Delft University of Technology

Adaptive data-driven reduced-order modelling techniques for nuclear reactor analysis

Alsayyari, F.S.

DOI

Publication date
2020

Document Version
Final published version

Citation (APA)

Important note
To cite this publication, please use the final published version (if applicable).
Please check the document version above.

ADAPTIVE DATA-DRIVEN REDUCED-ORDER MODELLING TECHNIQUES FOR NUCLEAR REACTOR ANALYSIS

Dissertation

for the purpose of obtaining the degree of doctor
at Delft University of Technology
by the authority of the Rector Magnificus Prof.dr.ir. T.H.J.J. van der Hagen
chair of the Board for Doctorates
to be defended publicly on Tuesday 6 October 2020 at 10:00 o'clock

by

Fahad ALSAYYARI

Magister en Ingeniería, Instituto Balseiro, Argentina
born in Riyadh, Saudi Arabia.

This dissertation has been approved by

Promotor: Prof.dr.ir. J.L. Kloosterman
Promotor: Dr.ir. D. Lathouwers
Copromotor: Dr. Z. Perkó

Composition of the doctoral committee:

Rector Magnificus,	chairperson
Prof.dr.ir. J.L. Kloosterman,	Delft University of Technology, promotor
Dr.ir. D. Lathouwers,	Delft University of Technology, promotor
Dr. Z. Perkó,	Delft University of Technology, copromotor

Independent members:

Prof.dr. A. Cammi,	Politechnical U. Milan, Italy
Prof.dr. J.C. Ragusa,	Texas A&M U., USA
Prof.dr. W.H.A. Schilders,	TU Eindhoven
Prof.dr.ir. A.W. Heemink,	Delft University of Technology
Prof.dr. P. Dorenbos,	Delft University of Technology, reserve member

To my best teachers,
my parents, Sulaiman and Norah.

CONTENTS

SUMMARY

L ARGE-SCALE complex systems require high-fidelity models to capture the dynamics of the system accurately. For example, models of nuclear reactors capture multi-physics interactions (e.g., radiation transport, thermodynamics, heat transfer, and fluid mechanics) occurring at various scales of time (prompt neutrons to burn-up calculations) and space (cell and core calculations). The complexity of these models, however, renders their use intractable for applications relying on repeated evaluations, such as control, optimization, uncertainty quantification, and sensitivity studies.

Reduced-order modelling (ROM) is an effective technique to reduce the complexity of such models in order to achieve a manageable computational cost. ROM methods rely on mathematical tools to replace the high-fidelity, expensive model with an efficient, low-dimensional model with a controlled level of accuracy. While different approaches for ROM exist in the literature, proper orthogonal decomposition (POD) is the most suited method for nonlinear systems (such as nuclear reactors). POD can be implemented in an intrusive setting, where access to the governing equations of the high-fidelity model is feasible, or in a nonintrusive (data-driven) setting using only data generated from the high-fidelity model. In practical nuclear reactor applications, most models are implemented with either closed-source or complex coupled codes that have been developed over many years to be certified by regulatory bodies. Thus, attempting to apply intrusive methods to such codes is impractical.

For this reason, this work develops a data-driven methodology based on POD to construct reduced-order models for nonlinear, large-scale nuclear reactor systems. The accuracy and efficiency of the data-driven POD method are known to be highly dependent on the sampling scheme, especially for high-dimensional problems. Reactor models are characterized by a large number of parameters, which often leads to the curse of dimensionality (i.e., the exponential increase in the computational resources with the increase in the parameter space dimensions). Therefore, a key challenge for any data-driven ROM method is to develop an effective sampling strategy for exploring large parameter spaces. In this work, we address this challenge with a novel approach using locally adaptive sparse grid techniques. Our approach iteratively adapts the sampling points to the problem without knowledge of the underlying governing equations. Additionally, we developed the adaptivity in both time and parameter spaces for steady-state and time-dependent systems, which allows for a wide range of potential applications.

We test our iterative approach on several numerical test problems of various degrees of nonlinearities, complexity, scale, and dimensionality. Eventually, we apply our approach to a full three-dimensional model of the molten salt fast reactor (MSFR), which represents the largest test in scale and dimension with 30 input parameters and 220,972 degrees of freedom. Our approach provides means to set the required tolerance on the error in the reduced-order model. The results of the test problems demonstrated the success of the method in terms of providing a reduced-order model with an error within the required

tolerance. Furthermore, the method includes a greediness parameter that controls the efficiency of the sampling scheme, which allowed for even higher dimensionality applications by identifying and disregarding irrelevant dimensions after the first few iterations. Finally, the structure of the developed adaptive sparse grid technique provides a tool for characterizing the nonlinearities of the model with respect to each parameter without accessing the governing equations.

The focus of this research is on nuclear reactor analysis problems. However, the challenge of developing a ROM method for a complex nonlinear system in a nonintrusive manner is present in many science and engineering applications. Because of the non-intrusiveness of our approach, no adaptations are required for applications in general large-scale engineering problems.

SAMENVATTING

G ROOTSCHALIGE complexe systemen vereisen modellen met een hoge betrouwbaarheid om de dynamiek van het systeem nauwkeurig te kunnen vatten. Bijvoorbeeld, modellen van nucleaire reactoren beschrijven multi-fysische interacties (bijvoorbeeld stralingstransport, thermodynamica, warmteoverdracht en vloeistofmechanica) die een rol spelen op verscheidene tijdschalen (van prompt neutronen tot opbrandberekeningen) en ruimteschalen (cel- en kernberekeningen). De complexiteit van deze modellen maakt dit soort modellen onbruikbaar voor toepassingen die zich baseren op herhaalde evaluaties, zoals controle, optimalisatie, het kwantificeren van onzekerheden en gevoeligheidsanalyses.

Zogenaamde *Reduced-order Modelling* (ROM) is een effectieve techniek om de complexiteit van dit soort modellen te reduceren waardoor de rekentijden beheersbaar blijven. ROM methoden baseren zich op wiskundige technieken om een model met een hoge betrouwbaarheid en hoge rekenkosten te vervangen door een efficiënt, laag-dimensionaal model met een gecontroleerde nauwkeurigheid. Alhoewel verschillende benaderingen voor ROM in de literatuur te vinden zijn, is *proper orthogonal decomposition* (POD) de beste methode voor niet-lineaire systemen (zoals nucleaire reactoren). POD kan worden geïmplementeerd in een intrusieve context, waar toegang tot de beschrijvende vergelijkingen van het model mogelijk is, of binnen een niet-intrusieve (data-gedreven) context waarin slechts gebruik gemaakt wordt van data die gegenereerd is in het nauwkeurige model. In praktische nucleaire toepassingen worden de meeste modellen geïmplementeerd met gesloten broncode of d.m.v. complexe gekoppelde codes, waarbij het jaren duurde om ze te ontwikkelen en goed te laten keuren door officiële instanties. Het toepassen van intrusieve methoden op dergelijke codes is dan ook niet praktisch.

Daarom wordt in dit onderzoek een data-gedreven methodologie ontwikkeld die gebaseerd is op POD om een gereduceerd model te construeren voor niet-lineaire, grootschalige nucleaire reactorsystemen. De nauwkeurigheid en efficiëntie van de data-gedreven POD methode staan erom bekend dat ze sterk afhankelijk zijn van het bemonsteringsschema, vooral bij hoog-dimensionale problemen. Reactormodellen worden gekenmerkt door gebruik van vele parameters, wat vaak leidt tot de vloek van de dimensionaliteit (de exponentiële toename in de benodigde rekencapaciteit door de toename van de dimensies van de parameterruimte). Vanwege deze reden is de grote uitdaging voor iedere data-gedreven ROM methode om een effectieve bemonsteringsstrategie te ontwikkelen om ruimten met veel parameters te verkennen. In dit onderzoek gaan we de uitdaging aan door een nieuwe benadering te introduceren die gebruik maakt van lokaal adaptieve *sparse grid* technieken. Onze aanpak kiest de gekozen monsters op iteratieve wijze zonder kennis van de onderliggende beschrijvende vergelijkingen van het probleem. Daarnaast hebben we adaptiviteit in zowel tijd en parameterruimtes voor stationaire en tijdsafhankelijke systemen ontwikkeld, wat ervoor zorgt dat het model voor vele potentiële toepassingen kan worden ingezet.

We testen onze iteratieve aanpak op verschillende numerieke testproblemen met verschillende gradaties van niet-lineariteit, complexiteit, schaal en dimensionaliteit. Uiteindelijk passen we onze techniek toe op een volledig driedimensionaal model van de snel-spectrum variant van de gesmolten zout reactor, de zogenaamde *Molten Salt Fast Reactor* (MSFR). Deze test vertegenwoordigt de grootste test, zowel in schaal als dimensie met 30 inputparameters en 220.972 vrijheidsgraden. Onze aanpak biedt de mogelijkheid om de vereiste tolerantie op de fout in het gereduceerd model in te stellen. De testresultaten laten zien dat het een succesvolle methode is om een gereduceerd model met een fout binnen de vereiste tolerantie te ontwikkelen. Bovendien bevat de methode ook een *greediness* parameter die de efficiëntie van het samplingschema controleert, waardoor het model ook toepasbaar is op toepassingen met een nog hogere dimensionaliteit door het identificeren en negeren van irrelevante dimensies na de eerste iteraties. Tenslotte verschaft de structuur van de ontwikkelde adaptieve *sparse grid* techniek een manier om de niet-lineariteiten van het model te karakteriseren met betrekking tot elke parameter zonder gebruikmaking van de beschrijvende vergelijkingen.

De nadruk van dit onderzoek ligt op problemen uit de kernreactoranalyse. Echter, de uitdaging met betrekking tot het ontwikkelen van een ROM methode voor complexe niet-lineaire systemen bestaat in vele andere wetenschappelijke en engineering toepassingen. Vanwege het niet-intrusieve karakter van onze methodiek, kan deze zonder aanpassingen worden toegepast op generieke grootschalige engineeringproblemen

1

INTRODUCTION

1.1. MOTIVATION

I N many science and engineering applications, mathematical models are indispensable to predict the behaviour of a system. However, modelling large-scale, complex systems is a challenging task. In particular, nuclear reactors are examples of such complex systems where the modelling process involves capturing the interactions between radiation transport, heat transfer, fluid mechanics, and structural analysis. Due to the limited computational resources in the past, numerical simulation of nuclear reactors used to be carried out with several decoupled models tackling each field and scale separately.

However, the trend in the nuclear industry has shifted towards interdisciplinary high-fidelity models, which often seek to provide comprehensive solutions to coupled problems involving multi-physics phenomena. This trend is driven by the increase in the computational power of today's computer hardware. In addition, regulations have moved towards requirements based on the best-estimate-plus-uncertainty approach instead of the traditional conservative approach. This calls for higher demand on high-fidelity models. However, because of the massive computational resources required by these models, they are not suitable for the so-called *many-query* applications– that is, applications where many repeated evaluations of the model are needed, such as design optimization, control, and uncertainty quantifications.

Therefore, in order to achieve savings in computational cost for such applications, models are often simplified. The simplification can be done based on the physics of the problem. For example, the spatial dimensionality may be reduced (e.g., coolant flow within a reactor core may be reduced to one-dimensional flow), or a particular phenomenon may be neglected (e.g., reactor's structure heating due to radiation). Furthermore, based on the prior knowledge of the problem, discretization may be adapted to have finer mesh in areas of interest and coarse meshes in less important areas. Knowledge about the symmetry can also be exploited to model only part of the system. All these techniques require physical insight into the problem to achieve the desired reduction in complexity. This class of techniques can be called *operational* model order reduction [1]. The chal-

1

1

lenge in applying this kind of reduction lies in having a sufficiently deep understanding of the physics of the problem.

An alternative approach to reduce the complexity of the problem is the so-called reduced-order modeling (ROM), which, depending on the context and the field of study, can be defined in several ways. However, concisely, ROM is a collection of methods derived using optimizing mathematical tools that aim to replace a high-fidelity, complex model with an efficient, low-dimensional model with a controlled level of accuracy. ROM methods have applications in fields of control, design, optimization, and uncertainty quantification across many engineering disciplines [2–11].

ROM is a strong candidate to be applied in the many-query context for nuclear reactor applications. This is especially true for the Generation IV reactors, such as the Molten Salt Reactor (MSR), where expertise in understanding their dynamics is limited. ROM methods can also be appreciated in the design phase of these new reactors to optimize the selection of parameters and the design of controllers. Moreover, having real-time simulation capabilities is essential for training and educational purposes of the new reactors. A difficulty commonly encountered in solving reactor models is the treatment of a large number of input parameters (cross sections, thermal-hydraulics, and material parameters). This fact causes reactor models to be prone to the so-called curse of dimensionality – that is, the exponential increase in computational time with the increase in input parameters.

Hence, this research is motivated by the need for ROM methods in nuclear reactor applications that can alleviate the computational burden of high-dimensional studies.

1.2. ROM METHODS

Different ROM methods can achieve the required reduction. They all share an *offline phase* where the models are developed using costly computations and an *online phase* where the models are evaluated using inexpensive algorithms [12]. It is important to highlight that the concept of ROM is not recent. A simple interpolating function or a truncated Taylor series expansion can be considered as two of the earliest forms of ROM. However, as a rigorous set of tools, this technique first appeared in the area of systems and control theory. Later on, these techniques were further developed by numerical mathematicians and computational scientists [1].

Several survey papers on the different ROM approaches can be found in the literature, such as [12–16]. All ROM methods can be broadly classified into two main categories. On the one hand are methods that drive the reduced model by utilizing the original governing equations of the high-fidelity model. These are **intrusive** methods that can only be applied if access to the system's governing equations is available. On the other hand, **nonintrusive** methods do not require access to the governing equations. They build a surrogate model that replicates the output response based on a set of collected input-output statistical data. In this section, the main methods within each class are presented.

1.2.1. INTRUSIVE APPROACHES

Intrusive ROM methods are also called *projection-based* methods because most methods in this class follow the idea of projecting the governing equations of the original high-fidelity model onto a selected reduced subspace [13]. The projection is achieved by means of a Petrov-Galerkin projection, which can be illustrated as follows: Consider a general time-dependent Partial Differential Equation (PDE) in the form,

$$\frac{dy(x,t)}{dt} = \mathscr{L}(y(x,t)) + \mathscr{F}(y(x,t)), \tag{1.1}$$

where $\mathscr{L}(\cdot)$ is a linear operator and $\mathscr{F}(\cdot)$ is a nonlinear function, and $y(x,t)$ is the unknown function to be computed from a high-fidelity model, which depends on state space x, and time t. At this point, the equation is general such that $y(x,t)$ could be any physical quantity (e.g., neutron flux in a reactor or pressure in a thermal hydraulic loop or voltage in an electrical circuit model).

We first consider linear systems, as treatment of the nonlinear term will be explicitly discussed in Section 1.2.2. Hence, considering the linear operator only (i.e., neglecting the nonlinear term $\mathscr{F}(\cdot)$), Equation 1.1 can be rewritten in a discrete form using a discretization scheme (e.g., finite difference, finite volume or finite element) for the linear operator $\mathscr{L}(\cdot)$ with appropriate boundary and initial conditions as

$$\frac{d\boldsymbol{y}(t;\boldsymbol{\mu})}{dt} = A(\boldsymbol{\mu})\,\boldsymbol{y}(t;\boldsymbol{\mu}) + B(\boldsymbol{\mu})u(t), \tag{1.2}$$

where $\boldsymbol{y}(t;\boldsymbol{\mu}) \in R^n$ is the state vector of the system and n is the dimension of the system, $A(\boldsymbol{\mu}) \in R^{n \times n}$ is a discretization matrix of the linear operator $\mathscr{L}(\cdot)$, and $u(t)$ is the input signal. Without loss of generality, the system considered in this discussion will be assumed to be of a single input system. Thus, the input matrix $B(\boldsymbol{\mu}) \in R^n$. Moreover, we assume that the system is also dependent on some input parameter of interest $\boldsymbol{\mu} \in R^d$, where d is the dimension of the input domain such that $\boldsymbol{y}(t;\boldsymbol{\mu})$. The parameter $\boldsymbol{\mu}$ can represent geometry, material, boundary and/or initial conditions of the problem. We seek to evaluate Equation 1.2 at different values of $\boldsymbol{\mu}$. For the sake of convenience, the dependence on the input parameter $\boldsymbol{\mu}$ will not be shown explicitly but rather implied ($\boldsymbol{y}(t;\boldsymbol{\mu}) \equiv \boldsymbol{y}(t), A(\boldsymbol{\mu}) \equiv A, B(\boldsymbol{\mu}) \equiv B$).

Note that Equation 1.2 is a system of Ordinary Differential Equations (ODE) that, generally, can be solved directly. However, if the dimension of the system n is large, the computational burden for the simulation would be expensive. In order to reduce the dimensionality of the problem, we seek a Galerkin approximation of the form

$$\boldsymbol{y}(t) \approx \boldsymbol{y}_r(t) = V\boldsymbol{z}(t), \tag{1.3}$$

where $V \in R^{n \times r}$ is a transformation (or basis) matrix whose columns span a reduced subspace such that $r \ll n$ and $\boldsymbol{z}(t) \in R^r$. In addition, we define a projection matrix $W \in R^{n \times r}$ such that $W^T V = I$, where I is the identity matrix $(I \in R^{r \times r})$. Replacing Equation 1.3 in Equation 1.2 and multiplying by W^T yields

$$W^T V \frac{d\boldsymbol{z}(t)}{dt} = W^T A V \boldsymbol{z}(t) + W^T B u(t), \tag{1.4}$$

1

which can be written as

$$\frac{d\boldsymbol{z}(t)}{dt} = A_r \boldsymbol{z}(t) + B_r u(t),$$

$$(1.5)$$

where $A_r = W^T A V$, and $B_r = W^T B$.

It is evident that Equation 1.5 is a reduced form of Equation 1.2. If the basis spanning the columns of W^T and V are chosen appropriately, the dynamics of the high-fidelity model can be captured effectively with a reduced computational cost.

Projection-based ROM methods differ in the approach to compute the transformation and projection matrices, W^T and V. Constructing these matrices is part of the offline phase, which can be computationally demanding. Nevertheless, once the matrices are known, solving Equation 1.5 becomes a low-cost online computation, which can be repeated inexpensively at different input values. The remainder of the subsection covers the three main projection-based methods: Balanced Truncation, Krylov subspace methods and Proper Orthogonal Decomposition (POD).

BALANCED TRUNCATION

Balanced truncation is one of the most elaborate methods with a strong, rigorous mathematical derivation. The method was first suggested by Moore [17], which was initially developed for linear-time-invariant (LTI) systems in control theory applications. The idea is that a balanced reduction can be applied to a system such that the states, which are both difficult to observe and control, are truncated [18]. These states are measured from the so-called observability gram matrix ($Q \in R^{n \times n}$) and controllability gram matrix ($P \in R^{n \times n}$). The gramians are obtained by solving a system of Lyapunov equations. Then, the gramians are used to compute the transformation and projection matrices, W^T and V (see [12, 18] for a detailed description).

It can be shown that the error in the reduced model has an upper bound [18]. The advantages of balanced truncation are that the error is guaranteed for all input values and the reduced model preserves the stability in the original system. To deal with parametrized dynamical systems, one can build a separate reduced model locally for several sampled parameter. Then, a solution for a non-sampled parameter can be obtained either by directly interpolating between local reduced model outputs, or projecting the equations on an interpolated local bases space. Alternatively, one can concatenate the local bases spaces for a single global basis space, which is then used for one global reduced model. However, the error bound is not guaranteed for models of varying parameters ($\boldsymbol{\mu}$)[12]. Moreover, solving the Lyapunov equations is intractable for high-dimensional, parameter-varying systems [19]. Some efforts to overcome this difficulty include Krylov iterative methods [20] and low rank approximation algorithms [21–23].

KYRLOV SUBSPACE METHODS

Krylov iterative methods are among the most powerful tools in linear algebra to deal with large-scale, sparse problems[1]. In fact, they are used in the balanced truncation method to efficiently solve the Lyapunov equations. However, not to be confused with this technique, by Krylov subspace methods, we refer to methods that are also called *moment matching* methods or *Padé approximation* methods. The concept is to construct a reduced model

---

[1]IEEE computer society included Krylov subspace methods as one of the top 10 algorithms of the century [24].

with a transfer function that matches the original model up to a certain degree around a selected point.

The method can be illustrated by first transforming the original model in Equation 1.2 to the frequency domain using the Laplace transform,

$$sY(s) = AY(s) + BU(s),$$ (1.6)

where the zero initial condition is assumed. Then, the transfer function is defined as

$$G(s) = \frac{Y(s)}{U(s)} = (sI - A)^{-1}B,$$ (1.7)

with the assumption that $(sI - A)$ is non-singular.

The transfer function can be rewritten to include a selected frequency s_0,

$$G(s) = (sI - A)^{-1}B = ((s - s_0)I - (A - s_0 I))^{-1}B.$$ (1.8)

Then, expanding the transfer function with Taylor series around the selected s_0,

$$
\begin{aligned}
G(s) &= ((s - s_0)I - (A - s_0 I))^{-1}B \\
&= -\underbrace{(A - s_0 I)^{-1}B}_{m_0} - \underbrace{(A - s_0 I)^{-2}B}_{m_1}(s - s_0) - \ldots - \underbrace{(A - s_0 I)^{-(j+1)}B}_{m_j}(s - s_0)^j - \ldots
\end{aligned}
$$ (1.9)

The vectors $\boldsymbol{m}_j = (A - s_0 I)^{-(j+1)}B$ are called *moments* of the system [25]. One can note that these moments actually span a Krylov subspace,

$$K_q(M, \boldsymbol{r}) = span\{\boldsymbol{r}, M\boldsymbol{r}, M^2\boldsymbol{r}, \ldots M^{q-1}\boldsymbol{r}\},$$ (1.10)

where the matrix $M = (A - s_0 I)^{-1}$ and the vector $\boldsymbol{r} = (A - s_0 I)^{-1}B$.

It can be proven that by selecting the columns of the transformation matrix V to span this Krylov subspace, the moments of the reduced model will match the original model up to the first q moments, where q is the size of the Krylov subspace (K_q) [26]. It is apparent that the choice for the selected frequency s_0 affects the quality of the approximation. If $s_0 = 0$, the reduced model will have a better approximation of the original system in the steady-state region. On the other hand, if $s_0 \to \infty$, the moments are called Markov parameters, and the reduced model will result in a better approximation of the transient (high-frequency) region.

Krylov ROM methods can reduce large scale systems efficiently. For this reason, they are commonly used in electronic circuit simulations. However, the stability of the reduced model is not guaranteed, even if the original model is stable. Furthermore, an upper bound error cannot be defined for the reduced system. To reduce the error in the approximation, one can match moments for multiple expansion points. This approach is called *rational interpolation* [12, 25].

Proper Orthogonal Decomposition

The origin of the proper orthogonal decomposition (POD) can be traced back to the paper by Pearson [27] in 1901. In that paper, a statistical technique to extract the dominant characteristics from a set of data was suggested. The idea was to represent the data with a set of basic *principle components*. The method was later developed independently by Hotelling, Loeve, Karhunen, and other scientists [28]. In 1967, Lumley [29] introduced the technique to solve PDE by applying the method to model coherent structures in turbulent flows. Then, an important development to the method occurred in 1987 when Sirovich [30] introduced the *method of snapshots*. Currently, POD can be found across many fields of research under different names; some of the other names are empirical orthogonal functions (usually in meteorology and geophysics), principal component analysis (for discrete random process), common factor analysis, Karhunen-Loeve expansion (for continuous random process), and Hotelling transformation (in image and signal processing) [28]. In the context of ROM, the POD method seeks an approximation that minimizes the error in L_2 norm. The following discussion presents the discrete POD theory as in [31]. The more general continuous POD theory can be found in [28, 32].

If the unknown vector function to be approximated $\left(\boldsymbol{y}(t)\right)$ is sampled at some t_k, then, we require that the error in the approximation of Equation 1.3 is minimized in the ℓ_2 norm sense,

$$E_k = \min_V \|\boldsymbol{y}(t_k) - V\boldsymbol{z}(t_k)\|_{\ell_2}. \tag{1.11}$$

If $\boldsymbol{y}(t)$ is sampled p times $\{t_1, t_2, \ldots, t_p\}$, the sum of the errors is computed as

$$E = \sum_{k=1}^{p} \|\boldsymbol{y}(t_k) - V\boldsymbol{z}(t_k)\|_{\ell_2}. \tag{1.12}$$

We seek to find the basis vectors $\{v_1, v_2, \ldots, v_r\}$ spanning the columns of V and coefficients $\{z_1, z_2, \ldots, z_r\}$ for $\boldsymbol{z}(t)$ that solves the minimization problem Equation 1.11. A constraint is imposed on the columns of transformation matrix V such that they are orthonormal. That is

$$< v_i, v_j > = \begin{cases} 1 & i = j, \\ 0 & i \neq j, \end{cases} \tag{1.13}$$

where v_i is the ith column of the matrix V, and $< \cdot, \cdot >$ is the scalar product. The sampled snapshots can be collected in a matrix

$$M = \left[\boldsymbol{y}(t_1), \, \boldsymbol{y}(t_2), \, \boldsymbol{y}(t_3), \ldots, \boldsymbol{y}\left(t_p\right)\right] \qquad \in R^{\text{n x p}}. \tag{1.14}$$

Then, it can be shown [31] that the solution to the minimization problem is achieved by having the basis vectors to be the first r eigenvectors corresponding to the r largest eigenvalues of the covariance matrix C defined by

$$C = MM^T. \tag{1.15}$$

The eigenvalue of each basis vector is related to the energy (or importance) of that basis vector. If only the first r eigenvectors are chosen, the error in the approximation can

be quantified using the discarded eigenvalues as follows:

$$E_r = \frac{\sum_{k=r+1}^{n} \lambda_k}{\sum_{k=1}^{n} \lambda_k}, \tag{1.16}$$

where λ_k is the kth eigenvalue. This error has an important implication on selecting the size of the basis space r as one can set an upper bound criteria γ_{tr} such that the truncated basis vectors have low contributions (i.e., $E_r < \gamma_{tr}$). Usually, r is selected such that $r \ll n$, where n is the dimension of the original system. The same result can be reached by performing a singular value decomposition (SVD) on the snapshot matrix (proof can be found in [33]). In this case, the basis vectors are the first r left singular vectors $\{v_1, \ldots, v_r\}$ of the SVD, where they are arranged in an order of decreasing singular values ($\{\sigma_i | i = 1, \ldots, r\}$). In this case, the square of the singular values are equal to the eigenvalues of the covariance matrix (i.e., $\lambda_i = \sigma_i^2$) [31]. It is important to note that the snapshot method is not restricted to time-dependent functions. The parameter t can be a pseudo parameter for any combination of parameters $\boldsymbol{\mu}$ and time t of interest.

Once the transformation matrix $V \in R^{n \times r}$ is selected The projection matrix can be chosen such that $W = V$, which satisfies $W^T V = V^T V = I$ because of the orthogonality of the basis.

The orthogonality condition also provides means to compute the coefficients in $\boldsymbol{z}(t)$ at the sampled points as

$$\boldsymbol{y}(t_k) = V\boldsymbol{z}(t_k) \Rightarrow \boldsymbol{z}(t_k) = V^T \boldsymbol{y}(t_k). \tag{1.17}$$

One of the most important features of POD is the ability to represent the sampled data with the highest accuracy compared to any other representation of the same order [34]. However, note that the error in Equation 1.16 quantifies the error in approximating the sampled snapshots. It is not a rigorous error for the reduced model. For any other value of t not included in the snapshot, an upper bound error cannot be guaranteed. For this reason, the selection of the sampled point is of great importance for the success of POD. The derivation of an upper bound error is one of the main challenges in POD approach [12]. Nevertheless, if the sampled snapshots are dense enough to cover the range of dynamics in the system, γ can be taken as a rough indicator for the error in the reduced model.

An extension of the POD method is the Reduced Basis (RB) method [35]. In the RB method, an *a posteriori* error estimation can be derived for the PDE. The error is derived such that its computation is independent from the dimension of the original model in order to be cheaply evaluated. Then, that error function is used to implement the POD with greedy sampling (i.e., iterative sampling) with an error check after each iteration until a certain criterion is met. Error bounds are available only for certain classes of PDEs (see [35–39] and the references therein). The advantage of the RB method is the considerable saving in the offline phase because the iterative greedy sampling approach selects snapshots in locations that have a contribution to the reduced basis. Therefore, oversampling issues are avoided, which also reduces the computational burden of the SVD.

Because of the truncation of the basis space in the POD approach, the reduced model is susceptible to instabilities even with a stable original model. The instability is induced

by truncating modes that have small energy magnitudes but are important for dissipating the energy of the system [40, 41].

1.2.2. Nonlinearity

Projection-based methods can significantly reduce the dimensionality of a large scale linear model, which, in turn, implies a great reduction in computational cost. However, in the nonlinear case, dimensionality reduction does not correlate linearly with the computational savings. This can be illustrated by considering the spatial discretization of Equation 1.1 with the nonlinear term as

$$\frac{d\mathbf{y}(t)}{dt} = A\mathbf{y}(t) + \mathscr{F}\big(\mathbf{y}(t)\big), \tag{1.18}$$

where $\mathbf{y}(t) \in R^n$ is a discretization of the unknown function $y(x, t)$, $A \in R^{n \times n}$ is a discretization matrix of the linear operator $\mathscr{L}(\cdot)$, and $\mathscr{F}$ is a nonlinear function acting on each component of the vector $\mathbf{y}(t)$. A projection onto a subspace is performed in similar manner to the linear case. That is

$$\mathbf{y}(t) \approx \mathbf{y_r}(t) = V\mathbf{z}(t). \tag{1.19}$$

Then, projecting Equation 1.18 onto the subspace V with a projection matrix W^T yields,

$$W^T V \underbrace{\frac{d\mathbf{z}(t)}{dt}}_{\text{r x 1}} = \underbrace{W^T AV}_{\text{r x r}} \mathbf{z}(t) + \underbrace{W^T}_{\text{r x n}} \mathscr{F} \underbrace{(V\mathbf{z}(t))}_{\text{n x 1}}. \tag{1.20}$$

The dimension of the linear terms is reduced, which implies that computing these terms is not dependent on the original dimension of the problem n. However, the nonlinear term $\mathscr{F}(\cdot)$ is still dependent on the original dimension of the system. The nonlinear function needs to be evaluated n times, which results in an inefficient reduced model if n is large.

A direct linearization with Taylor series expansion can overcome the costly computations. Taylor expansion was implemented successfully with Krylov subspace methods in [42, 43] and with balanced truncations in [44]. However, linearization is mostly limited to quadratic expansion because accounting for higher-order terms increases the computational complexity dramatically. Higher accuracy can be achieved with bilinearization of the model, as explained in [45–47]. Nevertheless, linearization and bilinearization methods are both inherently limited to local accuracy. To have a more global accuracy, the Trajectory-Piecewise-Linear (TPWL) method was suggested [48]. The idea is to employ a first order linearization at several selected expansion points. Then, a model for the system is obtained by combining these models with a weighted sum. TPWL can be applied in combination with POD [49], Krylov subspace [50] and balanced truncation [51]. However, the choice for the expansion points is extremely important for the success of the model. Moreover, some nonlinear functions cannot be represented adequately with piecewise low order polynomials.

