the inspiring conversations. I would also like to thank Dr. Carlo Fiorina for helping me through my struggles with OpenFOAM, paper reviews and the opportunity to visit Laussane. Special thanks to Jun Shi for lending me his numerical model of the Molten Salt Reactor Experiment for the simulations in Chapter 7 of this dissertation. Furthermore, I am especially grateful to Charles "Rob" Schafer, Logan Harbour, Patrick Behne, Bobert Terner, Andrew Hermosillo, Mateusz Marciniak and other students at Texas A&M University for the fun time spent together in the Office and their help with adjusting to a new way of life in a foreign country.

Lastly, but most importantly, I would like to express my deepest gratitude to my family in Hungary, and Linnea of College Station without whose support I would not have been able to reach this point in my professional carrier and personal growth. Örökké hálás maradok ezért!

CONTRIBUTORS AND FUNDING SOURCES

Contributors

This work was supported by the dissertation committee consisting of Professors Jean C. Ragusa (chair), Mauricio E. Tano Retamales (co-chair), Marvin L. Adams (member), Jim E. Morel (member) of the Department of Nuclear Engineering and Professor Adrea Bonito (member) of the Department of Mathematics.

The Full-Order numerical model of the Molten Salt Reactor Experiment used for the training of the Reduced-Order Models in Chapter 7 has been provided by Jun Shi from University of California, Berkeley.

All other work conducted for the dissertation was completed by the student independently.

Funding Sources

This research was carried out with funding received from the DOE Office of Nuclear Energy's NEUP grant No. DE-NE0008651.

TABLE OF CONTENTS

Page

LIST OF FIGURES

FIGURE Page

LIST OF TABLES

TABLE Page

1. INTRODUCTION

This chapter is dedicated to discuss the concept of model order reduction including the motivation and possible applications together with a brief, literature-based review of the specific techniques which provide alternatives to the one used in this work. The method in this work is reviewed in detail later, in Chapter 2. Since the aim of this dissertation is to apply model order reduction to typical problems arising during the simulations of Molten Salt Reactors (MSRs), a history-based review of the rector concept is included in this chapter as well. Lastly, we conclude this chapter by summarizing the aims and content of this dissertation.

1.1 Motivation for Model Order Reduction

The modeling and simulation of complex systems is common in many fields of science, including nuclear engineering. Oftentimes, this requires the discretization, in phase-space, of a system of partial differential equations, yielding an algebraic system of equations of possibly large size whose solution can be, in many cases, computationally expensive to obtain. In this work, these computationally expensive models are referred to as Full-Order Models (FOMs). In most cases, simulations using high-fidelity FOMs may not be optimal for multi-query applications where one has to carry out a large number of simulations with changing model parameters. Examples of such applications include design optimization and uncertainty quantification, where solutions are needed for each new realization of the uncertain parameters and/or for each change in the design parameters. Model order reduction consists of a set of empirical and mathematical techniques that aims at lowering the computational complexity of the FOMs, by creating Reduced-Order Models (ROMs) that sacrifice a controllable degree of accuracy for gains in simulation speed. Thus, the utilization of fast-running ROMs can greatly alleviate the computational burden of a multi-query problem.

It is worth noting that, in this work, we consistently call these simplified models ROMs; however, many other disciplines employ another terminology for such concepts. In machine learning

and data science they are often called surrogates or emulators, meaning that they mimic the behavior of an underlying response function with (hopefully) requiring less computational effort to evaluate.

Physics-informed ROMs have been used for decades in nuclear engineering. Examples include the utilization of sub-channel codes instead of CFD or point reactor kinetics instead of time-dependent neutron transport equations. However, in recent years new techniques have emerged and can be categorized into several, sometimes overlapping groups based on whether they are physics-informed or data-driven, intrusive or non-intrusive, to mention a few. In the following sections, an overview of some of the model order reduction techniques commonly used today is given by grouping the methods based on if they are intrusive or non-intrusive. The intrusiveness of a method, in this work, refers to the need to access the operators in the FOM. If a method does not require the knowledge of such operators, it is considered to be non-intrusive.

1.2 Non-intrusive Approaches

Let us assume that there is a specific Quantity of Interest (QoI) for which the Full-Order Model needs to be evaluated. This can be the solution at specific locations, extreme values of the solution or the full solution state vector itself, to name a few. For the time being, this response will be denoted as

$$\mathrm{QoI}(\theta(t, \boldsymbol{r}; \boldsymbol{\mu}), \boldsymbol{\mu})$$

expressing that this quantity depends on a space-, time-, and parameter-dependent solution, $\theta(\boldsymbol{r}, t; \boldsymbol{\mu})$, and possibly on additional model parameters as well. Altogether d model parameters are assumed, therefore the vector containing these can be expressed as $\boldsymbol{\mu} = [\mu_1, ..., \mu_d]^T$. A common feature of non-intrusive methods is that they treat the FOM as a black box, meaning that the means of obtaining the response of the FOM and the dependence of the QoI on the solution are not considered; the FOM only generates data which is later used to generate approximate functions for the QoI. This also means that every non-intrusive method is data-driven at the same time. Among the oldest non-intrusive ROMs, one can mention polynomial regression and fitting techniques or

by simply choosing the nearest point in a data set. However, for the sake of brevity, only a few recently emerging techniques within nuclear engineering are reviewed in this section.

1.2.1 Gaussian Processes

A Gaussian Process is defined as the collection of random variables whose joint distribution is Gaussian. The following brief review follows the one in [8] with the approximated function being our quantity of interest. Furthermore, it is assumed here that the QoI is a scalar-valued function with the elements of $\boldsymbol{\mu}$ as arguments. The extension of the following description to vector-valued problems is also discussed in [8]. Next, we assume that the quantity of our interest can be simulated as a Gaussian Process, or

$$\mathrm{QoI}(\boldsymbol{\mu}) \sim \mathcal{GP}(\boldsymbol{m}(\boldsymbol{\mu}), \boldsymbol{k}(\boldsymbol{\mu}, \boldsymbol{\mu}')), \tag{1.1}$$

where $\boldsymbol{m}(\boldsymbol{\mu})$ and $\boldsymbol{k}(\boldsymbol{\mu}, \boldsymbol{\mu}')$ are mean and covariance functions that fully specify the given Gaussian Process. The mean function is considered to be constant zero since in most cases the available data can be centered around the mean values. The kernel function, on the other hand, describes the covariance between different random variables used in the multivariate Gaussian. The choice of this function can add additional information about the system which can considerably influence the accuracy of the predictions given by the ROM. The interested reader is referred to [9] for the comparison of different kernel functions.

It is assumed that there are N_s training samples available with the corresponding $\boldsymbol{\mu}_i$ parameter vectors and $\mathrm{QoI}(\boldsymbol{\mu}_i)$ $(i = 1, ..., N_s)$ evaluations organized in matrix $\boldsymbol{M}_\mu$ and vector $\boldsymbol{q}$, respectively. Using the information in these samples, we would like to estimate the function values ($\boldsymbol{q}'$) at an additional n test parameter samples organized in matrix $\boldsymbol{M}'_\mu$. These values can be computed using the conditioning of the joint distribution which include all of the test and training samples:

$$\begin{bmatrix} \boldsymbol{q} \\ \boldsymbol{q}' \end{bmatrix} \sim \mathcal{N}\left(\boldsymbol{0}, \begin{bmatrix} K(\boldsymbol{M}_\mu, \boldsymbol{M}_\mu) + \sigma_n^2 I & K(\boldsymbol{M}_\mu, \boldsymbol{M}'_\mu) \\ K(\boldsymbol{M}'_\mu, \boldsymbol{M}_\mu) & K(\boldsymbol{M}'_\mu, \boldsymbol{M}'_\mu) \end{bmatrix} \right). \tag{1.2}$$

It turns out that due to the properties of a Gaussian distribution (see [10] and Appendix A in [11]),

the mean ($\overline{\boldsymbol{q}'}$) and covariance ($\mathrm{cov}(\boldsymbol{q}')$) of QoI at the test samples can be determined as:

$$\overline{\boldsymbol{q}'} = K(\boldsymbol{M}'_\mu, \boldsymbol{M}_\mu)[K(\boldsymbol{M}_\mu, \boldsymbol{M}_\mu) + \sigma_n^2 I]^{-1}\boldsymbol{q} \tag{1.3a}$$

and

$$\mathrm{cov}(\boldsymbol{q}') = K(\boldsymbol{M}'_\mu, \boldsymbol{M}'_\mu) - K(\boldsymbol{M}'_\mu, \boldsymbol{M}_\mu)[K(\boldsymbol{M}_\mu, \boldsymbol{M}_\mu) + \sigma_n^2 I]^{-1}K(\boldsymbol{M}_\mu, \boldsymbol{M}'_\mu), \tag{1.3b}$$

where $K(\boldsymbol{M}', \boldsymbol{M})$ is a covariance matrix whose elements are generated by evaluating the covariance function with all the possible combinations of the parameter vectors in the argument matrices. Furthermore, σ_n^2 denotes the variance of a Gaussian random variable responsible for modeling the noise (not knowing the training values exactly) in the system. In Gaussian Processes trained with the results of numerical simulations, this can be the stopping tolerance of the nonlinear or linear iteration throughout the solution process, for example.

In nuclear engineering, Gaussian Processes have been utilized for uncertainty quantification of thermal hydraulics simulations [12], dose mapping [13] and nuclear component degradation [14], to name a few.

1.2.2 Polynomial Chaos Expansion

The first use of Polynomial Chaos Expansion (PCE) for stochastic differential equations was presented in [15] in an intrusive setting and the corresponding non-intrusive methods have quickly followed [16]. The review below is based on [17] with the notation adjusted to the content of this paper. Again, for the sake of brevity, it is assumed that the quantity of interest is a scalar-valued function of the model parameters. It must be noted, however, that this procedure can be easily extended to vector-valued problems as well.

PCE is based on the polynomial expansion of the output (QoI) in the uncertain model parameters ($\boldsymbol{\mu}$):

$$\mathrm{QoI}(\boldsymbol{\mu}) \approx \sum_{j=0}^{n} c_j \mathcal{P}_j(\boldsymbol{\mu}), \tag{1.4}$$

where n is the number of polynomial terms used and c_j and $\mathcal{P}_j$ are the corresponding expansion coefficients and multi-dimensional polynomial functions. It can be shown that $\mathcal{P}_j$ can be written as the product of polynomials depending on single model parameters (μ_i) as [16]:

$$\mathcal{P}_j(\boldsymbol{\mu}) = \prod_{i=1}^{d} P_{i,j}(\mu_i). \tag{1.5}$$

The number n and the degree of the one-dimensional polynomials $P_{i,j}$ is, most of the times, selected using a maximum allowed degree over all the dimensions and constructing all the possible combinations from $P_{i,j}$ that satisfy this condition. The type of the polynomials typically depend on the assumed probability distribution of the model parameter. For the corresponding distribution-polynomial pairs, the reader is referred to [15, 16]. In this setting, the only remaining unknowns are the expansion coefficients. These can be determined using multiple approaches, like linear regression or spectral projection. In case of spectral projection, one needs to evaluate the following expression for each coefficient:

$$c_j = \frac{\int \mathrm{QoI}(\boldsymbol{\mu})\mathcal{P}_j(\boldsymbol{\mu})w(\boldsymbol{\mu})d\boldsymbol{\mu}}{\int \mathcal{P}_j^2(\boldsymbol{\mu})w(\boldsymbol{\mu})d\boldsymbol{\mu}}, \tag{1.6}$$

where $w(\boldsymbol{\mu})$ is a weighting function determined using the assumed distributions of the model parameters. The integrals can be evaluated using Monte Carlo methods or different quadrature rules, such as tensor grids or Smolyak sparse grids. When the expansion coefficients are available, every evaluation of the QoI simplifies into the evaluation of the polynomial approximate in Eq. (1.4) which is of low computational cost. Furthermore, another desirable feature of PCE is that the statistical moments and sensitivities of QoI can be computed analytically.

PCE based ROMs have been used in nuclear engineering for the uncertainty propagation of prompt fission neutron spectrum [18], criticality problems [19] and multiphysics problems [20], to name a few.

1.2.3 Interpolation using Sparse Grids

Interpolation on sparse grids using global polynomial functions have been introduced in [21]. Later, it has been extended to dimension-wise adaptive interpolation on sparse grids in [22] and to locally defined spline interpolation in [23]. The essence of the method is that the function of interest (QoI) is approximated by its interpolant in the parameter space:

$$\mathrm{QoI}(\mu_i) \approx \mathcal{I}^l(\mu_i) = \sum_{j=1}^{m_l} \mathrm{QoI}(\mu_{i,j}^l)\mathcal{P}_j^l(\mu_i), \tag{1.7}$$

where $\mathcal{I}^l$ denotes the interpolant at the hierarchical level l of the grid, while m_l denotes the number of interpolation points and $\mathcal{P}_j^l$ the associated polynomial basis functions, respectively. Note that first, only a one-dimensional problem is considered in one of the model parameters (μ_i). This will be extended when the multidimensional grids are introduced. The same notation is used as for the PCE in the previous subsection. However, in this method, the polynomials may have local support instead of the global support required for PCE. The only remaining question is where to anchor these basis points. When the number of model parameters, d, is low, one can just use a tensor product interpolant of different dimensions as

$$\mathrm{QoI}(\boldsymbol{\mu}) \approx \boldsymbol{\mathcal{I}}(\boldsymbol{\mu}) = \mathcal{I}^{l_1} \otimes \cdots \otimes \mathcal{I}^{l_d} = \sum_{j_i=1}^{m_{l_1}} \cdots \sum_{j_d=1}^{m_{l_d}} \mathrm{QoI}(\mu_{j_1}^{l_1}, ..., \mu_{j_d}^{l_d}) \left(\mathcal{P}_{j_1}^{l_1}(\mu_1) \otimes \cdots \otimes \mathcal{P}_{j_d}^{l_d}(\mu_d)\right), \tag{1.8}$$

where the interpolation points are also generated as the tensor product of the anchor points in each dimension. However, in high dimensional parameter spaces, the number of evaluations needed for a certain accuracy might be too large, therefore the use of sparse grids becomes necessary. These sparse grids are usually the linear combination of low-resolution tensor product grids with a certain rule for the summation. In case of the Smolyak sparse grid, the interpolant can be constructed as follows:

$$\mathrm{QoI}(\boldsymbol{\mu}) \approx \boldsymbol{\mathcal{I}}_q^{\mathrm{Smol}}(\boldsymbol{\mu}) = \sum_{q-d+1 \le |\boldsymbol{l}| \le q} (-1)^{q-|\boldsymbol{l}|} \binom{d-1}{q-|\boldsymbol{l}|} \left(\mathcal{I}^{l_1} \otimes \cdots \otimes \mathcal{I}^{l_d}\right), \tag{1.9}$$

where $|\boldsymbol{l}| = l_1 + ... + l_d$ and we assume that $q \geq d$ and q is the level of the sparse grid itself. This approach can be further improved with hierarchical adaptive sparse grids [23].

Interpolation using sparse grids is not commonly used within the nuclear engineering community, however applications can be noted in the treatment of few-group cross-sections [24] and uncertainty quantification of Molten Salt Reactors [25].

1.2.4 Dynamic Mode Decomposition

Lastly, Dynamic Mode Decomposition (DMD) [26] is a useful reduction tool for mainly linear and nonlinear time-dependent problems (hence the name) where the quantity of interest can be represented as a series of related data vectors. Let us denote this time-dependent vector of QoIs with $\boldsymbol{q}(t)$. Note, there is no parameter dependence involved in this case. The process starts with collecting N_s snapshots of $\boldsymbol{q}(t)$ by saving these vectors at (for now) equidistant times from t_0 to t_{N_s}. These snapshots are then organized into two snapshot matrices in the following manner:

$$\boldsymbol{X} = [\boldsymbol{q}(t_0)\ \boldsymbol{q}(t_1)\ \cdots\ \boldsymbol{q}(t_{N_s-1})] \quad \text{and} \quad \boldsymbol{X'} = [\boldsymbol{q}(t_1)\ \boldsymbol{q}(t_2)\ \cdots\ \boldsymbol{q}(t_{N_s})]. \tag{1.10}$$

As a next step, it is assumed that subsequent snapshots are connected by a linear operator, $\boldsymbol{A}$, therefore $\boldsymbol{X'}$ can be expressed in terms of $\boldsymbol{X}$ as

$$\boldsymbol{X'} = \boldsymbol{A}\boldsymbol{X}.$$

However, this $\boldsymbol{A}$ operator is unknown and computing a low-order approximation ($\boldsymbol{A}_r$) is the key in DMD. This can be computed by the following procedure:

$$\boldsymbol{A}_r = \boldsymbol{U}^T\boldsymbol{X'}\boldsymbol{W}\boldsymbol{\Sigma}^{-1}, \tag{1.11}$$

where $\boldsymbol{U}$, $\boldsymbol{W}$ and $\boldsymbol{\Sigma}$ are matrices obtained through the Singular Value Decomposition (SVD) of matrix $\boldsymbol{X}$. In many scenarios, matrix $\boldsymbol{X}$ can be reconstructed accurately with the first r basis vectors in $\boldsymbol{U}$, therefore it is enough to use the first r columns of $\boldsymbol{U}$, $\boldsymbol{\Sigma}$ and $\boldsymbol{W}$ to construct $\boldsymbol{A}_r$. Therefore, in most of the cases the size of $\boldsymbol{A_r}$ is less than the number of snapshots ($r \ll N_s$), resulting in savings in computation time. As a next step, $\boldsymbol{A_r}$ is diagonalized using its eigendecom-

position:

$$\boldsymbol{A}_r = \boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^T, \tag{1.12}$$

where $\boldsymbol{V}$ and $\boldsymbol{\Lambda}$ are matrices containing the eigenvectors and eigenvalues of $\boldsymbol{A}_r$. This operation can be extremely fast if the rank of $\boldsymbol{A}_r$ is small, that is, when only a few columns of $\boldsymbol{U}$ are kept. Using the computed eigenvalues and eigenvectors, the approximate and exact DMD modes ($\Psi_{DMD,i}$, $i = 1, ..., r$) can be computed as:

$$\boldsymbol{\Psi}_{DMD}^{\text{app}} = \boldsymbol{U}^r\boldsymbol{V} \quad \text{and} \quad \boldsymbol{\Psi}_{DMD}^{\text{ex}} = \boldsymbol{\Lambda}^{-1}X'\boldsymbol{W}\boldsymbol{\Sigma}^{-1}\boldsymbol{V}. \tag{1.13}$$

For more information about the difference in the two types of DMD modes, the reader is referred to [27]. These DMD modes can then be used to reconstruct the dynamics of the system QoI by the following expression:

$$\text{QoI}(x,t) \approx \sum_{i=1}^{r} a_i \Psi_{DMD,i}(x) e^{\ln(\lambda_i)\frac{t}{\Delta t}}, \tag{1.14}$$

where a_i coefficients can be computed by $\boldsymbol{a} = \boldsymbol{\Psi}_{DMD}^{-1}\boldsymbol{q}(t_0)$. Using this expression, the dynamics of the quantity of interest can be predicted or tracked back in time. This method has mostly been applied to fluid flow problems; however, recent studies has showed that it can be used for neutronics problems as well to study time-eigenvalues of a system, see for example in [28] and [29].

1.3 Intrusive Approaches

The intrusive methods are fundamentally different in a sense that the operators of the underlying mathematical model are needed for the construction of the ROMs. In this section, few of the most common intrusive techniques are reviewed using a linear time-invariant (LTI) problem of the following form:

$$\frac{d\boldsymbol{\theta}(\boldsymbol{t})}{dt} = \boldsymbol{A}\boldsymbol{\theta}(t) + \boldsymbol{b}u(t), \tag{1.15}$$

$$o(t) = \boldsymbol{c}^T\boldsymbol{\theta}(t), \tag{1.16}$$

where $\boldsymbol{\theta}(t) \in \mathbb{R}^N$ is a discretized state vector with N denoting the degrees of freedom. Furthermore, $u(t) \in \mathbb{R}$ is an input function, $o(t) \in \mathbb{R}$ is an output function, $\boldsymbol{A} \in \mathbb{R}^{N \times N}$ is a discretized linear operator, and $\boldsymbol{b} \in \mathbb{R}^N$ and $\boldsymbol{c} \in \mathbb{R}^N$ are the input and output distribution vectors. Lastly, we assume that $\boldsymbol{\theta}(t_0) = \mathbf{0}$ initial condition is used for simplicity. It is apparent that only single-input single-output cases are considered in the demonstration problem; however, the presented techniques can be generalized for multi-input and multi-output cases as well. A common feature of these methods that they employ a projection onto a carefully selected reduced subspace generated for the state vector. The projection approach and the approximation of the state vector depend strongly on the selected method.

It must be mentioned that the Reduced Basis method used in this work also belongs to the group of intrusive ROM techniques, however it is discussed in greater details later, in Chapter 2.

1.3.1 Krylov Subspace Methods

The theory for Krylov subspace methods for efficient reduced-order modeling was developed in the 1970s, but its application became widespread in the early 1990s [30], mainly for electrical circuit related problems. The following description of the essence of the method follows the one in [31] with the notation adjusted to Eq. 1.16. As a first step, the Laplace transform of system (1.16) is taken:

$$s\boldsymbol{\theta}(\boldsymbol{s}) = \boldsymbol{A}\boldsymbol{\theta}(s) + \boldsymbol{b}u(s), \tag{1.17}$$

$$o(s) = \boldsymbol{c}^T\boldsymbol{\theta}(s), \tag{1.18}$$

where we can introduce $\boldsymbol{X} = -(\boldsymbol{A} - s_0\boldsymbol{I})^{-1}$ and $\boldsymbol{r} = (\boldsymbol{A} + s_0\boldsymbol{I})^{-1}\boldsymbol{b}$ to compute the transfer function $H(s)$ (that connects $o(s)$ and $u(s)$ directly), around an expansion point s_0:

$$H(s) = \boldsymbol{c}^T(\boldsymbol{A} + s_0\boldsymbol{I} - s_0\boldsymbol{I} + s\boldsymbol{I})^{-1}\boldsymbol{b} = \boldsymbol{c}^T(\boldsymbol{I} - (s - s_0)\boldsymbol{X})^{-1}\boldsymbol{r}. \tag{1.19}$$

At this point, matrix $(\boldsymbol{I} - (s - s_0)\boldsymbol{X})^{-1}$ in the transfer function can be expanded into Neumann series to get the following form:

$$H(s) = \boldsymbol{c}^T(\boldsymbol{I} - (s - s_0)\boldsymbol{X})^{-1}\boldsymbol{r} = \boldsymbol{c}^T\boldsymbol{r} + (\boldsymbol{c}^T\boldsymbol{X}\boldsymbol{r})(s - s_0) + (\boldsymbol{c}^T\boldsymbol{X}^2\boldsymbol{r})(s - s_0)^2 + \dots, \quad (1.20)$$

where $(\boldsymbol{c}^T\boldsymbol{X}^i\boldsymbol{r})$ are called the moments about s_0. The reduced-order modeling is based on n-th Padé approximation [32] of $H(s)$, denoted here by $H_n(s)$. This approximation matches the moments in the expansion up to $2n$. There are different techniques to compute this approximation starting from the classical moment matching algorithm to the more advanced Padé via Lánczos (PVL) [33] method which is one of the most used methods today. PVL utilizes the fact that the moments in the expansion can be computed by taking products of the right and left Krylov spaces:

$$\mathcal{K}_n(\boldsymbol{X}, \boldsymbol{r}) = \text{span}\{\boldsymbol{r}, \boldsymbol{X}\boldsymbol{r}, ..., \boldsymbol{X}^{n-1}\boldsymbol{r}\} \quad \text{and} \quad \mathcal{K}_n(\boldsymbol{X}, \boldsymbol{r}) = \text{span}\{\boldsymbol{c}, \boldsymbol{X}^T\boldsymbol{c}, ..., (\boldsymbol{X}^T)^{n-1}\boldsymbol{c}\}. \quad (1.21)$$

Unfortunately, using the basis vectors above for the moment-matching algorithms result in ill-conditioned systems which puts a limitation on the maximum order of the Padé approximation. To circumvent this problem, the Lánczos process is used to create basis vectors that span the same Krylov spaces. The first n Lánczos basis vectors $\boldsymbol{v}_i$, $i = 1, ..., n$, for the right Krylov space and the corresponding basis vectors in the left Krylov space $\boldsymbol{w}_i$, can be used to transform matrix $\boldsymbol{X}$ into a $n \times n$ tridiagonal matrix, $\boldsymbol{T}_n$, as follows:

$$\boldsymbol{W}_n^T\boldsymbol{X}\boldsymbol{V}_n = \boldsymbol{D}_n\boldsymbol{T}_n, \quad (1.22)$$

where $\boldsymbol{D}_n$ is a diagonal matrix, and $\boldsymbol{W}$ and $\boldsymbol{V}$ matrices contain the left and right basis vectors, respectively. Using these basis vectors, one can not only compute the n-th Padé approximation, but can reduce the system in the time domain by approximating the time-dependent state vector as the linear combination of the right Lánczos vectors and projecting (left multiplying) system 1.16

using the left Lánczos vectors:

$$\begin{aligned}\boldsymbol{D}_n\frac{d\tilde{\boldsymbol{\theta}}}{dt} =&\boldsymbol{A}_r\tilde{\boldsymbol{\theta}}(t)+\boldsymbol{b}_r u(t),\\ o(t) =&\boldsymbol{c}_r^T\tilde{\boldsymbol{\theta}}(t),\end{aligned}$$

where $\boldsymbol{A}_r = \boldsymbol{W}_n^T\boldsymbol{A}\boldsymbol{V}_n$, $\boldsymbol{b}_r = \boldsymbol{W}_n^T\boldsymbol{b}$ and $\boldsymbol{c}_r = \left(\boldsymbol{c}\boldsymbol{V}_n^T\right)^T$. This system is smaller than the original and therefore obtaining the solution takes less effort. It must be mentioned, that obtaining the basis vectors using the Lanczos process is computationally efficient, since it is a recursive procedure that can be terminated at the required number of basis functions. However, parametric and nonlinear problems pose a considerable challenge for this technique. These have been investigated in [34] and [35], respectively.

1.3.2 Balanced Truncation

Balanced truncation for model order reduction was originally proposed in [36] and is still widely used in electric circuit modeling [37]. The overview of the method in this section follows the one in [37] with slight adjustments to the notation in Eq. (1.16). The idea behind this method is that the modes of the original system that are difficult to control through $u(t)$ while being unobservable through $o(t)$ can be truncated. Many variants have been developed, but the most common is the Lyapunov truncation which requires the generation of two Gram matrices: a controllability matrix, $\boldsymbol{W}_c$, and an observability matrix, $\boldsymbol{W}_o$. These can be obtained by solving two Lyapunov problems. For example (1.16), these equations are the following:

$$\boldsymbol{A}\boldsymbol{W}_c + \boldsymbol{W}_c\boldsymbol{A}^T + \boldsymbol{b}\boldsymbol{b}^T = 0, \tag{1.23}$$

$$\boldsymbol{A}^T\boldsymbol{W}_o + \boldsymbol{W}_o\boldsymbol{A} + \boldsymbol{c}^T\boldsymbol{c} = 0, \tag{1.24}$$

Using these matrices one can generate a transformation matrix, $\boldsymbol{T}$ and its inverse, $\boldsymbol{T}^{-1}$, which can transform both $\boldsymbol{W}_c$ and $\boldsymbol{W}_o$ into the same diagonal form $\boldsymbol{\Sigma}$ with $\boldsymbol{T}\boldsymbol{W}_c\boldsymbol{T}^T = \boldsymbol{T}^{-T}\boldsymbol{W}_o\boldsymbol{T}^{-1} = \boldsymbol{\Sigma}$. State space models where $\boldsymbol{W}_c$ and $\boldsymbol{W}_o$ are both diagonal matrices are called balanced in

control terminology, hence the name of the method. One can expand the state vector as the linear combination of the first r most important column vectors of the transformation matrix, $\boldsymbol{\theta} \approx \boldsymbol{T}_r\tilde{\boldsymbol{\theta}}$, where the importance of each vector can be determined by the magnitude of the corresponding diagonal element in $\boldsymbol{\Sigma}$, also called as Hankel eigenvalues. Using this reduced transformation matrix $\boldsymbol{T}_r$, the reduced-order model for the LTI can be expressed as:

$$\begin{aligned}\frac{d\tilde{\boldsymbol{\theta}}}{dt} =& \boldsymbol{A}_r\tilde{\boldsymbol{\theta}}(t) + \boldsymbol{b}_r u(t),\\ o(t) =& \boldsymbol{c}_r^T\tilde{\boldsymbol{\theta}}(t),\end{aligned}$$

where $\boldsymbol{A}_r = \boldsymbol{T}_r^{-1}\boldsymbol{A}\boldsymbol{T}_r$, $\boldsymbol{b}_r = \boldsymbol{T}_r^{-1}\boldsymbol{B}$ and $\boldsymbol{c}_r = \left(\boldsymbol{c}\boldsymbol{T}_r^T\right)^T$. This system is smaller than the original and therefore obtaining the solution takes less effort. It must be mentioned, however, that solving a Lyapunov equation has a $O(N^3)$ operation cost meaning that Lyapunov truncation is not practical with the current computational resources if $N > 10^5$. Furthermore, this method in the presented form is not capable of simulating parametric and nonlinear systems. These issues have been addressed in [38] and [39], respectively.

1.3.3 Proper Generalized Decomposition

For the review of Proper Generalized Decomposition (PGD) [40], we step back to the non-discretized form of the state vector $\boldsymbol{\theta}$. It is assumed that the solution function depends on a spatial coordinate and time so the corresponding continuous problem can be written as

$$\frac{\partial\theta(t,x)}{\partial t} = A\theta(t,x) + b(x)u(t), \tag{1.25}$$

where $\theta(t,x)$ is the solution field, A is a linear operator and $b(x)$ is a source field. In case of PGD, one approximates the multi-variant solution function with the sums of products of univariant functions as follows:

$$\theta(x,t) \approx \theta_N(x,t) = \sum_{i=1}^{N} X_i(x)T_i(t), \tag{1.26}$$

where the constituent functions can be computed in an iterative procedure. Let $\theta_n(x,t)$ denote the n-th iterate and assume an initial function for both $X_1(x)$ and $T_1(t)$. Using the recurrence in the summation, the solution at iteration step n can be expressed as the solution at the previous iteration step and an additional term as:

$$\theta_n(x,t) = \sum_{i=1}^{n-1} X_i(x)T_i(t) + X_n(x)T_n(t) = \theta_{n-1}(x,t) + X_n(x)T_n(t). \tag{1.27}$$

Plugging this approximation into Eq. (1.25) and taking the weak form using a test function $\theta^*(x,t)$ we obtain the following system:

$$\begin{aligned}\int_{\Upsilon\times[0,T]} \theta^*(x,t)\left(X_n(x)\frac{dT_n(t)}{dt} - AX_n(x)T_n(t)\right)d\Upsilon dt = \\ \int_{\Upsilon_x\times[0,T]} \theta^*(x,t)\left(\sum_{i=1}^{n-1}\left[X_i(x)\frac{dT_i(t)}{dt} - AX_i(x)T_i(t)\right] + b(x)u(t)\right)d\Upsilon dt,\end{aligned} \tag{1.28}$$

where $\theta^*(x,t)$ is chosen to be either $X_n^*(x)T_n(t)$ or $X_n(x)T_n^*(t)$ with the asterisk denoting the function which is kept unchanged for the given step of a fixed-point iteration. If the spatial test function is unchanged, the equation reduces to a one-dimensional nonlinear problem in time, while in the other scenario it reduces to a one-dimensional nonlinear problem in space. These problems can be discretized and solved repeatedly. The fixed-point iteration can be implemented by alternating the one-dimensional solves until the n-th components of the update converge. Therefore, this approach is often called the method of alternating directions. It is visible that PGD decouples a multi-dimensional solve into multiple one-dimensional solves meaning that the complexity of the system scales linearly with the number of dimensions instead of exponentially. In case of parametric cases, the only additional step is to extend approximation (1.26) with one one-dimensional function for each parameter. Furthermore, PGD can be utilized for nonlinear problems as well, for example see the multiphysics problems in [41]. The utilization of PGD for neutronics problems typically encountered in nuclear engineering has been explored in [42, 43, 44].

1.4 Molten Salt Reactors

In this section, the history of the Molten Salt Reactor (MSR) technology is briefly reviewed with special emphasis on two concepts, the Molten Salt Reactor Experiment (MSRE) and the Molten Salt Fast Reactor (MSFR), which are used for the numerical examples in this dissertation.

Probably the most distinguishing feature of MSRs is that the nuclear fuel is in liquid form, dissolved in a salt. Salts are typically based on fluorides or chlorides and include one or more of the following compounds: LiF, NaF, BeF_2, ZrF_4, KF, NaCl, $MgCl_2$. MSRs have highly promising features in terms of sustainability and safety. They can be designed to have strong negative-only reactivity feedback; they operate at atmospheric pressure; they allow for an online removal of gaseous fission products; and they give the possibility to drain the fuel salt in passively cooled and critically-safe tanks in case of emergency. Furthermore, the typically high boiling point of the molten salts allow high temperature and consequently high efficiency energy conversion.

The first MSR designs were developed during the Aircraft Reactor Experiment (ARE) in the 1950s at Oak Ridge National Laboratory (ORNL, USA) [45]. Following this experiment, a new, $8\ MW_{th}$ demonstrating reactor was designed at ORNL which was under operation between 1965 and 1969 [46]. This project was called the Molten Salt Reactor Experiment (MSRE) and the gathered data has been the baseline for MSR-related code-validation up until this day. Figure 1.1 shows the structure of the core of the MSRE. The molten salt fuel enters the core through a flow distributor, down-comer and lower plenum. The salt then enters the channels between the moderator stingers made of graphite and, assuming an operation at non-zero power, warms up due to the fission reactions within the fluid. The warm fluid then leaves the graphite channels and, through the upper plenum, the core as well. The reactor mainly operated with LiF-BeF_2-ZrF_4-UF_4 fuel with temperatures slightly above $900\ K$. The experience gathered during this program lead to the design of an industrial $1\ GW_e$, thermal spectrum reactor, the Molten Salt Breeder Reactor (MSBR). The MSBR project, however, was terminated in 1976 because of the sodium-cooled fast reactors seemed to be more favorable alternatives. The MSR research in the USA dwindled down to low-priority, low-funding efforts over the following decades. However, the safety requirements

in the nuclear sector have changed over these decades and MSRs have been selected as one of the six concepts for further investigation by the Generation-IV International Forum [47].

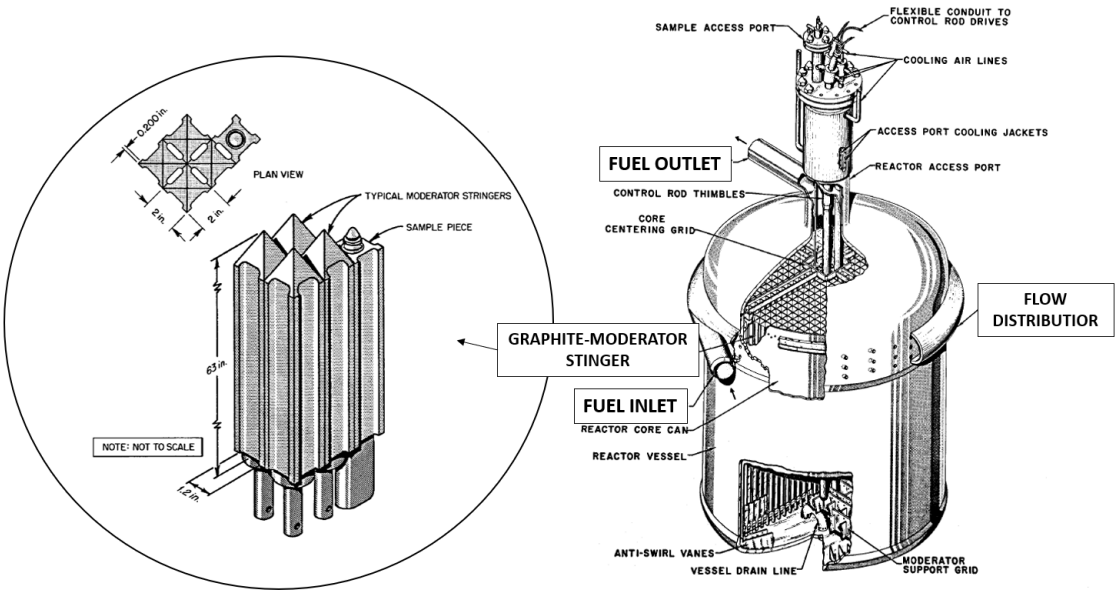


Figure 1.1: The reactor core (right) and the graphite stingers (left) used in the Molten Salt Reactor Experiment. Modified from [1].

Since the termination of the MSBR project, the international research efforts on MSRs split into two main directions. One direction kept the original, thermal reactor design of the MSBR with certain modifications. Good examples for such concepts are the Chinese Thorium Molten Salt Reactor (TMSR) [48] or the Japanese FUJI [49]. The other direction of the development chose a system with fast neutron spectrum. Examples of such designs include: the Russian MOlt en Salt Actinide Recycler and Transforming (MOSART) project [50], the Molten Chloride Fast Reactor (MCFR) developed by Terrapower (USA) [51] and the Molten Salt Fast Reactor (MSFR) concept, developed in the European EVOL (Evaluation and Viability of Liquid Fuel Fast Reactors) [52, 53] and then SAMOFAR (Safety Assessment of the Molten Salt Fast Reactor) [54] programs. Later is selected as a representative fast spectrum system for the examples in this work. The reactor core of the MSFR is depicted in Figure 1.2. It is visible that this design is simpler in a sense that

the reactor core cavity is entirely filled with the fuel salt, which is a common feature of MSRs with fast neutron spectrum. Advantages of this approach are the very hard neutron spectrum which can be used for breeding and transmutation, and the considerable reduction in pumping power due to the lack of flow resistances. On the other hand, this concept exhibits new challenges in the design process due to the complicated heat and mass-transfer phenomena [55]. The MSFR is supposed to operate with LiF-ThF_4-UF_4 salt with temperatures between 900 K and 1100 K.

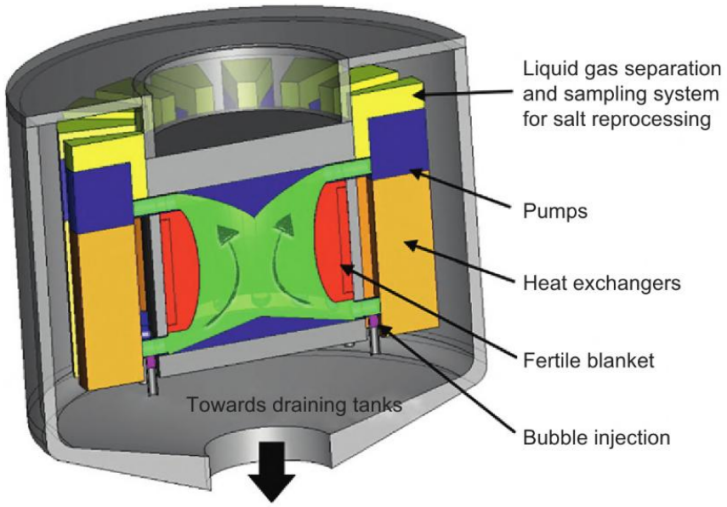


Figure 1.2: The reactor core of the Molten Salt Fast Reactor. Reprinted from [2].

1.5 The Aim and Structure of the Dissertation

The goal of this dissertation is to develop an intrusive Proper Orthogonal Decomposition aided Reduced Basis (POD-RB) ROM which can be used for multi-query tasks involving the simulations of liquid-fuel MSRs. Following the introduction, the content of the dissertation is structured as follows. Chapter 2 gives an overview of the chosen intrusive ROM technique with the help of a simple 1D example. Following this, Chapter 3 discusses the governing equations used to describe the behavior of MSRs together with the discretization techniques used to obtain the corresponding numerical models. Chapter 4 then introduces the derivation of the corresponding POD-RB ROMs, while Chapter 5 discusses the developed computational framework together with different methods

for the evaluation of the ROMs and a proposed approach that can be used for the uncertainty quantification and sensitivity analysis of the original system by the utilization of ROMs as emulators. Following these chapters, several numerical examples are provided. First, the ROM is applied to parameterized steady-state end transient zero-power simulations of the MSFR in Chapter 6. Then, in Chapter 7, the ROM is applied to parameterized steady-state simulation of the MSRE at nominal power level, with the assumption that the flow fields are fixed. This assumption ensures that the efficiency of the developed multiphysics ROMs for the handling the temperature-dependent cross sections can be assessed. The last example, discussed in Chapter 8, consists of parameterized, fully-coupled steady-state simulations of the MSFR with a 23-dimensional parameter space. Lastly, Chapter 9 summarizes the content of the dissertation with special emphasis on the novel contributions and possible approaches to improve on the presented work.

2. PROCESS OF THE INTRUSIVE POD-RB METHOD

The aim of reduced basis methods is to decrease the number of degrees of freedom in a simulation by using global basis functions chosen in a specific manner instead of local basis functions common in Finite Element Methods (FEM) or Finite Volume Methods (FVM). In the problems presented in this work, an intrusive Proper Orthogonal Decomposition aided Reduced Basis (POD-RB) technique is used to decrease the number of spatial degrees of freedom. The choice of the method relies on two assumptions:

i The intrusiveness of the method has the potential of decreasing the number of computationally expensive model evaluations needed for training the ROMs. This is based on the intuition that the intrusive process incorporates additional information about the differential operators of the full-order models.

ii Intrusive POD-RB ROMs have been proven to be efficient for parametric nonlinear problems, which is ideal for simulations of MSRs. For specific applications of POD-RB ROMs, see the literature review in Section 2.5 later in this chapter.

Moreover, this chapter covers the general process of creating intrusive POD-RB ROMs and serves as an outline and reference for the more detailed, MSR-specific sections later in this dissertation. Figure 2.1 summarizes the necessary steps in this process, while Sections 2.1-2.3 discuss each step in more detail with the help of a simple, diffusion-reaction problem on a one-dimensional domain. Even though this example may be simple, it can already showcase the crucial details one needs to account for when creating intrusive POD-RB ROMs. The results for the 1D problem have been generated using a custom program coded in `python`.

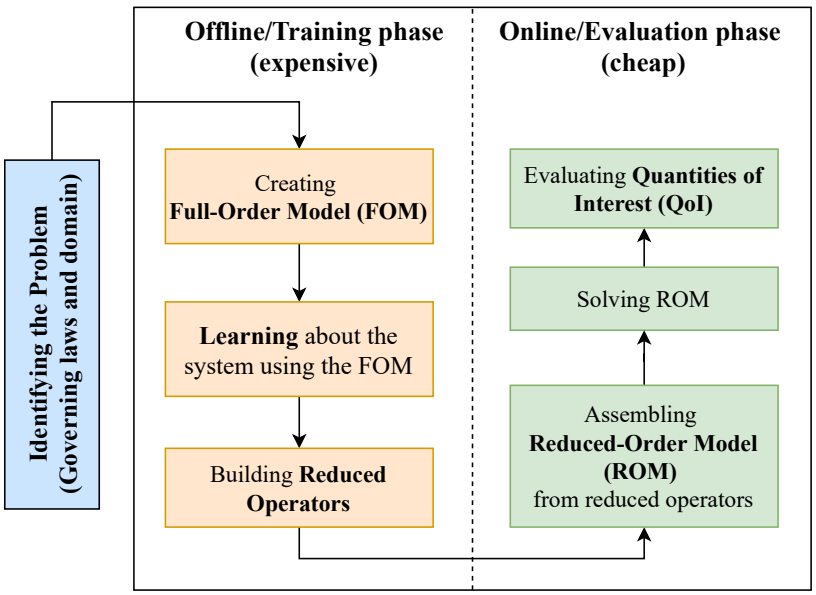


Figure 2.1: Flow chart of the Reduced-Order Modeling procedure carried out in this work.

2.1 Identifying the Problem

The process starts with the description of a mathematical model, i.e., a set of governing partial differential equations in a specific domain, together with the corresponding initial and boundary conditions. In this work, these governing equations describe the fluid flows, heat transfer, and neutronics phenomena in MSRs and are discussed in Section 3.1 in detail. For the time being, the general form of these equations is used which can be described as

$$\frac{d\theta(\boldsymbol{r},t;\boldsymbol{\mu})}{dt} + A(\boldsymbol{r},t;\boldsymbol{\mu})\theta(\boldsymbol{r},t;\boldsymbol{\mu}) + F(\theta(\boldsymbol{r},t;\boldsymbol{\mu}),\boldsymbol{r},t;\boldsymbol{\mu}) = S(\boldsymbol{r},t;\boldsymbol{\mu}), \quad \boldsymbol{r} \in \Upsilon, \tag{2.1}$$

where $\theta(\boldsymbol{r},t)$ is a general solution field that can be either scalar- (temperature, pressure) or vector-valued (velocity). Furthermore, A denotes a general liner operator, $F()$ a general, nonlinear function and S a general, external source term. All of these terms can depend on time, t, spatial position, $\boldsymbol{r} = [x, y, z]^T$, and $\boldsymbol{\mu}$, a vector containing model parameters (e.g., material or geometrical properties). The domain of interest and its boundary are denoted by Υ and Γ, respectively. Lastly, the governing law is supplemented with adequate initial and boundary conditions which can be

case-specific and, therefore, are not discussed in detail here.

Even though at certain parts of this chapter we utilize this general notation, the main steps of the procedure are presented for a simplified one-dimensional diffusion-reaction problem as well. This example has been specifically chosen to show the typical challenges and the corresponding solutions encountered in the generation of intrusive POD-RB ROMs, without the additional complexity associated with higher-dimensional problems. The example reads as:

$$\begin{cases} -\frac{\partial}{\partial x}\left(D(x)\frac{\partial \phi(x)}{\partial x}\right) + \Sigma(x)\phi(x) = S_{\text{ext}}(x) + F(\phi(x)), & x \in [0,1], \\ \phi(x) = 0, & x = 0,\ x = 1 \end{cases} \tag{2.2}$$

where $\phi(x)$ denotes the solution field, while the diffusion coefficient, $D(x)$, reaction coefficient, $\Sigma(x)$, and external source term, $S_{\text{ext}}(x)$, are given by:

$$D(x) = \begin{cases} D_1, & x < 0.5, \\ D_2, & x > 0.5 \end{cases} \qquad \Sigma(x) = \begin{cases} \Sigma_1, & x < 0.5, \\ \Sigma_2, & x > 0.5 \end{cases} \qquad S_{\text{ext}}(x) = \begin{cases} S_1, & x < 0.5, \\ S_2, & x > 0.5, \end{cases}$$

The nonlinear source term in this specific case is as follows:

$$F(\phi(x)) = e^{-\phi^2(x)}.$$

This term introduces an additional source which is considerable when the solution is a small number, but rapidly decreases as the solution increases. It is visible that the terms in Eq. (2.2) depend on model parameters D_1, D_2, Σ_1, Σ_2, S_1 ,S_2. These parameters are considered to be uncertain in this example with distributions shown in Table 2.1. Therefore, in this scenario, the parameter vector can be defined as: $\boldsymbol{\mu} = [D_1, D_2, \Sigma_1, \Sigma_2, S_1, S_2]^T$, and the dependence of the solution on these parameters will be explicitly noted as $\phi(x; \boldsymbol{\mu})$. In this setting, $\mathcal{U}(a, b)$ denotes a uniform distribution with a lower bound of a and an upper bound of b.

Table 2.1: The distributions of the parameters used for the 1D example.

Parameter	Distribution	Parameter	Distribution
D_1	$\mathcal{U}(1.0, 9.0)$	D_2	$\mathcal{U}(1.0, 9.0)$
Σ_1	$\mathcal{U}(0.2, 1.8)$	Σ_2	$\mathcal{U}(0.2, 1.8)$
S_1	$\mathcal{U}(0.2, 1.8)$	S_2	$\mathcal{U}(0.2, 1.8)$

2.2 Offline or Training Phase

Once the problem has been clearly identified, we then enter the offline, or training phase, which consists of three major steps. First, the selected mathematical model is discretized, resulting in a Full-Order Model (FOM). Then, the Full-Order model is exercised to collect data about the system, which is used to build reduced subspaces. The basis functions of the reduced subspaces are then used to create reduced operators for the ROMs. These steps are discussed in detail in the following subsections.

2.2.1 Creating the Full-Order Model (FOM)

This process entails the introduction of discretization schemes in space, time, and for neutronics problems, energy and potentially angle as well. The discretization schemes used for the MSR-specific problems in this work are discussed in Section 3.2 in detail; however, it is important to establish at this point that, for spatial discretization, we use the cell-centered Finite Volume Method (FVM) [56] for every problem in this work. This entails that the spatial domain is split into N non-overlapping cells and the solution is approximated to be constant over each cell. Let us consider the general problem described in Eq. (2.1) first. After the spatial discretization, it yields the following nonlinear system of Ordinary Differential Equations (ODEs):

$$\boldsymbol{M}_{\Upsilon}\frac{d\boldsymbol{\theta}(t;\boldsymbol{\mu})}{dt} + \boldsymbol{A}(t;\boldsymbol{\mu})\boldsymbol{\theta}(t;\boldsymbol{\mu}) + \boldsymbol{F}(\boldsymbol{\theta}(t;\boldsymbol{\mu}), t;\boldsymbol{\mu}) = \boldsymbol{S}(t;\boldsymbol{\mu}), \tag{2.3}$$

where $\boldsymbol{\theta}(t;\boldsymbol{\mu}) \in \mathbb{R}^N$ is the spatially discretized solution vector, $\boldsymbol{M}_\Upsilon \in \mathbb{R}^{N\times N}$ is a mass-matrix, $\boldsymbol{A}(t;\boldsymbol{\mu}) \in \mathbb{R}^{N\times N}$ is the discretized linear operator, $\boldsymbol{F}(\boldsymbol{\theta}(t;\boldsymbol{\mu}),t;\boldsymbol{\mu}) \in \mathbb{R}^N$ is the discretized non-linear function and $\boldsymbol{S}(t;\boldsymbol{\mu}) \in \mathbb{R}^N$ is the discretized external source term.

For the 1D problem outlined in Eq. (2.2), $x \in [0,1]$ domain has been split to $N = 200$, non-overlapping cells and the problem has been discretized using cell-centered FVM with a linear interpolation, while the derivative of $\phi(x)$ is approximated at the cell-boundaries with a central differencing scheme. Using the notation already established for the general problem, the resulting discretized system can be expressed as:

$$\boldsymbol{A}(\boldsymbol{\mu})\boldsymbol{\phi}(\boldsymbol{\mu}) - \boldsymbol{F}(\boldsymbol{\phi}(\boldsymbol{\mu})) = \boldsymbol{S}(\boldsymbol{\mu})\,, \tag{2.4}$$

where $\boldsymbol{A}(\boldsymbol{\mu}) \in \mathbb{R}^{200\times 200}$ is the discretized linear operator that consists of the diffusion and reaction terms, $\boldsymbol{F}(\boldsymbol{\phi}(\boldsymbol{\mu})) \in \mathbb{R}^{200}$ is the discretized nonlinear function , $\boldsymbol{Q}(\boldsymbol{\mu}) \in \mathbb{R}^{200}$ is the discretized source term while $\boldsymbol{\phi}(\boldsymbol{\mu}) \in \mathbb{R}^{200}$ is the discretized solution vector.

2.2.2 Definition of Inner Products Used

Before diving into the details of the generation of reduced subspaces and operators, the basic notation for inner products needs to be established. In this work, the volumetric inner product of vector fields $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$ or scalar fields v_1 and v_2 is expressed as

$$\langle \boldsymbol{v}_1, \boldsymbol{v}_2 \rangle_\Upsilon = \int_\Upsilon \boldsymbol{v}_1 \cdot \boldsymbol{v}_2 \, d\Upsilon \approx \sum_{k=1}^{N} \boldsymbol{v}_{1,k} \cdot \boldsymbol{v}_{2,k} V_k \tag{2.5a}$$

$$\langle v_1, v_2 \rangle_\Upsilon = \int_\Upsilon v_1 v_2 \, d\Upsilon \approx \sum_{k=1}^{N} v_{1,k} v_{2,k} V_k \tag{2.5b}$$

where the solution fields in the finite volume approximation are cell-wise constant in cell k with volume V_k. Similarly, the boundary inner products can be expressed as

$$\langle \boldsymbol{v}_1, \boldsymbol{v}_2 \rangle_\Gamma = \int_\Gamma \boldsymbol{v}_1 \cdot \boldsymbol{v}_2 \, d\Gamma \approx \sum_{f=1}^{N_{bound}} \boldsymbol{v}_{1,f} \cdot \boldsymbol{v}_{2,f} |\boldsymbol{S}_f|, \tag{2.6a}$$

$$\langle v_1, v_2 \rangle_\Gamma = \int_\Gamma v_1 v_2 \, d\Gamma \approx \sum_{f=1}^{N_{bound}} v_{1,f} v_{2,f} |\boldsymbol{S}_f|, \tag{2.6b}$$

where $|\boldsymbol{S}_f|$ denotes the area of face f, with $f = 1, ..., N_{bound}$. Again, the fact that, in the finite volume approximation, the value of the variable is constant on the face was used.

2.2.3 Learning About the System

In reduced basis methods, the solution fields are approximated using basis functions with global support on the specific spatial domain. Let us take the discretized solution field $\boldsymbol{\theta}$ from the general formulation for example. The reduced basis approximation of $\boldsymbol{\theta}$ can be expressed as:

$$\boldsymbol{\theta}(t; \boldsymbol{\mu}) \approx \tilde{\boldsymbol{\theta}}(t; \boldsymbol{\mu}) = \sum_{i=1}^{r_\theta} \boldsymbol{\psi}_i^\theta c_i^\theta(t; \boldsymbol{\mu}) = \boldsymbol{\Psi}^\theta \boldsymbol{c}^\theta, \tag{2.7}$$

where $\boldsymbol{\Psi}^\theta$ is a matrix containing space-dependent $\boldsymbol{\psi}_i^\theta$ basis vectors of a reduced subspace and $\boldsymbol{c}^\theta$ is a vector which contains the time- and parameter-dependent c_i^θ coordinates of the approximate solution $\tilde{\boldsymbol{\theta}}$ in this subspace. It is important to note that each $\boldsymbol{\psi}_i^\theta$ basis vector corresponds to a $\psi_i^\theta(\boldsymbol{r})$ basis function which is the functional expression of the finite volume solution. Furthermore, r_θ is the rank of the subspace of $\boldsymbol{\theta}$. To obtain this subspace, one needs to survey the time- and parameter-dependent manifold of $\boldsymbol{\theta}$ which is considered to be a learning process. In case of POD-RB ROMs, this process can be separated into two steps: (1) collection of data and (2) the extraction of the basis functions (subspaces) for the fields of interest.

2.2.3.1 Collecting Data

In this step, the FOM is used to collect information about the dominant characteristics of the system. This involves exercising the FOM with different samples of the model parameter vector $\boldsymbol{\mu}_i$ ($i = 1, ..., N_\mu$), and building a data base using the corresponding solution fields. In practice, this means that the discretized solution vector is saved at N_τ time-instances into a snapshot matrix for each parameter sample. Let us consider the general equation described in Eq. (2.3) first. In this

case, the solution vector $\boldsymbol{\theta}$ is saved in the following snapshot matrix:

$$\boldsymbol{R}_\theta = [\boldsymbol{\theta}(\boldsymbol{\mu}_1, t_1), \ldots, \boldsymbol{\theta}(\boldsymbol{\mu}_{N_\mu}, t_{N_\tau})] \in \mathbb{R}^{N \times N_S}, \tag{2.8}$$

where $N_s = N_\mu \times N_\tau$ is the total number of snapshots collected throughout the training phase. The types of solution fields gathered for the MSR related problems in this work are summarized in Section 4.1.

For our one-dimensional example, two fields have been saved throughout the training phase: the solution $\boldsymbol{\phi}$ and the nonlinear source term $\boldsymbol{F}$. The latter is necessary for the efficient handling of nonlinear terms at reduced-order level; this will become clear in Section 2.2.4. Let

$$\boldsymbol{R}_\phi = [\boldsymbol{\phi}(\boldsymbol{\mu}_1), \ldots, \boldsymbol{\phi}(\boldsymbol{\mu}_{N_s})] \in \mathbb{R}^{200 \times N_s} \qquad \text{and}$$
$$\boldsymbol{R}_F = [\boldsymbol{F}(\boldsymbol{\phi}(\boldsymbol{\mu}_1)), \ldots, \boldsymbol{F}(\boldsymbol{\phi}(\boldsymbol{\mu}_{N_s}))] \in \mathbb{R}^{200 \times N_s}$$

denote the snapshot matrices which contain the saved solution vectors $\boldsymbol{\phi}$ and nonlinear source vectors $\boldsymbol{F}$, respectively. Since this example involves a steady-state problem, the number of snapshots, N_s, is equal to the number of parameter vector samples, N_μ. Altogether $N_s = 30$ snapshots have been captured by creating samples of $\boldsymbol{\mu}$ using Latin Hypercube Sampling (LHS) and executing the FOM with each sample. The first 7 snapshots for both $\boldsymbol{\phi}$ and $\boldsymbol{F}$ are presented in Figure 2.2.

2.2.3.2 Extracting the Basis Vectors of the Reduced Subspaces

The general idea behind obtaining an adequate subspace is to use the information already available in the collected snapshots. This assumes that the snapshots are representative of the full solution manifold, i.e., the fundamental characteristics of the latter are captured in the snapshots. It is important to note that the snapshot matrices may contain redundant information if the stored fields are similar. A good example would be one snapshot being equivalent to another one times a constant. Obviously, in this case the two snapshots contain the same information about the shape of the solution vector and the only variable describing the solution is the mentioned constant alone.

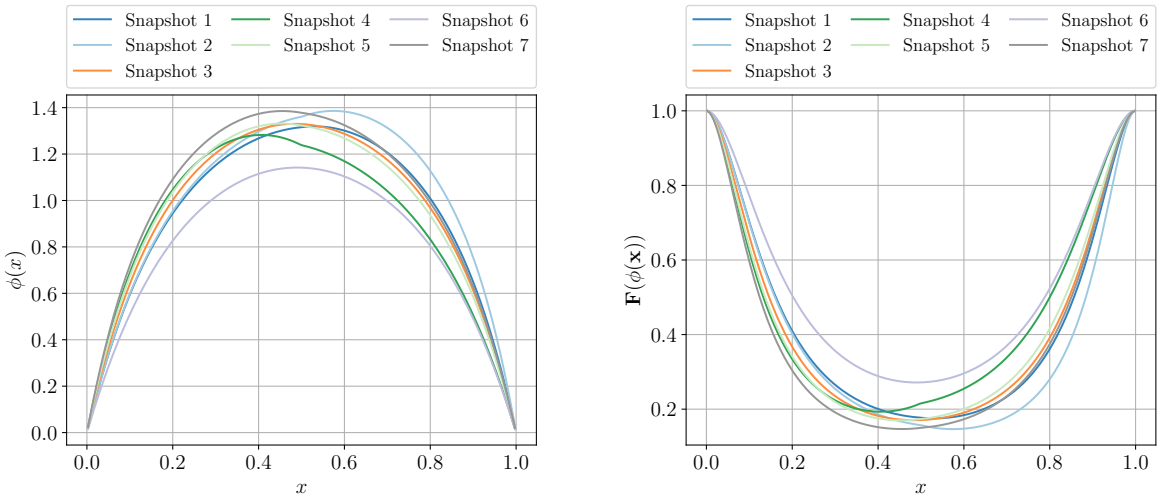


Figure 2.2: First seven snapshots of the solution vector $\boldsymbol{\phi}$ (left) and the nonlinear function $\boldsymbol{F}(\boldsymbol{\phi})$ (right).

To filter out such redundant information, Proper Orthogonal Decomposition (POD) [57, 58, 59] is utilized in this work. This involves the discovery of a reduced set of orthonormal basis functions that span a subspace of fixed rank that is closest to the snapshots in a global L^2 sense. We use these optimal basis functions to build our reduced subspaces for the solution vectors. Using the general solution θ as an example, this optimization problem can be formulated as:

$$\text{span}\{\psi_i^\theta\} = \underset{\text{span}\{\psi_i^\theta\}}{\arg\min} \sum_{s=1}^{N_s} \left\| \theta_s - \sum_{i=1}^{r_\theta} \left\langle \psi_i^\theta, \theta_s \right\rangle_\Upsilon \psi_i^\theta \right\|_{L^2}^2, \quad \text{with} \quad \left\langle \psi_i^\theta, \psi_j^\theta \right\rangle_\Upsilon = \delta_{i,j}. \tag{2.9}$$

The basis functions ψ_i^θ of this subspace $\text{span}\{\psi_i^\theta\}$ are often called POD modes and the two expressions will be used interchangeably hereafter. The solution of optimization problem (2.9) can be obtained using the generalized Singular Value Decomposition (SVD) of the snapshot matrix. Indeed, it can be proved, that the orthonormal basis vectors spanning a space of rank r_θ closest (in an L^2 sense) to the columns of the snapshot matrix are the first r_θ left singular vectors of the constrained SVD [60, 61]:

$$\boldsymbol{R}_\theta = \boldsymbol{\Psi}^\theta \boldsymbol{\Delta}^\theta \boldsymbol{V}^\theta, \tag{2.10}$$

where $\boldsymbol{\Psi}^{\theta}$ is the left singular matrix and is enforced to be unitary with respect to a mass matrix $\boldsymbol{M}_{\Upsilon}$ ($\boldsymbol{\Psi}^{\theta,T}\boldsymbol{M}_{\Upsilon}\boldsymbol{\Psi}^{\theta} = \boldsymbol{I}$). For more details on the constrained SVD, the interested reader is referred to [60]. Furthermore, $\boldsymbol{\Delta}^{\theta}$ denotes the singular value matrix and $\boldsymbol{V}^{\theta}$ is the unitary right singular value matrix. In finite volume methods $\boldsymbol{M}_{\Upsilon}$ is a diagonal matrix which contains the volumes of each cell in the mesh, therefore computing it is relatively inexpensive. Depending on the dimensions of $\boldsymbol{R}_{\theta}$, obtaining the left singular vectors might be costly in terms of computation time. In this work, it is assumed that the number of snapshots is considerably less than the spatial degrees of freedom ($N_s \ll N$), therefore the following process has been used to compute the POD modes (left singular vectors):

1. Using the snapshots in $\boldsymbol{R}_{\theta}$, a correlation matrix is computed with its elements defined as:

$$\boldsymbol{\mathcal{C}}^{\theta}_{i,j} = \langle \theta_i, \theta_j \rangle_{\Upsilon}, \quad i,j = 1,..,N_s, \tag{2.11}$$

   note, that this is equivalent to the following matrix-matrix product formulation:

$$\boldsymbol{\mathcal{C}}^{\theta} = \boldsymbol{R}^{T}_{\theta}\boldsymbol{M}_{\Upsilon}\boldsymbol{R}_{\theta}. \tag{2.12}$$

2. The eigendecomposition of this correlation matrix is computed as

$$\boldsymbol{\mathcal{C}}^{\theta} = \boldsymbol{V}^{\theta}\boldsymbol{W}^{\theta}\boldsymbol{V}^{\theta,T}, \tag{2.13}$$

   where $\boldsymbol{V}^{\theta}$ matrix contains the eigenvectors while $\boldsymbol{W}^{\theta}$ the eigenvalues (in descending order) of the correlation matrix. Since the size of the correlation matrix is $N_s \times N_s$, obtaining this decomposition is relatively cheap. Furthermore, it must be noted that the the singular values (σ_i) in $\boldsymbol{\Delta}^{\theta}$ are related to the eigenvalues (ω^{θ}_i) of the correlation matrix $\boldsymbol{W}^{\theta}$ through $\omega^{\theta}_i = (\sigma^{\theta}_i)^2$.

3. Using the eigenvalue and eigenvector matrices and the snapshots themselves, the basis vec-

tors ($\boldsymbol{\psi}_i^{\theta}$) of the reduced subspace of $\boldsymbol{\theta}$ can be reconstructed as

$$\boldsymbol{\psi}_i^{\theta} = \frac{1}{\sqrt{\omega_i^{\theta}}} \sum_{k=0}^{N_s} \theta_k \boldsymbol{V}_{i,k}^{\theta} \,. \tag{2.14}$$

Based on [61], the magnitudes of the singular/eigenvalues are good indicators of the importance of the corresponding basis functions for reconstructing the snapshots. Oftentimes, the number of relevant modes to keep is much smaller than the number of snapshots, $r_\theta \ll N_s$. In many cases r_θ can be determined using the eigenvalues as:

$$r_\theta = \underset{1 \leq r \leq N_s}{\arg\min} \left(\frac{\sum_{k=1}^{r} \omega_k^{\theta}}{\sum_{k=1}^{N_s} \omega_k^{\theta}} > 1 - \tau \right) , \tag{2.15}$$

where τ is an allowed relative error indicator. In the POD-ROM community $(1 - \tau)$ is also called the energy retention limit. It must be mentioned, however, that the error controlled by the energy retention limit is not the same as the error between the solution of the full-order model and the approximate solution from the reduced-order-model ($||\theta - \tilde{\theta}||$). The former only controls the error in the reconstruction of the snapshots, while the latter has additional components like the adequacy of the snapshots to represent the solution manifold. For instance, if the sampling of the parameter space is poor, the snapshots may represent only the solution manifold at parts of the parameter space. Consequently, the reduced-basis approximation may yield erroneous results at segments of the parameter space not represented in the snapshots. This error can be controlled by proper parameter sampling and snapshot taking frequency in case of transient simulations.

In case of the one-dimensional example, two subspaces are extracted from the snapshots: one for the solution vector $\boldsymbol{\phi}$ and another for the nonlinear function $\boldsymbol{F}$. The normalized eigenvalues of the correlation matrices built from the corresponding snapshots are presented for both fields in Figure 2.3. The normalization in this case was carried out using the sum of all eigenvalues, as shown in Eq. (2.15).

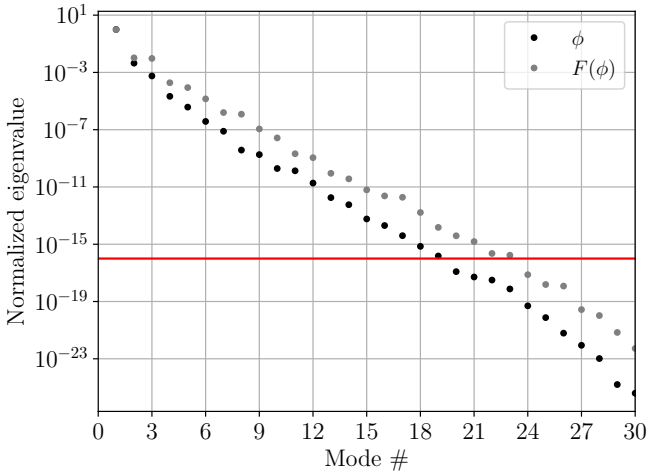


Figure 2.3: Normalized scree plot of the eigenvalues of the correlation matrices.

The red line in Figure 2.3 indicates the allowed error (τ) which would help to choose suitable dimensions for the subspaces. It is visible that $\boldsymbol{\phi}$ would need 20 basis functions to keep the reconstruction error in the snapshots below $\tau = 10^{-16}$, while the same for $\boldsymbol{F}$ is 24. Furthermore, the first 6 basis functions of the reduced subspaces for both $\boldsymbol{\phi}$ and $\boldsymbol{F}$ are presented in Figure 2.4.

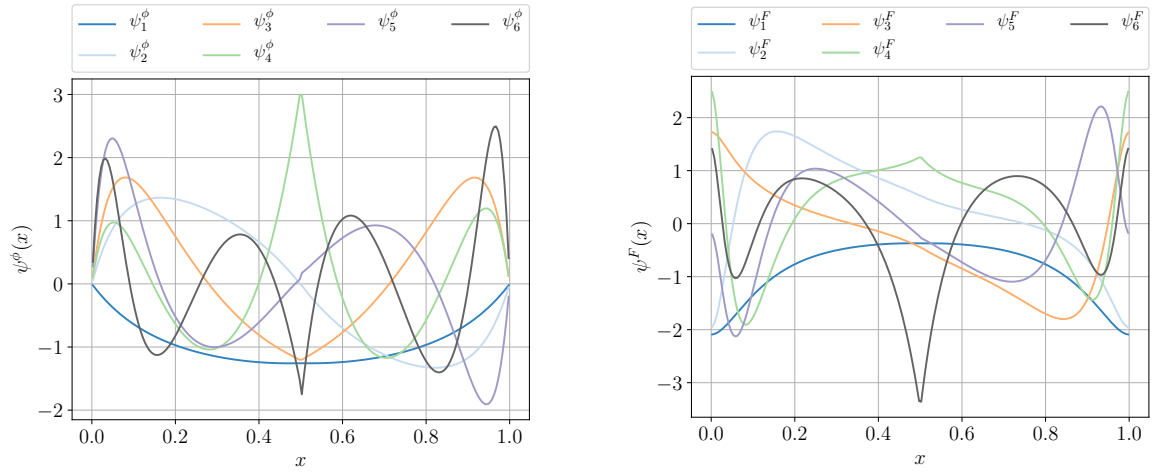


Figure 2.4: First six POD modes of the solution vector $\boldsymbol{\phi}$ (left) and the nonlinear function $\boldsymbol{F}(\boldsymbol{\phi})$ (right).

It is visible that due to the fact that the basis functions are just linear combinations of the snapshots, ψ^{ϕ} also respect the homogeneous Dirichlet boundary conditions.

2.2.4 Building Reduced Operators

The last step in the training procedure is the construction of reduced operators using the FOM and the available global basis functions. The general, spatially discretized formulation in Eq. (2.3) is considered first. Again, it is assumed that the solution can be well approximated by a vector from the subspace spanned by the global basis vectors:

$$\boldsymbol{\theta}(\boldsymbol{\mu}, t) \approx \tilde{\boldsymbol{\theta}}(\boldsymbol{\mu}, t) = \sum_{i=0}^{r_\theta} \boldsymbol{\psi}_i^\theta c_i^\theta(\boldsymbol{\mu}, t), = \boldsymbol{\Psi}^\theta \boldsymbol{c}^\theta(\boldsymbol{\mu}, t) \tag{2.16}$$

Using this approximation in Eq. (2.3) and a Galerkin projection (left multiplication by $\boldsymbol{\Psi}^{\theta,T}$ in this setting), the following equation is obtained:

$$\frac{\partial \boldsymbol{c}^\theta(\boldsymbol{\mu}, t)}{\partial t} + \boldsymbol{\Psi}^{\theta,T} \boldsymbol{A}(\boldsymbol{\mu}) \boldsymbol{\Psi}^\theta \boldsymbol{c}^\theta(\boldsymbol{\mu}, t) + \boldsymbol{\Psi}^{\theta,T} \boldsymbol{F}(\boldsymbol{\Psi}^\theta \boldsymbol{c}^\theta(\boldsymbol{\mu}, t), \boldsymbol{\mu}) = \boldsymbol{\Psi}^{\theta,T} \boldsymbol{S}\,, \tag{2.17}$$

where $\boldsymbol{\Psi}^{\theta,T} \boldsymbol{M}_\Upsilon \boldsymbol{\Psi}^\theta = \boldsymbol{I}$ has been used. This can be further simplified by introducing $\boldsymbol{A}^r(\boldsymbol{\mu}) = \boldsymbol{\Psi}^{\theta,T} \boldsymbol{A}(\boldsymbol{\mu}) \boldsymbol{\Psi}^\theta \in \mathbb{R}^{r_\theta \times r_\theta}$ and $\boldsymbol{S}^r = \boldsymbol{\Psi}^{\theta,T} \boldsymbol{S} \in \mathbb{R}^{r_\theta}$ reduced operator and source terms:

$$\frac{\partial \boldsymbol{c}^\theta(\boldsymbol{\mu}, t)}{\partial t} + \boldsymbol{A}^r(\boldsymbol{\mu}) \boldsymbol{c}^\theta(\boldsymbol{\mu}, t) + \boldsymbol{\Psi}^{\theta,T} \boldsymbol{F}(\boldsymbol{\Psi}^\theta \boldsymbol{c}^\theta(\boldsymbol{\mu}, t), \boldsymbol{\mu}) = \boldsymbol{S}^r\,. \tag{2.18}$$

It is visible that at this point the only unknowns in the system are the elements of $\boldsymbol{c}^\theta(\boldsymbol{\mu}, t)$, which means that the number of spatial unknowns is reduced from N to $r_\theta \ll N$. Despite this, solving this equation might not necessarily be faster than solving Eq. (2.3) because of two reasons:

i If operator $\boldsymbol{A}$ and source term $\boldsymbol{S}$ have a general dependence on time and model parameters, the corresponding reduced terms have to be created from the full-order ones at every time step or new parameter value. This operation scales with $2 \times r^\theta \times N$ which might result in negligible time-savings at reduced-level.

ii The nonlinear term has to be evaluated at each nonlinear iteration. This operation also scales with N meaning that the time required to solve the ROM would still scale with the size of the original system, which is not optimal.

The techniques mostly used in the ROM community to tackle these problems are covered in the following subsections.

2.2.4.1 Affine Decomposition of the Operators

To be able to create efficient POD-RB ROMs, it is important to ensure that no full-order operation (scaling with N) is performed during the evaluation phase. For the linear operators and source vector, this is satisfied if these terms in the FOM have an affine dependence on time and model parameters. In other words, if they can be expressed as the sum of products of parameter- and time-independent constituent operators and source terms ($\boldsymbol{A}^{(i)}$ and $\boldsymbol{S}^{(i)}$) and parameter- and time-dependent scalar functions ($f_A^{(i)}(t;\boldsymbol{\mu})$ and $f_S^{(i)}(t;\boldsymbol{\mu})$) as

$$\boldsymbol{A}(t;\boldsymbol{\mu}) = \sum_i f_A^{(i)}(t;\boldsymbol{\mu})\boldsymbol{A}^{(i)} \quad \text{and} \quad \boldsymbol{S}(t;\boldsymbol{\mu}) = \sum_i f_S^{(i)}(t;\boldsymbol{\mu})\boldsymbol{S}^{(i)}. \tag{2.19}$$

In such cases the constituent operators can be reduced one-by-one and the complete reduced operators can be computed by multiplying and summing small dense matrices:

$$\boldsymbol{A}^r(t;\boldsymbol{\mu}) = \boldsymbol{\Psi}^{\theta,T}\boldsymbol{A}(t;\boldsymbol{\mu})\boldsymbol{\Psi}^{\theta} = \sum_i f_A^{(i)}(t;\boldsymbol{\mu})\boldsymbol{\Psi}^{\theta,T}\boldsymbol{A}^{(i)}\boldsymbol{\Psi}^{\theta} = \sum_i f_A^{(i)}(t;\boldsymbol{\mu})\boldsymbol{A}^{(i),r} \quad \text{and}$$
$$\boldsymbol{S}^r(t;\boldsymbol{\mu}) = \boldsymbol{\Psi}^{\theta,T}\boldsymbol{S}(t;\boldsymbol{\mu}) = \sum_i f_S^{(i)}(t;\boldsymbol{\mu})\boldsymbol{\Psi}^{\theta,T}\boldsymbol{S}^{(i)} = \sum_i f_S^{(i)}(t;\boldsymbol{\mu})\boldsymbol{S}^{(i),r}.$$

This means that the assembly of such terms at reduced-order level scales with the rank of the reduced system, r^{θ}, and no full-order operations are needed.

Using the one-dimensional problem as an example, the linear operator $\boldsymbol{A}(\boldsymbol{\mu})$ has the following affine decomposition:

$$\boldsymbol{A}(\boldsymbol{\mu}) = -D_1\boldsymbol{K}_1 - D_2\boldsymbol{K}_2 + \Sigma_1\boldsymbol{M}_1 + \Sigma_2\boldsymbol{M}_2, \tag{2.20}$$

where D_1, D_2, Σ_1 and Σ_2 are the diffusion and reaction coefficients in material regions 1 and 2, while $\boldsymbol{K}_1$, $\boldsymbol{K}_2$, $\boldsymbol{M}_1$ and $\boldsymbol{M}_2$ are the corresponding region-wise stiffness and mass matrices computed by discretizing (using a weak form):

$$K_z(x) = \frac{\partial}{\partial x}\delta_z(x)\frac{\partial}{\partial x} \quad \text{and} \quad M_z(x) = \delta_z(x),$$

where $\delta_z(x)$ denotes a characteristic function which returns a value of 1 if x is in material zone z and 0 otherwise. These region-wise matrices can be then reduced using the basis function for $\boldsymbol{\phi}$ as:

$$\boldsymbol{K}_z^r = \boldsymbol{\Psi}^{\phi,T}\boldsymbol{K}_z\boldsymbol{\Psi}^{\phi} \quad \text{and}$$
$$\boldsymbol{M}_z^r = \boldsymbol{\Psi}^{\phi,T}\boldsymbol{M}_z\boldsymbol{\Psi}^{\phi}$$

and the final reduced linear operator can be constructed simply by adding these constituent matrices as:

$$\boldsymbol{A}^r(\boldsymbol{\mu}) = -D_1\boldsymbol{K}_1^r - D_2\boldsymbol{K}_2^r + \Sigma_1\boldsymbol{M}_1^r + \Sigma_2\boldsymbol{M}_2^r. \tag{2.21}$$

The same can be shown for source term $\boldsymbol{S}_{\text{ext}}(\boldsymbol{\mu})$ as well, since the external source function is considered to be region-wise constant. It is important to note that these reduced constituent operators and source terms are pre-computed once for all at the end of the training phase and reused every time the ROM needs to be evaluated at a new parameter or time instance, which makes the evaluation process of the ROM fast.

2.2.4.2 Treatment of the Nonlinear Terms

As already discussed, the solution of Eq. (2.18) still requires evaluating the full-order nonlinear function, of size N. This can be computationally expensive, often resulting in limited computation time savings when solving the reduced order model, compared to the original full-order model. To refer back to the previous subsection, the problem is that the nonlinear term is not affine (by

definition) in the elements of vector $\boldsymbol{c}^{\theta}$.

Depending on the nonlinear function itself, multiple approaches can be used to tackle this problem. The first approach is only applicable for nonlinearities which have polynomial-like behavior. This means that the term can be factorized as the product of linear operators times the solution function. For simplicity, let us consider the following example:

$$F(\theta) = \boldsymbol{\nabla} \cdot (\theta \boldsymbol{\nabla} \theta),$$

where the problem comes from multiplying θ by the gradient of itself. If the reduced-basis approximate $\tilde{\theta} = \sum_{i}^{r_\theta} \psi_i^\theta c_i^\theta$ is plugged into this expression we get:

$$F\left(\sum_{i}^{r_\theta} \psi_i^\theta c_i^\theta\right) = \boldsymbol{\nabla} \cdot \left(\sum_{i}^{r_\theta} \psi_i^\theta c_i^\theta \boldsymbol{\nabla} \left(\sum_{j}^{r_\theta} \psi_j^\theta c_j^\theta\right)\right) = \sum_{i}^{r_\theta} c_i^\theta \left(\boldsymbol{\nabla} \cdot \psi_i^\theta \boldsymbol{\nabla} \left(\sum_{j}^{r_\theta} \psi_j^\theta c_j^\theta\right)\right),$$

where the last equality used the fact that the divergence operator is linear. As the next step, the discretization of $\boldsymbol{\nabla} \cdot \psi_i^\theta \boldsymbol{\nabla}$ operator can be carried out and the resulting matrix is denoted by $\boldsymbol{F_i}$. Using this, the discretized form of the nonlinearity can be expressed as:

$$\boldsymbol{F}\left(\sum_{i}^{r_\theta} \boldsymbol{\psi}_i^\theta c_i^\theta\right) = \sum_{i}^{r_\theta} c_i^\theta \left(\boldsymbol{F_i} \boldsymbol{\Psi}^\theta \boldsymbol{c}^\theta\right).$$

To create the reduced operator, one needs to project this expression onto the subspace. Using a Galerkin projection, the final expression for the reduced nonlinear term becomes:

$$\boldsymbol{F}^r(\boldsymbol{c}_i^\theta) = \boldsymbol{\Psi}^{\theta,T} \boldsymbol{F}\left(\sum_{i}^{r_\theta} \boldsymbol{\psi}_i^\theta c_i^\theta\right) = \sum_{i}^{r_\theta} c_i^\theta \left(\boldsymbol{\Psi}^{\theta,T} \boldsymbol{F_i} \boldsymbol{\Psi}^\theta \boldsymbol{c}^\theta\right) = \sum_{i}^{r_\theta} c_i^\theta \boldsymbol{F}_i^r \boldsymbol{c}^\theta = \boldsymbol{c}^{\theta,T} \overline{\overline{\boldsymbol{F}}}^r \boldsymbol{c}^\theta,$$

where $\overline{\overline{\boldsymbol{F}}}^r$ denotes a third order tensor where each slice can be computed as: $\overline{\overline{\boldsymbol{F}}}_i^r = \left(\boldsymbol{\Psi}^{\theta,T} \boldsymbol{F_i} \boldsymbol{\Psi}^\theta\right)$. This procedure can be applied for higher order polynomial-like nonlinearities; however, the order of tensor $\overline{\overline{\boldsymbol{F}}}^r$ increases with the increasing polynomial order as well. This approach has been extensively utilized for the treatment of advection terms in the MSR specific reduced-order models.

A good example can be the advection term in the Navier-Stokes equation: $\boldsymbol{\nabla} \cdot (\rho \boldsymbol{u} \otimes \boldsymbol{u})$, which is a quadratic function of the velocity ($\boldsymbol{u}$, see Section 3.2 for notation).

A second option, which can be used for arbitrary nonlinear functions, is to use the Discrete Empirical Interpolation Method (DEIM) introduced in [62]. This requires the construction of a second snapshot matrix for the the nonlinear function:

$$\boldsymbol{R}_{\boldsymbol{F}} = [\boldsymbol{F}(\boldsymbol{\theta}(\boldsymbol{\mu}_1, t_1)), \ldots, \boldsymbol{F}(\boldsymbol{\theta}(\boldsymbol{\mu}_{N_\mu}, t_{N_\tau}))]. \tag{2.22}$$

Using these snapshots, a spatial global basis is generated for the nonlinear function using POD:

$$\boldsymbol{\Psi}^F = [\boldsymbol{\psi}_1^F, \ldots, \boldsymbol{\psi}_{r_F}^F] \in \mathbb{R}^{N \times r_F}. \tag{2.23}$$

These basis functions are used to subsequently interpolate $\boldsymbol{F}$. For this, DEIM also selects r_F distinct interpolation elements (corresponding to nodes or cells on the mesh) in vector $\boldsymbol{F}$ in order to assemble the DEIM interpolation selection matrix:

$$\boldsymbol{P} = [\boldsymbol{e}_{p_1}, \ldots, \boldsymbol{e}_{p_{r^F}}] \in \mathbb{R}^{N \times r^F}, \tag{2.24}$$

where $\boldsymbol{e}_i$ is the i-th canonical unit vector. The process to select these interpolation points is also adapted from [62] and is summarized in Algorithm 1. The method selects degrees of freedom for the interpolation where the best linear combination of the preceding basis functions has the highest error.

Once the interpolation points are obtained, the DEIM interpolant of $\boldsymbol{F}$ can be computed as:

$$\boldsymbol{\Psi}^F (\boldsymbol{P}^T \boldsymbol{\Psi}^F)^{-1} \boldsymbol{P}^T \boldsymbol{F}(\boldsymbol{\theta}(t;\boldsymbol{\mu}), t, \boldsymbol{\mu}) \approx \boldsymbol{\Psi}^F (\boldsymbol{P}^T \boldsymbol{\Psi}^F)^{-1} \boldsymbol{P}^T \boldsymbol{F}(\boldsymbol{\Psi}^\theta \boldsymbol{c}^\theta(t;\boldsymbol{\mu}), t, \boldsymbol{\mu})\,. \tag{2.25}$$

By plugging this into Eq. (2.18) one can obtain the following reduced system:

$$\frac{\partial \boldsymbol{c}^\theta(\boldsymbol{\mu}, t)}{\partial t} + \boldsymbol{A}^r(t;\boldsymbol{\mu}) \boldsymbol{c}^\theta(t;\boldsymbol{\mu}) + \boldsymbol{\Psi}^{\theta,T} \boldsymbol{\Psi}^F (\boldsymbol{P}^T \boldsymbol{\Psi}^F)^{-1} \boldsymbol{P}^T \boldsymbol{F}(\boldsymbol{\Psi}^\theta \boldsymbol{c}^\theta(t;\boldsymbol{\mu}), t; \boldsymbol{\mu}) = \boldsymbol{S}^r(t;\boldsymbol{\mu})\,. \tag{2.26}$$

Algorithm 1: Determination of interpolation points on the mesh. [62]

Result: $\boldsymbol{P}$
initialization: $\boldsymbol{\Psi}^F$;
$p_1 = \arg\max_i(|\boldsymbol{\psi}_1^F|_i)$;
$\boldsymbol{\Psi} = [\boldsymbol{\psi}_1^F]$; $\boldsymbol{P} = [\boldsymbol{e}_{p_1}]$;
for $m = 2$ to r_F **do**
  Solve $(\boldsymbol{P}^T\boldsymbol{\Psi})\boldsymbol{c} = \boldsymbol{P}^T\boldsymbol{\psi}_m^\theta$ for $\boldsymbol{c}$;
  $\boldsymbol{\delta} = \boldsymbol{\psi}_m^\theta - \boldsymbol{\Psi}\boldsymbol{c}$;
  $p_m = \arg\max_i(|\boldsymbol{\delta}|_i)$;
  $\boldsymbol{\Psi} = [\boldsymbol{\Psi} \;\; \boldsymbol{\psi}_m^F]$; $\boldsymbol{P} = [\boldsymbol{P} \;\; \boldsymbol{e}_{p_m}]$;
end

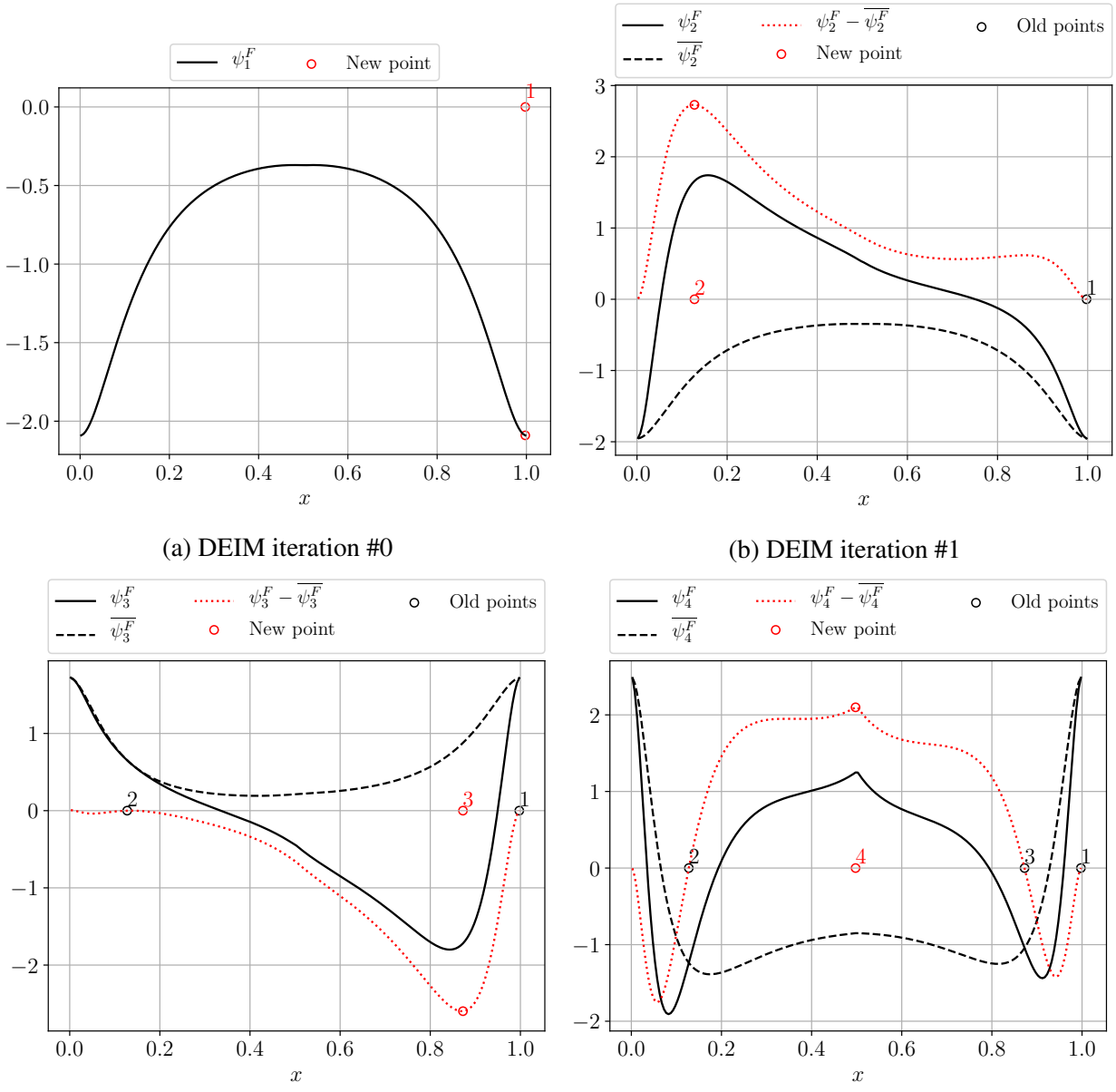
It is important to note that $\boldsymbol{P}^T\boldsymbol{F}$ extracts only a few ($r_F \ll N$) entries of $\boldsymbol{F}$ that need to be evaluated. This results in large computation time savings because discretization of partial differential equations usually results in local connectivity between an unknown and its neighbors, meaning that evaluating $\boldsymbol{P}^T\boldsymbol{F}(\boldsymbol{\Psi}^\theta\boldsymbol{c}^\theta(\boldsymbol{\mu},t),\boldsymbol{\mu})$ is fast. Furthermore, $\boldsymbol{\Psi}^{\theta,T}\boldsymbol{\Psi}^F(\boldsymbol{P}^T\boldsymbol{\Psi}^F)^{-1} \in \mathbb{R}^{r_x \times r_F}$ is a small matrix that can be pre-computed once during the training phase and can be reused every time $\boldsymbol{\mu}$ or $\boldsymbol{c}^\theta(\boldsymbol{\mu},t)$ changes.

This approach has been utilized to treat the $F(\phi) = e^{-\phi^2}$ nonlinear term in the one-dimensional example problem as well. The snapshots and basis functions for $\boldsymbol{F}$ have been already presented in Figures 2.2 and 2.4. The interpolation points have been selected using Algorithm 1. The first four steps of the iteration are visualized in Figure 2.5 with red and black circles indicating the newly and previously selected cells at the given iteration step, respectively. Recall that the new points are chosen to be at the maximum of the absolute difference between the actual and approximate POD modes ($|\psi_i^F - \overline{\psi_i^F}|$). In this scenario the reduced nonlinear term takes the following form:

$$\boldsymbol{\Psi}^{\phi,T}\boldsymbol{\Psi}^F(\boldsymbol{P}^T\boldsymbol{\Psi}^F)^{-1}\boldsymbol{P}^T e^{-(\boldsymbol{\Psi}^\phi\boldsymbol{c}^\phi(\boldsymbol{\mu}))^2} = \boldsymbol{\Psi}^{\phi,T}\boldsymbol{\Psi}^F(\boldsymbol{P}^T\boldsymbol{\Psi}^F)^{-1}e^{-(\boldsymbol{P}^T\boldsymbol{\Psi}^\phi\boldsymbol{c}^\phi(\boldsymbol{\mu}))^2}, \tag{2.27}$$

where $\boldsymbol{P}^T$ could enter the nonlinear term because it is evaluated cell-wise; or in other words, the value of the nonlinear function in a cell only depends on the solution in the same cell alone.

DEIM has been extensively utilized in this work for the approximation of the temperature and density feedback effects in neutronics problems and computing flow resistances in the reduced fluid dynamics sub-problem. These applications are discussed in Section 4.3 in detail. For additional information on DEIM, we refer the reader to [62, 63].



(a) DEIM iteration #0

(b) DEIM iteration #1

(c) DEIM iteration #2

(d) DEIM iteration #3

Figure 2.5: First four steps in the algorithm which determines the DEIM interpolation points on the mesh.

2.3 Online or Evaluation Phase

The subsequent, online phase includes the assembly and evaluation of the reduced equation, Eq. (2.26), together with the evaluation of the Quantities of Interest (QoIs).

2.3.1 Assembly and Solution of the Reduced System

If the original operators have affine decompositions and the nonlinearities are resolved using either a polynomial form or using DEIM, this phase simplifies into operations with small, dense matrices which are of low computational cost.

As an example, the expanded form of the reduced equation for the one-dimensional diffusion-reaction problem can be expressed as:

$$-\sum_{i=1}^{2}\left(D_i\boldsymbol{K}_i^r+\Sigma_i\boldsymbol{M}_i^r\right)\boldsymbol{c}^{\phi}+\boldsymbol{B}^r e^{-(\boldsymbol{P}^T\boldsymbol{\Psi}^{\phi}\boldsymbol{c}^{\phi}(\boldsymbol{\mu}))^2}=\sum_{i=1}^{2}S_i\boldsymbol{S}_i^r, \tag{2.28}$$

where $\boldsymbol{B}=\boldsymbol{\Psi}^{\phi,T}\boldsymbol{\Psi}^F(\boldsymbol{P}^T\boldsymbol{\Psi}^F)^{-1}\in\mathbb{R}^{r_\theta\times r_F}$. It is visible that in this case the elements of $\boldsymbol{\mu}=[D_1,D_2,\Sigma_1,\Sigma_2,S_1,S_2]^T$ model parameter vector are just parameters, meaning that, when the constituent matrices and vectors ($\boldsymbol{K}_i^r$, $\boldsymbol{M}_i^r$, $\boldsymbol{B}^r$, $\boldsymbol{S}_i^r$) are available, the assembly and solution of the system scales with $r_\phi, r_F \ll N$.

2.3.2 Evaluation of the Quantities of Interest

When the solution of the nonlinear reduced equation is obtained, the approximate solution can be reconstructed using Eq. (2.7). Then, using the approximate solution, the required Quantities of Interest (QoIs) can be extracted. Good examples for QoIs can be the extreme, point or average values of the solution, norms of the solution, the gradient of the solution, etc. This can be a costly procedure, compared to the solution of the ROM itself, since the size of the basis functions is N for scalar valued problems and $3N$ for vector valued problems. However, certain QoIs do not require the reconstruction of the approximate solution vector. For example, the point and average values

of the solution can be reconstructed as:

$$\theta(\boldsymbol{r}_0) = \sum_{i=1}^{r_\theta} \psi_i^\theta(\boldsymbol{r}_0) c_i^\theta \quad \text{and} \tag{2.29}$$

$$\overline{\theta} = \sum_{i=1}^{r_\theta} \overline{\psi}_i^\theta c_i^\theta, \tag{2.30}$$

where the only additional items that have to be stored are the point values ($\psi_i^\theta(\boldsymbol{r}_0)$) or the averages ($\overline{\psi}_i^\theta$) of the basis functions. These can be computed/saved at the offline phase stage, meaning that in such cases the evaluation of the QoI using a ROM can be quite efficient.

In case of the one-dimensional example problem, a convergence study has been carried out. A new sample batch of 100 parameter vectors has been created for testing ROMs built with different numbers of basis functions. The members of the test batch do not overlap with the members of the parameter samples used for collecting snapshots in the training phase. The number of bases for both $\boldsymbol{\phi}$ and $\boldsymbol{F}$ has been increased simultaneously. The new parameter samples have been inserted into the FOM and different ROMs and the mean L^2 difference have been computed between the solution vectors from the FOM and the approximate solution vectors from the ROMs. This also necessitates the reconstruction of the full approximate solution, which is a considerable effort compared to other low-order operations in the ROM. Figure 2.6 presents the decreasing error and increasing computation time with the increase in the rank of the reduced-subspace. The average computation time for the FOMs varied between 38 and 41 ms meaning that a speedup of 27–205 can be achieved per new parameter sample depending on the required accuracy.

2.4 Important Remarks

Lastly, it must be noted that it is only beneficial to train ROMs if the cost of the offline phase can be justified. In other words, if the training of a ROM with adequate accuracy takes less effort than the original task itself. Typical applications where this condition is met are uncertainty quantification and design optimization, where the original task is the evaluation of the FOM with hundreds or thousands of different input parameter combinations. In such cases, only a fraction of

the parameter samples are used to generate solutions which can be utilized as training data to generate a ROM. The rest of the parameter samples are then evaluated using the slightly less accurate, but orders of magnitude faster ROM. This results in an overall saving in computation time and allows better parameter space exploration for design optimization and richer statistics for uncertainty quantification.

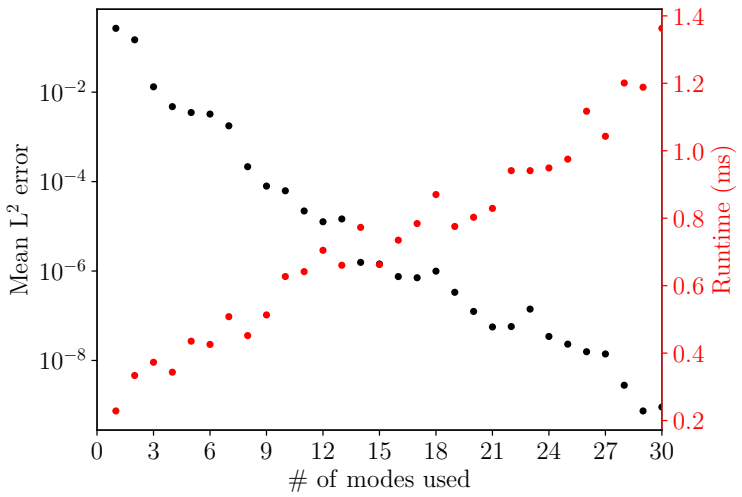


Figure 2.6: The mean L^2 error of the test set for the one-dimensional diffusion-reaction problem as function of then umber of POD modes used.

For the 1D example, we see that the ROM built using 30 snapshots (15 POD modes) is already accurate, exhibiting a mean L^2 error of approximately 10^{-6} over the validation set. Considering that the generation of the ROM takes only 5.45 ms and that every simulation using the FOM takes approximately 40 ms, the utilization of the ROM for additional computations with new parameter samples is justified.

2.5 Application of POD-RB ROMs

The original concept of POD can be traced back to the beginning of the 20-th century [57], but has gained an increased recognition over the last three decades in Computational Fluid Dynamics (CFD) and turbulence modeling [64, 65]. POD-RB ROMs are now commonly used for steady-

state and transient fluid flow simulations with finite element [66] and finite volume [67, 68] spatial discretizations. Recently, multiple POD-RB methods have been developed for the reduced-order modeling of the Reynolds Averaged Navier Stokes (RANS) equations [68, 69] providing a tool for handling turbulent fluid flows with high Reynolds numbers as well. Furthermore, the technique shows a great promise in controlling applications of fluid flows, especially when spatial effects in the system have to be considered [70]. On the other side of the spectrum, the applicability of POD-RB has been investigated for diffusion dominated Darcy flows in porous media as well [71] and is also frequently used in oceanography and meteorology for solving the shallow water equations [72] or for reducing layer dynamics [73, 74]. Moreover, it is used in the field of mechanics for describing system dynamics [75], vibrations [76] and elastic-plastic structures [77].

In the nuclear engineering community, applications of POD-RB-ROMs can be noted in reactor kinetics problems [78, 79, 80, 81], and fixed source, steady state neutral particle transport applications [82, 83, 84]. The utilization of POD-ROM for diffusion k-eigenvalue problems connected to reactor physics has been established in [85, 86, 87, 88]. The method has also been applied to the burnup analysis in [89]. Multiphysics model-order reduction is however still at experimental phase. One of the first approaches, discussed in [90], investigates a fuel pin in a lead-cooled reactor with cross-sections having a linear temperature dependence. Other works, like the one published in [91], uses a similar approach for Molten Salt Reactors but use logarithmic fuel temperature dependence for the cross sections and handles the coupling with Discrete Empirical Interpolation method. This direction is followed in this work with certain extensions. Another approach, described in [92], has already been developed for multiphysics simulations of Molten Salt Reactors operating in the laminar flow regime with clean fluid. This model used a Radial Basis Function (RBF) interpolation scheme for the treatment of temperature dependent cross-sections which resulted in decoupled heat transfer and neutronics equations at reduced-order level.

Compared to previous efforts, this work focuses on fully-coupled POD-RB ROMs applied to problems with fluid dynamics, heat transfer and neutronics phenomena. This, to the best of the author's knowledge, has not been investigated before. Furthermore, instead of clean fluid applica-

tions, this work focuses on flows with porous medium treatment which allows the simulation of homogenized fluid-structure interactions like heat exchange or flow resistance.

3. GOVERNING EQUATIONS AND FULL-ORDER MODELS FOR MOLTEN SALT REACTORS

This chapter introduces the mathematical models (governing equations) that need to be solved for Molten Salt Reactors. The discretization of the governing laws is also discussed, leading to the generation of so-called Full-Order Models.

3.1 The Governing Equations

Before initiating the ROM procedure, we select a set of partial differential equations, including initial and boundary conditions, that adequately describe the underlying physics of the problem at hand. Due to the fact that the fuel is dissolved in the coolant, the simulation of MSRs requires a complex multiphysics handling with coupled neutronics, fluid dynamics and heat transfer subproblems. The general structure of the multiphysics problem is visualized in Figure 3.1 together with the interactions between the different subproblems. The corresponding partial differential equations including these interactions in each subproblem are discussed in detail in the following subsections. In this work, the physical domain and its boundary are denoted by Υ and Γ, respectively.

3.1.1 Fluid Dynamics Subproblem

To avoid the explicit modeling of the complex structural elements located in nuclear reactor cores, a porous medium treatment [93, 94, 95, 96] is adopted in this work. This assumes that the structures are homogenized and interact with the fluid on a macro-scale, allowing coarse mesh treatment of the dominant physics. The following subsections discuss the way the mass and linear momentum conservation equations are treated in porous media together with additional aspects that need to be considered for the MSR related problems in this work.

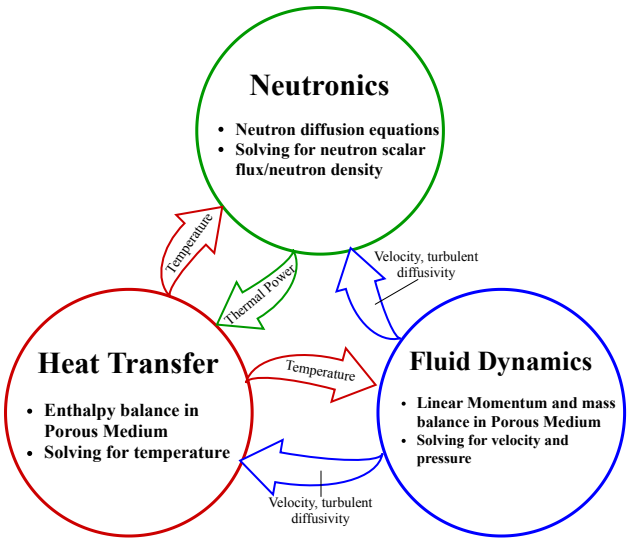


Figure 3.1: The simulated subproblems and the interactions between them.

3.1.1.1 Mass and Momentum Conservation in Porous Medium

The derivation of the mathematical model utilized can be found in [93] and only the final form is presented here. The mass and linear momentum conservation equations for a liquid in a porous medium with structural elements can be expressed as:

$$\boldsymbol{\nabla} \cdot \rho \boldsymbol{u}_D = 0\,, \qquad \boldsymbol{r} \in \Upsilon, \quad (3.1)$$

$$\frac{\partial \rho \boldsymbol{u}_D}{\partial t} + \frac{1}{\gamma}\boldsymbol{\nabla} \cdot (\rho \boldsymbol{u}_D \otimes \boldsymbol{u}_D) = \boldsymbol{\nabla} \cdot \left((\eta + \eta_t) \left[\boldsymbol{\nabla} \boldsymbol{u}_D + (\boldsymbol{\nabla} \boldsymbol{u}_D)^T \right] \right) - \gamma \boldsymbol{\nabla} p_t + \gamma \rho \boldsymbol{g} + \gamma \boldsymbol{F}_p + \gamma \boldsymbol{F}_{fr} + \gamma \rho \boldsymbol{g} \beta_{th} (T - T_{\text{ref}})\,, \qquad \boldsymbol{r} \in \Upsilon. \quad (3.2)$$

where $\boldsymbol{u}_D = \boldsymbol{u}_D(\boldsymbol{r}, t)$ denotes the Reynolds-averaged Darcy velocity vector field, $p_t = p_t(\boldsymbol{r}, t)$ the Reynolds-averaged total pressure field, $T = T(\boldsymbol{r}, t)$ the Reynolds-averaged temperature field, T_{ref} a reference temperature, η the molecular dynamic viscosity, ρ the density (taken as constant here) and β_{th} the thermal expansion coefficient. Furthermore, $\boldsymbol{g}$ is the gravitational acceleration, $\gamma = \gamma(\boldsymbol{r})$ is the porosity expressing the volume fraction occupied by the fluid, while $\boldsymbol{F}_p = \boldsymbol{F}_p(\boldsymbol{r}, t)$ and $\boldsymbol{F}_{fr} = \boldsymbol{F}_{fr}(\boldsymbol{r}, t)$ are volumetric linear momentum sources and sinks resulting from a pump and structures moving or obstructing the flow. It is important to note that the Darcy velocity can

be expressed as

$$\boldsymbol{u}_D(\boldsymbol{r},t) \equiv \gamma(\boldsymbol{r})\boldsymbol{u}(\boldsymbol{r},t), \tag{3.3}$$

where $\boldsymbol{u}(\boldsymbol{r},t)$ is the real velocity vector field. The conservation equations are solved for the Darcy velocity and the real velocity can be converted back using the expression above. This substitution ensures that the solution of the equation system is smooth in space. The pressure jump terms at the interfaces of different porous medium zones, discussed in [93], are not taken into account in this work because of the assumption that they are negligible compared to the pressure drops in the porous medium zones. It is also apparent that Eqs. (3.1)-(3.2) simplify to the Reynolds-Averaged Navier-Stokes (RANS) equations [97] with Boussinesq eddy viscosity approximation [98] and Boussinesq buoyancy approximation [99] if the fluid is clean, i.e., $\gamma = 1$. In this work, a Newtonian fluid model is assumed, meaning that the molecular dynamic viscosity is constant. The eddy viscosity, denoted by η_t, is used to model the additional dissipation effects introduced by turbulence.

The CFD solvers within the `OpenFOAM`© framework, and consequently in GeN-Foam as well, use a slightly different formulation for Eq. (3.2) by introducing a corrected pressure [100] using the following identity:

$$\boldsymbol{\nabla} p_t - \rho\boldsymbol{g} = \boldsymbol{\nabla}\left(p_t - \rho\boldsymbol{g}\cdot\boldsymbol{r}\right) = \boldsymbol{\nabla} p, \tag{3.4}$$

where $p = p(\boldsymbol{r},t)$ is the total pressure minus the hydrostatic pressure. Eq. (3.4) can be derived using Green's identity and the fact that ρ and $\boldsymbol{g}$ are constant fields. Plugging this back into Eqs. (3.2)–(3.1), we obtain

$$\boldsymbol{\nabla}\cdot\rho\boldsymbol{u}_D = 0\,, \qquad \boldsymbol{r}\in\Upsilon, \tag{3.5}$$

$$\begin{aligned}\frac{\partial\rho\boldsymbol{u}_D}{\partial t} + \frac{1}{\gamma}\boldsymbol{\nabla}\cdot(\rho\boldsymbol{u}_D\otimes\boldsymbol{u}_D) =& \boldsymbol{\nabla}\cdot\left((\eta+\eta_t)\left[\boldsymbol{\nabla}\boldsymbol{u}_D + (\boldsymbol{\nabla}\boldsymbol{u}_D)^T\right]\right) - \gamma\boldsymbol{\nabla} p \\ &+ \gamma\boldsymbol{F}_p + \gamma\boldsymbol{F}_{fr} + \gamma\rho\boldsymbol{g}\beta_{th}(T - T_{\text{ref}})\,, \qquad \boldsymbol{r}\in\Upsilon.\end{aligned} \tag{3.6}$$

These conservation equations are supplemented with appropriate initial and boundary conditions.

For an inlet-outlet problem (open loop), the most common boundary conditions can be written as

$$\boldsymbol{u}_D = \boldsymbol{u}_{D,in} \qquad \boldsymbol{r} \in \Gamma_{\text{in}}, \tag{3.7}$$

$$\boldsymbol{u}_D = \boldsymbol{0} \qquad \boldsymbol{r} \in \Gamma_{\text{wall}}, \tag{3.8}$$

$$\boldsymbol{\nabla} p \cdot \boldsymbol{n} = 0 \qquad \boldsymbol{r} \in \Gamma_{F,in} \cup \Gamma_{\text{wall}}, \tag{3.9}$$

$$\boldsymbol{n} \cdot \boldsymbol{\nabla} \boldsymbol{u}_i = 0 \qquad i \in [x, y, z],\ \boldsymbol{r} \in \Gamma_{\text{out}}, \tag{3.10}$$

$$p = p_{\text{out}} \qquad \boldsymbol{r} \in \Gamma_{\text{out}}, \tag{3.11}$$

where Γ_{in} denotes the inlet, Γ_{out} the outlet, and Γ_{wall} the wall segments of the boundary and $\boldsymbol{n}$ denotes the surface normal. For a closed-loop system, the boundary conditions simplify to

$$\boldsymbol{u}_D = \boldsymbol{0} \qquad \boldsymbol{r} \in \Gamma_{\text{wall}}, \tag{3.12}$$

$$\boldsymbol{\nabla} p \cdot \boldsymbol{n} = 0 \qquad \boldsymbol{r} \in \Gamma_{\text{wall}}. \tag{3.13}$$

The absence of Dirichlet boundary conditions, the incompressible fluid, and the fact that Eq. (3.6) contains only the gradient of the pressure entail that more than one pressure field can satisfy the problem. All of the possible solutions are just shifted by a constant pressure value. Thus, in a closed-loop system, the pressure needs to be specified at a chosen location ($\boldsymbol{r}_0$) in the domain to set the correct operating state:

$$p_t(\boldsymbol{r}_0, t) = p_{t,0} \quad \text{or} \quad p(\boldsymbol{r}_0, t) = p_{t,0} - \rho \boldsymbol{g} \cdot \boldsymbol{r}_0 \,. \tag{3.14}$$

Moreover, in many porous medium applications a no-slip boundary condition is preferred for the velocity at the walls which can be expressed as:

$$\boldsymbol{u}_D \cdot \boldsymbol{n} = 0, \quad \boldsymbol{r} \in \Gamma_{\text{wall}}, \tag{3.15}$$

meaning that the normal component of the velocity is enforced to be zero. Finally, the initial

velocity and pressure fields, $\boldsymbol{u}_D(\boldsymbol{r}, 0)$ and $p(\boldsymbol{r}, 0)$, are given and satisfy the boundary conditions.

3.1.1.2 Treatment of Volumetric Momentum Sources and Sinks

In the linear momentum conservation equation, terms $\boldsymbol{F}_{fr}$ and $\boldsymbol{F}_p$ describe the linear momentum sources and sinks resulting from the interactions between the fluid and the homogenized structure in the porous medium.

In this work, it is assumed that the momentum source of the pump, $\boldsymbol{F}_p$, is a vector field with region-wise constant values in the N_{pump} pump regions (Υ_{pump}) and zero everywhere else:

$$\boldsymbol{F}_p(\boldsymbol{r}) = \begin{cases} \boldsymbol{F}_{p,i} & \boldsymbol{r} \in \Upsilon_{pump,i}, \quad i = 1, ..., N_{pump} \\ \boldsymbol{0} & \boldsymbol{r} \in \Upsilon / \left(\bigcup_i \Upsilon_{pump,i} \right). \end{cases}$$

The flow resistance, on the other hand, cannot be considered independent of the fluid velocity. Based on [93, 101, 102], the following expression is used to approximate each component of this force:

$$\boldsymbol{F}_{fr,i} = -\frac{f_i \rho \boldsymbol{u}_{D,i}^2}{2 D_h \gamma^2}, \tag{3.16}$$

where D_h denotes the the hydraulic diameter of the structure and $f_{D,i}$ is the friction factor in direction $i = \{x, y, z\}$. This friction factor can be computed using empirical correlation functions in the following form (Blasius correlation [103]):

$$f_{D,i} = A_{f_{D,i}} \mathrm{Re}^{B_{f_{D,i}}}, \tag{3.17}$$

where the Reynolds number, Re, is determined using the real velocity as

$$\mathrm{Re} = \frac{|\boldsymbol{u}| D_h \rho}{\eta}, \tag{3.18}$$

and $A_{f_{D,i}}$ and $B_{f_{D,i}}$ parameters are specified based on factors such as the flow regime or the geometry.

3.1.1.3 Treatment of Turbulent Flows in Porous Medium

Turbulent flows are handled using Reynolds-averaged quantities in Eqs. (3.5)-(3.6) together with the Boussinesq eddy viscosity approximation [98]. This means that the molecular dynamic viscosity is augmented by a turbulent viscosity (η_t) which is a nonlinear function of the velocity and pressure, and can be computed for clean fluids using different eddy viscosity models such as the $k-\epsilon$ or $k-\omega$ models [97, 104]. In this work, the $k-\epsilon$ eddy viscosity model is used; this requires the solution of two additional equations for the turbulent kinetic energy k and its dissipation rate ϵ. The standard implementation for clean fluids in `OpenFOAM`©, and consequently in GeN-Foam as well, is based on the following formulation [105]:

$$\frac{\partial \rho k}{\partial t} + \boldsymbol{\nabla} \cdot (\rho \boldsymbol{u} k) - \boldsymbol{\nabla} \cdot (\rho D_{t,k} \boldsymbol{\nabla} k) = S_{t,k} - \rho\epsilon \tag{3.19}$$

$$\frac{\partial \rho \epsilon}{\partial t} + \boldsymbol{\nabla} \cdot (\rho \boldsymbol{u} \epsilon) - \boldsymbol{\nabla} \cdot (\rho D_{t,\epsilon} \boldsymbol{\nabla} \epsilon) = \frac{C_1 \epsilon}{k} \left(S_{t,k} + C_3 \frac{2}{3} k \boldsymbol{\nabla} \cdot \boldsymbol{u} \right) - C_2 \rho \frac{\epsilon^2}{k} \tag{3.20}$$

where $D_{t,k}$ and $D_{t,\epsilon}$ are the effective turbulent kinetic energy and dissipation diffusivities, $S_{t,k}$ the turbulent kinetic energy production rate, while C_1, C_2 and C_3 are constant parameters which might depend on the specific problem. For porous regions, however this formulation breaks down and, based on [93], these equations have to be modified by introducing equilibrium turbulent kinetic energies k_0, equilibrium dissipation rates ϵ_0, and a convergence rate $\lambda_{\epsilon/k}$ as follows:

$$\frac{\partial \rho k}{\partial t} + \boldsymbol{\nabla} \cdot (\rho \boldsymbol{u}_D k) - \boldsymbol{\nabla} \cdot (\rho D_{t,k} \boldsymbol{\nabla} k) = \delta(\gamma) \left[S_{t,k} - \rho\epsilon \right] + (1 - \delta(\gamma)) \lambda_{\epsilon/k} \rho \left[k_0 - k \right] \tag{3.21}$$

$$\begin{aligned} \frac{\partial \rho \epsilon}{\partial t} + \boldsymbol{\nabla} \cdot (\rho \boldsymbol{u}_D \epsilon) - \boldsymbol{\nabla} \cdot (\rho D_{t,\epsilon} \boldsymbol{\nabla} \epsilon) = & \delta(\gamma) \left[\frac{C_1 \epsilon}{k} \left(S_{t,k} + C_3 \frac{2}{3} k \boldsymbol{\nabla} \cdot \boldsymbol{u} \right) - C_2 \rho \frac{\epsilon^2}{k} \right] \\ & + (1 - \delta(\gamma)) \lambda_{\epsilon/k} \rho \left[\epsilon_0 - \epsilon \right] \end{aligned} \tag{3.22}$$

where $\delta(\gamma(\boldsymbol{r}))$ is a function that is 1 where $\gamma(\boldsymbol{r}) = 1$ (clean fluid) and 0 everywhere else. Methods to approximate the k_0 and ϵ_0 parameters are also discussed in [93]. These surrogate equations are supplemented with appropriate boundary conditions that are discussed in [97]. After solving these

equations, the turbulent thermal diffusivity α_t and viscosity η_t can be computed as:

$$\eta_t = C_\mu \frac{\rho k^2}{\epsilon}, \quad \text{and} \quad \alpha_t = \frac{\eta_t}{\text{Pr}_t}, \tag{3.23}$$

where C_μ is a constant which, most of the times, takes the value of 0.09 [105] and Pr_t is the turbulent Prandtl number which is usually between 0.75 and 0.95.