It is important to highlight that balanced truncation and Krylov subspace methods are only valid in the linear case. Therefore, linearization is essential for their applicability. POD, on the other hand, is valid even for nonlinear models. For this reason, POD is

preferred for highly nonlinear systems. The only difficulty that arises, in this case, is the computational cost of the nonlinear term. Nevertheless, POD methods can exploit the data generated from the snapshots to build an approximation for the nonlinear term. This is the basis for the Empirical Interpolation Method (EIM) and its variant: Discrete Empirical Interpolation (DEIM) [52, 53]. In this approach, snapshots of the nonlinear function obtained from the high-fidelity evaluations are stored in a separate matrix. Then, a POD approach is applied to generate a separate subspace basis for the nonlinear term. The coefficient values are then interpolated to solve for the function values at the required point. The method is similar to the nonintrusive POD described in the following section. However, this approach requires that the nonlinear term has a known analytical form or that the solver can export snapshots of the nonlinear term separately.

1.2.3. NONINTRUSIVE APPROACHES
Nonintrusive methods are also called surrogate-based, data-fit, and pattern identification. The concept is based on collecting data from the high-fidelity model (or an experiment) as much as affordably possible. Then, the data is analysed to build a model that captures the relationship between the input of interest and the desired output. Unlike intrusive methods, these methods do not require access to the governing equations of the system. This advantage allows nonintrusive methods to be applied to virtually any problem without restrictions. However, due to the lack of the underlying physical structure in constructing these models, careful selection of the snapshots points is of utmost importance in nonintrusive methods [54]. Broadly, two classes of nonintrusive methods can be identified. The first, which can be called grey-box (or structured) methods, attempts to recover the physical structure of the problem by inferring an assumed operator from the data. The second class is *black-box* (or *unstructured*) methods, which are constructed purely based on the generated data without any physical insight into the system.

GREY-BOX MODELLING
In grey-box modeling, an assumed structured form for the system is constructed based on some knowledge of the system. An example of grey-box ROM methods is the Dynamic Mode Decomposition (DMD), which was first suggested in [55]. DMD approximates the operator of a dynamic system by fitting the generated data in an optimal least square sense. If the data are generated at fixed intervals, a linear mapping from each snapshot to the next can be assumed as

$$y(t_{i+1}) = Ay(t_i), \tag{1.21}$$

where $y(t_i)$ is a snapshot generated at t_i and A is the system matrix (or operator) to be estimated. While the mapping is true if the system is linear, nonlinear systems can only be approximated with such linear mapping. After successive generation of snapshots, the snapshots matrix can be shown to span a Krylov subspace as follows:

$$K_q(A, y_1) = span\{y_1, Ay_1, A^2 y_1, \dots A^{q-1} y_1\}, \tag{1.22}$$

where $y_i = y(t_i)$. The eigenvectors and eigenvalues of the matrix A can be estimated from the data using Krylov algorithms. Once A is known, the system is propagated in time. The approach can also be applied to a steady-state system parametrized with a

1

single parameter. However, the method is not directly applicable to multi-parametric problems [56].

A different grey-box approach is the Loewner framework [57], which is a nonintrusive version of the rational interpolation approach described under Krylov subspace methods (Section 1.2.1). In this approach, a reduced model for the system is constructed by interpolating measurements of the transfer function in the frequency domain. This approach was extended to construct a reduced model from time-domain data [58]. However, reduced models in the Loewner framework are only applicable to LTI systems. Another approach that is similar to DMD is the operator inference approach [59]. In this approach, the generated data are fitted to a parametrized dynamic model with nonlinear terms of low order polynomials. Further development to generalize this work to higher and non-polynomial nonlinearities suggested using auxiliary variables to lift the generated data to a quadratic form. Then, apply the operator inference approach to the lifted system [60]. However, defining the lifting maps is problem specific and requires characterization of the nonlinear term, which is an intrusive step.

BLACK-BOX MODELLING

Black-box methods are closer to *machine learning* techniques. They use generated data to fit a surrogate model mapping a defined input space to the desired output space, regardless of the physics of the problem. Classical machine learning methods were developed primarily in the computer sciences and statistics field to identify patterns in big data. Therefore, they are usually trained on an abundance of data. However, in computational science and engineering applications (both numerical and experimental), data are typically expensive to generate. Therefore, an important challenge to overcome for black-box ROM methods is to build an accurate model with limited data.

The predominant surrogates are the polynomial surface response method (SRM), methods using radial basis functions (RBF), and Kriging. Excellent survey papers comparing the different methods can be found in [61–64]. General guidelines can be found in these papers on their application based on complexity and flexibility. However, one common conclusion all nonintrusive comparative studies reach is the non-existence of a single method for all types of problems. Certain methods may outperform others depending on the problem considered, but predicting which method delivers the best results is difficult beforehand.

Applying the surrogate models directly on each state or response of the system is expensive for large-scale systems and can lead to inconsistencies in the physics or boundary conditions of the problem. A recent development in this area to address such issues combines the POD method with a surrogate model [65]. This approach starts in a similar way to the projection-based version by constructing a reduced basis space from snapshots of the system. However, instead of projecting the high-fidelity model equations onto the reduced basis space to solve for the POD coefficients, data-fit surrogate models for the POD expansion coefficients are employed. This is achievable because the coefficient values at the snapshot points can be computed without any projection, as shown in Equation 1.17. The problem, then, becomes training a surrogate model for the coefficients of the POD basis vectors. The surrogate model can be a simple interpolation or splines as in [66] or more advanced techniques such as RBF [31, 67–70]. Gaussian regression process (or Kriging) is another option to build the surrogate model [71–73].

Alternatively, classical machine learning techniques such as neural networks can be used to learn the surrogate model [74–81]. A comparison between different machine learning methods for POD-based ROM modelling has also been investigated [82]. Another interesting approach suggests using a sparse grid interpolant to find the coefficient [83, 84].

1.2.4. ROM in Nuclear Reactor Applications

Although limited in quantity, most of the work on ROM methods for nuclear applications has focused on projection-based POD methods. The reason can be attributed to the superior performance of POD in nonlinear problems compared to Krylov or balanced truncation methods. Projection-based POD has been applied to solve the eigenvalue problem [85–89], for pin-by-pin reactor core calculations [90], in fuel burnup calculations [91], in thermal hydraulics modeling [92], in stability analysis [93, 94], in spent fuel pool modeling [95], and to model the lead cooled fast reactor [96].

On the other hand, nonintrusive approaches have not been fully adopted in the nuclear community. Only a limited number of publications can be found on the topic. Failure domains in nuclear systems have been identified using machine learning techniques [97]. DMD has been employed to model the MSFR [98]. In addition, nonintrusive POD method based on Range Finding Algorithm (RFA) has been used in [99, 100] to build the reduced basis space (referred to as active subspace) combined with a simple polynomial regression surrogate for the POD coefficients.

1.3. Research Objectives

Most of the computer codes in practical reactor physics applications are either closed-source or legacy codes that have benefited from years of development and gone through a rigorous process of certification by regulatory bodies. Such codes are difficult to access or modify for intrusive approaches. For this reason, there is a pressing need for novel and creative nonintrusive approaches in the field of nuclear applications. Additionally, while smart sampling strategies are developed for intrusive approaches, such as the greedy algorithm in the RB method, they are lacking in nonintrusive approaches.

Therefore, the goal of this research is to develop a nonintrusive methodology for constructing a reduced-order model in applications involving large-scale, complex models of nuclear reactors. Particularly, the research has the following contributions:

- Offer a systematic nonintrusive ROM method that can work with any general PDE solver including the validated, high-fidelity reactor physics codes;

- Address the key challenges in constructing reduced-order models for systems with high-dimensional input parameter spaces both in steady-state and transient applications;

- Develop a criterion for adaptive sampling strategies in nonintrusive settings;

- As an application for the developed methodology, analyse the large-scale Molten Salt Fast Reactor (MSFR) and perform a parametric study for uncertainty quantification and sensitivity analysis.

1

Because nuclear reactor models are nonlinear, the focus of the research is on POD methods since they offer better handling of the nonlinearity compared to balanced truncations and Krylov subspace methods. The nonintrusive-POD route is of particular interest because of the need for nonintrusive approaches in the nuclear community. Exploring the use of sparse grids to deal with higher-dimensional parameter spaces is an underpinning of this work.

1.4. THESIS ORGANIZATION

The thesis is organized as a collection of articles. Each chapter is written as a self-contained scientific paper. The order of the chapters correlates with the progress of the research. For this reason, some overlapping between the chapters can be observed, especially in the theoretical formulation section of each chapter since the developed theory in one chapter is built upon in the subsequent work. The remainder of the thesis is organized as follows: **Chapter** 2 compares two nonintrusive POD methods: RBF and sparse grids interpolant. Then, **Chapter** 3 presents a nonintrusive adaptive POD algorithm for parametrized steady-state PDE. The algorithm is demonstrated on three numerical examples. **Chapter** 4 tests the developed algorithm on a larger-scale two-dimensional system of fuelled molten salt with an input parameter space of 27 dimensions. In this chapter, we compare two approaches for handling multiple outputs. The chapter also demonstrates an approach to using the constructed reduced model in uncertainty and (both local and global) sensitivity analysis. **Chapter** 5 extends the developed algorithm to time-dependent parametrized problems. We propose an approach for selecting snapshots that is fully adaptive in both time and parameter spaces. Three test cases were presented in this chapter to show the effectiveness of the time adaptive approach. In **Chapter** 6, the developed algorithm is applied to a high-fidelity three-dimensional MSFR model for steady-state and transient analysis. In the steady-state analysis, a study of 30 model parameters was conducted for uncertainty quantification and sensitivity analysis. For the transient analysis, a transient reduced-order model is built for the fission power and temperature distributions as a function of the flow in the secondary loop. Finally, conclusions and recommendations are discussed in **Chapter** 7.

REFERENCES

[1] W. H. A. Schilders, H. A. van der Vorst, and J. Rommes, eds., *Model Order Reduction: Theory, Research Aspects and Applications* (Springer Berlin Heidelberg, 2008).

[2] U. Baur, P. Benner, A. Greiner, J. Korvink, J. Lienemann, and C. Moosmann, *Parameter preserving model order reduction for MEMS applications,* Mathematical and Computer Modelling of Dynamical Systems **17**, 297 (2011).

[3] K. Bizon, G. Continillo, L. Russo, and J. Smuła, *On POD reduced models of tubular reactor with periodic regimes,* Computers & Chemical Engineering **32**, 1305 (2008).

[4] R. Bourguet, M. Braza, and A. Dervieux, *Reduced-order modeling of transonic flows around an airfoil submitted to small deformations,* Journal of Computational Physics **230**, 159 (2011).

[5] M. W. Hess and P. Benner, *A reduced basis method for microwave semiconductor devices with geometric variations,* COMPEL - The international journal for computation and mathematics in electrical and electronic engineering **33**, 1071 (2014).

[6] T. Lieu and C. Farhat, *Adaptation of Aeroelastic Reduced-Order Models and Application to an F-16 Configuration,* AIAA Journal **45**, 1244 (2007).

[7] A. Placzek, D.-M. Tran, and R. Ohayon, *A nonlinear POD-Galerkin reduced-order model for compressible flows taking into account rigid body motions,* Computer Methods in Applied Mechanics and Engineering **200**, 3497 (2011).

[8] P. Vermeulen, A. Heemink, and C. T. Stroet, *Reduced models for linear groundwater flow models using empirical orthogonal functions,* Advances in Water Resources **27**, 57 (2004).

[9] M. Xu, P. van Overloop, and N. van de Giesen, *Model reduction in model predictive control of combined water quantity and quality in open channels,* Environmental Modelling & Software **42**, 72 (2013).

[10] A. Marquez, J. J. E. Oviedo, and D. Odloak, *Model Reduction Using Proper Orthogonal Decomposition and Predictive Control of Distributed Reactor System,* Journal of Control Science and Engineering **2013**, 1 (2013).

[11] D. Amsallem, S. Deolalikar, F. Gurrola, and C. Farhat, *Model predictive control under coupled fluid-structure constraints using a database of reduced-order models on a tablet,* in *21st AIAA Computational Fluid Dynamics Conference* (American Institute of Aeronautics and Astronautics, 2013).

[12] P. Benner, S. Gugercin, and K. Willcox, *A Survey of Projection-Based Model Reduction Methods for Parametric Dynamical Systems,* SIAM review **57**, 483 (2015).

[13] U. Baur, P. Benner, and L. Feng, *Model order reduction for linear and nonlinear systems: A system-theoretic perspective,* Archives of Computational Methods in Engineering **21**, 331 (2014).

[14] A. C. Antoulas, D. C. Sorensen, and S. Gugercin, *A survey of model reduction methods for large-scale systems,* Contemporary Mathematics **280**, 193 (2001).

[15] F. Lihong, *Review of model order reduction methods for numerical simulation of nonlinear circuits,* **167**, 576 (2005).

[16] F. Chinesta, A. Huerta, G. Rozza, and K. Willcox, *Model Reduction Methods,* in *Encyclopedia of Computational Mechanics Second Edition* (American Cancer Society, 2017) pp. 1–36.

[17] B. Moore, *Principal component analysis in linear systems: Controllability, observability, and model reduction,* IEEE Transactions on Automatic Control **26** (1981), 10.1109/tac.1981.1102568.

[18] S. Gugercin and A. C. Antoulas, *A survey of model reduction by balanced truncation and some new results,* International Journal of Control **77**, 748 (2004).

[19] T. Bui-Thanh, K. Willcox, O. Ghattas, and B. van Bloemen Waanders, *Goal-oriented, model-constrained optimization for reduction of large-scale systems,* Journal of Computational Physics **224**, 880 (2007).

[20] V. Druskin, L. Knizhnerman, and V. Simoncini, *Analysis of the Rational Krylov Subspace and ADI Methods for Solving the Lyapunov Equation,* SIAM Journal on Numerical Analysis **49**, 1875 (2011).

[21] P. Benner, J.-R. Li, and T. Penzl, *Numerical solution of large-scale Lyapunov equations, Riccati equations, and linear-quadratic optimal control problems,* Numerical Linear Algebra with Applications **15**, 755 (2008).

[22] J. R. Li and J. White, *Low-Rank Solution of Lyapunov Equations,* SIAM Review **46**, 260 (2004).

[23] T. Penzl, *A Cyclic Low-Rank Smith Method for Large Sparse Lyapunov Equations,* SIAM Journal on Scientific Computing **21**, 1401 (1999).

[24] J. Dongarra and F. Sullivan, *Guest editors introduction to the top 10 algorithms,* Computing in Science & Engineering **2**, 22 (2000).

[25] Z. Bai, *Krylov subspace techniques for reduced-order modeling of large-scale dynamical systems,* Applied Numerical Mathematics **43**, 9 (20029).

[26] B. Salimbahrami and B. Lohmann, *Krylov subspace methods in linear model order reduction: Introduction and invariance properties,* in *Sci. Rep. Institute of Automation* (University of Bremen, 2002).

[27] K. Pearson, *On lines and planes of closest fit to systems of points in space,* The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science **2**, 559 (1901).

[28] A. Quarteroni and G. Rozza, eds., *Reduced Order Methods for Modeling and Computational Reduction* (Springer Science mathplus Business Media, 2014).

[29] J. Lumley, *Coherent structures in turbulence,* in *Transition and Turbulence* (Elsevier, 1981) pp. 215–242.

[30] L. Sirovich, *Turbulence and the dynamics of coherent structures part I: coherent structures,* Quarterly of applied mathematics **45**, 561 (1987).

[31] V. Buljak, *Inverse Analyses with Model Reduction: Proper Orthogonal Decomposition in Structural Mechanics* (Springer, Berlin, 2012).

[32] K. Kunisch and S. Volkwein, *Galerkin proper orthogonal decomposition methods for parabolic problems,* Numerische Mathematik **90**, 117 (2001).

[33] Y. Liang, H. Lee, S. Lim, W. Lin, K. Lee, and C. Wu, *Proper Orthogonal Decomposition and its Applications—Part I: Theory,* Journal of Sound and Vibration **252**, 527 (2002).

[34] N. Aubry, *On the hidden beauty of the Proper Orthogonal Decomposition,* in *Studies in Turbulence*, Vol. 2 (Springer New York, 1992) pp. 264–265.

[35] G. Rozza, D. B. P. Huynh, and A. T. Patera, *Reduced Basis Approximation and a Posteriori Error Estimation for Affinely Parametrized Elliptic Coercive Partial Differential Equations,* Archives of Computational Methods in Engineering **15**, 229 (2008).

[36] J. S. Hesthaven, G. Rozza, and B. Stamm, *Certified Reduced Basis Methods for Parametrized Partial Differential Equations* (Springer International Publishing, 2016).

[37] D. B. P. Huynh and A. T. Patera, *Reduced basis approximation and a posteriori error estimation for stress intensity factors,* International Journal for Numerical Methods in Engineering **72**, 1219 (2007).

[38] D. Klindworth, M. A. Grepl, and G. Vossen, *Certified reduced basis methods for parametrized parabolic partial differential equations with non-affine source terms,* Computer Methods in Applied Mechanics and Engineering **209-212**, 144 (2012).

[39] A. Quarteroni, G. Rozza, and A. Manzoni, *Certified reduced basis approximation for parametrized partial differential equations and applications,* Journal of Mathematics in Industry **1**, 3 (2011).

[40] M. Couplet, P. Sagaut, and C. Basdevant, *Intermodal energy transfers in a proper orthogonal decomposition–Galerkin representation of a turbulent separated flow,* Journal of Fluid Mechanics **491**, 275 (2003).

[41] 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 **311**, 151 (2016).

[42] Y. Chen and J. White, *A quadratic method for nonlinear model order reduction,* in *International Conference on Modeling and Simulation of Microsystems* (2000) pp. 477–480.

[43] J. Chen and S. M. Kang, *An algorithm for automatic model-order reduction of nonlinear MEMS devices,* in *2000 IEEE International Symposium on Circuits and Systems (ISCAS)*, Vol. 2 (Presses Polytech. Univ. Romandes, 2000) pp. 445–448.

[44] K. Fujimoto and D. Tsubakino, *On computation of nonlinear balanced realization and model reduction,* in *2006 American Control Conference* (Institute of Electrical and Electronics Engineers (IEEE), 2006).

[45] P. Benner and T. Breiten, *Krylov-Subspace Based Model Reduction of Nonlinear Circuit Models Using Bilinear and Quadratic-Linear Approximations,* in *Progress in Industrial Mathematics at ECMI 2010*, edited by M. Günther, A. Bartel, M. Brunk, S. Schöps, and M. Striebel (Springer Berlin Heidelberg, 2012) pp. 153–159.

[46] M. Condon and R. Ivanov, *Nonlinear systems –algebraic gramians and model reduction,* COMPEL - The international journal for computation and mathematics in electrical and electronic engineering **24**, 202 (2005).

[47] J. R. Phillips, *Projection frameworks for model reduction of weakly nonlinear systems,* in *Proceedings of the 37th Annual Design Automation Conference* (ACM, 2000) pp. 184–189.

[48] M. J. Rewienski, *A trajectory piecewise-linear approach to model order reduction of nonlinear dynamical systems,* Ph.D. thesis, Massachusetts Institute of Technology (2003).

[49] D. Gratton and K. Willcox, *Reduced-order, trajectory piecewise-linear models for nonlinear computational fluid dynamics,* in *34th AIAA Fluid Dynamics Conference and Exhibit* (American Institute of Aeronautics and Astronautics, 2004).

[50] M. Rewienski and J. White, *A trajectory piecewise-linear approach to model order reduction and fast simulation of nonlinear circuits and micromachined devices,* in *Proceedings of the 2001 IEEE/ACM International Conference on Computer-aided Design,* Vol. 22 (Institute of Electrical and Electronics Engineers (IEEE), 2003) pp. 155–170.

[51] D. Vasilyev, M. Rewienski, and J. White, *A TBR-based Trajectory Piecewise-linear algorithm for generating accurate low-order models for nonlinear analog circuits and MEMS,* in *Proceedings of the 40th Annual Design Automation Conference* (ACM, 2003) pp. 490–495.

[52] S. Chaturantabut and D. C. Sorensen, *Nonlinear model reduction via discrete empirical interpolation,* SIAM Journal on Scientific Computing **32**, 2737 (2010).

[53] H. Antil, M. Heinkenschloss, and D. C. Sorensen, *Application of the Discrete Empirical Interpolation Method to Reduced Order Modeling of Nonlinear and Parametric Systems,* in *Reduced Order Methods for Modeling and Computational Reduction,* edited by Q. A. R. Gianluigi (Springer International Publishing, 2014) pp. 101–136.

[54] A. I. J. Forrester, A. Sóbester, and A. J. Keane, *Engineering Design Via Surrogate Modelling* (Wiley-Blackwell, 2008).

[55] P. J. Schmid, *Dynamic mode decomposition of numerical and experimental data,* Journal of Fluid Mechanics **656**, 5 (2010).

[56] J. H. Tu, , C. W. Rowley, D. M. Luchtenburg, S. L. Brunton, and J. N. Kutz, *On dynamic mode decomposition: Theory and applications,* Journal of Computational Dynamics **1**, 391 (2014).

[57] A. Antoulas, A. Ionita, and S. Lefteriu, *On two-variable rational interpolation,* Linear Algebra and its Applications **436**, 2889 (2012).

[58] B. Peherstorfer, S. Gugercin, and K. Willcox, *Data-driven reduced model construction with time-domain loewner models,* SIAM Journal on Scientific Computing **39**, A2152 (2017).

[59] B. Peherstorfer and K. Willcox, *Data-driven operator inference for nonintrusive projection-based model reduction,* Computer Methods in Applied Mechanics and Engineering **306**, 196 (2016).

[60] E. Qian, B. Kramer, B. Peherstorfer, and K. Willcox, *Lift & learn: Physics-informed machine learning for large-scale nonlinear dynamical systems,* Physica D: Nonlinear Phenomena **406**, 132401 (2020).

[61] R. R. Barton, *Simulation optimization using metamodels,* in *Proceedings of the 2009 Winter Simulation Conference (WSC)* (Winter Simulation Conference, 2009) pp. 230–238.

[62] N. V. Queipo, R. T. Haftka, W. Shyy, T. Goel, R. Vaidyanathan, and P. K. Tucker, *Surrogate-based analysis and optimization,* Progress in Aerospace Sciences **41**, 1 (2005).

[63] T. Simpson, J. Poplinski, P. N. Koch, and J. Allen, *Metamodels for computer-based engineering design: survey and recommendations,* Engineering with Computers **17**, 129 (2001).

[64] R. Jin, W. Chen, and T. Simpson, *Comparative studies of metamodeling techniques under multiple modeling criteria,* in *8th Symposium on Multidisciplinary Analysis and Optimization* (American Institute of Aeronautics and Astronautics, 2000).

[65] P. Breitkopf and R. F. Coelho, eds., *Multidisciplinary Design Optimization in Computational Mechanics* (John Wiley & Sons, Inc., 2013).

[66] H. V. Ly and H. T. Tran, *Modeling and Control of Physical Processes using Proper Orthogonal Decomposition,* Mathematical and Computer Modelling **33**, 223 (2001).

[67] C. Audouze, F. D. Vuyst, and P. B. Nair, *Nonintrusive reduced-order modeling of parametrized time-dependent partial differential equations,* Numerical Methods for Partial Differential Equations **29**, 1587 (2013).

[68] M. Guénot, I. Lepot, C. Sainvitu, J. Goblet, and R. Filomeno Coelho, *Adaptive Sampling Strategies for Non-intrusive POD-based Surrogates,* Engineering Computations **30**, 521 (2013).

[69] D. Xiao, F. Fang, C. Pain, and G. Hu, *Non-intrusive reduced-order modelling of the Navier-Stokes equations based on RBF interpolation,* International Journal for Numerical Methods in Fluids **79**, 580 (2015).

[70] S. Walton, O. Hassan, and K. Morgan, *Reduced order modelling for unsteady fluid flow using proper orthogonal decomposition and radial basis functions,* Applied Mathematical Modelling **37**, 8930 (2013).

1

[71] N. Nguyen and J. Peraire, *Gaussian functional regression for output prediction: Model assimilation and experimental design,* Journal of Computational Physics **309**, 52 (2016).

[72] M. Xiao, P. Breitkopf, R. F. Coelho, C. Knopf-Lenoir, M. Sidorkiewicz, and P. Villon, *Model reduction by CPOD and Kriging,* Structural and Multidisciplinary Optimization **41**, 555 (2009).

[73] M. Guo and J. S. Hesthaven, *Data-driven reduced order modeling for time-dependent problems,* Computer Methods in Applied Mechanics and Engineering **345**, 75 (2019).

[74] J. Hesthaven and S. Ubbiali, *Non-intrusive reduced order modeling of nonlinear problems using neural networks,* Journal of Computational Physics **363**, 55 (2018).

[75] A. T. Mohan and D. V. Gaitonde, *A Deep Learning based Approach to Reduced Order Modeling for Turbulent Flow Control using LSTM Neural Networks,* (2018), arXiv:1804.09269 .

[76] F. Regazzoni, L. Dedè, and A. Quarteroni, *Machine learning for fast and reliable solution of time-dependent differential equations,* Journal of Computational Physics **397**, 108852 (2019).

[77] R. Hu, F. Fang, C. Pain, and I. Navon, *Rapid spatio-temporal flood prediction and uncertainty quantification using a deep learning method,* Journal of Hydrology **575**, 911 (2019).

[78] O. San, R. Maulik, and M. Ahmed, *An artificial neural network framework for reduced order modeling of transient flows,* Communications in Nonlinear Science and Numerical Simulation **77**, 271 (2019).

[79] Z. Deng, Y. Chen, Y. Liu, and K. C. Kim, *Time-resolved turbulent velocity field reconstruction using a long short-term memory LSTM-based artificial intelligence framework,* Physics of Fluids **31**, 075108 (2019).

[80] S. Pawar, S. M. Rahman, H. Vaddireddy, O. San, A. Rasheed, and P. Vedula, *A deep learning enabler for nonintrusive reduced order modeling of fluid flows,* Physics of Fluids **31**, 085101 (2019).

[81] H. F. S. Lui and W. R. Wolf, *Construction of reduced-order models for fluid flows using deep feedforward neural networks,* Journal of Fluid Mechanics **872**, 963 (2019).

[82] R. Swischuk, L. Mainini, B. Peherstorfer, and K. Willcox, *Projection-based model reduction: Formulations for physics-based machine learning,* Computers & Fluids **179**, 704 (2019).

[83] B. Peherstorfer, *Model Order reduction of Parametrized Systems with Sparse Grid Learning Techniques,* Ph.D. thesis, Technische Universität München, München (2013).

[84] D. Xiao, F. Fang, C. Pain, and I. Navon, *A parameterized non-intrusive reduced order model and error analysis for general time-dependent nonlinear partial differential equations and its applications,* Computer Methods in Applied Mechanics and Engineering **317**, 868 (2017).

[85] A. G. Buchan, C. C. Pain, F. Fang, and I. M. Navon, *A POD reduced-order model for eigenvalue problems with application to reactor physics,* International Journal for Numerical Methods in Engineering **95**, 1011 (2013).

[86] 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 Rreactor Spatial Kinetics,* Annals of Nuclear Energy **71**, 217 (2014).

[87] J. P. Senecal and W. Ji, *Characterization of the proper generalized decomposition method for fixed-source diffusion problems,* Annals of Nuclear Energy **126**, 68 (2019).

[88] P. German and J. C. Ragusa, *Reduced-order modeling of parameterized multi-group diffusion k-eigenvalue problems,* Annals of Nuclear Energy **134**, 144 (2019).

[89] Z. M. Prince and J. C. Ragusa, *Application of proper generalized decomposition to multigroup neutron diffusion eigenvalue calculations,* Progress in Nuclear Energy **121**, 103232 (2020).

[90] A. Cherezov, R. Sanchez, and H. G. Joo, *A reduced-basis element method for pin-by-pin reactor core calculations in diffusion and SP_3 approximations,* Annals of Nuclear Energy **116**, 195 (2018).

[91] C. Castagna, M. Aufiero, S. Lorenzi, G. Lomonaco, and A. Cammi, *Development of a reduced order model for fuel burnup analysis,* Energies **13**, 890 (2020).

[92] L. Vergari, A. Cammi, and S. Lorenzi, *Reduced order modeling approach for parametrized thermal-hydraulics problems: inclusion of the energy equation in the POD-FV-ROM method,* Progress in Nuclear Energy **118**, 103071 (2020).

[93] D. Prill and A. Class, *Semi-automated proper orthogonal decomposition reduced order model non-linear analysis for future BWR stability,* Annals of Nuclear Energy **67**, 70 (2014).

[94] R. Manthey, A. Knospe, C. Lange, D. Hennig, and A. Hurtado, *Reduced order modeling of a natural circulation system by proper orthogonal decomposition,* Progress in Nuclear Energy **114**, 191 (2019).

[95] J. Y. Escanciano and A. G. Class, *POD-Galerkin modeling of a heated pool,* Progress in Nuclear Energy **113**, 196 (2019).

[96] 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 **87**, 198 (2016).

1

1

[97] F. D. Maio, A. Bandini, E. Zio, A. Alfonsi, and C. Rabiti, *An approach based on Support Vector Machines and a K-D Tree search algorithm for identification of the failure domain and safest operating conditions in nuclear systems,* Progress in Nuclear Energy **88**, 297 (2016).

[98] A. D. Ronco, C. Introini, E. Cervi, S. Lorenzi, Y. S. Jeong, S. B. Seo, I. C. Bang, F. Giacobbo, and A. Cammi, *Dynamic mode decomposition for the stability analysis of the Molten Salt Fast Reactor core,* Nuclear Engineering and Design **362**, 110529 (2020).

[99] Y. Bang, H. S. Abdel-Khalik, M. A. Jessee, and U. Mertyurek, *Hybrid reduced order modeling for assembly calculations,* Nuclear Engineering and Design **295**, 661 (2015).

[100] D. Huang, H. Abdel-Khalik, C. Rabiti, and F. Gleicher, *Dimensionality reducibility for multi-physics reduced order modeling,* Annals of Nuclear Energy **110**, 526 (2017).

2

A Nonintrusive POD Approach Using Classical Sparse Grids

Reduced order models are effective in reducing the computational burden of large-scale complex systems. Proper Orthogonal Decomposition (POD) is one of the most important methods for such application. Nevertheless, problems parametrized on high dimensional spaces require computations of an enormous number of simulations in the offline phase. In this chapter, the use of sparse grids is suggested to select the sampling points in an efficient manner. The method exploits the hierarchical nature of the Smolyak algorithm to select the sparse grid level based on the singular values of the POD basis. Then, a nonintrusive reduced order model is built using Smolyak's combination technique. The proposed method was tested and compared with Radial Basis Functions in two nuclear applications. The first was a one-dimensional slab solved as a diffusion eigenvalue problem and the second was the two-dimensional IAEA benchmark problem. In both cases, the results showed that while Radial Basis Functions resulted in a faster reduced order model, Smolyak's model provided superior accuracy.

2.1. INTRODUCTION

MODELING nuclear reactors is a challenging task that involves capturing the interaction between multi-physics phenomena occurring at various scales. In the reactor design process, high fidelity simulation tools are often used to provide a comprehensive solution for the coupled inter-disciplinary problem. Nevertheless, even with the increasing power of today's supercomputers, high fidelity models require a tremendous amount of computational time and memory allocation. For applications of design optimization, uncertainty quantification and control, where many repeated model evaluations are needed, such models are rendered extremely expensive.

Reduced Order Modeling is an effective technique to reduce the dimensionality of large-scale complex systems. The reduction is achieved by replacing the high fidelity model with a low-dimensional efficient model capturing the prominent dynamics of the system. The reduced model can then be used to provide fast solutions with a controlled level of accuracy. Different reduced order modeling techniques can be found in the literature. However, Proper Orthogonal Decomposition (POD) is the favoured method for nonlinear systems [2]. POD was first introduced as a statistical technique to extract dominant characteristics from a set of data. The idea was to represent the data with a set of basic principle components. As a reduced order method, the method was later developed by Lumely [3] to model coherent structures in turbulent flow. The POD method is based on sampling the high fidelity model at several points in the parameter space to construct the so-called snapshot matrix. Then, a reduced basis is created through a Singular Value Decomposition (SVD). The original high fidelity model is then projected onto the created reduced basis space by means of a Galerkin projection. The generation of the snapshot matrix and the building of the model are accomplished in the offline phase, which is executed only once. Afterwards, in the online phase, the generated reduced model can be run inexpensively at any desired parameter point. In reactor physics applications, POD model order reduction was applied to the eigenvalue problem of the diffusion equation in Ref. [4] and to the time-dependent diffusion equation in Ref. [5].

However, projection-based POD methods are code intrusive, which is a major limitation. For legacy codes where access to the governing equations is not possible, the approach is not applicable. For such cases, a slightly different nonintrusive POD technique can be employed. The idea is to benefit from the orthogonality of the subspace basis to generate the Galerkin expansion coefficients at the sampled points. Thus, the coefficient values at the snapshots points are computed without any projection. Then, a surrogate model can be constructed to compute the solution at any required non-sampled point. In literature, different surrogate models have been suggested to compute the expansion coefficients. For lower dimensional problems, direct interpolation or splines can be used as in Ref. [6]. On the other hand, high-dimensionality problems require more advanced techniques. Radial Basis Function (RBF) is one of the commonly used methods in such applications [7].

Nevertheless, the accuracy of the POD method to provide a solution at a non-sampled point is directly affected by the choice of the sampling scheme. The snapshots need to capture the entire dynamics of the model within the desired range. Moreover, in the nonintrusive approach, the sampling points should be dense enough for the surrogate model to reproduce a reliable predictive solution at non-sampled locations. Thus, for

nonintrusive methods, the sampling strategy becomes even more relevant. In addition, problems parametrized on high dimensional spaces are prone to the curse of dimensionality (i.e., the exponential increase of the computational time with the increase in the number of dimensions). In these cases, the efficient selection of the sampling points is crucial for any practical application. Latin Hypercube Sampling (LHS) can be an efficient sampling technique to address this issue. However, the lack of adaptivity can be limiting in nonlinear cases. An extension of LHS was suggested in Ref. [8]. The technique improves the initial snapshot matrix by adaptively selecting new points based on the "influence" of the new point on the snapshot matrix.

In this chapter, a different sampling method based on sparse grids is suggested. Sparse grids were first introduced by Smolyak [9] and, ever since, has been used to cope with the curse of dimensionality in multivariate integrations and interpolations. It involves preserving the interpolation property for the unidimensional formula by a specific combination of the tensorized product [10]. In the context of reduced order models, sparse grids were suggested by Peherstorfer [11] to be used as a machine learning tool to build a reduced order model. The approach was tested on heat transfer problems. Also, Xiao [12] presented a method of propagating the expansion coefficients through time with the use of a sparse grid interpolant. The method was tested on the Navier-Stokes equations. This chapter, however, presents an approach to exploit the hierarchical nature of the Smolyak's algorithm and select the sparse grid level based on the singular values of the POD basis. Then, an efficient surrogate reduced order model is built using a Smolyak interpolant. The approach can be extended to higher dimensional problems inexpensively. Although the nonintrusive approach is considered in this chapter, it is important to note that the proposed sampling method can equally be combined with a Galerkin-POD approach. In this work, the method is tested on two cases. The first is a one-dimensional diffusion eigenvalue problem and the second is the 2D IAEA benchmark problem. A comparison between Smolyak's interpolant and RBF method is presented.

2.2. THEORY

2.2.1. PROPER ORTHOGONAL DECOMPOSITION

In nonintrusive applications, the high fidelity model is considered as a black box mapping a given input to the desired output. The model, thus, can be seen as an unknown function $\boldsymbol{f} : [0\ 1]^d \rightarrow \mathbb{R}^n$, where d is the dimension of the input defined in the unit hypercube. The function $\boldsymbol{f}(x; \lambda)$ is dependent on state x and the input parameters of interest λ. In a Galerkin expansion, the function can be written as a linear combination of basis functions:

$$\boldsymbol{f}(x; \lambda) = \sum_{i=1}^{r} c_i(\lambda)\, \boldsymbol{u}_i(x) \tag{2.1}$$

where c_i are the expansion coefficients which depend on the input parameter λ and $\boldsymbol{u}_i(x)$ are the basis functions.

The POD method seeks an approximation of the function that minimizes the error in L_2 norm [13],

$$\min_{\boldsymbol{u}_i(x)} \left\| \boldsymbol{f}(x) - \sum_{i=1}^{k} c_i\, \boldsymbol{u}_i(x) \right\|_{L2} \tag{2.2}$$

The basis functions are chosen such that they are orthonormal. Thus, the coefficients c_i can be computed as follows:

$$c_i = \int_\Omega \boldsymbol{f}(x)\,\boldsymbol{u}_i(x)\,d\Omega \tag{2.3}$$

The solution to the minimization problem can be reached with the Singular Value Decomposition (SVD) as follows [7]:

1. Sample the function $\boldsymbol{f}$ at some preselected sampling points λ_k;

2. Arrange the solutions to construct the snapshot matrix $M = \{\boldsymbol{f}(\lambda_1), \boldsymbol{f}(\lambda_2), \ldots, \boldsymbol{f}(\lambda_p)\}$, where p is the number of simulations;

3. Perform Singular Value Decomposition on the snapshot matrix $M \rightarrow UDV$ to obtain a matrix U whose columns are the left singular vectors, matrix V whose columns are right singular vectors and a diagonal matrix D with entries σ_k corresponding to the singular values of the snapshot matrix arranged in a descending order.

If the number of non-zero singular values is w, it can be shown that the rank of the snapshot matrix is also w. The POD basis vectors (modes) can be selected as the first r left singular vectors of the matrix U (where $r \leq w$). If r is chosen to be strictly less than w, an approximation error can be quantified using the singular values (σ),

$$E = \frac{\sum_{k=r+1}^{n} \sigma_k^2}{\sum_{k=1}^{n} \sigma_k^2} \tag{2.4}$$

2.2.2. SPARSE GRIDS

The snapshots for the POD method can be generated by different methods depending on the sampling scheme. However, computing the function $\boldsymbol{f}$ at every possible combination is unrealistic, especially for high dimensional problems. Therefore, in this chapter, the sampling points are generated on a sparse grid. The idea is to select a set of nodes for each dimension in the parameter space. Then, the points are tensorized in a specific way to construct the sparse grid. Many choices are possible for the unidimensional nodes. The only requirement is to choose the nodes in a nested manner (i.e., $X_i \subset X_{i+1}$), where X_i is the set of nodes for a given index i. An overview of different possible sparse grid choices can be found in [14].

In this work, the sparse grid is combined with the POD method in order to build a nonintrusive model. This imposes an additional constraint on the selection of the unidimensional nodes. This is because the nodes need to be separated enough in the parameter space to produce enriched POD modes covering the complete range of dynamics of the system. Nevertheless, such selection of nodes might not be the ideal scheme for the interpolation. In many studies, Chebyshev nodes were found to perform better than uniform sampling [14]. However, Chebyshev nodes produce more points very close to each other at the boundary and fewer points in the central region. This increases the risk of overlooking some of the dynamics at the inner region. Therefore, in order to achieve
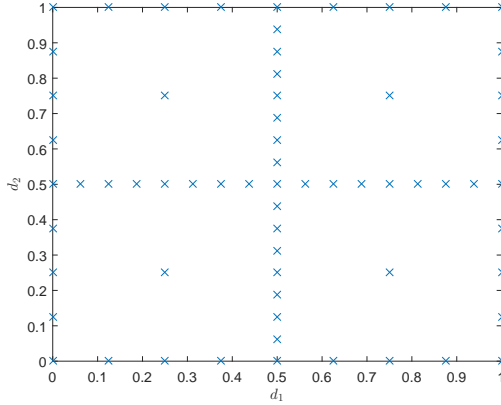
Figure 2.1: Sparse grid points for $d = 2$ and $l = 4$.

maximum separation of points over the entire parameter domain, equidistant nodes are chosen to generate the sampling points with the following formula:

$$m_i = \begin{cases} 1 & \text{if } i = 1 \\ 2^{i-1} + 1 & \text{if } i > 1 \end{cases}$$

$$x_j^i = \begin{cases} \dfrac{j-1}{m_i - 1} & \text{for } j = 1, 2, ..., m_i \quad \text{if } m_i > 1 \\ 0.5 & \text{for } j = 1 \qquad\qquad \text{if } m_i = 1 \end{cases} \tag{2.5}$$

Where i is the index for any dimension d and x_j^i are the nodes in set X_i. The points are generated in the hypercube $[0\ 1]^d$ which can then be scaled accordingly.

The Smolyak algorithm can be applied to combine the unidimensional nodes into a sparse grid by satisfying the following condition:

$$q - d + 1 \le |i| \le q \tag{2.6}$$

Where d is the dimension, and $|i| = i_1 + i_2 + ... + i_d$ with i_d being an integer index in dimension d. q is a parameter such that $q \ge d$. The level (l) of the sparse grid can be defined by $l = q - d$. Figure 2.1 shows the sparse grid for $d = 2$ and $l = 4$. Table 2.1 presents the number of points for the first 7 levels for different dimensions.

The generated points can then be used to construct the snapshot matrix for the POD. One can select the level to have the maximum affordable number of snapshots. However, we propose an adaptive level selection by computing the SVD for each level successively and setting a criterion based on the decay of the singular values. The concept is based on the fact that the singular values are representative of the energy of each POD mode [13]. Thus, higher singular values indicate POD modes that are contributing more and are considered more important than modes with lower values. Consequently, adding points to the snapshot which result only in lower singular values is analogous to including higher order terms in a Taylor expansion.

Therefore, after a sufficient number of points, the main dynamics is captured and any added point will only change the lower singular values. Thus, benefiting from the hierarchical nature of the sparse grids, we suggest comparing the slope of decay for the leading singular values with each level increase. Then, the appropriate level is selected when no change in the slopes is observed. However, the "leading" singular values still need to be defined properly. In the proposed algorithm, they are defined with respect to the highest singular value, that is the first singular values within a margin of the highest singular value (σ_i / σ_1). The margin in this work was taken as 10^{-4}.

Thus, the following algorithm is proposed to select the minimum sparse grid level needed to build a reduced model:

Algorithm
Starting with $l = 1$,

1. generate the sparse points for level l;

2. sample the model at the generated points;

3. construct the snapshot matrix;

4. perform the SVD on the snapshots;

5. compute the logarithmic decay slope for the leading singular values;

6. compare the computed decay slope with that of level $l - 1$. If the absolute difference is more than a given tolerance, increase l and repeat step 1; otherwise, return U and D.

The algorithm returns the POD modes in the left singular matrix U and the singular values in the diagonal of D. Equation 2.4 can be used to choose the number of POD modes (r) appropriately. The expansion coefficients (c_i) can be computed from Equation 2.3 for all parameter points within the training set. Then, in order to compute the solution for a new parameter value, one needs to interpolate between the obtained coefficients. However, interpolation for high dimensional functions is challenging. Therefore, Smolyak's combination technique is used to tensorize unidimensional interpolation functions. Due

Table 2.1: Number of sparse grid points by level and dimension.

Level (l)	$d = 2$	$d = 3$	$d = 4$	$d = 5$	$d = 6$
0	1	1	1	1	1
1	5	7	9	11	13
2	13	25	41	61	85
3	29	69	137	241	389
4	65	177	401	801	1457
5	145	441	1105	2433	4865
6	321	1073	2929	6993	15121

to the selection of equally spaced nodes, local piecewise multilinear functions are chosen as basis functions. The piecewise linear functions are defined as follows [14]:

$$a_{x_1} = 1 \quad \text{if } i = 1$$

$$a_{x_i^j}(x) = \begin{cases} 1 - (m_i - 1) \cdot |x - x_j^i| & \text{if } |x - x_j^i| < \dfrac{1}{m_i - 1} \\ 0 & \text{otherwise} \end{cases} \quad (2.7)$$

Where $a(x)$ is the local basis function and m_i is defined as in Eq. 2.5. The interpolant for one dimension can then be computed as follows:

$$U^i(c) = \sum_{x_i \in X_i} a_{x^i}(x) c(x) \quad (2.8)$$

Then, as shown in [10], the Smolyak combination technique can be used to construct the multidimensional interpolant:

$$A_{q,d} = \sum_{q-d+1 \leq |i| \leq q} (-1)^{q-|i|} \binom{d-1}{q-|i|} (U^{i_1}(c) \otimes \dots \otimes U^{i_d}(c)) \quad (2.9)$$

2.2.3. RADIAL BASIS FUNCTION
A different method to compute the expansion coefficients is using Radial Basis Functions (RBF). RBF assumes a surrogate of the form:

$$c(\lambda) = \sum_{i=1}^{p} \alpha_i g_i(\|\lambda - \lambda_i\|) \quad (2.10)$$

where p is the number of sample points, α_i are coefficients to be determined. In principle, the kernel function $g_i(\|\lambda - \lambda_i\|$ can be any function of the norm between the required λ and the sampled λ_j. In this study, the multi-quadratic kernel was selected,

$$g_i(\|\lambda - \lambda_i\|) = \sqrt{\|\lambda - \lambda_i\|^2 + \gamma^2} \quad (2.11)$$

where γ is a shape parameter to be tuned. The coefficients α can be found by replacing the obtained values of c_i in Equation 2.10 and solving the resultant system of linear equations. Once α_i are obtained, which is done only once in the offline phase, values for c_i can be computed in the online phase for any new parameter λ. It is important to highlight that the selection of the shape parameter has an effect on the accuracy of the interpolation [15]. The shape parameter in this work was selected by dividing the data into a training set and a testing set. The parameter was then optimized by cross validation of the two sets to minimize the error.

2.3. RESULTS AND ANALYSIS
Two cases were chosen to test the proposed algorithm. In both cases, the two-group diffusion eigenvalue equation is solved. Assumptions of no up scattering and no fast
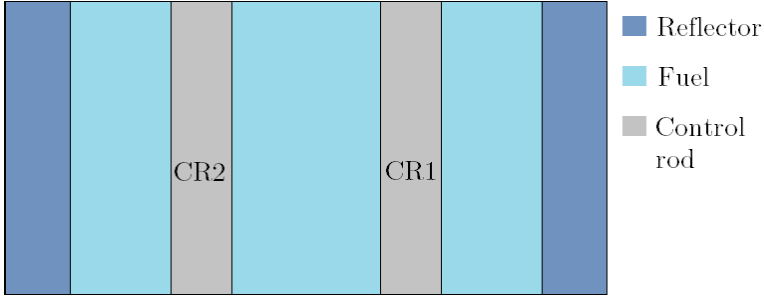
Figure 2.2: Geometry of the slab in Case1.

fissions were made for the equation,

$$-\nabla \cdot D_1 \nabla \phi_1 + (\Sigma_{a1} + \Sigma_{12})\, \phi_1 = \frac{1}{k}\, \nu \Sigma_{f2}\, \phi_2$$
$$-\nabla \cdot D_2 \nabla \phi_2 + \Sigma_{a2}\phi_2 = \Sigma_{12}\phi_1$$

(2.12)

where D_g is the group diffusion coefficient, Σ_{ag} is the group absorption cross section, Σ_{12} is the down scattering cross section, k is the multiplication factor, ϕ_1 and ϕ_2 are the fast and thermal flux respectively.

In each test case, Equation 2.12 is first solved with a well-established numerical method (Finite Difference in the first case and Finite Element in the second). Then, the reduced models (RBF and Smolyak) were built and assessed with respect to that reference solution. All computations were performed in a Matlab environment. It is important to note that in both cases the reference solution itself was fast enough to be solved inexpensively at any desired point. Thus, building a reduced order model was not required in those cases. Nevertheless, the test cases were selected for illustrative purposes only.

2.3.1. TEST CASE 1
The first case is an eigenvalue problem solved with a Finite Difference scheme. Equation 2.12 is solved for one spatial direction. Figure 2.2 shows the slab geometry, which has a total thickness of 396 cm. The flux at the boundaries is assumed to be zero. The fuel region is reflected at both ends with a reflector of thickness 15.4 cm at each side. Two control rods are introduced in the fuel region, each with equal thickness of 15.4 cm. The domain was discretized into a total of 780 mesh points. The percentage insertion of each control rod is considered as an input parameter for the model. Thus, the model is parametrized on a 2D space. For this test case, only the thermal flux is considered as an output for the model. Nevertheless, the same algorithm can be applied to the fast flux.

The algorithm was applied to the model and resulted in a sparse grid level selection of 5 (145 points). The resulting singular values are shown in Figure 2.3a. For comparison purposes, the singular values of the previous level ($l = 4$) are also plotted in the same figure. A close-up view of the first singular values can be seen in Figure 2.3b. Indeed, by examining the singular values, it is evident that most of the points added in level 5

contributed only to the lower part of the plot. In fact, the algorithm revealed that the maximum absolute slope change between the two levels in the leading singular values was 0.03.

To build the reduced models, Equation 2.3 was first used to compute the expansion coefficients. Then, two nonintrusive models were built with the obtained POD modes and coefficients. The first was RBF with Equation 2.10 and the second was Smolyak's interpolant as in Equation 2.9. Both models were tested with 121 points that were not part of the training set. The maximum error in L_2 norm was found to be 17% for the RBF model and 9% for the Smolyak model. This case was observed when control rod 1 was inserted 30% and control rod 2 inserted 5%. Figure 2.4a shows the flux for this case. In Figure 2.4b, a different selected point is shown where also the Smolyak model had outperformed RBF. The reference model had a runtime of around 10 s. On the other hand, both RBF and Smolyak models achieved a considerable saving in computational time. Smolyak model needed 0.1 s for a single simulation. The RBF model was even faster than Smolyak by a factor of 10. The offline time to assemble both models was around 1451 s.

(a) Singular values for all modes.

(b) A close-up view of the first 21 singular values.
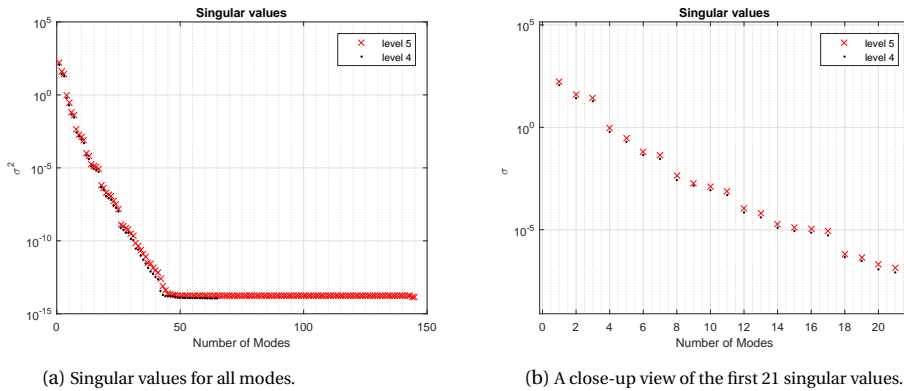
Figure 2.3: Test Case 1: Singular values for $l = 5$ (145points) compared with the previous level of $l = 4$ (65points).

2.3.2. TEST CASE 2
The second tested case is the 2D IAEA benchmark problem as described in Ref. [16]. The core has two fuel zones with five control rods and a reflector. The geometry of the core can be seen in Figure 2.5. The steady state, 2-group diffusion equation (Equation 2.12) is solved in two spatial dimensions. The cross sections of the different regions are reported in Table 2.2. The benchmark problem also provides the axial buckling $B_{z,g}^2 = 0.8 \times 10^{-4}$. Thus, a term $D_g B_{z,g}^2$ is added to the removal term of Equation 2.12. The boundary conditions are assumed to be vacuum at the external boundary ($J_g^{\text{in}} = 0$) and symmetry at the inner boundaries ($\nabla \phi_g = 0$). This problem was solved with a Finite Element Method (FEM) on an unstructured mesh employed in a Matlab environment. The FEM model is considered as the reference model for the problem. The insertion percentage of the control rods were considered as input to the model. Thus, a nonintrusive reduced order model of the FEM is built with 5 input parameters corresponding to each control rod

(a) CR1 inserted 30% and CR2 inserted 5% (RBF error = 17% and Smolyak = 9%).

(b) CR1 inserted 10% and CR2 inserted 60% (error of RBF = 5%, and Smolyak = 0.1%).
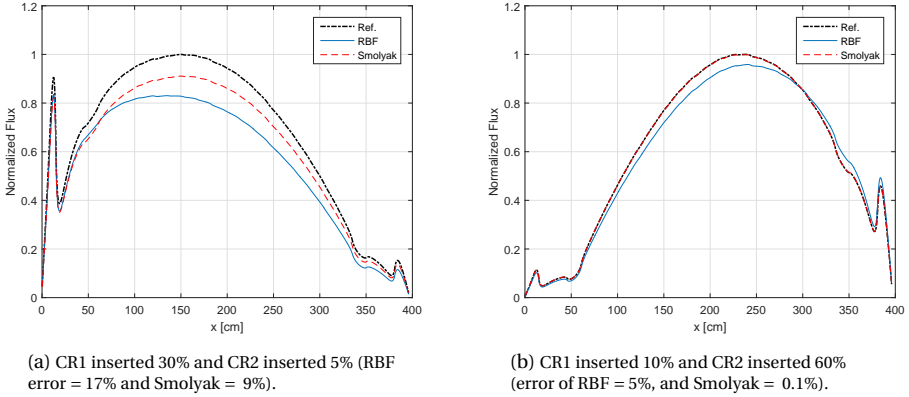
Figure 2.4: Test Case 1: Normalized thermal flux.

position. The model output was considered to be the thermal flux.
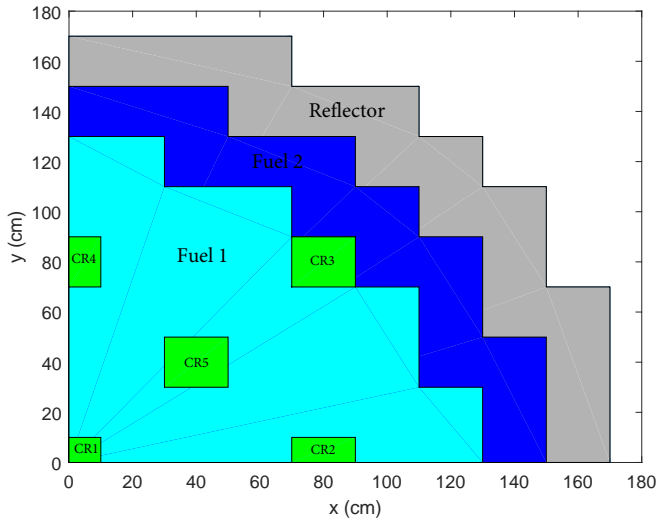
Figure 2.5: Geometry of the 2D IAEA benchmark problem as implemented for Test Case 2.

The algorithm selected sparse grid level 4, which resulted in 801 simulation points. The leading singular values are plotted in Figure 2.6 along with the singular values of the previous level. The figure shows the similarity of the decay between level 4 and level 3. This indicates that most of the new level points had little contribution to the dominant singular values. The resultant POD modes were truncated to $r = 50$ in order to build the reduced RBF and Smolyak models. The expansion coefficients were obtained by applying Equation 2.3. The models were then tested with 1500 points that were not part of the training set. The points were generated on a uniform full grid spanning the parameter

Table 2.2: Cross Section data for the different regions in Test Case 2.

	Fuel 1	Fuel 2	Control Rod	Reflector
D_1 [cm]	1.5	1.5	1.5	2
D_2 [cm]	0.4	0.4	0.4	0.3
Σ_{12} [cm^{-1}]	0.02	0.02	0.02	0.04
Σ_{a1} [cm^{-1}]	0.01	0.01	0.01	0
Σ_{a2} [cm^{-1}]	0.08	0.085	0.13	0.01
$\nu\Sigma_{f2}$ [cm^{-1}]	0.135	0.135	0.135	0

domain. The error for both models is compared in Figure 2.7. The reference FEM model computed a single simulation in about 3 s. RBF model needed 2404 s to assemble the model (offline time) and 0.005 s for a single simulation (online time). On the other hand, Smolyak model needed almost the same time for the offline phase and 0.3 s for the online phase. The results show that while RBF was faster in running a single simulation, Smolyak model outperformed RBF by a considerable margin. All error analyses were assessed in the L_2 norm. The average error for the RBF was 1.03% whereas the average Smolyak's error was found to be 0.08%. The maximum observed RBF error was 2.7%. The same configuration resulted in a Smolyak error of 0.04%. This case is shown in Figure 2.8. The flux is given along the x-axis and along the diagonal line ($y = x$). On the other hand, Figure 2.9 shows the configuration that resulted in a maximum Smolyak error, which was found to be 0.3%. This configuration resulted in an RBF error of 2.7%.
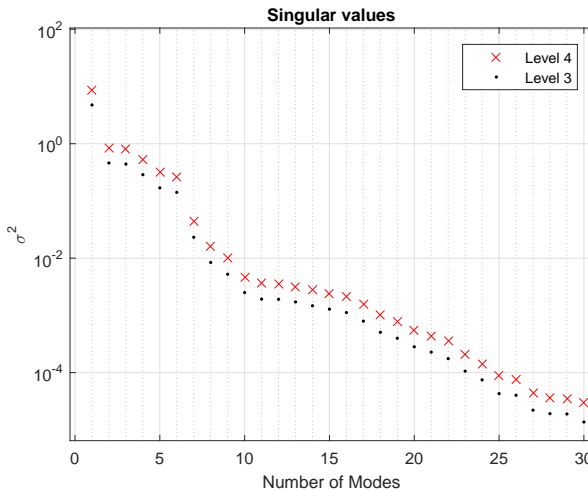
Figure 2.6: Test Case 2: Leading singular values plotted in comparison with the singular values of the previous level.
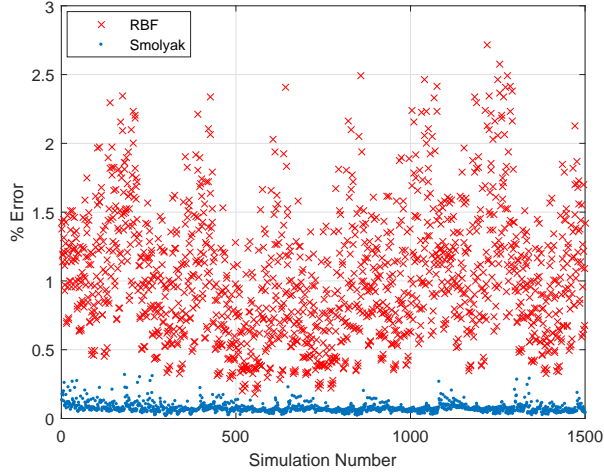
Figure 2.7: Test Case 2: error comparison between RBF and Smolyak models for 1500 simulation points.

2.4. CONCLUSIONS

Building a reduced order POD model for high dimensional problems can be achieved efficiently with the use of sparse grids. The suggested algorithm of comparing the singular values of the different spars grid levels provides a valuable tool for the efficient selection of the sampling points. Thus, the offline time can be reduced significantly in problems parametrized on high dimensional spaces. Moreover, combining sparse grid sampling with Smolyak's combination technique results in an effective nonintrusive reduced order model. In this work, Smolyak's interpolant was tested with equally spaced nodes and piecewise linear interpolation. Also, a comparison with RBF was presented for two nuclear problems. In both cases, the results showed that while RBF resulted in a faster reduced order model, Smolyak's model provided a superior accuracy. Nevertheless, different RBF kernels were not studied and the shape parameter was selected by manual optimization. As suggested in Ref. [15], the accuracy of the RBF can be improved with advanced optimization.

Although the sparse grid approach generated an efficient set of sampling points compared to the full grid tensorization, the number of points still depends on the dimension d. As is evident from Table 2.1, the number of points increases sharply with the increase in d and l. Therefore, the algorithm may result in an infeasible number of simulations for really high dimensions. Similarly, models with strong nonlinearities will need a higher level of sparse grids. In these cases, adaptive sparse grids can be employed to increase the sampling points only in regions of higher interest. Therefore, in future work, we will investigate the use of adaptive sparse grids in building reduced order models.
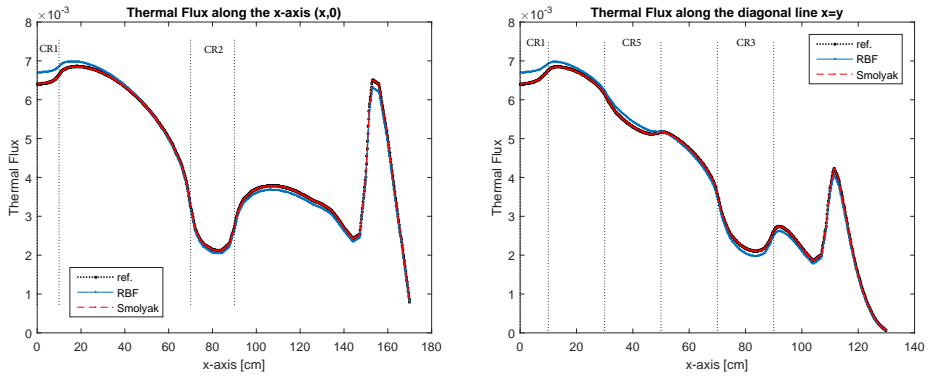
Figure 2.8: Test Case 2: Thermal flux for the configuration that resulted in a maximum RBF error (control rods insertion percentages were [$CR1 = 10\%$, $CR2 = 90\%$, $CR3 = 58\%$, $CR4 = 90\%$, $CR5 = 10\%$]). RBF error was 2.7% and Smolyak's error was 0.04%.
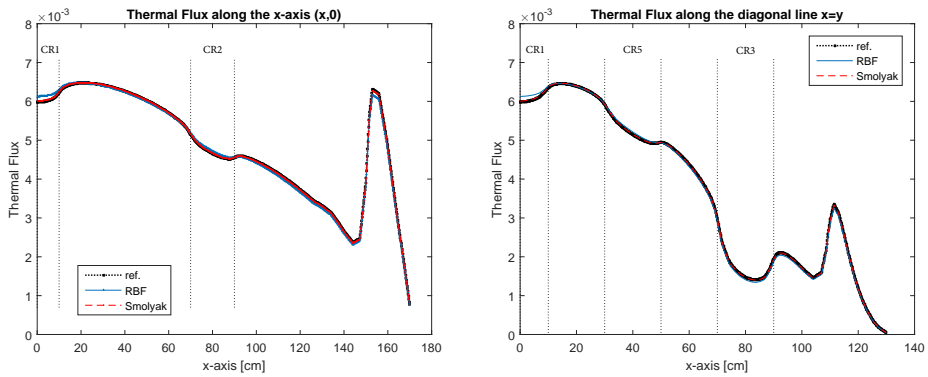
Figure 2.9: Test Case 2: Thermal flux for the configuration that resulted in a maximum Smolyak model error (control rods insertion percentages were [$CR1 = 10\%$, $CR2 = 10\%$, $CR3 = 90\%$, $CR4 = 10\%$, $CR5 = 10\%$]). RBF error was 1.8% and Smolyak's error was 0.3%.