Finally, the computation of the buoyancy term requires the knowledge of the temperature which is obtained by solving the heat transfer sub-problem discussed in the following subsection.

3.1.2 Heat Transfer Subproblem

To be able do determine the temperature of the system, a porous medium enthalpy equation is solved coupled to the fluid dynamics equations:

$$\begin{aligned}\frac{\partial \gamma \rho c_p T}{\partial t} + \boldsymbol{\nabla} \cdot (\boldsymbol{u}_D \rho c_p T) = \boldsymbol{\nabla} \cdot (\gamma \left[k_l + c_p \alpha_t\right] \boldsymbol{\nabla} T) \\ - h A_V (T - T_{ext}) + \gamma \int_0^\infty \Sigma_p(\boldsymbol{r}, E') \phi(\boldsymbol{r}, E') dE', \quad \boldsymbol{r} \in \Upsilon\end{aligned} \tag{3.24}$$

where c_p and k_l are the heat capacity and thermal conductivity of the fluid, while h is the heat transfer coefficient, T_{ext} is an external temperature and A_V is the volumetric area of the heat sink. Furthermore, it is visible that additional turbulent mixing effects are considered through a diffusion term utilizing α_t turbulent diffusivity. This formulation has been adapted from [93] by neglecting the terms used for kinetic energy conservation. This simplification is based on the assumption that the kinetic energy of the fluid is considerably less than its enthalpy. The last term on the right hand side of the equation expresses a volumetric heat deposition resulted from the fission reactions induced by the neutrons. This is handled using a power cross section $\Sigma_p(\boldsymbol{r}, E')$ and the neutron scalar flux $\phi(\boldsymbol{r}, E')$. These quantities are discussed in Section 3.1.3 in detail. For simplicity, zero gradient boundary conditions are employed for the temperature field in this work:

$$\boldsymbol{\nabla} T \cdot \boldsymbol{n} = 0, \quad \boldsymbol{r} \in \Gamma. \tag{3.25}$$

Lastly, it is assumed that the initial temperature field $T(\boldsymbol{r}, 0)$ is known and satisfies the boundary conditions.

3.1.3 Neutronics Subproblem

For the analysis of nuclear reactors, knowing the distribution of neutrons is essential. Several methods have been developed to determine this variable. Solution techniques utilizing Monte-Carlo approaches simulate particle histories in the reactor from birth to death (i.e., absorption or leakage) and give a statistical estimate of neutron density based on the observed population [106]. Despite their capability to simulate detailed geometries, Monte Carlo methods are rarely used in a multiphysics setting. The reason behind this is that obtaining an accurate, high-resolution estimate for neutron distribution in the reactor requires an immense amount of simulated histories, making this method computationally expensive. Other methods formulate a neutron balance equation which can be discretized in the independent variables and solved numerically. The most accurate mathematical model describing the balance of neutrons in an infinitesimal small space-energy-direction phase-space volume around $(\boldsymbol{r}, E, \boldsymbol{\Omega})$ at time t is the Neutron Transport Equation [107]. Discrete ordinates (S_N) and spectral methods using spherical harmonic expansion (P_N) are commonly used to solve this equation [108]. Due to the six-dimensional phase-space, however, solving the neutron transport equation is computationally demanding and is not widely used in multiphysics applications. Fortunately, several approximations of the neutron transport equation have been developed to obtain models which are easier to solve. The neutron diffusion equation belongs to the family of these simplified models and can be derived using multiple methods, including asymptotics on the neutron transport equation [109, 110, 107]. This equation yields good results for large and homogeneous reactors if the dominant reaction between the neutrons and the matter is scattering. Since the cores of MSRs are highly homogeneous and the scattering ratios are typically high, the neutron diffusion equation is used in this work. The results presented in [55] support this approach as well. This section briefly reviews the neutron diffusion based equations for the simulation of different MSR-related problems in this work.

3.1.3.1 Balance Equation for Neutrons

The equation describing the balance of neutrons in the reactor can be expressed as follows [107]:

$$\begin{aligned}\frac{1}{v(E)}\frac{\partial\phi(E,\boldsymbol{r},t)}{\partial t} = \quad & \boldsymbol{\nabla}\cdot[D(E,\boldsymbol{r})\boldsymbol{\nabla}\phi(E,\boldsymbol{r},t)] \\ & -\Sigma_t(E,\boldsymbol{r})\phi(E,\boldsymbol{r},t) \\ & +\frac{\big(1-\beta(\boldsymbol{r})\big)\chi_p(E,\boldsymbol{r})}{\mathrm{k_{eff}}}\int_0^\infty \nu(E',\boldsymbol{r})\Sigma_f(E',\boldsymbol{r})\phi(E',\boldsymbol{r},t)dE' \\ & +\int_0^\infty \Sigma_s(E'\rightarrow E,\boldsymbol{r})\phi(E',\boldsymbol{r},t)dE' \\ & +\chi_d(E,\boldsymbol{r})\sum_{i=1}^{G_d}\lambda_i(\boldsymbol{r})C_i(\boldsymbol{r},t), \quad \boldsymbol{r}\in\Upsilon \end{aligned} \tag{3.26}$$

where $v(E)$ denotes the velocity of neutrons, $\phi(E',\boldsymbol{r},t)$ the neutron scalar flux, $D(E,\boldsymbol{r})$ the diffusion coefficient, $\Sigma_t(E,\boldsymbol{r})$ macroscopic total cross section, β the total delayed neutron fraction, $\chi_p(E,\boldsymbol{r})$ the prompt neutron spectrum, $\nu(E,\boldsymbol{r})$ the fission neutron yield, $\Sigma_f(E',\boldsymbol{r})$ the macroscopic fission cross section, $\mathrm{k_{eff}}$ the effective multiplication factor, $\Sigma_s(E'\rightarrow E,\boldsymbol{r})$ is the differential macroscopic scattering cross section from energy E' to E. Furthermore, $\chi_d(E,\boldsymbol{r})$ is the delayed neutron spectrum , G_d is the number of delayed neutron groups, while $\lambda_i(\boldsymbol{r})$ and $C_i(\boldsymbol{r},t)$ are the corresponding decay constant and precursor concentration in precursor group i. The terms on the right hand side of the equation describe, in order, the neutron leakage, the neutron loss due to reactions, the neutron source from fission, the neutron source from scattering and the neutron source from the decay of the delayed neutron precursors. It must be mentioned that, in this equation, $\mathrm{k_{eff}}$ is just a constant that can be used to change the fission neutron yield across the entire reactor. This is a common tool to insert/withdraw reactivity into/from the system. The problems presented in this work use either reflective (homogeneous Neumann)

$$-D(E,\boldsymbol{r},t)\boldsymbol{\nabla}\phi(E,\boldsymbol{r},t)\cdot\boldsymbol{n}(\boldsymbol{r})=0, \quad \boldsymbol{r}\in\Gamma_{\text{Refl}} \tag{3.27}$$

zero scalar flux (homogeneous Dirichlet)

$$\phi(E, \boldsymbol{r}, t) = 0, \quad \boldsymbol{r} \in \Gamma_{\text{Dir}} \tag{3.28}$$

or albedo (homogeneous Robin)

$$-D(E, \boldsymbol{r}, t)\boldsymbol{\nabla}\phi(E, \boldsymbol{r}, t) \cdot \boldsymbol{n}(\boldsymbol{r}) = \gamma_r \phi(E, \boldsymbol{r}, t), \quad \boldsymbol{r} \in \Gamma_{\text{Alb}} \tag{3.29}$$

boundary conditions for the scalar flux, where Γ_{Refl}, Γ_{Dir} and Γ_{Alb} are the corresponding segments of Γ and $\boldsymbol{n}$ is the surface normal vector. In case of albedo boundary condition, $\gamma_r = \frac{1}{2}$ ensures a zero-incoming current boundary condition, while decreasing this parameter allows the simulation of external reflectors.

3.1.3.2 Balance Equation for Delayed Neutron Precursors

To close Eq. (3.26), additional equations are needed to describe the balance of the delayed neutron precursors in each group. These equations are especially important in the case of Molten Salt Reactors, where the fluid can move precursors out of the high importance regions of the reactor core. Implications of this delayed neutron drift can be the oscillation of reactivity in time and a considerable decrease in the effective delayed neutron fraction. Furthermore, in zones treated as porous medium, the fluid transporting the delayed neutron precursors does not occupy the entire volume, meaning that the volumetric density of the delayed neutrons have to be adjusted with the porosity. The equations describing the balance of delayed neutron precursors in porous media can be expressed by modifying the clean fluid equation in [111] as follows:

$$\begin{aligned}\frac{\partial\left[\gamma C_i^*(\boldsymbol{r}, t)\right]}{\partial t} + \boldsymbol{\nabla} \cdot \left[\boldsymbol{u}_D(\boldsymbol{r}, t) C_i^*(\boldsymbol{r}, t)\right] =& \boldsymbol{\nabla} \cdot \left(\left[\frac{\alpha_l(\boldsymbol{r})}{\rho} + \frac{\alpha_t(\boldsymbol{r}, t)}{\rho}\right] \boldsymbol{\nabla} C_i^*(\boldsymbol{r}, t)\right) \\ &+ \frac{\beta_i(\boldsymbol{r})}{\mathrm{k_{eff}}} \int_0^\infty \nu(E', \boldsymbol{r}) \Sigma_f(E', \boldsymbol{r}) \phi(E', \boldsymbol{r}, t) dE' \\ &- \lambda_i(\boldsymbol{r}) \gamma C_i^*(\boldsymbol{r}, t), \quad \boldsymbol{r} \in \Upsilon, \quad i \in [0, ..., G_d]\end{aligned} \tag{3.30}$$

where $\boldsymbol{u}_D(\boldsymbol{r},t)$ denotes the Darcy velocity vector field, $C_i^* = \frac{C_i}{\gamma}$ is the corrected precursor concentration, $\alpha_l(\boldsymbol{r})$ is the laminar diffusivity of the precursors, and $\beta_i(\boldsymbol{r})$ the delayed neutron fraction in group i. This equation is solved for the corrected precursor concentrations and the C_i is reconstructed afterwards. The substitution is important, since C_i^* is smooth in space, while C_i is not. Furthermore, $\alpha_l(\boldsymbol{r})$ can be computed using the molecular dynamic viscosity of the fluid, η, and the Schmidt number, Sc, as [101]:

$$\alpha_l = \frac{\eta}{\mathrm{Sc}}, \tag{3.31}$$

and $\alpha_t(\boldsymbol{r},t)$ is an additional diffusivity caused by turbulent mixing (defined in Section 3.1.2) in the fluid and is determined following the procedure described in Section 3.1.1. In case of laminar fluid flows the value of α_t is zero. In this work, only homogeneous Neumann boundary conditions are used for the corrected precursor equations:

$$\boldsymbol{\nabla} C_i^*(\boldsymbol{r},t)\cdot\boldsymbol{n}(\boldsymbol{r}) = 0, \quad \boldsymbol{r}\in\Gamma, \quad i\in[0,...,G_d]. \tag{3.32}$$

3.1.3.3 Eigenvalue Problems

In certain cases one is interested in the steady-state scalar flux and precursor concentrations. However, the system of (3.26)-(3.30) with the time-derivatives set to zero and homogeneous boundary conditions might not have a solution. In fact a steady state solution is only guaranteed if a volumetric or boundary external source is present and the reactor is subcritical or if no external sources are present and the reactor is critical. Therefore, in these situations, $\mathrm{k_{eff}}$ is assumed to be a free parameter that can be tuned to ensure that the system is critical in a sense that the number of generated neutrons equals to the number of neutrons lost. This transforms the mentioned system into a generalized eigenvalue problem where $\frac{1}{\mathrm{k_{eff}}}$ serves as the eigenvalue (denoted by $\frac{1}{k^*}$ momentarily) and the scalar flux with the corrected delayed neutron precursor concentrations form the

eigenvectors:

$$-\boldsymbol{\nabla}\cdot[D\boldsymbol{\nabla}\phi]+\Sigma_t\phi-\int_0^\infty\Sigma_s(E'\rightarrow E)\phi dE'-\chi_d\sum_{i=1}^{G_d}\lambda_i\gamma C_i^*=\frac{\big(1-\beta\big)\chi_p}{k^*}\int_0^\infty\nu\Sigma_f\phi dE' \tag{3.33}$$

$$\boldsymbol{\nabla}\cdot(\boldsymbol{u}_D C_i^*)-\boldsymbol{\nabla}\cdot\left(\left[\frac{\alpha_l}{\rho}+\frac{\alpha_t}{\rho}\right]\boldsymbol{\nabla}C_i^*\right)+\lambda_i\gamma C_i^*=\frac{\beta_i}{k^*}\int_0^\infty\nu\Sigma_f\phi dE' \tag{3.34}$$

Note that from now on, for the sake of being concise, the arguments of the field variables and cross sections are only shown when crucial for the formulation. In most practical applications, only the largest k^* is needed, thus the system can be solved using power iteration [107]. The largest k^* is commonly denoted by k_{eff}. During the power iteration, the eigenvectors are normalized to ensure that the thermal power of the reactor is a fixed value, P_{th}:

$$P_{th}=\int_\Upsilon\int_0^\infty\Sigma_p(\boldsymbol{r},E)\phi(\boldsymbol{r},E)dEd\Upsilon, \tag{3.35}$$

where $\Sigma_p(\boldsymbol{r},E)$ is a power cross section.

3.1.3.4 Temperature-dependent Cross Sections

To add an additional level of complexity to the problem, we consider the fact that the cross sections in Eqs. (3.26) and (3.30) are temperature-dependent. This can be expressed using the general notation of a cross section as:

$$\Sigma(E,\boldsymbol{r})=\Sigma(E,\boldsymbol{r},T), \tag{3.36}$$

where $T=T(\boldsymbol{r},t)$ denotes the temperature field. This dependence includes the effects of the Doppler broadening of the resonances in the cross sections together with the changing density of the fuel salt. This also means that the neutronics equations have to be coupled not only to the fluid dynamics equations, but to the energy equation as well. The treatment of temperature-dependent cross sections is addressed in Section 3.2 in detail.

3.1.3.5 Adjoint Eigenvalue Problems

In most MSRs, the effective fraction of delayed neutrons is an important parameter. This differs from the physical fractions, since during operation, a certain amount of delayed neutrons is moved by the fluid to lower neutron importance regions. In this work, only the steady-state delayed neutron fractions are investigated. To be able to determine these, the adjoint problem of Eqs. (3.33)-(3.34) has to be solved for the neutron importance fields. Based on [112], the adjoint problem for moving precursors can be expressed as

$$-\boldsymbol{\nabla}\cdot\left[D\boldsymbol{\nabla}\phi^{\dagger}\right]+\Sigma_t\phi^{\dagger}-\int_0^{\infty}\Sigma_s(E\rightarrow E')\phi^{\dagger}dE'=\frac{(1-\beta)\nu\Sigma_f}{k^*}\int_0^{\infty}\chi_p\phi^{\dagger}dE'+\frac{\nu\Sigma_f}{k^*}\sum_{i=1}^{G_d}\beta_i\gamma C_i^{*,\dagger} \tag{3.37}$$

$$\boldsymbol{\nabla}\cdot\left(-\boldsymbol{u}_D C_i^{*,\dagger}\right)-\boldsymbol{\nabla}\cdot\left(\left[\frac{\alpha_l}{\rho}+\frac{\alpha_t}{\rho}\right]\boldsymbol{\nabla}C_i^{*,\dagger}\right)+\lambda_i\gamma C_i^{*,\dagger}=\lambda_i\sum_{i=1}^{G_e}\chi_{d,i}\phi^{\dagger}, \tag{3.38}$$

where $\phi^{\dagger}$ denotes the neutron importance and $C_i^{*,\dagger}$ is the corrected precursor importance. The original precursor importance can be computed as $C_i^{\dagger}=\gamma C_i^{*,\dagger}$, where γ is the porosity. The same homogeneous boundary conditions can be used for these equations than the ones described for Eqs. (3.33)-(3.34). It is important to mention that, for the adjoint flux, the state (velocity, temperature, density) of the system is fixed to the solution fields arising from the forward problem potentially coupled with the fluid-dynamics and energy equations. Therefore, this equation system is not coupled to any of the other physics, it is solved in a standalone manner. Assuming that the steady-state scalar flux, delayed neutron precursor concentrations and the neutron importance are available, the following expression can be derived for the effective delayed neutron fraction [113] for precursor group i:

$$\beta_{\text{eff},i}=\frac{\int_0^{\infty}\int_{\Upsilon}\phi^{\dagger}\chi_d\lambda_i C_i d\Upsilon dE}{\int_0^{\infty}\int_{\Upsilon}\phi^{\dagger}\left(\chi_d\sum_{j=0}^{G_d}\lambda_j C_j+\chi_p\int_0^{\infty}\nu\Sigma_f\phi dE'\right)d\Upsilon dE}, \tag{3.39}$$

and the total effective delayed neutron fraction (β_{eff}) can be computed as the sum of the group-wise fractions $\beta_{\text{eff}} = \left(\sum_i \beta_{\text{eff},i} \right)$.

3.2 Full-Order Model

The Full-Order Model (FOM) consists of the discretized forms of the governing equations described in Section 3.1. This section briefly summarizes the spatial, temporal, and, for the neutronics subproblem, the energy discretization schemes applied in this work. The solvers which create and solve the FOM have been implemented in GeN-Foam (Generalized Nuclear Foam) [93], an OpenFoam® [114] based finite-volume multiphysics framework specifically developed for nuclear applications. This framework has been chosen because, besides being open-source, it grants easy access to discretized operators which makes the generation of intrusive POD-RB ROMs convenient.

3.2.1 Discretization in Space

The spatial discretization in OpenFOAM is based on the Finite Volume Method (FVM) [115, 116] which is widely used in CFD applications due to its stability properties, and the fact that it can handle arbitrary cell shapes. Furthermore, since this method is conservative, it is preferred in applications involving conservation laws. The general idea behind this discretization scheme is reviewed in this subsection. It involves the introduction of a spatial mesh by splitting the computational domain Υ into N non-overlapping cells, $\Upsilon_i,\ (i = 1, ..., N)$, with volumes of V_i. Figure 3.2 shows an example of a cell in the mesh and introduces the general notation needed for the subsequent derivations.

A general time-dependent diffusion-advection-reaction problem is considered to review the discretization approach in the following form:

$$\frac{\partial(\rho\theta)}{\partial t} + \boldsymbol{\nabla} \cdot (\rho \boldsymbol{u} \theta) = \boldsymbol{\nabla} \cdot (D_\theta \boldsymbol{\nabla} \theta) + F(\theta, \boldsymbol{u}) + Q_\theta, \tag{3.40}$$

where θ is a conserved variable, ρ is the physical density, $\boldsymbol{u}$ is the velocity vector field, D_θ is

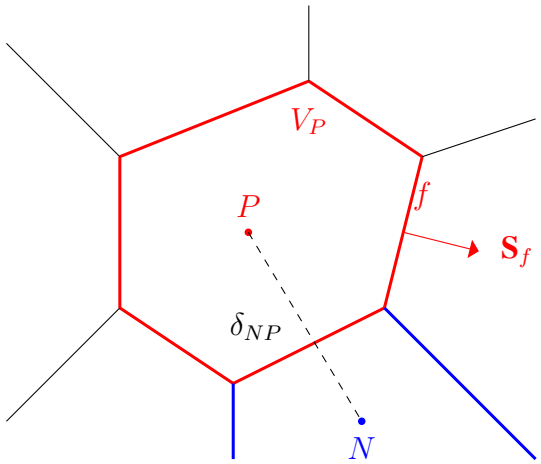


Figure 3.2: A general cell with centroid at P used for the finite volume discretization. (N - centroid of neighboring cell, f - face index, $\mathbf{S}_f$ - surface vector)

the diffusion coefficient, while $F(\theta, \boldsymbol{u})$ expresses a general nonlinear reaction term and Q_θ is an external source term. Note that every equation in Section 3.1 has a similar form. The spatial discretization schemes for different terms can be derived by integrating Eq. (3.40) over a cell Υ_P (with volume V_P) and using the Gauss divergence theorem on the diffusion and advection terms. This yields the following form using the notation in Figure 3.2:

$$\frac{\partial}{\partial t}\int_{\Upsilon_P}(\rho\theta)dV + \int_{\Gamma_P}\rho\theta\boldsymbol{u}\cdot\boldsymbol{n}dS = \int_{\Gamma_P}D_\theta\boldsymbol{\nabla}\theta\cdot\boldsymbol{n}dS + \int_{\Upsilon_P}F(\theta,\boldsymbol{u})dV + \int_{\Upsilon_P}Q_\theta dV, \tag{3.41}$$

where Γ_P denotes the entire boundary of the cell. As a next step, the conserved variable is assumed to be constant, θ_P over the cell and θ_f over side f of the boundary of the cell. Furthermore, it is assumed that the material properties (density, diffusion coefficient, etc.) and the value of the nonlinear function are constants within the cell and on the boundary segments as well. Using these assumptions the integrals in Eq. (3.41) can be expressed as:

$$\frac{\partial}{\partial t}(\rho_P\theta_P V_P) + \sum_f \rho_f\theta_f\boldsymbol{u}_f\cdot\boldsymbol{S}_f = \sum_f D_{\theta,f}\boldsymbol{\nabla}\theta_f\cdot\boldsymbol{S}_f + F(\theta_P,\boldsymbol{u}_P)V_P + Q_{\theta,P}V_P, \tag{3.42}$$

It is apparent that the order of accuracy of the discretization scheme depends on the approximation

of the face values and gradients using the cell values. In most of the cases, this involves linear approximations for the face values:

$$\theta_f \approx w\theta_P + (1-w)\theta_N, \tag{3.43}$$

with N denoting the centroid of a neighboring cell, while the normal gradients at the face can be approximated as:

$$\boldsymbol{\nabla}\theta_f \cdot \boldsymbol{S}_f \approx \frac{1}{\cos\left(\alpha_{\delta \boldsymbol{S}}\right)} \frac{\theta_N - \theta_P}{|\boldsymbol{\delta}_{PN}|} |\boldsymbol{S}_f| + \boldsymbol{\nabla}\theta_f^{\text{old}} \cdot \left(\frac{\boldsymbol{S}_{f,i}}{|\boldsymbol{S}_f|} - \frac{1}{\cos\left(\alpha_{\delta \boldsymbol{S}}\right)} \frac{\boldsymbol{\delta}_{PN}}{|\boldsymbol{\delta}_{PN}|} \right) |\boldsymbol{S}_f|, \tag{3.44}$$

where the second term expresses an explicit correction for cases when the face surface normal is not parallel to the vector connecting the cell centers ($\boldsymbol{S}_f \nparallel \boldsymbol{\delta}_{PN}$). Furthermore, $\alpha_{\delta \boldsymbol{S}}$ denotes the angle between $\boldsymbol{S}_f$ and $\boldsymbol{\delta}_{PN}$. These approximations yield a second order scheme in space. However, in certain cases this is not achievable due to oscillations caused by the advection term. In these scenarios, an upwind scheme is used to compute the face values:

$$\theta_f \approx \begin{cases} \theta_P & \text{if} \quad \boldsymbol{u} \cdot \boldsymbol{S}_f > 0 \\ \theta_N & \text{if} \quad \boldsymbol{u} \cdot \boldsymbol{S}_f < 0 \end{cases}$$

which yields a first order scheme. Nevertheless, it is visible that by repeating the same process for every cell in the mesh and applying the proper boundary conditions, the partial differential equation has been transformed into a system of ordinary differential equation where the unknowns are the values of θ at the cell centers.

3.2.2 Discretization in Time

Following the discretization in space we are left with a system of ordinary differential equation in a form of:

$$\boldsymbol{M}\frac{\partial \boldsymbol{\theta}}{\partial t} = \mathbf{A}(\boldsymbol{\theta})\boldsymbol{\theta} + \mathbf{Q}_\theta(\boldsymbol{\theta}), \tag{3.45}$$

where $\boldsymbol{\theta}$ is a vector containing the values of θ_P for each cell, $\boldsymbol{M}$ is a diagonal mass matrix, $\mathbf{A}(\boldsymbol{\theta})$ is a system matrix that can potentially depend on $\boldsymbol{\theta}$, while $\mathbf{Q}_\theta(\boldsymbol{\theta})$ is a source term which might also depend on $\boldsymbol{\theta}$. This equation still needs to be integrated in time over a time step $\Delta t_i = [t_i, t_{i+1}]$:

$$\boldsymbol{M} \int_{t_i}^{t_{i+1}} \frac{\partial \boldsymbol{\theta}}{\partial t} dt = \int_{t_i}^{t_{i+1}} \left[\mathbf{A}(\boldsymbol{\theta})\boldsymbol{\theta} + \mathbf{Q}_\theta(\boldsymbol{\theta})\right] dt, \tag{3.46}$$

The handling of the left hand side is simple, however the integration of the right hand side requires more attention. In general, several approximations are used to integrate the right hand side in time. Typical approaches include the approximation of the integral by the weighted sum of the integrand evaluated at different points within the time step. Obviously, the order of the integration scheme depends on the selection of these locations and the corresponding weights. In this work, an implicit Euler discretization scheme has been used which yields first order accuracy:

$$\boldsymbol{M}\boldsymbol{\theta}(t_i) - \boldsymbol{M}\boldsymbol{\theta}(t_{i+1}) = \left[\mathbf{A}(\boldsymbol{\theta}(t_{i+1}))\boldsymbol{\theta}(t_{i+1}) + \mathbf{Q}_\theta(\boldsymbol{\theta}(t_{i+1}))\right] \Delta t_i \tag{3.47}$$

and requires the solution of a nonlinear equation system at each time-step. This means that a fixed-point iteration is necessary at every time step to resolve the nonlinearities. This can be achieved by lagging the solution in the nonlinear operators and source terms:

$$\boldsymbol{M}\boldsymbol{\theta}(t_i) - \boldsymbol{M}\boldsymbol{\theta}(t_{i+1})^{l+1} = \left[\mathbf{A}(\boldsymbol{\theta}(t_{i+1})^{l})\boldsymbol{\theta}(t_{i+1})^{l+1} + \mathbf{S}_\theta(\boldsymbol{\theta}(t_{i+1})^{l})\right] \Delta t_i, \tag{3.48}$$

where l is the index in the fixed-point iteration. In this case a linear algebraic equation system needs to be solved at every iteration step which requires a linear solver with appropriate preconditioning.

3.2.3 Discretization in Neutron Energy

Looking at Eq. (3.26) and Eq. (3.30) we notice that the equations are still continuous in neutron energy, meaning that this independent variable needs to be discretized before applying the spatial and temporal discretization schemes mentioned in the previous subsections. In this work, the

multigroup diffusion equation has been used which can be derived from Eq. (3.26) and Eq. (3.30) with splitting the energy spectrum into small intervals, as shown in Figure 3.3. It is visible that the

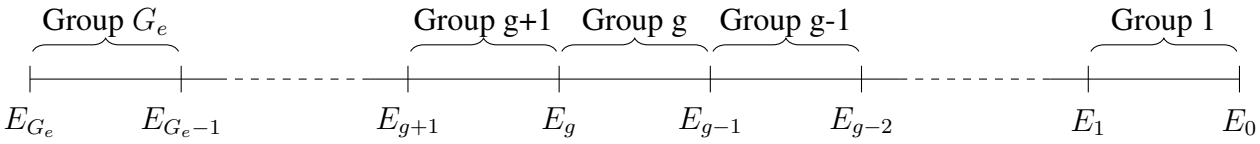


Figure 3.3: The discretized energy spectrum.

energy spectrum is split into G_e intervals or groups. It is common practice to assign index 1 to the interval with the highest energy and G_e to the lowest energy group. By integrating Eq. (3.26) over a $[E_g, E_{g+1}]$ interval one can derive the multigroup diffusion equations for the neutron scalar flux:

$$\frac{1}{v_g}\frac{\partial \phi_g}{\partial t} = \boldsymbol{\nabla}\cdot[D_g\boldsymbol{\nabla}\phi_g] - \Sigma_{t,g}\phi_g + \frac{(1-\beta)\chi_{p,g}}{\mathrm{k_{eff}}}\sum_{g'=1}^{G_e}\nu_{g'}\Sigma_{f,g'}\phi_{g'} + \sum_{g'=1}^{G_e}\Sigma_{s,g'\to g}\phi_{g'} + \chi_{d,g}\sum_{i=1}^{G_d}\lambda_i C_i, \quad \boldsymbol{r}\in\Upsilon, \quad g\in[1,...,G_e], \tag{3.49}$$

where ϕ_g is the scalar flux in energy group g defined as:

$$\phi_g = \phi_g(\boldsymbol{r},t) = \int_{E_g}^{E_{g+1}} \phi(E,\boldsymbol{r},t)dE, \tag{3.50}$$

while v_g, D_g, $\Sigma_{t,g}$, $\chi_{p,g}$, ν_g, $\Sigma_{f,g}$, $\Sigma_{s,g'\to g}$, $\chi_{d,g}$ are space-dependent group constants which can be derived from the coefficients in Eq. (3.26) using different averaging techniques. In this paper, these coefficients are generated using the Serpent 2 Monte Carlo particle transport code [117]. The same types of boundary conditions can be applied to each energy group problem as the ones described

in Section 3.1. Similarly, the corresponding precursor equations can be expressed as:

$$\frac{\partial \gamma C_i^*}{\partial t} + \boldsymbol{\nabla} \cdot [\boldsymbol{u}_D C_i^*] = \boldsymbol{\nabla} \cdot \left(\left[\frac{\alpha_l}{\rho} + \frac{\alpha_t}{\rho} \right] \boldsymbol{\nabla} C_i^* \right) + \frac{\beta_i}{k_{\text{eff}}} \sum_{g'=1}^{G_e} \nu_{g'} \Sigma_{f,g'} \phi_{g'} - \lambda_i \gamma C_i^*, \quad \boldsymbol{r} \in \Upsilon. \quad i \in [0, ..., G_d] \quad (3.51)$$

The adjoint problem can be discretized in a similar manner resulting in a multigroup adjoint diffusion problem.

3.2.4 Treatment of Temperature-dependent Cross Sections

Lastly, the treatment of the temperature-dependent cross sections have to be addressed as well. As already mentioned, this work considers two main effects that result from the changing temperature of the molten salt fuel:

i Due to the fact that the nuclear fuel is within the coolant, the changing coolant temperature can result in neutronics Doppler effects. This is a negative feedback meaning that the increasing fuel temperature decreases the reactivity of the system.

ii The density of the coolant is temperature-dependent as well, meaning that the increasing temperature decreases the coolant density. This is also a negative feedback meaning that the increasing temperature decreases the amount of fuel within the core causing a drop in reactivity.

The implementation in GeN-Foam requires the definition of three cross section libraries: one for a reference point and one for each feedback at a perturbed state. Using these libraries, the temperature-dependent cross sections are computed by the following interpolation/extrapolation schemes:

$$\Sigma(\boldsymbol{r}, T, \rho(T)) \approx \Sigma(\boldsymbol{r}, T_{ref}, \rho(T_{ref})) + \delta_{FT}(\boldsymbol{r}) \left(\sqrt{T} - \sqrt{T_{ref}} \right) + \delta_{FD}(\boldsymbol{r}) \left(\rho(T) - \rho(T_{ref}) \right), \quad (3.52)$$

where the δ coefficients can be determined as:

$$\delta_{FT}(\boldsymbol{r}) = \left[\frac{\Sigma(\boldsymbol{r}, T_{ref}, \rho(T_{ref})) - \Sigma(\boldsymbol{r}, T_{pert}, \rho(T_{ref}))}{\sqrt{T_{ref}} - \sqrt{T_{pert}}} \right] \tag{3.53}$$

$$\delta_{FD}(\boldsymbol{r}) = \left[\frac{\Sigma(\boldsymbol{r}, T_{ref}, \rho(T_{ref})) - \Sigma(\boldsymbol{r}, T_{ref}, \rho(T_{pert}))}{\rho(T_{ref}) - \rho(T_{pert})} \right]. \tag{3.54}$$

This work assumes that the density of the molten salt has a linear temperature dependence in the following form:

$$\rho(T) = \rho(T_{ref}) + \beta_{th}\rho(T_{ref})(T - T_{ref}),$$

where β_{th} is the thermal expansion coefficient. Using this, the interpolation can be simplified into a purely temperature-dependent form:

$$\Sigma(\boldsymbol{r}, T) \approx \Sigma(\boldsymbol{r}, T_{ref}, \rho(T_{ref})) + \delta_{FT}(\boldsymbol{r}) \left(\sqrt{T} - \sqrt{T_{ref}} \right) + \delta_{FD}(\boldsymbol{r})\beta_{th}\rho(T_{ref}) \left(T - T_{ref} \right). \tag{3.55}$$

For fast spectrum reactors, the density effect stays the same; however, for the fuel temperature effect, a logarithmic interpolation is used:

$$\Sigma(\boldsymbol{r}, T) \approx \Sigma(\boldsymbol{r}, T_{ref}, \rho(T_{ref})) + \delta_{FT}(\boldsymbol{r}) \ln \left(\frac{T}{T_{ref}} \right) + \delta_{FD}(\boldsymbol{r})\beta_{th}\rho(T_{ref}) \left(T - T_{ref} \right). \tag{3.56}$$

where the δ_{FT} factor is computed as:

$$\delta_{FT}(\boldsymbol{r}) = \left[\frac{\Sigma(\boldsymbol{r}, T_{ref}, \rho(T_{ref})) - \Sigma(\boldsymbol{r}, T_{pert}, \rho(T_{ref}))}{\ln \left(\frac{T_{ref}}{T_{pert}} \right)} \right]. \tag{3.57}$$

3.2.5 Iteration scheme

This subsection briefly reviews the iteration strategy used to solve the above mentioned system of discretized partial differential equations. GeN-Foam utilizes a multi-layer fixed-point iteration to obtain solutions for the fields of interest. Figure 3.4 shows this iteration scheme which was derived from the original version presented in [93] with applying the incompressible fluid flow

constraint. The PIMPLE (PISO+SIMPLE) algorithm [118] is used to solve the discretized fluid dynamics equations. This involves the creation of a Poisson equation for pressure using the already discretized velocity equation [116]. It involves an inner loop iterating on pressure and an outer loop iterating on velocity, temperature and the $k - \epsilon$ equations. It is visible, that the velocity-temperature coupling is resolved first and only when this has converged, progresses the solver to the neutronics-temperature coupling. In case of a steady-state computation, the process is the

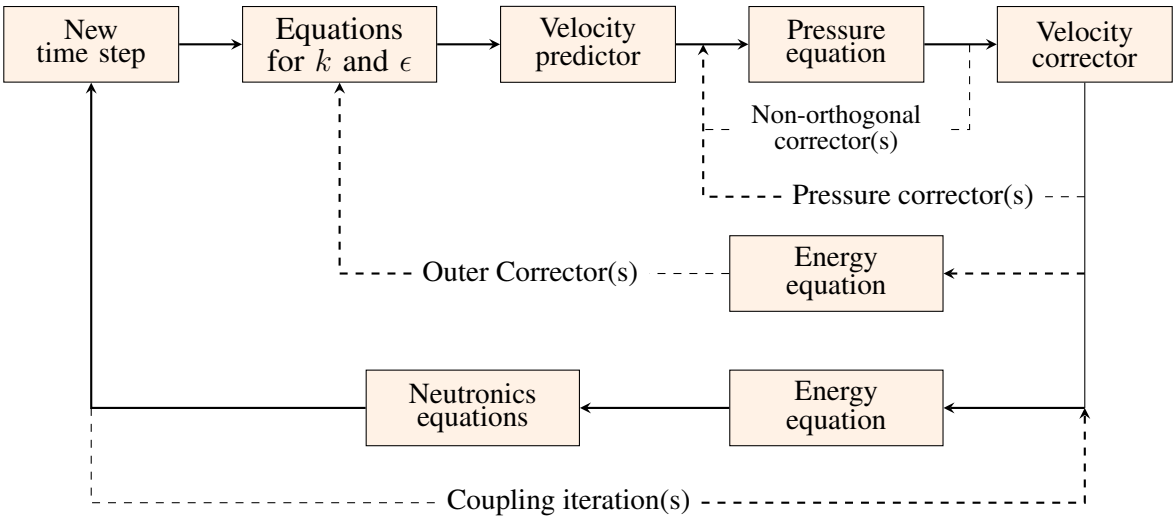


Figure 3.4: The coupling scheme of the Full-Order Models. (solid lines - the default coupling, dashed lines - possible inner iterations)

same; the simulations are run until the problems reach steady state using pseudo-time stepping instead of real time.

As previously outlined in Figure 2.1, the training phase consists of three main steps. First, a Full-Order Model (FOM) is created by discretizing the underlying governing laws on a specific domain. This process, together with the MSR-related governing laws have been covered in Chapter 3. The remaining two steps, namely the data-driven learning of reduced subspaces, and the computation of reduced operators are discussed in the following sections in detail. For the sake of better clarity, the two sub processes of the data-driven learning step, namely: (i) collection of the data and (ii) the generation of the basis functions of the subspaces, are discussed separately.

4.1 Data Collection

As part of the learning process, this step consists of collecting data about the time- and parameter-dependent solution manifolds. For this, the method of snapshots is utilized [64], in which FOM simulations are performed to obtain N_s snapshots of the system at different time instances and/or input/uncertain parameter values. Let us assume that the FOM is exercised using N_μ different parameter samples and snapshots are captured at N_τ different time instances. With this, the total number of snapshots is $N_s = N_\mu \times N_\tau$ for a parametric transient scenarios. For steady-state simulations, the number of snapshots equals the number of parameter samples ($N_s = N_\mu$). In the case of finite volume discretization, a snapshot is a vector containing the values of a scalar field (e.g., scalar flux, temperature, etc.) or vector field (e.g., velocity) at the center of each mesh cell. These solution vectors are then collected in snapshot matrices denoted by $\boldsymbol{R}_*$. Therefore, the dimension of the snapshot matrices is $N \times N_s$ for scalar fields and $3N \times N_s$ for vector fields in a three dimensional domain, where N is the number of cells in the mesh. For the fluid dynamics equations, the Darcy velocity and corrected pressure fields are saved in every scenario:

$$\boldsymbol{R}_{\boldsymbol{u}_D} = [\boldsymbol{u}_{D,1}, ..., \boldsymbol{u}_{D,N_s}], \tag{4.1}$$

$$\boldsymbol{R}_p = [p_1, ..., p_{N_s}]. \tag{4.2}$$

while for turbulent flows, the eddy viscosity and eddy diffusivity fields need to be saved as well:

$$\boldsymbol{R}_{\eta_t} = [\eta_{t,1}, ..., \eta_{t,N_s}], \tag{4.3}$$

$$\boldsymbol{R}_{\alpha_t} = [\alpha_{t,1}, ..., \alpha_{t,N_s}], \tag{4.4}$$

For porous medium simulations, the obstructing force ($\boldsymbol{F}_{\text{fr}}$) is saved for each of the N_Z porous medium zones in the geometry:

$$\boldsymbol{R}_{\boldsymbol{F}_{fr,z}} = [\boldsymbol{F}_{fr,z,1}, ..., \boldsymbol{F}_{fr,z,N_s}], \quad z = 1, .., N_Z. \tag{4.5}$$

For the heat transfer subproblem, one needs to save the resulting temperature fields:

$$\boldsymbol{R}_T = [T_1, ..., T_{N_s}], \tag{4.6}$$

and, depending on the interpolation method chosen for the treatment of the temperature-dependent cross sections, either the square root of the temperature $\sqrt{T}$ or the logarithm of the temperature $\ln(T)$. For the sake of generality, this auxiliary temperature field will be denoted by

$$T^{\text{aux}} = \begin{cases} \sqrt{T}, & \text{if square root treatment (thermal reactor)} \\ \ln(T), & \text{if logarithmic treatment (fast reactor)} \end{cases}$$

and are saved in the corresponding snapshot matrix:

$$\boldsymbol{R}_{T^{\text{aux}}} = [T_1^{\text{aux}}, ..., T_{N_s}^{\text{aux}}]. \tag{4.7}$$

Lastly, for the neutronics subproblem we generally save the neutron flux for each energy group and the corrected concentration for each delayed neutron precursor group:

$$\boldsymbol{R}_{\phi_g} = [\phi_{g,1}, ..., \phi_{g,N_s}], \quad g = 1, ..., G_e, \tag{4.8}$$

$$\boldsymbol{R}_{C_i^*} = [C_{i,1}^*, ..., C_{i,N_s}^*], \quad i = 1, ..., G_d, \tag{4.9}$$

while for adjoint problems, the neutron and corrected precursor importances are saved in a similar fashion:

$$\boldsymbol{R}_{\phi_g^\dagger} = [\phi_{g,1}^\dagger, ..., \phi_{g,N_s}^\dagger], \quad g = 1, .., G_e, \tag{4.10}$$

$$\boldsymbol{R}_{C_i^{\dagger,*}} = [C_{i,1}^{\dagger,*}, ..., C_{i,N_s}^{\dagger,*}], \quad i = 1, .., G_d. \tag{4.11}$$

4.2 Generating Reduced Subspaces

In reduced basis techniques, the solution fields are approximated using basis functions with global support on domain Υ. This section discusses the specific fields that need to be approximated together with the steps to obtain the corresponding basis functions for the heat transfer, fluid dynamics and neutronics subproblems.

4.2.1 Reduced Bases for Heat Transfer Subproblem

The field variable associated with this subproblem is the temperature, which can be approximated using the corresponding global basis-functions (ψ^T) as:

$$T \approx \widetilde{T} = \sum_{i=1}^{r_T} \psi_i^T c_i^T = \boldsymbol{\Psi}^T \boldsymbol{c}^T. \tag{4.12}$$

The basis functions of the reduced subspace are generated using the POD procedure from Section 2.2.3 without any modifications. The heat transfer subproblem has an additional auxiliary temperature field which is necessary for the coupling with the neutronics subproblem. This auxiliary field ($\sqrt{T}$ or $\ln(T)$ depending on the cross section interpolation method) is approximated

as:

$$T^{\text{aux}} \approx \widetilde{T}^{\text{aux}} = \sum_{i=1}^{r_{T^{\text{aux}}}} \psi_i^{T^{\text{aux}}} c_i^{T^{\text{aux}}} = \boldsymbol{\Psi}^{T^{\text{aux}}} \boldsymbol{c}^{T^{\text{aux}}}. \tag{4.13}$$

The basis functions for the auxiliary temperature has also been generated using the generic POD technique in Section 2.2.3.

4.2.2 Reduced Bases for Fluid Dynamics Subproblem

In this work two reduction techniques are used for the fluid dynamics equations. The first technique, introduced in [119] and extended to turbulent flows and finite volume approximation in [68], employs a one-equation model with an additional physics-based approximation at reduced level. This method is referred to as POD-FV-ROM in [120], but in this work we denote it as one-equation ROM to make the comparison to the other approach conspicuous. The second approach employs a two-equation ROM with supremizer stabilization [67, 69, 121]. The manner in which the POD bases are generated is slightly different between the one- and two-equation ROMs. In case of the two-equation ROM, the fluid dynamics related solution fields are approximated as:

$$\boldsymbol{u}_D \approx \widetilde{\boldsymbol{u}}_D = \sum_{i=1}^{r_{\boldsymbol{u}_D}} \psi_i^{\boldsymbol{u}_D} c_i^{\boldsymbol{u}_D} = \boldsymbol{\Psi}^{\boldsymbol{u}_D} \boldsymbol{c}^{\boldsymbol{u}_D}, \quad \text{and} \quad p \approx \widetilde{p} = \sum_{i=1}^{r_p} \psi_i^p c_i^p = \boldsymbol{\Psi}^p \boldsymbol{c}^p, \tag{4.14}$$

where the corresponding basis functions are obtained using the general POD procedure discussed in Section 2.2.3. It is important to mention that instead of the real velocity, we approximate the Darcy velocity, since it is smooth in space. This property is advantageous in numerical methods involving the weak formulation of the given problem (like POD-RB), since the smoother the functions are, the better they can be approximated with global basis functions. Moreover, the fluid dynamics subproblem has two auxiliary variables as well: the turbulent dynamic viscosity (η_t) and the turbulent diffusivity (α_t). These can also be approximated as:

$$\eta_t \approx \widetilde{\eta}_t = \sum_{i=1}^{r_{\eta_t}} \psi_i^{\eta_t} c_i^{\eta_t} = \boldsymbol{\Psi}^{\eta_t} \boldsymbol{c}^{\eta_t}, \quad \text{and} \quad \alpha_t \approx \widetilde{\alpha}_t = \sum_{i=1}^{r_{\alpha_t}} \psi_i^{\alpha_t} c_i^{\alpha_t} = \boldsymbol{\Psi}^{\alpha_t} \boldsymbol{c}^{\alpha_t}, \tag{4.15}$$

where the bases are obtained using the general POD procedure. These fields are crucial for the simulation of the additional dissipation effects in turbulent flows.

In the case of the one-equation method, the mentioned fields are approximated slightly differently. This approach assumes that there is a close correlation between the dynamics of the velocity, pressure, eddy viscosity and eddy diffusivity modes, therefore the mentioned fields can be expressed as:

$$\boldsymbol{u}_D \approx \widetilde{\boldsymbol{u}}_D = \sum_{i=1}^{r_{\boldsymbol{u}_D}} \psi_i^{\boldsymbol{u}_D} c_i^{\boldsymbol{u}_D} = \boldsymbol{\Psi}^{\boldsymbol{u}_D} \boldsymbol{c}^{\boldsymbol{u}_D} , \qquad p \approx \widetilde{p} = \sum_{i=1}^{r_{\boldsymbol{u}_D}} \psi_i^{p} c_i^{\boldsymbol{u}_D} = \boldsymbol{\Psi}^{p} \boldsymbol{c}^{\boldsymbol{u}_D} \tag{4.16}$$

$$\eta_t \approx \widetilde{\eta}_t = \sum_{i=1}^{r_{\boldsymbol{u}_D}} \psi_i^{\eta_t} c_i^{\boldsymbol{u}_D} = \boldsymbol{\Psi}^{\eta_t} \boldsymbol{c}^{\boldsymbol{u}_D} , \qquad \alpha_t \approx \widetilde{\alpha}_t = \sum_{i=1}^{r_{\boldsymbol{u}_D}} \psi_i^{\alpha_t} c_i^{\boldsymbol{u}_D} = \boldsymbol{\Psi}^{\alpha_t} \boldsymbol{c}^{\boldsymbol{u}_D} . \tag{4.17}$$

Note that the coefficients of the velocity subspace ($\boldsymbol{c}^{\boldsymbol{u}_D}$) are used for every field. Consequently, the basis vectors of the reduced subspaces have to be obtained in a slightly different manner as well. After obtaining the eigendecomposition of the velocity correlation matrix as

$$\boldsymbol{\mathcal{C}}^{\boldsymbol{u}_D} = \boldsymbol{V}^{\boldsymbol{u}_D} \boldsymbol{W}^{\boldsymbol{u}_D} \boldsymbol{V}^{\boldsymbol{u}_D, T}, \tag{4.18}$$

the modes for all of the additional fields are selected as follows

$$\psi_i^{\boldsymbol{u}_D} = \frac{1}{\sqrt{\omega_i^{\boldsymbol{u}_D}}} \sum_{k=0}^{N_s} \boldsymbol{u}_{D,k} \boldsymbol{V}_{i,k}^{\boldsymbol{u}_D} , \qquad \psi_i^{p} = \frac{1}{\sqrt{\omega_i^{\boldsymbol{u}_D}}} \sum_{k=0}^{N_s} p_k \boldsymbol{V}_{i,k}^{\boldsymbol{u}_D} , \tag{4.19}$$

$$\psi_i^{\eta_t} = \frac{1}{\sqrt{\omega_i^{\boldsymbol{u}_D}}} \sum_{k=0}^{N_s} \eta_{t,k} \boldsymbol{V}_{i,k}^{\boldsymbol{u}_D} , \qquad \psi_i^{\alpha_t} = \frac{1}{\sqrt{\omega_i^{\boldsymbol{u}_D}}} \sum_{k=0}^{N_s} \alpha_{t,k} \boldsymbol{V}_{i,k}^{\boldsymbol{u}_D} . \tag{4.20}$$

It is visible that the eigenvectors and the eigenvalues of the velocity correlation matrix are used to construct the subspaces for every other field variable. This means that, pressure, eddy viscosity and turbulent diffusivity modes are required to follow the same dynamics as the velocity modes. This approximation was first introduced in [119] and has been numerically demonstrated to yield accurate ROMs for some practical applications with inlet-outlet clean (no porous medium) fluid flows [120].

Lastly, there is one more auxiliary field which needs to be approximated in case if the obstructing force ($\boldsymbol{F}_{\text{fr}}$) in the porous medium is considerable. The treatment of this field is the same for both ROM approaches:

$$\boldsymbol{F}_{fr,z} \approx \widetilde{\boldsymbol{F}}_{fr,z} = \sum_{i=1}^{r_{\boldsymbol{F}_{fr,z}}} \psi_i^{\boldsymbol{F}_{fr,z}} c_i^{\boldsymbol{F}_{fr,z}} = \boldsymbol{\Psi}^{\boldsymbol{F}_{fr,z}} \boldsymbol{c}^{\boldsymbol{F}_{fr,z}}. \tag{4.21}$$

where z is the index of the porous medium zone. This means that a separate basis is generated for the flow resistance in every porous region. These bases are obtained using the regular POD procedure for both the one-equation and the two-equation techniques.

4.2.3 Reduced Bases for the Neutronics Subproblem

A natural question arises during the generation of reduced subspaces for the neutronics subproblem: "How should the multi-group nature of the neutron flux and the precursor concentration be addressed?". This topic has been partially investigated in [88] on conventional nuclear reactor benchmark problems with stationary fuel and without considering precursors. Based on the study, there are three main approaches:

1. Full neutronics state vectors (ξ) are created by concatenating the corresponding snapshots of the neutron fluxes and precursor concentrations:

$$\xi_i = \begin{bmatrix} \phi_{1,i} \\ \vdots \\ \phi_{G_e,i} \\ C^*_{1,i} \\ \vdots \\ C^*_{G_d,i} \end{bmatrix}, \quad i = 1, ..., N_s. \tag{4.22}$$

   One can organize these merged snapshots into a snapshot matrix $\boldsymbol{R}_\xi$ and carry out the general POD procedure (Section 2.2.3) to obtain basis functions ψ_i^ξ, $i = 1, ..., r_\xi$, for these combined state vectors. In this scenario, the weighting matrix $\boldsymbol{M}$ is a block-diagonal matrix with

$G_e + G_d$ blocks of $\boldsymbol{M}_\Upsilon$. In this case the neutronics field variables are approximated as:

$$\xi \approx \widetilde{\xi} = \sum_{i=1}^{r_\xi} \psi_i^\xi c_i^\xi = \boldsymbol{\Psi}^\xi \boldsymbol{c}^\xi. \tag{4.23}$$

This means that every group flux and precursor concentration field shares the same coefficients in the reduced subspace. Without precursors, this approach yields accurate results for fixed source and time-dependent problems. However, it also exhibits a serious weakness when applied to eigenvalue problems. Based on [88], it turns out that if the neutronics subproblem is reduced using this subspace, spurious complex eigenvalues emerge at reduced-order level with larger magnitudes than the largest eigenvalue of the FOM ($\mathrm{k_{eff}}$). This would pose serious difficulties, since the present work focuses on the comparison of the eigenvalues with largest magnitudes only.

2. The second approach improves on the previous one by splitting the basis functions ψ_i^ξ into sections corresponding to each neutron flux and precursor group:

$$\psi_i^\xi = \begin{bmatrix} \psi_{i,1}^\xi \\ \vdots \\ \psi_{i,G_e}^\xi \\ \psi_{i,G_e+1}^\xi \\ \vdots \\ \psi_{i,G_e+G_d}^\xi \end{bmatrix} \begin{matrix} \rightarrow & \psi_i^{\phi_1} \\ \vdots & \vdots \\ \rightarrow & \psi_i^{\phi_{G_e}} \\ \rightarrow & \psi_i^{C_1^*} \\ \vdots & \vdots \\ \rightarrow & \psi_i^{C_{G_d}^*} \end{matrix}, \quad i = 1, ..., r_\xi. \tag{4.24}$$

This allows the utilization of different coefficients for the corresponding segments of the combined state vector during the approximation:

$$\phi_g \approx \widetilde{\phi}_g = \sum_{i=1}^{r_\xi} \psi_i^{\phi_g} c_i^{\phi_g} = \boldsymbol{\Psi}^{\phi_g} \boldsymbol{c}^{\phi_g}, \quad g = 1, ..., G_e, \quad \text{and} \tag{4.25}$$

$$C_g^* \approx \widetilde{C}_g^* = \sum_{i=1}^{r_\xi} \psi_i^{C_g^*} c_i^{C_g^*} = \boldsymbol{\Psi}^{C_g^*} \boldsymbol{c}^{C_g^*}, \quad g = 1, ..., G_d. \tag{4.26}$$

This approach allows a group-wise reduction of the neutronics equations. The details of

this reduction method are discussed in Section 4.3.2.1. Without the presence of precursors, based on [88], this approach does not exhibit the issue of the method discussed under the previous bullet. One disadvantage of this method is that the bases for the group fluxes and precursor concentrations are not group-wise orthonormal and optimal. Furthermore, every group needs to use r_ξ basis functions even when there are groups with smoother parameter dependence which would require fewer.

3. To circumvent the shortcomings of the previous two approaches, [88] proposed another method which generates reduced subspaces for the group fluxes and precursor concentrations one-by-one without concatenating the snapshots into a single state vector. In this case, the general POD procedure presented in Section 2.2.3 is carried out using the original snapshot matrices $\boldsymbol{R}_{\phi_g}$ and $\boldsymbol{R}_{C_g^*}$. This allows a flexible selection of the number of bases for different energy and precursor groups for the approximation:

$$\phi_g \approx \widetilde{\phi}_g = \sum_{i=1}^{r_{\phi_g}} \psi_i^{\phi_g} c_i^{\phi_g} = \boldsymbol{\Psi}^{\phi_g} \boldsymbol{c}^{\phi_g}, \quad g = 1, ..., G_e, \quad \text{and} \tag{4.27}$$

$$C_g^* \approx \widetilde{C}_g^* = \sum_{i=1}^{r_{C_g^*}} \psi_i^{C_g^*} c_i^{C_g^*} = \boldsymbol{\Psi}^{C_g^*} \boldsymbol{c}^{C_g^*}, \quad g = 1, ..., G_d. \tag{4.28}$$

Furthermore, this approach does not exhibit issues in case of eigenvalue problems either.

Of course, the same approaches exist for the adjoint problem as well. Due to the improved flexibility and robustness in case of eigenvalue problems, the last approach has been utilized in this work. This means that a group-wise basis is built for every forward and adjoint group flux and precursor concentration. Similarly to the Darcy velocity, the corrected precursor concentrations are approximated here instead of the real concentrations, due to the fact that they are smooth in space. This intuitively suggests, that they can be better approximated using global basis functions.

4.3 Generating the Reduced Operators

The next step in the training phase is the generation of reduced operators. In this work, they are obtained by using the approximate solutions in Section 4.2 in the governing equations in 3.1 and applying a Galerkin projection. This means that the same basis vectors are used for the projection, as for the approximation. The following subsections contain the necessary details for deriving the final forms of the reduced equations for each subproblem.

4.3.1 Reduced Fluid Dynamics Equations

As already mentioned in terms of the generation of the reduced subspaces, there are two methods for the generation of fluid dynamics ROMs. The main difference between the two approaches is the number of reduced equations they employ. The derivation of these two ROMs are discussed in the following subsections together with the handling of the nonlinear terms and certain stabilization approaches.

4.3.1.1 The Two-equation Reduced-Order Model

To derive this reduced-order model, approximations (4.14), (4.15) and (4.12) are plugged into Eqs. (3.5)-(3.6) and the bases in (4.14) are used for the projection:

$$\Big\langle \psi_i^{\boldsymbol{u}_D},\ \frac{\partial \rho \widetilde{\boldsymbol{u}}_D}{\partial t} + \frac{1}{\gamma}\boldsymbol{\nabla}\cdot(\rho\widetilde{\boldsymbol{u}}_D \otimes \widetilde{\boldsymbol{u}}_D) - \boldsymbol{\nabla}\cdot\Big((\eta+\widetilde{\eta})\left[\boldsymbol{\nabla}\widetilde{\boldsymbol{u}}_D + (\boldsymbol{\nabla}\widetilde{\boldsymbol{u}}_D)^T\right]\Big)$$
$$+\gamma\boldsymbol{\nabla}\widetilde{p} - \gamma\boldsymbol{F}_p - \gamma\widetilde{\boldsymbol{F}}_{fr} - \gamma\rho\boldsymbol{g}\beta_{th}(\widetilde{T}-T_{ref})\Big\rangle_{\Upsilon} = 0 \quad i = 1, ..., r_{\boldsymbol{u}_D}, \quad (4.29)$$

$$\langle \psi_i^p,\ \boldsymbol{\nabla}\cdot\rho\widetilde{\boldsymbol{u}}_D\rangle_{\Upsilon} = 0, \quad i = 1, ..., r_p, \quad (4.30)$$

Note that the approximate solutions can be expressed as the linear combinations of the corresponding basis functions, and therefore, the above formulation can be written purely in terms of basis function-dependent reduced operators and time- and parameter-dependent coefficients. Before discussing the complete reduced equations, it is noted that in this work, ρ, η, β_{th}, T_{ref} and $\boldsymbol{g}$ are constants while γ can vary in space, but not in time or parameter space. Using these assumptions,

the two-equation reduced fluid dynamics equations can be expressed as:

$$\rho \boldsymbol{M}\dot{\boldsymbol{c}}^{\boldsymbol{u}_D} + \rho \boldsymbol{c}^{\boldsymbol{u}_D,T}\overline{\overline{\boldsymbol{C}}}\boldsymbol{c}^{\boldsymbol{u}_D} - \boldsymbol{c}^{\eta,T}\overline{\overline{\boldsymbol{T}}}\boldsymbol{c}^{\boldsymbol{u}_D} - \eta \boldsymbol{D}\boldsymbol{c}^{\boldsymbol{u}_D} + \boldsymbol{P}\boldsymbol{c}^{p} + \Gamma(\boldsymbol{B}\boldsymbol{c}^{\boldsymbol{u}_D} - |\boldsymbol{u}_{D,in}|\boldsymbol{S}_r^{BD}) \tag{4.31}$$

$$-\sum_{z=1}^{Z}\left(|\boldsymbol{F}_{p,z}|\boldsymbol{S}_{p,z} - \boldsymbol{S}_{fr,z}\boldsymbol{c}_z^{\boldsymbol{F}_{fr}}\right) - \rho\beta_{th}(\boldsymbol{A}\boldsymbol{c}^{T} - T_{ref}\boldsymbol{S}_T) = 0, \tag{4.32}$$

$$\rho \boldsymbol{G}\boldsymbol{c}^{\boldsymbol{u}_D} = 0 \tag{4.33}$$

where the matrices and vectors are denoted by **bold**, while higher order tensors have been distinguished by adding an additional double bar symbol on the top. Furthermore, index z denotes the porous medium zones in the domain. The elements of the low-order operators in the equations above are computed as:

$$\boldsymbol{M}_{i,j} = \left\langle \psi_i^{\boldsymbol{u}_D}, \psi_j^{\boldsymbol{u}_D} \right\rangle_{\Upsilon} \qquad \overline{\overline{\boldsymbol{C}}}_{i,j,k} = \left\langle \psi_j^{\boldsymbol{u}_D}, \frac{1}{\gamma}\boldsymbol{\nabla}\cdot(\psi_i^{\boldsymbol{u}_D}\otimes\psi_k^{\boldsymbol{u}_D}) \right\rangle_{\Upsilon}$$

$$\boldsymbol{D}_{i,j} = \left\langle \psi_i^{\boldsymbol{u}_D}, \boldsymbol{\nabla}\cdot\left[\boldsymbol{\nabla}\psi_j^{\boldsymbol{u}_D} + (\boldsymbol{\nabla}\psi_j^{\boldsymbol{u}_D})^T\right] \right\rangle_{\Upsilon} \qquad \boldsymbol{P}_{i,j} = \left\langle \psi_i^{\boldsymbol{u}_D}, \gamma\boldsymbol{\nabla}\psi_j^{p} \right\rangle_{\Upsilon}$$

$$\boldsymbol{B}_{i,j} = \left\langle \psi_i^{\boldsymbol{u}_D}, \psi_j^{\boldsymbol{u}_D} \right\rangle_{\Gamma_{in}} \qquad \boldsymbol{S}_{p,z,i} = \left\langle \psi_i^{\boldsymbol{u}_D}, \gamma\frac{\delta_z(\boldsymbol{r})\boldsymbol{F}_{p,z}}{|\boldsymbol{F}_{p,z}|} \right\rangle_{\Upsilon}$$

$$\boldsymbol{S}_{r,i}^{BD} = \left\langle \psi_i^{\boldsymbol{u}_D}, \frac{\boldsymbol{u}_{in}}{|\boldsymbol{u}_{in}|} \right\rangle_{\Gamma_{in}} \qquad \boldsymbol{G}_{i,j} = \left\langle \psi_i^{p}, \boldsymbol{\nabla}\cdot\psi_j^{\boldsymbol{u}_D} \right\rangle_{\Upsilon},$$

$$\overline{\overline{\boldsymbol{T}}}_{i,j,k} = \left\langle \psi_j^{\boldsymbol{u}_D}, \boldsymbol{\nabla}\cdot\left(\psi_i^{\eta_t}\left[\boldsymbol{\nabla}\psi_k^{\boldsymbol{u}_D} + (\boldsymbol{\nabla}\psi_k^{\boldsymbol{u}_D})^T\right]\right) \right\rangle_{\Upsilon} \qquad \boldsymbol{S}_{fr,z,i,j} = \left\langle \psi_i^{\boldsymbol{u}_D}, \delta_z(\boldsymbol{r})\psi_j^{\boldsymbol{F}_{fr,z}} \right\rangle_{\Upsilon}$$

$$\boldsymbol{A}_{i,j} = \left\langle \psi_i^{\boldsymbol{u}_D}, \boldsymbol{g}\psi_j^{T} \right\rangle_{\Upsilon} \qquad \boldsymbol{S}_{T,i} = \langle \psi_i^{\boldsymbol{u}_D}, \boldsymbol{g} \rangle_{\Upsilon}$$

In this context, $\delta_z(\boldsymbol{r})$ is a selection function (also known as characteristic function) which returns 1 if $\boldsymbol{r}$ is in porous medium zone z and 0 otherwise. Furthermore, The term $\Gamma(\boldsymbol{B}\boldsymbol{c}^{\boldsymbol{u}_D} - |\boldsymbol{u}_{D,in}|\boldsymbol{S}_r^{BD})$ in the reduced momentum equation is responsible for weakly enforcing the inlet boundary conditions. It contains a constant penalty factor Γ which, in practice, should be large enough to enforce the boundary condition, but not too large in order to avoid ill-conditioned reduced operators. This approach has been widely used in methods involving the weak formulation (e.g., spectral methods, finite element methods) of equations describing fluid flows [122, 123]. Tensor $\overline{\overline{\boldsymbol{T}}}$ is only used for

turbulent problems, in laminar scenarios it is a zero tensor.

Before system (4.29)-(4.30) can be solved to obtain $\boldsymbol{c}^{\boldsymbol{u}_D}$ and $\boldsymbol{c}^p$, three main questions need to be addressed:

i In this form, the two-equation ROM is not stable due to the fact that the Ladyzhenskaya-Babuska-Brezzi (LBB) condition [124] is not necessarily satisfied.

ii The coefficients of the eddy viscosity are still unknown. The computation of these coefficients from reduced forms of eddy-viscosity models is not utilized by the fluid-dynamics ROM community, therefore a data-based approach, Radial-Basis Function (RBF) interpolation is chosen in this work.

iii The coefficients of the flow resistances are still unknown. In this work, the Discrete Empirical Interpolation Method (DEIM) is used for the determination of these models based on the coefficients of the Darcy velocity.

These concerns are addressed in detail in the subsequent sections. Lastly, it is important to note that the coefficients of the temperature ($\boldsymbol{c}^T$) are fixed to the latest solution of the reduced heat transfer equation during the fluid ROM solves.

4.3.1.2 Stabilization of the Two-Equation ROM

It must be noted that as the linear combination of divergence-free snapshots, the velocity basis functions are divergence-free as well. This entails that the entries of matrix $\boldsymbol{G}$ are zero, rendering the simultaneous determination of the pressure and velocity coefficients impossible. In other words, the reduced system does not satisfy the Ladyzhenskaya-Babuska-Brezzi (LBB) condition for saddle point problems [124] which can be written as:

$$\inf_{\psi_j^p} \sup_{\psi_i^{\boldsymbol{u}_D}} \frac{\left\langle \psi_j^p, \boldsymbol{\nabla} \cdot \psi_i^{\boldsymbol{u}_D} \right\rangle}{||\psi_i^{\boldsymbol{u}_D}||_{H^1} ||\psi_j^p||_{L^2}} = \inf_{\psi_j^p} \sup_{\psi_i^{\boldsymbol{u}_D}} \frac{\boldsymbol{G}_{i,j}}{||\psi_i^{\boldsymbol{u}_D}||_{H^1} ||\psi_j^p||_{L^2}} = \beta > 0. \tag{4.34}$$

for the porous Navier-Stokes equations. Based on [67], there are two approaches to tackle this problem for cases with finite volume spatial discretization. One can either construct a Pressure

Poisson Equation at the reduced-order level and solve it together with the reduced momentum equations, or augment the velocity reduced subspace with supremizer basis functions [125]. Due to the fact that the Pressure Poisson approach yields models with high errors in the pressure fields, the supremizer stabilization has been used in this work. The essence of this method is that additional stabilizing, so called supremizer basis functions are added to the velocity space. In practice, approximate supremizer modes are computed with the following procedure [125]. First, a supremizer field, $\boldsymbol{s}$, is computed for every pressure snapshot during the offline phase by solving the following Poisson problem

$$\Delta \boldsymbol{s}_i = -\boldsymbol{\nabla} p_i, \qquad \boldsymbol{r} \in \Upsilon,\ i = 1, ..., N_s \tag{4.35}$$

$$\boldsymbol{s}_i = \boldsymbol{0}, \qquad \boldsymbol{r} \in \Gamma. \tag{4.36}$$

These supremizer fields are then saved into a supremizer snapshot matrix,

$$\boldsymbol{R}_s = [\boldsymbol{s}_1, ..., \boldsymbol{s}_{N_s}]$$

and the same POD procedure as described in Section 2.2.3 is applied to obtain the supremizer modes. Following this, the velocity basis has to be augmented using these modes in the following manner:

$$\boldsymbol{\Psi}^{\boldsymbol{u}_D} = [\psi_1^{\boldsymbol{u}_D}\ \psi_2^{\boldsymbol{u}_D}\ \cdots\ \psi_{r_{\boldsymbol{u}_D}}^{\boldsymbol{u}_D}\ \psi_1^{s}\ \cdots \psi_{r_s}^{s}]$$

A simple rule of thumb is that as many supremizer modes should be added to the velocity basis as pressure modes are used. Fewer modes may be used; however, including more supremizer modes may considerably decrease the accuracy of the velocity field. Also note that the velocity basis functions are not orthonormal after the enrichment.

4.3.1.3 Determination of the Coefficients of Eddy Viscosity

The following description is partially reprinted from [3]. In case of the two-equation ROMs and turbulent flows, an efficient way of computing the $\boldsymbol{c}^{\eta_t} = \boldsymbol{f}(\boldsymbol{c}^{\boldsymbol{u}}, \boldsymbol{c}^{p})$ closure is essential. For flows

with heat and precursor transfer, the coefficients of the eddy diffusivity, α_t, need to be computed as well. At the full-order level, the eddy viscosity and diffusivity are determined by using various eddy viscosity models. The reduction of the surrogate equations in these models is not common practice in the reduced-order modeling community due to the fact that they are highly nonlinear and their applicability can be case-dependent. Alternatively, the Radial Basis Function (RBF) interpolation approach proposed in [69, 121, 126] is used in this work. In general, RBF interpolation has been proven to be an efficient technique for multidimensional scattered data [127]. In this section, the method is demonstrated using the eddy viscosity only, however the same technique is used for the eddy diffusivity as well. In this method, the original functional dependence of the turbulent viscosity $\boldsymbol{c}^{\eta_t} = \boldsymbol{f}(\boldsymbol{c}^{\boldsymbol{u}}, \boldsymbol{c}^{p})$ is approximated by a function that depends only on the input/uncertain parameters and time $\boldsymbol{c}^{\eta_t} \approx \boldsymbol{f}(\boldsymbol{\mu}, t)$. This means that, during the snapshot generation part in the offline phase, the extended parameter vectors $\boldsymbol{\pi} = [\boldsymbol{\mu}, t]$ have to be saved for every snapshot. One can notice that time is considered to be just another parameter in this approach, which limits the applicability of the method for potential extrapolation in time.

As a first step, we compute the L^2 coordinates of the eddy viscosity snapshots. This is achieved by generating scalar products in the following form

$$c_{i,k}^{\eta_t} = \langle \eta_k, \psi_i^{\eta_t} \rangle_{\Upsilon}\,, \quad i = 1, ..., r_{\eta_t,t} \quad \text{and} \quad k = 1, ..., N_s\,. \tag{4.37}$$

In other words, $c_{i,k}^{\eta_t}$ is the L^2 coordinate of snapshot k corresponding to basis function i in the reduced subspace built for the eddy viscosity. Now, these L^2 coordinates can be interpolated using the stored $\boldsymbol{\pi}$ vectors and radial basis functions. This scheme can be written in the following form for the i-th component of $\boldsymbol{c}^{\eta_t}$:

$$c_i^{\eta_t}(\boldsymbol{\pi}^*) = \sum_{k=1}^{N_s} \xi(||\boldsymbol{\pi}^* - \boldsymbol{\pi}_k||_{L^2}) w_{i,k}, \tag{4.38}$$

where $\boldsymbol{\pi}^*$ is a new extended parameter vector sample for which $c_{t,i}^{\eta}$ needs to be determined. Moreover, ξ denotes the radial basis functions with w weights. In this work Gaussian radial basis

functions are used, which can be defined as

$$\xi(r_{i,j}) = e^{-c_a r_{i,j}^2}, \tag{4.39}$$

with an attenuation parameter c_a and $r_{i,j} = ||\boldsymbol{\pi}_i - \boldsymbol{\pi}_j||_{L^2}$. At this point the only missing pieces are the weights used for the interpolation. These are determined for every component of $\boldsymbol{c}^{\eta_t}$ independently. For the i-th component the weights can be obtained by solving a linear system of equations where the j-th row is written as

$$\sum_{k=1}^{N_s} \xi(||\boldsymbol{\pi}_j - \boldsymbol{\pi}_k||_{L^2}) w_{i,j} = c_{i,k}^{\eta_t}, \quad \text{with} \quad j,k = 1, ..., N_s. \tag{4.40}$$

An advantage of this method is that the computation of the L^2 coefficients of the eddy viscosity and the interpolation weights can be done during the training phase alone. After this, in the online/evaluation phase, when $\boldsymbol{\pi}^*$ changes, only Eq. (4.38) has to be evaluated. However, this method exhibits weaknesses on two fronts. First, if the number of snapshots is large, evaluating Eq. (4.38) at each time step can be still considerable compared to the other operations at reduced level. Second, this method cannot be used for extrapolation, meaning that $\boldsymbol{\pi}^*$ should be within the bounds of the training set.

Finally, it is important to note that in multi-dimensional parameter space, the accuracy of this method can be improved if the interpolation is carried out on a reference domain where the distances between different samples do not vary considerably. The sample points for the numerical examples in this paper are considered to be the nodes of an orthogonal grid. This grid is presented on the left side of Figure 4.1. Depending on the absolute values of uncertain parameters, the distances in one specific direction can be considerably larger than in others. Due to the fact that in this work the attenuation parameter in the radial basis function is considered to be the same for every direction and interpolation point, this may have a negative effect on the accuracy. Therefore, the original parameter grid is mapped onto a uniform grid. This ensures that the variation of the solution in every direction has the same importance. One can think about this mapping as a combi-

nation of scaling and translation in each direction. The linear mapping between this reference grid and the original one is also presented in Figure 4.1.

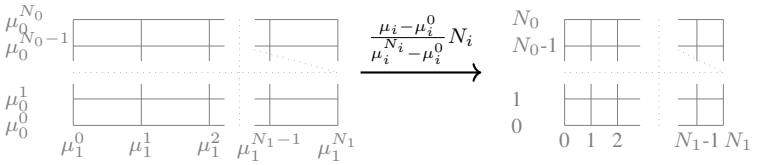


Figure 4.1: Transformation of the parameter space used for the radial basis function interpolation. Reprinted from [3]. (left: original orthogonal grid, right: transformed/projected grid.)

4.3.1.4 Determination of the Coefficients of the Flow Resistance

The coefficients for every porous medium zone, $\boldsymbol{c}_z^{\boldsymbol{F}_{fr}}$, are computed using the DEIM procedure described in Section 2.2.4.2. In this work, the correlation function described in Eq. (3.16) is used, therefore the coefficients are determined as:

$$\boldsymbol{c}_z^{\boldsymbol{F}_{fr}} = \left(\boldsymbol{P}_z^T \boldsymbol{\Psi}^{\boldsymbol{F}_{fr,z}}\right)^{-1} \boldsymbol{P}_z^T \boldsymbol{F}_{fr}(\boldsymbol{\Psi}^{\boldsymbol{u}_D} \boldsymbol{c}^{\boldsymbol{u}_D}), \tag{4.41}$$

where $\boldsymbol{P}_z^T$ is the matrix which selects the interpolation points in porous medium zone z.

4.3.1.5 One-Equation Reduced-Order Model

The one equation ROM has been first introduced for finite volume discretization in [68] and has been previously referred to as POD-FV-ROM. In this scenario, every approximated fluid flow-related field (except the flow resistance) is assumed to have the same coefficients, therefore it is enough to reduce and solve the momentum equation:

$$\Big\langle \psi_i^{\boldsymbol{u}_D},\ \frac{\partial \rho \widetilde{\boldsymbol{u}}_D}{\partial t} + \frac{1}{\gamma}\boldsymbol{\nabla}\cdot(\rho \widetilde{\boldsymbol{u}}_D \otimes \widetilde{\boldsymbol{u}}_D) - \boldsymbol{\nabla}\cdot\Big((\eta+\widetilde{\eta})\left[\boldsymbol{\nabla}\widetilde{\boldsymbol{u}}_D + (\boldsymbol{\nabla}\widetilde{\boldsymbol{u}}_D)^T\right]\Big)$$
$$+\gamma\boldsymbol{\nabla}\widetilde{p} - \gamma\boldsymbol{F}_p - \gamma\widetilde{\boldsymbol{F}}_{fr,} - \gamma\rho\boldsymbol{g}\beta_{th}(\widetilde{T} - T_{ref})\Big\rangle_{\Upsilon} = 0, \quad i = 1, ..., r_{\boldsymbol{u}_D}, \tag{4.42}$$

As a next step, the approximate fields are expressed as the linear combinations of the corresponding basis functions and the operators in Eq. (4.42) are reduced to arrive to following equation:

$$\rho \boldsymbol{M}\dot{\boldsymbol{c}}^{\boldsymbol{u}_D} + \rho \boldsymbol{c}^{\boldsymbol{u}_D,T}\overline{\overline{\boldsymbol{C}}}\boldsymbol{c}^{\boldsymbol{u}_D} - \boldsymbol{c}^{\boldsymbol{u}_D,T}\overline{\overline{\boldsymbol{T}}}\boldsymbol{c}^{\boldsymbol{u}_D} - \eta \boldsymbol{D}\boldsymbol{c}^{\boldsymbol{u}_D} + \boldsymbol{P}\boldsymbol{c}^{\boldsymbol{u}_D} + \Gamma(\boldsymbol{B}\boldsymbol{c}^{\boldsymbol{u}_D} - |\boldsymbol{u}_{D,in}|\boldsymbol{S}_r^{BD}) - \sum_{z=1}^{Z}\left(|\boldsymbol{F}_{p,z}|\boldsymbol{S}_{p,z} - \boldsymbol{S}_{fr,z}\boldsymbol{c}_z^{\boldsymbol{F}_{fr}}\right) - \rho\beta_{th}(\boldsymbol{A}\boldsymbol{c}^T - T_{ref}\boldsymbol{S}_T) = 0, \quad (4.43)$$

where the matrices and vectors are computed using the formulas described in Section 4.3.1.1 with the basis functions determined as in Eq. (4.19). The coefficients for flow resistances are computed using the method in Section 4.3.1.4.

4.3.1.6 Notes on Accuracy

Before moving on to the neutronics equations, it must be mentioned that the accuracy of both fluid ROM methods can be limited. In case of the two-equation ROM, this limitation comes from two main sources. Part of it comes from the inclusion of the supremizer modes. The presence of these modes is necessary for stabilization purposes, but they do not represent any physics and their overuse can even decrease the accuracy in velocity and, in certain cases, in pressure as well. The second part of the discrepancy between the two-equation ROM and the FOM comes from the fact that PIMPLE iteration is used to solve Eqs. (3.6)-(3.5) in the FOM, which requires the construction of the numerical approximation of a Pressure Poisson equation in `OpenFOAM`© [116]. This means that the two-equation ROM presented in this paper is not entirely numerically consistent with the FOM. Since only the momentum equation is used in case of the one-equation method, we have this numerical consistency. However, the physics-based approximation may not apply in certain scenarios, which can result in considerable errors. For clean fluid, a comparison of these techniques is available in [3] where these limitations are discussed in great detail with the help of two numerical examples.

Aside from the previously mentioned factors, there can be an additional contributor to the error between the the fluid ROMs and the FOM. The origin of this error is the advection term, discussed

in Section 3.2, which can be discretized as:

$$\int_{\Upsilon_P} \boldsymbol{\nabla} \cdot (\theta \rho \boldsymbol{u}) dV = \int_{\Gamma_P} \theta \rho \boldsymbol{u} \cdot \boldsymbol{n} dS \approx \sum_f \theta_f \rho_f \boldsymbol{u}_f \cdot \boldsymbol{S}_f, \tag{4.44}$$

saying that the scheme requires the value of variable θ on the faces of a given cell Υ_P. However, in upwind discretization schemes the value of θ on the face depends on the velocity $\boldsymbol{u}_f$ on the same face:

$$\theta_f \approx \begin{cases} \theta_{\text{own}} & \text{if} \quad \boldsymbol{u}_f \cdot \boldsymbol{S}_f > 0 \\ \theta_{\text{neighbor}} & \text{if} \quad \boldsymbol{u}_f \cdot \boldsymbol{S}_f < 0, \end{cases}$$

During the reduction of the advection terms with a velocity approximated by the sum of basis functions, the discretized operator in one cell will become:

$$\sum_i \theta_f \rho_f \sum_{j=1}^{r_{\boldsymbol{u}}} \psi_{j,f}^{\boldsymbol{u}} \boldsymbol{c}^{\boldsymbol{u}} \cdot \boldsymbol{S}_f. \tag{4.45}$$

The issue in this case is that these integrals are evaluated separately for each basis function used for $\boldsymbol{u}$, meaning that depending on the current basis function, a different value of θ_f is used. This means that the discretization scheme for the individual modes will not necessarily be the same as the one with the reconstructed velocity. This only causes problems for mainly first order upwind schemes and is still the topic of ongoing research. For higher order schemes, where the values are interpolated to the faces or for very fine meshes, this discrepancy can be negligible. Nevertheless, most of the models used for real-life applications utilize first order upwind schemes with relatively coarse mesh meaning that this issue can be a considerable contributor to errors between the ROMs and the FOM.

4.3.2 Reduced Neutronics Equations

As already discussed in Section 4.2.3, a group-wise reduction is used in this work which involves the generation of bases for group fluxes and precursor concentrations. This section reviews the generation of reduced equations for these field variables together with the treatment of the

temperature- and density-dependent cross sections at reduced-order level.

4.3.2.1 Reduced Equations for the Group Fluxes

Using the approximations in Section 4.2.3 for the group fluxes and precursor concentrations, the reduced equations for the scalar flux in group g can be derived from the following weak form:

$$\left\langle \psi_k^{\phi_g}, \frac{1}{v_g}\frac{\partial\tilde{\phi}_g}{\partial t} - \boldsymbol{\nabla}\cdot\left[D_g\boldsymbol{\nabla}\tilde{\phi}_g\right] + \Sigma_{t,g}\tilde{\phi}_g - \frac{(1-\beta)\chi_{p,g}}{\mathrm{k_{eff}}}\sum_{g'=1}^{G_e}\nu_{g'}\Sigma_{f,g'}\tilde{\phi}_{g'} - \sum_{g'=1}^{G_e}\Sigma_{s,g'\to g}\tilde{\phi}_{g'} - \chi_{d,g}\sum_{i=1}^{G_d}\lambda_i\gamma\tilde{C}_i^*\right\rangle_\Upsilon = 0, \quad k = 1, ..., r_{\phi_g} \tag{4.46}$$

For the time being, the temperature and density dependence of the cross sections is neglected, temperature-dependent scenarios will be discussed separately, in Section 4.3.2.5. As a next step, the expansions in Section 4.2.3 are plugged into Eq. (4.46), and the reduced equation for the scalar flux in group g can be expressed as:

$$\left(\sum_{z=1}^{N_z}\frac{1}{v_g^z}\boldsymbol{M}_{g,g}^z\right)\dot{\boldsymbol{c}}^{\phi_g} + \left(\sum_{z=1}^{N_z} -D_g^z\boldsymbol{K}_g^z + \Sigma_{t,g}^z\boldsymbol{M}_{g,g}^z\right)\boldsymbol{c}^{\phi_g} = \frac{1}{\mathrm{k_{eff}^r}}\sum_{g'=1}^{G_e}\left(\sum_{z=1}^{N_z}(1-\beta^z)\chi_{p,g}^z\nu_{g'}^z\Sigma_{f,g'}^z\boldsymbol{M}_{g,g'}^z\right)\boldsymbol{c}^{\phi_{g'}} + \sum_{g'=1}^{G_e}\left(\sum_{z=1}^{N_z}\Sigma_{s,g'\to g}^z\boldsymbol{M}_{g,g'}^z\right)\boldsymbol{c}^{\phi_{g'}} + \sum_{i=1}^{G_d}\left(\sum_{z=1}^{N_z}\chi_{d,g}^z\lambda_i^z\boldsymbol{P}_{g,i}^z\right)\boldsymbol{c}^{C_i^*} \tag{4.47}$$

which assumes that the cross sections are region-wise constant with the index z denoting the corresponding material region. For a similar, but considerably simpler example, see Chapter 2. Using the characteristic function $\delta_z(\boldsymbol{r})$ which returns 1 if $\boldsymbol{r}$ is in material region z and 0 otherwise, the elements of the operators in the reduced flux equation can be expressed as

$$\left(\boldsymbol{M}_{g,g'}^z\right)_{i,j} = \left\langle \psi_i^{\phi_g}\ ,\ \delta_z(\boldsymbol{r})\psi_j^{\phi_{g'}}\right\rangle_\Upsilon, \qquad \left(\boldsymbol{K}_g^z\right)_{i,j} = \left\langle \psi_i^{\phi_g}\ ,\ \boldsymbol{\nabla}\cdot\left(\delta_z(\boldsymbol{r})\boldsymbol{\nabla}\psi_j^{\phi_g}\right)\right\rangle_\Upsilon, \tag{4.48}$$

$$\left(\boldsymbol{P}_{g,k}^z\right)_{i,j} = \left\langle \psi_i^{\phi_g}\ ,\ \delta_z(\boldsymbol{r})\gamma\psi_j^{C_k^*}\right\rangle_\Upsilon. \tag{4.49}$$

These operators are the same for the reduced eigenvalue equations as well. However, in such scenarios, it has to be ensured that the coefficients of the eigenvectors are normalized in a way

which is consistent with the normalization of the flux at full-order level. As discussed in Eq. (3.35), the eigenvectors of the FOM are normalized to ensure a pre-defined power output:

$$P_{th} = \int_{\Upsilon} \int_{0}^{\infty} \Sigma_p(\boldsymbol{r}, E)\phi(\boldsymbol{r}, E) dE d\Upsilon \approx \sum_{g=1}^{G_e} \langle \Sigma_{p,g}\phi^g \rangle_{\Upsilon} \approx \sum_{g=1}^{G_e} \sum_{i=1}^{r_{\phi_g}} \left\langle \Sigma_{p,g}\psi_i^{\phi_g} \right\rangle_{\Upsilon} c_i^{\phi_g}, \tag{4.50}$$

which ensures that the coefficients $c_i^{\phi_g}$ are normalized in a way that the reconstructed eigenvector is directly comparable with the one obtained from the FOM.

Before moving on to the reduced equations for the precursors concentrations, the treatment of the boundary conditions is addressed. The formulation in Eq. (4.47) works for only specific boundary conditions like zero value or reflective. This is due to the fact that the basis functions computed using snapshots with homogeneous Neumann or Dirichlet conditions on the boundaries conserve these properties. However, in case of homogeneous Robin boundary conditions, for example, this is not satisfied. For such cases, the computation of $\boldsymbol{K}_g^z$ is slightly different. To discuss this difference we first have to understand how the differential operators are approximated in the OpenFoam® framework. As already described in Section 3.2, the integral of the Laplacian operator in the neutron diffusion equation is approximated in a single cell as:

$$\int_{\Upsilon_P} \boldsymbol{\nabla} \cdot (D_g \boldsymbol{\nabla} \phi_g)\, dV \approx \sum_f D_{g,f} \boldsymbol{\nabla} \phi_{g,f} \cdot \boldsymbol{S}_f, \tag{4.51}$$

which is the weighted sum of the surface normal gradients. Therefore, the Laplacian is approximated within cell P as a constant (still FVM) by simply dividing this value by the volume of the cell (V_P):

$$[\boldsymbol{\nabla} \cdot (D_g \boldsymbol{\nabla} \phi_g)]_P \approx \frac{\sum_f D_{g,f} \boldsymbol{\nabla} \phi_{g,f} \cdot \boldsymbol{S}_f}{V_P}. \tag{4.52}$$

If the cell is on the boundary, this expression can be split into three components:

$$\frac{\sum_f D_{g,f} \boldsymbol{\nabla} \phi_{g,f} \cdot \boldsymbol{S}_f}{V_P} = \frac{\sum_f^{N_{\text{int}}} D_{g,f} \boldsymbol{\nabla} \phi_{g,f} \cdot \boldsymbol{S}_f}{V_P} + \frac{\sum_f^{N_{\text{ext}}} C_{\text{int}}^{\text{BC}} \phi_{g,f} \boldsymbol{S}_f}{V_P} + \frac{\sum_f^{N_{\text{ext}}} C_{\text{ext}}^{\text{BC}} \boldsymbol{S}_f}{V_P}, \tag{4.53}$$

where N_{int} and N_{ext} are the internal and external (where BC is applied) faces of the cell, while $C_{\text{int}}^{\text{BC}}$ and $C_{\text{ext}}^{\text{BC}}$ are the boundary condition-dependent coefficients which add an additional term on the diagonal and to the source term of the linear system, respectively. In case of homogeneous Robin boundary conditions $C_{\text{ext}}^{\text{BC}} = 0$ and $C_{\text{int}}^{\text{BC}} = \gamma_r$, meaning that the last term does not have to be computed. In case of homogeneous Neumann conditions both $C_{\text{ext}}^{\text{BC}}$ and $C_{\text{int}}^{\text{BC}}$ are zero. In case of homogeneous Dirichlet conditions, however $C_{\text{ext}}^{\text{BC}} = 0$, while $C_{\text{int}}^{\text{BC}} = \frac{1}{h}$, where h is the distance between the center of the boundary face and the cell center. If this is carried out for every cell, the reduced diffusion term in Eq. (4.47) splits into three additional terms:

$$\left(\sum_{z=1}^{N_z} -D_g^z \boldsymbol{K}_g^z\right) \boldsymbol{c}^{\phi_g} + \left(\sum_{b=1}^{N_{\text{BC}}} \boldsymbol{K}_g^{\text{int},b}\right) \boldsymbol{c}^{\phi_g} + \left(\sum_{b=1}^{N_{\text{BC}}} \boldsymbol{k}_g^{\text{ext},b}\right), \tag{4.54}$$

where the elements of the modified reduced terms can be computed using the reduced-basis approximates together with a Galerkin projection similarly to the original case. However, in this scenario there are additional terms corresponding to the potential contributions to the system matrix $\left(\boldsymbol{K}_g^{\text{int},b}\right)$ and source vector $\left(\boldsymbol{k}_g^{\text{ext},b}\right)$ coming from the boundary conditions. This practice has been followed for every differential operator whenever necessary in the following subproblems as well. In case of the fluid dynamics subproblem, a penalty treatment was chosen to enforce non-zero Dirichlet Boundary conditions.

4.3.2.2 Reduced Equations for the Precursor Concentrations

Similarly to the flux equations, the reduced form of the precursor equation for group i can be derived from the following weak form

$$\left\langle \psi_k^{C_i^*}, \frac{\partial \widetilde{C}_i^*}{\partial t} + \boldsymbol{\nabla} \cdot \left[\boldsymbol{u}_D \widetilde{C}_i^*\right] - \boldsymbol{\nabla} \cdot \left(\left[\frac{\alpha_l}{\rho} + \frac{\alpha_t}{\rho}\right] \boldsymbol{\nabla} \widetilde{C}_i^*\right) - \frac{\beta_i}{\text{k}_{\text{eff}}} \sum_{g'=1}^{G_e} \nu_{g'} \Sigma_{f,g'} \widetilde{\phi}_{g'} + \lambda_i \widetilde{C}_i^* \gamma \right\rangle_{\Upsilon} = 0, \quad k = 1, ..., r_{C_i^*} \tag{4.55}$$

where the approximations in Eq. (4.26) and (4.25) together with Eq. (4.14) are used to generate the reduced operators as:

$$\left(\sum_{z=1}^{N_z} \boldsymbol{M}_i^z\right) \dot{\boldsymbol{c}}^{C_i^*} + \boldsymbol{c}^{\boldsymbol{u}_D} \overline{\overline{\boldsymbol{A}}}_i \boldsymbol{c}^{C_i^*} - \frac{\alpha_l}{\rho} \boldsymbol{K}_i \boldsymbol{c}^{C_i^*} - \boldsymbol{c}^{\alpha_t} \frac{1}{\rho} \overline{\overline{\boldsymbol{D}}}_i \boldsymbol{c}^{C_i^*} = - \left(\sum_{z=1}^{N_z} \lambda_i^z \boldsymbol{M}_i^z\right) \boldsymbol{c}^{C_i^*} + \frac{1}{\mathrm{k}_{\mathrm{eff}}^r} \sum_{g'=1}^{G_e} \left(\sum_{z=1}^{N_z} \beta^z \nu_{g'}^z \Sigma_{f,g'}^z \boldsymbol{F}_{i,g'}^z\right) \boldsymbol{c}^{\phi_{g'}}, \quad (4.56)$$

where region-wise constant neutronics properties are assumed. The elements of the matrices and tensors in the equation above can be computed using the characteristic function $\delta_z(\boldsymbol{r})$ as

$$(\boldsymbol{M}_k^z)_{i,j} = \left\langle \psi_i^{C_k^*} \, , \, \delta_z(\boldsymbol{r}) \psi_j^{C_k^*} \right\rangle_\Upsilon , \qquad \left(\boldsymbol{F}_{k,g}^z\right)_{i,j} = \left\langle \psi_i^{C_k^*} \, , \, \delta_z(\boldsymbol{r}) \psi_j^{\phi_g} \right\rangle_\Upsilon , \quad (4.57)$$

$$(\boldsymbol{K}_l)_{i,j} = \left\langle \psi_i^{C_k^*} \, , \, \boldsymbol{\nabla}^2 \psi_j^{C_k^*} \right\rangle_\Upsilon , \qquad \left(\overline{\overline{\boldsymbol{A}}}_l\right)_{i,j,k} = \left\langle \psi_i^{C_l^*} \, , \, \boldsymbol{\nabla} \cdot (\psi_k^{\boldsymbol{u}_D} \psi_j^{C_l^*}) \right\rangle_\Upsilon , \quad (4.58)$$

$$\left(\overline{\overline{\boldsymbol{D}}}_l\right)_{i,j,k} = \left\langle \psi_i^{C_k^*} \, , \, \boldsymbol{\nabla} \cdot (\psi_k^{\alpha_t} \boldsymbol{\nabla} \psi_j^{C_k^*}) \right\rangle_\Upsilon . \quad (4.59)$$

In this reduced-order model, the coefficients for the velocity and eddy diffusivity are given by solving the fluid dynamics ROM and keeping them fixed during the fixed-point iterations of the reduced neutronics subproblem.

4.3.2.3 Reduced Form of the Adjoint Problem

Similarly to the forward problem discussed above, the reduced equations for the adjoint flux and precursor concentrations can be derived by taking the inner product of Eqs. (3.37) and (3.38) with the approximate multi-group solutions and $\psi_j^{\phi_g^\dagger}$ and $\psi_k^{C_i^{*\dagger}}$ as test functions ($g = 1, .., G_e$, $i = 1, ..., G_d$, $j = 1, .., r_{\phi_g^\dagger}$, $k = 1, .., r_{C_i^{*\dagger}}$). In this section only the final forms of the equations are discussed with the assumption that the neutronics parameters are material region-wise constant.

The reduced-order equation for the adjoint flux in group g can therefore be expressed as:

$$\left(\sum_{z=1}^{N_z} -D_g^z \boldsymbol{K}_g^z + \Sigma_{t,g}^z \boldsymbol{M}_{g,g}^z\right) \boldsymbol{c}^{\phi_g^\dagger} = \frac{1}{\mathrm{k}_{\mathrm{eff}}^r} \sum_{g'=1}^{G_e} \left(\sum_{z=1}^{N_z} (1-\beta^z) \chi_{p,g'}^z \nu_g^z \Sigma_{f,g}^z \boldsymbol{M}_{g,g'}^z\right) \boldsymbol{c}^{\phi_{g'}^\dagger} + \sum_{g'=1}^{G_e} \left(\sum_{z=1}^{N_z} \Sigma_{s,g\to g'}^z \boldsymbol{M}_{g,g'}^z\right) \boldsymbol{c}^{\phi_{g'}^\dagger} + \frac{1}{\mathrm{k}_{\mathrm{eff}}^r} \sum_{i=1}^{G_d} \left(\sum_{z=1}^{N_z} \nu_g^z \Sigma_{f,g}^z \beta_i^z \boldsymbol{P}_{g,i}^z\right) \boldsymbol{c}^{C_i^{*,\dagger}} \quad (4.60)$$

where the elements of the reduced matrices can be computed as:

$$\left(\boldsymbol{M}_{g,g'}^z\right)_{i,j} = \left\langle \psi_i^{\phi_g^\dagger} \,,\, \delta_z(\boldsymbol{r}) \psi_j^{\phi_{g'}^\dagger} \right\rangle_\Upsilon, \qquad \left(\boldsymbol{K}_g^z\right)_{i,j} = \left\langle \psi_i^{\phi_g^\dagger} \,,\, \boldsymbol{\nabla} \cdot \left(\delta_z(\boldsymbol{r}) \boldsymbol{\nabla} \psi_j^{\phi_g^\dagger}\right) \right\rangle_\Upsilon, \quad (4.61)$$

$$\left(\boldsymbol{P}_{g,k}^z\right)_{i,j} = \left\langle \psi_i^{\phi_g^\dagger} \,,\, \delta_z(\boldsymbol{r}) \gamma \psi_j^{C_k^{\dagger,*}} \right\rangle_\Upsilon. \quad (4.62)$$

The corresponding adjoint precursor equations can be expressed as:

$$\boldsymbol{c}^{\boldsymbol{u}_D} \overline{\overline{\boldsymbol{A}}}_i \boldsymbol{c}^{C_i^{\dagger,*}} - \frac{\alpha_l}{\rho} \boldsymbol{K}_i \boldsymbol{c}^{C_i^{\dagger,*}} - \boldsymbol{c}^{\alpha_t} \frac{1}{\rho} \overline{\overline{\boldsymbol{D}}}_i \boldsymbol{c}^{C_i^{\dagger,*}} = -\left(\sum_{z=1}^{N_z} \lambda_i^z \boldsymbol{M}_i^z\right) \boldsymbol{c}^{C_i^{\dagger,*}} + \sum_{g'=1}^{G_e} \left(\sum_{z=1}^{N_z} \lambda_i^z \chi_{d,g',i}^z \boldsymbol{F}_{i,g'}^z\right) \boldsymbol{c}^{\phi_{g'}^\dagger}, \quad (4.63)$$

where the reduced operators are defined as:

$$\left(\boldsymbol{M}_k^z\right)_{i,j} = \left\langle \psi_i^{C_k^{\dagger,*}} \,,\, \delta_z(\boldsymbol{r}) \psi_j^{C_k^{\dagger,*}} \right\rangle_\Upsilon, \qquad \left(\boldsymbol{F}_{k,g}^z\right)_{i,j} = \left\langle \psi_i^{C_k^{\dagger,*}} \,,\, \delta_z(\boldsymbol{r}) \psi_j^{\phi_g^\dagger} \right\rangle_\Upsilon, \quad (4.64)$$

$$\left(\boldsymbol{K}_l\right)_{i,j} = \left\langle \psi_i^{C_k^{\dagger,*}} \,,\, \boldsymbol{\nabla}^2 \psi_j^{C_k^{\dagger,*}} \right\rangle_\Upsilon, \qquad \left(\overline{\overline{\boldsymbol{A}}}_l\right)_{i,j,k} = \left\langle \psi_i^{C_l^{\dagger,*}} \,,\, \boldsymbol{\nabla} \cdot (\psi_k^{\boldsymbol{u}_D} \psi_j^{C_l^{\dagger,*}}) \right\rangle_\Upsilon, \quad (4.65)$$

$$\left(\overline{\overline{\boldsymbol{D}}}_l\right)_{i,j,k} = \left\langle \psi_i^{C_k^{\dagger,*}} \,,\, \boldsymbol{\nabla} \cdot (\psi_k^{\alpha_t} \boldsymbol{\nabla} \psi_j^{C_k^{\dagger,*}}) \right\rangle_\Upsilon. \quad (4.66)$$

4.3.2.4 Determination of β_{eff}

As already mentioned, the determination of the effective delayed neutron precursor yield is crucial in Molten Salt Reactors. As we will show, it is possible to generate an approximate value for this quantity without reconstructing the approximate solutions with the ROMs. This means that it is enough to have the coefficients for the adjoint flux, $\boldsymbol{c}^{\phi_g^\dagger}$, the forward flux, $\boldsymbol{c}^{\phi_g}$, and the forward precursor concentrations, $\boldsymbol{c}^{C_i^*}$ to compute this quantity, therefore no full-order operation

is performed. This can be shown by first inserting the approximations (4.26) and (4.25) into Eq. (3.39) and using the multi-group nature of the problem as

$$\beta_{\text{eff},i} = \frac{\sum_{g=1}^{G_e} \left\langle \widetilde{\phi}_g^{\dagger} \, , \, \chi_{d,g} \lambda_i \gamma \widetilde{C}_i^{*} \right\rangle_{\Upsilon}}{\sum_{g=1}^{G_e} \left\langle \widetilde{\phi}_g^{\dagger} \, , \, \chi_{d,g} \sum_{j=0}^{G_d} \lambda_j \gamma \widetilde{C}_j^{*} + \chi_{p,g} \sum_{g'=1}^{G_e} \nu_{g'} \Sigma_{f,g'} \widetilde{\phi}_{g'} \right\rangle}, \tag{4.67}$$

which can be further simplified into the following form by assuming that the cross sections are region-wise constant:

$$\beta_{\text{eff},i} = \frac{\sum_{g=1}^{G_e} \boldsymbol{c}^{\phi_g^{\dagger},T} \left(\sum_{z=1}^{N_z} \chi_{d,g}^z \lambda_i^z \boldsymbol{A}_{g,i}^z \right) \boldsymbol{c}^{C_i^*}}{\sum_{g=1}^{G_e} \boldsymbol{c}^{\phi_g^{\dagger},T} \left[\sum_{j=0}^{G_d} \left(\sum_{z=1}^{N_z} \chi_{d,g}^z \lambda_j^z \boldsymbol{A}_{g,j}^z \right) \boldsymbol{c}^{C_j^*} + \sum_{g'=1}^{G_e} \left(\sum_{z=1}^{N_z} \chi_{p,g}^z \nu_{g'}^z \Sigma_{f,g'}^z \boldsymbol{F}_{g,g'}^z \right) \boldsymbol{c}^{\phi_{g'}} \right]}, \tag{4.68}$$

where the corresponding matrices can be generated using the characteristic function $\delta_z(\boldsymbol{r})$ and the porosity γ as

$$\left(\boldsymbol{A}_{g,l}^z\right)_{i,j} = \left\langle \psi_i^{\phi_g^{\dagger}} \, , \, \delta_z(\boldsymbol{r}) \gamma \psi_j^{C_l^*} \right\rangle_{\Upsilon} \quad \text{and} \quad \left(\boldsymbol{F}_{g,g'}^z\right)_{i,j} = \left\langle \psi_i^{\phi_g^{\dagger}} \, , \, \delta_z(\boldsymbol{r}) \psi_j^{\phi_{g'}} \right\rangle_{\Upsilon}. \tag{4.69}$$

These matrices are precomputed in the training phase and used only when β_{eff} needs to be evaluated. The total effective delayed neutron yield is then computed as the sum of the group-wise yields.

4.3.2.5 Treatment of The Neutronics Cross Sections

So far, the group constants in the neutronics equations have been considered to be material zone-wise constant. This is a good approximation for zero-power reactors, where the temperatures of the fuel, moderator, coolant and other structural elements do not change from a reference value. In power reactors, however, this is not the case and, as already stated in Section 3.2.4, the group constants in the neutronics equations are temperature- and density-dependent. Ultimately, both feedback can be traced back to the changing temperature. While the density is assumed to change

linearly with the temperature, the Doppler feedback is proportional to the logarithm or square root of the temperature, depending upon whether the nuclear reactor operates with a predominantly fast or thermal neutron spectrum. Therefore, the cross sections can be described as:

$$\Sigma(T) \approx \Sigma_{\mathrm{ref}} + \delta_{FT}\left(T^{\mathrm{aux}} - T^{\mathrm{aux}}_{\mathrm{ref}}\right) + \delta_{FD}\beta_{th}\rho\left(T - T_{\mathrm{ref}}\right), \tag{4.70}$$

where T^{aux} is an auxiliary temperature which is either the logarithm of the square root of the absolute temperature. The definitions of δ_{FT} and δ_{FD} are available in Section 3.2.4, and they are assumed to be material region-wise constant as well together with Σ_{ref}, while β_{th}, ρ and T_{ref} are global constants. In this section, we present the way the temperature dependence of the group constants are handled on a reaction term in the forward flux equation with a general cross section:

$$\Sigma_g \phi_g, \tag{4.71}$$

but it must be mentioned that this method can be (and has been) applied to every term in each expression that contained temperature dependent cross sections. In the original approach involving a Galerkin projection and temperature-independent group constants, the corresponding reduced-order term of Eq. (4.71) becomes:

$$\left(\sum_{z=1}^{N_z} \Sigma_g^z \boldsymbol{M}_{g,g}^z\right) \boldsymbol{c}^{\phi_g}, \tag{4.72}$$

where z denotes the material zone index, $\boldsymbol{M}_{g,g}^z$ the corresponding mass matrix (defined in Section 4.3.2.1) and $\boldsymbol{c}^{\phi_g}$ the coefficients of the ROM. Now, we recall that throughout the generation of reduced subspaces, basis functions of T and T^{aux} have already been computed. These basis functions will be used here to derive a temperature dependent form of Eq. (4.72). As a first step, expression (4.70) is plugged into Eq. (4.71) and the field variables are approximated as described

in Section 4.2:

$$\Sigma_g(T)\phi_g \approx \left[\Sigma_{\text{ref}} + \delta_{FT}\left(T^{\text{aux}} - T^{\text{aux}}_{\text{ref}}\right) + \delta_{FD}\beta_{th}\rho\left(T - T_{\text{ref}}\right)\right]\phi_g$$
$$\approx \left[\Sigma_{\text{ref}} + \delta_{FT}\left(\widetilde{T}^{\text{aux}} - T^{\text{aux}}_{\text{ref}}\right) + \delta_{FD}\beta_{th}\rho\left(\widetilde{T} - T_{\text{ref}}\right)\right]\widetilde{\phi}_g. \quad (4.73)$$

Next, we take the weak form with using the basis functions of ϕ_g as test functions:

$$\left\langle \psi_j^{\phi_g}\ ,\ \left[\Sigma_{\text{ref}} + \delta_{FT}\left(\widetilde{T}^{\text{aux}} - T^{\text{aux}}_{\text{ref}}\right) + \delta_{FD}\beta_{th}\rho\left(\widetilde{T} - T_{\text{ref}}\right)\right]\widetilde{\phi}_g \right\rangle_{\Upsilon},\ j = 1, ..., r_{\phi_g}. \quad (4.74)$$

These inner products are then evaluated using the basis functions of T, T^{aux} and ϕ_g, entailing that the equation can be written in the following, simplified form:

$$\left(\sum_{z=1}^{N_z} \Sigma^z_{\text{ref}}\boldsymbol{M}^z_{g,g} + \delta^z_{FT}\left(\boldsymbol{c}^{T^{\text{aux}},T}\overline{\overline{\boldsymbol{A}}}^z_{FT,g} - T^{\text{aux}}_{\text{ref}}\boldsymbol{M}^z_{g,g}\right) + \delta^z_{FD}\beta_{th}\rho\left(\boldsymbol{c}^{T,T}\overline{\overline{\boldsymbol{A}}}^z_{FD,g} - T_{\text{ref}}\boldsymbol{M}^z_{g,g}\right)\right)\boldsymbol{c}^{\phi_g}, \quad (4.75)$$

where the fact that the feedback coefficient (δ_{F*}-s) are material-wise constant is used. The elements of the reduced operators (matrices/tensors) can be computed using the $\delta_z(\boldsymbol{r})$ characteristic functions as

$$\left(\boldsymbol{M}^z_{g,g}\right)_{i,j} = \left\langle \psi_i^{\phi_g}\ ,\ \delta_z(\boldsymbol{r})\psi_j^{\phi_g}\right\rangle_{\Upsilon}, \qquad \left(\overline{\overline{\boldsymbol{A}}}^z_{FT,g}\right)_{i,j,k} = \left\langle \psi_i^{\phi_g}\ ,\ \delta_z(\boldsymbol{r})\psi_k^{T^{\text{aux}}}\psi_j^{\phi_g}\right\rangle_{\Upsilon}, \quad (4.76)$$

$$\left(\overline{\overline{\boldsymbol{A}}}^z_{FD,g}\right)_{i,j,k} = \left\langle \psi_i^{\phi_g}\ ,\ \delta_z(\boldsymbol{r})\psi_k^{T}\psi_j^{\phi_g}\right\rangle_{\Upsilon}. \quad (4.77)$$

This approach can be applied for every other term involving group constants in the equations for the forward/adjoint scalar flux and precursor concentration, the heat transfer equation, the expression for β_{eff} and the normalization of the flux in case of eigenvalue problems. Furthermore, the coefficients $\boldsymbol{c}^T$ are computed in the reduced heat transfer equation (discussed soon) while the elements of $\boldsymbol{c}^{T^{\text{aux}}}$ are determined using the DEIM method discussed in Section 2.2.4.2. The DEIM procedure computes coefficients for $\boldsymbol{c}^{T^{\text{aux}}}$ based on the actual $\boldsymbol{c}^T$.

Note that in this work, DEIM is utilized to connect the absolute temperature with the auxiliary

temperature instead of connecting the absolute temperature with every nonlinear term involving the temperature-dependent group constants. This results in savings in both computation time and memory. The method used in this work has been published in [91] for the fuel temperature feedback.

4.3.3 Reduced Heat Transfer Equation

The heat transfer equation is reduced using the approximations discussed in Section 4.2 together with considering the multi-group approach for the integrals in the heat source. Applying a Galerking projection with the basis functions of the temperature (ψ_i^T, $i = 1, ..., r_T$), we arrive to the following form:

$$\left\langle \psi_i^T \, , \, \frac{\partial \gamma \rho c_p \widetilde{T}}{\partial t} + \boldsymbol{\nabla} \cdot (\boldsymbol{u}_D \rho c_p \widetilde{T}) - \boldsymbol{\nabla} \cdot \left(\gamma \left[k_l + c_p \alpha_t \right] \boldsymbol{\nabla} \widetilde{T} \right) + h A_V (\widetilde{T} - T_{ext}) - \gamma \sum_{g=1}^{G_e} \left\langle \Sigma_{p,g} \widetilde{\phi}_g \right\rangle \right\rangle = 0. \quad (4.78)$$

Before describing the final form of the reduced heat transfer equation, we note that it is assumed that A_v, h and Σ_p are region-wise constant, while c_p, ρ and k_l are constants on the entire domain. With these in mind, the reduced heat transfer equation becomes

$$\begin{aligned} \rho c_p \boldsymbol{M}_\gamma \dot{\boldsymbol{c}}^T + \rho c_p \boldsymbol{c}^{\boldsymbol{u}_d, T} \overline{\overline{\boldsymbol{A}}} \boldsymbol{c}^T - k_l \boldsymbol{K}_l \boldsymbol{c}^T - c_p \boldsymbol{c}^{\alpha_t, T} \overline{\overline{\boldsymbol{K}}}_t \boldsymbol{c}^T = \\ - \left(\sum_{z=1}^{N_z} h^z A_V^z \boldsymbol{M}^z \right) \boldsymbol{c}^T + \left(\sum_{z=1}^{N_z} h^z A_V^z T_{\text{ext}} \boldsymbol{s}_h^z \right) + \sum_{g=1}^{G_e} \left(\sum_{z=1}^{N_z} \Sigma_{p,g} \boldsymbol{S}_{p,g}^z \right) \boldsymbol{c}^{\phi_g}, \quad (4.79) \end{aligned}$$

where the elements of the reduced operators can be computed using the following expressions

$$(\boldsymbol{M}_\gamma)_{i,j} = \left\langle \psi_i^T \, , \, \gamma \psi_j^T \right\rangle_\Upsilon , \qquad \left(\overline{\overline{\boldsymbol{A}}} \right)_{i,j,k} = \left\langle \psi_i^T \, , \, \boldsymbol{\nabla} \cdot \left(\psi_k^{\boldsymbol{u}_D} \psi_j^T \right) \right\rangle_\Upsilon , \quad (4.80)$$

$$(\boldsymbol{K}_l)_{i,j} = \left\langle \psi_i^T \, , \, \boldsymbol{\nabla} \cdot \left(\gamma \boldsymbol{\nabla} \psi_j^T \right) \right\rangle_\Upsilon , \qquad \left(\overline{\overline{\boldsymbol{K}}}_t \right)_{i,j,k} = \left\langle \psi_i^T \, , \, \boldsymbol{\nabla} \cdot \left(\gamma \psi_k^{\alpha_t} \boldsymbol{\nabla} \psi_j^T \right) \right\rangle_\Upsilon , \quad (4.81)$$

$$(\boldsymbol{M}^z)_{i,j} = \left\langle \psi_i^T \, , \, \delta_z(\boldsymbol{r}) \psi_j^T \right\rangle_\Upsilon , \qquad (\boldsymbol{s}_h^z)_i = \left\langle \psi_i^T \delta_z(\boldsymbol{r}) \right\rangle_\Upsilon , \quad (4.82)$$

$$\left(\boldsymbol{S}_{p,g}^z \right)_{i,j} = \left\langle \psi_i^T \, , \, \delta_z(\boldsymbol{r}) \gamma \psi_j^{\phi_g} \right\rangle_\Upsilon . \quad (4.83)$$

Again, characteristic function $\delta_z(\boldsymbol{r})$ is used to select the corresponding mesh domain for zone z. It must be mentioned that for the sake of simplicity, a constant power cross section was assumed in

the formulation above. For temperature-dependent cross sections, one has to use the formulation from Section 4.3.2.5. This equation is solved for $\boldsymbol{c}^T$, while $\boldsymbol{c}^{\alpha_t}$ and $\boldsymbol{c}^{\boldsymbol{u}_D}$ ensure the coupling with the fluid dynamics sub-problem and $\boldsymbol{c}^{\phi_g}$ are obtained by solving the neutronics equations.

5. REDUCED-ORDER MODEL FRAMEWORK AND ONLINE EVALUATION PHASE

The evaluation/online phase of the ROM is covered in this section. Based on Figure 2.1, this includes the assembly of the reduced equations derived in Chapter 4 together with the iterative solution of the problem and the evaluation of the Quantities of Interest (QoIs). To demonstrate the applicability of the methods discussed so far, a computational framework has been developed as an extension of finite volume method based multiphysics solver GeN-Foam. A short overview of this framework is presented as well. We conclude this chapter by discussing the error indicators and other measures used to assess the quality of the ROMs together with the approach we utilize for uncertainty quantification and sensitivity analysis throughout the numerical examples presented later in this work.

5.1 Assembling the Reduced-Order Model

As already discussed in Chapter 2, the assembly of the reduced-order equations is a fast operation, given that the full-order operators have an affine behavior in the parameters. In the formulations discussed in Chapter 4, the equations are definitely affine in the neutronics and thermophysical parameters. This means that once the reduced constituent operators are precomputed in the offline or training phase, the assembly of the ROM requires operations which involve the summation and multiplication with a scalar of small, dense matrices and vectors. Therefore, this phase is computationally cheap compared to the execution of the FOM.

5.2 Iteration Strategy

The reduced-order equations are solved using a fixed-point iteration scheme with multiple inner iterations between different sub-problems. The general iteration strategy used in this work is presented in Figure 5.1. It must be mentioned that for certain primitive solvers within the built ROM framework, this iteration scheme can be much simpler. The presented flowchart assumes that a ROM for every physics component is solved in a turbulent flow domain and these components are coupled to each other. Furthermore, we present the iteration strategy with a transient problem,

however in case of steady state scenarios, there is only one time step, from the initial guess to the final solution.

The iteration within a time step starts with determining the reduced-order coefficients of the turbulent viscosity and diffusivity. Since these are handled by RBF interpolation and depend only on time and model parameters, it is enough to do it once, at the beginning of each time step. Then, we enter the outer-iteration cycle which couples the fluid ROM, energy ROM and neutronics ROM solves. To converge the nonlinearities present in the fluid ROM system (i.e., convection term and the flow resistance), an inner cycle iteration has been implemented. It is visible that in contrast to the FOM, the solution of the ROM is not segregated, therefore does not need pressure correctors and non-orthogonal correctors. Another inner cycle exists to iterate between the fluid and energy ROMs in case of buoyant flows. This is especially important for buoyancy-driven scenarios. Another inner iteration cycle follows this in order to resolve the coupling between the energy and neutronics equations. This is only considered if the cross sections are temperature-dependent.