REFERENCES

[1] F. Alsayyari, D. Lathouwers, and J. L. Kloosterman, *A Nonintrusive POD Approach for High Dimensional Problems using Sparse Grids,* in *International Conference on Mathematics and Computational Methods applied to Nuclear Science and Engineering (M&C 2017)* (Jeju, South Korea, 2017).

[2] P. Benner, S. Gugercin, and K. Willcox, *A Survey of Projection-Based Model Reduction Methods for Parametric Dynamical Systems,* SIAM review **57**, 483 (2015).

[3] J. L. Lumley, *The Structure of Inhomogeneous Turbulent Flows,* Atmospheric Turbulence and Radio Wave Propagation **790**, 166 (1967).

[4] A. G. Buchan, C. C. Pain, F. Fang, and I. M. Navon, *A POD reduced-order model for eigenvalue problems with application to reactor physics,* International Journal for Numerical Methods in Engineering **95**, 1011 (2013).

[5] 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 Rreactor Spatial Kinetics,* Annals of Nuclear Energy **71**, 217 (2014).

[6] H. V. Ly and H. T. Tran, *Modeling and Control of Physical Processes using Proper Orthogonal Decomposition,* Mathematical and Computer Modelling **33**, 223 (2001).

[7] V. Buljak, *Inverse Analyses with Model Reduction: Proper Orthogonal Decomposition in Structural Mechanics* (Springer, Berlin, 2012).

[8] M. Guénot, I. Lepot, C. Sainvitu, J. Goblet, and R. Filomeno Coelho, *Adaptive Sampling Strategies for Non-intrusive POD-based Surrogates,* Engineering Computations **30**, 521 (2013).

[9] S. A. Smolyak, *Quadrature and interpolation formulas for tensor products of certain classes of functions,* in *Dokl. Akad. Nauk SSSR*, Vol. 4 (1963) pp. 240–243.

[10] V. Barthelmann, E. Novak, and K. Ritter, *High Dimensional Polynomial Interpolation on Sparse Grids,* Advances in Computational Mathematics **12**, 273 (2000).

[11] B. Peherstorfer, *Model Order reduction of Parametrized Systems with Sparse Grid Learning Techniques,* Ph.D. thesis, Technische Universität München, München (2013).

[12] D. Xiao, F. Fang, A. Buchan, C. Pain, I. Navon, and A. Muggeridge, *Non-intrusive Reduced Order Modelling of the Navier–Stokes Equations,* Computer Methods in Applied Mechanics and Engineering **293**, 522 (2015).

[13] P. Holmes, J. L. Lumley, and G. Berkooz, *Turbulence, Coherent Sructures, Dynamical Systems and Symmetry* (Cambridge University Press, Cambridge, 1996).

[14] A. Klimke, *Uncertainty Modeling using Fuzzy Arithmetic and Sparse Grids,* Ph.D. thesis, Universität Stuttgart, Stuttgart (2006).

[15] P. Chandrashekarappa and R. Duvigneau, *Radial Basis Functions and Kriging Meta-models for Aerodynamic Optimization*, Research Report RR–6151 (INRIA, 2007).

[16] Argonne National Laboratory, *Argonne Code Center: Benchmark Problem Book*, ANL-7416 (Suppl. 2, 1977).

2

3

LOCALLY ADAPTIVE SPARSE GRIDS FOR PARAMETRIZED SYSTEMS

In this chapter, we study integrating the locally adaptive sparse grids with the POD method to develop a novel nonintrusive POD-based reduced order model. In our proposed approach, the locally adaptive sparse grid is used to adaptively control the sampling scheme for the POD snapshots, and the hierarchical interpolant is used as a surrogate model for the POD coefficients. An approach to efficiently update the surpluses of the sparse grids with each POD snapshots update is also introduced. The robustness and efficiency of the locally adaptive algorithm are increased by introducing a greediness parameter, and a strategy to validate the reduced model after convergence. The proposed validation algorithm can also enrich the reduced model around regions of detected discrepancies. Three numerical test cases are presented to demonstrate the potential of the proposed adaptive-POD algorithm. The first is a nuclear reactor point kinetics, the second is a general diffusion problem, and the last is a variation of the analytical Morris function. The results show that the developed algorithm reduced the number of model evaluations compared to the classical sparse grid approach. The built reduced models captured the dynamics of the reference systems with the desired tolerances. The non-intrusiveness and simplicity of the method provide great potential for a wide range of practical large scale applications.

3.1. INTRODUCTION

COMPLEX systems are described by interactions of multi-physics phenomena that occur at various scales. Capturing such inter-disciplinary interactions requires building complex coupled models. Nuclear reactors, for example, are described by interactions of radiation transport, heat transfer, and fluid dynamics. High fidelity simulation tools are often used to provide a solution for such coupled problems. Nevertheless, even with the increasing power of today's supercomputers, high fidelity models require a tremendous amount of computational time and memory allocation. For applications such as design optimization, uncertainty quantification, and control, where many repeated model evaluations are needed, such models are too expensive.

Reduced Order Modeling (ROM) is an effective technique to reduce the dimensionality of large-scale complex systems. It replaces the high fidelity model with a low-dimensional efficient model preserving only the prominent dynamics of the system. The reduced model can then be used to provide fast solutions with a controlled level of accuracy. Different reduced order modeling techniques have been proposed in literature. Benner et al. (2015) [2] presented a comprehensive survey on ROM with numerous examples on their applications. Proper Orthogonal Decomposition (POD) is the favoured method for nonlinear systems [3]. POD was first introduced as a statistical technique to extract dominant characteristics from a set of data. As a reduced order method, the method was later developed by Lumley (1967) [4] to model coherent structures in turbulent flows.

The POD method is based on sampling the high fidelity model at several points in parameter space to construct a so-called snapshot matrix. Then, a reduced basis is created through Singular Value Decomposition (SVD). The original high fidelity model can then be projected onto the created reduced basis space by means of a Galerkin projection. The snapshot matrix and the model are built during the offline phase, which is executed only once. Afterwards, during the online phase, the reduced model can be run inexpensively at any desired parameter point. POD-Galerkin has been studied extensively in many fields, such as in fluid dynamics to model compressible flows [5–7] and incompressible flows [8–10]. In reactor physics applications, POD was applied to the eigenvalue problem of the diffusion equation [11], the time-dependent diffusion equation [12], and to model the coolant pool of the Lead-cooled Fast Reactor [13]. A variant of this approach called the Reduced Basis method [14–16] was developed by adding an a posteriori error estimator and a greedy sampling scheme to model parametrized partial differential equations.

Projection-based POD methods are code intrusive, which is a major limitation. For legacy codes where access to the governing equations is unattainable, this approach cannot be applied. In such cases, a slightly different nonintrusive POD technique can be employed. The idea is to benefit from the orthogonality of the subspace basis to generate the Galerkin expansion coefficients at the sampled points. Then, a surrogate model can be constructed to compute the solution at any required non-sampled point. Different surrogate models have been suggested to compute the expansion coefficients. For lower dimensional problems, direct interpolation or splines can be used [17]. Problems with higher dimensionality require more advanced techniques. The use of Radial Basis Function (RBF) is one of the common methods for such problems [3, 18]. Hesthaven and Ubbiali (2018) [19] used neural networks to build a surrogate model for the coefficients. Gaussian regression can also be used to build the surrogate model [20].

For the ROM to produce an accurate representation of the original model, the snapshots need to capture the entire dynamics of the original model within the desired range. Consequently, the choice of the sampling scheme directly affects the accuracy of the POD method. Moreover, in the nonintrusive approach, the sampling points should be dense enough for the surrogate model to reproduce a reliable solution at non-sampled locations. Thus, the sampling strategy becomes even more relevant for nonintrusive methods. In addition, problems parametrized on high dimensional spaces are prone to the curse of dimensionality, that is the exponential increase of the computational time with the increase in the number of dimensions. In these cases, the efficient selection of the sampling points is crucial for any practical application. Latin Hypercube Sampling (LHS) can be an efficient way to address this issue [21]. However, the lack of adaptivity is a limitation in nonlinear cases. Guenot et al. (2013) [22] proposed an extension of LHS that improves the initial snapshot matrix by adaptively selecting new points based on the "influence" of the new point on the snapshot matrix.

A different sampling method based on sparse grids can also be applied. Sparse grids were first introduced by Smolyak (1963) [23] and, ever since, have been used to cope with high dimensional multivariate integration and interpolation problems. The method builds a hierarchical grid that preserves the properties for the unidimensional rule by a specific combination of the tensorized product [24]. In the context of reduced order models, Peherstorfer (2013) [25] suggested the use of sparse grids as a machine learning tool for reduced order models, which was tested on heat transfer problems. In addition, Xiao et al. (2015) [26] presented a method of propagating the expansion coefficients through time with the use of a sparse grid interpolant, which was demonstrated on the Navier-Stokes equations. Elman et al. (2011) [27] suggested the use of Reduced Basis method to further reduce the computational burden of stochastic collocation methods based on sparse grids.

Sparse grids were developed under the assumption that the function to be approximated is sufficiently smooth. In this case, the algorithm provides optimal selection of subspaces contributing to the interpolant function. However, in many applications, the smoothness of the function is unknown and cannot be established a priori. To that end, adaptive strategies can be employed to modify the classical sparse grids algorithm. In cases where the multivariate function is only sensitive to certain dimensions, the anisotropic sparse grid approach is suitable. In this strategy, which is also called (global) adaptive sparse grids, the grid is constructed by placing more points along certain dimensions that have higher importance. The importance of each dimension is identified during the construction stage by testing and comparing all dimensions [28]. Chen and Quarteroni (2015) [29] combined the anisotropic adaptive sparse grid with the reduced basis method for error estimation.

However, the dimension adaptive approach falls short of identifying regions with steep gradients or discontinuities. For these cases, an alternative local adaptive strategy can be more effective. In fact, one of the earliest work on sparse grids by Zenger (1990) [30] suggested the use of local adaptivity for non-smooth functions. The objective of this strategy is to identify certain regions of higher importance and only refine the grid within these regions by benefiting from the hierarchical structure of the grids [31]. Griebel (1998) [32] showed how locally adaptive sparse grids can be used to adaptively discretize partial

3

differential equation. In this implementation, Dirichlet boundary condition is assumed, and the unidimensional rule is chosen to not place any points along the boundaries. Pflueger (2010) [33] extended this idea by modifying the basis functions in a way that extrapolates towards the boundaries. The author used this algorithm for classification problems. This approach is suitable if the value at the boundaries is not important and only an estimate is required. Ma and Zabaras (2009) [34] used a unidimensional rule that places points at the boundaries for an adaptive collocation method. The authors then used the algorithm with an Anchored-ANOVA approach to model stochastic processes [35]. Applications of the locally adaptive sparse grids can also be seen in tracking function discontinuities in high dimensional spaces [36], high dimensional integrations [37], and economic modeling [38].

In this chapter, our goal is to exploit the hierarchical nature of the adaptive sparse grids in order to efficiently build a POD-based reduced order model in a nonintrusive manner. We present an approach to utilize the local adaptivity in order to identify regions of high importance for the POD development. In our nonintrusive approach, no assumption is made on the value of the model at the boundaries of the parameter domain. Therefore, we follow the work by Ma and Zabaras (2009) [34] in defining the unidimensional rule for the adaptive sparse grids. We also introduce an approach to iteratively update the surpluses of the sparse grids as the POD modes develop. We suggest a criterion for the refinement strategy based on the physical space rather than the surpluses of the sparse grids. We also extend the locally adaptive algorithm by introducing a parameter that controls the greediness of the algorithm in generating the snapshots. Additionally, a strategy to validate and update the reduced model is proposed, which increases the robustness of the algorithm. Although the nonintrusive approach is considered in this chapter, the proposed algorithm can equally be combined with a Galerkin-POD approach.

The remainder of this chapter is organized as follows: in Section 3.2 the problem is formulated and the POD method is introduced. Section 3.3 presents the sparse grids as an interpolation technique by first introducing the classical sparse grids and subsequently the locally adaptive version. In this section, the refinement strategy and the proposed validation algorithm are also presented. The combined adaptive-POD algorithm is presented in Section 3.4 along with the method of updating the surpluses. Three applications are presented in Section 3.5 that test the proposed algorithm numerically. The first is a neutron point kinetics problem in 5 dimensions presented in two cases: one is strongly nonlinear and the second case is weakly nonlinear. The second application is a general diffusion problem in 18 dimensions, and the last application is an analytical function of 20 dimensions. Finally, our conclusions are discussed in Section 3.6.

3.2. Proper Orthogonal Decomposition

Physical phenomena are modelled by capturing the dynamics of the system in a set of governing equations. These equations can then be solved numerically by some discretization technique. The solution of the discretized model results in the state (or field) vector describing the state of the system, which in turn is a function of some design parameters that characterise features of the system (geometry, materials,...,etc.). The discretized model can be written in the form

$$\mathbf{R}(\mathbf{y}(\mathbf{x}), \mathbf{x}) = \mathbf{0}, \tag{3.1}$$

where $\mathbf{y} \in \mathbb{R}^n$ is a vector with n state variables, and $\mathbf{x} \in \mathbb{R}^d$ is a vector of the design parameters with dimension d. For high fidelity models, the dimension of the state vector (n) is usually large, which renders solving the model to be computationally expensive. ROM aims at recasting the high fidelity model into a simpler model with dimension $r < n$. The model can then be solved with reduced computational effort. The POD method approximates the state vector in terms of basis vectors (in a discrete analogy to Fourier expansion) as

$$\mathbf{y}(\mathbf{x}) \approx \sum_{h=1}^{r} c_h(\mathbf{x})\mathbf{u}_h, \tag{3.2}$$

where c_h is the amplitude of the basis vector $\mathbf{u}_h$. The POD basis vectors (also called POD modes) are data-driven, that is they are built based on data collected from the model to describe the state vector. The amplitude c_h is a function of the design parameter $\mathbf{x}$.

The POD method is based on sampling the model at different design parameter values. Each state solution is a snapshot of the model at a certain parameter value. The snapshots are collected in the snapshot matrix

$$\mathbf{M} = [\mathbf{y}(\mathbf{x}_1), \mathbf{y}(\mathbf{x}_2), \dots, \mathbf{y}(\mathbf{x}_p)] \quad \in \mathbb{R}^{n \times p}, \tag{3.3}$$

where p is the number of sampling points. The goal of the POD method is to find the optimal basis vectors in some subspace $\mathbb{V}$ of dimension $r << n$ that minimizes the error of the approximation in the L_2 norm. Once the basis vectors are known, the amplitude $c_h(\mathbf{x})$ can be computed either intrusively, by projecting the governing equations, or non-intrusively using regression methods. Our approach is non-intrusive, and therefore the model can be considered as a black box, mapping a given input to the desired output. We can write a functional minimizing the approximation error in the L_2 norm as follows [39]:

$$E = \min_{\mathbf{u}_h} \sum_{j=1}^{p} \left\| \mathbf{y}(\mathbf{x}_j) - \sum_{h=1}^{r} c_h(\mathbf{x}_j)\mathbf{u}_h \right\|_{L2}. \tag{3.4}$$

The basis vectors are chosen such that they are orthonormal (i.e., $< \mathbf{u}_g, \mathbf{u}_h > = \delta_{gh}$). The POD basis solves the minimization problem in Equation 3.4. They can be obtained with the Singular Value Decomposition (SVD) as the left singular vectors of the snapshot matrix [3]. Using orthogonality of the modes, the value of the amplitude $c_h(\mathbf{x})$ at parameter value $\mathbf{x}_j$ can be computed as

$$c_h(\mathbf{x}_j) = < \mathbf{u}_h, \mathbf{y}(\mathbf{x}_j) >, \tag{3.5}$$

If the number of non-zero singular values is g, it can be shown that the rank of the snapshot matrix is also g. The POD basis vectors are formed by the first r left singular vectors (where $r \le g$). If r is chosen to be strictly less than g, a POD truncation error can be quantified using the singular values (σ) as follows:

$$e_r = \frac{\sum_{k=r+1}^{n} \sigma_k^2}{\sum_{k=1}^{n} \sigma_k^2}. \tag{3.6}$$

To set a criterion for selecting the number of POD modes (r), a cut-off threshold (γ_{tr}) can be defined such that

$$e_r < \gamma_{\text{tr}} \qquad \forall r \in [1, \dots n]. \tag{3.7}$$

Note that the truncation error only quantifies the error in representing the state solutions included in the snapshot matrix. However, with sufficient sampling points, it can be used as an indicator for the error in representing (new) solutions not included in the snapshot matrix.

3.3. Sparse Grids for Interpolation

In order to generate the snapshot matrix, we need to explore the parameter space by sampling the model at discrete points. Choosing an appropriate number of sampling points is a key challenge for any sampling strategy. Covering the entire range of dynamics of the unknown function is imperative for a successful construction of the ROM. In addition, extending the sampling strategy to high dimensional problems is another challenge that must be addressed. Different sampling schemes have been studied to determine an optimal set of sampling points for increased space coverage (e.g., LHS). However, such methods select the sampling points a priori without any insight into the function being sampled. This can lead to overlooking some localized nonlinearities or discontinuities. To reduce such risk, uniform sampling with small intervals can be used. However, this strategy is prohibitively expensive.

Sparse grids can be very effective for problems of high dimensionality. The construction of the sparse grids is a hierarchical approach that successively builds the new grid based on previous grid selection. Such construction is suitable for adaptive strategies since it can be used to build an algorithm that will terminate whenever a desired accuracy is reached. We will first introduce the classical sparse grid method then present our approach for implementing adaptivity.

3.3.1. Classical Sparse Grids

Sparse grids are constructed based on selecting a set of nodes separately for each dimension. These nodes are generated in a hierarchical manner by levels, where each level is assigned an integer index i. The unidimensional nodes are then tensorized to form the final sparse grids. We first consider one dimension, then generalize it to the multidimensional case.

Many choices are possible for selecting the unidimensional nodes. While the nodes can be disjoint (non-nested) as in [27] and [40], nested nodes are more convenient and efficient since function evaluations in this case are not repeated with increasing the sparse grid level. Therefore, we choose the nodes in a nested manner, that is $\mathcal{X}^i \subseteq \mathcal{X}^{i+1}$, where $\mathcal{X}^i$ is the set of nodes at the index level i. Since nodes are nested, we can also define the difference set as $\mathcal{X}_\Delta^{i+1} = \mathcal{X}^{i+1} \backslash \mathcal{X}^i$, where $\mathcal{X}_\Delta^{i+1}$ is a set that contains only the newly added nodes at level $i + 1$. An overview of different possible sparse grid choices can be found in [41]. We generate the nodes in the range [0, 1] which is then scaled according to the physical range of the parameters in $\mathcal{X}^i$. In order to avoid confusion, we reserve the use of the term "node" for the unidimensional point while a multidimensional vector of coordinates formed by nodes along each dimension is given the term "point".

Ultimately, we aim to combine sparse grids with the POD method in order to build a nonintrusive ROM model. This imposes an additional constraint on the selection of the unidimensional nodes. This is because the nodes need to be separated sufficiently in parameter space to produce enriched POD modes covering the complete range of dynamics of the model. Nevertheless, such selection of nodes might not be ideal for interpolation. In many studies, Chebyshev nodes were found to perform better than uniform sampling [41]. However, Chebyshev nodes produce more nodes very close to each other at the boundaries and fewer nodes in the central region, increasing the risk of overlooking some dynamics in the inner region. Therefore, in order to achieve maximum separation of nodes over the entire parameter domain, we choose the equidistant rule to generate the nodes.

A Smolyak interpolant is built for the amplitudes $c(x)$ (from Equation 3.5, where the index h is dropped for notational convenience) by considering an operator $U^i(c)(x)$ that approximates the function $c(x)$ by an expansion as follows:

$$c(x) \approx U^i(c)(x) = \sum_{x_j^i \in \mathcal{X}^i} c(x_j^i) a_{x_j^i}^i(x), \tag{3.8}$$

where i is the level index, $\mathcal{X}^i$ is the set of nodes x_j^i at level i, $a_{x_j^i}^i(x)$ are basis functions, and $c(x_j^i)$ is the function value evaluated at the support nodes x_j^i. In principle, different choices for the basis functions are possible. However, due to our selection of equidistant nodes, any choice of (global) polynomial basis function is likely to yield a poor approximation because of Runge's phenomenon [24]. Additionally, polynomial functions are not suitable for local adaptive strategies since their support covers the entire domain. Piecewise multi-linear functions, on the other hand, are flexible because they have local support and thus can be used to refine specific regions of the domain. These basis functions, which are also called the hat functions because of their shape, satisfy $a_{x_j^i}^i(x) \in C([0,1])$, $a_{x_j^i}^i(x_j^i) = 1$, $a_{x_j^i}^i(y_j^i) = 0 \ \forall \ y_j^i \in \mathcal{X}^i$, $x_j^i \neq y_j^i$. For equidistant type nodes, we can define the basis functions as follows [41]:

$$a_{x^1}^1 = 1 \quad \text{if } i = 1,$$

$$a_{x_j^i}^i(x) = \begin{cases} 1 - (m^i - 1) \cdot |x - x_j^i|, & \text{if } |x - x_j^i| < \dfrac{1}{m^i - 1} \\ 0, & \text{otherwise,} \end{cases} \tag{3.9}$$

where m_i and the equidistant nodes x_j^i are defined as follows:

$$m^i = \begin{cases} 1 & \text{if } i = 1, \\ 2^{i-1} + 1 & \text{if } i > 1, \end{cases} \tag{3.10}$$

$$x_j^i = \begin{cases} 0.5 & \text{for } j = 1 & \text{if } m^i = 1, \\ \dfrac{j-1}{m^i - 1} & \text{for } j = 1, 2, ..., m^i & \text{if } m^i > 1. \end{cases} \tag{3.11}$$

m^i represents the number of nodes at level i (cardinality of $\mathcal{X}^i$).

Before generalizing to the multivariate case, we first define the difference formula

$$\Delta^i(c)(x) = U^i(c)(x) - U^{i-1}(c)(x), \tag{3.12}$$

with $U^0 = 0$. As a consequence of selecting nested nodes, the interpolant at level i can always recreate the interpolant at level $i-1$ (i.e., $U^{i-1}(c)(x) = U^i(U^{i-1}(c)(x))$). Therefore, Equation 3.12 can be rewritten in terms of the basis functions in Equation 3.8 as,

$$\Delta^i(c)(x) = \sum_{x_j^i \in \mathscr{X}^i} a^i_{x_j^i}(x) c(x_j^i) - \sum_{x_j^i \in \mathscr{X}^i} a^i_{x_j^i}(x)(U^{i-1}(c)(x_j^i)), \tag{3.13}$$

$$= \sum_{x_j^i \in \mathscr{X}^i} a^i_{x_j^i}(x)(c(x_j^i) - U^{i-1}(c)(x_j^i)). \tag{3.14}$$

Since the interpolant is completely represented at level $i-1$, $(c(x_j^i) - U^{i-1}(c)(x_j^i)) = 0, \forall x_j^i \in \mathscr{X}^{i-1}$. Thus,

$$\Delta^i(c)(x) = \sum_{x_j^i \in \mathscr{X}_\Delta^i} a^i_{x_j^i}(x)(c(x_j^i) - U^{i-1}(c)(x_j^i)). \tag{3.15}$$

This means that the interpolant needs to be evaluated only at the newly added nodes at each level increase. Thus, we can redefine x_j^i as the jth element in the set $\mathscr{X}_\Delta^i$. Knowing that the number of newly added nodes (cardinality of $\mathscr{X}_\Delta^i$) is $m_\Delta^i = m^i - m^{i-1}$, the difference formula can be written as,

$$\Delta^i(c)(x) = \sum_{j=1}^{m_\Delta^i} a^i_{x_j^i}(x)(c(x_j^i) - U^{i-1}(c)(x_j^i)). \tag{3.16}$$

Hence, the sum only runs over the newly added elements that are stored in $\mathscr{X}_\Delta^i$. The contributions of the previous level need not to be considered.

The Smolyak algorithm can be applied to combine the unidimensional nodes into sparse grids by satisfying the following condition:

$$d \le |\mathbf{i}| \le l + d, \tag{3.17}$$

where d is the number of dimensions, $|\mathbf{i}| = i_1 + i_2 + ... + i_d$ with i_n being the index level along dimension n, and l defines the level of the sparse grids. Therefore, the points of the sparse grid at level l are formed by the set

$$B_{l,d} = \bigcup_{d \le |\mathbf{i}| \le l+d} (\mathscr{X}^{i_1} \otimes \mathscr{X}^{i_2} \otimes ... \otimes \mathscr{X}^{i_d}) \tag{3.18}$$

Figure 3.1 shows the development of the sparse grids from level $l = 0$ to $l = 4$ for a two dimensional space ($d = 2$). Table 3.1 lists the number of sparse grid points generated per level and dimension.

The unidimensional formulation in Equation 3.8 can be extended to the multivariate case using the tensor product operation

$$(U^{i_1}(c)(x_1) \otimes \cdots \otimes U^{i_d}(c)(x_d)) = \sum_{j_1=1}^{m^{i_1}} \cdots \sum_{j_d=1}^{m^{i_d}} c(x_{j_1}^{i_1}, \ldots, x_{j_d}^{i_d}) \cdot (a^{i_1}_{x_{j_1}^{i_1}}(x_1) \otimes \cdots \otimes a^{i_d}_{x_{j_d}^{i_d}}(x_d)). \tag{3.19}$$
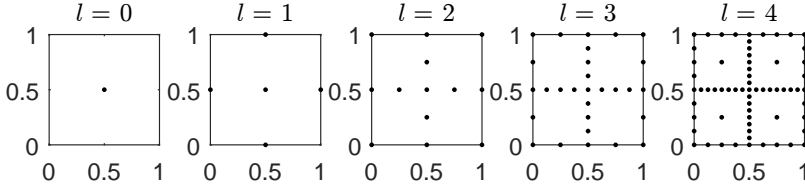
Figure 3.1: Sparse grid points for $d = 2$ and different values of l.

Table 3.1: Number of sparse grid points by level and dimension.

Level (l)	$d = 2$	$d = 3$	$d = 4$	$d = 5$	$d = 6$
0	1	1	1	1	1
1	5	7	9	11	13
2	13	25	41	61	85
3	29	69	137	241	389
4	65	177	401	801	1457
5	145	441	1105	2433	4865
6	321	1073	2929	6993	15121

This shows that building the interpolant needs $\prod_{k=1}^{d} m^{i_k}$ function evaluations, which increases exponentially with the dimension. Smolyak combination technique can be used to reduce the number of function evaluations. The idea is based on the fact that not all points contribute equally to the interpolant, some have a minimal contribution which can be neglected. Therefore, a hierarchically structured algorithm can be built that includes points iteratively until a desired accuracy is reached.

The Smolyak combination technique forms the multivariate interpolant from the univariate difference formula (Equation 3.16) as follows:

$$A_{l,d}(c)(\mathbf{x}) = \sum_{|\mathbf{i}| \leq l+d} \Delta^{i_1}(c)(x_1) \otimes \cdots \otimes \Delta^{i_d}(c)(x_d), \tag{3.20}$$

where l and $|\mathbf{i}|$ are defined as in Equation 3.17. This Equation can be split into two parts,

$$A_{l,d}(c)(\mathbf{x}) = \underbrace{\sum_{|i| < l+d} (\Delta^{i_1}(c)(x_1) \otimes \cdots \otimes \Delta^{i_d}(c)(x_d))}_{A_{l-1,d}(c)(\mathbf{x})} + \underbrace{\sum_{|i| = l+d} (\Delta^{i_1}(c)(x_1) \otimes \cdots \otimes \Delta^{i_d}(c)(x_d))}_{\Delta A_{l,d}(c)(\mathbf{x})} .$$

$$\tag{3.21}$$

The first term ($A_{l-1,d}(c)(x)$) represents the interpolant value at the previous interpolation level and the second term ($\Delta A_{l,d}(c)(x)$) is the interpolation contributions from the newly added points. Since the points at each level are defined as a subset of the next level

$(\mathcal{X}^i \subseteq \mathcal{X}^{i+1})$, the second term can be written in terms of the basis functions as

$$\Delta A_{l,d}(c)(\mathbf{x}) = \sum_{|\mathbf{i}|=l+d} \sum_{\mathbf{j}} (a^{i_1}_{x^{i_1}_{j_1}}(x_1) \otimes \cdots \otimes a^{i_d}_{x^{i_d}_{j_d}}(x_d)) \cdot \underbrace{(c(x^{i_1}_{j_1},\ldots,x^{i_d}_{j_d}) - A_{l-1,d}(c)(x^{i_1}_{j_1},\ldots,x^{i_d}_{j_d}))}_{w^{\mathbf{i}}_{\mathbf{j}}},$$

(3.22)

where $\mathbf{j}$ is a multi-index $(j_1,\ldots,j_d)$, $j_k = 1,\ldots,m^{i_k}_\Delta$, $k = 1,\ldots,d$, and $m^{i_k}_\Delta$ is the number of newly added nodes along dimension k. Note that any point $\mathbf{x}^{\mathbf{i}}_{\mathbf{j}} = (x^{i_1}_{j_1},\ldots,x^{i_d}_{j_d})$ at level l is included in the set $B_{l,d}$ (from Equation 3.18). The term denoted by $w^{\mathbf{i}}_{\mathbf{j}}$ is called *hierarchical surplus* [41] which is the difference between the true function values at the newly added points and the corresponding approximation of the interpolant at the previous level. Therefore, these coefficients are simply a correction of the interpolant at level $l-1$ to the actual values. Figure 3.2 illustrates the progress of the interpolant for a simple one-dimensional function.
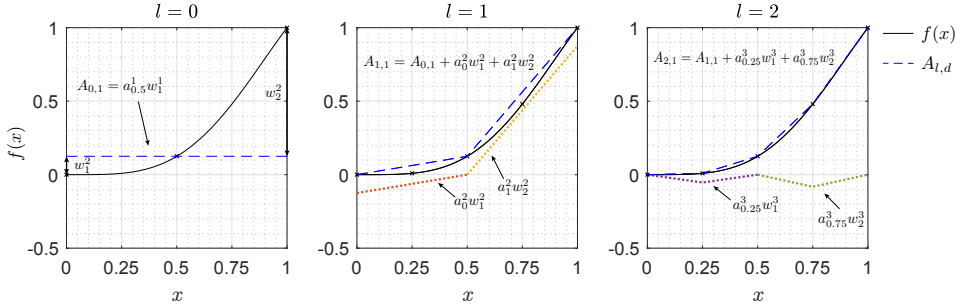
Figure 3.2: The progression of the interpolant $A_{l,d}$ for a one dimensional function $f(x) = x^2 \sin\left(\frac{\pi}{2}x\right)^2$ from the first level $l = 0$ to level $l = 2$. The weights w^i_j and the basis functions $a^i_{x_j}$ are also shown. The generated nodes are $\mathcal{X}^1_\Delta = \{0.5\}$ at level $l = 0$, $\mathcal{X}^2_\Delta = \{0,1\}$ at level $l = 1$, and $\mathcal{X}^3_\Delta = \{0.25, 0.75\}$ at level $l = 2$.