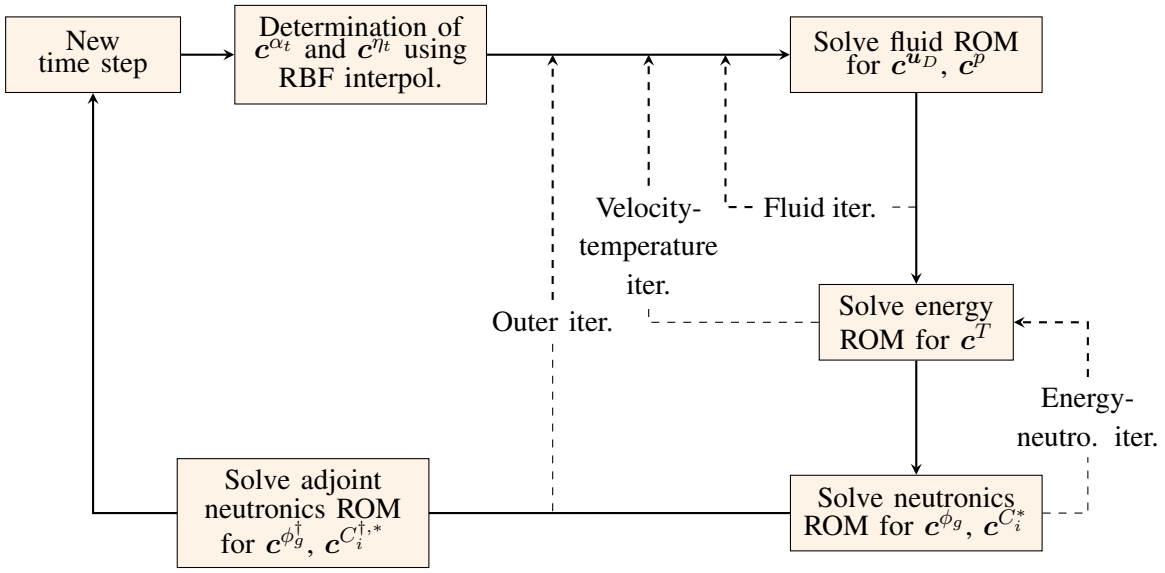


Figure 5.1: The coupling scheme of the Reduced-Order Models. (solid lines - the default coupling, dashed lines - possible inner iterations)

It is visible that the adjoint problem, if needed, is only solved at the end of the time step. The reason behind this is that it is not coupled to any of the problems. It uses the temperature and velocity field already computed using the forward problem without affecting them. Every iteration cycle has a corresponding tolerance τ which can be set in the input files. The outer iteration terminates when the maximum relative l^2 error in the physics-based reduced coefficients is below τ_{outrer}.

Lastly, the initial guesses/conditions for the ROM coefficients of different sub-problems can be obtained by projecting the corresponding initial guesses/conditions of the FOM onto the reduced sub-space:

$$\boldsymbol{M}^{\theta}_{\Upsilon}\boldsymbol{c}^{\theta} = \boldsymbol{s}_0, \tag{5.1}$$

where $\boldsymbol{M}^{\theta}_{\Upsilon}$ is a mass matrix created using the $\left\langle \psi^{\theta}_i, \psi^{\theta}_j \right\rangle_{\Upsilon}$ inner products, while $(\boldsymbol{s}_0)_i = \left\langle \psi^{\theta}_i, \theta_0 \right\rangle_{\Upsilon}$ with θ_0 denoting the FOM initial guess. It must be mentioned that in most cases, due to the construction of the basis functions, M^{θ}_{Υ} is the identity matrix. In this case it is enough to take the inner product of the FOM initial guess and the basis functions to compute the coefficients. However, in cases like the supremizer-augmented velocity treatment, the basis functions are not necessarily orthonormal, therefore the mass matrix needs to be constructed as well.

5.3 Computational Framework

To be able to demonstrate the applicability of the model-order reduction technique discussed above, a computational framework has been developed using the `OpenFOAM`© [114] finite volume library and additional routines included in GeN-Foam [93], a similarly `OpenFOAM`© based multiphysics solver developed for nuclear applications. The created ROM framework utilizes the same input files as GeN-Foam to get the model parameters, however requires an additional file containing the parameters for the control of the reduced operations. The structure of the ROM framework is presented in Figure 5.2. Since the development of the framework started as an extension of GeN-Foam, it has been named GeN-ROM[1]. It must be mentioned, however, that the ROM

[1]Is available under `https://gitlab.com/peter.german/gen-rom` (currently private, will be public when project is completed)

framework is coded in a way which allows the utilization of additional OpenFOAM© based solvers which are not included in GeN-Foam.

For the generation of snapshots, the framework uses a python script which runs the FOM with different parameters and collects the necessary snapshots. This means that practically any OpenFOAM© based solver can be used for the generation of snapshots. Beyond this point, the computation of basis functions, the generation of the operators in the ROM together with the solution of the ROM equations is coded in C++ using OpenFOAM© libraries.

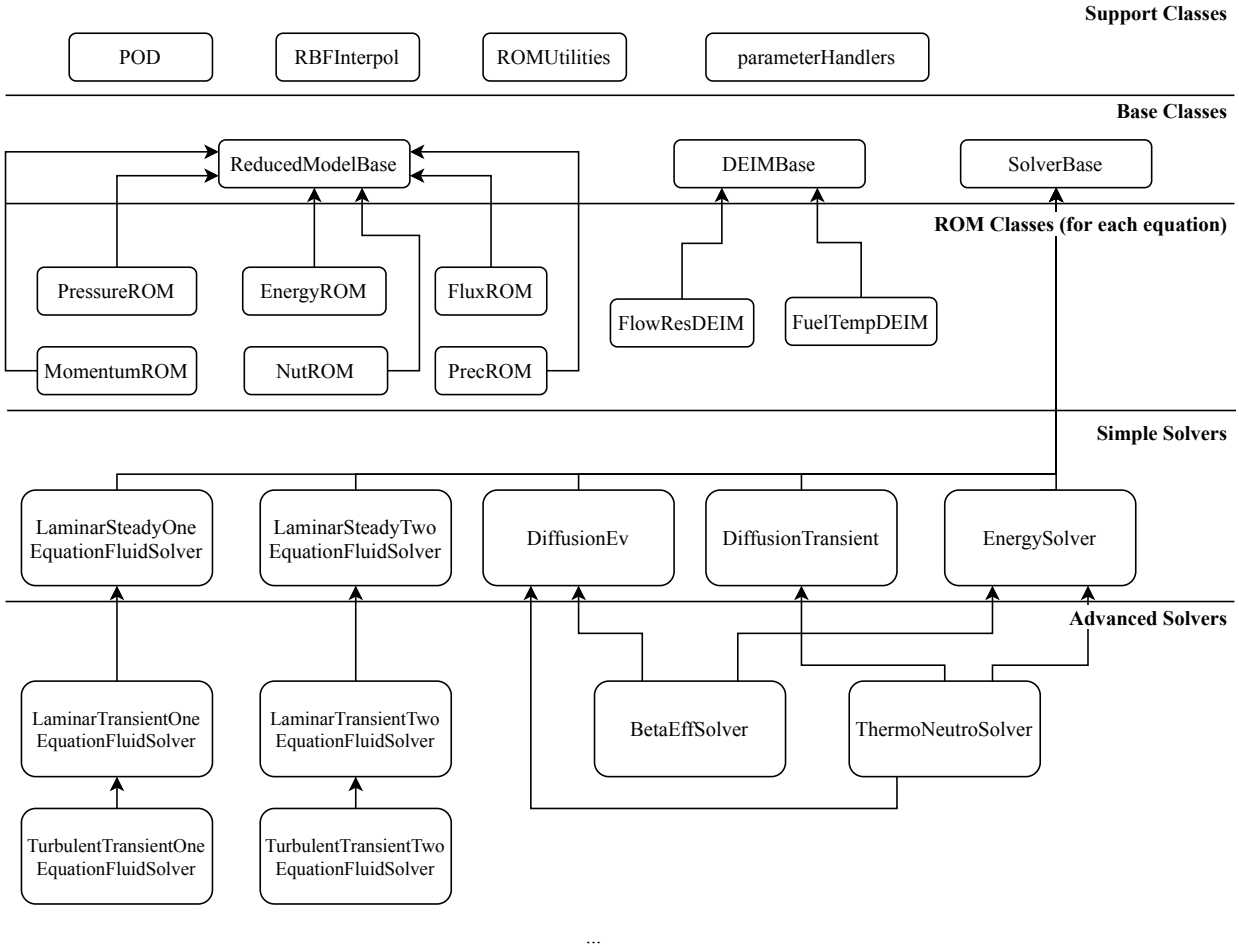


Figure 5.2: The structure of the computational framework (GeN-ROM) created. The arrows denote the inheritance paths of classes in C++.

As shown in Figure 5.2, a separate ROM class has been created for every equation that needs to be solved. The framework already has implementations for clean and porous medium incompressible fluid dynamics equations, the energy equation and neutronics transient or eigenvalue computations. These ROM classes are then used to create primitive solvers like laminar steady-state fluid dynamics solver, standalone energy or neutronics solvers. These primitive solvers can then be combined into more advanced solvers which consider the coupling between different physics. The solver needed for a given computation can be selected using the input file containing the ROM controls.

5.4 Indicators Used for ROM Evaluation

In this work, the relative global L^2 error is used as a measure of accuracy for the ROMs. This can be epressed for an arbitrary solution field θ as:

$$e_\theta = \frac{||\theta_{FOM} - \theta_{ROM}||_{L^2,\Upsilon}}{||\theta_{FOM}||_{L^2,\Upsilon}} = \frac{\sqrt{\langle (\theta_{FOM} - \theta_{ROM}), (\theta_{FOM} - \theta_{ROM}) \rangle_\Upsilon}}{\sqrt{\langle \theta_{FOM}, \theta_{FOM} \rangle_\Upsilon}}, \tag{5.2}$$

where the FOM and ROM subscripts denote the fields from the full-order model and the reconstructed fields from the ROMs, respectively. The average value of these relative errors over the time steps when the solution fields are reconstructed using the ROM is defined as:

$$\overline{e_\theta} = \frac{1}{N_{T_s}} \sum_{i=1}^{N_{T_s}} \delta\theta_i \,, \tag{5.3}$$

where N_{T_s} denotes the number of time instances for which the approximate fields are reconstructed. For simulations involving eigenvalue problems, the absolute difference between the eigenvalue of the ROM and the eigenvalue of the FOM,

$$\Delta \mathrm{k_{eff}} = \left| \mathrm{k_{eff}^{FOM}} - \mathrm{k_{eff}^{ROM}} \right| ,$$

is used to assess the quality of the ROM. For problems, where the effective delayed neutron fraction is evaluated, we use the same absolute difference:

$$\Delta\beta_{\text{eff}} = \left|\beta_{\text{eff}}^{\text{FOM}} - \beta_{\text{eff}}^{\text{ROM}}\right| ,$$

In addition, the acceleration (speed-up) achieved by the ROM compared to the FOM is defined as follows:

$$\Theta = \frac{\tau_{FOM}}{\tau_{ROM}}, \tag{5.4}$$

where τ_{FOM} and τ_{ROM} are the solution times of the FOM and the ROM, respectively.

5.5 Approach for Uncertainty Quantification and Sensitivity Analysis

Once an adequate ROM is generated, it can be used as an emulator of the FOM to carry out uncertainty propagation or to determine the sensitivity of the QoIs to input parameters. Due to the considerable reduction in computation time, the ROMs allow the utilization of Monte Carlo methods. This means that a large number (N_e) of parameter samples ($\boldsymbol{\mu}_i,\ i = 1, ..., N_e$) are generated using the assumed probability distributions, and the emulator is executed to obtain the corresponding values of the QoIs which can be used for the estimation of relevant statistical moments as:

$$\overline{\text{QoI}} = \frac{1}{N_e}\sum_{i=1}^{N_e} \text{QoI}(\boldsymbol{\mu}_i) \quad \text{and} \quad \sigma_{\text{QoI}} = \sqrt{\frac{1}{N_e - 1}\sum_{i=1}^{N_e} \left(\text{QoI}(\boldsymbol{\mu}_i) - \overline{\text{QoI}}\right)^2}, \tag{5.5}$$

where $\overline{\text{QoI}}$ estimates the mean and σ_{QoI} the standard deviation of the QoI. The confidence intervals of these estimators can also be estimated using a bootstrap (resampling with replacement) method where multiple subsamples of $\text{QoI}(\boldsymbol{\mu}_i)$ are chosen to estimate the distribution of the statistical moments themselves.

In many scenarios, the contribution of the uncertainty in different model parameters to the uncertainty in the QoI can be rather different. This is not only due to the fact that the relative uncertainty in the model parameters is different, but because the sensitivity of the QoI to changes

in the input parameter is different as well. In this work, the contributions to the uncertainty in the QoI are analyzed using Sobol Indices which can be derived by first taking the Sobol Decomposition of $\mathrm{QoI}(\boldsymbol{\mu})$ as [128]:

$$\mathrm{QoI}(\mu_1, ..., \mu_d) = \mathrm{QoI}_0 + \sum_{i=1}^{d} \mathrm{QoI}_i(\mu_i) + \sum_{i \leq i < j \leq d} \mathrm{QoI}_{i,j}(\mu_i, \mu_j) + ... + \mathrm{QoI}_{12...d}(\mu_1, ..., \mu_d) \tag{5.6}$$

with QoI_0 being the average of $\mathrm{QoI}(\boldsymbol{\mu})$ over the parameter domain $\mathcal{D}_\mu$. Furthermore, we assume that the summands in Eq. (5.6) are orthogonal with respect to the probability density function (PDF) of $\boldsymbol{\mu}$ and their mean values are zero [129]. It must be noted that the functions in the expansion can be determined in a straightforward sequential manner starting from QoI_0 and using the previously computed summands to determine the next function. For more information on the exact algorithm, the interested reader is referred to [128]. If we take the variance of $\mathrm{QoI}(\boldsymbol{\mu})$ and use the previously mentioned assumptions, we get:

$$\mathrm{Var}[\mathrm{QoI}(\boldsymbol{\mu})] = \sigma^2_{\mathrm{QoI}} = \int_{\mathcal{D}_\mu} \mathrm{QoI}^2(\boldsymbol{\mu}) d\boldsymbol{\mu} - \mathrm{QoI}^2(\boldsymbol{\mu}) = \sum_{i=1}^{d} \sigma_i^2 + \sum_{i \leq i < j \leq d} \sigma_{ij}^2 + ... + \sigma_{12...d}^2, \tag{5.7}$$

meaning that the variance of the original function can be expressed as the sum of the variances of the member functions ($\sigma^2_{i_1 i_2 ... i_s}$, $s \leq d$) in the Sobol expansion. Normalizing the equation above by σ^2_{QoI} we get the following expression:

$$1 = \sum_{i=1}^{N_\mu} S_i + \sum_{i \leq i < j \leq d} S_{ij} + ... + S_{12...d}, \tag{5.8}$$

where $S_{i_1 i_2 ... i_s}$ are called the Sobol Idices. These indices describe the relative contributions of the variances of the model parameters together with the combined effects to the overall variance in the QoI. Furthermore, the total Sobol Index with respect to model parameter μ_k ($k = 1, ..., d$) is defined as the sum of all partial sensitivity coefficients which include it.

There are multiple ways to compute estimates for these indices. A Monte Carlo sampling based approach is described by Saltelli in [130], while another, Polynomial Chaose Expanson-based (PCE) technique is discussed in [128]. In this work, we wish to reuse the sample-output combinations already used for the estimation of the statistical moments, meaning that Saltelli's method is not applicable since it requires the additional evaluation of the model with specifically

chosen parameter vectors. For this reason, the PCE-based approach has been implemented. As already introduced in Chapter 1, the PCE of our QoI can be expressed as:

$$QoI(\boldsymbol{\mu}) \approx \widehat{QoI}(\boldsymbol{\mu}) = \sum_{k}^{N_p} QoI_k \Psi_i(\boldsymbol{\mu}) = \sum_{k}^{N_p} QoI_i \prod_{j=1}^{N_\mu} \mathcal{P}_{i_j}(\mu_j), \quad \text{for all} \quad \sum_{j}^{d} i_j < N_{\text{max}}, \tag{5.9}$$

where QoI_k are the expansion coefficients for basis functions Ψ_k. In case of PCE, these basis functions can be constructed as the products of univariate polynomials, $\mathcal{P}_{i_j}$ of degree i_j. Furthermore, d denotes the number of parameters, N_p is the number of expansion terms, while N_{max} is the maximum polynomial degree allowed in the expansion. Clearly, N_p can be determined using d and N_{max}. The type of polynomial basis functions are typically chosen using the assumed distributions of the uncertain parameters. For the corresponding distribution-polynomial pairs, see [15, 16]. Since we only use uniformly distributed variables in this work, Legendre polynomials are used to build the basis. It is important to mention that similarly to the Sobol decomposition, the multivariate basis functions of the PCE are orthogonal with respect to the PDF of $\boldsymbol{\mu}$. The QoI_k expansion coefficients can be determined using multiple methods including least squares regression or spectral projection evaluated with quadrature sets or Monte Carlo methods. Since the training parameter-output data base is already given, a least squares regression algorithm is utilized in this work which minimizes $\sum_i ||QoI(\boldsymbol{\mu}_i) - \widehat{QoI}(\boldsymbol{\mu}_i)||^2,\ (i = 1, ..., N_e)$. This process has been carried out using the `openTURNS` python library [131].

Once the coefficients of the PCE are given, the Sobol Indices can be easily computed. This utilizes the fact that the basis functions of the PCE satisfy the requirements of the Sobol decomposition. In other words, by computing the PCE, we obtain the Sobol decomposition of $\widehat{QoI}(\boldsymbol{\mu})$. The only step remaining is the rearrangement of the PCE expansion terms with respect to their parameter-dependence and the computation of the corresponding Sobol Indices by [128]:

$$S^{PCE}_{i_1 i_2 \dots i_s} = \frac{1}{\sum_{i=1}^{N_p} QoI_i^2 E[\Psi_i^2]} \sum_{\alpha \in \mathcal{I}_{i_1 i_2 \dots i_s}} QoI_\alpha^2 E[\Psi_\alpha^2], \tag{5.10}$$

where $E[f]$ denotes the expectation of function f and $\mathcal{I}_{i_1 i_2 \ldots i_s}$ contains the indices of the terms in the PCE which can be grouped together to generate the $i_1 i_2 \ldots i_s$-th term in the Sobol decomposition. This means that once the expansion coefficients of the PCE are computed, the determination of the Sobol Indices becomes a simple task which involves the computation of the square integrals of polynomials and the summation of the squares of expansion coefficients.

Furthermore, the fitting of a PCE using the available data allows the determination of local sensitivity coefficients in the parameter space. These sensitivity coefficients describe the relative change in the QoI caused by the relative change in an input parameter (μ_k) at $\boldsymbol{\mu}_0$ in the parameter space as:

$$\kappa_{\mu_k}\big|_{\boldsymbol{\mu}_0} = \frac{\partial \mathrm{QoI}}{\partial \mu_k}\bigg|_{\boldsymbol{\mu}_0} \frac{\mu_{0k}}{\mathrm{QoI}(\boldsymbol{\mu}_0)} \approx \frac{\mu_{0k}}{\widehat{QoI}(\boldsymbol{\mu}_0)} \frac{\partial}{\partial \mu_k} \sum_{i}^{N_p} QoI_i \prod_{j=1}^{N_\mu} \mathcal{P}_{i_j}(\mu_{0j}) = \frac{\mu_k}{\widehat{QoI}(\boldsymbol{\mu})} \frac{\partial}{\partial \mu_k} \sum_{i}^{N_p} QoI_i \frac{\partial \mathcal{P}_{i_k}(\mu_k 0)}{\partial \mu_k} \prod_{j=1, j\neq k}^{N_\mu} \mathcal{P}_{i_j}(\mu_0 j). \quad (5.11)$$

It is visible that the computation of the sensitivity coefficient simplifies to the evaluation of polynomial functions and weighting the results by the corresponding expansion coefficients.

Lastly, we note that the final values of the Sobol Indices and the sensitivity coefficients are burdened by multiple errors in this process such as:

1. The approximation of $\mathrm{QoI}(\boldsymbol{\mu})$ by $\mathrm{QoI}^{ROM}(\boldsymbol{\mu})$
2. The approximation of $\mathrm{QoI}(\boldsymbol{\mu})^{ROM}$ by a fitted PCE which contains the error of the least squares regression together with the error caused by discarding the terms with a polynomial degree above $N_{\max}$ in the expansion.

Nevertheless, in most of the cases these errors are negligible compared to variation of the $\mathrm{QoI}(\boldsymbol{\mu})$, therefore we assume that $\widehat{QoI}$ can adequatly emulate the original system. The numerical examples in the following sections will confirm this assumption.

6. REDUCED-ORDER MODELING OF ZERO-POWER REACTORS

Following the discussion of the generation process of POD-RB-ROMs, we turn our attention to several numerical examples. In this chapter, specifically, the applicability of the devised ROM methods to zero-power steady-state and transient simulations is assessed. In these scenarios, the thermal power of the reactor is negligible, therefore, buoyancy effects and the temperature-dependence of neutronics cross sections are not considered, only the coupling between the neutronics and fluid dynamics ROMs is investigated. ROMs are generated for a 2D axisymmetric model of the Molten Salt Fast Reactor (MSFR) and examples with both laminar and turbulent flows are presented. In case of steady-state simulations, the propagation of uncertainties arising from fission cross sections, diffusion coefficients, and the pumping force is investigated. The quantities of interest, in this case, are the effective multiplication factor ($\mathrm{k_{eff}}$) and the effective delayed neutron fraction (β_{eff}). In transient scenarios, we assume that the inserted reactivity and the change in the pumping force are model parameters with the quantity of interest being the relative reactor power at the end of the transients (normalized by the initial power level).

6.1 The Full-Order Model (FOM) of the Molten Salt Fast Reactor (MSFR)

In this work, a 2D axisymmetric model of the MSFR is used as the FOM. A wedge geometry and a hexahedral mesh have been constructed using the open-source mesh generation tool SALOME [132] based on the design proposed in [133] and [134]. The mesh, presented in Figure 6.1, contains altogether $N = 16,140$ cells, grouped into four mesh zones: one for the pump (Zone P), one for the heat exchanger (Zone HX), one between the pump and the heat exchanger (Zone I) and the last one for all the remaining cells in the domain (Zone C). Since the specific design is not publicly available, the heat exchanger is treated as a true porous medium with an approximate fluid fraction of $\gamma = 0.4$. Every other region is assumed to be occupied entirely by molten salt. To avoid numerical checker-boarding close to the high-magnitude momentum sources and sinks (pump and heat exchanger), the mesh has been refined close to the interfaces of these regions. The same mesh

has been used for simulations with turbulent and laminar scenarios. Furthermore, the thickness of the mesh layers closest to the boundary have been determined in a way that allows the use of wall functions in case of turbulent simulations with nominal flow velocities.

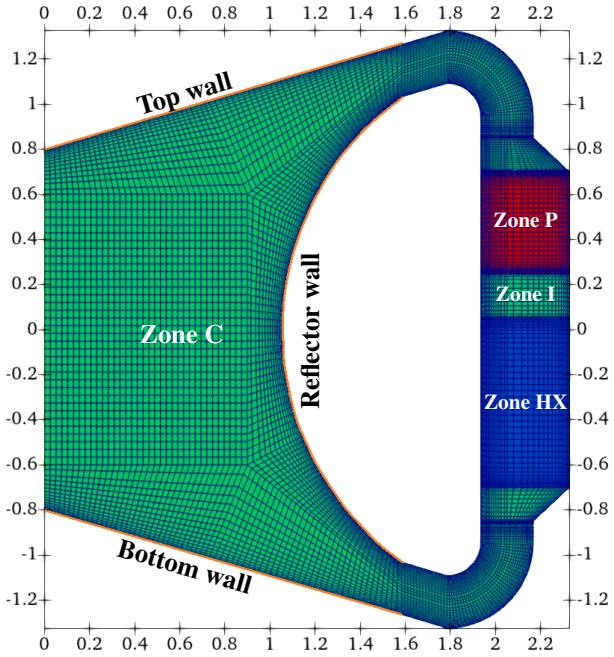


Figure 6.1: Geometry and dimensions (in m) used for the MSFR. Modified from [3]. (red: pump, blue: heat exchanger, green: clean fluid, orange: distinguished boundary segments)

In this work, both steady-state and transient cases are considered with two additional sub-cases in each category. First, a laminar scenario is assumed by keeping the velocity of the fluid close to nominal values and increasing the dynamic viscosity of the fuel salt. This is important, since decreasing the velocity while keeping the molecular viscosity at a low value would essentially decouple the two subproblems by reducing the precursor drift to a negligible level. The second case considers turbulent flows using the Reynolds-Averaged Porous Medium Navier-Stokes equations with Boussinesq eddy viscosity approximation and a porous medium $k - \epsilon$ eddy viscosity model. For more information on these models, see Chapter 3. The boundary conditions used for both the fluid dynamics and neutronics subproblems are summarized in Table 6.1. It is visible that

homogeneous Robin boundary conditions have been employed for the neutron scalar flux to take into account the effect of neutron reflectors. To simulate the axial reflection, $\gamma_r = 0.05$ has been used at the top and bottom walls, while $\gamma_r = 0.08$ has been used for the wall facing the radial reflector. For the definition of γ_r in terms of the incoming and outgoing currents, the reader is referred to [93]. Every additional boundary segment is assumed to face vacuum, therefore at these segments a $\gamma_r = 0.5$ has been utilized.

Table 6.1: The boundary conditions used for the fluid dynamics and neutronics subproblems.

Field Name	**Symbol**	**Boundary condition**
Superficial velocity	$\boldsymbol{u}_D$	$\boldsymbol{u}_D = 0$
Corrected pressure	p	$\boldsymbol{\nabla} p \cdot \boldsymbol{n} = 0$
Turbulent kinetic energy	k	wall function
Discipation rate	ϵ	wall function
Scalar flux in energy group i	ϕ_i	$-D_i \boldsymbol{\nabla} \phi_i \cdot \boldsymbol{n} = \gamma_r \phi_i$
Adjoint flux in energy group i	$\phi_i^\dagger$	$-D_i \boldsymbol{\nabla} \phi_i^\dagger \cdot \boldsymbol{n} = \gamma_r \phi_i^\dagger$
Corr. precursor concentration in group i	C_i^*	$\boldsymbol{\nabla} C_i^* \cdot \boldsymbol{n} = 0$
Corr. adjoint prec. conc. in group i	$C_i^{\dagger,*}$	$\boldsymbol{\nabla} C_i^{\dagger,*} \cdot \boldsymbol{n} = 0$

The same spatial discretization schemes have been used for both transient and steady-state simulations. The advection term in the momentum equation is discretized using a vanLeer scheme [135] which is second order in space. The advection terms in the forward and adjoint precursor equations together with those in the $k-\epsilon$ model are, however, discretized using a first order upwind scheme. The diffusion terms, on the other hand, have been treated using a linear gradient estimator (second order in space) in every equation.

Based on [7], a molten salt composition of $77.5\mathrm{LiF} - 19.95^{232}\mathrm{ThF}_4 - 2.55^{233}\mathrm{UF}_4$ (mol %)

has been chosen. The thermophysical properties of the molten salt have been determined using the data available in [7] with the exception of the laminar scenarios where the molecular viscosity has been intentionally increased. The values used for the relevant thermophysical properties are summarized in Table 6.2. The fluid is assumed to be at 900 K, which is above the melting point of the given salt composition. Due to the negligible thermal power, the flow is considered to be isothermal.

Table 6.2: The thermophysical parameters used in the fluid dynamics suproblem for laminar and turbulent scenarios.

Parameter name	**Symbol**	**Value (laminar)**	**Value (turbulent)**
Physical density	ρ	4125 $\frac{kg}{m^3}$	
Molecular viscosity	η	15 $Pa \cdot s$	0.01 $Pa \cdot s$

In both laminar and turbulent cases, the pump serves as the volumetric momentum source, while the heat exchanger obstructs the flow with a volumetric force that depends on the velocity itself. The exact form of this functional dependence is also discussed in Chapter 3. The parameters of the porous medium emulating the heat exchanger are summarized in Table 6.3.

For steady-state simulations, the magnitude of the pumping power ($|\boldsymbol{F}_p|$) is considered to be an uncertain/design parameter with a uniform distribution within the $[60\ \frac{kN}{m^3}, 100\ \frac{kN}{m^3}]$ interval. The same interval has been used for laminar and turbulent scenarios. This interval ensures that the Reynolds number in the core cavity is $225-310$ for laminar and $9.6\times10^5-1.3\times10^6$ for turbulent simulations, respectively.

The group constants in the multigroup diffusion equations in the neutronics submodule have been generated using Serpent 2 Monte Carlo particle transport code [117] from the cross section libraries available at 900 K. Altogether six energy groups and eight precursor groups have been

used. The energy group structure is presented in Table 6.4, while the default, 8-group structure of Serpent has been utilized for the precursors.

Table 6.3: The parameters of the porous medium heat exchanger used in the fluid dynamics suproblem.

Parameter name	**Symbol**	**Value**
Fluid fraction	γ	0.4 $\frac{m^3}{m^3}$
Hydraulic diam. of the heat exchanger	D_h	0.02 m
Flow res. coefficient in heat exchanger	A_{f_D}	0.687
Flow res. exponent in heat exchanger	B_{f_D}	-0.25

Table 6.4: The energy group structure used for the 2D model of the MSFR [4].

Group	**Upper bound (MeV)**	**Lower bound (MeV)**
1		2.231E-00
2	2.231E-00	4.979E-01
3	4.979E-01	2.479E-02
4	2.479E-02	5.531E-03
5	5.531E-03	2.485E-04
6	2.485E-04	

For steady-state simulations, it is assumed that the diffusion coefficients (D_i) and neutron yield times fission cross section ($\nu\Sigma_{f,i}$) are uncertain in mesh region C (which includes the core

cavity, see Figure 6.1). These coefficients are approximated with uniformly distributed random variables within the $\pm 10\%$ interval around their nominal values. This means that the problem has altogether 13 design/uncertain parameters that can be aggregated into a parameter vector: $\boldsymbol{\mu} = [D_1, ..., D_6, \nu\Sigma_{f,1}, ..., \nu\Sigma_{f,6}, |\boldsymbol{F}_p|]$. The quantities of our interest in steady-state simulations are the effective multiplication factor and the effective total delayed neutron fraction, meaning that our goal is to use our reduced-order models to approximate the $\mathrm{k_{eff}} = \mathrm{k_{eff}}(\boldsymbol{\mu})$ and $\beta_{\mathrm{eff}} = \beta_{\mathrm{eff}}(\boldsymbol{\mu})$ functions.

In case of time-dependent simulations, a reactivity driven transient is considered with the following timeline (also shown in Figure 6.2):

1. The reactor is initially critical with a volumetric pumping force of $|\boldsymbol{F}_{p,0}| = 30\frac{kN}{m^3}$. The fields at the initial time step are obtained from a steady-sate k-eigenvalue computation to ensure that the reactor is in a critical state in a sense that the production and the disappearance of the neutrons are equal. The $\mathrm{k_{eff}}$ where this is satisfied is 0.981969 for laminar and 0.981711 for turbulent simulations, respectively.

2. At T_0, an unintentional reactivity insertion occurs. The reactor becomes supercritical, but not prompt supercritical. The increase in reactivity is simulated by decreasing k_{eff} in the time-dependent multi-group neutron diffusion equations. The change in the effective multiplication factor $\delta\mathrm{k_{eff}}$ is an uncertain parameter in these scenarios varying in the [25 pcm, 55 pcm] interval with uniform distribution. Note that in the time-dependent scenario $\mathrm{k_{eff}}$ is just a factor which can modify the fission cross section times the fission neutron yield ($\nu\Sigma_f$) across the whole reactor. The decrease in this factor leads to an increase in the fission neutron yield rendering the reactor supercitical in a sense that the production of neutrons will be higher than the disappearance. An example for this could be the accidental, relatively homogeneous injection of additional fissile material into the fuel salt.

3. Between T_1 and T_2, the pumping power is increased to $|\boldsymbol{F}_p| = F_p$ with a linear ramp. This decreases the reactivity by transporting an increasing fraction of delayed neutron precursors

from the high importance zones of the reactor core. The magnitude of the new pumping force is considered to be a design/uncertain parameter with a uniform distribution in the [60, 100] $\frac{kN}{m^3}$ interval for both laminar and turbulent scenarios. In the subsequent chapters, instead of using the absolute value of the pumping force we use a relative pumping power, denoted by $a_{F_p} = \frac{F_p}{|\boldsymbol{F}_{p,0}|}$, which varies in the $[2.0,\ 3.\dot{3}]$ interval.

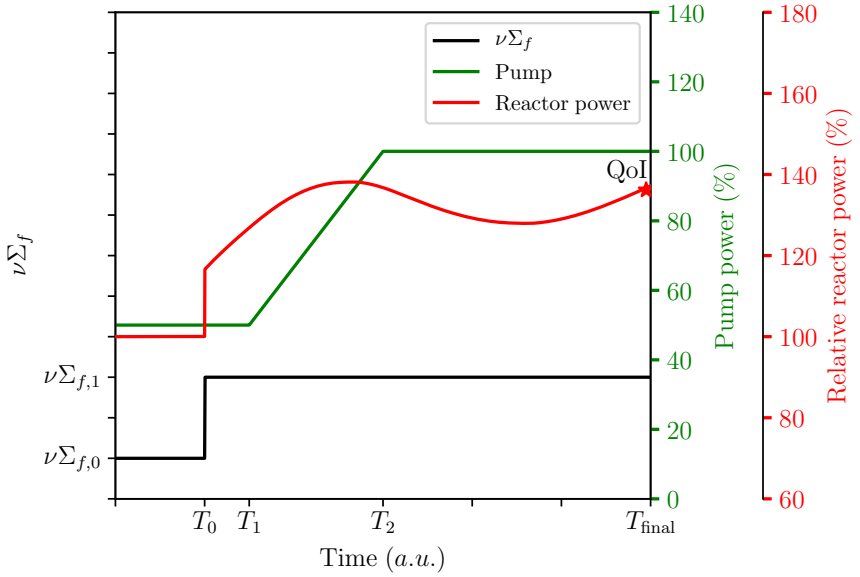


Figure 6.2: Timeline of the transients in case of simulations with zero-power MSFR.

The described transient has been specifically selected to investigate whether the reduced-order models can reconstruct: (i) the prompt jump following the initial reactivity insertion and (ii) the oscillation in power caused by the drift of the delayed neutron precursors. To ensure that the initial jump in power is resolved, an adaptive time-stepping technique is used to modify the next time step based on the relative power change:

$$\Delta t_{\text{new}} = \min\left(\min\left(\frac{C_{P_{th}}}{\frac{|P_{th}-P_{th}^{old}|}{P_{th}^{old}}}, 1 + 0.1\frac{C_{P_{th}}}{\frac{|P_{th}-P_{th}^{old}|}{P_{th}^{old}}}\right), 1.2\right)\Delta t_{\text{old}}, \tag{6.1}$$

where $C_{P_{th}}$ factor describes the maximum allowed relative power change in each time step. The initial time step has been set to 10^{-6} s, while the maximum allowed time step is 0.001 s. As described in Chapter 3, backward Euler discretization scheme is used to integrate the equations in time.

In the transient scenario, only the jump in $\mathrm{k_{eff}}$ and the final relative pumping power are considered to be uncertain, giving a parameter vector of $\boldsymbol{\mu} = [\delta \mathrm{k_{eff}}, a_{F_p}]$. The quantity of interest in this case is the relative reactor power $\left(P_{th}^{\mathrm{rel}}(t) = \frac{P_{th}(t)}{P_{th}(T_0)}\right)$ attained at $t = T_{\mathrm{final}}$. Therefore, in these scenarios we attempt to approximate the $P_{th}^{\mathrm{rel}}(T_{\mathrm{final}}; \boldsymbol{\mu})$ function with our reduced-order model.

6.2 Reduced-Order Models for Steady-State Simulations

The construction of the reduced-order models for steady-state simulations begins with the collection of snapshots at different states of the system. Due to the computationally expensive FOM, only 20 parameter vectors have been sampled in the 13 dimensional parameter space using Latin Hypercube Sampling (LHS). These parameter samples are fed to the FOM and the steady-state solution fields are saved into corresponding snapshot matrices. As a next step, the spatial basis functions are extracted from these snapshot matrices using POD. For this, correlation matrices are generated from the snapshot matrices and the eigen-decompositions of the correlation matrices are obtained. The decay of the eigenvalues on the diagonal of the eigenvalue matrix is a good indicator of the error in approximating (by an L^2 projection) the snapshots using the linear combination of the generated POD modes. The decay in the normalized (using the largest eigenvalue) eigenvalues of the correlation matrices is presented in Figure 6.3 for laminar and in Figure 6.4 for turbulent simulations. It must be mentioned, however, that this error is not necessarily indicative of the expected L^2 difference between the fields of the ROM and the FOM, since that depends on additional factors such as the ability of the snapshots to represent the solution manifold and the other approximations introduced at reduced-order level.

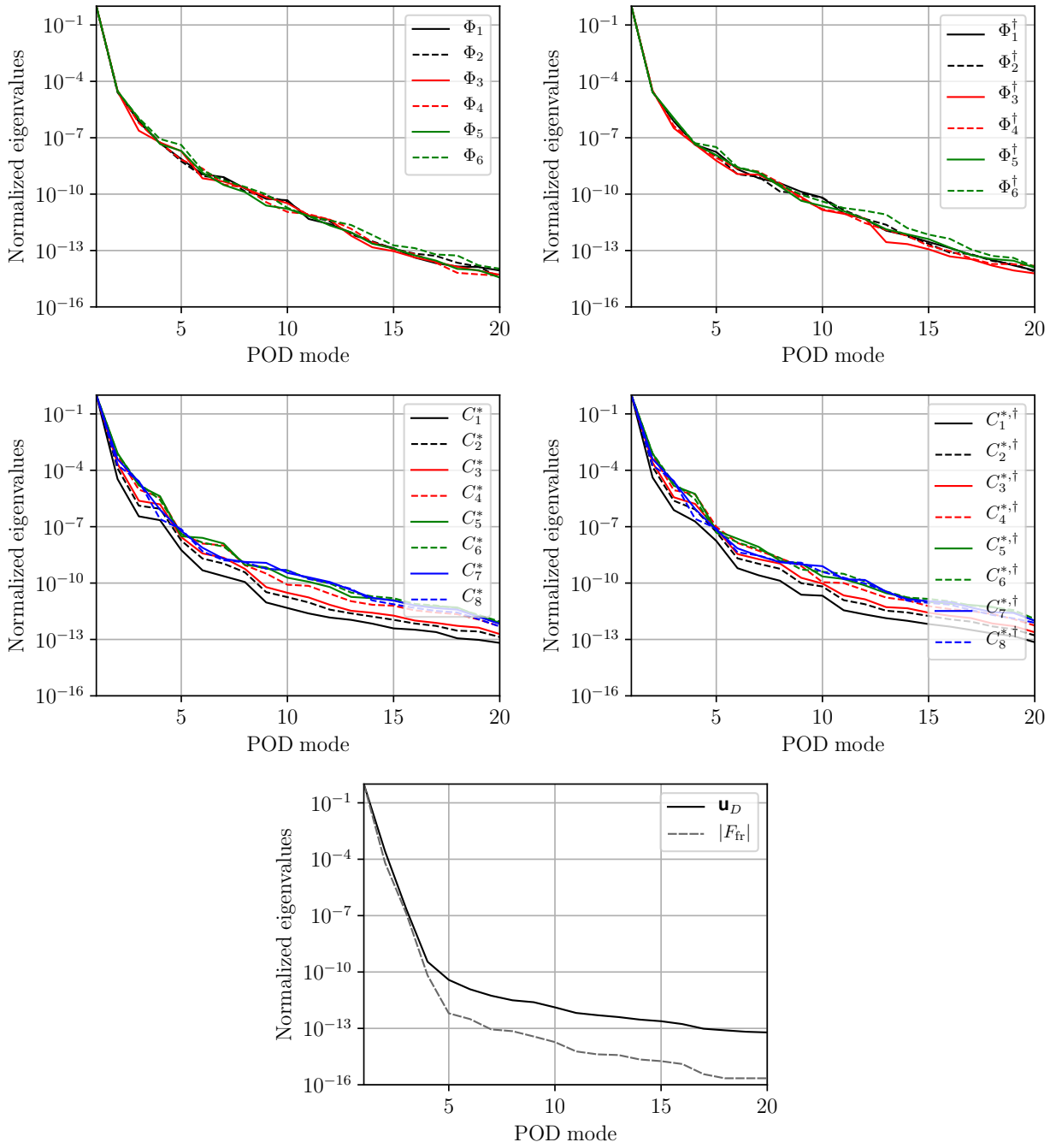


Figure 6.3: Scree plots of the normalized eigenvalues of the correlation matrices built from the corresponding snapshot matrices obtained from steady-state simulations with a laminar fluid model.

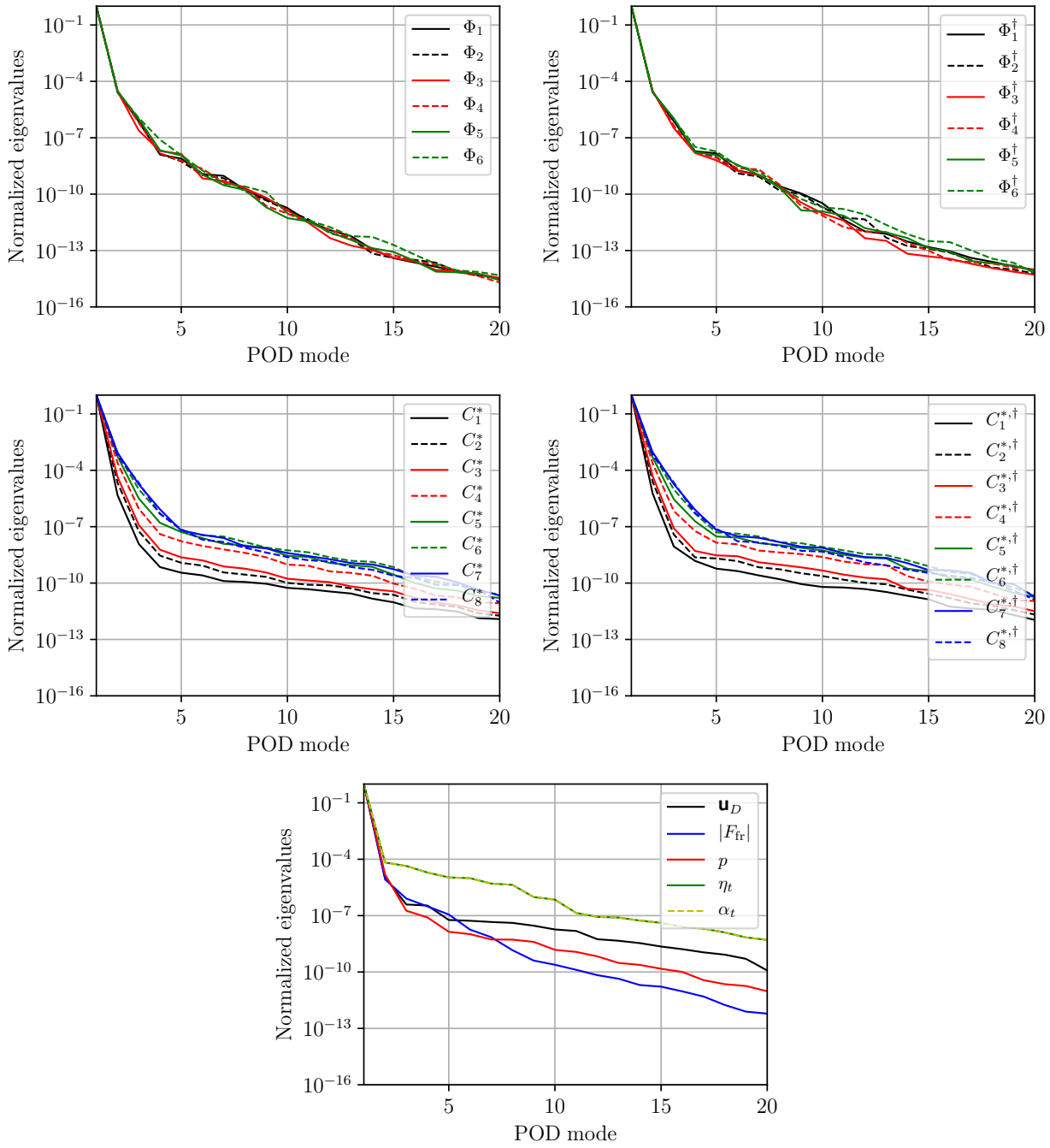


Figure 6.4: Scree plots of the normalized eigenvalues of the correlation matrices built from the corresponding snapshot matrices obtained from steady-state simulations with a turbulent fluid model.

Nevertheless, by observing the steepness of the decay curves we can anticipate whether the given field can be approximated in a lower-dimensional subspace. The steeper the decay is, the

better the function can be approximated. As it is visible in the scree plots, the eigenvalues of every single field decay rapidly, meaning that the system can be most likely represented by a low-rank model. This is not surprising, since the MSFR is an exceedingly homogeneous reactor concept in a sense that the solution fields are predominantly smooth in space. It is interesting to note that the eigenvalues of the forward and adjoint group fluxes decay in a similar manner, independent of the energy groups. This is again the consequence of the homogeneous zone together with the fast neutron spectrum. Furthermore, it is visible that the longer the half-life of the neutron precursor group (group 1 - longest half-life, group 8 - shortest half-life), the faster the decay of the corresponding eigenvalues is. This can be explained by the fact that the precursors with the longest half-life are distributed relatively evenly by the flow within the core due to their slow nuclear decay. In case of the groups with the shortest half-lives, most of the neutron precursors do not leave the core before decaying, thus introducing a considerable spatial variation which is also reflected in the decay of the corresponding eigenvalues.

As a next step, multiple reduced-order models are generated using different number of extracted basis functions. The accuracy of these reduced-order models is evaluated using a set of new parameter vectors which do not overlap with the ones used for the collection of the snapshots. Altogether 30 new parameter vectors have been generated using LHS. Based on the conclusions in [3], namely that the one-equation ROM for the fluid dynamics gives good results for laminar fluid flows while being faster than its two-equation counterpart, the one equation ROM has been utilized for laminar cases. For the turbulent scenarios, however, the two-equation ROM is utilized since the physics-based approximations in the one-equation ROM do not hold anymore.

The accuracy of the multiphyics ROMs for laminar steady-state simulations is evaluated with different numbers of POD modes determined by varying the truncation parameter τ (see Chapter 4 for more information). Generally, the decrease in the truncation parameter entails an increase in the number of POD modes used for the reduced-basis expansion. The results of the experiments with the different truncation parameters for laminar scenarios are presented in Tables 6.5. Due to the large number of solution fields, only the ones considered representative are shown. For the

neutronics subproblem, these are the adjoint and forward scalar flux in energy group 5, and the adjoint and forward precursors concentration in precursor group 6. However, it must be mentioned that the trends are the same for the other solution fields. It is visible that the decrease in the truncation limit entails the increase in accuracy. With $\tau = 10^{-7}$, the maximum error in $\mathrm{k_{eff}}$ and β_{eff} is below 1 pcm and 0.5 pcm, respectively.

Table 6.5: The average and maximum errors over the validation set in case of laminar steady-state zero-power simulations MSFR.

τ	$\overline{\Delta \mathrm{k_{eff}}}$ (pcm)	max($\Delta \mathrm{k_{eff}}$) (pcm)	$\overline{e_{\phi_5}}$ (%)	max(e_{ϕ_5}) (%)	$\overline{e_{C_6}}$ (%)	max(e_{C_6}) (%)	$\overline{e_{u_D}}$ (%)	max(e_{u_D}) (%)
10^{-4}	2.79	6.64	0.49	1.20	0.46	0.93	0.50	1.18
10^{-5}	1.09	2.23	0.15	0.28	0.22	0.48	0.05	0.12
10^{-6}	0.83	2.07	0.13	0.34	0.12	0.28	0.05	0.12
10^{-7}	0.17	0.62	0.03	0.06	0.06	0.10	0.02	0.04

τ	$\overline{\Delta \beta_{\mathrm{eff}}}$ (pcm)	max($\Delta \beta_{\mathrm{eff}}$) (pcm)	$\overline{e_{\phi_5^\dagger}}$ (%)	max($e_{\phi_5^\dagger}$) (%)	$\overline{e_{C_6^{\dagger,*}}}$ (%)	max($e_{C_6^{\dagger,*}}$) (%)
10^{-4}	0.338	0.942	0.48	1.28	0.45	0.96
10^{-5}	0.179	0.480	0.15	0.28	0.24	0.52
10^{-6}	0.080	0.254	0.13	0.31	0.14	0.30
10^{-7}	0.011	0.035	0.03	0.05	0.07	0.10

Furthermore, it is also visible that the corresponding average and maximum relative L^2 errors are below 0.07% and 0.1%. Since the expected variation in the quantities of interest in case of steady-state computations is higher than the errors shown over the validation set at a truncation limit of 10^{-7}, the last model has been selected to carry out the uncertainty propagation in the following chapter. The final number of POD modes used for each physics component is summarized

in Table 6.6. It is visible that 3-5 POD modes are enough to describe the variations in the fields of interest. Again, the reason behind this is that the reactor concept is homogeneous and the smooth solution fields can be described using only a few global basis functions.

Table 6.6: The ranks of the subspaces used for the construction of the final ROMs for laminar steady-state simulations.

Field	**Rank**	**Field**	**Rank**	**Field**	**Rank**	**Field**	**Rank**	**Field**	**Rank**
ϕ_1	3	C_2^*	4	$\phi_1^\dagger$	3	$C_2^{\dagger,*}$	4	$\boldsymbol{u}_D$	3
ϕ_2	3	C_3^*	4	$\phi_2^\dagger$	3	$C_3^{\dagger,*}$	4	p	3
ϕ_3	3	C_4^*	4	$\phi_3^\dagger$	3	$C_4^{\dagger,*}$	5	$\boldsymbol{F}_{fr}$	3
ϕ_4	3	C_5^*	4	$\phi_4^\dagger$	3	$C_5^{\dagger,*}$	4		
ϕ_5	3	C_6^*	4	$\phi_5^\dagger$	3	$C_6^{\dagger,*}$	4		
ϕ_6	4	C_7^*	4	$\phi_6^\dagger$	3	$C_7^{\dagger,*}$	4		
C_1^*	4	C_8^*	4	$C_1^{\dagger,*}$	4	$C_8^{\dagger,*}$	4		

Lastly, it must be mentioned that the FOM in laminar scenarios with a coupled steady-state fluid dynamics and forward k-eigenvalue computation have 258,240 degrees of freedom which increases to 484,200 for cases where the adjoint problem has to be solved. These numbers are reduced in the ROM to 57 and 108 unknowns, respectively. Computing a steady-state fluid flow together with a forward k-eigenvalue problem took approximately 2400 s on average, while the ROM was solved in about 0.08 s. This means that a speedup of 3.0×10^4 is obtained with the ROM. However, there are two additional aspects that may change this factor. First, the full-order solver, GeN-Foam, uses a transient simulation to obtain the steady state, which is not efficient in terms of computation time. Second, the ROM still needs a considerable amount of file handling such as reading the POD modes, mesh files and reduced matrices and writing out results. This slows down the ROMs considerably, since solving the reduced equation system takes approximately only $6-8\times10^{-3}$ s. If

β_{eff} needs to be determined as well, the solve time of the FOM increases to 2600 s on average, while the execution time for the ROM, mainly due to the increased amount of file handling, increases to 0.15 s. This yields an approximate speedup factor of 1.7×10^4. The achieved speedup is considered to be satisfactory for the present work, however it must be noted that by storing the necessary files (matrices, POD modes) in the memory of the computer, this factor can be considerably improved.

A similar experiment has been carried out for turbulent cases as well. Due to the fact that the supremizer stabilization of the two-equation ROM can decrease its accuracy, the number of basis functions used in the fluid ROM have to be determined in a different manner; the truncation limit is not necessarily indicative. In this work, we opt for an iterative strategy using the validation set. For this, the number of velocity modes are increased first and then the corresponding number of pressure and supremizer modes are determined in a way that the average relative L^2 error in the velocity and pressure are the lowest with maintaining a stable ROM. The number of eddy viscosity and diffusivity modes is determined using a $\tau = 10^{-6}$ truncation parameter. Below this, the gain in the accuracy of the velocity and precursor concentrations from including additional eddy viscosity and diffusivity modes is negligible. The resulting number of modes and average and maximum L^2 errors for the final fluid ROM are summarized in Table 6.7.

Table 6.7: The used POD modes for the fluid dynamics subproblem together with the resulting average and maximum errors over the validation set in case of turbulent steady-state simulations.

Field	**Rank**	$\overline{e}$ **(%)**	**max(e) (%)**
$\boldsymbol{u}_D$	2 (+1)	0.34	0.68
p	1	0.42	0.92
ν_t	10	0.13	0.50
α_t	10	0.13	0.50

In case of the velocity field, the additional one mode indicates that one supremizer mode was

added to the velocity space. This means that the rank of the velocity subspace is 3. Determining the number of POD modes separately for the fluid dynamics FOM can be justified by the fact that the neutronics subproblem does not influence the field variables of the fluid dynamics submodule at all (zero-power assumption).

Using the selected fluid dynamics ROM, a convergence study has been carried out for the neutronics ROM using the validation set and changing the value of the truncation parameter. The corresponding errors over the validation set are shown in Table 6.8. It is visible that overall the error decreases with the decreasing truncation limit. The final model corresponding to $\tau = 10^{-7}$ yields an average error in the effective multiplication factor and effective delayed neutron fraction of 0.3 pcm and 0.11 pcm with the corresponding maximum errors being 0.93 pcm and 0.3 pcm, respectively. Furthermore, the L^2 error of every field of interest is below 0.28%.

Table 6.8: The average and maximum errors over the validation set in case of turbulent steady-state simulations.

τ	$\overline{\Delta \mathrm{k}_{\mathrm{eff}}}$ (pcm)	$\max(\Delta \mathrm{k}_{\mathrm{eff}})$ (pcm)	$\overline{e_{\phi_5}}$ (%)	$\max(e_{\phi_5})$ (%)	$\overline{e_{C_6^*}}$ (%)	$\max(e_{C_6^*})$ (%)
10^{-4}	3.21	7.09	0.49	1.19	0.35	0.70
10^{-5}	1.53	2.82	0.15	0.28	0.26	0.45
10^{-6}	1.19	2.66	0.13	0.34	0.26	0.41
10^{-7}	0.30	0.93	0.02	0.05	0.20	0.28
τ	$\overline{\Delta \beta_{\mathrm{eff}}}$ (pcm)	$\max(\Delta \beta_{\mathrm{eff}})$ (pcm)	$\overline{e_{\phi_5^\dagger}}$ (%)	$\max(e_{\phi_5^\dagger})$ (%)	$\overline{e_{C_6^{\dagger,*}}}$ (%)	$\max(e_{C_6^{\dagger,*}})$ (%)
10^{-4}	0.45	1.10	0.48	1.26	0.36	0.79
10^{-5}	0.23	0.62	0.15	0.27	0.21	0.37
10^{-6}	0.19	0.51	0.13	0.31	0.21	0.43
10^{-7}	0.11	0.30	0.03	0.03	0.18	0.28

The model corresponding to the neutronics truncation limit of 10^{-7} has been selected for further analysis and uncertainty propagation. The number of POD modes used for each physics component in the neutronics subproblem is summarized in Table 6.9.

Table 6.9: The ranks of the subspaces used for the construction of the final neutronics ROM for turbulent steady-state simulations.

Field	**Rank**	**Field**	**Rank**	**Field**	**Rank**	**Field**	**Rank**
ϕ_1	3	C_2^*	2	$\phi_1^\dagger$	3	$C_2^{\dagger,*}$	2
ϕ_2	3	C_3^*	3	$\phi_2^\dagger$	3	$C_3^{\dagger,*}$	3
ϕ_3	3	C_4^*	3	$\phi_3^\dagger$	3	$C_4^{\dagger,*}$	4
ϕ_4	3	C_5^*	5	$\phi_4^\dagger$	3	$C_5^{\dagger,*}$	5
ϕ_5	3	C_6^*	6	$\phi_5^\dagger$	3	$C_6^{\dagger,*}$	6
ϕ_6	3	C_7^*	5	$\phi_6^\dagger$	3	$C_7^{\dagger,*}$	5
C_1^*	2	C_8^*	5	$C_1^{\dagger,*}$	2	$C_8^{\dagger,*}$	5

Lastly, it must be mentioned that solving a steady state fluid problem together with a forward eigenvalue problem took 4200 s on a single processor core. This is decreased to 2600 s if 4 processor cores are used. The poor scaling is due to the low number of distributed spatial degrees of freedom, meaning that the specific communication between processors is considerable. Meanwhile, executing the ROM for the same problem took 0.09 s on average. Again, the overwhelming majority of this execution time is spent with reading and writing files, only 7×10^{-3} s is needed for the solution of the reduced system. In case of simulations when the quantity of interest is β_{eff}, the simulation time with the FOM increased to 4400 s on average, while the execution time of the ROM increased to 0.19 s. This means that the speedup factors varied between 2.3×10^5 and 4.7×10^5 on average. Altogether, for a simple k-eigenvalue problem, the number of degrees of freedom is decreased from 290,520 to 69, while for cases that require the solution of the adjoint

problem as well, it decreased to from 516,480 to 119.

6.3 Uncertainty Quantification and Sensitivity Analysis for Steady-State Simulations

With the generated ROMs as emulators, the propagation of the uncertainty in the cross sections and the pumping force is investigated in this section. Additionally, a sensitivity study is carried out on the quantities of interest. As already discussed in Section 6.1, there are altogether 13 uncertain parameters in this scenario: $\boldsymbol{\mu} = [D_1, ..., D_6, \nu\Sigma_{f,1}, ..., \nu\Sigma_{f,6}, |\boldsymbol{F}_p|]$, where the diffusion coefficients and cross sections have a uniform distribution within the $\pm 10\%$ interval around their expected values, while the pumping force is uniformly distributed in the $[60, 100]\frac{kN}{m^3}$ interval. Since the simulation times with the ROMs are orders of magnitude shorter compared to that of the FOM, a simple Monte Carlo approach can be used here with a large number of experiments. The goal is to estimate the mean and standard deviation of the effective multiplication factor and the effective total delayed neutron fraction. For this, 5×10^4 random samples are drawn using the distributions of the uncertain parameters. The experiments have been carried out for both turbulent and laminar flows. The estimated mean values and standard deviations together with the corresponding 95% confidence intervals are presented in Table 6.10. The confidence intervals have been computed using a bootstrap (re-sampling with replacement) method with 5,000 cycles. The corresponding histograms are depicted in Figures 6.5 and 6.6 using `seaborn` [136]. A kernel density estimator (smoothed curve) has also been included.