3.3.2. LOCALLY ADAPTIVE SPARSE GRIDS

The local adaptive method can be illustrated by showing the unidimensional nodes $\mathcal{X}^i_\Delta$ in a tree-like structure. Figure 3.3 shows such a tree where the depth of the tree has been assigned the level index i. The root of the tree has a single node $\mathcal{X}^1 = \{0.5\}$. It is evident that nodes are added at each level to half the distances between the nodes in the previous levels. Therefore, each node has an ancestry as shown in the tree structure (Figure 3.3). Each node has one parent and two children, except at index level $i = 2$ where each node has only one child. This ancestry dependence can be extended to multidimensional points by relating each point to a set of neighbouring points called forward points. Specifically, a forward point to $\mathbf{x}$ is a point on the grid that shares all nodes with $\mathbf{x}$ except along one dimension where the forward point node is a child of the node of $\mathbf{x}$. To that end, we define a forward neighbourhood operator $\Psi(\mathscr{S})$ that operates on a set
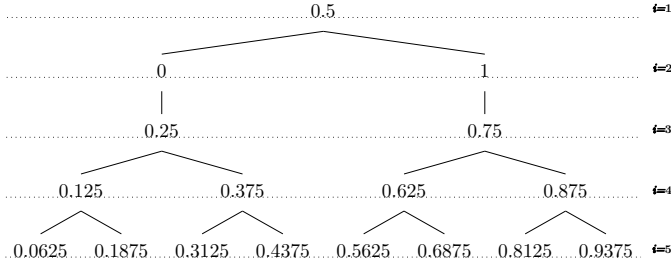
Figure 3.3: Tree structure for the nodes in $\mathscr{X}_\Delta^i$ where the depth is assigned the level index i. Nodes are added at each level to half the distances between the nodes in the previous level.

of points $\mathscr{S} = \{\mathbf{x}_q | q = 1,\dots, n\}$ and returns all forward points for all points in $\mathscr{S}$ as follows:

$$\Psi(\mathscr{S}) = \{(y_1,\dots,y_d)| \exists i, q : b(y_i) = x_{q,i} \wedge y_j = x_{q,j} \; \forall j \neq i, \; q \in [1,\dots,n], \; j, i \in [1,\dots d]\}, \tag{3.23}$$

where $b(x)$ is a function that returns the parent of a node x from the tree. We also define a backward point for $\mathbf{x}$ as a point with a parent node along one of the dimensions of $\mathbf{x}$. A backward neighbourhood operator $\Psi^{-1}(\mathscr{S})$ that operates on the set $\mathscr{S}$ and returns the set of all backward points can be defined as

$$\Psi^{-1}(\mathscr{S}) = \{(y_1,\dots,y_d)| \exists i, q : b(x_{q,i}) = y_i \wedge y_j = x_{q,j} \; \forall j \neq i, \; q \in [1,\dots,n], \; j, i \in [1,\dots d]\}. \tag{3.24}$$

Each point on the grid is surrounded by $2d$ forward points and d backward points. However, because of the exception at tree level 2 where nodes generate only one child, points that contain a node from level 2 have less than $2d$ forward points. Additionally, points that contain the root node 0.5 have less than d backward points because the parent function $b(x)$ returns the root itself for the root node (i.e., $b(0.5) = 0.5$). Note that the forward points are not unique since the same forward point can be generated from different points. Thus, it is important to keep track of the generated forward points to avoid duplication of points. By applying the backward neighbourhood operator successively, we can define the set of ancestor points $\Gamma(\mathscr{S})$ for all points $\mathbf{x}$ in $\mathscr{S}$

$$\Gamma(\mathscr{S}) = \bigcup_{q=1}^{L} \left(\Psi^{-1}\right)^q (\mathscr{S}), \tag{3.25}$$

where $\left(\Psi^{-1}\right)^L (\mathscr{S}) = (0.5,\dots,0.5)$. The set of ancestors for a point $\mathbf{x}$ represents all points with basis functions that contribute to the construction of the interpolant at point $\mathbf{x}$. Figure 3.4 shows an example point $\mathbf{x} = (0.25, 0.75)$ with its forward points. The figure also shows a point on the boundary containing a node from level 2 $\mathbf{x} = (0.75, 0)$, which has less than $2d$ forward points.

Locally adaptive algorithm
The basic idea of the locally adaptive algorithm is to set a criterion for selecting important points then refining the grid iteratively by adding only the forward points of the selected
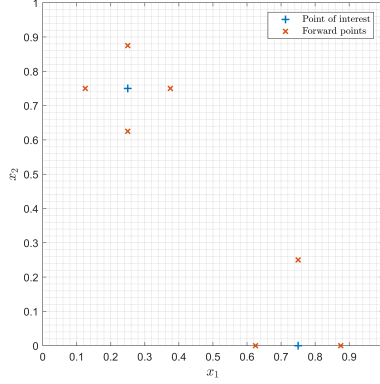
Figure 3.4: A 2-dimensional examples showing the four forward points of the points $\mathbf{x} = (0.25, 0.75)$ and the three forward points of the point $\mathbf{x} = (0.75, 0)$.

important points.

Let $\mathcal{Z}^{k-1}$ be the set of important points, and I^{k-1} be a set of inactive points that were considered unimportant at iteration $k-1$. During the next iteration (k), the algorithm generates a testing set $\mathcal{T}^k$ from the forward points of $\mathcal{Z}^{k-1}$, and I^{k-1} as

$$\mathcal{T}^k = \Psi(\mathcal{Z}^{k-1}) \cup I^{k-1}, \tag{3.26}$$

and identifies a subset $\mathcal{Z}^k \subseteq \mathcal{T}^k$ that is considered important, which is then added to the grid $\mathcal{X}^k$ (i.e., $\mathcal{X}^k = \mathcal{Z}^k \cup \mathcal{X}^{k-1}$). This process is repeated until some global criterion is met.

In order to identify the important points $\mathcal{Z}^k$, we need to define a local error measure (ϵ_j^k). A point $\mathbf{x}_j$ in the testing set $\mathcal{T}^k$ is considered important and is admitted in $\mathcal{Z}^k$ if it has an error (ϵ_j^k) above a certain threshold (γ_{int})

$$\mathcal{Z}^k = \{\mathbf{x}_j \in \mathcal{T}^k | \epsilon_j^k > \gamma_{\text{int}}\}. \tag{3.27}$$

Points that do not meet this criterion are stored in the inactive set

$$\mathcal{I}^k = \{\{\mathcal{T}^k \backslash \mathcal{Z}^k\} \cup \mathcal{I}^{k-1}\}. \tag{3.28}$$

At each iteration, we need to evaluate the interpolant at the testing points $\mathcal{T}^k$ in order to compute ϵ_j^k. However, the interpolant in Equation 3.21 is written in terms of the (global) level l which is not relevant in the adaptive scheme because points are added based on their location and ancestry. Therefore, we can rewrite the interpolant in Equation 3.21 and Equation 3.22 for any point $\mathbf{x} = (x_1, \dots, x_d)$ in terms of iteration k as

$$A_{k,d}(c)(\mathbf{x}) = A_{k-1,d}(c)(\mathbf{x}) + \Delta A_{k,d}(c)(\mathbf{x}), \tag{3.29}$$

$$\Delta A_{k,d}(c)(\mathbf{x}) = \sum_{n=1}^{m_k^\Delta} w_n^k \Theta_n(\mathbf{x}), \tag{3.30}$$

where $m_k^{\Delta} = card(Z^k)$, and Θ_n is the d-dimensional basis function for the point $\mathbf{x}_n \in \mathcal{Z}_k$,

$$\Theta_n(\mathbf{x}) = \prod_{p=1}^{d} a_{x_{n,p}^{i_p}}^{i_p}(x_p), \tag{3.31}$$

where $\mathbf{x}_n$ has support nodes $(x_{n,1}^{i_1}, \dots, x_{n,d}^{i_d})$, and i_p is the level (tree depth) index for the support node $x_{n,p}^{i_p}$. The surplus corresponding to $\mathbf{x}_n$ is defined as

$$w_n^k = c(\mathbf{x}_n) - A_{k-1,d}(\mathbf{x}_n). \tag{3.32}$$

Once the important points are identified, they are added to the set $\mathcal{X}^k$ and their corresponding surpluses are stored in the set $\mathcal{W}^k$. The hierarchical surplus as defined in Equation 3.32 is a natural candidate for the local error measure ϵ_j^k. These surpluses are defined locally (for each point) and represent the deviation from the true value. This criterion was applied in [34, 36]. However, Griebel (1998) [32] showed that taking the absolute value of the surpluses is too sharp of an indicator and can lead to a non-terminating algorithm in some cases. The author suggested weighing the surpluses with the integral of the corresponding basis functions in order to give more importance to points with basis functions that have wider support. This criterion was used in [33, 35, 37]. In our implementation, we combine the adaptivity with the POD to model a physical field. Therefore, we choose a local error measure based on physical space rather than the surpluses which are defined in parameter space. The local error measure in our approach is presented in Section 3.4 after presenting the method of integrating the POD with the adaptive sparse grids.

To highlight the difference between the classical and the locally adaptive sparse grids, Figure 3.5 shows an analytical function that was sampled using both approaches. The first 65 sampled points from both algorithms are marked on the figure. The classical sparse grids algorithm resulted in a uniform sampling regardless of the function's behaviour. The adaptive algorithm, on the other hand, was more efficient by spending more points around the steep gradient and fewer points in the smooth region.

INCLUDING ANCESTOR POINTS IN THE ADAPTIVITY

A refinement criterion based entirely on the local error measure can lead to premature termination of the algorithm. This is observed when the true function value intersects (or closely intersects) the interpolant at the forward points. For example, the adaptive algorithm based only on a local error measure never converged when tested on the Rosenbrock function in 2D, defined as $f(x, y) = 100(y - x^2)^2 + (1 - x)^2$ and $x, y \in (0, 1)$. The reason is that the value of this function at the root point $(0.5, 0.5)$ and at one of its forward points $(0.5, 0)$ are equal. Therefore, the surplus computed at the point $(0.5, 0)$ is zero which falsely implies that the interpolant is accurate around this point. As a consequence, the algorithm stops refining around the point $(0.5, 0)$ and assumes a constant interpolation between $(0.5, 0.5)$ and $(0.5, 0)$, which is incorrect. The missing behaviour around $(0.5, 0)$ cannot be recovered at subsequent iterations. This is significant because the basis function at $(0.5, 0)$ has support that spans half the domain.

This issue can be mitigated by redefining the selection of the important points to include the ancestors. Including missing ancestors has been discussed as part of building

3
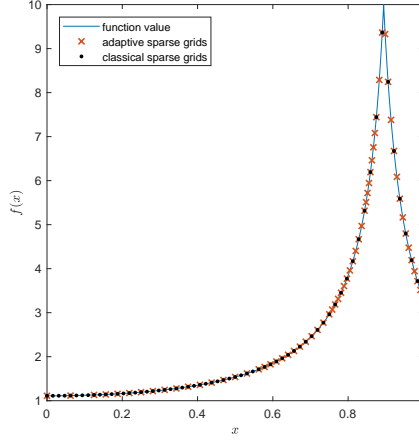
Figure 3.5: A comparison showing the difference in sampling the function $f(x) = \frac{1}{|0.8 - x^2| + 0.1}$ between adaptive and classical sparse grids algorithms. The first 65 sampled points from both algorithms are marked, which shows the adaptive algorithm selecting more points around the peak and fewer at the smooth region.

an accurate hierarchical interpolant (see [33]). Bungartz et. al (2003) [37] used the missing ancestors as support points for the d-polynomial basis functions where they contribute to the calculations of the surpluses of their descendants. In our approach, the goal is to use the adaptivity to explore the parameter space. Therefore, the ancestors are not only included for building the interpolant but also to search in the forward points of these ancestors for any possible missing behaviour. In this case, the local error indicator ϵ_j^k is still a measure for the importance of the region around the point but the algorithm prioritizes the search in the vicinity of ancestors before moving to the forward points. This is important because the basis functions for ancestors have wider support compared to the descendants. However, in order to keep the number of evaluations reduced, not all ancestors are included in the important set.

The important set is redefined to take into account the ancestors as follows:

$$\mathscr{C}^k = \{\mathbf{x}_j \in \mathscr{T}^k \,|\, \epsilon_j^k > \gamma_{\text{int}}\},$$
$$\mathcal{I}_a^k = \{\mathbf{x}_j \in \mathscr{C}^k \,|\, \Gamma(\mathbf{x}_j) \subseteq \mathscr{X}^{k-1}\},$$
$$\mathcal{I}_b^k = \{\mathbf{y}_i \in \Gamma(\mathbf{x}_j) \,|\, \mathbf{x}_j \in \mathscr{C}^k, \, \Gamma(\mathbf{x}_j) \cap \mathscr{C}^k = \emptyset \,\wedge\, \mathbf{y}_i \notin \mathscr{X}^{k-1} \wedge \Gamma(\mathbf{y}_i) \subseteq \mathscr{X}^{k-1}\},$$
$$\mathcal{I}^k = \mathcal{I}_a^k \cup \mathcal{I}_b^k. \tag{3.33}$$

In this definition, we first identify a set of candidate points $\mathscr{C}^k$ containing all points with an error above the defined threshold γ_{int}. Then, the important points set $\mathcal{I}^k$ is formed by two parts. The first $\mathcal{I}_a^k$ are points within the candidate points $\mathscr{C}^k$ that have all their ancestors already included in the sparse grid $\mathscr{X}^{k-1}$ from previous iterations. The second $\mathcal{I}_b^k$ are the missing ancestors of candidate points with partial ancestry in $\mathscr{X}^{k-1}$. However, candidate points that have any ancestor point also as a candidate will not be considered because the error at these points is likely to be high due to the error at that ancestor.

Such points will be added to the inactive set to be tested again in the next iteration after including the ancestor point first. Applying this strategy to the aforementioned Rosenbrock function resulted in the algorithm converging with 967 points to a relative error of 1%.

Such definition for the set of important points (Equation 3.33) enhances the quality of exploring the parameter space because for every important point identified in iteration k, all $2d$ forward points will be tested in the next iteration (Equation 3.26). However, for high dimensional problems where the model is linear (or almost linear) along certain dimensions, refining the grid in all dimensions unnecessarily increases the number of model evaluations. For such cases, we can control the number of model evaluations by introducing a parameter μ that tunes the greediness of the sampling scheme. This parameter reduces the number of points in the testing set $\mathcal{T}^{k+1}$ as follows:

$$\mathcal{T}^{k+1} = \left\{ \mathbf{x}_j \in \Psi(\mathcal{Z}^k) \,\middle|\, \frac{card(\Psi^{-1}(\mathbf{x}_j) \cap \mathcal{X}^k)}{card(\Psi^{-1}(\mathbf{x}_j))} \geq 1 - \mu \right\} \cup \mathcal{I}^k, \qquad (3.34)$$

where the operator $card(\mathcal{Y})$ returns the cardinality of a set $\mathcal{Y}$, and μ is the greediness parameter that has a value $\in [0,1]$. For every forward point in $\Psi(\mathcal{Z}^k)$, the fraction of its backward points that are included in the important set $\mathcal{X}^k$ is required to be greater than or equal to $1 - \mu$ in order for this point to be tested. Note that each point in a d-dimensional grid has up to d backward points. The algorithm is greedy for $\mu = 1$ because all forward points will be admitted and tested. For $\mu = 0$, a point will only be tested if all of its backward points were important, which directs the algorithm to avoid searching regions with no important points. The concept of only considering points whose backward points are important is inspired by the (anisotropic) dimension adaptive sparse grids, where indices of important grids are identified based on the importance of the indices of its backward neighbours (from previous iterations) [28]. By extending this concept to the local adaptivity, we have better control of the number of model evaluations, at the expense of exploring the parameter space less thoroughly.

VALIDATION

Including the ancestors can only mitigate the premature termination issue but not resolve it completely. This can be observed, for example, in the one-dimensional sine function $\sin(2\pi x)$, which has the same value at the root point $\{0.5\}$ and at its forward points $\{0,1\}$. Both forward points, in this case, will have zero surpluses ($w_1^2 = w_2^2 = 0$). Therefore, the algorithm will still terminate without any further refinement even with including the ancestors rule. This issue can also arise in multidimensional functions. We can try to define a different error criterion to circumvent this case. However, as stated by Griebel [32], for any given error criteria, we can always find a function that will cause the algorithm to terminate prematurely. Therefore, we propose to include a validation step after convergence that will test the model at randomly generated points. If any of the random points results in an error greater than the tolerance, the algorithm enriches the model with more points around that point. This is achieved by considering all points from the inactive set with basis functions contributing to the interpolant at the random

validation points

$$\mathcal{Q} = \{\mathbf{x}_n \in \mathcal{I}^k \mid |x_{r,p} - x_{n,p}^{i_p}| < \frac{1}{m_n^{i_p} - 1}, \forall i_p \neq 1, p = 1, \dots, d\}, \tag{3.35}$$

where $\mathbf{x}_r = (x_{r,1}, \dots, x_{r,d})$ is the tested random point, $x_{n,p}^{i_p}$ is the support node of the point $\mathbf{x}_n$ along dimension p with tree depth i_p, and $m_n^{i_p}$ is defined as in Equation 3.10. The points in $\mathcal{Q}$ are then considered candidate points (i.e., $\mathcal{C}^k = \mathcal{Q}$) and the adaptive algorithm is resumed. The basic principle here is that if the random point has an error above the tolerance, it implies that the algorithm incorrectly categorized one of the contributing basis functions as not important (added to the inactive set) and failed to build an accurate interpolant in that region. Therefore, reconsidering these points as candidate points will enrich the model around this region.

3.4. ADAPTIVE-POD ALGORITHM

The adaptive algorithm is used to guide the sampling scheme for the POD method. At each iteration, the high fidelity model $\mathbf{y}(\mathbf{x})$ is sampled at new grid points $\mathcal{T}^k$. Then based on the predefined error measure, an important subset $\mathcal{Z}^k$ is identified and added to $\mathcal{X}^k$. The snapshots corresponding to the points in $\mathcal{Z}^k$ are then added to the snapshots set $\mathcal{F}_k = \{\mathbf{y}(\mathbf{x}_j) | \mathbf{x}_j \in \mathcal{X}^k, j = 1, \dots m_k\}$. We then perform a SVD on the snapshots and obtain new POD modes $\{\mathbf{u}_h | h = 1, \dots r_k\}$. Each iteration will result in a new set of POD modes $\mathbf{u}_h$ and, consequently, a new set of amplitudes $c_h(\mathbf{x})$. Moreover, the number of POD modes might increase from iteration to the next because of the addition of new snapshots. As a consequence, the number of functions to be interpolated (amplitudes of the POD modes) also increases. In this section, we propose a scheme that is able to keep track of the changes in the amplitudes with minimized computational cost.

At iteration k, the high fidelity model is approximated as

$$\mathbf{y}(\mathbf{x}) \approx \sum_{h=1}^{r_k} c_h(\mathbf{x})\mathbf{u}_h. \tag{3.36}$$

Using the orthogonality of the POD modes, we can define the amplitudes at any point $\mathbf{x}_i$ as

$$c_h(\mathbf{x}_i) = <\mathbf{y}(\mathbf{x}_i), \mathbf{u}_h>. \tag{3.37}$$

We aim to approximate $c_h(\mathbf{x})$ with the interpolant $A_{k,d}(c_h)(\mathbf{x})$ (Equation 3.30) and eventually build a ROM $\mathbf{y}_a(\mathbf{x})$ approximating $\mathbf{y}(\mathbf{x})$ such that

$$\mathbf{y}_a(\mathbf{x}) = \sum_{h=1}^{r_k} A_{k,d}(c_h)(\mathbf{x})\mathbf{u}_h. \tag{3.38}$$

The interpolant $A_{k,d}(c_h)(\mathbf{x})$ depends on the grid points $\mathcal{X}^k$ and the surpluses $\mathcal{W}_h^k$. For every amplitude (c_h), a specific interpolant is built with a corresponding set of surpluses $\mathcal{W}_h^k$.

At iteration $k+1$, the adaptive algorithm selects a new set of grid points $\mathcal{Z}^{k+1}$. Then, the set of snapshots are updated, $\mathcal{F}^{k+1} = \{\mathbf{y}(\mathbf{x}_j) | \mathbf{x}_j \in \mathcal{X}^{k+1}, j = 1, \dots m_{k+1}\}$, where $\mathcal{X}^k \subset \mathcal{X}^{k+1}$ and $m_{k+1} > m_k$. As a consequence, we obtain a new set of POD modes $\{\hat{\mathbf{u}}_h | h = 1, \dots r_{k+1}\}$ and corresponding amplitudes $\{\hat{c}_h(\mathbf{x}) | h = 1, \dots r_{k+1}\}$. The POD model at iteration $k+1$ can be written as

$$\mathbf{y}_a(\mathbf{x}) = \sum_{g=1}^{r_{k+1}} \hat{c}_h(\mathbf{x})\hat{\mathbf{u}}_h. \tag{3.39}$$

In principle, $\hat{c}_h(\mathbf{x})$ is a new function of $\mathbf{x}$ and is not related to $c_h(\mathbf{x})$ because the POD modes are different (i.e., $\mathbf{u}_h \neq \hat{\mathbf{u}}_h$). Therefore, in order to construct the ROM as in Equation 3.38, a new interpolant for $\hat{c}_h$ ($A_{k+1,d}(\hat{c}_h)(\mathbf{x})$) needs to be rebuilt hierarchically starting from the first iteration. From Equation 3.29, $A_{k+1,d}(\hat{c}_h)(\mathbf{x})$ is formed by two parts

$$A_{k+1,d}(\hat{c}_h)(\mathbf{x}) = A_{k,d}(\hat{c}_h)(\mathbf{x}) + \Delta A_{k+1,d}(\hat{c}_h)(\mathbf{x}). \tag{3.40}$$

The first term $A_{k,d}(\hat{c}_h)(\mathbf{x})$ is the interpolant from the previous iterations, which needs to be computed first for all $\mathbf{x}_j \in \mathcal{X}^k$ in order to obtain the unknown surpluses $\hat{\mathcal{W}}_h^k$. Recomputing the entire interpolant for all previous points at each iteration is inefficient and counter-productive to the hierarchical structure of the Smolyak algorithm. However, since the grid points are selected in a nested manner, we can find an efficient way to update the surpluses from the previous iterations without having to recompute the interpolant hierarchically. Once these surpluses are updated, the surpluses for the new points in $\mathcal{Z}^{k+1}$ can be computed for the second term $\Delta A_{k+1,d}(\hat{c}_h)(\mathbf{x})$ as in Equation 3.32.

To update the surpluses $\hat{\mathcal{W}}_h^k$, we first notice that the amplitudes from the two consecutive iterations k and $k+1$ are not equal, that is

$$\hat{c}_h(\mathbf{x}) \neq c_h(\mathbf{x}). \tag{3.41}$$

In physical space, however, assuming negligible SVD truncation error, both POD models from Equation 3.36 and Equation 3.39 are defined to reproduce the exact snapshots at the points in $\mathcal{X}^k$ because the points are nested ($\mathcal{X}^k \subset \mathcal{X}^{k+1}$). Therefore,

$$\sum_{g=1}^{r_{k+1}} \hat{c}_g(\mathbf{x}_j)\hat{\mathbf{u}}_g = \sum_{h=1}^{r_k} c_h(\mathbf{x}_j)\mathbf{u}_h \qquad \forall \mathbf{x}_j \in \mathcal{X}^k. \tag{3.42}$$

We can use the orthogonality property and project the equation onto $\hat{\mathbf{u}}_g$ to obtain the amplitudes

$$\hat{c}_g(\mathbf{x}_j) = \sum_{h=1}^{r_k} c_h(\mathbf{x}_j) <\mathbf{u}_h, \hat{\mathbf{u}}_g> \qquad \forall \mathbf{x}_j \in \mathcal{X}^k, g = 1, \dots, r_{k+1}. \tag{3.43}$$

Since the interpolant $A_{k,d}(\hat{c}_g)(\mathbf{x})$ is a function of the surpluses rather than the function $\hat{c}_g$, it is more convenient to find a relation between $\hat{w}_j^k$ and w_j^k. We first note that the set $\mathcal{X}^k$ is formed by the union of the important points from all previous iterations, that is

$$\mathcal{X}^k = \bigcup_{l=1}^{k} \mathcal{Z}^l. \tag{3.44}$$

The definition of the surpluses in Equation 3.32 can be written for amplitude $c_h(\mathbf{x}_j)$ at any iteration l as

$$w_{j,h}^l = c_h(\mathbf{x}_j) - A_{l-1,d}(c_h)(\mathbf{x}_j), \qquad \forall \mathbf{x}_j \in \mathcal{Z}^l. \tag{3.45}$$

Substituting Equation 3.45 in Equation 3.43

$$\hat{w}_{j,g}^l + A_{l-1,d}(\hat{c}_g)(\mathbf{x}_j) = \sum_{h=1}^{r_k} \left[w_{j,h}^l + A_{l-1,d}(c_h)(\mathbf{x}_j) \right] < \mathbf{u}_h, \hat{\mathbf{u}}_g >, \tag{3.46}$$

where Equation 3.46 holds $\forall \mathbf{x}_j \in \mathcal{Z}^l, \ l = 1, \ldots, k,$ and $g = 1, \ldots, r_{k+1}$.

We can further reduce Equation 3.46 by noticing that $A_{0,d}(\hat{c}_g)(\mathbf{x}) = A_{0,d}(c_h)(\mathbf{x}) = 0$. Therefore, for $l = 1$,

$$\hat{w}_{j,g}^1 = \sum_{h=1}^{r_k} \left[w_{j,h}^1 \right] < \mathbf{u}_h, \hat{\mathbf{u}}_g > \quad \forall \mathbf{x}_j \in \mathcal{Z}^1, \ g = 1, \ldots, r_{k+1}. \tag{3.47}$$

For $l = 2$,

$$\hat{w}_{j,g}^2 + A_{1,d}(\hat{c}_g)(\mathbf{x}_j) = \sum_{h=1}^{r_k} \left[w_{j,h}^2 + A_{1,d}(c_h)(\mathbf{x}_j) \right] < \mathbf{u}_h, \hat{\mathbf{u}}_g > \quad \forall \mathbf{x}_j \in \mathcal{Z}^2, \ g = 1, \ldots, r_{k+1}. \tag{3.48}$$

Note that both interpolants $A_{k,d}(\hat{c}_g)(\mathbf{x})$ and $A_{k,d}(c_h)(\mathbf{x})$ share the same support nodes.

From the definition of the interpolant in Equation 3.29 and Equation 3.30, it follows that

$$A_{1,d}(c)(\mathbf{x}_j) = \Delta A_{1,d} = \sum_{n=1}^{m_1^\Delta} w_n^1 \Theta_n(\mathbf{x}_j). \tag{3.49}$$

Thus, Equation 3.48 becomes

$$\hat{w}_{j,g}^2 + \sum_{n=1}^{m_1^\Delta} \hat{w}_{n,g}^1 \Theta_n(\mathbf{x}_j) = \sum_{h=1}^{r_k} \left[w_{j,h}^2 + \sum_{n=1}^{m_1^\Delta} w_{n,h}^1 \Theta_n(\mathbf{x}_j) \right] < \mathbf{u}_h, \hat{\mathbf{u}}_g > \quad \forall \mathbf{x}_j \in \mathcal{Z}^2, \ g = 1, \ldots, r_{k+1}. \tag{3.50}$$

Using Equation 3.47 to replace $\hat{w}_{n,g}^1$ in Equation 3.50, we get

$$\hat{w}_{j,g}^2 + \sum_{n=1}^{m_1^\Delta} \sum_{h=1}^{r_k} w_{n,h}^1 < \mathbf{u}_h, \hat{\mathbf{u}}_g > \Theta_n(\mathbf{x}_j) = \sum_{h=1}^{r_k} \left[w_{j,h}^2 + \sum_{n=1}^{m_1^\Delta} w_{n,h}^1 \Theta_n(\mathbf{x}_j) \right] < \mathbf{u}_h, \hat{\mathbf{u}}_g >, \tag{3.51}$$

for all $\mathbf{x}_j \in \mathcal{Z}^2$, and $g = 1, \ldots, r_{k+1}$, which simplifies to

$$\hat{w}_{j,g}^2 = \sum_{h=1}^{r_2} w_{j,h}^2 < \mathbf{u}_h, \hat{\mathbf{u}}_g > \qquad \forall \mathbf{x}_j \in \mathcal{Z}^2, \ g = 1, \ldots, r_{k+1}. \tag{3.52}$$

Recursively, we find a general expression for $\hat{w}_{j,g}^k$ as

$$\hat{w}_{j,g}^k = \sum_{h=1}^{r_k} w_{j,h}^k < \mathbf{u}_h, \hat{\mathbf{u}}_g > \qquad \forall \mathbf{x}_j \in \mathcal{X}^k, \ g = 1, \ldots, r_{k+1}. \tag{3.53}$$

Since w_j^k is simply a measure of the deviation of the interpolant from the true value, this result means that we can simply obtain the surpluses at any iteration $k+1$ by taking this difference from the previous iterations to the physical space first, then projecting onto the new space formed by $\hat{\mathbf{u}}_g$. Equation 3.53 can be used first to update the surpluses for all previous points $\mathscr{X}^k$ before computing the surpluses at the new points $\mathscr{Z}^{k+1}$ (Equation 3.32).

To define the local error measure ϵ_j^k, we choose the L_2-norm error in physical space as follows:

$$\epsilon_j^k = \left\| \mathbf{y}(\mathbf{x}_j) - \sum_{h=1}^{r_k} A_{k,d}(c_h)(\mathbf{x}_j)\mathbf{u}_h \right\|_{L_2} \qquad \forall \mathbf{x}_j \in \mathscr{X}_k^\Delta, \tag{3.54}$$

and the candidate points $\mathscr{C}^k$ are selected as

$$\mathscr{C}^k = \{\mathbf{x}_j \in \mathscr{T}^k | \epsilon_j^k > (\gamma_{\text{int}} \left\| \mathbf{y}(\mathbf{x}_j) \right\|_{L_2} + \zeta_{\text{abs}})\}, \tag{3.55}$$

where γ_{int} is the interpolation threshold and ζ_{abs} is the absolute tolerance, which is introduced to deal with functions of small magnitudes. The important points $\mathscr{Z}^k$ are then selected by considering the ancestors as in Equation 3.33.

Additionally, we can define a global error estimate ($\epsilon_{\max}^k$) in the L_∞ norm

$$\epsilon_{\max}^k = \max_j \epsilon_j^k. \tag{3.56}$$

The algorithm is terminated once the global error is below a given relative tolerance ζ_{rel}. A trivial choice for the tolerance is to be equal to the selected threshold (i.e., $\zeta_{\text{rel}} = \gamma_{\text{int}}$). However, setting a different global tolerance is useful to avoid non-terminating algorithm or oversampling the high fidelity model. The algorithm terminating criterion that takes into account both the relative and absolute error is introduced as

$$\epsilon_{\max}^k < (\zeta_{\text{rel}} \left\| \mathbf{y}(\mathbf{x}_j) \right\| + \zeta_{\text{abs}}). \tag{3.57}$$

The algorithm is summarized in Algorithm 1 and the validation algorithm is summarized in Algorithm 2. Note that not all sampled points are included in the snapshots. The POD modes are formed only by the snapshots corresponding to the points in $\mathscr{X}^k$. Points that are not considered important (not included in $\mathscr{Z}^k$) are not included in the snapshots. This strategy reduces the computational cost of performing the SVD.

3.5. APPLICATIONS

In this section, we present three different numerical tests. The first is a reactor physics problem with 5 dimensions. This problem is presented in two cases in order to test the adaptive-POD algorithm in two different settings, the first is a strongly nonlinear setting and the second is weakly nonlinear. The second numerical test is a general diffusion problem with 18 dimensions, which demonstrates the performance of the algorithm in higher dimensionality. The validation algorithm is also tested in this problem. The final test case is an oscillatory analytical function in 20 dimensions, which tests the algorithm in identifying regions of strong nonlinearity.

Algorithm 1 Adaptive-POD

Require: an interpolation threshold γ_{int}, a relative tolerance ζ_{rel}, an absolute tolerance ζ_{abs}, a POD truncation threshold γ_{tr}, a greediness parameter μ, and a target model $y(\mathbf{x})$.

Ensure: grid points $\mathscr{X}^k$ and their surpluses $\mathscr{W}_h^k$, inactive set $\mathscr{I}^k$, and POD modes $\mathbf{u}_h$.

1: **Initialization**:
   - set $k = 1$, $\mathscr{X}^1 = \mathscr{Z}^1 = \{(0.5,\ldots,0.5)\}$, $\mathscr{I}^1 = \{\emptyset\}$, $\epsilon_1^1 = \inf$
   - evaluate the model at $\mathbf{y}(0.5,\ldots,0.5)$ and add the resulting snapshot to $\mathscr{F}$
   - perform SVD on the snapshots in $\mathscr{F}$ to obtain $\mathbf{u}_1$
   - compute the amplitude $c(0.5,\ldots,0.5) = <\mathbf{y}(0.5,\ldots,0.5),\mathbf{u}_1>$
   - set $\mathscr{W}_1^1 = \{c_1(0.5,\ldots,0.5)\}$

2: **while** any $\epsilon_j^k > (\zeta_{\text{rel}} \left\|\mathbf{y}(\mathbf{x}_j)\right\| + \zeta_{\text{abs}})$ **do**

3:    $k = k + 1$

4:    compute $\mathscr{T}^k = \left\{ \mathbf{x}_j \in \Psi(\mathscr{Z}^{k-1}) \;\middle|\; \frac{card(\Psi^{-1}(\mathbf{x}_j) \cap \mathscr{X}^{k-1})}{card(\Psi^{-1}(\mathbf{x}_j))} \geq 1 - \mu \right\} \cup \mathscr{I}^{k-1}$

5:    **for all** $\mathbf{x}_j \in \mathscr{T}^K$ **do**

6:       evaluate $\mathbf{y}(\mathbf{x}_j)$

7:       compute $A_{k-1,d}(c)(\mathbf{x}_j)$ as in Equation 3.29

8:       compute $\epsilon_j^k = \left\| \mathbf{y}(\mathbf{x}_j) - \sum_{h=1}^{r_k} A_{k-1,d}^h(c_h)(\mathbf{x}_j)\mathbf{u}_h \right\|_{L_2}$

9:    **end for**

10:    find the candidate points $\mathscr{C}^k = \{\mathbf{x}_j \in \mathscr{T}^k | \epsilon_j^k > (\gamma_{\text{int}} \left\|\mathbf{y}(\mathbf{x}_j)\right\|_{L_2} + \zeta_{\text{abs}})\}$.

11:    find $\mathscr{Z}_a^k$, $\mathscr{Z}_b^k$ and $\mathscr{Z}^k$ as in Equation 3.33

12:    compute $\mathscr{X}^k = \mathscr{Z}^k \cup \mathscr{X}^{k-1}$

13:    update the inactive set $\mathscr{I}^k = \{\mathscr{T}^k \backslash \mathscr{Z}^k\} \cup \mathscr{I}^{k-1}$

14:    add the snapshots corresponding to the points in $\mathscr{Z}^k$ to $\mathscr{F}$

15:    perform SVD on the snapshots in $\mathscr{F}$ to obtain new POD modes $\hat{u}_g$

16:    truncate the POD modes such that $e_r < \gamma_{\text{tr}}$

17:    use Equation 3.53 to update the surpluses for points in $\mathscr{X}^{k-1}$

18:    compute the surpluses at the selected points in $\mathscr{Z}^k$ as in Equation 3.32

19: **end while**

3.5.1. TEST CASE 1: POINT KINETICS

Models of nuclear reactors are complex because they aim to capture the dynamics of multi-physics phenomena occurring at various scales. The Point Kinetics model is a simple approximation that models the temporal evolution of the reactor power as a function of perturbations in the reactor. In this model, the spatial behaviour is not considered, and the reactor is condensed into a single point, hence the name. The model is described by a set of ordinary differential equations as follows [42]:

$$\frac{dP(t)}{dt} = \frac{\rho(t) - \beta}{\Lambda} P(t) + \lambda C(t), \tag{3.58}$$

$$\frac{dC(t)}{dt} = \frac{\beta}{\Lambda} P(t) - \lambda C(t), \tag{3.59}$$

Algorithm 2 Adaptive-POD Validation

Require: an interpolation threshold γ_{int}, a relative tolerance ζ_{rel}, an absolute tolerance ζ_{abs}, a POD truncation threshold γ_{tr}, a greediness parameter μ, a target model $y(\mathbf{x})$, grid points $\mathscr{X}^k$ and their surpluses $\mathscr{W}_h^k$, inactive set $\mathscr{I}^k$, POD modes u_h, and a number of random points to be tested ν.

Ensure: grid points $\mathscr{X}^k$ and their surpluses $\mathscr{W}_h^k$, inactive set $\mathscr{I}^k$, and POD modes u_h.

1: generate a set of ν random points $\mathscr{V}$
2: **for all** $\mathbf{x}_r \in \mathscr{V}$ **do**
3:   evaluate $\mathbf{y}(\mathbf{x}_r)$
4:   compute $A_{k,d}(c)(\mathbf{x}_r)$ as in Equation 3.29
5:   compute $\epsilon_r = \left\| \mathbf{y}(\mathbf{x}_r) - \sum_{h=1}^{r_k} A_{k,d}^h (c_h)(\mathbf{x}_r)\mathbf{u}_h \right\|_{L_2}$
6: **end for**
7: **while** any $\epsilon_r > (\zeta_{\text{rel}} \left\| \mathbf{y}(\mathbf{x}_r)\right\| + \zeta_{\text{abs}})$ **do**
8:   find $\mathscr{Q}$ as in Equation 3.35
9:   $k = k+1$
10: set $\mathscr{C}^k = \mathscr{Q}$
11: perform *Algorithm* 1 starting from step 11.
12: recompute ϵ_r for all $x_r \in \mathscr{V}$
13: **end while**

---

where $P(t)$ is the reactor power at time t, ρ is the reactivity of the reactor, β is the effective fraction of delayed neutrons, Λ is the neutron generation time, λ is the one-group decay constant, and $C(t)$ is the one-group precursors concentration.

A reactor is usually controlled by control rods which mostly influence the reactivity of the reactor. A positive reactivity $\rho > \beta$ causes the reactor power to be unstable in a very short period of time, a state called supercritical. On the other hand, $0 < \rho < \beta$ also causes a surge of power but at a much-reduced rate which allows for enough time to compensate and control the reactor. Therefore, the time evolution of the power is strongly nonlinear when the reactivity insertion is close to β. This problem is similar to the one presented by Perkó et al. (2014) [43], where the uncertainty in the maximum power was studied using adaptive polynomial chaos.

In this example, we consider a transient problem by assuming an insertion of a large positive reactivity that triggers the emergency safety system of the reactor. Thus, starting from an initial stable reactor at $t < 0$ ($\rho = 0$), the reactor is perturbed with a step reactivity insertion $\rho(t) = \rho_1 > 0$ at $t = 0$, causing the power to increase exponentially. At $t = t_s$, a strong negative step reactivity is inserted $\rho(t) = \rho_2 < 0$, which simulates the insertion of shutdown rods to stop the reactor. Under these assumptions, we can solve Equation 3.58 and Equation 3.59. For $0 \le t \le t_s$, we get

$$P(t) = P_0 \left[\frac{w^- - \frac{\rho_1}{\Lambda}}{w^- - w^+} e^{w^+ t} - \frac{w^+ - \frac{\rho_1}{\Lambda}}{w^- - w^+} e^{w^- t} \right], \tag{3.60}$$

$$C(t) = \frac{\beta P_0}{\Lambda \lambda} \left[\frac{w^-}{w^- - w^+} e^{w^+ t} - \frac{w^+}{w^- - w^+} e^{w^- t} \right], \tag{3.61}$$

where

$$w^{\pm} = \frac{\rho_1 - \beta - \lambda\Lambda \pm \sqrt{(\beta - \rho_1 + \lambda\Lambda)^2 + 4\lambda\Lambda\rho_1}}{2\Lambda}, \tag{3.62}$$

and P_0 is the initial power level.

The solution of the power for $t > t_s$ is

$$P(t) = \left[\frac{P_s w_s^- - P_s \frac{\rho_2 - \beta}{\Lambda} - \lambda C_s}{w_s^- - w_s^+} \right] e^{w_s^+(t - t_s)} - \left[\frac{P_s w_s^+ - P_s \frac{\rho_2 - \beta}{\Lambda} - \lambda C_s}{w_s^- - w_s^+} \right] e^{w_s^-(t - t_s)}, \tag{3.63}$$

where P_s and C_s are respectively the power and precursors concentration at $t = t_s$, and

$$w_s^{\pm} = \frac{\rho_2 - \beta - \lambda\Lambda \pm \sqrt{(\beta - \rho_2 + \lambda\Lambda)^2 + 4\lambda\Lambda\rho_2}}{2\Lambda}. \tag{3.64}$$

We study this model under variations of 5 parameters P_0, λ, Λ, ρ_1, and β. We assume a uniform distribution with $\pm 5\%$ change from the nominal values. We consider two settings. The first is a case where the reactivity insertion is close to the supercritical state ($\rho_1 = 0.9\beta$), which describes strong nonlinear behaviour. The second setting is selected such that the reactor is slightly further from the strong nonlinear behaviour, $\rho_1 = 0.7\beta$. Nominal values of both settings are shown in Table 3.2.

Table 3.2: Nominal values and the studied range of variations for Setting 1 and Setting 2 of the Point Kinetics model.

Parameter	Setting 1	Setting 2	Change
ρ_1	0.9β	0.7β	$\pm 5\%$
λ	$0.09441\,\frac{1}{s}$	$0.09441\,\frac{1}{s}$	$\pm 5\%$
Λ	$0.478 \times 10^{-6}\,\text{s}$	$0.478 \times 10^{-6}\,\text{s}$	$\pm 5\%$
P_0	$1\,\text{W}$	$1\,\text{W}$	$\pm 5\%$
β	0.00403	0.00403	$\pm 5\%$
ρ_2	-5β	-5β	0
t_s	10^{-2}s	10^{-2}s	0

For both settings the relative tolerance (ζ_{rel}) was set to 10^{-2}, absolute tolerance (ζ_{abs}) was 10^{-4}, interpolation threshold (γ_{int}) was 5×10^{-3}, truncation threshold (γ_{tr}) was 10^{-12}, and the greediness parameter μ was 1. A ROM model was built for each setting separately. In Setting 1, the algorithm stopped after sampling 944 points while for the second setting the algorithm required only 139 points, which is expected since Setting 2 is further from the unstable region. However, in both settings, only about 16% of the sampled points were included in the ROM. We then tested the models on 10000 randomly generated points that were not part of the training set. In order to show the progress of the ROM per iteration, Figure 3.6 shows the maximum relative L_2 error as a function of the number of model evaluations by testing the model at the random points after each iteration. On the same figure, we also compare the error with a different ROM model built using the classical sparse grid method (no adaptivity). The classical model is equivalent to setting the interpolation threshold γ_{int} and the absolute tolerance ζ_{abs} to be zero.

The advantage of the adaptive strategy compared to the classical is clear for Setting 1 (Figure 3.6a) as the tolerance was achieved with a much-reduced number of evaluations. The reason is that the classical algorithm adds points along all dimensions equally in every iteration whereas the adaptive algorithm only adds points in regions of higher error. As a consequence, the adaptive algorithm samples the most sensitive dimensions more densely. This is evident from the number of unique nodes in every dimension selected by the algorithm, which is shown in Table 3.3. For the initial power level, the algorithm sampled 5 nodes only, which are the first five nodes: the (center) root node 0.5, the children of the root node (the boundary nodes) 0 and 1, and their children 0.25 and 0.75. This means that the algorithm built a linear interpolant with the nodes 0.5, 0, and 1, then when it tested the interpolant at the nodes 0.25, and 0.75 the error was sufficiently low and no further refinement was needed. This is expected because the initial power level is a linear scaling factor, as can be seen from the solution in Equation 3.60. For the decay constant (λ), the algorithm assumes a constant nominal value because only 3 nodes were sampled. The root node results in the nominal value but when testing on its children (boundary nodes), the error was sufficiently low to stop any further refinement along this dimension. The neutron generation time (Λ) also has minimal effect on the power evolution, which is evident in the low number of selected unique nodes. As expected, the model is most sensitive to the reactivity (ρ_1) and the delayed neutron fraction (β), which the adaptive algorithm could recognize by adding more points along these dimensions.

Figure 3.7 shows a projection of the evaluated points in Setting 1 on the $\rho_1 - \beta$ plane. This figure illustrates the region where the algorithm selected the most points. The region corresponding to higher reactivity and lower delayed neutron fraction was considered the most important (mathematically $\rho_1 \approx \beta$). This was expected because a lower fraction of the delayed neutrons means the reactor responds stronger to perturbations with instantaneous prompt neutrons. Reactors with lower β are closer to the instability and harder to control. Higher reactivity insertion ρ_1 also causes the neutron population inside the reactor to multiply much faster. Additionally, Figure 3.6 shows the effect of reducing the tolerances by a factor of 10 (i.e., $\zeta_{rel} = 10^{-3}$, $\zeta_{abs} = 10^{-5}$, $\gamma_{int} = 5 \times 10^{-4}$). The algorithm produced a ROM that achieved the required tolerance with a higher number of model evaluations. Nevertheless, the adaptive algorithm in this case reduced the number of model evaluations compared to the classical approach by a factor of 6.

Table 3.3: Number of unique nodes in each dimension selected by the adaptive-POD algorithm in the Point Kinetics problem.

	P_0	λ	Λ	ρ_1	β
Number of nodes in Setting 1	5	3	5	31	29
Number of nodes in Setting 2	5	3	4	7	7

For Setting 2, both the adaptive and the classical models reached the required tolerance at similar rates, as shown in Figure 3.6b. The adaptivity had no advantage here since the problem is almost linear. Reducing the threshold brings the adaptive method closer to the classical performance because more points are admitted in the ROM per iteration. For example, Figure 3.6b shows that both the adaptive and the classical approaches

(a) Maximum relative L_2 error for the
strongly nonlinear Setting 1.

(b) Maximum relative L_2 error for the
weakly nonlinear Setting 2.
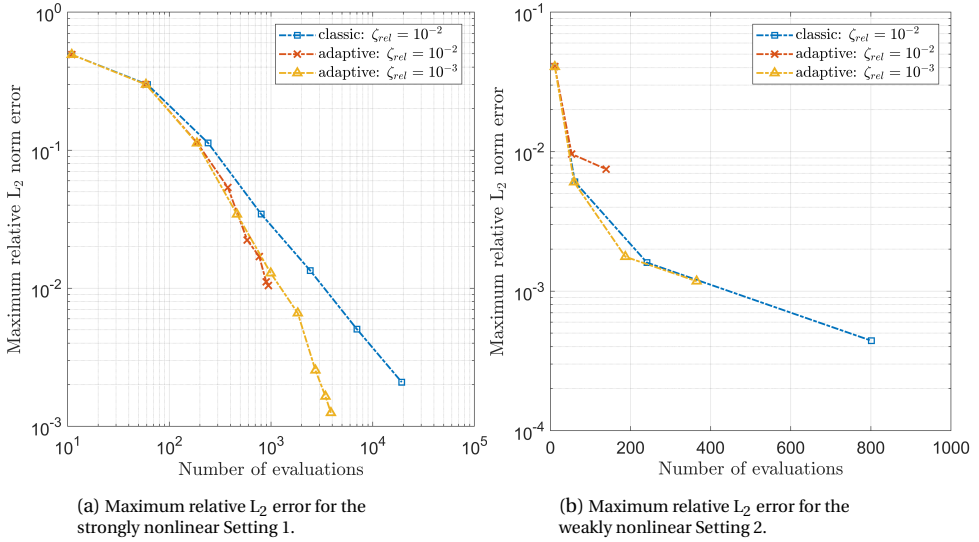
Figure 3.6: Point Kinetics: Maximum relative L_2 error as a function of the number of evaluations. The classical and the adaptive algorithms are compared. The effect of reducing all tolerances of the adaptive algorithm by a factor of 10 is shown (marked with $\zeta_{\text{rel}} = 10^{-3}$).

sampled 61 points by the second iteration. However, when tested on the random points, the classical approach had a lower error compared to the adaptive because the adaptive marked some of these points as inactive and did not include them in the ROM. Although Figure 3.6b shows the error to be lower than the tolerance by the second iteration, the adaptive algorithm converged after three iterations because the error estimate ($\epsilon_{\max}^k$) was still slightly above the tolerance at the second iteration. Reducing the threshold caused the adaptive to include all 61 points in the ROM, which also resulted in an error almost following the classical model for all iterations. Nevertheless, the adaptive algorithm is able to stop the sampling and converge to the required relative tolerance of 10^{-3} after 367 points while the classical approach can only stop after a predefined number of points, which in this case was 801 points.

Figure 3.8a and Figure 3.9a show the distribution of the relative L_2 error resulting from the tests for Setting 1 and Setting 2, respectively. In Setting 1, 99.98% of the points resulted in an error less than the tolerance of 1% while in Setting 2 all tested points resulted in errors less than the set tolerance. The maximum error found in Setting 1 was 1.05%. This point is simulated and compared to the reference solution in Figure 3.8b. The figure shows the power increase at $t = 0$ due to the positive reactivity ($\rho_1 = 0.00375$), which is then sharply reduced at $t = 10^{-2}$s due to the insertion of the shutdown rods ($\rho_2 = -5\beta$). In Setting 2, the maximum error was 0.75% which is shown in Figure 3.9b along with the corresponding reference solution. The initial power increase is due to a perturbation of $\rho_1 = 0.00293$ and the decrease is again due to the shutdown rods. The adaptive-POD model produced the simulations for the 10000 points in 3 seconds while the reference model required 15 seconds.
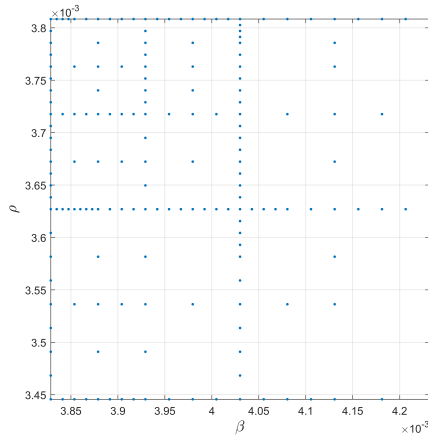
Figure 3.7: Point Kinetics: Projection of the sampled points on the $\rho_1 - \beta$ plane for Setting 1 with $\zeta_{\text{rel}} = 10^{-2}$.
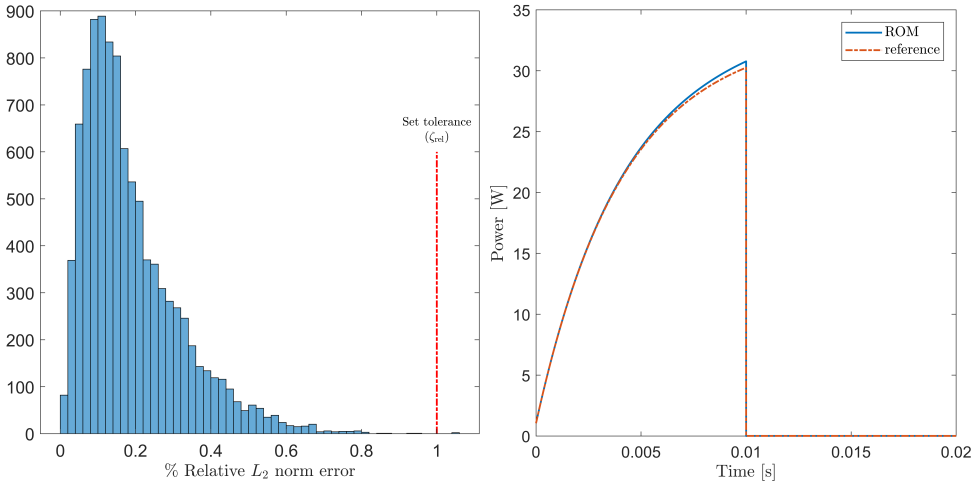
(a) Histogram of the relative L_2 error. 99.98% of the points resulted in an error less than the set tolerance.

(b) Simulation comparisons between the reference model and adaptive-POD ROM at the point of maximum error (1.05%).

Figure 3.8: Point Kinetics Setting 1: Histogram and simulation of the point of maximum error, which resulted from testing the ROM model on 10000 randomly generated points. The adaptive-POD model produced the simulations faster than the reference model by a factor of 5.

(a) Histogram of the relative L_2 error. 100% of the points resulted in an error less than the set tolerance.

(b) Simulation comparisons between the reference model and adaptive-POD ROM at the point of maximum error (0.75%).
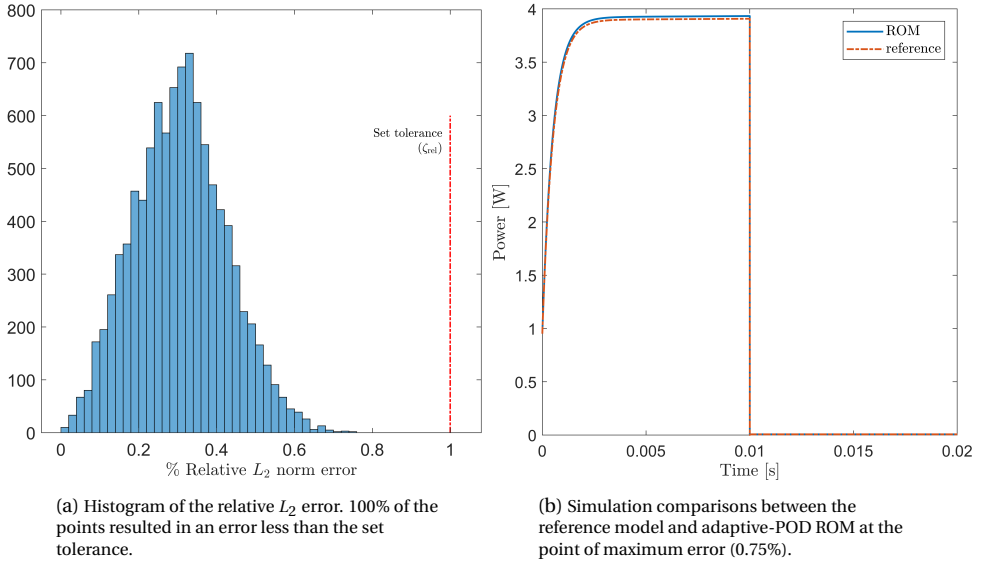
Figure 3.9: Point Kinetics Setting 2: Histogram and simulation of the point of maximum error, which resulted from testing the ROM model on 10000 randomly generated points. The adaptive-POD model produced the simulations faster than the reference model by a factor of 5.

In order to test the effect of the greediness parameter, we rerun the algorithm with reduced values of μ and tested the resultant ROM models on the same 10000 random points. We compare the number of model evaluations, the percentage of these evaluations utilized by the ROM, the maximum error found and the percentage of tested points with errors below the set tolerance. The results are summarized in Table 3.4. As expected, reducing the greediness parameter decreased the number of model evaluations and improved the utilization of these points. It can also be seen that this improvement compromised the accuracy of the model to some extent. For example, selecting $\mu = 0$ increased the maximum error for setting 1 to 4.7% and decreased the number of points below the tolerance to 89%. However, this model was built with only about 17% of the points needed by the default (greedy) model. The table also shows a case of reducing the interpolation threshold γ_{int} to 10^{-3} while $\mu = 0$. The model, in this case, matched the default model in terms of accuracy with 58% less points. The significant reduction in the number of points with decreasing μ can also be seen in Setting 2 and similar conclusions can be drawn.

3.5.2. TEST CASE 2: DIFFUSION

The diffusion equation has application in many scientific disciplines. We consider the time independent diffusion equation with a space dependent diffusion coefficient and a removal term

$$\boldsymbol{\nabla} \cdot D(\mathbf{r})\boldsymbol{\nabla}\phi(\mathbf{r}) + \alpha(\mathbf{r})\phi(\mathbf{r}) = Q(\mathbf{r}), \tag{3.65}$$

Table 3.4: Point Kinetics: Results of comparing different ROM models by varying the greediness parameter μ.

	μ	Number of model evaluations	% of utilized points	Maximum relative L_2 error	% of points $< \zeta_{\text{rel}}$
Setting 1	1	944	15.9%	1.05%	99.98%
	0.5	529	26.6%	1.11%	99.98%
	0	163	60.12%	4.78%	89.38%
	0^*	401	66.83%	1.06%	99.98%
Setting 2	1	139	15.8%	0.75%	100%
	0.5	91	24.1%	0.75%	100%
	0	43	46.5%	0.96%	100%
	0^*	77	28.6%	0.27%	100%

* $\gamma_{\text{int}} = 10^{-3}$

where $\phi(\mathbf{r})$ is the diffusing material at location $\mathbf{r}$, $D(\mathbf{r})$ is the diffusion coefficient at $\mathbf{r}$, $\alpha(\mathbf{r})$ is the removal coefficient, and $Q(\mathbf{r})$ is the source function. We consider a checkerboard domain with 9 regions as shown in Figure 3.10. Each region is considered to be homogeneous with constant properties across the region. Only the lower left corner (Region 1) has a uniform unit source, that is

$$Q(\mathbf{r}) = 1 \qquad \forall \, \mathbf{r} \in \text{Region 1}, \tag{3.66}$$

while the source is zero for all other regions. The boundary conditions were taken to be

$$\begin{aligned} \phi(\mathbf{r}) &= 0 & \text{on } S_D, \\ \hat{n} \cdot \boldsymbol{\nabla}\phi(\mathbf{r}) &= 0 & \text{on } S_N, \end{aligned} \tag{3.67}$$

where $\hat{n}$ is the outgoing normal vector, S_N is the outer boundary segment for the source region and S_D is the outer boundary segment for all other regions as shown in Figure 3.10. The diffusion equation was solved on a two dimensional plane (i.e., $\mathbf{r} = (x, y)$). We consider the model to be parametrized with respect to the diffusion and removal coefficients. Therefore, we have a total of 18 dimensions corresponding to the diffusion coefficients (D_n) and removal coefficients (α_n) within every region n. The diffusion coefficients were taken to be in the range [0.3 1.7], and the removal coefficient in [0.0075 0.0425], where the nominal value of the solution is the center of the range. The equation was solved using a Finite Element (FE) code, which was considered to be the reference model. We then build a ROM for the FE model using the adaptive-POD algorithm with relative tolerance (ζ_{rel}) set to 10^{-2}, absolute tolerance (ζ_{abs}) set to 10^{-4}, interpolation threshold (γ_{int}) set to 5×10^{-3}, truncation threshold (γ_{tr}) set to 10^{-12} and a value of 1 for the greediness parameter μ.

The algorithm converged after 8815 sampling points. As a benchmark showing the progress of the error during the construction stage, the model was tested on 1000 randomly generated points after each iteration. Figure 3.11a shows the maximum relative error per iteration as a function of the number of evaluations. The classical sparse grids

error is also shown in Figure 3.11a for comparison. Both the adaptive and the classical sparse grids needed a relatively large number of model evaluations due to the high dimensionality of the problem. However, the number of model evaluations used by the adaptive algorithm was less by a factor of 10 compared to the classical approach. The adaptive algorithm recognized that the model was more sensitive to property changes in the source region and sampled more points there compared to other regions. Figure 3.13 shows the total number of unique nodes selected in each region (i.e., the number of unique diffusion coefficient nodes and the number of unique removal coefficient nodes). The same figure shows that the algorithm placed more importance on the diffusion coefficient than the removal coefficient. In fact, the algorithm considered the removal coefficient to be constant in all regions at the nominal value except in the source region where it was considered to be a linear factor. Moreover, the projection of the points on the plane of diffusion and removal coefficients in the source region (Figure 3.14) shows that more importance was given to lower values of the diffusion coefficient because the solution is smoother for higher diffusivity.

The adaptive-POD algorithm sampled 8815 points but only 336 points were included in the ROM. The small fraction of important points indicates that the algorithm was oversampling the model, which can be attributed to the higher dimensionally of this problem. Despite that the algorithm eventually recognized the important dimensions, all dimensions needed to be searched at the initial steps, which increased the number of evaluations. As a test for the refinement criterion, we run the algorithm with the same tolerances again but by not including the ancestor points. Not including the ancestor points is equivalent to marking all candidate points C^k as important without considering their ancestry first (i.e., Equation 33 becomes $Z^k = C^k$). The algorithm without the ancestors converged after 12313 sampling points. The same set of 1000 random points used to test the default ROM was used to test the ROM resulting from this strategy. The resulting error per iteration is shown in Figure 3.11a. From this figure, it can be seen that both strategies were equivalent for the first three iterations. However, the strategy of including the ancestors reduced the error further and eventually converged with less model evaluations. This indicates that by not including the ancestor points in this problem, some behaviour of the reference model is overlooked, which results in increased model evaluations to compensate for the missing dynamics.