It can be observed that every distribution is close to a Gaussian. It is also visible that the confidence intervals of the mean values are tight, meaning that the presented estimates can be considered accurate. Surprisingly, the difference in the mean values of $\mathrm{k_{eff}}$ for the two fluid flows is not statistically significant. The reason behind this is that the quality of the fluid flow influences only the emission of the delayed neutrons only, which is a small fraction of the total neutron population. This suggests (will be confirmed later) that the variation of other parameters influences $\mathrm{k_{eff}}$ stronger compared to the pumping force. For β_{eff}, on the other hand, we see that the mean value depends considerably on the flow domain. The difference in the mean values (32.25 pcm) is statistically significant.

Table 6.10: The estimated mean values and standard deviations (with confidence intervals) of k_{eff} and β_{eff} for different flow regimes.

$\mathbf{k_{eff}}$				
Flow Regime	**Mean**	**95% CI**	**Std. Dev.**	**95% CI**
Laminar	0.981805	[0.981553, 0.982057]	0.028775	[0.028616, 0.028927]
Turbulent	0.981467	[0.981216, 0.981718]	0.028642	[0.028485, 0.028793]
$\boldsymbol{\beta_{eff}}$ **(pcm)**				
Flow Regime	**Mean**	**95% CI**	**Std. Dev.**	**95% CI**
Laminar	142.92	[142.89, 142.96]	4.00	[3.98, 4.02]
Turbulent	111.90	[111.87, 111.93]	3.53	[3.51, 3.55]

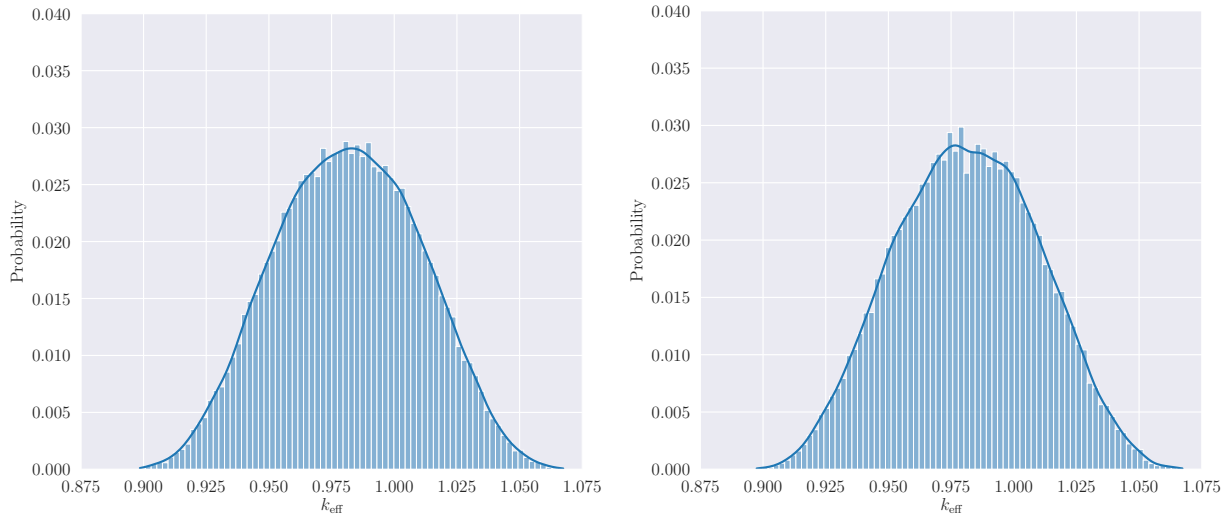


Figure 6.5: The distribution of k_{eff} in case of laminar (left) and turbulent (right) steady-state computations, obtained using 5×10^4 random samples.

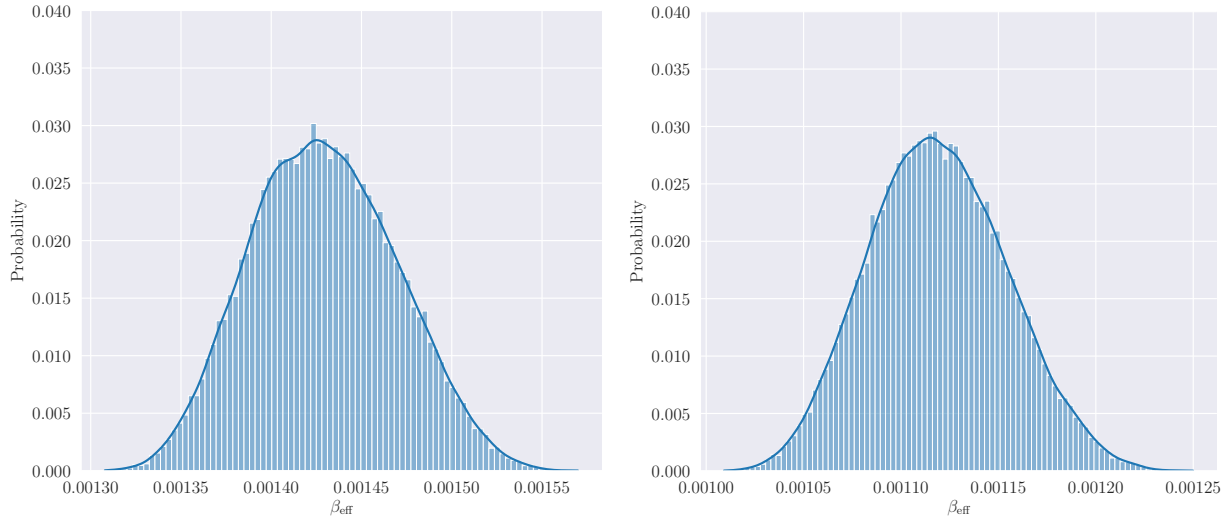


Figure 6.6: The distribution of β_{eff} in case of laminar (left) and turbulent (right) steady-state computations, obtained using 5×10^4 random samples.

Additionally, the joint distribution plots of k_{eff} and β_{eff} are presented in Figure 6.7. It is visible that the two QoIs are correlated in both laminar and turbulent scenarios.

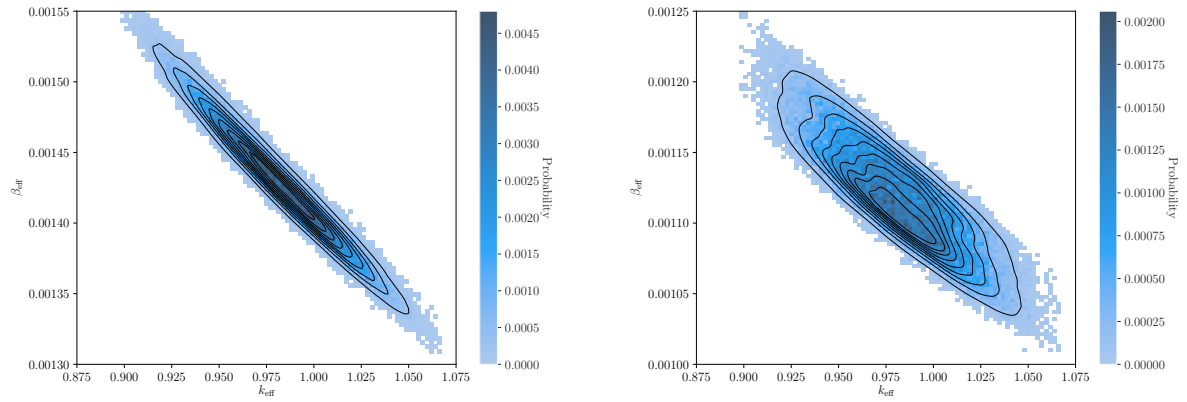


Figure 6.7: The joint distributions of k_{eff} and β_{eff} in case of laminar (left) and turbulent (right) steady-state computations, obtained using 5×10^4 random samples.

A possible explanation for this can be the effect of the changing fission cross sections. Increas-

ing the fission cross section leads to an increase in $\mathrm{k_{eff}}$, while based on Eq. (3.39), it results in a decrease in β_{eff}. The difference in the spread of the data points suggests that the pumping force has a stronger influence in the turbulent scenario. These phenomena will be further investigated using the Sobol Index and Sensitivity Analyses later in this section.

To determine the contributions to the variance from the uncertain parameters, a Sobol Index Analysis has been carried out. For this, the 5×10^4 samples are used to fit a Polynomial Chaos Expansion (PCE) model using a least squares method, as described in Section 5.5. The expansion contained polynomials up to degree $N_{\mathrm{max}} = 4$ meaning that altogether $N_p = 2380$ coefficients have been computed. The quality of the fit has been determined using the absolute residuals and the relative errors over the training set. The corresponding values are summarized in Table 6.11.

Table 6.11: The residuals and relative errors of the fitted PCE over the training set for steady-state zero-power scenarios.

	$\mathbf{k_{eff}}$		$\boldsymbol{\beta_{\mathrm{eff}}}$	
Flow Regime	**Residual**	**Relative Error**	**Residual**	**Relative Error**
Laminar	6.8×10^{-11}	2.8×10^{-13}	2.4×10^{-10}	1.8×10^{-6}
Turbulent	9.8×10^{-10}	5.8×10^{-11}	4.5×10^{-10}	8.5×10^{-6}

The resulting expansion can be used to determine the Sobol Indices analytically. The reader is referred to [128] (or Section 5.5) for the derivation of the connection between the PCE coefficients and the Sobol Indices. The indices for laminar fluid flows are presented in Figure 6.8, while the corresponding results for turbulent flows are presented in Figure 6.9. It must be mentioned, that the error between the FOM and the ROM and the PCE fitting error are not quantified in these figures. Still, the relative values of the indices can be considered indicative, since we assume that the mentioned errors are negligible.

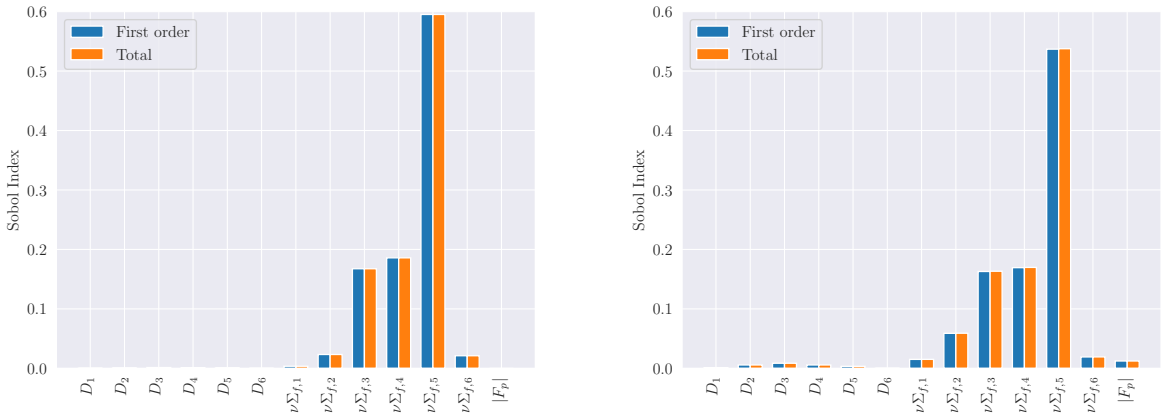


Figure 6.8: First order and total Sobol Indices of k_{eff} (left) and β_{eff} (right) for laminar steady-state computations.

As noted while observing the histograms of k_{eff}, the contribution of the uncertainty in the pumping power is negligible to the contribution of the uncertainties of the fission cross sections. This can be explained by the fact that the variation in pumping power introduces little variation in the precursor concentrations whose contributions to the total neutron population is low by default.

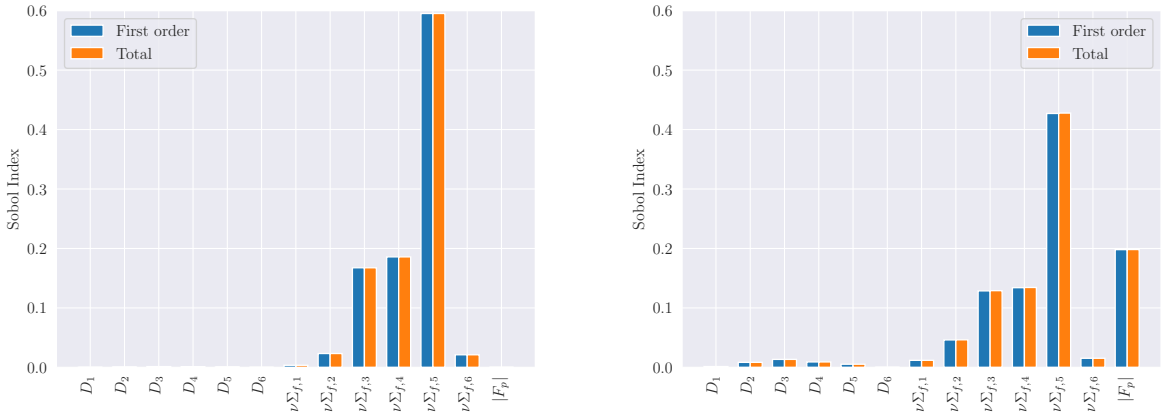


Figure 6.9: First order and total Sobol Indices of k_{eff} (left) and β_{eff} (right) for turbulent steady-state computations.

Furthermore, the increased molecular viscosity (for laminar cases) influences the diffusion of the delayed neutron precursors as well, which decreases the impact of the changing velocity field. What is also worth mentioning is that the diffusion coefficients have close to negligible contributions compared to the fission cross sections. These results are similar to the findings published in [25], which were obtained using a (non-intrusive) POD-based sparse grid interpolation technique. In case of β_{eff}, however, we see that both the pumping force and the diffusion coefficients have meaningful contributions. In the laminar scenario, these contributions are still less than those of the fission cross sections; however, for turbulent flows the uncertainty in pumping power becomes the second biggest contributor. The reason behind this is that the precursor concentrations have a higher contribution to β_{eff} than k_{eff}. Also, the contribution of the diffusion coefficients can be explained by the fact that their change influences the shape of the adjoint flux which has a high impact on β_{eff}.

After the Sobol Index analysis, we investigate the local sensitivity coefficients of the QoIs at the midpoint of the parameter space. Due to the similarity between the results of the laminar and turbulent scenarios, only the turbulent results are presented here in Tables 6.12 and 6.13 for k_{eff} and β_{eff}, respectively.

Table 6.12: Local sensitivity coefficients of the effective multiplication factor at the middle of the parameter space in case of turbulent zero-power steady-state simulations.

$\boldsymbol{\mu_k}$	$\boldsymbol{\kappa_{\mu_k}^{\mathrm{k_{eff}}}}$	$\boldsymbol{\mu_k}$	$\boldsymbol{\kappa_{\mu_k}^{\mathrm{k_{eff}}}}$	$\boldsymbol{\mu_k}$	$\boldsymbol{\kappa_{\mu_k}^{\mathrm{k_{eff}}}}$	$\boldsymbol{\mu_k}$	$\boldsymbol{\kappa_{\mu_k}^{\mathrm{k_{eff}}}}$
D_1	-4.11×10^{-3}	D_5	-1.18×10^{-2}	$\nu\Sigma_{f,3}$	2.08×10^{-1}	$\lvert F_p \rvert$	-1.06×10^{-4}
D_2	-1.13×10^{-2}	D_6	-8.42×10^{-4}	$\nu\Sigma_{f,4}$	2.19×10^{-1}		
D_3	-1.87×10^{-2}	$\nu\Sigma_{f,1}$	3.02×10^{-2}	$\nu\Sigma_{f,5}$	3.92×10^{-1}		
D_4	-1.49×10^{-2}	$\nu\Sigma_{f,2}$	7.75×10^{-2}	$\nu\Sigma_{f,6}$	7.39×10^{-2}		

As expected, the increase in D leads to a decrease in k_{eff}, while the increase in $\nu\Sigma_f$ has an op-

posite effect. In case of β_{eff}, the signs of the corresponding sensitivity coefficients are the opposite: the increase in D leads to an increase in β_{eff} through more homogeneous importance fields, while the increase in $\nu\Sigma_f$ results in a decreaseing β_{eff} due to the increase in the prompt fission term in Eq. (3.39). Furthermore, we see that both k_{eff} and β_{eff} decrease with the increasing pumping force. In terms of magnitudes, similarly to the Sobol Indices, both QoIs are most sensitive to the changes in $\nu\Sigma_f$.

Table 6.13: Local sensitivity coefficients of the effective delayed neutron fraction at the middle of the parameter space in case of turbulent zero-power steady-state simulations.

$\boldsymbol{\mu_k}$	$\boldsymbol{\kappa_{\mu_k}^{\beta_{\text{eff}}}}$	$\boldsymbol{\mu_k}$	$\boldsymbol{\kappa_{\mu_k}^{\beta_{\text{eff}}}}$	$\boldsymbol{\mu_k}$	$\boldsymbol{\kappa_{\mu_k}^{\beta_{\text{eff}}}}$	$\boldsymbol{\mu_k}$	$\boldsymbol{\kappa_{\mu_k}^{\beta_{\text{eff}}}}$
D_1	2.08×10^{-2}	D_5	4.01×10^{-2}	$\nu\Sigma_{f,3}$	-1.96×10^{-1}	$\lvert F_p\rvert$	-9.57×10^{-2}
D_2	5.05×10^{-2}	D_6	2.89×10^{-3}	$\nu\Sigma_{f,4}$	-2.00×10^{-1}		
D_3	6.36×10^{-2}	$\nu\Sigma_{f,1}$	-5.95×10^{-2}	$\nu\Sigma_{f,5}$	-3.58×10^{-1}		
D_4	5.26×10^{-2}	$\nu\Sigma_{f,2}$	-1.18×10^{-1}	$\nu\Sigma_{f,6}$	-6.76×10^{-2}		

Lastly, the required computational effort is compared to a fictional scenario where the uncertainty quantification and sensitivity analysis are carried out with the FOM using one processor core only. In this scenario, the overall speedup factor can be quantified by the following expression:

$$SF = \frac{5 \times 10^4 \cdot \tau_{FOM}}{20 \cdot \tau_{FOM} + \tau_{train} + 5 \times 10^4 \cdot \tau_{ROM}} \approx \frac{5 \times 10^4 \cdot \tau_{FOM}}{20 \cdot \tau_{FOM} + 5 \times 10^4 \cdot \tau_{ROM}}, \tag{6.2}$$

where τ_{FOM} and τ_{ROM} is the computation times necessary to solve the FOM and ROM, respectively. Furthermore, τ_{train} denotes the time necessary for training the ROMs and, for steady-state simulations, we assume that it is negligible compared to the other costs. After plugging in the relevant solve times, this speedup factor turns out to be between 2,184 and 2,372. Of course, if the number of Monte Carlo experiments is increased to obtain better estimates of the statistical mo-

ments, SF would increase as well until it reaches the speedups previously measured in Section 6.2.

6.4 Reduced-Order Models for Transient Scenarios

The generation of the reduced-order models for the transient scenario outlined in Section 6.1 begins with data collection. In this work, 9 transients are carried out with different model parameter vectors. We recall that this problem has two model parameters: the step change in the effective multiplication factor ($\delta \mathrm{k_{eff}}$) and the relative pumping force (a_{F_p}) at the end of the ramp. The quantity of interest is the relative reactor power (normalized by the initial power) at end of the simulation. In this example, the step change in $\mathrm{k_{eff}}$ is introduced at $t = 0\ s$, the ramp in the pumping force starts at $t = 1\ s$ and ends at $t = 4\ s$, while the simulation is terminated at $t = 6\ s$. For the laminar scenario, a 3×3 tensor product grid has been used for sampling parameter vectors, while LHS has been applied in the turbulent setting. The choice of LHS can be justified by the fact that it yields a better resolution in the pumping force dependence of the solution, which is crucial to increase the accuracy of the RBF interpolation when computing the coefficients of the eddy viscosity and eddy diffusivity. It is assumed that the step in the effective multiplication factor has a uniform distribution in the $\delta \mathrm{k_{eff}} \in [25\ \mathrm{pcm},\ 55\ \mathrm{pcm}]$ interval, while the final relative pumping power changes uniformly in $a_{F_p} \in \left[2.0,\ 3.\dot{3}\right]$ interval. The relative reactor power curves of the training transients are presented in Figure 6.10. Visually, we can note that the final relative power is more sensitive to the initial step decrease in the multiplication factor than the pumping power. The reason behind this is simple: the reactor does not start with stationary fluid but with a flow field obtained with a decreased pumping force. Nevertheless, it is visible that the change in pumping force does influence the QoI and the extent of this effect depends on the relation between the magnitude of the change in $\mathrm{k_{eff}}$ and the perturbation introduced by the pump.

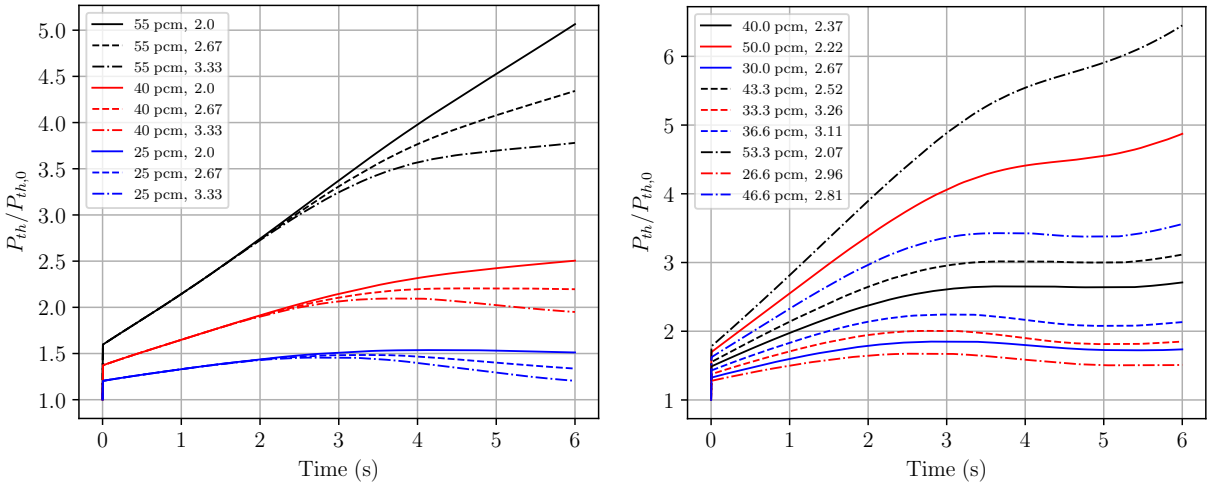


Figure 6.10: Reactor power as function of time for the training transients in case of laminar (left) and turbulent (right) zero-power experiments with the MSFR.

For laminar simulations, the solution fields are captured at every 0.05 s yielding a total of 1,080 snapshots. For turbulent simulations, to ensure a reasonable accuracy for the RBF interpolation of the turbulent viscosity and diffusivity, a slightly finer step size, 0.03 s, has been chosen. This yields 1,800 snapshots per solution field altogether. The basis functions of the reduced subspaces are then extracted from the snapshot matrices using the methods described in Section 4.2.2. Similarly to the steady-state problem, the one-equation fluid dynamics ROM has been used for the laminar scenario, while the two-equation method is utilized for the turbulent computations. The decay in the eigenvalues of the field-wise correlation matrices is shown in Figures 6.11 and 6.12 for the laminar and turbulent scenarios, respectively. It is visible that in both cases the decay in the eigenvalues of the group fluxes is extremely steep suggesting that only a few modes can be enough to reconstruct these fields. This is not surprising, since none of the model parameters change the shape of the flux, unlike in the steady-state experiments where the varying diffusion coefficient introduced notable change in the spatial distribution. Furthermore, one can conclude that the other solution fields show a considerably slower decay suggesting that many POD modes are needed for an accurate approximation.

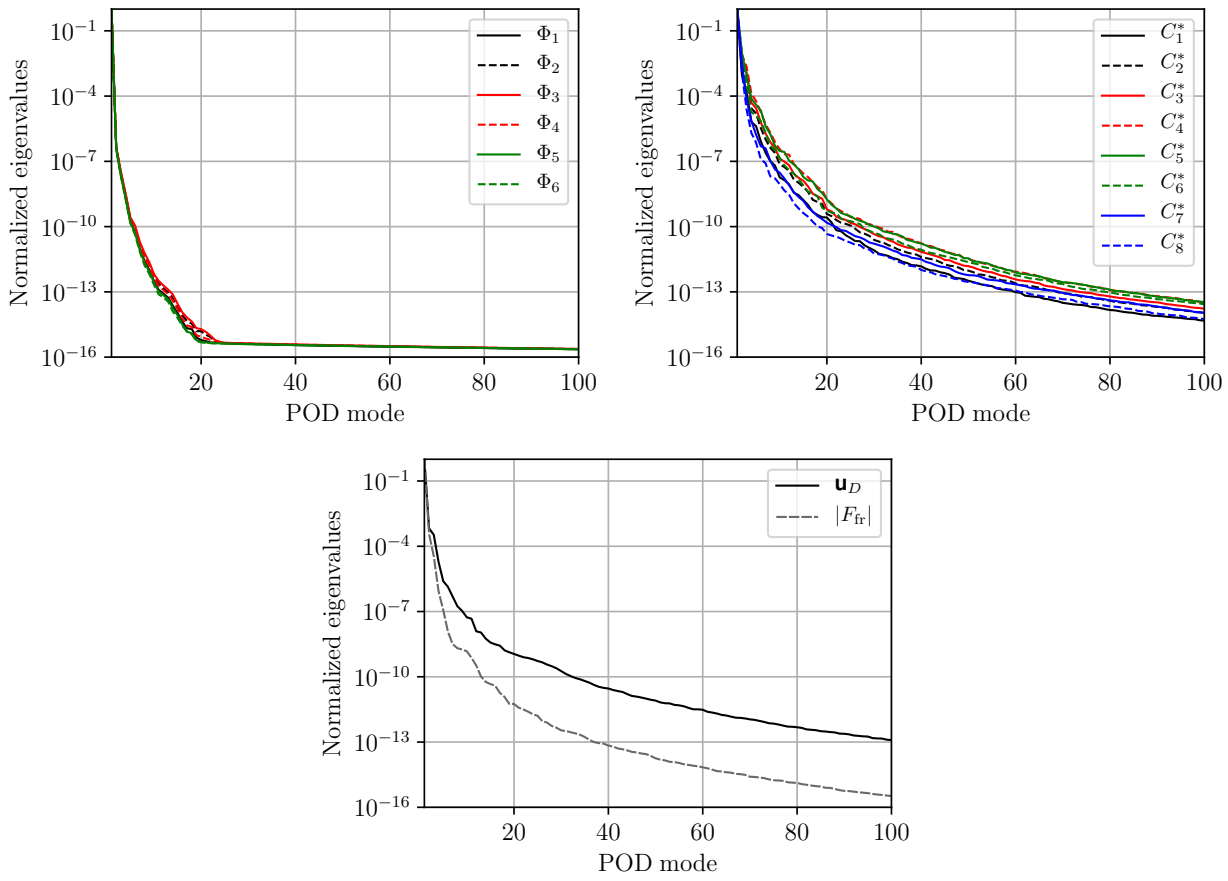


Figure 6.11: Scree plots of the normalized eigenvalues of the correlation matrices built from the corresponding snapshot matrices obtained from transient simulations with a laminar fluid model.

In both laminar and turbulent scenarios, the decay in the eigenvalues of the precursor groups with the longest and shortest half-lives is the fastest. This is not surprising, since the change in the flow pattern has the least influence on these precursor groups. It can also be observed that the decay in all of the precursor groups is much slower in the turbulent scenario. The reason behind this is that the problem is less diffusive and therefore the change in the spatial distributions is less smooth. This is true for the fields of interest in the fluid dynamics subproblem as well. Apparently, the time and parameter dependence of the eddy viscosity and diffusivity is not easy to capture with global basis functions, since these curves show the slowest decay.

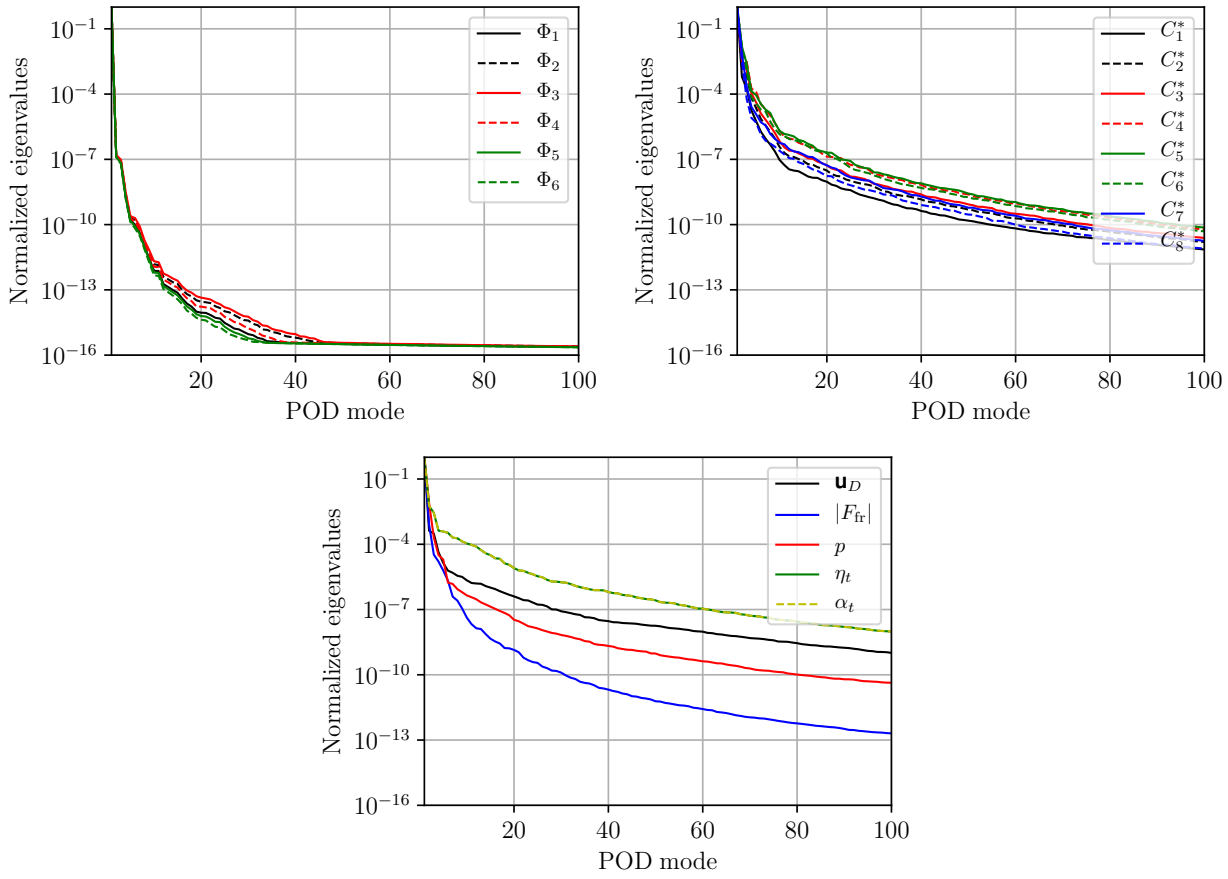


Figure 6.12: Scree plots of the normalized eigenvalues of the correlation matrices built from the corresponding snapshot matrices obtained from transient simulations with a turbulent fluid model.

The number of POD modes per field variable used to construct the ROMs has been determined slightly differently for the laminar and turbulent scenarios. Due to the fact that the laminar fluid ROM is more robust, simply the truncation limit can be utilized. For the testing of the created ROMs, 10 additional parameter vectors are drawn using LHS. The average and relative maximum errors in the QoI and field variables over the test set as a function of the truncation limit is presented in Table 6.14. It is visible that beyond $\tau = 10^{-6}$ the results do not improve, therefore this model has been chosen for uncertainty quantification purposes. We also note that the errors are significantly higher than those presented in steady-state scenarios. This is the result of the inaccuracy of the fluid dynamics ROM and the advection terms discussed in Section 4.3.1.6. Nevertheless, the

average error in both the field variables and the QoI is less than 1%, with maximum errors around 2%. This is considered to be adequate for the uncertainty quantification tasks where the expected parameter-dependent change in these quantities is much higher (see Figure 6.10).

Table 6.14: The average and maximum errors over the validation set in case of laminar transient simulations.

τ	$\overline{e_P}$ (%)	max(e_P) (%)	$\overline{e_{\phi_5}}$ (%)	max(e_{ϕ_5}) (%)	$\overline{e_{C_6}}$ (%)	max(e_{C_6}) (%)	$\overline{e_{u_D}}$ (%)	max(e_{u_D}) (%)
10^{-3}	1.29	3.06	0.94	3.10	2.14	7.02	4.41	8.41
10^{-4}	0.67	1.46	0.36	1.42	0.76	2.50	1.31	2.95
10^{-5}	0.70	1.53	0.35	1.49	0.59	2.03	1.12	1.88
10^{-6}	0.55	1.12	0.37	1.12	0.59	1.93	1.08	1.92
10^{-7}	0.56	1.12	0.39	1.37	0.61	2.11	1.12	1.96

The number of modes for each of the field variables are summarized in Table 6.15 for the model generated using $\tau = 10^{-6}$. It is visible that it is enough to use one spatial basis vector for the group fluxes. This is to be expected based on the decay curves discussed before. It can also be observed that even though the number of model parameters is considerably lower, the number of modes used for the precursor concentrations and the fields in the fluid dynamics ROM are higher. This means that the time evolution of these fields introduces considerable changes in the spatial distributions.

The same procedure has been repeated for the turbulent transient ROMs with slight modifications. Due to the accuracy concerns described in Section 4.3.1.6, the turbulent fluid dynamics ROM cannot be generated by using the truncation limit alone. For this reason, the fluid dynamics ROM has been tested separately using an iterative process for the selection of the number of modes necessary for the best accuracy over the validation set.

Table 6.15: The ranks of the subspaces used for the construction of the final ROMs for laminar transient simulations.

Field	Rank	Field	Rank	Field	Rank	Field	Rank
ϕ_1	1	ϕ_6	1	C_5^*	9	p	6
ϕ_2	1	C_1^*	6	C_6^*	8	$\boldsymbol{F}_{fr}$	4
ϕ_3	1	C_2^*	7	C_7^*	6		
ϕ_4	1	C_3^*	8	C_8^*	4		
ϕ_5	1	C_4^*	9	$\boldsymbol{u}_D$	6		

In the iterative process, the number of eddy viscosity modes has been fixed to 37 which corresponds to $\tau = 10^{-5}$. The number of modes selected for the field variables in the fluid dynamics ROM is listed in Table 6.16 together with the corresponding average and maximum L^2 errors over the transients in the validation set. The (+3) for $\boldsymbol{u}_D$ indicates that the velocity space has been augmented with 3 supremizer modes, meaning that altogether 19 basis functions haven been used. Moreover, altogether 6 modes have been used for the flow resistance, however the corresponding errors have not been analyzed separately.

Table 6.16: The used POD modes for the fluid dynamics subproblem together with the resulting average and maximum errors over the validation set in case of turbulent transient simulations.

Field	Rank	$\overline{e}$ (%)	max(e) (%)
$\boldsymbol{u}_D$	16 (+3)	1.84	4.22
p	3	1.55	9.53
ν_t	37	5.60	31.84
α_t	37	5.60	31.84

It is visible that the RBF interpolation for the eddy viscosity/diffusivity yields high errors. This is due to the fact that the advection dominated problem in time and parameter space is not smooth and the 9×200 interpolation points are still not enough. Unfortunately, further refinement is not practical since the evaluation of the increasing number of interpolation terms can considerably increase the runtime of the ROM. A possible solution for this problem can be the utilization of the sparsity in the parameter space by the evaluation of the interpolation terms that correspond to the points within a certain radius of the given parameter-time vector. This, however has not been analyzed in the frame of this dissertation. For this reason, and based on the results presented in [3], we conclude that the RBF interpolation is better suited for parametric steady-state scenarios. Lastly, we note that an attenuation parameter of $c_a = 0.16$ has been used for the RBF interpolation on the reference grid. Two other parameter values have been investigated: $c_a = 0.09$ and $c_a = 0.25$; however, both yielded considerably higher average and maximum errors.

Using the pre-generated fluid dynamics ROM, the neutronics ROM has been run with selecting the number of basis functions using the truncation limit. The relative errors over the validation set for the QoI and different representative fields are summarized in Table 6.17.

Table 6.17: The average and maximum errors over the validation set in case of turbulent transient zero-power simulations.

τ	$\overline{e_P}$ (%)	max(e_P) (%)	$\overline{e_{\phi_5}}$ (%)	max(e_{ϕ_5}) (%)	$\overline{e_{C_6}}$ (%)	max(e_{C_6}) (%)
10^{-1}	7.06	17.70	10.83	28.92	18.60	32.05
10^{-2}	7.50	14.45	8.93	21.43	10.75	20.63
10^{-3}	1.18	3.56	0.88	3.66	1.95	4.55
10^{-4}	2.54	3.51	0.87	3.61	1.82	4.44
10^{-5}	2.48	4.54	0.79	4.55	1.34	4.63

It is visible that $\tau = 10^{-3}$ yields the best ROM in terms of QoI. Even though the average errors in the field variable slightly decrease, the maximum errors increase in certain cases which may introduce high errors in the QoI. Therefore, the ROM with $\tau = 10^{-3}$ has been selected for uncertainty quantification purposes. This ROM yields an average error of slightly more than 1% for the QoI which is less than the expected total variation in the parameter space.

Lastly, the gain in computational time needs to be addressed. Since an adaptive time-stepping scheme has been used, the runtimes for the FOMs varied considerable depending on the relative change in reactor power. Using one core of a processor, the FOM required 4,000-20,000 s per transient, while the turbulent scenarios were slightly slower with their 4,600-26,000 s. Similarly, due to the adaptive time-stepping, the execution time of the laminar ROMs varied in the 16-23 s interval, while the same for the turbulent ROM is 59-120 s. The source of the considerable difference between the execution times of the laminar and turbulent ROMs is the RBF interpolation of 37 turbulent viscosity and diffusivity modes at 9×200 anchor points at every single timestep. This means that the laminar ROM yields a single-run speedup factor of 250-870, while the same for the turbulent ROM is 78-216.

6.5 Uncertainty Quantification and Sensitivity Analysis for Transient Scenarios

Using the two ROMs (laminar or turbulent) as emulators, we can explore the parameter-dependent response surface of the problem using a simple Monte Carlo approach. We assume that the jump in $\mathrm{k_{eff}}$ and the relative change in pumping power are uncertain parameters in this case with uniform distributions. Using this knowledge, we draw 10,000 new parameter samples for the laminar and 1,400 for the turbulent scenario. The turbulent case has fewer test samples because the ROM is considerable slower. The ROMs are then evaluated using these input parameters and the statistical moments of the QoIs are computed. The mean and standard deviation of the relative reactor power at the end of the simulation are presented in Table 6.18. The corresponding confidence intervals have been computed using a bootstrap (re-sampling with replacement) method with 5,000 cycles. The distributions of the QoI for the two different scenarios are presented in Figure 6.13.

Table 6.18: Mean and standard deviation of the relative power at the end of the transient together with the corresponding confidence intervals over the test set.

Flow Regime	**Mean**	**95% CI**	**Std. Dev.**	**95% CI**
Laminar	2.4029	[2.3865, 2.4198]	0.8617	[0.8512, 0.8726]
Turbulent	2.9961	[2.9290, 3.0643]	1.2799	[1.2339, 1.3245]

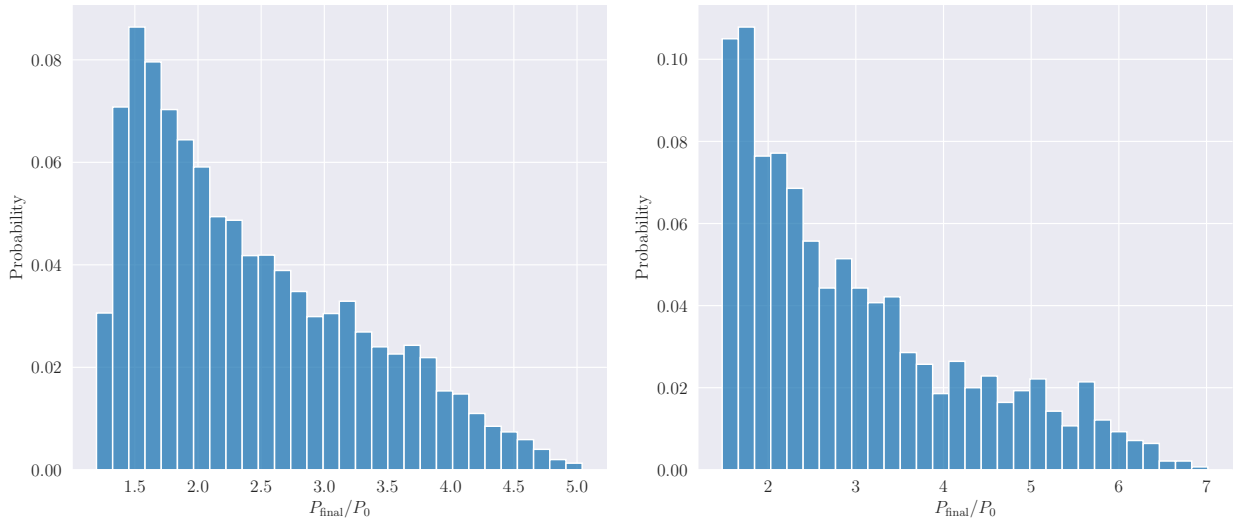


Figure 6.13: The distribution of the relative reactor power in case of zero-power laminar (left) and turbulent (right) transients.

It is visible that the distributions are exponential-like. This suggests that the change in k_{eff} has a larger impact on the time evolution of the power than the change in pumping power. However, it is also visible that the highest probability spot for the laminar scenario is not the lowest relative power, but a slightly higher one. This is attributed to the presence of pump which decreases the relative powers in the mid-low ranges of δk_{eff}.

This effect is not visible in the turbulent scenario suggesting that the change in pumping power has a lower influence on the QoI. To quantify this effect, a Sobol Index Analysis is performed

together with a local sensitivity analysis. The process is similar to the one used for the steady-state results. First, a Polynomial Chaos Expansion is computed using a least squares regression, as described in Section 5.5. Using the computed expansion coefficients, the Sobol Indices and Sensitivity Coefficients can be easily determined. The sum of the residuals and relative errors of the PCE at the training samples for both laminar ant turbulent cases are summarized in Table 6.19. It is important to note that the PCE needs a maximum allowed degree of 4 for the laminar scenario to keep the relative error below 10^{-5}, while the same for the turbulent model is 5.

Table 6.19: The residuals and relative errors of the fitted PCE over the training set in case of zero-power transient simulations.

Flow Regime	**Max. degree**	**Residual**	**Relative Error**
Laminar	4	1.3×10^{-5}	2.4×10^{-6}
Turbulent	5	7.9×10^{-5}	5.3×10^{-6}

The computed Sobol Indices are presented in Table 6.20. As expected, the indices corresponding to the change in the multiplication factor have considerably higher magnitudes. It is also visible that in case of turbulent simulations, the effect of the change in pumping power is even lower. A possible explanation for this phenomenon can be that, assuming a simple point-kinetics approximation without pump, the same change in $\mathrm{k_{eff}}$ causes larger power variation if β_{eff} is lower. Indeed, mainly due to the different velocity profile, the effective delayed neutron yield is lower in case of turbulent scenarios meaning that the same change in the multiplication factor should yield larger variation in the relative power. Also, it is important to note that even though the absolute change in $\mathrm{k_{eff}}$ is the same for both laminar and turbulent scenarios, the corresponding relative change is higher for turbulent cases since the steady-state $\mathrm{k_{eff}}$ is slightly lower: 0.981711 compared to the 0.981969 of the laminar simulation. Furthermore, we recall that in case of the steady-state examples, the interaction between uncertain parameters has been negligible because the total Sobol

Indices turn out to be almost identical to the first order indices. This is different for transient scenarios. Here we see that the total indices are higher than the first order indices, indicating that the effects of the two parameters are clearly coupled through time.

Table 6.20: First order and total Sobol Indices of the parameters for zero-power laminar and turbulent transients.

Flow Regime	**Laminar**	**Tubulent**
First order SI $\delta \mathrm{k_{eff}}$	0.9441	0.9737
Total SI $\delta \mathrm{k_{eff}}$	0.9525	0.9827
First order SI a_{F_p}	0.0475	0.0173
Total SI a_{F_p}	0.0559	0.0265

Similarly to the steady-state cases, a local sensitivity analysis has been carried out using the fitted PCE models at the midpoint of the parameter space. Table 6.21 presents the sensitivity coefficients. It can be observed that the increase in $\mathrm{k_{eff}}$ leads to the increase in the final power, while the increase in pumping power has an opposite effect. Furthermore, based on the magnitudes of the coefficients, we see that the QoI is 3-6 times more sensitive to the change in $\mathrm{k_{eff}}$ (change in fission cross section) than to the change in pumping power. This is in agreement with the conclusions drawn in case of the Sobol Index analysis.

Table 6.21: Local sensitivity coefficients of the quantities interest for transient zero-power simulations at the middle of the parameter space.

Flow Regime	**Laminar**	**Tubulent**
$\delta \mathrm{k_{eff}}$ Sensitivity	1.5311	1.8013
a_{F_p} Sensitivity	-0.5254	-0.3112

Lastly, the savings in computational time are estimated for the uncertainty quantification task by assuming that the ROMs are 560 and 147 times faster on average than the corresponding laminar and turbulent FOMs. With this, the overall speedup factor for the laminar ROM can be quantified by using the formula in Eq. (6.2). This results in an approximate speedup of 391 for the laminar ROM, while the same for the turbulent scenario is 82, mainly due to the lower number of parameter samples in the Monte Carlo study of the ROM. Altogether, it can be concluded that, due to the deficiencies of the RBF interpolation in the turbulent ROM, the computational speedup is rather limited for transient scenarios.

7. REDUCED-ORDER MODELING OF TEMPERATURE FEEDBACK

Using a second numerical example, we demonstrate the efficiency of the method developed for the treatment of temperature-dependent cross sections at reduced-order level. For this, a 2D model of the Molten Salt Reactor Experiment (MSRE) has been adopted. The original model is based on the one developed within the work published in [5] and only slight modifications have been applied. In this example, precomputed fluid flow fields are used, meaning that the fluid dynamics equations are not solved; only the coupled neutronics and heat transfer subproblems are considered. Two combined temperature feedback mechanisms are investigated: the Doppler broadening of the resonances of the cross sections and the expansion of the molten salt with the increasing temperature. These are tested using a steady-state simulation where the quantities of our interest (QoIs) are the effective multiplication factor ($\mathrm{k_{eff}}$) and the maximum system temperature ($T_{\max}$). Additionally, an uncertainty and sensitivity analysis is carried out on the two QoIs using six uncertain model parameters.

7.1 Full-Order Model (FOM) of the Molten Salt Reactor Experiment (MSRE)

The 5-degree wedge geometry simulating a 2D axisymmetric model of the MSRE and the corresponding mesh for the full-order model have been prepared using GMSH [137] and are presented in Figure 7.1. Several approximations have been used in the mesh generation process:

(i) the complex structural elements in the core, upper and lower plenum and heat exchanger are homogenized into porous zones,

(ii) the elements of the external loop (heat exchanger, pump, pipelines) are modeled with a simplified geometry with the salt residence times matched to measurements [5] and

(iii) the downcomer wall is assumed to be infinitesimal thin.

Approximation (ii) could be ensured by fine-tuning the geometry and porosity factors in each of the mesh zones in the external loop. The mesh consists of 38,562 cells altogether, distributed in 11

mesh regions highlighted in Figure 7.1.

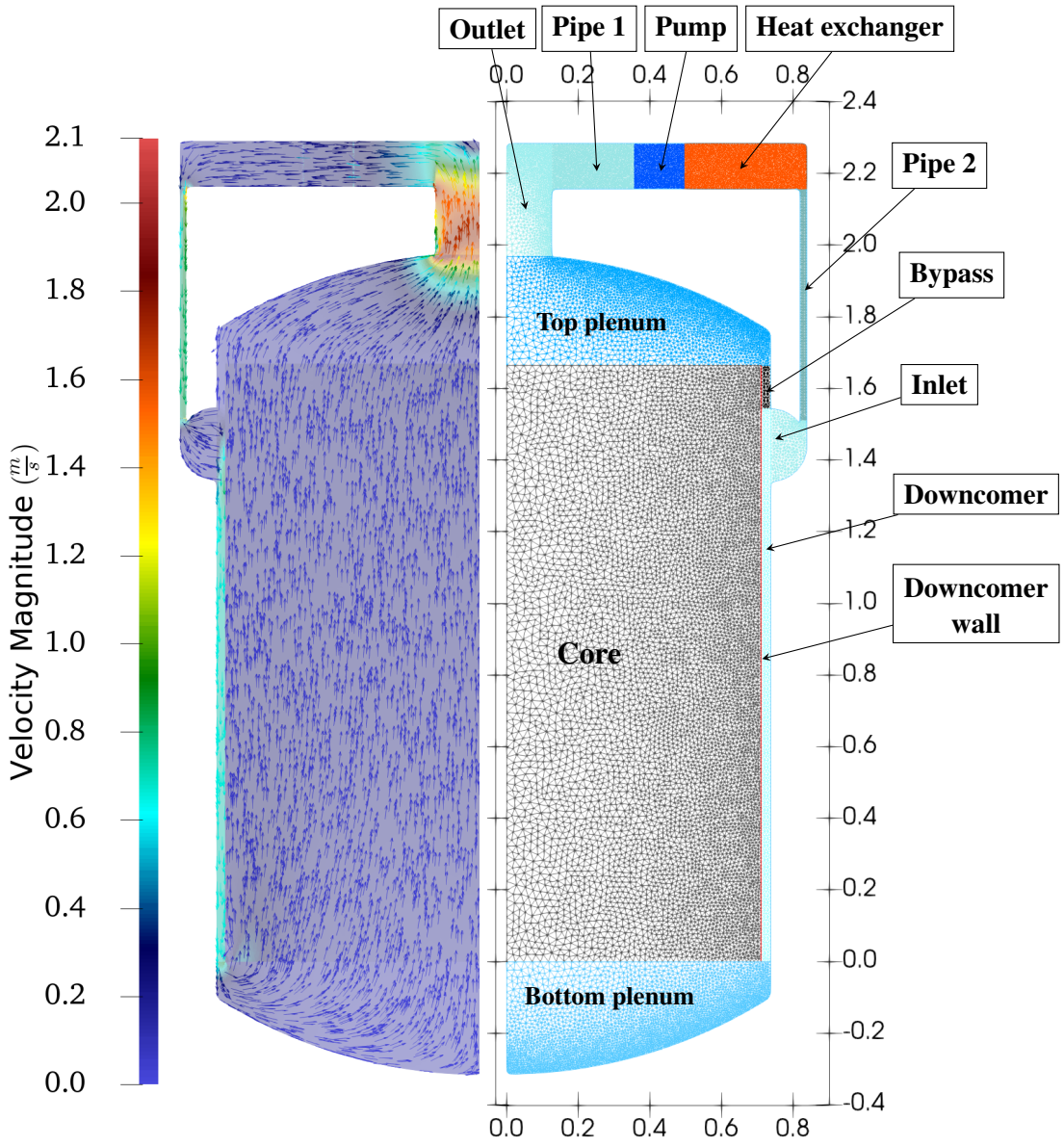


Figure 7.1: Geometry and mesh of the MSRE together with the used real velocity field. Dimensions are in m.

As already mentioned in the introduction, the velocity and turbulent viscosity/diffusivity fields are assumed to be fixed throughout the simulations in this chapter, meaning that the fluid dynamics subproblem is not solved. The used fluid dynamics fields (velocity, turbulent viscosity/diffusivity)

have been obtained by solving the stand-alone Porous Medium Navier-Stokes equations with nominal thermophysical parameters and 1,200 gpm flow rate. The resulting velocity field is included in Figure 7.1 as well.

The temperature of the system is obtained by solving the heat transfer subproblem. We assume that the thermophysical properties of the fuel salt are constant in this subproblem with the values listed in Table 7.1 which corresponds to a 911 K reference temperature. Based on [5, 6], the following fuel salt composition has been used: 62.5LiF-31.6BeF_2-5.1ZrF_4-0.8UF_4 (mol%).

Table 7.1: The thermophysical properties of the molten salt in the MSRE [5, 6].

Parameter name	**Symbol**	**Value**
Density	ρ	$2327.5\ \frac{g}{cm^3}$
Heat capacity	c_p	$1967.8\ \frac{J}{kgK}$
Thermal expansion coefficient	β_{th}	$2 \times 10^{-4}\ \frac{1}{K}$
Reference temperature	T_0	$911\ K$
Dynamic viscosity	η	$0.00785\ Pa \cdot s$
Prandtl number	Pr	10.7

The reactor core, upper and lower plenums together with the heat exchanger are modeled as porous medium with fluid fractions determined using the original design of the MSRE. The corresponding fluid fractions (γ) are summarized in Table 7.2. Every additional zone is assumed to contain clean fluid. The heat exchanger acts as a homogenized heat sink. Lastly, the porous medium enthalpy equations are solved using homogeneous Neumann ($\boldsymbol{\nabla} T \cdot \boldsymbol{n} = 0$) boundary conditions.

Table 7.2: The fluid fractions of the porous medium zones in the MSRE [5].

Zone	γ	**Zone**	γ
Core	0.2677	Bottom plenum	0.7908
Top plenum	0.85	Heat exchanger	0.9477

The neutronics subproblem consists of two energy and six delayed neutron precursor groups. The utilization of the 2-group structure can be justified based on the comparison discussed in [5], namely that the difference between the 2-group and a 23-group structure is only 25 pcm in the steady-state effective multiplication factor. The two energy groups used are separated at 0.625 eV. The homogenized cross sections for each material region have been determined using Serpent 2 [117] Monte Carlo transport code. As discussed in Chapter 3, the temperature-dependence of the cross sections is handled using three different cross section libraries: a reference library ($T_{\text{fuel}} = 911\ K$, $\rho = 2327.5\ \frac{g}{cm^3}$), and two other libraries with perturbed fuel temperatures ($T_{\text{fuel}} = 1200\ K$, $\rho = 2327.5\ \frac{g}{cm^3}$) and fuel salt densities ($T_{\text{fuel}} = 911\ K$, $\rho = 2184.6\ \frac{g}{cm^3}$). The actual value of the cross sections is determined using a linear interpolation in the direction of the fuel salt density and a square root interpolation in the direction of the fuel temperature, respectively. See Section 3.2 for more information on the handling of the temperature-dependent cross sections. The equations of the group fluxes are solved using vacuum boundary conditions ($D_g \boldsymbol{\nabla} \phi_g \cdot \boldsymbol{n} = -0.5\phi_g$, $g = 1, 2$), while homogeneous Neumann boundary conditions have been applied for the corrected precursor concentrations ($\boldsymbol{\nabla} C_i^* \cdot \boldsymbol{n} = 0$, $i = 1, .., 6$).

Lastly, it must be mentioned that six uncertain model parameters are considered. The parameters together with their corresponding distributions are summarized in Table 7.3. It is visible that four out of the six parameters ($A_V,\ \alpha_{HX},\ T_{HX},$ Pr) come from the porous medium enthalpy equation, while the remaining parameters ($P_{th},\ \beta_{th}$) directly influence the neutronics equations. As before, $\mathcal{U}(a,\ b)$ denotes a uniform distribution in the $[a, b]$ interval.

Table 7.3: The distributions of the uncertain model parameters for the steady-state simulations of the MSRE.

Parameter Name	**Symbol**	**Distribution**
Volumetric surface	A_V	$\mathcal{U}(110,\ 150)\ \frac{m^2}{m^3}$
Heat transfer coefficient	α_{HX}	$\mathcal{U}(1.8 \times 10^4,\ 2.2 \times 10^4)\ \frac{W}{m^2 K}$
External coolant temperature	T_{HX}	$\mathcal{U}(700,\ 800)\ K$
Prandtl number	Pr	$\mathcal{U}(9.6, 11.8)$
Thermal expansion coefficient	β_{th}	$\mathcal{U}(1.8 \times 10^{-4},\ 2.2 \times 10^{-4})\ \frac{1}{K}$
Reactor power	P_{th}	$\mathcal{U}(1.0 \times 10^5,\ 1.22 \times 10^5)\ \frac{W_{th}}{5\ \text{degree}}$

7.2 Reduced-Order Model (ROM) of the Molten Salt Reactor Experiment (MSRE)

The generation of the ROM starts with data collection. For this purpose, a training parameter sample set of size 20 has been prepared using Latin Hypercube Sampling (LHS). The FOM is then executed using the samples in this set and the group fluxes, corrected precursor concentrations, the temperature and auxiliary temperature fields are saved into corresponding snapshot matrices. Following this, the spatial basis functions are extracted from the snapshots by POD, which involves the generation of correlation matrices using the snapshots. By observing the eigenvalue spectrum of these correlation matrices, one can get a hint if the given fields can be accurately represented in a low-order subspace. The eigenvalues of the built correlation matrices in this work are plotted in Figure 7.2. It is visible that every curve shows a steep decay indicating that the problem exhibits a low-dimensional behavior. Similarly to the steady-state ROMs in Chapter 6, the eigenvalues of the correlation matrices in case of the precursor concentrations decay faster when the half life of the given precursor group is longer. The reason behind this is the same: precursors with longer half-life are distributed more evenly in the system by the fluid, while the short lived precursors due to their faster nuclear decay introduce a considerable spatial variation.