The solution resulting from the ROM at the point with the maximum error is shown in Figure 3.12a and the reference solution at the same point is shown in Figure 3.12b and the absolute difference is in Figure 3.12c. The adaptive-POD model outperformed the reference model in the time required to simulate 1000 points by a factor of 40. This factor represents the computational time required to simulate 1000 points with the reference model over the computational time required to simulate the same points with the ROM model. However, the histogram of the error in Figure 3.11b shows that only 15.6% of the tested points resulted in an error above the required tolerance. At this stage, the validation algorithm was started in order to reduce this percentage. We initiated the validation stage with the top 20 random points corresponding to the highest relative error. The validation algorithm required an additional 14532 points before convergence. However, only 21% of these points were included in the ROM. Then, we tested this model with a new set of 1000 random points. With this model, all tested points were less than the tolerance with the

Table 3.5: Diffusion: Results of comparing different ROM models by varying the greediness parameter μ.

μ	Number of model evaluations	% of utilized points	Maximum relative L_2 error	% of points $< \zeta_{\text{rel}}$
1	8,815	3.8%	1.6%	84.4%
0.5	1,720	9.8%	2.3%	77.4%
0	385	41.6%	3.1%	41.6%
0*	1,577	40.0%	1.0%	99.9%

* $\gamma_{\text{int}} = 10^{-3}$

maximum relative error found to be 0.6%. With the additional validation points, the total number of model evaluations was 23,347.

In Table 3.5, we compare models built with different values of the greediness parameter μ. Evidently, a greedy algorithm is an overkill for this problem because the percentage of utilized points was very small. By choosing a μ value of 0, the number of model evaluations was massively reduced. This can be explained by the fact that most dimensions of this problem were linear or only mildly non-linear. From Figure 3.13, we can deduce that, out of the 18 dimensions, 8 were considered constant (only 3 nodes were sampled along each of them), and 6 were considered linear (with 5 nodes). The greedy setting ($\mu = 1$), however, caused the algorithm to constantly search along these irrelevant dimensions as well in every iteration. By setting $\mu = 0$, the algorithm disregarded every point that had a backward point that was seen to be unimportant in previous iterations. Naturally, such non-greedy strategy resulted in overlooking some of the dynamics and the accuracy was deteriorated somewhat (e.g., 3.1% maximum error vs 1.6%). Nevertheless, reducing the value for the interpolation threshold compensated for the lower accuracy and resulted in a ROM model that outperformed the greedy model both in terms of the accuracy and the number of evaluations.
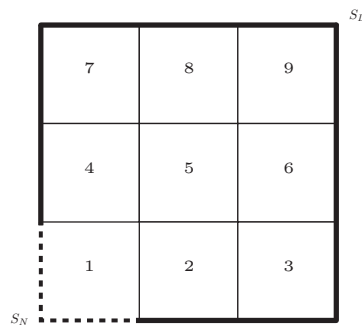
Figure 3.10: Domain of the diffusion problem showing the numbered 9 regions and the boundary segments S_D and S_N.

3

(a) Maximum relative L$_2$ error as a function of the number of evaluations for the diffusion problem.

(b) Histogram of the relative L_2 error. 84.4% of the points resulted in an error less than the set tolerance of 1%.
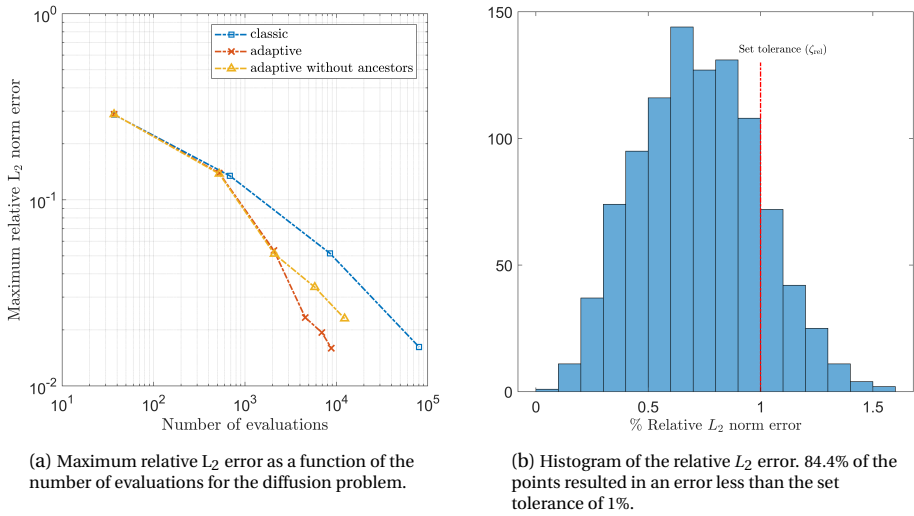
Figure 3.11: Diffusion: Maximum relative error per iteration after testing on 1000 random points and histogram of the relative error at the same points for the adaptive-POD ROM after convergence.

(a) Solution of the adaptive-POD ROM at the point with maximum error (1.6%).

(b) Solution of the reference FE code at the point with maximum error.

(c) The spatial absolute difference between the ROM and the FE code at the point with maximum error.
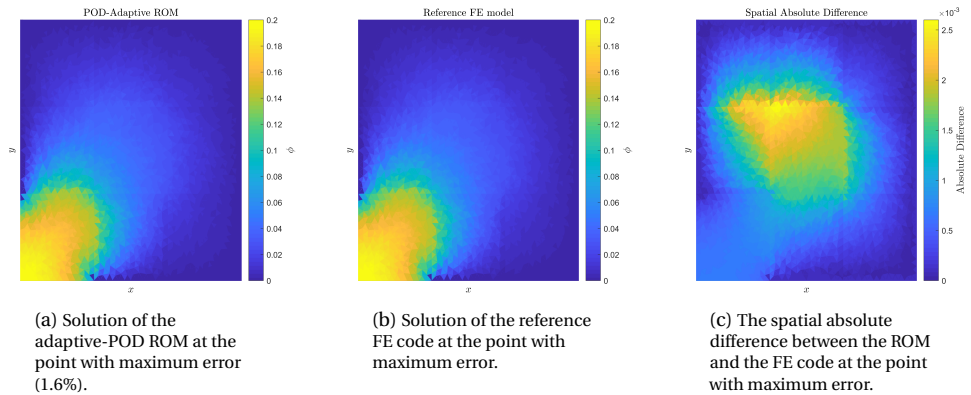
Figure 3.12: Diffusion: Comparison between the adaptive-POD ROM and the reference model at the point with maximum error resulting from testing on 1000 random points. The ROM produced the simulation faster than the reference model by a factor of 40.
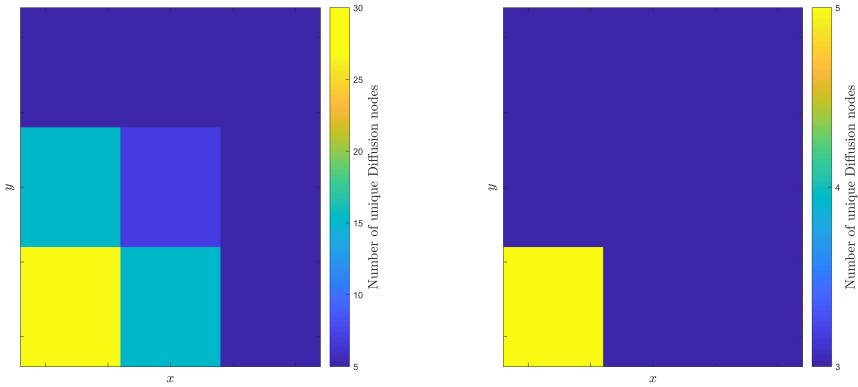
(a) Number of Diffusion coefficient nodes in each region.

(b) Number of Removal coefficient nodes in each region.

Figure 3.13: Diffusion: number of nodes in each region selected by the adaptive algorithm showing the importance of the regions as discovered by the algorithm.
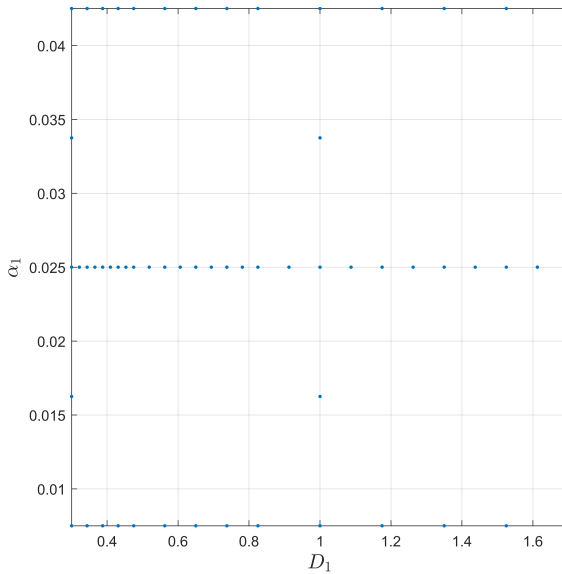
Figure 3.14: Diffusion: Projection of the sampled points on the plane of the diffusion and removal coefficients in the source region $(D_1 - \alpha_1)$.

3.5.3. TEST CASE 3: MODIFIED MORRIS FUNCTION

The Morris function is a single valued function in 20 dimensions that was developed to test optimization algorithms [44]. In order to test the adaptive-POD algorithm, we propose a modified version by having the output as a field defined over a 2-dimensional plane. We propose the modified Morris function as follows:

$$f(\mathbf{x}) = \sum_{i=1}^{20} \beta_i w_i \sin(i\pi x_1) + \sum_{i<j}^{20} \beta_{i,j} w_i w_j \sin(i\pi x_1) \sin(j\pi x_2)$$

$$+ \sum_{i<j<l}^{20} \beta_{i,j,l} w_i w_j w_l \sin(i\pi x_1) \sin(j\pi x_2) \cos(l\pi x_1)$$

$$+ \sum_{i<j<l<s}^{20} \beta_{i,j,l,s} w_i w_j w_l w_s \sin(i\pi x_1) \sin(j\pi x_2) \cos(l\pi x_1) \cos(s\pi x_2),$$

$$\tag{3.68}$$

where $x_1, x_2 \in [0,1]$, and

$$w_i = \begin{cases} 2\left(\frac{1.1\lambda_i}{\lambda_i + 0.1} - 0.5\right), & \text{for } i = 3,5,7 \\ 2(\lambda_i - 0.5), & \text{otherwise,} \end{cases} \tag{3.69}$$

where λ_i is the input parameter defined uniformly on the interval $[0,1]$ and the constants β defined as

$$\beta_i = \begin{cases} 20, & \text{for } i = 1,\dots,10 \\ (-1)^i, & \text{otherwise,} \end{cases} \qquad \beta_{i,j,l} = \begin{cases} -10, & \text{for } i,j,l = 1,\dots,5 \\ 0, & \text{otherwise,} \end{cases}$$

$$\beta_{i,j} = \begin{cases} -15, & \text{for } i,j = 1,\dots,6 \\ (-1)^{i+j}, & \text{otherwise,} \end{cases} \qquad \beta_{i,j,l,s} = \begin{cases} 5, & \text{for } i,j,l,s = 1,\dots,4 \\ 0, & \text{otherwise.} \end{cases}$$

$$\tag{3.70}$$

The modified Morris function is parametrized on 20 dimensions corresponding to λ_i. The function is linear in all dimensions except three: λ_3, λ_5, and λ_7. A ROM was built using the adaptive-POD algorithm with relative tolerance (ζ_{rel}) equal to 5×10^{-3}, absolute tolerance (ζ_{abs}) set to 5×10^{-4}, interpolation threshold (γ_{int}) equal to 5×10^{-3}, truncation threshold (γ_{tr}) equal to 10^{-12}, and the greediness parameter $\mu = 1$.

This problem was found to be challenging to the algorithm due to the higher dimensionality and the oscillatory nature of the solution. The algorithm converged after sampling 44,297 points, but only 1874 points were admitted to the ROM, which implied that the algorithm was oversampling. The produced ROM model was tested on 1000 random points after each iteration. Figure 3.15a shows the maximum relative error resulting from the tests as a function of the number of evaluations. For comparison, the error from a ROM with classical sparse grid sampling is also shown on the same figure. The classical approach requires significantly more points. The main reason for the difference in performance is that the adaptive-POD algorithm correctly recognized that λ_3, λ_5, λ_7 are nonlinear and selected 31 unique nodes for λ_3, 31 nodes for λ_5, and 29 nodes for λ_7. For all other dimensions, 5 nodes were sampled, which is an accurate linear assumption.

Table 3.6: Modified Morris: Results of comparing different ROM models by varying the greediness parameter μ.

μ	Number of model evaluations	% of utilized points	Maximum relative L_2 error	% of points $< \zeta_{\text{rel}}$
1	44,296	3.8%	0.88%	44.3%
0.5	19,852	8.9%	1.11%	14.8 %
0	9,928	17.2%	1.15%	13.3%
0^*	13,872	22.4%	0.3%	100%

* $\gamma_{\text{int}} = 10^{-3}$

3

Moreover, lower values of λ_i were found to be more important than higher values. This is illustrated in Figure 3.16, which shows the projection of the sampled points on the $\lambda_1 - \lambda_3$ plane. The selected important points are also marked on the figure. For λ_1, the algorithm selected 5 nodes and marked only 3 as important (the center root node and the boundaries). On the other hand, λ_3 is nonlinear and was more refined in the lower region. Identifying this important region enabled the algorithm to reduce the number of evaluations considerably compared to the classical approach. Comparison between the ROM and the reference solution at the point of maximum error is shown in Figure 3.17a and Figure 3.17b while Figure 3.17c shows the absolute difference between the two solutions. Histogram of the error (Figure 3.15b) shows that the error almost followed a normal distribution around the set tolerance but only 44.3% of the points were strictly below the tolerance. The reference model for the modified Morris function was implemented with nested loops which required about 3 seconds to complete a single evaluation. The time to evaluate 1000 points with the reference model was about 3241 seconds while the ROM model needed only 33 seconds to evaluate the same points, which is a reduction by about a factor of 100 in computational time.

On Figure 3.15a, we also show the effect of reducing the greediness of the algorithm (decreasing μ). Table 3.6 summarizes the differences in performance between these models. As expected, lower values of the μ converged faster but with reduced accuracy. We notice again that the non-greedy ROM model with $\mu = 0$ and reduced interpolation threshold ($\gamma_{\text{int}} = 10^{-3}$) was the best compromise between accuracy and efficiency. In order to highlight the difference in the sampling scheme with respect to varying μ, Figure 3.16 shows the projection of the sampled points on the $\lambda_1 - \lambda_3$ plane for the two cases of greedy $\mu = 1$ and non-greedy $\mu = 0$. At the boundaries of λ_1, we notice that, for every refinement along λ_3, the algorithm also searches the children of these boundaries (nodes 0.25, and 0.75). However, λ_1 is linear and these points never reveal any significant dynamics to be marked important. This search creates an unnecessary line of points at the inner nodes $\lambda_1 = 0.25$, and 0.75. Of course, this line is repeated for all linear dimensions, which is one of the causes for the additional cost in model evaluations compared to the non-greedy setting. For the case of $\mu = 0$, this line is eliminated because the algorithm stopped searching dimension λ_1 after this dimension was found to be linear along the line $\lambda_3 = 0.5$. This successfully reduced the number of model evaluations without loss in accuracy.

(a) Maximum relative L_2 error as a function of the number of evaluations for the diffusion problem.

(b) Histogram of the relative L_2 error. 44.3% of the points resulted in an error less than the set tolerance of 0.5%.
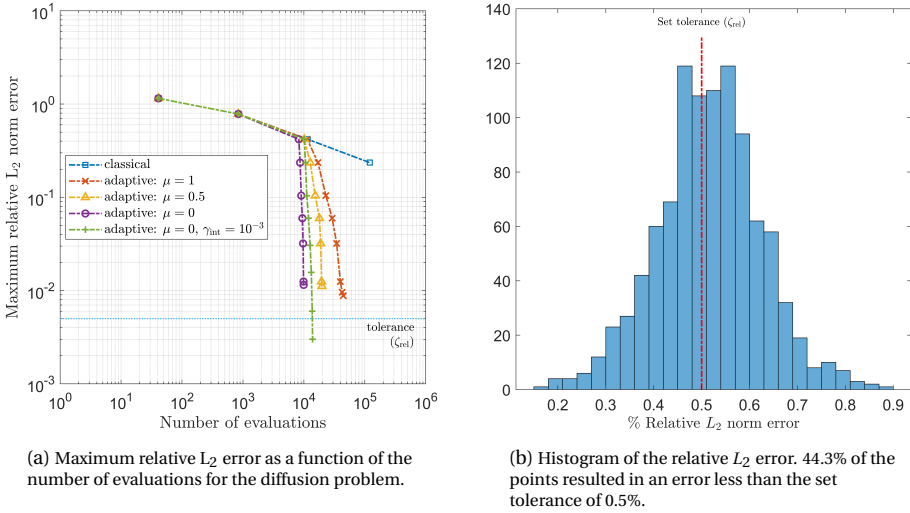
Figure 3.15: Modified Morris: Maximum relative error per iteration after testing on 1000 random points and histogram of the relative error at the same points for the adaptive-POD ROM after convergence

3.6. CONCLUSIONS

We have presented a practical approach for integrating the locally adaptive sparse grid technique with the POD method to develop a nonintrusive ROM algorithm. The local adaptivity provides an effective sampling scheme for the POD snapshots while the hierarchical interpolant builds a surrogate model for the POD amplitudes. For increased robustness of the refinement strategy, the ancestor points are prioritized when selecting the important points. We also introduced a greediness parameter that provides additional control of the number of model evaluations during the construction phase. Additionally, a validation algorithm was presented with the purpose of not only to certify the model after convergence but also enhancing it once a discrepancy is detected. In addition, an efficient way of updating the surpluses with every POD modes update was presented, which reduces the computational burden of recalculating the interpolant after each iteration.

The developed adaptive-POD algorithm was tested numerically on three applications. In all tests, the algorithm considered the system as a black box without any knowledge of the equations being solved. Table 3.7 summarizes the results of all test cases for values of $\mu = 0$ and $\gamma_{\text{int}} = 10^{-3}$. Every ROM model was then tested on random points that were not part of the training set. In all test cases, the built ROM model was able to reduce the computational time and provide solutions in good agreements with the reference model (errors were within the set tolerances). The computational time to perform these simulations was reduced by a factor of 5 in the Point kinetics case, by a factor of 40 in the Diffusion case, and by a factor of 100 in the modified Morris case. We compared the POD-adaptive with the classical (non-adaptive) sparse grids technique. The adaptive-POD reduced the number of model evaluations compared to the classical approach significantly. This was most evident in the nonlinear cases (i.e., Point Kinetics Setting 1, Diffusion, and the modified Morris function). For linear and smooth problems, as in the

(a) Projection of the sampled points for the greedy algorithm ($\mu = 1$).

(b) Projection of the sampled points for the non greedy algorithm ($\mu = 0$).
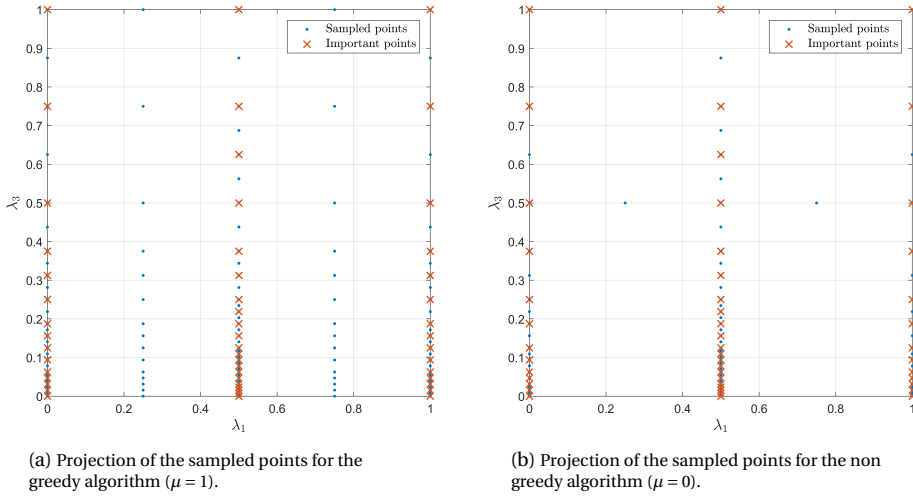
Figure 3.16: Modified Morris: Projection of the sampled points on the plane of $\lambda_1 - \lambda_3$ comparing the selected points between the greedy ($\mu = 1$) and the non-greedy ($\mu = 0$) algorithms. The selected important points ($\mathcal{X}^k$) are marked with crosses.

Point Kinetics Setting 2, both the adaptive and the classical approaches provided similar performance. For all applications not only did the algorithm recognize the important dimensions but it also recognized the important range within that dimension. Therefore, although the method is locally adaptive, it is also implicitly (globally) dimension adaptive.

The greediness parameter reduced the total number of model evaluations and improved the utilization of the sampled points significantly. As expected, problems with higher dimensions tend to have a smaller percentage of utilized points. The models for the Diffusion and the modified Morris problems utilized around 4% of the points whereas the lower dimensional Point Kinetics model included 15% of the points. This indicates that the algorithm explored the parameter space for these problems by oversampling the reference model. By reducing the greediness parameter, the quantity and utilization of

Table 3.7: Summary of results for all test cases using $\mu = 0$ and $\gamma_{\text{int}} = 10^{-3}$.

	Point Kinetics Setting 1	Point Kinetics Setting 2	Diffusion	Modified Morris
Dimensionality	5	5	18	20
Number of model evaluations	401	77	1577	13,872
Speed-up factor[*]	5	5	40	100
Maximum relative L_2 error	1.06%	0.27%	1.0%	0.3%
% of points $< \zeta_{\text{rel}}$	99.9%	100%	99.9%	100%

[*] The speed-up factor is the ratio of the computational time required to simulate 1000 points with the reference model to the computational time required to simulate the same points with the ROM model.

(a) Simulation of the adaptive-POD ROM at the point with maximum error (0.88%).

(b) Solution of the reference model at the point with maximum error.

(c) The spatial absolute difference between the ROM and the reference model at the point with maximum error.
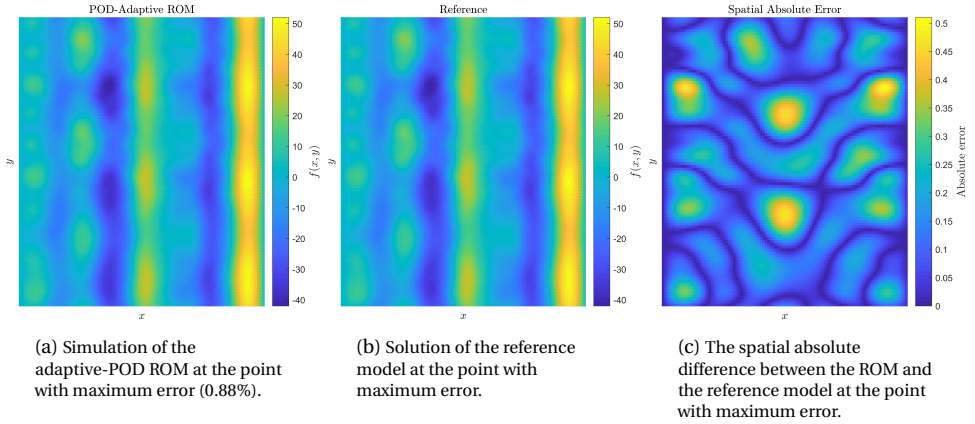
Figure 3.17: Modified Morris: Comparison between the adaptive-POD ROM and the reference model at the point with maximum error resulting from testing on 1000 random points. The ROM produced the simulation faster than the reference model by a factor of 100.

the sampled point were improved. It was shown that this improvement compromised the accuracy of the model to some extent. Nevertheless, the lost accuracy was recovered with a reduced interpolation threshold (γ_{int}), and a compromise between accuracy and efficiency was achieved. While the risk of overlooking localized dynamics is increased for a reduced μ, the developed validation stage can reveal any missing dynamics and update the ROM model in such cases.

The algorithm can still be improved with the regards to the sampling and utilization of points. For large scale, very high-dimensional problems (e.g., in the hundreds), the main limitation of the algorithm is the large number of model evaluations potentially required. Knowledge of the physics of the system can be helpful to screen and reduce the dimensionality a priori. Additionally, limiting the range of the input parameters will restrict the space in which the algorithm searches the domain. We can also use a different unidimensional rule that limits the scope of the search to specific important regions. For example, we can choose a rule with no boundary points if we know that the sensitivity of the model to variations at the boundaries are small. Moreover, using higher order basis functions is an interesting area to investigate for reducing the number of model evaluation and improving the utilization of the points. These areas are subjects of future research.

References

[1] F. Alsayyari, Z. Perkó, D. Lathouwers, and J. L. Kloosterman, *A nonintrusive reduced order modelling approach using Proper Orthogonal Decomposition and locally adaptive sparse grids,* Journal of Computational Physics **399**, 108912 (2019).

[2] P. Benner, S. Gugercin, and K. Willcox, *A Survey of Projection-Based Model Reduction Methods for Parametric Dynamical Systems,* SIAM review **57**, 483 (2015).

[3] V. Buljak, *Inverse Analyses with Model Reduction: Proper Orthogonal Decomposition in Structural Mechanics* (Springer, Berlin, 2012).

[4] J. L. Lumley, *The Structure of Inhomogeneous Turbulent Flows,* Atmospheric Turbulence and Radio Wave Propagation **790**, 166 (1967).

[5] R. Bourguet, M. Braza, and A. Dervieux, *Reduced-order modeling of transonic flows around an airfoil submitted to small deformations,* Journal of Computational Physics **230**, 159 (2011).

[6] A. Placzek, D.-M. Tran, and R. Ohayon, *A nonlinear POD-Galerkin reduced-order model for compressible flows taking into account rigid body motions,* Computer Methods in Applied Mechanics and Engineering **200**, 3497 (2011).

[7] C. W. Rowley, T. Colonius, and R. M. Murray, *Model reduction for compressible flows using POD and Galerkin projection,* Physica D: Nonlinear Phenomena **189**, 115 (2004).

[8] M. Bergmann and L. Cordier, *Optimal control of the cylinder wake in the laminar regime by trust-region methods and POD reduced-order models,* Journal of Computational Physics **227**, 7813 (2008).

[9] Z. Wang, I. Akhtar, J. Borggaard, and T. Iliescu, *Proper orthogonal decomposition closure models for turbulent flows: A numerical comparison,* Computer Methods in Applied Mechanics and Engineering **237-240**, 10 (2012).

[10] J. Baiges, R. Codina, and S. R. Idelsohn, *Reduced-Order Modelling Strategies for the Finite Element Approximation of the Incompressible Navier-Stokes Equations,* in *Computational Methods in Applied Sciences* (Springer International Publishing, 2014) pp. 189–216.

[11] A. G. Buchan, C. C. Pain, F. Fang, and I. M. Navon, *A POD reduced-order model for eigenvalue problems with application to reactor physics,* International Journal for Numerical Methods in Engineering **95**, 1011 (2013).

[12] 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 Rreactor Spatial Kinetics,* Annals of Nuclear Energy **71**, 217 (2014).

[13] 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 **46**, 263 (2017).

[14] A. Quarteroni, G. Rozza, and A. Manzoni, *Certified reduced basis approximation for parametrized partial differential equations and applications,* Journal of Mathematics in Industry **1**, 3 (2011).

[15] G. Rozza, D. B. P. Huynh, and A. T. Patera, *Reduced Basis Approximation and a Posteriori Error Estimation for Affinely Parametrized Elliptic Coercive Partial Differential Equations,* Archives of Computational Methods in Engineering **15**, 229 (2008).

[16] M. A. Grepl and A. T. Patera, *A posteriori error bounds for reduced-basis approximations of parametrized parabolic partial differential equations,* ESAIM: Mathematical Modelling and Numerical Analysis **39**, 157 (2005).

[17] H. V. Ly and H. T. Tran, *Modeling and Control of Physical Processes using Proper Orthogonal Decomposition,* Mathematical and Computer Modelling **33**, 223 (2001).

[18] C. Audouze, F. D. Vuyst, and P. B. Nair, *Nonintrusive reduced-order modeling of parametrized time-dependent partial differential equations,* Numerical Methods for Partial Differential Equations **29**, 1587 (2013).

[19] J. Hesthaven and S. Ubbiali, *Non-intrusive reduced order modeling of nonlinear problems using neural networks,* Journal of Computational Physics **363**, 55 (2018).

[20] N. Nguyen and J. Peraire, *Gaussian functional regression for output prediction: Model assimilation and experimental design,* Journal of Computational Physics **309**, 52 (2016).

[21] A. Giunta, S. Wojtkiewicz, and M. Eldred, *Overview of modern design of experiments methods for computational simulations,* in *41st Aerospace Sciences Meeting and Exhibit* (American Institute of Aeronautics and Astronautics, 2003).

[22] M. Guénot, I. Lepot, C. Sainvitu, J. Goblet, and R. Filomeno Coelho, *Adaptive Sampling Strategies for Non-intrusive POD-based Surrogates,* Engineering Computations **30**, 521 (2013).

[23] S. A. Smolyak, *Quadrature and interpolation formulas for tensor products of certain classes of functions,* in *Dokl. Akad. Nauk SSSR*, Vol. 4 (1963) pp. 240–243.

[24] V. Barthelmann, E. Novak, and K. Ritter, *High Dimensional Polynomial Interpolation on Sparse Grids,* Advances in Computational Mathematics **12**, 273 (2000).

[25] B. Peherstorfer, *Model Order reduction of Parametrized Systems with Sparse Grid Learning Techniques,* Ph.D. thesis, Technische Universität München, München (2013).

[26] D. Xiao, F. Fang, A. Buchan, C. Pain, I. Navon, and A. Muggeridge, *Non-intrusive Reduced Order Modelling of the Navier–Stokes Equations,* Computer Methods in Applied Mechanics and Engineering **293**, 522 (2015).

[27] R. S. Tuminaro, E. T. Phipps, C. W. Miller, and H. C. Elman, *Assessment of collocation and Galerkin approaches to linear diffusion equations with random data,* International Journal for Uncertainty Quantification **1**, 19 (2011).

[28] T. Gerstner and M. Griebel, *Dimension-adaptive tensor-product quadrature,* Computing **71**, 65 (2003).

[29] P. Chen and A. Quarteroni, *A new algorithm for high-dimensional uncertainty quantification based on dimension-adaptive sparse grid approximation and reduced basis methods,* Journal of Computational Physics **298**, 176 (2015).

[30] C. Zenger, *Sparse Grids,* in *Parallel Algorithms for Partial Dierential Equations,* Notes on Numerical Fluid Mechanics, Vol. 31, edited by W. Hackbusch (Vieweg, 1990) pp. 241 –251.

[31] H.-J. Bungartz and M. Griebel, *Sparse grids,* Acta Numerica **13**, 147 (2004).

[32] M. Griebel, *Adaptive sparse grid multilevel methods for elliptic PDEs based on finite differences,* Computing **61**, 151 (1998).

[33] D. Pflüger, B. Peherstorfer, and H.-J. Bungartz, *Spatially adaptive sparse grids for high-dimensional data-driven problems,* Journal of Complexity **26**, 508 (2010).

[34] X. Ma and N. Zabaras, *An adaptive hierarchical sparse grid collocation algorithm for the solution of stochastic differential equations,* Journal of Computational Physics **228**, 3084 (2009).