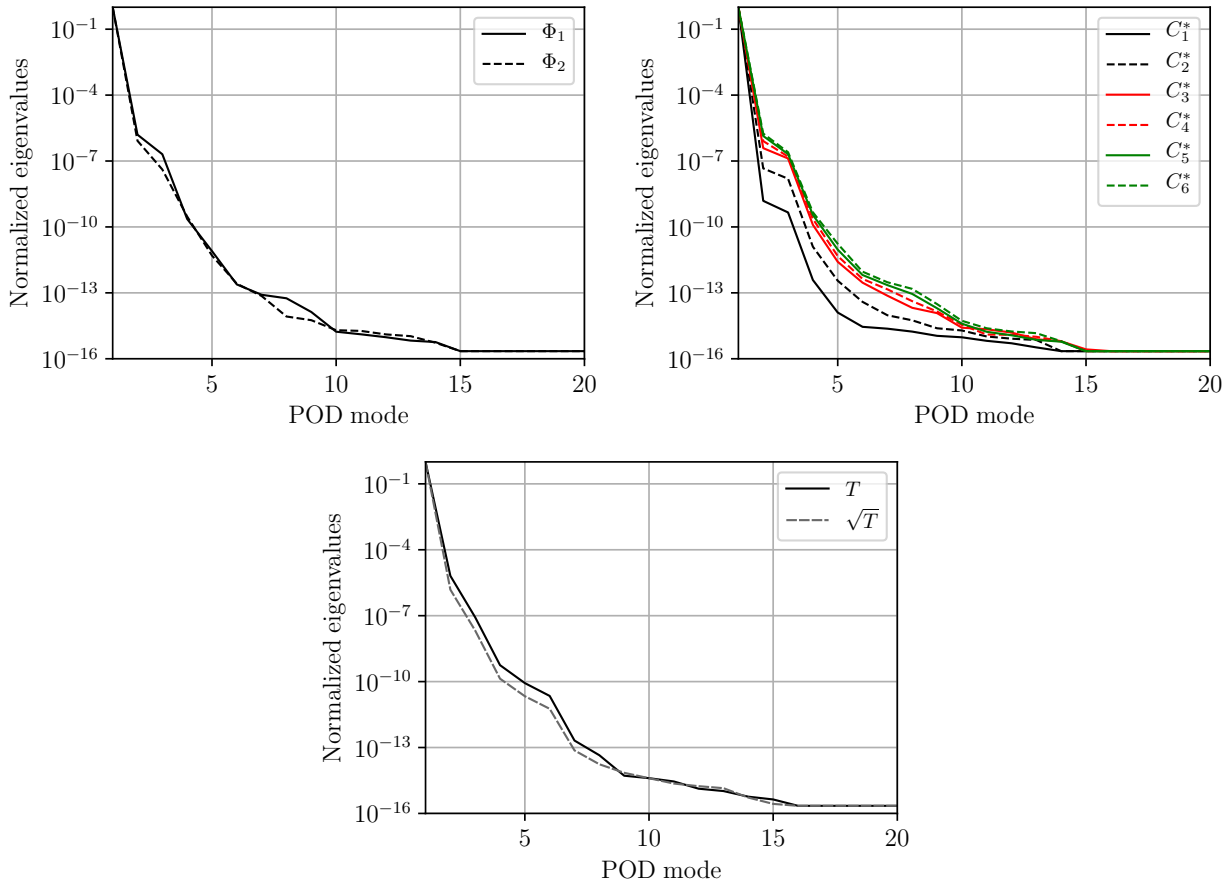


Figure 7.2: Scree plot of the normalized eigenvalues of the correlation matrices built from the corresponding snapshot matrices obtained from steady-state simulations of the MSRE.

Using the snapshots and the eigendecomposition of the correlation matrices, the spatial basis functions are generated for the 10 fields of interest. Figure 7.3 presents the first three spatial basis functions (POD modes) for the scalar flux in the thermal energy group, the precursor concentration in group 5 and the temperature. We see that a first mode for the neutronics fields depicts an average distribution, while the second and third modes introduce the axial asymmetries mainly in the reactor core. It is slightly different in case of the temperature since instead of axial asymmetries, we see that the higher order modes represent the asymmetries in the whole loop, including the heat exchanger.

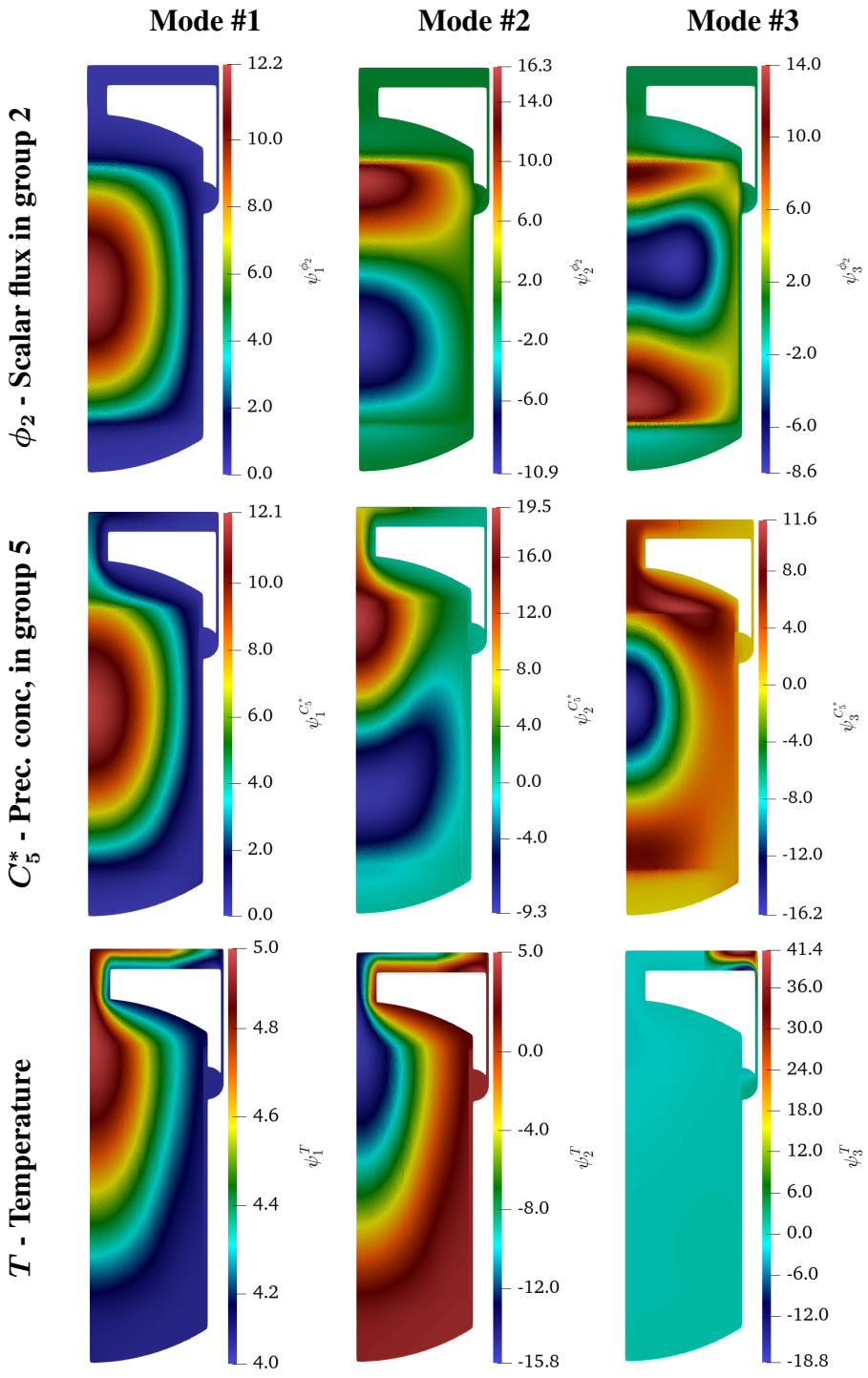


Figure 7.3: The first three POD modes (columns) of the scalar flux in the thermal energy group (first row), the precursor concentration in group 5 (second row) and the temperature (third row) for the steady-state simulations of the MSRE.

As a next step, multiple ROMs are created using different number of POD modes. To test these ROMs, a new parameter sample set (validation set) has been created using LHS. This set contains 30 new samples with no overlap with the training set. The number of POD modes used for the different ROMs has been determined using different values for the energy truncation limit (τ) described in Chapter 2. Then, both the ROM and FOM have been executed using the model parameter samples in the validation set and the generated effective multiplication factors and the solution fields are compared. The statistics of the absolute difference in $\mathrm{k_{eff}}$ and the relative L^2 difference in the representative solution fields are summarized in Table 7.4. It is visible that the decreasing truncation limit results in ROMs with higher accuracy. We note, however, that the increase in accuracy is almost negligible at high values for τ. The reason behind this is that the decay curves of the fields of interest are very steep and it takes multiple orders of magnitude change in τ to include additional modes. Nevertheless, it can be concluded that the method for the treatment of the temperature-dependent cross sections results in accurate ROMs in this case. Using $\tau = 10^{-10}$, the maximum absolute difference in the eigenvalues of the ROM and the FOM over the validation set is 0.36 pcm, which is considerably less than the expected variation in the parameter space.

Table 7.4: The average and maximum errors over the validation set in case of steady-state simulations of the MSRE.

τ	$\overline{\Delta \mathrm{k_{eff}}}$ (pcm)	max($\Delta \mathrm{k_{eff}}$) (pcm)	$\overline{e_{\phi_2}}$ (%)	max(e_{ϕ_2}) (%)	$\overline{e_{C_5}}$ (%)	max(e_{C_5}) (%)	$\overline{e_T}$ (%)	max(e_T) (%)
10^{-5}	14.21	49.30	0.090	0.191	0.108	0.212	0.278	0.760
10^{-6}	13.15	40.20	0.093	0.190	0.115	0.234	0.195	0.589
10^{-7}	11.65	35.61	0.059	0.167	0.050	0.161	0.193	0.584
10^{-8}	0.39	1.23	0.003	0.009	0.003	0.008	0.006	0.018
10^{-9}	0.40	1.23	0.003	0.009	0.003	0.008	0.006	0.018
10^{-10}	0.08	0.36	0.001	0.001	0.001	0.001	0.003	0.012

Furthermore, the relative L^2 difference in the fields of interest is always lower than 0.003%, which also suggest a good agreement between the results of the FOM and ROM. Therefore, the ROM built with $\tau = 10^{-10}$ has been selected for further use for uncertainty quantification. The number of POD modes used for the construction of this ROM is presented in Table 7.5 for every approximated field. It is visible that 3-5 spatial basis functions are enough to describe the system with great accuracy. This is not surprising, since just like the MSFR, the MSRE is also an exceedingly homogeneous reactor which is further homogenized using the porous medium treatment and the additional approximations in the geometry. Another interesting thing to note is that while the temperature field needs to be approximated with 5 basis functions, it is enough to use 4 for its square root.

Table 7.5: The ranks of the subspaces used for the construction of the final ROMs for the steady-state simulations of the MSRE.

Field	**Rank**	**Field**	**Rank**
ϕ_1	4	C_4^*	4
ϕ_2	4	C_5^*	4
C_1^*	3	C_6^*	4
C_2^*	3	T	5
C_3^*	4	$\sqrt{T}$	4

Lastly, the time savings resulting from using a ROM instead of the FOM for as single run is evaluated. The solution of a single steady-state problem on a single core of a processor takes approximately 1,200 s using the FOM. Compared to this, the ROM built using $\tau = 10^{-10}$ takes only 0.4 s. This means that the ROM results in an approximate overall speedup factor of 3,000. It must be mentioned, however, that the majority of the 0.4 s needed for the execution of the ROM is spent with file input-output operations, since the pure solution time of the ROM is only 0.019 s. In

repeated computations, when the ROM is executed multiple times and the precomputed matrices do not need to be loaded separately for every run, a considerably higher speedup factor can be achieved.

7.3 Uncertainty Quantification and Sensitivity Analysis

Using the previously created ROM as emulator, a parameter-study has been carried out. The six uncertain model parameters introduced for the FOM together with their assumed distributions are summarized in Table 7.3. We recall that the quantities of interest in this case are the effective multiplication factor $\mathrm{k_{eff}}$ and the maximum temperature T_{max}. Therefore, we are are using the emulator to approximate the $\mathrm{k_{eff}} = \mathrm{k_{eff}}(\boldsymbol{\mu})$ and $T_{\mathrm{max}} = T_{\mathrm{max}}(\boldsymbol{\mu})$ functions. First, we attempt to propagate the uncertainties in the input parameters to the quantities of interest. Since the ROM is significantly faster than the FOM, a simple Monte Carlo sampling approach has been chosen. This means that 50,000 new parameter samples have been drawn using the corresponding probability distribution and the results of the simulations are tallied.

The estimated means and standard deviations of the QoIs over the test set are presented in Table 7.6. The corresponding 95% confidence intervals are included as well. The confidence intervals have been computed using a bootstrap (re-sampling with replacement) method with 5,000 cycles in this case. The histograms of the QoIs are depicted in Figure 7.4. It is visible that both histograms are slightly skewed, they deviate from a Gaussian. This deviation is further analyzed with the normality plots in Figure 7.5. These plots compare the logarithmic distribution from the experiments to that of a fitted Gaussian curve. It is visible that the distribution of the effective multiplication factor exhibits a larger deviation from Gaussian, since not only the tails (before 5% and beyond 95%) are off the fitted Gaussian, but the middle region as well. In case of the maximum temperature, the mid-section of the distribution conforms a Gaussian, however as we approach the tails, the deviation increases considerably.

Table 7.6: Mean and standard deviation of k_{eff} and T_{max} together with the corresponding confidence intervals for the steady-state simulations with the MSRE.

$\mathbf{k_{eff}}$			
Mean	**95% CI**	**Std. Dev. (pcm)**	**95% CI (pcm)**
1.030949	[1.030924, 1.030975]	287.4	[285.9, 288.8]
$\boldsymbol{T_{max}}$ **(K)**			
Mean	**95% CI**	**Std. Dev.**	**95% CI**
1015.10	[1014.78, 1015.42]	36.73	[36.54, 36.93]

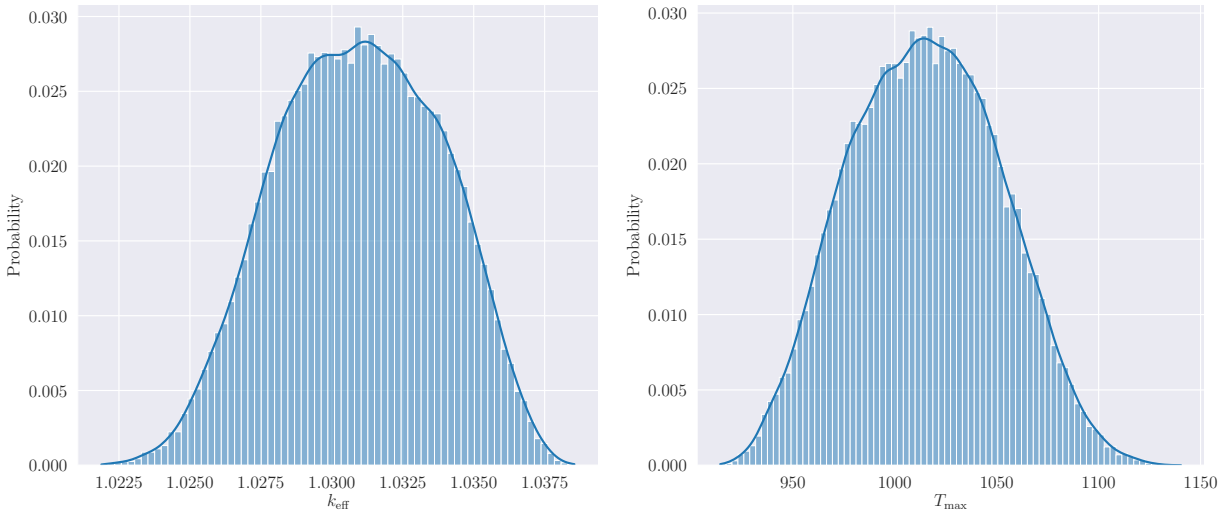


Figure 7.4: The distribution of the effective multiplication factor (left) and the maximum temperature (right) for steady-state simulation with the MSRE.

In order to investigate which uncertain parameter has the highest impact on the variance of the QoIs, a Sobol Index Analysis is carried out. Similarly to the methodology discussed in Chapter 6, a Polynomial Chaos Expansion (PCE) is fitted onto the results from the 50,000 tests samples using a least squares regression. This has been carried out using the `OpenTURNS` [131] library, as

described in 5.5. Since the distributions of the uncertain variables are assumed to be uniform, Legendre polynomials have been used for the PCE model. The maximum allowed polynomial degrees together with the resulting number of expansion terms and the residuals and relative errors of the fitting process are summarized in Table 7.7. The Sobol Indices can then be computed using the expansion coefficients in the fitted PCE.

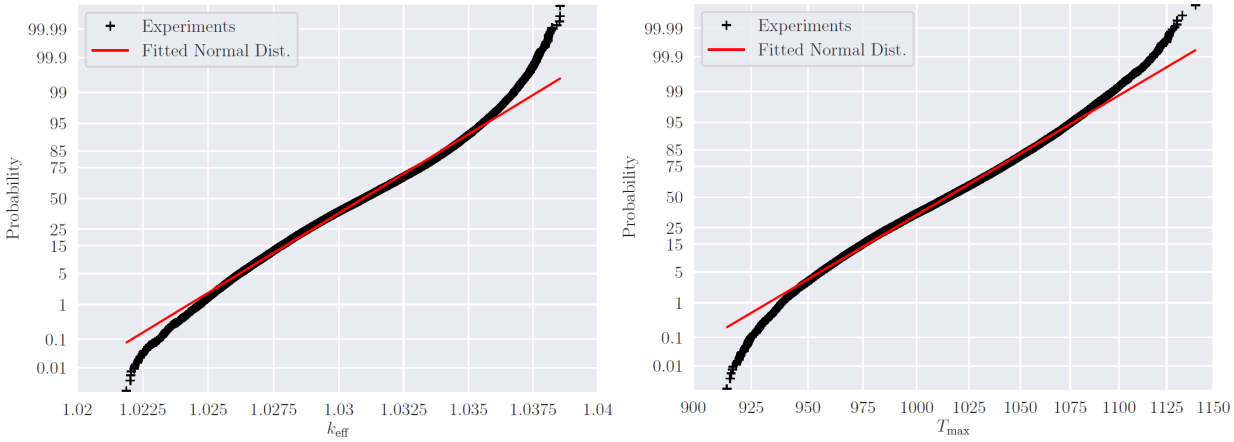


Figure 7.5: Normality plots of the effective multiplication factor (left) and the maximum temperature (right) for steady-state simulations with the MSRE.

It is visible that more terms are needed to reduce the relative error in the PCE to about 10^{-10} in case of the effective multiplication factor. Nevertheless, we can also observe that both in terms of residuals and relative errors, the PCE fit shows good statistics, therefore we conclude that the fitted models are satifactory for the prediction of the Sobol Indices. The Sobol Indices computed using the PCE are presented in Figure 7.6. It must be mentioned that the error in the fitting of the PCE and the difference between the ROM emulator and the actual result from the FOM are not quantified in the figure and these tasks are beyond the scope of this dissertation. Note, however, that due to the error statistics presented in Tables 7.4 and 7.7, we assume that these errors are negligible.

Table 7.7: Important parameters together with the residuals and relative errors of the fitted PCE over the training set in case of steady-state simulations with the MSRE.

QoI	Max. degree	Number of terms	Residual	Relative Error
k_{eff}	6	924	1.3×10^{-10}	1.0×10^{-10}
T_{max}	5	462	1.4×10^{-6}	6.9×10^{-11}

The results show that the variation in the temperature of the external coolant in the heat exchanger is the most dominant contributor to the uncertainty in both QoIs. Furthermore, while the uncertainty in the heat transfer coefficient is the second largest contributor in case of k_{eff}, it is only the third in case of T_{max}, since T_{max} is more sensitive to the thermal power which is a direct source in the heat transfer subproblem. The reason behind the difference in the ordering is that the magnitude of the heat source does not influence the effective multiplication factor directly, only through the temperature feedback.

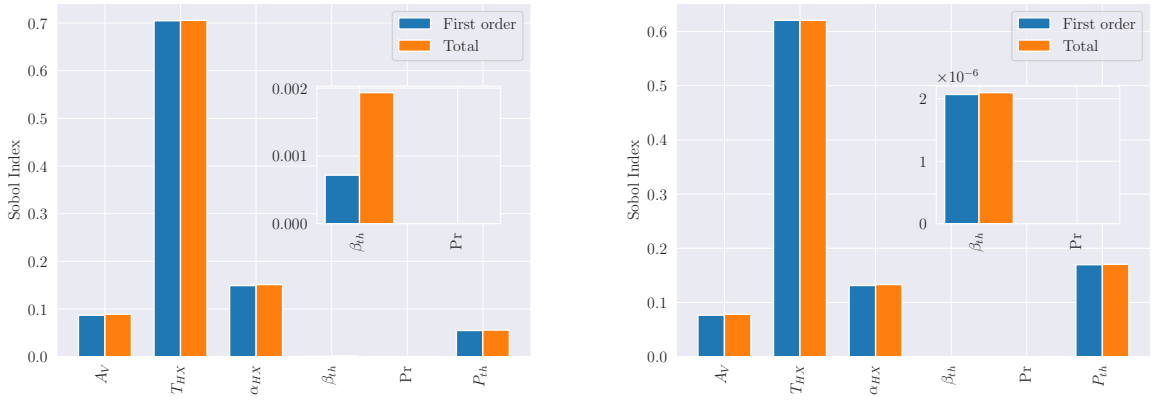


Figure 7.6: Sobol Indices of the effective multiplication factor (left) and maximum temperature (right) in case of steady-state simulations with the MSRE.

We also point out that the total indices are generally 1-2% higher than the first order indices,

meaning that there is a low level of interaction between the chosen uncertain parameters, with one exception: the Prandtl number. The total thermal expansion coefficient index for $\mathrm{k_{eff}}$ is more than three times the magnitude of its first order counterpart meaning that the interactions with other parameters is higher than the first order effect. This is not surprising, since thermal expansion coefficient only influences the density feedback which depends on the temperature which is determined using the other parameters.

Following this, the sensitivities of the QoIs to the model parameters at the middle of the parameter space are investigated using the results in Table 7.8. The sensitivity coefficients can also be generated using the previously fitted PCE using the method described in Section 5.5.

Table 7.8: The local sensitivity coefficients of the quantities of interest at midpoint of the parameter space in the MSRE.

Parameter	Symbol	$\boldsymbol{\kappa^{\mathrm{k_{eff}}}}$	$\boldsymbol{\kappa^{T_{\max}}}$
Volumetric surface	A_V	8.93×10^{-3}	-1.09×10^{-3}
External temperature	T_{HX}	-6.07×10^{-2}	7.43×10^{-1}
Heat transfer coefficient	α_{HX}	8.93×10^{-3}	-1.09×10^{-3}
Thermal expansion coefficient	β_{th}	1.43×10^{-3}	-9.07×10^{-4}
Prandtl number	Pr	9.07×10^{-7}	2.42×10^{-5}
Reactor power (per 5 degree wedge)	P_{th}	-1.12×10^{-2}	2.59×10^{-1}

We see that with one exception, the signs of the sensitivities of $\mathrm{k_{eff}}$ and $T_{\max}$ are opposite. This is not surprising since the changes in parameter values that lead to the increase in the system temperature generally lead to the decrease in $\mathrm{k_{eff}}$ through the negative feedback coefficients. The increase in Prandtl number results in a decrease in thermal conductivity which, in turn, leads to the increase in the maximum temperature. Furthermore we see that both QoIs have negligible sensitivities to Pr which is not surprising since the turbulent mixing effects almost completely

overshadow the heat conduction in the fluid. Similarly to the Sobol Indices, it turns out that the QoIs are most sensitive to the external coolant temperature in the heat exchanger.

Lastly, the gain in computational time is quantified by comparing the time needed to generate the 50,000 samples with the ROM to a theoretical scenario where the same task is carried out using the FOM. The equation described in Section 6.3 can be used for this comparison by plugging in $\tau_{ROM} = 0.4\ s$ and $\tau_{FOM} = 1200\ s$ in. With this, we get a speedup factor of $SF = 1,363$ which proves that using ROMs for tasks like uncertainty quantification can be highly beneficial.

8. REDUCED-ORDER MODELING OF A FULLY-COUPLED PROBLEM

Following the examples of partially coupled systems, we turn our attention to cases that require the solution of all three subproblems discussed in Section 3.1. In this chapter, we present the applicability of the developed POD-RB method to a parameterized steady-state problem using a 2D model of the Molten Salt Fast Reactor (MSFR) at nominal power level. The constructed ROMs are then utilized for the uncertainty quantification and sensitivity analysis of the system at hand. Model parameters include group constants in the neutronics subproblem, pumping force and thermal expansion coefficient from the fluid dynamics subproblem and other thermophysical properties and heat exchanger parameters from the heat transfer subproblem. Altogether 23 model parameters are considered. The quantities of interest (QoIs) are the effective multiplication factor ($\mathrm{k_{eff}}$) together with the maximum temperature in the system (T_{max}). A similar example has been investigated in [25] using a 3D model and a neutron transport solver instead of the diffusion approximation employed in this work. Whenever possible, we emphasize the similarities and differences between the results in this section and those presented in [25].

8.1 The Full-Order Model (FOM) for Fully-coupled Simulations of the MSFR

The FOM used for the current example is based on the model discussed in Chapter 6 with certain extensions. Therefore, only the additional details not mentioned in Section 6.1 are covered here. The most important difference is that the power level in this scenario is not negligible, therefore the heat transfer subproblem needs to be solved as well. This means that buoyancy effects and the temperature-dependence of the cross sections are both taken into account. Furthermore, this example involves turbulent fluid flows only. The important thermophysical properties have been imported from Table 6.2. Beyond these properties, three additional fluid parameters need to be defined for the characterization of the fluid dynamics and energy equations. These parameters are presented in Table 8.1. As before, $\mathcal{U}(a, b)$ denotes a uniform distribution within the [a,b] interval. In this scenario, the thermal expansion coefficient and the Prandtl number are uncertain

parameters. Since the salt characteristics are not entirely fixed for the MSFR yet, the used values are approximates based on [7].

Table 8.1: The thermophysical properties of the molten salt in the MSFR [7].

Parameter name	**Symbol**	**Value**
Heat capacity	c_p	$1600\ \frac{J}{kgK}$
Thermal expansion coefficient	β_{th}	$\mathcal{U}(1.8 \times 10^{-4},\ 2.2 \times 10^{-4})\ \frac{1}{K}$
Reference temperature	T_0	$900\ K$
Prandtl number	Pr	$\mathcal{U}(7.2,\ 8.8)$

The thermal conductivity of the fluid can be computed using the heat capacity (c_p), the dynamic viscosity (η) and the Prandtl number (Pr) of the fluid as:

$$k_l = \frac{c_p \eta}{\mathrm{Pr}}, \tag{8.1}$$

meaning that the thermal conductivity is uncertain as well. Similarly to Section 6.1, the heat exchanger is treated as a porous medium. The parameters used to describe the flow resistance in the porous region are the same as presented in Table 6.2. However, in this case, the heat exchanger serves as a heat sink and can be characterized by the parameters in Table 8.2. It must be mentioned that these parameters are approximate values since the exact design is not publicly available yet.

Table 8.2: The porous medium properties of the heat exchanger in the MSFR.

Parameter name	**Symbol**	**Value**
Volumetric surface	A_V	$200\ \frac{m^2}{m^3}$
Heat transfer coefficient	α_{hx}	$\mathcal{U}(0.8 \times 10^5,\ 1.2 \times 10^5)\ \frac{W}{m^2 K}$
External temperature	T_{hx}	$\mathcal{U}(850,\ 950)\ K$

The pump (Zone P in Figure 6.1) is treated as a homogeneous momentum source. Since no specific design is available in the literature, we assume that the volume fraction of the fluid is 100% ($\gamma = 1.0$) in this region. The magnitude of the pumping force (momentum source) is assumed to be an uncertain model parameter with a distribution highlighted in Table 8.3.

Table 8.3: The pumping force used in the fluid dynamics subproblem.

Parameter name	**Symbol**	**Value**
Pumping force	$\lvert F_p \rvert$	$\mathcal{U}(60,\ 100)\ \frac{kN}{m^3}$

The boundary conditions used for the fluid dynamics and neutronics fields are exactly the same as the ones discussed in Section 6.1. Since the energy equation is also solved in this case, a boundary condition needs to be specified for the temperature as well. We assume that the reactor is well insulated, therefore a homogeneous Neumann condition can be used ($\nabla T \cdot \boldsymbol{n} = 0$) for every wall segment. The same condition is used on the symmetry planes as well. Throughout the computations we assume that the reactor power is fixed at $3\ GW_{th}$.

As in Chapter 6, the steady-state neutronics fields are obtained through the solution of a multi-group diffusion k-eigenvalue problem. The group constants used for the neutron diffusion prob-

lems have been generated using the Serpent 2 Monte Carlo transport code [117]. Similarly to Chapter 6, 6 energy groups and 8 precursor groups have been employed. The energy group structure has been summarized in Table 6.4. Altogether 3 cross section databases have been generated: one with nominal parameters ($T_{\text{salt}} = 900\ K$, $\rho_{\text{salt}} = 4125\ \frac{kg}{m^3}$), one with perturbed salt temperature ($T_{\text{salt}} = 1500\ K$, $\rho_{\text{salt}} = 4125\ \frac{kg}{m^3}$) and a last one with perturbed salt density ($T_{\text{salt}} = 900\ K$, $\rho_{\text{salt}} = 3419\ \frac{kg}{m^3}$). The temperature-dependent cross sections are determined using these libraries by a logarithmic interpolation in the direction of fuel temperature (Doppler feedback) and a linear interpolation in the direction of salt density. For more information on the treatment of the parameterized cross sections, see Section 3.2.4. We assume that the diffusion coefficients ($D_i,\ i = 1, ..., 6$), fission neutron yield times fission cross sections ($\nu\Sigma_{f,i},\ i = 1, ..., 6$) and removal cross-sections ($\Sigma_{r,i},\ i = 1, ..., 6$) are uncertain model parameters in the reactor core cavity (Zone C in Figure 6.1). This means that the neutronics subproblem contributes with 18 uncertain parameters. The uncertain group constants are summarized in Table 8.4.

Table 8.4: The distributions of the uncertain group constants for the fully-coupled computations of the MSFR.

Group const.	**Distribution**	**Group const.**	**Distribution**
D_1	$\mathcal{U}(0.021,\ 0.026)$ cm	$\nu\Sigma_{f,4}$	$\mathcal{U}(0.561,\ 0.685)\ \frac{1}{\text{cm}}$
D_2	$\mathcal{U}(0.014,\ 0.017)$ cm	$\nu\Sigma_{f,5}$	$\mathcal{U}(1.320,\ 1.61)\ \frac{1}{\text{cm}}$
D_3	$\mathcal{U}(0.009,\ 0.011)$ cm	$\nu\Sigma_{f,6}$	$\mathcal{U}(4.283,\ 5.235)\ \frac{1}{\text{cm}}$
D_4	$\mathcal{U}(0.011,\ 0.013)$ cm	$\Sigma_{r,1}$	$\mathcal{U}(6.210,\ 7.590)\ \frac{1}{\text{cm}}$
D_5	$\mathcal{U}(0.010,\ 0.012)$ cm	$\Sigma_{r,2}$	$\mathcal{U}(3.560,\ 4.351)\ \frac{1}{\text{cm}}$
D_6	$\mathcal{U}(0.010,\ 0.012)$ cm	$\Sigma_{r,3}$	$\mathcal{U}(1.597,\ 1.952)\ \frac{1}{\text{cm}}$
$\nu\Sigma_{f,1}$	$\mathcal{U}(0.517,\ 0.633)\ \frac{1}{\text{cm}}$	$\Sigma_{r,4}$	$\mathcal{U}(1.785,\ 2.182)\ \frac{1}{\text{cm}}$
$\nu\Sigma_{f,2}$	$\mathcal{U}(0.338,\ 0.413)\ \frac{1}{\text{cm}}$	$\Sigma_{r,5}$	$\mathcal{U}(1.474,\ 1.801)\ \frac{1}{\text{cm}}$
$\nu\Sigma_{f,3}$	$\mathcal{U}(0.370,\ 0.452)\ \frac{1}{\text{cm}}$	$\Sigma_{r,6}$	$\mathcal{U}(3.210,\ 3.923)\ \frac{1}{\text{cm}}$

It must mentioned that even though the group constants are considered uncertain, the parameters describing their temperature-dependence (δ_{FD} and δ_{FT} coefficients in Eq. (4.73)) are not.

This means that altogether 23 uncertain parameters are in the fully-coupled system, organized in parameter vector $\boldsymbol{\mu}$. The quantities of interest, in this scenario, are the maximum temperature in the system (T_{max}) and the effective multiplication factor (k_{eff}). Therefore, this example aims at generating a reduced-order model which can adequately approximate the $\text{k}_{\text{eff}} = \text{k}_{\text{eff}}(\boldsymbol{\mu})$ and $T_{\text{max}} = T_{\text{max}}(\boldsymbol{\mu})$ functions.

Lastly, we review the used finite volume discretization techniques in the FOM. For the discretization of the diffusion terms we used a scheme based on linear interpolation in every case. The discretization of the advection terms, however, depends on the equation itself. In the fluid dynamics equations, a vanLeer [135] scheme has been used. In every other equation involving the transport of passive scalars, due to stability reasons, a simple upwind scheme is used. The steady-state solutions are obtained through a transient simulation which employed the fixed-point iteration scheme described in Figure 3.4. A maximum of 6 velocity correctors have been employed with 2 additional pressure non-orthogonal correctors per cycle. A maximum of 20 non-linear iterations have been used per timestep. This could change adaptively depending on the residuals of the velocity, pressure and energy equations.

8.2 Reduced-Order Model (ROM) of the Fully-coupled System

The generation of reduced-order models begins with the collection of data in the form of snapshots of the solutions. In this case, we save snapshots of the scalar flux in each energy group ($\phi_i,\ i = 1, ..., 6$), the corrected precursor concentrations ($C_i^*,\ i = 1, .., 8$), the temperature (T) together with the logarithmic temperature ($log(T)$), the Darcy velocity ($\boldsymbol{u}_D$), the corrected pressure (p), the eddy viscosity (ν_t) and diffusivity (α_t) and the flow-resistance in the heat exchanger ($\boldsymbol{F}_{\text{fr}}$). The snapshots of these fields are taken using 30 randomly selected (using LHS) points in the parameter space. Note that 30 samples in the 23-dimensional parameter space is considered sparse, however, due to the high cost of the solution of the FOM, and the homogeneous nature of the reactor concept, it is assumed to be an adequate number. The basis functions for these fields

are then extracted from these snapshots using the POD-procedure discussed in Chapter 2. The decay of eigenvalues for each correlation matrix built from the snapshots (indicative of the error in approximating the snapshots) is presented in Figure 8.1.

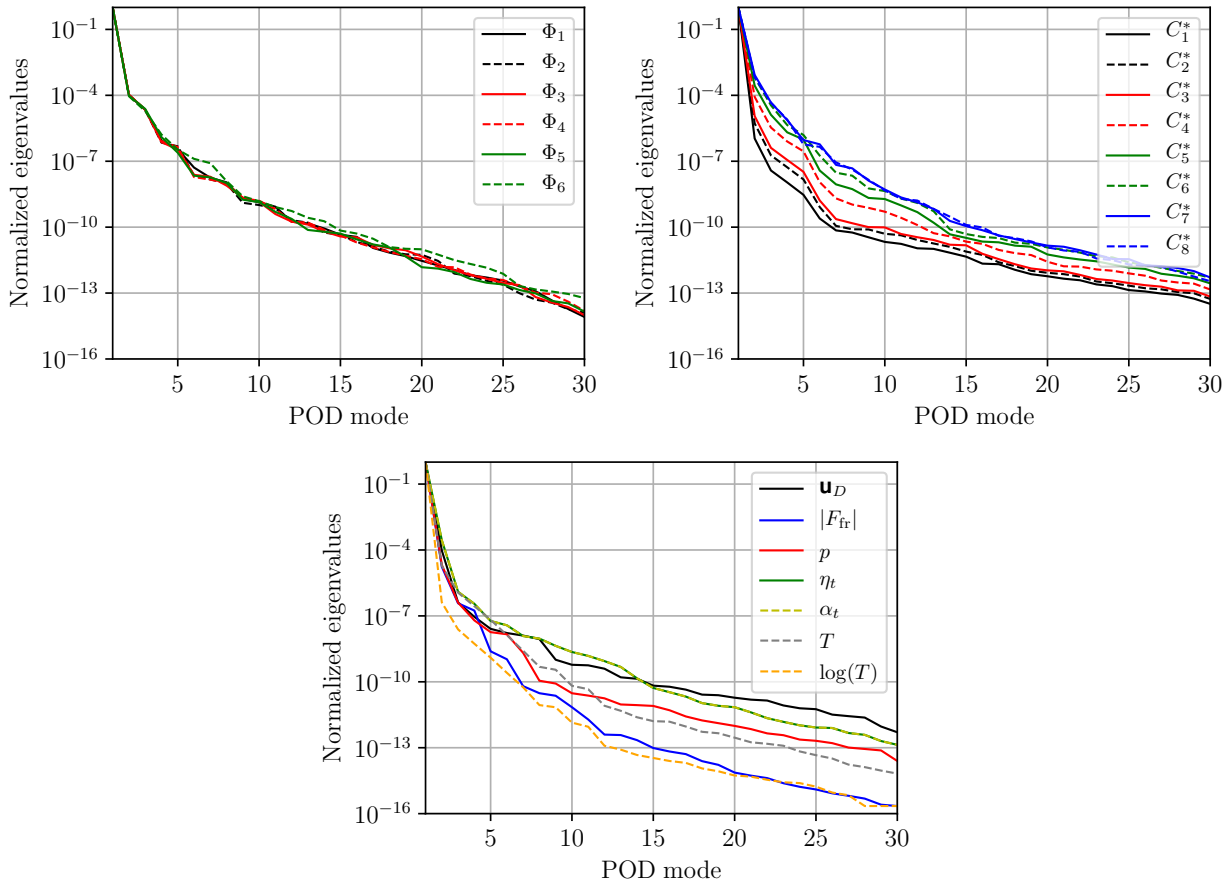


Figure 8.1: Scree plots of the normalized eigenvalues of the correlation matrices built from the corresponding snapshot matrices obtained from steady-state fully-coupled simulations of the MSFR.

We see that the decay of the eigenvalues is much slower compared to the steady-state cases investigated in Chapter 6. This foreshadows that more basis functions (POD modes) are necessary for the accurate reconstruction of the solution fields. Furthermore it is also visible that the precursor concentrations in groups with longer half-lives need fewer POD modes for the same accuracy. As the half-lives of the precursors decrease, more and more basis functions are needed for the

reconstruction to handle the local spatial variations caused by the changing fluid flow.

Due to the stability issues of the one-equation ROM for turbulent closed-loop problems [3], a two-equation fluid ROM has been utilized in this example. The coefficients of the eddy viscosity are therefore determined using the Radial Basis Function (RBF) interpolation technique described in Section 4.3.1.3. Instead of using the full parameter vector, however, only the magnitude of the pumping force is used as the input for the RBF interpolation. Discarding the other parameter dimensions can be justified by the fact that the neutronics cross sections and other heat transfer related parameters only influence the fluid dynamics through the buoyancy term which has a considerably lower impact on the flow field than the pumping force in the problem at hand (operation at nominal power). The results presented later in this chapter justify this approach as well.

Following this, 20 new parameter vectors are sampled for validation purposes. The FOM is then executed using the parameters in the validation set and the results are compared to those obtained from the ROM. The average and maximum relative L^2 errors over the 20-sample validation set are presented in Table 8.5 as function of the truncation limit (τ). This limit is used for the determination of the number of basis functions for each solution field, with one exception. The number of pressure modes is chosen to ensure stability but minimize the pollution effects introduced by corresponding supremizer modes. For more information on this pollution, see Section 4.3.1.1. Therefore, for every case shown below the number of velocity modes is specified using a truncation limit and the corresponding number of pressure modes (and supremizer modes) is chosen to maximize the accuracy in velocity.

In general, we see that the relative L^2 errors decrease with the inclusion of additional basis functions to the reduced subspaces. We also see that the accuracy in the eddy viscosity does not change considerably beyond $\tau = 10^{-6}$. This error is attributed to the approximation error introduced by the RBF interpolation. Nevertheless, the maximum error over the validation set is 1.35% which is considered adequate and justifies the choice of shrinking the parameter space to the pumping force only. Every other field of interest has a maximum L^2 error below 0.52% which is considered good in this scenario. The corresponding convergence in the absolute errors in the

effective multiplication factor is presented by the black curves in Figure 8.5.

Table 8.5: The average and maximum L^2 errors of the representative fields over the validation set in case of steady-state fully-coupled turbulent simulations of the MSFR.

τ	$\overline{e_{\phi_5}}$ (%)	max(e_{ϕ_5}) (%)	$\overline{e_{C_7}}$ (%)	max(e_{C_7}) (%)	$\overline{e_T}$ (%)	max(e_T) (%)
10^{-2}	0.75	1.64	2.82	5.17	0.50	1.24
10^{-4}	0.51	1.14	0.59	1.25	0.50	1.24
10^{-6}	0.17	0.40	0.29	0.53	0.15	0.36
10^{-8}	0.06	0.10	0.42	0.51	0.09	0.12
10^{-10}	0.03	0.05	0.41	0.52	0.07	0.09
τ	$\overline{e_{u_D}}$ (%)	max(e_{u_D}) (%)	$\overline{e_p}$ (%)	max(e_p) (%)	$\overline{e_{\nu_t}}$ (%)	max(e_{ν_t}) (%)
10^{-2}	0.94	1.91	0.48	1.02	1.63	3.17
10^{-4}	0.91	1.86	0.50	1.11	0.73	1.33
10^{-6}	0.40	0.74	0.49	1.10	0.73	1.35
10^{-8}	0.24	0.30	0.09	0.17	0.73	1.35
10^{-10}	0.26	0.31	0.10	0.17	0.73	1.35

We see that before starting to decrease, the error in $\mathrm{k_{eff}}$ increased when the truncation limit is decreased to $\tau = 10^{-5}$. This increase is, however, not present in the L^2 errors. The reason behind this behavior is the difference in the decay of the eigenvalues of the correlation matrices built for the temperature and the logarithmic temperature, shown in Figure 8.1. It is visible that the same truncation limit results in different number of modes for these two fields. This means that while the density feedback (driven by the temperature) is treated properly by including additional local effects by each additional basis function, the fuel temperature (Doppler, driven by the logarithmic

temperature) feedback cannot handle these local effects because the corresponding spatial basis functions are not included yet. As we start including more than 1 basis function for the logarithmic temperature at $\tau = 10^{-7}$, the errors start to decrease considerably.

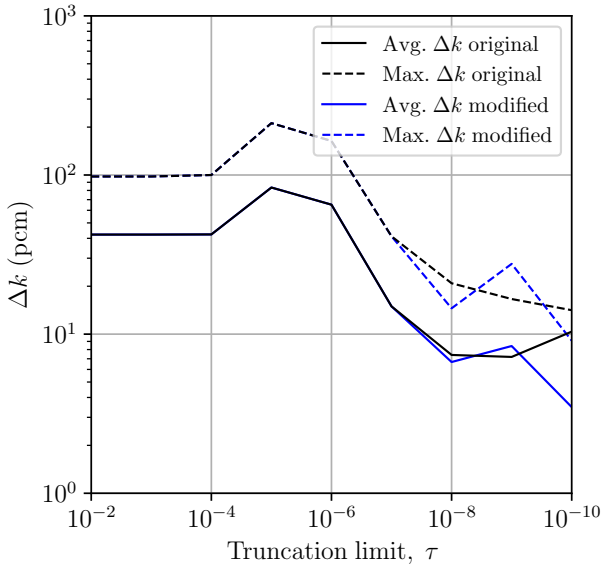


Figure 8.2: Decay of the maximum and average absolute error in k_{eff} using the original method (black curves) and the lower dimensional fluid ROM (blue curves).

At this point, we recall that the inclusion of too many velocity modes can also decrease the accuracy due to various effects like using upwind schemes for the precursor end enthalpy transport equations. For this reason, a test has been carried out where the number of basis functions used for the velocity, pressure, supremizer and flow resistance fields beyond $\tau = 10^{-7}$ are fixed to the ones determined at $\tau = 10^{-7}$. This scenario is depicted by the blue curves in Figure 8.1. It is visible that using fewer basis functions for the fluid dynamics fields yields more accurate ROMs both at $\tau = 10^{-8}$ and $\tau = 10^{-10}$, with $\tau = 10^{-9}$ being slightly worse than the original scenario. For 10^{-10} energy retention limit, specifically, the original, truncation limit based ROM gives average and maximum errors of 10.37 pcm and 14.14 pcm, while the modified ROM gives 3.48 pcm and 9.1 pcm for the same, respectively.

The L^2 errors which correspond to the modified (lower-order fluid ROM) multiphysics ROMs are presented in Table 8.6. It is visible that the modified ROMs yield better L^2 errors in terms of every solution field except temperature which is marginally better in case of the original ROM.

Table 8.6: The average and maximum L^2 errors of the representative fields over the validation set in case of steady-state fully-coupled turbulent simulations of the MSFR using the lower-dimensional fluid ROM.

τ	$\overline{e_{\phi_5}}$ (%)	max(e_{ϕ_5}) (%)	$\overline{e_{C_7}}$ (%)	max(e_{C_7}) (%)	$\overline{e_T}$ (%)	max(e_T) (%)
10^{-8}	0.03	0.05	0.31	0.38	0.09	0.14
10^{-10}	0.02	0.04	0.31	0.38	0.08	0.13
τ	$\overline{e_{u_D}}$ (%)	max(e_{u_D}) (%)	$\overline{e_p}$ (%)	max(e_p) (%)	$\overline{e_{\nu_t}}$ (%)	max(e_{ν_t}) (%)
10^{-8}	0.15	0.21	0.10	0.16	0.73	1.35
10^{-10}	0.15	0.21	0.10	0.16	0.73	1.35

Based on the results in this section, the modified ROM which corresponds to $\tau = 10^{-7}$ for the fluid dynamics fields and $\tau = 10^{-10}$ for every other field has been selected for uncertainty quantification and sensitivity analysis purposes. We assume that the 3 pcm mean and 9 pcm maximum error in the multiplication factor is much lower compared to the variation in the parameter space, which is on the order of 10,000 pcm. Besides the effective multiplication factor, this ROM yields good results in the reconstruction of the other field variables as well. The dimensions of the subspaces used in the selected ROM are summarized in Table 8.7. The (+2) in case of the velocity field denotes that two additional supremizer basis functions have been used. Furthermore, we see that the final subspaces utilize considerably more basis functions compared to the steady-state zero-power experiments in Chapter 6. This is not surprising, since the addition of new uncertain model parameters and the inclusion of temperature-dependent cross sections introduce additional

spatial variation in the solution fields.

Table 8.7: The ranks of the subspaces used for the construction of the final ROMs for turbulent fully-coupled multiphysics simulations of the MSFR. The ranks correspond to $\tau = 10^{-7}$ for the fluid dynamics fields and $\tau = 10^{-10}$ for every additional field.

Field	**Rank**	**Field**	**Rank**	**Field**	**Rank**	**Field**	**Rank**	**Field**	**Rank**
ϕ_1	15	ϕ_6	16	C_5^*	16	p	2	$\log(T)$	6
ϕ_2	15	C_1^*	9	C_6^*	17	$\boldsymbol{F}_{fr}$	4		
ϕ_3	14	C_2^*	11	C_7^*	18	ν_t	8		
ϕ_4	15	C_3^*	12	C_8^*	17	α_t	8		
ϕ_5	15	C_4^*	14	$\boldsymbol{u}_D$	4 (+2)	T	10		

Additionally, Figures 8.3 and 8.4 present the spatial distribution of the errors in the representative fields at the first parameter point in the validation set using the selected ROM. Figure 8.3 shows the errors in the velocity and temperature fields together with the corresponding solution fields of the FOM at the given parameter sample. It is visible that the maximum error in the velocity is approximately 3 orders of magnitude lower than the characteristic velocity of the original solution. The same is true for the temperature, where the maximum error is 2.73 K, while the maximum temperature in the system is 1,268 K. Following this, Figure 8.4 shows the solutions and corresponding errors for the neutron scalar flux in energy group 5 together with the concentration of delayed neutrons in group 7. It can be observed that the maximum errors in the flux are approximately 4 orders of magnitude lower than the characteristic flux values at the given sample. Most of the error is concentrated close to the central line of the reactor. A possible explanation for this can be the inaccuracy in the temperature in this region. The precursor concentrations, however, show considerably higher errors that are only 2 orders of magnitude lower than the characteristic concentrations of the FOM. The bulk of the error is concentrated in close vicinity of the axis plane

and the top reflector.

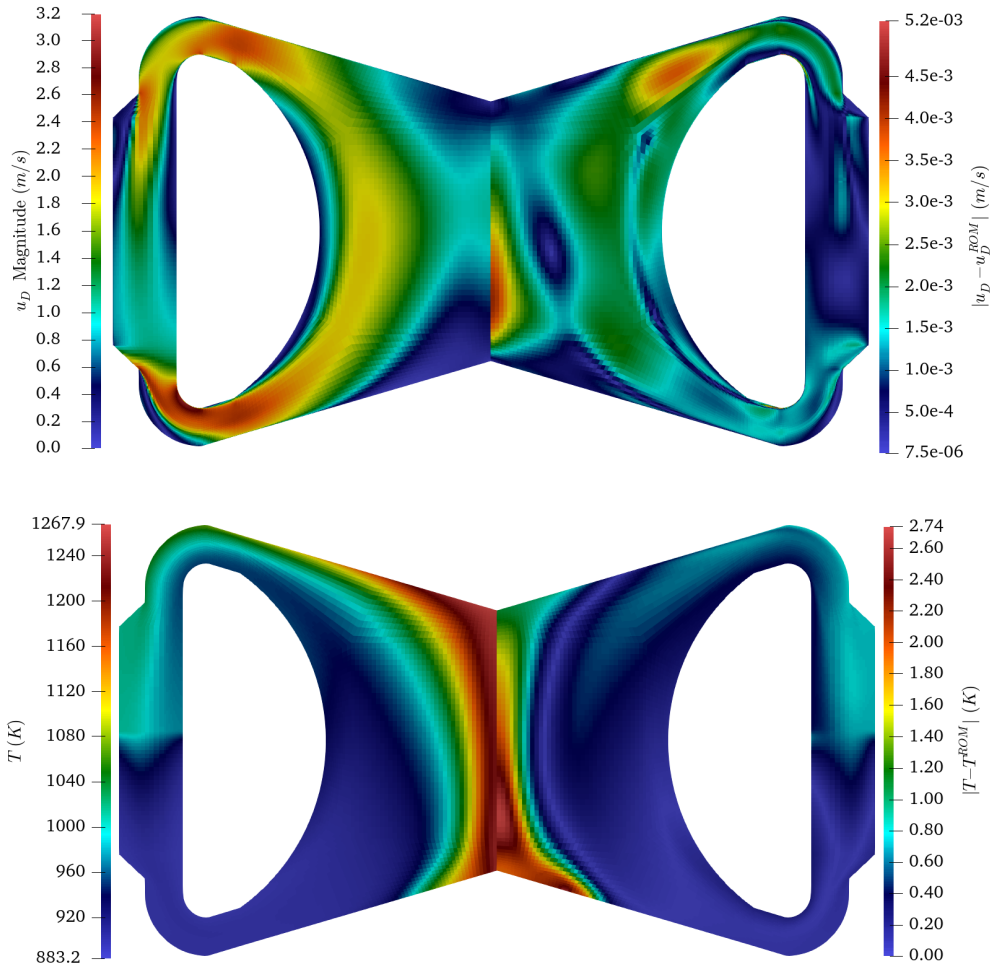


Figure 8.3: The velocity and temperature fields of the FOM (left) together with the absolute errors between the FOM and the ROM (right) for the first sample in the validation set used for the fully-coupled simulations of the MSFR.

At this point we can compare the performance of the generated ROMs to the similarly POD-based, but non-intrusive, sparse-grid interpolation ROMs used for a similar experiment in [25]. The non-intrusive ROMs are capable of reconstructing the temperature field with a maximum l^2 error of 0.02% and the effective multiplication factor with a maximum absolute error of 37 pcm. The same are 0.13% and 9.1 pcm for the intrusive ROM discussed here. An interesting difference,

however, is that while the non-intrusive ROM requires 63 and 1,639 model evaluations for the temperature and k_{eff}, the intrusive ROM needs only 30. We note, however, that the two examples are not one-to-one comparable due to the differences in the used FOMs and parameters.

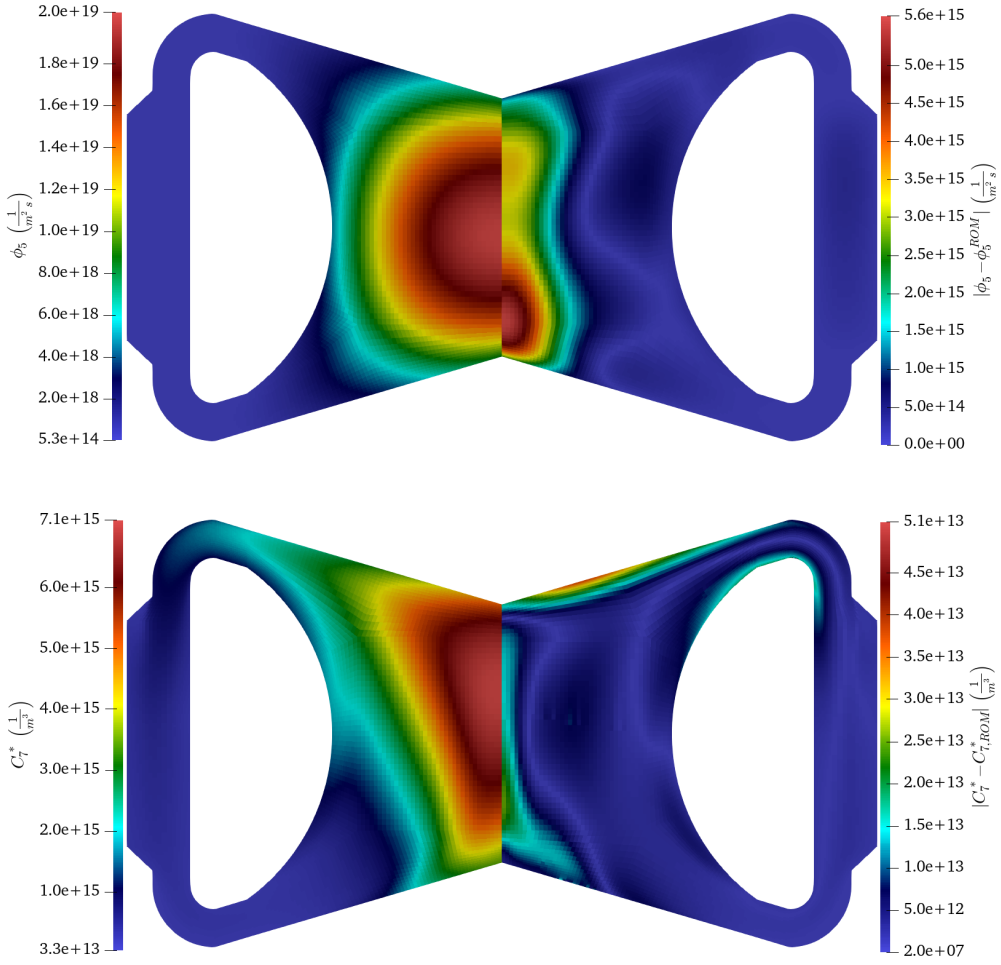


Figure 8.4: The representative scalar flux and precursor concentration of the FOM (left) together with the absolute errors between the FOM and the ROM (right) for the first sample in the validation set used for the fully-coupled simulations of the MSFR.

Lastly, the time required to solve the selected ROM is compared to that of the FOM. Due to the slow convergence and the fact that a transient simulation is used to obtain the steady state of the FOM, it takes a considerable, 21,000 s on average to solve the FOM per parameter sample on

a single core of a processor. The solution time of the ROM, on the other hand, is only 1.25 s on average. This yields an approximate speed-up factor of 16,800. This speed-up is the consequence of decreasing the number of degrees of freedom in the problem from 274,380 to 256.

8.3 Uncertainty Quantification and Sensitivity Analysis of the Fully-coupled Problem

Once the ROM is generated, it can be used as an emulator of the FOM for the purpose of uncertainty quantification and sensitivity analysis. First, we attempt to propagate the uncertainties in input parameters to the quantities of interest (QoIs). For the list of uncertain model parameters and their corresponding distributions, see Section 8.1. We note that the selected group constants have physical dependencies. A good example is the fission cross section (Σ_f) which, in a realistic scenario, would influence the removal cross section (Σ_r) and the diffusion coefficient (D) as well. In this example, however, our goal is to demonstrate the applicability of the ROMs for multi-query problems, therefore these dependencies are not resolved. Recall that two QoIs are defined for this problem, the effective multiplication factor ($\mathrm{k_{eff}}$) and the maximum temperature in the system. Since the ROM is orders of magnitude faster than the FOM, a Monte Carlo approach has been used to generate the distributions and the corresponding important statistical moments of the QoIs. Altogether 50,000 random samples have been used for the experiment. The resulting histograms for the two QoIs are plotted in Figure 8.5. The distributions are compared to a Gaussian using the normality plots in Figure 8.6. It is visible that the distribution of $\mathrm{k_{eff}}$ is seemingly closer to Gaussian, even though it is slightly skewed, especially in the tail regions. Nevertheless, this suggests that the QoI is influenced considerably by more than one model parameter. The maximum temperature, on the other hand, resembles a distorted uniform distribution which implies that it is influenced by predominantly only one model parameter. This will be confirmed later in this section. The corresponding statistical moments estimated using 50,000 samples are presented in Table 8.8. The confidence intervals of the statistical moments have also been estimated using a bootstrap (re-sampling with replacement) method with 5,000 cycles in this case. It is visible that the uncertainty on the effective multiplication factor is considerable with the standard deviation being close to 10,000 pcm. This emphasizes the importance of of accurate cross section libraries, since the used

uniform distributions with a $\pm 10\%$ variation result in a highly unreliable effective multiplication factor. The same can be stated about the maximum temperature.

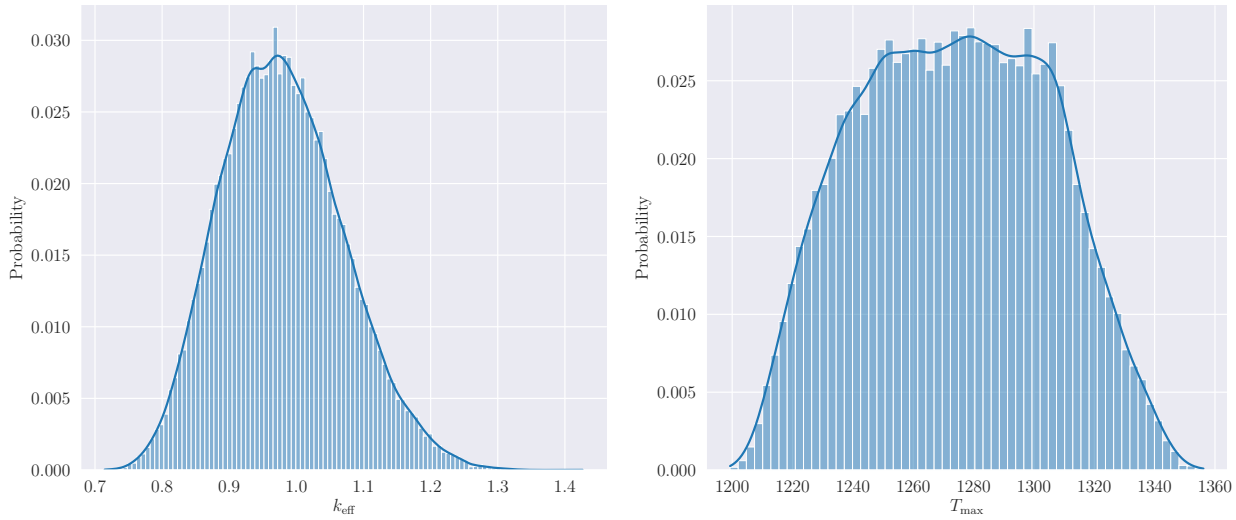


Figure 8.5: The histograms of the effective multiplication factor (left) and maximum temperature (right) for the fully-coupled problem using 50,000 parameter samples.

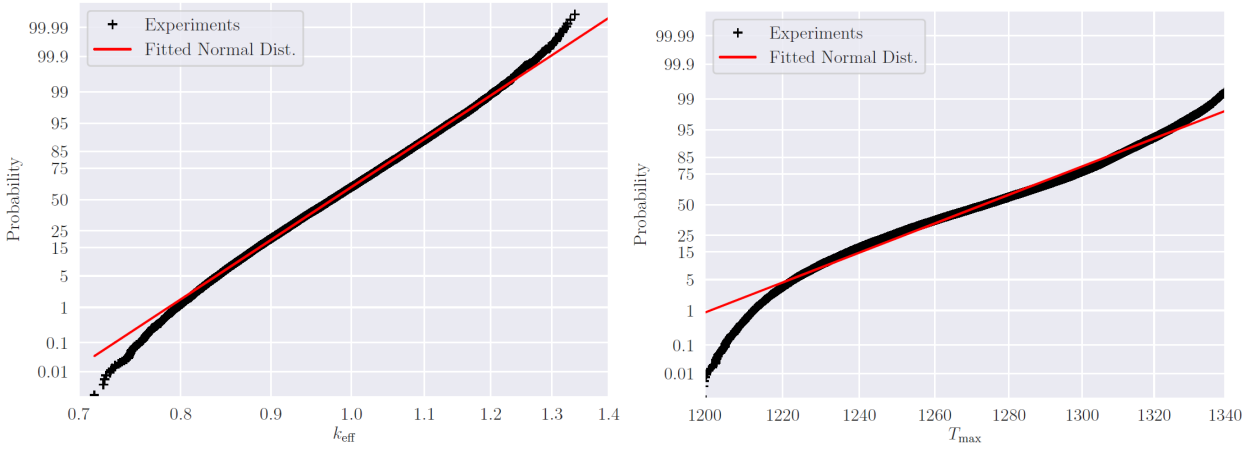


Figure 8.6: Normality plots of the effective multiplication factor (left) and the maximum temperature (right) in the fully coupled system using 50,000 parameter samples.

Table 8.8: Mean and standard deviation of $\mathrm{k_{eff}}$ and $T_{\max}$ together with the corresponding confidence intervals for the steady-state simulations of the fully-coupled problem.

$\mathbf{k_{eff}}$			
Mean	**95% CI**	**Std. Dev.**	**95% CI**
0.979730	[0.978934, 0.980525]	0.090254	[0.090254, 0.091316]
$\boldsymbol{T_{\max}}$ **(K)**			
Mean	**95% CI**	**Std. Dev.**	**95% CI**
1273.46	[1273.18, 1273.74]	31.47	[31.32, 31.61]

Following the analysis of the distributions and statistical moments, we turn our attention to the determination of the parameter-wise contributions to the uncertainty in the QoIs. This is supplemented by the sensitivity study of the QoIs. To perform these studies, the strategy discussed in Section 5.5 is employed. In this process, a Polynomial Chaos Expansion (PCE) is fitted onto the data of 50,000 samples. The PCE can then be used to generate global and local sensitivity indicators. The residuals and relative errors of the fitted PCE over the training set are summarized in Table 8.9.

Table 8.9: Quantities of interest together with the residuals and relative errors of the fitted PCE over the training set in case of steady-state fully-coupled simulations with the MSFR.

QoI	**Max. degree**	**Number of terms**	**Residual**	**Relative Error**
$\mathrm{k_{eff}}$	3	2,600	1.7×10^{-7}	1.7×10^{-7}
$\mathrm{k_{eff}}$	4	17,500	3.1×10^{-8}	5.8×10^{-9}
$T_{\max}$	3	2,600	3.6×10^{-3}	6.4×10^{-4}
$T_{\max}$	4	17,500	2.8×10^{-3}	4.1×10^{-4}

It is visible that both the residuals and relative errors are low in case of the effective multiplication factor. The approximation of the maximum temperature, on the other hand shows higher relative errors. Moreover, increasing the maximum degree of the polynomial expansion does not seem to reduce the error considerably. Increasing the maximum degree beyond 4 is not possible for the PCE since then number of terms would exceed the number of parameter samples. An explanation of this behavior can be that the $T_{\max} = T_{\max}(\boldsymbol{\mu})$ function is not smooth enough. Nevertheless, for this example, we consider the accuracy of the fitted PCE adequate. For more accurate results, adaptive PCE can be used or more training samples can be added to allow higher polynomial degrees.

The contribution of the variance in the model parameters to the variance in the QoIs can be analyzed using the corresponding Sobol Indices. These indices can be generated using the already fitted PCE. Figure 8.7 presents the Sobol Indices computed for the effective multiplication factor.

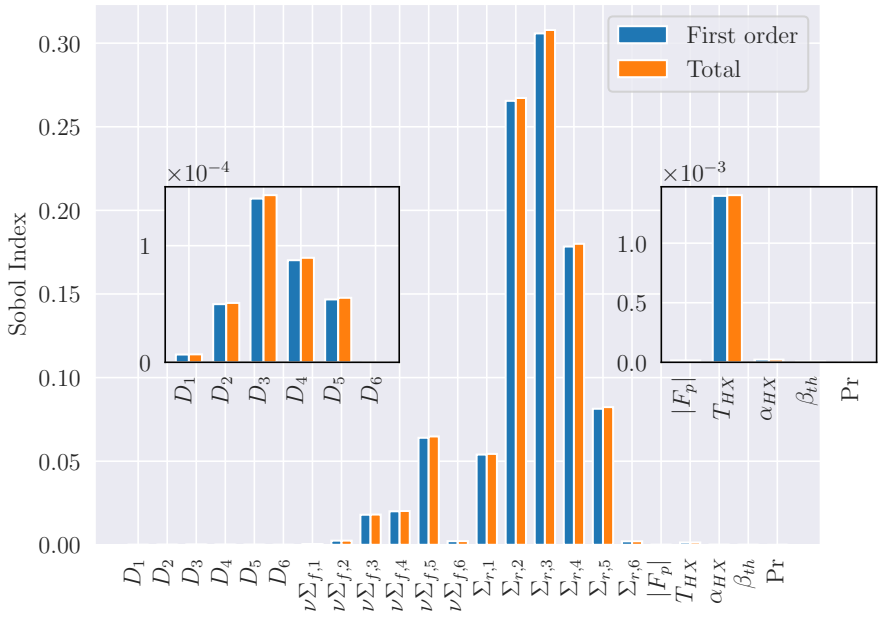


Figure 8.7: First order and total Sobol Indices of the effective multiplication factor for the fully-coupled MSFR model.

We see that the variance in the removal cross sections has the strongest impact on the variance of $\mathrm{k_{eff}}$. Following this, come the fission cross sections, external coolant temperature and the diffusion coefficients. The fact that the other thermophysical properties have little impact is not surprising, since the neutronics equations are expected to have the highest sensitivity to the model parameters directly used in them. To further investigate, the local sensitivity of the effective multiplication factor to the parameters at the middle of the parameter space are summarized in Table 8.10.

Table 8.10: The local sensitivity coefficients of $\mathrm{k_{eff}}$ at the middle point of the parameter space.

$\boldsymbol{\mu_k}$	$\boldsymbol{\kappa_{\mu_k}^{\mathrm{k_{eff}}}}$	$\boldsymbol{\mu_k}$	$\boldsymbol{\kappa_{\mu_k}^{\mathrm{k_{eff}}}}$	$\boldsymbol{\mu_k}$	$\boldsymbol{\kappa_{\mu_k}^{\mathrm{k_{eff}}}}$	$\boldsymbol{\mu_k}$	$\boldsymbol{\kappa_{\mu_k}^{\mathrm{k_{eff}}}}$
D_1	-4.03$\times10^{-3}$	$\nu\Sigma_{f,1}$	3.12$\times10^{-2}$	$\Sigma_{r,1}$	-3.68$\times10^{-1}$	$\lvert F_p\rvert$	2.53$\times10^{-3}$
D_2	-1.12$\times10^{-2}$	$\nu\Sigma_{f,2}$	8.01$\times10^{-2}$	$\Sigma_{r,2}$	-8.17$\times10^{-1}$	T_{HX}	-1.11$\times10^{-1}$
D_3	-1.87$\times10^{-2}$	$\nu\Sigma_{f,3}$	2.14$\times10^{-1}$	$\Sigma_{r,3}$	-8.77$\times10^{-1}$	α_{HX}	3.76$\times10^{-3}$
D_4	-1.48$\times10^{-2}$	$\nu\Sigma_{f,4}$	2.26$\times10^{-1}$	$\Sigma_{r,4}$	-6.69$\times10^{-1}$	β_{th}	-4.23$\times10^{-3}$
D_5	-1.16$\times10^{-2}$	$\nu\Sigma_{f,5}$	4.02$\times10^{-1}$	$\Sigma_{r,5}$	-4.51$\times10^{-1}$	Pr	4.16$\times10^{-5}$
D_6	-8.43$\times10^{-3}$	$\nu\Sigma_{f,6}$	7.40$\times10^{-2}$	$\Sigma_{r,6}$	-7.21$\times10^{-2}$		

By looking at the signs of the sensitivity coefficients, we see that the increase in diffusion coefficients and removal cross sections decrease the effective multiplication factor, while the opposite is true for the fission cross sections. Regarding the magnitudes, a similar behavior can be observed as in case of the Sobol Indices. We see that $\mathrm{k_{eff}}$ is most sensitive to the removal cross sections. Interestingly, the sign of the sensitivity coefficient of the pumping force is positive, meaning that the increase in pumping power leads to the increase in $\mathrm{k_{eff}}$. This is in contradiction with the zero-power simulations where the increase in pumping power leads to decreasing precursor concentration in the high-importance regions of the reactor core which, in turn, results in a decrease in $\mathrm{k_{eff}}$ as well. In this scenario, however, the increase in velocity does not only increases the loss of delayed neu-

tron precursors but improves the cooling of the system which results (through negative feedback) in a net increase in k_{eff}.

The same Sobol Index and Sensitivity Analyses have been carried out for the maximum temperature. The Sobol Indices are presented in Figure 8.8. It is visible that the variation in the maximum temperature is predominantly influenced by the variation in the pumping force and the external coolant temperature in the heat exchanger. This has been expected based on the shape of the histogram of the maximum temperature. Other notable contributors are the thermal expansion coefficient and the heat transfer coefficient in the heat exchanger.

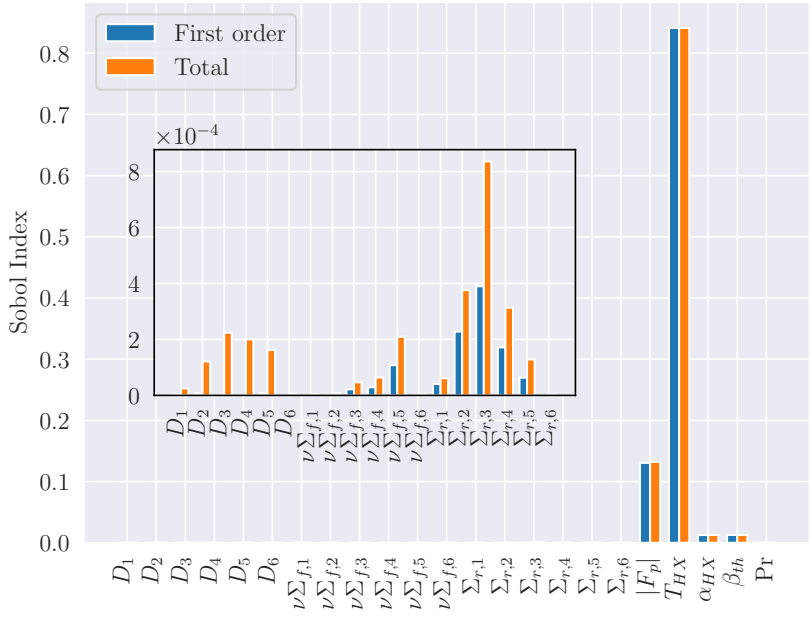


Figure 8.8: First order and total Sobol Indices of the maximum temperature in case of fully-coupled simulations.

It can also be observed that the Sobol Indices corresponding to the cross sections have negligible magnitudes. This is not surprising, since throughout the eigenvalue iteration in the neutronics solver, the solution fields are normalized to ensure that the reactor power is maintained at the nom-

inal level. This means that the source term in the enthalpy equation does not change considerably. On the other hand, the contribution of the variance in β_{th} is noticeable since it changes the shape of the flux and power distributions through the density feedback and contributes to the changing flow profiles through buoyancy effects. For the exact dependence, see Section 3.2.

The local sensitivity coefficients at the midpoint of the parameter space for each parameter are summarized in Table 8.11. We see that the increase in the pumping force, heat exchange coefficient and the thermal expansion coefficient all result in the decrease in maximum temperature. All of these parameters influence directly or indirectly (through the velocity and flux distributions) the cooling of the system. As expected, the QoI is most sensitive to the external coolant temperature.

Table 8.11: Local sensitivity coefficients of the maximum temperature at the middle of the parameter space in case of the fully-coupled problem.

$\boldsymbol{\mu}$	$\boldsymbol{\kappa}_{\boldsymbol{\mu_k}}^{\boldsymbol{T}_{\max}}$	$\boldsymbol{\mu}$	$\boldsymbol{\kappa}_{\boldsymbol{\mu_k}}^{\boldsymbol{T}_{\max}}$	$\boldsymbol{\mu}$	$\boldsymbol{\kappa}_{\boldsymbol{\mu_k}}^{\boldsymbol{T}_{\max}}$	$\boldsymbol{\mu}$	$\boldsymbol{\kappa}_{\boldsymbol{\mu_k}}^{\boldsymbol{T}_{\max}}$
D_1	-9.67$\times 10^{-4}$	$\nu\Sigma_{f,1}$	2.24$\times 10^{-4}$	$\Sigma_{r,1}$	-2.91$\times 10^{-3}$	$\lvert F_p \rvert$	-6.77$\times 10^{-2}$
D_2	-2.78$\times 10^{-3}$	$\nu\Sigma_{f,2}$	3.78$\times 10^{-4}$	$\Sigma_{r,2}$	-8.34$\times 10^{-3}$	T_{HX}	7.08$\times 10^{-1}$
D_3	-2.96$\times 10^{-3}$	$\nu\Sigma_{f,3}$	2.74$\times 10^{-3}$	$\Sigma_{r,3}$	-1.19$\times 10^{-2}$	α_{HX}	-2.37$\times 10^{-2}$
D_4	-2.21$\times 10^{-3}$	$\nu\Sigma_{f,4}$	2.73$\times 10^{-3}$	$\Sigma_{r,4}$	-7.27$\times 10^{-3}$	β_{th}	-4.58$\times 10^{-2}$
D_5	-4.96$\times 10^{-4}$	$\nu\Sigma_{f,5}$	6.52$\times 10^{-3}$	$\Sigma_{r,5}$	-4.58$\times 10^{-3}$	Pr	3.32$\times 10^{-4}$
D_6	2.27$\times 10^{-4}$	$\nu\Sigma_{f,6}$	4.60$\times 10^{-4}$	$\Sigma_{r,6}$	-2.43$\times 10^{-4}$		

Lastly, the savings in computation time resulted by the utilization of the ROMs is evaluated for the uncertainty quantification and sensitivity analysis. For this, we use the formula introduced in Eq. (6.2). An average execution time of 21,000 s is assumed for the FOM, while 1.25 s is used for the ROM. By plugging these numbers into Eq. (6.2), the overall approximate speedup, including the training phase, becomes $SF = 1,516$. Based on this, we conclude that the utilization of ROMs for the uncertainty quantification tasks can be highly beneficial.

9. CONCLUSIONS AND OUTLOOK

9.1 Summary

The work presented in this dissertation focuses on the development and implementation of intrusive multiphysics Reduced-Order Models (ROMs) for parametric problems in Molten Salt Reactors (MSRs). The motivation behind the generation of these ROMs is that high-fidelity Full-Order Models (FOMs) for complex multiphysics problems are computationally demanding, therefore they are are not suitable for multi-query problems like uncertainty quantification or design optimization. These multi-query problems require the repeated solution of the computationally expensive Full-Order Models (FOM) at a potentially large number of model parameter combinations. In this work, we propose that training ROMs and using them as emulators of the FOM can greatly accelerate such multi-query applications. The developed ROMs are based on the intrusive Proper Orthogonal Decomposition aided Reduced Basis technique (POD-RB), where the intrusive nature of the technique denotes that the discretized operators of the original, computationally expensive FOMs are used to create the ROMs. In the following paragraphs we summarize the main take-aways of each chapter in this dissertation. Whenever possible, we emphasize the novelty in the presented work.

Chapter 2 reviews the process of the generation of intrusive POD-RB ROMs using a one-dimensional diffusion-reaction example. In general, the construction of the ROMs starts with the collection of information using the system at hand. This means that snapshots of the solution fields are saved at different times and parameter values. Using the snapshots, reduced subspaces are generated for these solution fields with the help of POD. The original solutions are then expanded using the basis vectors of these subspaces. The final form of the ROMs can be derived by plugging in the expansions into the original mathematical model and projecting it using the basis vectors of the created subspaces (Galerkin projection). For the handling of non-quadratic nonlinearities, the Discrete Empirical Interpolation Method (DEIM) has been employed.

Chapter 3 presents the system of governing equations used for the description of the behavior of liquid-fuel MSRs together with the discretization techniques used to derive the FOMs. This work considers the incompressible Reynolds-Averaged Porous Medium Navier-Stokes equations for the description of fluid dynamics phenomenon. This is coupled with a porous medium enthalpy equation for the determination of temperature. The neutronics behavior of the system is described by the multigroup neutron diffusion equations supplemented with delayed neutron precursor balance equations in porous media. The neutronics equations use temperature-dependent group constants and consider the advection of the precursors. The buoyancy effects in the fluid are treated using a Boussinesq approximation. In this work, the governing equations are discretized using the cell-centered Finite Volume Method (FVM), while transient examples utilize implicit Euler discretization in time. GeN-Foam, an `OpenFOAM`©-based open-source multiphysics solver has been employed for the generation and solution of the FOMs. The nonlinearities in the system are resolved using a multi-layer fixed-point iteration between the fluid dynamics, heat transfer and neutronics subproblems.

Chapter 4 then presents the derivation of the corresponding multiphysics ROMs. More specifically, it describes in details the methods used for building the reduced subspaces for the solution fields, the generation of stable physics-wise ROMs and the multiphysics coupling between them. To the best knowledge of the author, this chapter contains the following novel contribution to the field of reduced-order modeling of nuclear systems:

- The derivation of POD-RB-ROMs for the incompressible Reynolds-Averaged Porous Medium Navier-Stokes equations with flow resistances included. This expands on the work published in [3] by the author.

- The derivation of a neutronics ROM based on group-wise reduction and the porous medium equations describing the delayed neutron precursor concentrations. This expands on the work published in [88] by the author.

- The derivation of a reduced-order system for the determination of the effective delayed neu-

tron fraction, involving the derivation of the reduced-order adjoint neutronics equations for liquid-fuel systems. The proposed method does not require any full-order operations, therefore is highly efficient in terms of computation time.

- The derivation of an efficient way of treating the temperature-dependent cross sections based on DEIM. The method is capable of handling the density and Doppler feedback separately. The presented method is efficient in terms of memory usage, extrapolability and computation cost. This expands on the method published in [91] by the author.

Chapter 5 presents the evaluation phase of the ROM. This includes the assemly and the solution of the reduced multiphysics system together with the extraction of the quantities of interest. Besides these, the chapter discusses methods to evaluate the accuracy of the ROMs and a possible approach to use the ROMs for uncertainty quantification and sensitivity analysis. To the best of the knowledge of the author this section contains the following novel contributions:

- The implementation of a multiphysics POD-RB-ROM framework (named GeN-ROM) using `OpenFOAM`©. The developed code can be used for simulations based on GeN-Foam, but also with other OpenFOAM-based solvers.

- The derivation of a multi-layered fixed point iteration scheme at reduced order level that yields stable multiphysics ROMs.

Following the details of the derivation and implementation of the multiphysics ROMs, several numerical examples are presented. The example in Chapter 6 demonstrates the applicability of the multiphysics ROMs to zero-power parametric simulations on a 2D model of the Molten Salt Fast Reactor (MSFR). Due to the zero-power assumption, only the neutronics and fluid dynamics subproblems are solved, the enthalpy equation does not need to be considered. In other words, the temperature-dependence of the cross sections and the buoyancy effects in the fluid dynamics subproblem are not taken into account. This example involves a steady-state problem with 13 uncertain parameters and a transient scenario with two uncertain parameters. The quantities of

interest are the effective multiplication factor and delayed neutron fraction for steady-state simulations and the power level at $t = 6\ s$ for the transient scenario. The accuracy of the generated ROMs are considered to be adequate in terms of both the solution fields and the quantities of interest. The steady-state ROMs, depending on the case, are capable of a single run speedup of 2.3×10^5-4.7×10^5, while the same for the transient ROMs was 78-870. The generated ROMs are then used for uncertainty quantification and sensitivity analysis of the given systems. The results indicate that the ROMs can be efficient emulators of the FOM, and can reduce the computation cost of the uncertainty quantification task by 1-3 orders of magnitude.

Chapter 7 presents another example of a partially coupled system. In this example, the fluid dynamics fields are considered to be fixed, therefore only the neutronics and heat transfers subproblems are solved. This example is specifically designed for the evaluation of the developed reduced-order method for the handling of temperature-dependent cross sections. The problem involves parameterized steady-state simulations of a 2D model created based on the Molten Salt Reactor Experiment (MSRE). Altogether 6 uncertain parameters are used in this scenario with the quantities of interest being the effective multiplication factor and maximum temperature in the system. The results indicate that the developed multiphysics ROMs yield accurate results and a considerable single-run approximate speedup of 3000. The generated ROM is then used for the uncertainty quantification and sensitivity analysis of the MSRE and the results indicate that the methodology can, in fact, be used for the acceleration of similar multi-query problems.

The last numerical example, presented in Chapter 8, demonstrates a case where all three of the subproblems are solved, coupled with each other. The model consists of a parameterized steady-state simulation using a 2D geometry of the MSFR. The uncertain model parameters in this scenario are group constants, parameters of the heat exchanger and the pump and other thermophysical properties. Altogether 23 model parameters are used. Similarly to the example of the MSRE, the quantities of interest are the effective multiplication factor and the maximum temperature of the system. The results indicate that the developed method can be used to generate accurate ROMs with single-run speedups on the order of 1000. The generated ROMs are then used

as emulators for the uncertainty quantification and sensitivity analysis of the system. Compared to other, recently developed, non-intrusive ROM techniques applied to the simulation of MSRs, the method in this paper yields similar accuracy with considerably fewer samples (snapshots in our case) needed for the training. Although, it must be mentioned that the used FOMs and uncertain parameters are not entirely comparable in the two works.

Overall, we conclude that the developed multiphysics ROM technique shows promising results for the steady-state and transient simulations of Molten Salt Reactors. Most likely, the main reason behind this is that MSRs are highly homogeneous systems, therefore the solution fields can be easily approximated by global basis functions.

9.2 Possible Improvements

Lastly, we give suggestions on possible further enhancements on the developed technique. These suggestions are made based on the difficulties faced during the simulations shown in Chapters 6, 7 and 8 and paper [3]:

- The one-equation fluid ROM and the supremizer stabilization-based two-equation fluid ROM approaches are not robust for turbulent simulations. We propose using a Discrete Empirical Interpolation method to solve a ROM for the numerically consistent segregated pressure equation derived for the SIMPLE/PIMPLE algorithms.

- The utilization of Radial Basis Function (RBF) interpolation for the determination of the coefficients of the eddy viscosity and diffusivity for the two-equation ROMs is not robust enough. In many scenarios, if the sampling of the parameters is not dense enough, the ROMs may exhibit stability issues due to the inaccuracy in the turbulent dissipation. For this reason, we propose either using a different, more robust multi-dimensional interpolation technique (e.g., neural networks or nearest neighbors) or reducing the auxiliary equations used for the determination of the turbulent viscosity and diffusivity. A possible approach would be the reduction of the $k - \epsilon$ equations using Discrete Empirical Interpolation Method (DEIM).

- The treatment of the advection terms in case of the passive scalar transport equations can considerably influence the accuracy of the ROM. If the FOM utilizes a first-order upwind scheme, the discretization depends on the velocity. This means that the quadratic approach used to generate the advection tensor in the ROMs would include errors due to the fact that different velocity modes will result in different discretization stencils compared to what the actual solution would use. This may not cause problems for high-resolution meshes, but it should be accounted for in case of simulations using a coarse-mesh. For this reason, we propose the utilization of the Discrete Empirical Interpolation method for resolving the advection terms in such cases.

REFERENCES

[1] R. Robertson, "MSRE design and operations report. part I. Description of reactor design," tech. rep., Oak Ridge National Lab., Tenn., 1965.

[2] I. Pioro, *Handbook of generation IV nuclear reactors*. Woodhead Publishing, 2016.

[3] P. German, M. Tano, J. C. Ragusa, and C. Fiorina, "Comparison of Reduced-Basis techniques for the model order reduction of parametric incompressible fluid flows," *Progress in Nuclear Energy*, vol. 130, p. 103551, 2020.

[4] C. Fiorina, *The Molten Salt Fast Reactor as a Fast-Spectrum Candidate for Thorium Implementation*. PhD thesis, Politecnico di Milano, 2013. `https://www.politesi.polimi.it/bitstream/10589/74324/1/2013_03_PhD_Fiorina.pdf`.

[5] J. Shi and M. Fratoni, "Gen-Foam Model and Benchmark of Delayed Neutron Precursor Drift in the Molten Salt Reactor Experiment," in *EPJ Web of Conferences*, vol. 247, p. 06040, EDP Sciences, 2021.

[6] R. Kedl, "Fluid Dynamic Studies of the Molten-Salt Reactor Eexperiment (MSRE) Core.," tech. rep., Oak Ridge National Lab., Tenn., 1970.

[7] C. Fiorina, D. Lathouwers, M. Aufiero, A. Cammi, C. Guerrieri, J. L. Kloosterman, L. Luzzi, and M. E. Ricotti, "Modelling and nalysis of the MSFR transient behaviour," *Annals of Nuclear Energy*, vol. 64, pp. 485–498, 2014.

[8] C. E. Rasmussen and C. Williams, "Gaussian processes for machine learning, vol. 1," *MIT press*, vol. 39, pp. 40–43, 2006.

[9] A. Wilson and R. Adams, "Gaussian process kernels for pattern discovery and extrapolation," in *International conference on machine learning*, pp. 1067–1075, 2013.

[10] R. Von Mises, *Mathematical theory of probability and statistics*. Academic Press, 2014.

[11] C. E. Rasmussen, "Gaussian processes in machine learning," in *Summer School on Machine Learning*, pp. 63–71, Springer, 2003.

[12] X. Wu, T. Kozlowski, H. Meidani, and K. Shirvan, "Inverse uncertainty quantification using the modular Bayesian approach based on Gaussian Process, Part 2: Application to TRACE," *Nuclear Engineering and Design*, vol. 335, pp. 417–431, 2018.

[13] B. A. Khuwaileh and W. A. Metwally, "Gaussian Process approach for dose mapping in radiation fields," *Nuclear Engineering and Technology*, 2020.

[14] P. Baraldi, F. Mangili, and E. Zio, "A prognostics approach to nuclear component degradation modeling based on Gaussian Process Regression," *Progress in Nuclear Energy*, vol. 78, pp. 141–154, 2015.

[15] D. Xiu and G. E. Karniadakis, "The Wiener–Askey polynomial chaos for stochastic differential equations," *SIAM journal on scientific computing*, vol. 24, no. 2, pp. 619–644, 2002.

[16] M. Eldred, C. Webster, and P. Constantine, "Evaluation of non-intrusive approaches for Wiener-Askey generalized polynomial chaos," in *49th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, 16th AIAA/ASME/AHS Adaptive Structures Conference, 10th AIAA Non-Deterministic Approaches Conference, 9th AIAA Gossamer Spacecraft Forum, 4th AIAA Multidisciplinary Design Optimization Specialists Conference*, p. 1892, 2008.

[17] M. Eldred and J. Burkardt, "Comparison of non-intrusive polynomial chaos and stochastic collocation methods for uncertainty quantification," in *47th AIAA aerospace sciences meeting including the new horizons forum and aerospace exposition*, p. 976, 2009.

[18] M. Rising, A. Prinja, and P. Talou, "Prompt fission neutron spectrum uncertainty propagation using polynomial chaos expansion," *Nuclear Science and Engineering*, vol. 175, no. 2, pp. 188–203, 2013.

[19] L. Gilli, D. Lathouwers, J. L. Kloosterman, T. Van der Hagen, A. Koning, and D. Rochman, "Uncertainty quantification for criticality problems using non-intrusive and adaptive polynomial chaos techniques," *Annals of Nuclear Energy*, vol. 56, pp. 71–80, 2013.

[20] Z. Perkó, "Sensitivity and Uncertainty Analysis of Coupled Reactor Physics Problems: Method Development for Multi-Physics in Reactors," 2015.

[21] V. Barthelmann, E. Novak, and K. Ritter, "High dimensional polynomial interpolation on sparse grids," *Advances in Computational Mathematics*, vol. 12, no. 4, pp. 273–288, 2000.

[22] T. Gerstner and M. Griebel, "Dimension–adaptive tensor–product quadrature," *Computing*, vol. 71, no. 1, pp. 65–87, 2003.

[23] W. A. Klimke, *Uncertainty modeling using fuzzy arithmetic and sparse grids*. Citeseer, 2006.

[24] D. Botes and P. M. Bokov, "Polynomial interpolation of few-group neutron cross sections on sparse grids," *Annals of Nuclear Energy*, vol. 64, pp. 156–168, 2014.

[25] F. Alsayyari, M. Tiberga, Z. Perkó, D. Lathouwers, and J. L. Kloosterman, "A nonintrusive adaptive reduced order modeling approach for a molten salt reactor system," *Annals of Nuclear Energy*, vol. 141, p. 107321, 2020.

[26] P. J. Schmid, "Dynamic mode decomposition of numerical and experimental data," *Journal of fluid mechanics*, vol. 656, pp. 5–28, 2010.

[27] J. H. Tu, C. W. Rowley, D. M. Luchtenburg, S. L. Brunton, and J. N. Kutz, "On dynamic mode decomposition: Theory and applications," *arXiv preprint arXiv:1312.0041*, 2013.

[28] R. G. McClarren, "Calculating time eigenvalues of the neutron transport equation with dynamic mode decomposition," *Nuclear Science and Engineering*, vol. 193, no. 8, pp. 854–867, 2019.

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

[30] R. W. Freund, "Reduced-order modeling techniques based on Krylov subspaces and their use in circuit simulation," in *Applied and computational control, signals, and circuits*, pp. 435–498, Springer, 1999.

[31] Z. Bai, "Krylov subspace techniques for reduced-order modeling of large-scale dynamical systems," *Applied numerical mathematics*, vol. 43, no. 1-2, pp. 9–44, 2002.

[32] G. Baker and P. Graves-Morris, "Padé approximants, cambridge university press, cambridge," 1996.

[33] P. Feldmann and R. W. Freund, "Efficient linear circuit analysis by Padé approximation via the Lanczos process," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 14, no. 5, pp. 639–649, 1995.

[34] Y. Yue and K. Meerbergen, "Parametric model order reduction of damped mechanical systems via the block Arnoldi process," *Applied Mathematics Letters*, vol. 26, no. 6, pp. 643–648, 2013.

[35] Y. Chen, J. White, *et al.*, "A quadratic method for nonlinear model order reduction," 2000.

[36] B. Moore, "Principal component analysis in linear systems: Controllability, observability, and model reduction," *IEEE transactions on automatic control*, vol. 26, no. 1, pp. 17–32, 1981.

[37] P. Rabiei and M. Pedram, "Model order reduction of large circuits using balanced truncation," in *Proceedings of the ASP-DAC'99 Asia and South Pacific Design Automation Conference 1999 (Cat. No. 99EX198)*, pp. 237–240, IEEE, 1999.

[38] P. Wittmuess, C. Tarin, A. Keck, E. Arnold, and O. Sawodny, "Parametric model order reduction via balanced truncation with Taylor series representation," *IEEE Transactions on Automatic Control*, vol. 61, no. 11, pp. 3438–3451, 2016.

[39] S. Lall, J. E. Marsden, and S. Glavaški, "Empirical model reduction of controlled nonlinear systems," *IFAC Proceedings Volumes*, vol. 32, no. 2, pp. 2598–2603, 1999.

[40] F. Chinesta, A. Ammar, A. Leygue, and R. Keunings, "An overview of the proper generalized decomposition with applications in computational rheology," *Journal of Non-Newtonian Fluid Mechanics*, vol. 166, no. 11, pp. 578–592, 2011.

[41] D. Néron and P. Ladevèze, "Proper generalized decomposition for multiscale and multiphysics problems," *Archives of Computational Methods in Engineering*, vol. 17, no. 4, pp. 351–372, 2010.

[42] Z. M. Prince and J. C. Ragusa, "Application of proper generalized decomposition to multigroup neutron diffusion eigenvalue calculations," *Progress in Nuclear Energy*, vol. 121, p. 103232, 2020.

[43] Z. M. Prince and J. C. Ragusa, "Space-energy separated representations for multigroup neutron diffusion using proper generalized decompositions," *Annals of Nuclear Energy*, vol. 142, p. 107360, 2020.

[44] Z. M. Prince and J. C. Ragusa, "Parametric uncertainty quantification using proper generalized decomposition applied to neutron diffusion," *International Journal for Numerical Methods in Engineering*, vol. 119, no. 9, pp. 899–921, 2019.

[45] E. Bettis and R. Schroeder, "The aircraft reactor experiment. Design and construction," *Nucl. Sci. Eng*, vol. 2, pp. 804–826, 1957.

[46] D. LeBlanc, "Molten salt reactors: A new beginning for an old idea," *Nuclear Engineering and Design*, vol. 240, no. 6, pp. 1644 – 1656, 2010.

[47] I. Pioro, "Introduction: Generation IV International Forum," in *Handbook of Generation IV Nuclear Reactors*, pp. 37–54, Elsevier, 2016.

[48] D. Zhang, L. Liu, M. Liu, R. Xu, C. Gong, J. Zhang, C. Wang, S. Qiu, and G. Su, "Review of conceptual design and fundamental research of molten salt reactors in china," *International Journal of Energy Research*, vol. 42, no. 5, pp. 1834–1848, 2018.

[49] T. Ishiguro, W. Van Rooijen, Y. Shimazu, and H. Mochizuki, "Design of a passive residual heat removal system for the FUJI-233Um molten salt reactor system," *Annals of Nuclear Energy*, vol. 64, pp. 398–407, 2014.

[50] V. Ignatiev, O. Feynberg, I. Gnidoi, A. Merzlyakov, A. Surenkov, V. Uglov, A. Zagnitko, V. Subbotin, I. Sannikov, A. Toropov, *et al.*, "Molten salt actinide recycler and transforming system without and with Th-U support: Fuel cycle flexibility and key material properties," *Annals of Nuclear Energy*, vol. 64, pp. 408–420, 2014.

[51] "Molten Chloride Fast Reactor Technology." `https://www.terrapower.com/our-work/molten-chloride-fast-reactor-technology/`. Accessed: 09-19-2020.

[52] D. Heuer, E. Merle-Lucotte, M. Allibert, M. Brovchenko, V. Ghetta, and P. Rubiolo, "Towards the thorium fuel cycle with molten salt fast reactors," *Annals of Nuclear Energy*, vol. 64, pp. 421 – 429, 2014.

[53] "EVOL (Project n^{o}249696) Final Report ." `https://cordis.europa.eu/docs/results/249/249696/final1-final-report-f.pdf`. Accessed: 09-19-2020.

[54] J. L. Kloosterman *et al.*, "Safety assessment of the molten salt fast reactor (SAMOFAR)," in *Molten Salt Reactors and Thorium Energy*, pp. 565–570, Elsevier, 2017.

[55] M. Tano Retamales, *Développement des modèles multi-physiques multi-échelle de caloporteurs sels fondus à haute température et validation expérimentale*. PhD thesis, Grenoble Alpes, 2018.

[56] J. H. Ferziger, M. Perić, and R. L. Street, *Computational methods for fluid dynamics*, vol. 3. Springer, 2002.

[57] K. Pearson, "LIII. On lines and planes of closest fit to systems of points in space," *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 2, no. 11, pp. 559–572, 1901.

[58] M. Rathinam and L. R. Petzold, "A new look at proper orthogonal decomposition," *SIAM Journal on Numerical Analysis*, vol. 41, no. 5, pp. 1893–1925, 2003.

[59] R. Pinnau, "Model reduction via proper orthogonal decomposition," in *Model Order Reduction: Theory, Research Aspects and Applications*, pp. 95–109, Springer, 2008.

[60] H. Abdi, "Singular value decomposition (SVD) and generalized singular value decomposition," *Encyclopedia of measurement and statistics*, pp. 907–912, 2007.

[61] J. Hopcroft and R. Kannan, "Foundations of data science." Microsoft, `https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/book-No-Solutions-Aug-21-2014.pdf`, 2014. Accessed: 06-25-2021.

[62] S. Chaturantabut and D. C. Sorensen, "Nonlinear model reduction via discrete empirical interpolation," *SIAM Journal on Scientific Computing*, vol. 32, no. 5, pp. 2737–2764, 2010.

[63] B. Peherstorfer, D. Butnaru, K. Willcox, and H. Bungartz, "Localized Discrete Empirical Interpolation Method," *SIAM Journal on Scientific Computing*, vol. 36, no. 1, pp. A168–A192, 2014.

[64] L. Sirovich, "Turbulence and the dynamics of coherent structures. III. Dynamics and scaling," *Quarterly of Applied Mathematics*, vol. 45, pp. 583–590, 10 1987.

[65] D. Hilberg, W. Lazik, and H. E. Fiedler, "The Application of Classical POD and Snapshot POD in a Turbulent Shear Layer with Periodic Structures," *Applied Scientific Research*, vol. 53, pp. 283–290, 1994.

[66] J. Du, F. Fang, C. C. Pain, I. Navon, J. Zhu, and D. A. Ham, "POD reduced-order unstructured mesh modeling applied to 2D and 3D fluid flow," *Computers & Mathematics with Applications*, vol. 65, no. 3, pp. 362–379, 2013.

[67] G. Stabile and G. Rozza, "Finite volume POD-Galerkin stabilised reduced order methods for the parametrised incompressible Navier–Stokes equations," *Computers & Fluids*, vol. 173, pp. 273–284, 2018.

[68] S. Lorenzi, A. Cammi, L. Luzzi, and G. Rozza, "POD-Galerkin method for finite volume approximation of Navier–Stokes and RANS equations," *Computer Methods in Applied Mechanics and Engineering*, vol. 311, pp. 151–179, 2016.

[69] S. Hijazi, G. Stabile, A. Mola, and G. Rozza, "Data-driven POD-Galerkin reduced order model for turbulent flows," *arXiv preprint arXiv:1907.09909*, 2019.

[70] S. Ravindran, "Reduced-order adaptive controllers for fluid flows using POD," *Journal of scientific computing*, vol. 15, no. 4, pp. 457–478, 2000.

[71] J. Li, T. Zhang, S. Sun, and B. Yu, "Numerical investigation of the POD reduced-order model for fast predictions of two-phase flows in porous media," *International Journal of Numerical Methods for Heat & Fluid Flow*, 2019.

[72] R. Ştefănescu and I. M. Navon, "POD/DEIM nonlinear model order reduction of an ADI implicit shallow water equations model," *Journal of Computational Physics*, vol. 237, pp. 95–114, 2013.

[73] Y. Cao, J. Zhu, Z. Luo, and I. Navon, "Reduced-order modeling of the upper tropical Pacific ocean model using proper orthogonal decomposition," *Computers & Mathematics with Applications*, vol. 52, no. 8-9, pp. 1373–1386, 2006.

[74] F. Fang, C. Pain, I. Navon, M. Piggott, G. Gorman, P. Farrell, P. Allison, and A. Goddard, "A POD reduced-order 4D-Var adaptive mesh ocean modelling approach," *International Journal for Numerical Methods in Fluids*, vol. 60, no. 7, pp. 709–732, 2009.

[75] G. Kerschen, J.-C. Golinval, A. F. Vakakis, and L. A. Bergman, "The method of proper orthogonal decomposition for dynamical characterization and order reduction of mechanical systems: an overview," *Nonlinear dynamics*, vol. 41, no. 1-3, pp. 147–169, 2005.

[76] I. Georgiou, "Advanced proper orthogonal decomposition tools: using reduced order models to identify normal modes of vibration and slow invariant manifolds in the dynamics of planar nonlinear rods," *Nonlinear dynamics*, vol. 41, no. 1-3, pp. 69–110, 2005.

[77] A. Corigliano, M. Dossi, and S. Mariani, "Model Order Reduction and domain decomposition strategies for the solution of the dynamic elastic–plastic structural problem," *Computer Methods in Applied Mechanics and Engineering*, vol. 290, pp. 127–155, 2015.

[78] A. Sartori, D. Baroli, A. Cammi, D. Chiesa, L. Luzzi, R. Ponciroli, E. Previtali, M. E. Ricotti, G. Rozza, and M. Sisti, "Comparison of a Modal Method and a Proper Orthogonal Decomposition approach for multi-group time-dependent reactor spatial kinetics," *Annals of Nuclear Energy*, vol. 71, pp. 217–229, 2014.

[79] A. Sartori, D. Baroli, A. Cammi, L. Luzzi, and G. Rozza, "A reduced order model for multigroup time-dependent parametrized reactor spatial kinetics," in *2014 22nd International Conference on Nuclear Engineering*, pp. V005T17A048–V005T17A048, American Society of Mechanical Engineers, 2014.

[80] D. Prill and A. Class, "Semi-automated proper orthogonal decomposition reduced order model non-linear analysis for future BWR stability," *Annals of Nuclear Energy*, vol. 67, pp. 70–90, 2014.

[81] S. Lorenzi, "An Adjoint Proper Orthogonal Decomposition method for a neutronics reduced order model," *Annals of Nuclear Energy*, vol. 114, pp. 245–258, 2018.

[82] A. G. Buchan, A. Calloo, M. G. Goffin, S. Dargaville, F. Fang, C. C. Pain, and I. M. Navon, "A POD reduced order model for resolving angular direction in neutron/photon transport problems," *Journal of Computational Physics*, vol. 296, pp. 138–157, 2015.

[83] P. Behne, J. Ragusa, and J. Morel, "Model Order Reduction for Sn Radiation Transport," *Proceedings of M&C2019, Portland, OR, USA*, pp. 2481–2491, 2016.

[84] Y. Sun, J. Yang, Y. Wang, Z. Li, and Y. Ma, "A POD reduced-order model for resolving the neutron transport problems of nuclear reactor," *Annals of Nuclear Energy*, vol. 149, p. 107799, 2020.

[85] A. Buchan, C. Pain, F. Fang, and I. Navon, "A POD reduced-order model for eigenvalue problems with application to reactor physics," *International Journal for Numerical Methods in Engineering*, vol. 95, no. 12, pp. 1011–1032, 2013.

[86] C. Wang and H. S. Abdel-Khalik, "Construction of accuracy-preserving surrogate for the eigenvalue radiation diffusion and/or transport problem," tech. rep., American Nuclear Society, Inc., 555 N. Kensington Avenue, La Grange Park, Illinois 60526 (United States), 2012.

[87] C. Wang, H. S. Abdel-Khalik, and U. Mertyurek, "CRANE: A New Scale Super-Sequence for Neutron Transport Calculations," *Proceeding of MC 2015, Nashville, TN, April 19*, vol. 23, 2015.

[88] P. German and J. C. Ragusa, "Reduced-order modeling of parameterized multi-group diffusion k-eigenvalue problems," *Annals of Nuclear Energy*, vol. 134, pp. 144–157, 2019.

[89] C. Castagna, M. Aufiero, S. Lorenzi, G. Lomonaco, and A. Cammi, "Development of a Reduced Order Model for Fuel Burnup Analysis," *Energies*, vol. 13, no. 4, p. 890, 2020.

[90] A. Sartori, A. Cammi, L. Luzzi, and G. Rozza, "A multi-physics reduced order model for the analysis of Lead Fast Reactor single channel," *Annals of Nuclear Energy*, vol. 87, pp. 198–208, 2016.

[91] P. German, J. C. Ragusa, and C. Fiorina, "Application of multiphysics model order reduction to doppler/neutronic feedback," *EPJ Nuclear Sciences & Technologies*, vol. 5, p. 17, 2019.

[92] D. M. Panico, "Development of C++ tool for Reduced Order Modelling of the Molten Salt Fast Reactor multiphysics: Sensitivity Analysis application (MSc Thesis)," 2019.

[93] C. Fiorina, I. Clifford, M. Aufiero, and K. Mikityuk, "GeN-Foam: a novel OpenFOAM® based multi-physics solver for 2D/3D transient analysis of nuclear reactors," *Nuclear Engineering and Design*, vol. 294, pp. 24–37, 2015.

[94] I. D. Clifford, "A hybrid coarse and fine mesh solution method for prismatic high temperature gas-cooled reactor thermal-fluid analysis, PhD thesis," 2013.

[95] R. Saurel and R. Abgrall, "A multiphase Godunov method for compressible multifluid and multiphase flows," *Journal of Computational Physics*, vol. 150, no. 2, pp. 425–467, 1999.

[96] K. Vafai, *Handbook of porous media*. Crc Press, 2015.

[97] T. Kajishima and K. Taira, *Computational fluid dynamics*. Springer, 2017.

[98] F. G. Schmitt, "About Boussinesq's turbulent viscosity hypothesis: historical remarks and a direct evaluation of its validity," *Comptes Rendus Mécanique*, vol. 335, no. 9-10, pp. 617–627, 2007.

[99] D. D. Gray and A. Giorgini, "The validity of the boussinesq approximation for liquids and gases," *International Journal of Heat and Mass Transfer*, vol. 19, no. 5, pp. 545–551, 1976.

[100] H. Rusche, *Computational fluid dynamics of dispersed two-phase flows at high phase fractions*. PhD thesis, University of London, 2002.

[101] F. P. Incropera and D. P. De Witt, "Fundamentals of heat and mass transfer 3rd edition," *Jon Wiley and Sons, NY*, 1990.

[102] A. Chenu, K. Mikityuk, and R. Chawla, "Pressure drop modeling and comparisons with experiments for single-and two-phase sodium flow," *Nuclear engineering and design*, vol. 241, no. 9, pp. 3898–3909, 2011.

[103] K. T. Trinh, "On the Blasius correlation for friction factors," *arXiv preprint arXiv:1007.2466*, 2010.

[104] R. Temam, *Navier-Stokes equations: theory and numerical analysis*, vol. 343. American Mathematical Soc., 2001.

[105] B. E. Launder and D. B. Spalding, "The numerical computation of turbulent flows," in *Numerical prediction of flow, heat transfer, turbulence and combustion*, pp. 96–116, Elsevier, 1983.

[106] I. Lux and L. Koblinger, *Monte Carlo particle transport methods*. CRC press, 2018.

[107] J. J. Duderstadt and L. J. Hamilton, "Nuclear reactor analysis. 1976," *John Wiley& Sons, NY*.

[108] R. Sanchez and N. J. McCormick, "A review of neutron transport approximations," *Nuclear Science and Engineering*, vol. 80, no. 4, pp. 481–535, 1982.

[109] J. Banasiak and J. Mika, "Diffusion limit for the linear boltzmann equation of the neutron transport theory," *Mathematical methods in the applied sciences*, vol. 17, no. 13, pp. 1071–1087, 1994.

[110] K. M. Case, *Introduction to the theory of neutron diffusion*, vol. 1. Los Alamos Scientific Laboratory, 1954.

[111] C. Nicolino, G. Lapenta, S. Dulla, and P. Ravetto, "Coupled dynamics in the physics of molten salt reactors," *Annals of Nuclear Energy*, vol. 35, no. 2, pp. 314–322, 2008.

[112] M. Aufiero, M. Brovchenko, A. Cammi, I. Clifford, O. Geoffroy, D. Heuer, A. Laureau, M. Losa, L. Luzzi, E. Merle-Lucotte, *et al.*, "Calculating the effective delayed neutron fraction in the molten salt fast reactor: analytical, deterministic and Monte Carlo approaches," *Annals of Nuclear Energy*, vol. 65, pp. 78–90, 2014.

[113] F. Mattioda, P. Ravetto, and G. Ritter, "Effective delayed neutron fraction for fluid-fuel systems," *Annals of Nuclear Energy*, vol. 27, no. 16, pp. 1523–1532, 2000.

[114] H. Jasak, A. Jemcov, Z. Tukovic, *et al.*, "OpenFOAM: A C++ library for complex physics simulations," in *International workshop on coupled methods in numerical dynamics*, vol. 1000, pp. 1–20, IUC Dubrovnik Croatia, 2007.

[115] H. K. Versteeg and W. Malalasekera, *An introduction to computational fluid dynamics: the finite volume method*. Pearson education, 2007.

[116] H. Jasak, "Error analysis and estimation for the finite volume method with applications to fluid flows. ICL, PhD thesis," 1996.

[117] J. Leppänen, M. Pusa, T. Viitanen, V. Valtavirta, and T. Kaltiaisenaho, "The Serpent Monte Carlo code: Status, development and applications in 2013," *SNA+ MC 2013-Joint International Conference on Supercomputing in Nuclear Applications+ Monte Carlo*, pp. 6–21, 2014.

[118] F. Moukalled, L. Mangani, M. Darwish, *et al.*, *The finite volume method in computational fluid dynamics*, vol. 113. Springer, 2016.

[119] M. Bergmann, C.-H. Bruneau, and A. Iollo, "Enablers for robust POD models," *Journal of Computational Physics*, vol. 228, no. 2, pp. 516–538, 2009.

[120] S. Lorenzi, A. Cammi, L. Luzzi, and G. Rozza, "A reduced order model for investigating the dynamics of the Gen-IV LFR coolant pool," *Applied Mathematical Modelling*, vol. 46, pp. 263–284, 2017.

[121] S. Hijazi, S. Ali, G. Stabile, F. Ballarin, and G. Rozza, "The effort of increasing Reynolds number in projection-based reduced order methods: from laminar to turbulent flows," *arXiv preprint arXiv:1807.11370*, 2018.

[122] Y. Bazilevs and T. J. Hughes, "Weak imposition of Dirichlet boundary conditions in fluid mechanics," *Computers & Fluids*, vol. 36, no. 1, pp. 12–26, 2007.

[123] A. Barbagallo, D. Sipp, and P. J. Schmid, "Closed-loop control of an open cavity flow using reduced-order models," *Journal of Fluid Mechanics*, vol. 641, pp. 1–50, 2009.

[124] A.-L. Gerner and K. Veroy, "Certified reduced basis methods for parametrized saddle point problems," *SIAM Journal on Scientific Computing*, vol. 34, no. 5, pp. A2812–A2836, 2012.

[125] F. Ballarin, A. Manzoni, A. Quarteroni, and G. Rozza, "Supremizer stabilization of POD–Galerkin approximation of parametrized steady incompressible Navier–Stokes equations," *International Journal for Numerical Methods in Engineering*, vol. 102, no. 5, pp. 1136–1161, 2015.

[126] S. Georgaka, G. Stabile, K. Star, G. Rozza, and M. J. Bluck, "A hybrid reduced order method for modelling turbulent heat transfer problems," *Computers & Fluids*, vol. 208, p. 104615, 2020.

[127] G. B. Wright, *Radial basis function interpolation: numerical and analytical developments*. University of Colorado at Boulder, 2003.

[128] B. Sudret, "Global sensitivity analysis using polynomial chaos expansions," *Reliability engineering & system safety*, vol. 93, no. 7, pp. 964–979, 2008.

[129] S. R. Arwade, M. Moradi, and A. Louhghalam, "Variance decomposition and global sensitivity for structural systems," *Engineering Structures*, vol. 32, no. 1, pp. 1–10, 2010.

[130] A. Saltelli, M. Ratto, T. Andres, F. Campolongo, J. Cariboni, D. Gatelli, M. Saisana, and S. Tarantola, *Global sensitivity analysis: the primer*. John Wiley & Sons, 2008.

[131] M. Baudin, A. Dutfoy, B. Iooss, and A.-L. Popelin, "OpenTURNS: An industrial software for uncertainty quantification in simulation," *arXiv preprint arXiv:1501.05242*, 2015.

[132] A. Ribes and C. Caremoli, "Salome platform component model for numerical simulation," in *31st annual international computer software and applications conference (COMPSAC 2007)*, vol. 2, pp. 553–564, IEEE, 2007.

[133] M. Aufiero, A. Cammi, O. Geoffroy, M. Losa, L. Luzzi, M. E. Ricotti, and H. Rouch, "Development of an OpenFOAM model for the Molten Salt Fast Reactor transient analysis," *Chemical Engineering Science*, vol. 111, pp. 390–401, 2014.

[134] H. Rouch, O. Geoffroy, P. Rubiolo, A. Laureau, M. Brovchenko, D. Heuer, and E. Merle-Lucotte, "Preliminary thermal–hydraulic core design of the Molten Salt Fast Reactor (MSFR)," *Annals of Nuclear Energy*, vol. 64, pp. 449–456, 2014.

[135] B. Van Leer, "Towards the ultimate conservative difference scheme. V. A second-order sequel to Godunov's method," *Journal of computational Physics*, vol. 32, no. 1, pp. 101–136, 1979.

[136] M. Waskom, O. Botvinnik, D. O'Kane, P. Hobson, S. Lukauskas, D. C. Gemperline, T. Augspurger, Y. Halchenko, J. B. Cole, J. Warmenhoven, J. de Ruiter, C. Pye, S. Hoyer, J. Vanderplas, S. Villalba, G. Kunter, E. Quintero, P. Bachant, M. Martin, K. Meyer, A. Miles, Y. Ram, T. Yarkoni, M. L. Williams, C. Evans, C. Fitzgerald, Brian, C. Fonnesbeck, A. Lee, and A. Qalieh, "`https://github.com/mwaskom/seaborn`: v0.8.1 (september 2017)," Sept. 2017.

[137] C. Geuzaine and J.-F. Remacle, "Gmsh: A 3-D finite element mesh generator with built-in pre-and post-processing facilities," *International journal for numerical methods in engineering*, vol. 79, no. 11, pp. 1309–1331, 2009.

APPENDIX A

LIST OF PUBLICATIONS

A.1 Accepted Journal Papers

Tano, M., **German, P.**, Ragusa, J., *Evaluation of pressure reconstruction techniques for Model Order Reduction in incompressible convective heat transfer*, Thermal Science and Engineering Progress, Vol. 23, 2021, 100841.

German, P., Tano, M., Ragusa, J. C., Fiorina, C., *Comparison of Reduced-Basis techniques for the model order reduction of parametric incompressible fluid flows*, Progress in Nuclear Energy, 130 (2020), 103551.

German, P., Ragusa, J. C., and Fiorina, C., *Application of multiphysics model order reduction to doppler/neutronic feedback*, EPJ Nuclear Sciences & Technologies 5, ARTICLE (2019): 17.

German, P., and Ragusa, J. C.. *Reduced-order modeling of parameterized multi-group diffusion k-eigenvalue problems*, Annals of Nuclear Energy 134 (2019): 144-157.

A.2 Journal Papers in Preparation

German, P., Ragusa, J. C., Tano, M., and Fiorina, C., *Reduced-order modeling of convective flows in porous media*, Fluids

German, P., Ragusa, J. C., Tano, M., and Fiorina, C., *Multiphysics reduced-order modeling of Molten Salt Reactors*, Progress in Nuclear Energy

A.3 Conference Proceedings

German, P., Tano, M., Ragusa, J. C., Fiorina, C., *Reduced-Order Modeling of Transient Flows in Closed-loop Systems*, PHYSOR 2020, Cambridge UK, April 2020

German, P., Tano, M., Ragusa, J. C., Fiorina, C., *Reduced-Order Modeling of Parametrized Multi-Physics Computations for the Molten Salt Fast Reactor*, M&C 2019 Portland, Oregon USA, August 25-29

German, P., Tano, M., Ragusa, J. C., Fiorina, C., *Reduced-Order Modeling of Fluid Flows in Closed-Loop Systems*, ANS Winter Meeting 2019 Washington DC, USA, November 17-21