[35] X. Ma and N. Zabaras, *An adaptive high-dimensional stochastic model representation technique for the solution of stochastic partial differential equations,* Journal of Computational Physics **229**, 3884 (2010).

[36] J. D. Jakeman, R. Archibald, and D. Xiu, *Characterization of discontinuities in high-dimensional stochastic problems on adaptive sparse grids,* Journal of Computational Physics **230**, 3977 (2011).

[37] H.-J. Bungartz and S. Dirnstorfer, *Multivariate quadrature on adaptive sparse grids,* Computing **71**, 89 (2003).

[38] J. Brumm and S. Scheidegger, *Using Adaptive Sparse Grids to Solve High-Dimensional Dynamic Models,* Econometrica **85**, 1575 (2017).

[39] P. Holmes, J. L. Lumley, and G. Berkooz, *Turbulence, Coherent Sructures, Dynamical Systems and Symmetry* (Cambridge University Press, Cambridge, 1996).

[40] F. Nobile, R. Tempone, and C. G. Webster, *A Sparse Grid Stochastic Collocation Method for Partial Differential Equations with Random Input Data,* SIAM Journal on Numerical Analysis **46**, 2309 (2008).

[41] A. Klimke, *Uncertainty Modeling using Fuzzy Arithmetic and Sparse Grids,* Ph.D. thesis, Universität Stuttgart, Stuttgart (2006).

3

[42] J. J. Duderstadt and L. J. Hamilton, *Nuclear Reactor Analysis* (John Wiley & Sons, Inc., 1976).

[43] Z. Perkó, L. Gilli, D. Lathouwers, and J. L. Kloosterman, *Grid and basis adaptive polynomial chaos techniques for sensitivity and uncertainty analysis,* Journal of Computational Physics **260**, 54 (2014).

[44] M. D. Morris, *Factorial sampling plans for preliminary computational experiments,* Technometrics **33**, 161 (1991).

3

4

UNCERTAINTY AND SENSITIVITY ANALYSIS OF A MOLTEN SALT REACTOR SYSTEM

We use a novel nonintrusive adaptive Reduced Order Modeling method to build a reduced model for a molten salt reactor system. Our approach is based on Proper Orthogonal Decomposition combined with locally adaptive sparse grids. Our reduced model captures the effect of 27 model parameters on k_{eff} of the system and the spatial distribution of the neutron flux and salt temperature. The reduced model was tested on 1000 random points. The maximum error in multiplication factor was found to be less than 50 pcm and the maximum L_2 error in the flux and temperature were less than 1%. Using 472 snapshots, the reduced model was able to simulate any point within the defined range faster than the high-fidelity model by a factor of 5×10^6. We then employ the reduced model for uncertainty and sensitivity analysis of the selected parameters on k_{eff} and the maximum temperature of the system.

4.1. Introduction

COMPLEX systems such as molten salt reactors impose a modeling challenge because of the interaction between multi-physics phenomena (radiation transport, fluid dynamics and heat transfer). Such complex interaction is captured with high-fidelity, coupled models. However, these models are computationally expensive for applications of uncertainty quantification, design optimization, and control, where many repeated evaluations of the model are needed. Reduced Order Modeling (ROM) is an effective tool for such applications. This technique is based on recasting the high fidelity, high dimensional model into a simpler, low dimensional model that captures the prominent dynamics of the system with a controlled level of accuracy. Many ROM approaches can be found in literature [2]. However, amongst studied ROM methods, Proper Orthogonal Decomposition (POD) is the suitable method for parametrized, nonlinear systems [3]. The POD approach is divided into two main phases: the first is the *offline* phase, where the reduced order model is constructed by solving the high fidelity model at several points in parameter space to obtain a reduced basis space; the second is the *online* phase, in which the reduced model is used to replace the high fidelity model in solving the system at any desired point with a reduced computational burden.

POD can be implemented intrusively by projecting the reduced basis onto the system's governing equations or non-intrusively by building a surrogate model for the POD coefficients. Many studies have successfully implemented projection based POD for nuclear applications [4–8].

However, for practical nuclear reactor applications, the intrusive approach is often challenging because these models are usually implemented with legacy codes that prohibit access to the governing equations, or built with coupled codes that renders modifying the governing equations a complicated task. In this case, a nonintrusive approach can be adopted to build a surrogate model for the coefficients of the POD basis. Simple interpolation or splines can be used [9] or for high-dimensional problems, Radial Basis Function (RBF) is usually employed [10]. Neural networks [11] and Gaussian regression [12] have also been studied to build the surrogate model. These approaches rely on standard sampling schemes (Monte Carlo, Latin Hypercube Sampling, tensorized uniform) to generate the snapshots. Such strategies do not take into account the dynamics of the problem and can be expensive for problems parametrized on high-dimensional spaces. Audouze et al. (2009) [13] suggested tackling this issue by combining the POD-RBF method with a greedy residual search. In this approach, the residual of the PDE is used as an error estimator by iteratively placing sampling points at locations that minimize the residual until a certain global criterion is achieved. However, this method requires repeated evaluations of the residual, which can be expensive in some solvers (e.g., matrix-free solvers) or unavailable for legacy solvers.

In this work, we propose the use of ROM method that combines the non-intrusive POD approach with the sparse grids technique [14] to build a reduced model of a fast-spectrum molten salt system. Our approach is implemented using a previously developed algorithm [15] that uses locally adaptive sparse grids as a sampling strategy for selecting the POD snapshots efficiently. The adaptivity is completely nonintrusive to the governing equations. In addition, the algorithm provides a criterion to terminate the iterations, which can be used as a heuristic estimation for the error in the developed reduced model.

In this work, we extend the algorithm to deal with multiple fields of outputs. In addition, we demonstrate how local derivatives can be computed for local sensitivity analysis. The liquid-fueled system under investigation is a simplified system that captures the main characteristics of the Molten Salt Fast Reactor [16]. An in-house multi-physics tool [17], coupling an S_N radiation transport code with an incompressible Navier-Stokes solver, was considered as the reference model of the molten salt system. We use the developed adaptive-POD (aPOD) algorithm to construct a ROM for this reference model. We then employ the built reduced model for an uncertainty and sensitivity analysis application to study the effect of the parameters on the maximum temperature and the multiplication factor. The uncertainty and sensitivity analysis was accomplished with extensive random sampling of the reduced model. Such approach is only achievable due to the efficiency provided by the reduced model over the reference model.

The remainder of this chapter is organized as follows: the POD method is briefly introduced in Section 4.2. Section 4.3 presents the sparse grids approach by introducing the interpolation technique first followed by the method for selecting the sampling points. The aPOD algorithm along with the approach to deal with multiple fields of outputs and computing the local derivatives are presented in Section 4.4. The model for the molten salt system is given in Section 4.5. The discussion of the results of constructing the reduced model along with the uncertainty and sensitivity analysis are in Section 4.6. Finally, conclusions are presented in Section 4.8.

4.2. Proper Orthogonal Decomposition

In a nonintrusive manner, the Proper Orthogonal Decomposition can build a ROM by considering the reference, high fidelity model as a black box mapping a given input to the desired output. Let the reference model $f(y; \mathbf{x})$ be dependent on state y and a vector of input parameters $\mathbf{x}$. We can then find an expansion approximating the model as follows:

$$f(y; \mathbf{x}) \approx \sum_{i=1}^{r} c_i(\mathbf{x}) \, u_i(y), \tag{4.1}$$

where c_i is the expansion coefficients which depends on the input parameter $\mathbf{x}$ and $u_i(y)$ is the corresponding basis function.

The POD method seeks to find the optimal basis functions $u_i(y)$ that minimizes the error in L_2 norm,

$$\min_{u_i(y)} \left\| f(y; \mathbf{x}) - \sum_{i=1}^{r} c_i(\mathbf{x}) u_i(y) \right\|_{L_2}. \tag{4.2}$$

The basis functions are chosen such that they are orthonormal. Thus, the coefficients $c_i(\mathbf{x})$ can be computed as

$$c_i(\mathbf{x}) = < f(y; \mathbf{x}), u_i(y) >, \tag{4.3}$$

where $< f(x), g(x) > = \int f(x) g(x) dx$.

Assuming that the reference model is discretized ($f(y; \mathbf{x}) \to \mathbf{f}(\mathbf{x})$), The POD snapshot method finds the solution to the minimization problem using the Singular Value Decomposition (SVD). This approach begins with sampling the reference model at discrete points in parameter space $[\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_p]$, where p is the number of sampling points. Then, the

corresponding outputs $[\mathbf{f}(\mathbf{x}_1),\dots,\mathbf{f}(\mathbf{x}_p)]$ can be arranged in a matrix M called the snapshot matrix. Finally, we obtain the basis vectors (also called POD modes) $\mathbf{u}_i$ as the the first r left singular vectors of the SVD on the matrix M, where r is chosen to be less than or equal to the rank of the matrix M. A truncation error can be quantified using the singular values of the SVD (σ) if r is chosen to be strictly less than the rank of M,

$$e_{\text{tr}} = \frac{\sum_{k=r+1}^{n} \sigma_k^2}{\sum_{k=1}^{n} \sigma_k^2}, \tag{4.4}$$

where n is the rank of M. e_{tr} quantifies the error in approximating the solutions contained in the snapshot matrix.

4.3. SPARSE GRIDS

For an accurate POD reduced model, the snapshots need to cover the entire dynamics of the reference model within the defined range of input parameters. Therefore, selecting an effective sampling strategy is crucial for the success of the reduced model. We propose an algorithm that is based on locally adaptive sparse grids to select the sampling points. The sparse grid algorithm builds a surrogate model for each of the POD coefficients using a Smolyak interpolant. Iteratively, the algorithm identifies a set of important points and samples their neighbouring points in the next iteration [18]. This process is repeated until a global convergence criterion is met. In this section we introduce the methods for the interpolation and the selection of the sampling points.

4.3.1. INTERPOLATION

The Smolyak interpolation is a hierarchical interpolant that can be implemented in an iterative manner such that the accuracy is increased with each iteration [19]. Different basis functions can be used for the interpolant. We choose piecewise linear functions with equidistant anchor nodes since they are suitable for local adaptivity. The equidistant anchor nodes, x_j^i, corresponding to level i are defined as [20]

$$m^i = \begin{cases} 1 & \text{if } i = 1, \\ 2^{i-1}+1 & \text{if } i > 1, \end{cases} \tag{4.5}$$

$$x_j^i = \begin{cases} 0.5 & \text{for } j = 1 & \text{if } m^i = 1, \\ \dfrac{j-1}{m^i-1} & \text{for } j = 1,2,\dots,m^i & \text{if } m^i > 1. \end{cases} \tag{4.6}$$

Each node defines a piecewise linear basis function ($a_{x_j^i}^i(x)$) as follows :

$$a_{x^1}^1 = 1 \quad \text{if } i = 1,$$
$$a_{x_j^i}^i(x) = \begin{cases} 1-(m^i-1)|x-x_j^i|, & \text{if } |x-x_j^i| < \dfrac{1}{m^i-1}, \\ 0, & \text{otherwise,} \end{cases}$$
$$\tag{4.7}$$
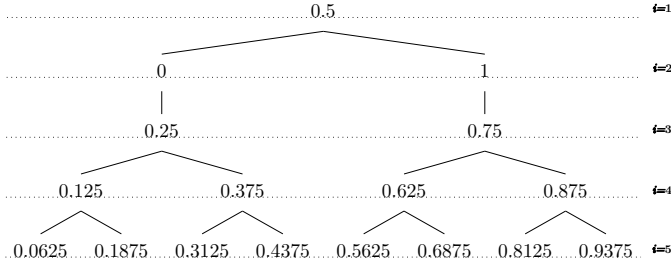
Figure 4.1: Tree structure for the anchor nodes of the basis functions where the depth is assigned a level index i. At each level, nodes are added at half the distances between the nodes in the previous levels.

The unidimensional nodes from Equation 4.6 can be shown in a tree structure (Figure 4.1) where the depth of the tree is assigned a level index i. The algorithm is iterative where at each iteration k, it defines a set of important points $\mathcal{Z}^k$. The criterion for selecting the important points is presented in Section 4.3.2. Once Z^k is identified, the interpolant at iteration k for a function ($c(\mathbf{x})$) depending on a d-dimensional input x can be given by

$$A_{k,d}(c)(\mathbf{x}) = A_{k-1,d}(c)(\mathbf{x}) + \Delta A_{k,d}(c)(\mathbf{x}) , \tag{4.8}$$

with $A_{0,d}(c)(\mathbf{x}) = 0$,

$$\Delta A_{k,d}(c)(\mathbf{x}) = \sum_{n=1}^{m_k^\Delta} w_n^k \Theta_n(\mathbf{x}), \tag{4.9}$$

where m_k^Δ is the cardinality of Z^k, and Θ_n is the d-variate basis function for the point $\mathbf{x}_n \in \mathcal{Z}_k$,

$$\Theta_n(\mathbf{x}) = \prod_{p=1}^{d} a_{x_{n,p}^{i_p}}^{i_p}(x_p), \tag{4.10}$$

where $\mathbf{x}_n$ has support nodes $(x_{n,1}^{i_1}, \dots, x_{n,d}^{i_d})$, and i_p is the level (tree depth) index for the support node $x_{n,p}^{i_p}$. w_n^k is called the *surplus* which is defined as

$$w_n^k = c(\mathbf{x}_n) - A_{k-1,d}(\mathbf{x}_n). \tag{4.11}$$

The union of the important points from all iterations up to k are collected in the set

$$\mathcal{X}^k = \bigcup_{l=1}^{k} \mathcal{Z}^l. \tag{4.12}$$

Because of the tree structure arrangement of the points, each point in the sparse grid ($\mathbf{x} = (x_1, \dots, x_d)$) has ancestry and descendant points. All the descendant points fall within the support of the basis function anchored at that point. The first generation descendants of a point are neighbouring points called *forward points*. The forward points for n points in the set $\mathcal{S} = \{\mathbf{x}_q | q = 1, \dots, n\}$ are defined with an operator $\Psi(\mathcal{S})$ as follows:

$$\Psi(\mathscr{S}) = \{(v_1, \ldots, v_d) | \exists i, q : b(v_i) = x_{q,i} \wedge v_j = x_{q,j}$$
$$\forall j \neq i, \ q \in [1, \ldots, n], \ j, i \in [1, \ldots d]\}, \tag{4.13}$$

where $b(x)$ is a function that returns the parent of a node x from the tree. Likewise, the first generation ancestor points are called backward points and defined with an operator $\Psi^{-1}(\mathscr{S})$ as follows:

$$\Psi^{-1}(\mathscr{S}) = \{(v_1, \ldots, v_d) | \exists i, q : b(x_{q,i}) = v_i \wedge v_j = x_{q,j}$$
$$\forall j \neq i, \ q \in [1, \ldots, n], \ j, i \in [1, \ldots d]\}. \tag{4.14}$$

Finally, an operator $\Gamma(\mathscr{S})$ that return all ancestors for the points in $\mathscr{S}$ can be defined as

$$\Gamma(\mathscr{S}) = \bigcup_{l=1}^{L} \left(\Psi^{-1}\right)^l (\mathscr{S}). \tag{4.15}$$

4.3.2. SELECTING THE IMPORTANT POINTS

The algorithm builds the reduced model in an iterative fashion. At each iteration, we generate a set of trial points to test the model. The model is then updated according to results of this test. Let the generated trial points be stored in the set $\mathscr{T}^k$, where k is the iteration number. The method for generating the trial points will be discussed in Section 4.4. For any point $\mathbf{x}_q \in \mathscr{T}^k$, we can define a local error measure ϵ_q^k in the L_2-norm as follows:

$$\epsilon_q^k = \left\| \mathbf{f}(\mathbf{x}_q) - \sum_{h=1}^{r_k} A_{k,d}(c_h)(\mathbf{x}_q)\mathbf{u}_h \right\|_{L_2}, \tag{4.16}$$

where r_k is the number of POD modes selected at iteration k. The number of POD modes is selected such that the truncation error (Equation 4.4) is below a defined tolerance γ_{tr}. Once ϵ_q^k is computed for all points in $\mathscr{T}^k$, we can select points with an error above a certain threshold to be stored as candidate points. The candidate points are defined as

$$\mathscr{C}^k = \{\mathbf{x}_q \in \mathscr{T}^k | \epsilon_q^k > (\gamma_{\mathrm{int}} \left\| \mathbf{f}(\mathbf{x}_q) \right\|_{L_2} + \zeta_{\mathrm{abs}})\}, \tag{4.17}$$

where γ_{int} is an interpolation threshold and ζ_{abs} is the absolute tolerance, which is introduced to deal with functions of small magnitude.

The candidate points indicate the regions in which the model needs to be enriched. To enrich the model, the ancestor points of these candidate points are first considered because ancestors have wider support. If all ancestors of the candidate points were considered important from previous iterations, that point is taken as important because the error at that point (ϵ_q^k) is above the desired threshold despite including the point's whole ancestry. This is formulated as follows:

$$\mathcal{I}_a^k = \{\mathbf{x}_q \in \mathscr{C}^k | \Gamma(\mathbf{x}_q) \subseteq \mathscr{X}^{k-1}\}. \tag{4.18}$$

On the other hand, if a point x_q in iteration k has an error ϵ_q^k above the threshold but has also an ancestor point y_i which was not included in the important set in the previous

iterations, x_q will not be marked important but its ancestor y_i will be marked important, because it is possible that the error ϵ_q^k was large due to missing the ancestor which has a wider support, that is

$$\mathcal{I}_b^k = \{\mathbf{y}_i \in \Gamma(\mathbf{x}_q) \mid \mathbf{x}_q \in \mathscr{C}^k, \ \Gamma(\mathbf{x}_q) \cap \mathscr{C}^k = \emptyset \ \wedge \ \mathbf{y}_i \notin \mathscr{X}^{k-1} \wedge \Gamma(\mathbf{y}_i) \subseteq \mathscr{X}^{k-1}\}. \qquad (4.19)$$

Then, the complete set of important points at iteration k is formed by Equation 4.18 and Equation 4.19 as

$$\mathcal{I}^k = \mathcal{I}_a^k \cup \mathcal{I}_b^k. \qquad (4.20)$$

4.4. ALGORITHM

Points that are not included in the important set $\mathcal{I}^k$ are added to the inactive set $\mathscr{I}^k$ to be tested in subsequent iterations. The trial set of the next iteration $(k+1)$ is generated as

$$\mathcal{T}^{k+1} = \left\{ \mathbf{x}_q \in \Psi(\mathcal{I}^k) \ \middle| \ \frac{card(\Psi^{-1}(\mathbf{x}_q) \cap \mathscr{X}^k)}{card(\Psi^{-1}(\mathbf{x}_q))} \ge 1 - \mu \right\} \cup \mathscr{I}^k, \qquad (4.21)$$

where $card(.)$ is the cardinality operator, and μ is a greediness parameter which has a value $\in [0,1]$. The trial set $(\mathcal{T}^{k+1})$ is formed by the forward points of $\mathcal{I}^k$. However, some of these forward points are excluded from being evaluated if they have some backward points not considered important in previous iterations. The number of excluded points is tuned with μ. For $\mu = 1$, all points are tested regardless of their ancestry (the algorithm in this case is more exploratory) whereas the algorithm is more efficient for $\mu = 0$ by not testing points that have any backward points not included in $\mathscr{X}^k$.

The trial set $(\mathcal{T}^{k+1})$ is then used to sample both the reduced model and the reference model to compute the error ϵ_q^{k+1}. Then, the important points $(\mathcal{I}^{k+1})$ are identified and added to the snapshot matrix. Each update to the snapshot matrix generates a complete new set of POD modes, which requires recomputing the interpolant $A_{k,d}(c)(\mathbf{x})$ because of its dependence on the POD modes. Specifically, the surpluses $(w_{q,h}^k)$ corresponding to POD mode u_h need to be recomputed with each POD update. The surpluses are just the deviations of the interpolant from the true value. Therefore, an easy way to update the surpluses after each iteration is as follows:

$$\hat{w}_{q,g}^k = \sum_{h=1}^{r_k} w_{q,h}^k <\mathbf{u}_h, \hat{\mathbf{u}}_g> \qquad g = 1, \ldots, r_{k+1}, \qquad (4.22)$$

where $\hat{\mathbf{u}}_g$ is the gth POD mode after updating the snapshot matrix, $\mathbf{u}_h$ is the hth POD mode before updating the snapshot matrix, $w_{q,h}^k$ is the surplus at iteration k corresponding to the point $\mathbf{x}_q \in \mathscr{X}^k$ and POD mode $\mathbf{u}_h$, and $\hat{w}_{q,g}^k$ is the updated surplus corresponding to $\mathbf{x}_q \in \mathscr{X}^k$ and $\hat{\mathbf{u}}_g$. For further reading regarding the adaptive sparse grids technique and the derivation of Equation 4.22, see [15] and the references within. Figure 4.2 summarizes the algorithm.

4.4.1. MULTIPLE OUTPUTS

To deal with models of multiple outputs, we can build a different ROM model for each output, which entails running the adaptive-POD algorithm separately for each output.
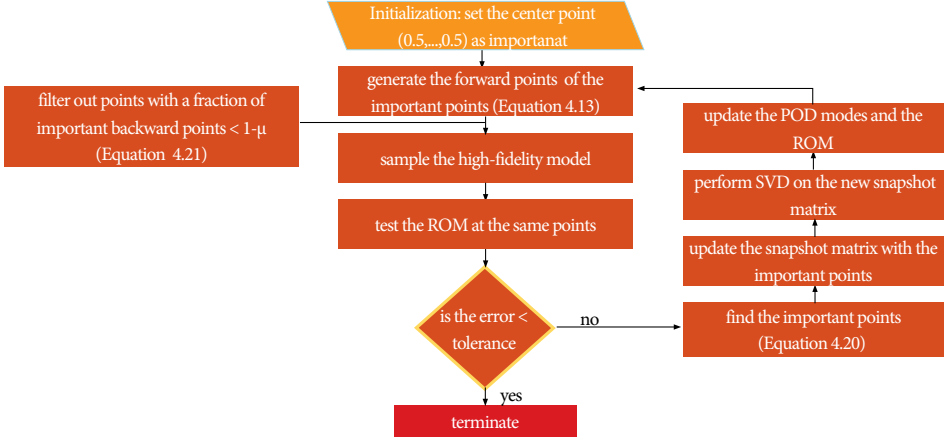
Figure 4.2: A graphical scheme of the adaptive-POD (aPOD) algorithm.

With such an approach, managing the output field data is important to prevent multiple costly evaluations of the same point. This can be achieved by storing all output fields for any full model evaluation in a data bank, which the algorithm is directed to access when a point is required more than once in different output field constructions. With this strategy, the separate runs of the algorithm are performed in series rather than parallel in order to avoid full evaluations of the same point. Another approach is to combine the output fields by stacking them into a composite vector which is then treated as a single output in the snapshot matrix. In this approach, only a single ROM is built to represent all outputs. Since the first approach is a straightforward application of the algorithm, in this section, we show how the second approach is implemented.

Let the outputs be represented by $\mathbf{f}_1(\mathbf{x}), \ldots, \mathbf{f}_o(\mathbf{x})$ where o is the number of output fields. The snapshot matrix is formed by stacking the output fields as

$$[(\mathbf{f}_1^{\mathrm{T}}(\mathbf{x}_1), \ldots, \mathbf{f}_o^{\mathrm{T}}(\mathbf{x}_1))^{\mathrm{T}}, \ldots, (\mathbf{f}_1^{\mathrm{T}}(\mathbf{x}_p), \ldots, \mathbf{f}_o^{\mathrm{T}}(\mathbf{x}_p))^{\mathrm{T}}]. \tag{4.23}$$

We can compute the local error measure (Equation 4.16) in each output $\mathbf{f}_s(\mathbf{x}_q)$ separately

$$\epsilon_{s,q}^k = \left\| \mathbf{f}_s(\mathbf{x}_q) - \sum_{h=1}^{r_k} A_{k,d}(c_h)(\mathbf{x}_q)\mathbf{u}_{s,h} \right\|_{L_2}. \tag{4.24}$$

Different interpolation thresholds and absolute tolerances can be defined for each output. A point x_q is admitted to the candidate set (Equation 4.17) if the corresponding error $\epsilon_{s,q}^k$ at any of the output fields ($s \in [1, \ldots, o]$) is greater than the defined threshold

$$\mathscr{C}^k = \{\mathbf{x}_q \in \mathscr{T}^k | \exists\, s = [1, \ldots, o] : \epsilon_{s,q}^k > \gamma_{\mathrm{int},s} \left\| \mathbf{f}_s(\mathbf{x}_q) \right\|_{L_2} + \zeta_{\mathrm{abs,s}}\}, \tag{4.25}$$

where $\gamma_{\mathrm{int},s}$ and $\zeta_{\mathrm{abs,s}}$ are respectively the interpolation threshold and the absolute tolerance defined for output $\mathbf{f}_s(\mathbf{x})$.

The algorithm is terminated when a global criterion is met. We define this criterion to be

$$\epsilon_{s,q}^k < (\zeta_{\mathrm{rel,s}} \left\| \mathbf{f}_s(\mathbf{x}_q) \right\|_{L_2} + \zeta_{\mathrm{abs,s}}), \qquad \forall \mathbf{x}_q \in \mathcal{T}^k, \ s = 1, \ldots, o, \tag{4.26}$$

where $\zeta_{\mathrm{rel,s}}$ is the global relative tolerance set for output $\mathbf{f}_s(\mathbf{x})$. Note that the multiple-outputs approach can yield a different performance compared to the single-output approach in terms of points selected for evaluations. This is because the POD basis is constructed differently. In the single-output approach, the POD modes are tailored to that output specifically whereas in the multiple-outputs approach the POD modes contain information for all output fields.

4.4.2. CALCULATION OF LOCAL SENSITIVITIES

To compute local sensitivities, we can find an analytical expression for the derivatives of each output with respect to the inputs. The derivative of the ROM model in Equation 4.1 with respect to the gth dimension x_g is

$$\frac{\partial}{\partial x_g} \mathbf{f}(\mathbf{x}) = \frac{\partial}{\partial x_g} \sum_{i=1}^r c_i(\mathbf{x}) \, \mathbf{u}_i. \tag{4.27}$$

The ROM model interpolates $c_i(\mathbf{x})$ with the operator $A_{k,d}(c)(\mathbf{x})$. Using Equation 4.8 and Equation 4.9, Equation 4.27 becomes

$$\frac{\partial}{\partial x_g} \mathbf{f}(\mathbf{x}) = \frac{\partial}{\partial x_g} \sum_{i=1}^r \left(\sum_{n=1}^{m_k^\Delta} w_{n,i}^k \, \Theta_n(\mathbf{x}) \right) \mathbf{u}_i, \tag{4.28}$$

$$= \sum_{i=1}^r \mathbf{u}_i \sum_{n=1}^{m_k^\Delta} w_{n,i}^k \frac{\partial}{\partial x_g} \prod_{p=1}^d a_{x_{n,p}^{i_p}}^{i_p} (x_p), \tag{4.29}$$

$$= \sum_{i=1}^r \mathbf{u}_i \sum_{n=1}^{m_k^\Delta} w_{n,i}^k \left[\frac{\partial}{\partial x_g} a_{x_{n,g}^{i_g}}^{i_g} (x_g) \right] \prod_{p \neq g}^d a_{x_{n,p}^{i_p}}^{i_p} (x_p), \tag{4.30}$$

where the derivative of the unidimensional basis function $\frac{\partial}{\partial x} a_{x_n^i}^i(x)$ (dropping the dependence on the dimension g) is computed as

$$\frac{\partial}{\partial x} a_{x^1}^1 = 0 \quad \text{if } i = 1,$$

$$\frac{\partial}{\partial x} a_{x_n^i}^i(x) = \begin{cases} -(m^i - 1)\dfrac{x - x_n^i}{|x - x_n^i|}, & \text{if } |x - x_n^i| < \dfrac{1}{m^i - 1}, x \neq x_n^i \\ 0, & \text{if } |x - x_n^i| \geq \dfrac{1}{m^i - 1} \\ \text{Not defined}, & \text{if } x = x_n^i, \end{cases} \tag{4.31}$$

It is evident that due to the choice of piecewise linear basis functions, our reduced model is non-differentiable at the anchor nodes x_j^i, which implies that we cannot compute local derivatives at the sampled snapshots, including the nominal point. However, we can compute the local derivatives at two points very close to the nominal values and average them out to have a measure of the local sensitivities at the nominal point.

4.5. Description of the Molten Salt Reactor System

In this work, we construct a reduced order model of a simplified system representative of the main characteristics of the Molten Salt Fast Reactor [16]: strong coupling between neutronics and thermal-hydraulics, fast spectrum, and transport of precursors. The problem was developed as a benchmark for multi-physics tools dedicated to liquid-fuel fast reactors [21, 22].

Figure 4.3 depicts the problem domain: a 2 m side square, 2-dimensional cavity filled with fluoride molten salt at initial temperature of 900 K. The cavity is surrounded by vacuum and insulated; salt cooling is simulated via a heat sink equal to $h(T_{ext} - T)$, where $T_{ext} = 900$ K and h is a volumetric heat transfer coefficient. Zero-velocity boundary conditions are applied to all walls except the top lid, which moves at $v_{lid} = 0.5$ ms^{-1}. The steady-state solution is sought with criticality eigenvalue calculations normalizing the reactor power to P_0. Fluid properties are constant with temperature and uniform in space. Neutronics data are condensed into 6 energy groups and temperature corrected only via density feedback, to avoid the complexities related to Doppler feedback modeling; delayed neutron precursors are divided into 8 families. The flow is laminar and buoyancy effects are modeled via the Boussinesq approximation. Cross sections are corrected according to

$$\Sigma(T) = \Sigma(T_{ref}) \frac{\rho(T)}{\rho(T_{ref})} = \Sigma(T_{ref}) \left(1 - \beta_{th}\left(T - T_{ref}\right)\right) \tag{4.32}$$

where $T_{ref} = 900$ K and $\rho(T_{ref})$ is the density at which macroscopic cross sections are provided. They correspond to the reference values chosen for the Boussinesq approximation. β_{th} is the thermal expansion coefficient. We refer to [21, 22] for a more detailed description of the problem.

An in-house multi-physics tool is used to model the molten salt system. It couples a solver for the incompressible Navier-Stokes equations (DGFlows) with a neutronics code solving the multi-group S_N Boltzmann equation coupled with the transport equations for the delayed neutron precursors (PHANTOM-S_N). Both codes are based on the Discontinuous Galerkin Finite Element method for space discretization. Figure 4.4 displays the structure of the multi-physics tool and the data exchanged between the codes. The average temperature on each element (T_{avg}) is outputted to PHANTOM-S_N, which applies the density feedback on cross sections taken from the library at 900 K, according
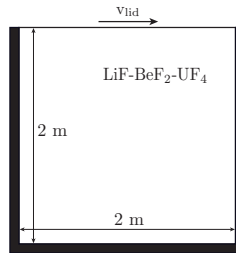

Figure 4.3: Simplified molten salt fast system: square cavity domain. It is insulated, surrounded by vacuum, and filled with molten fluoride salt at initial temperature of 900 K. The top lid moves with velocity $v_{lid} = 0.5$ ms^{-1}.
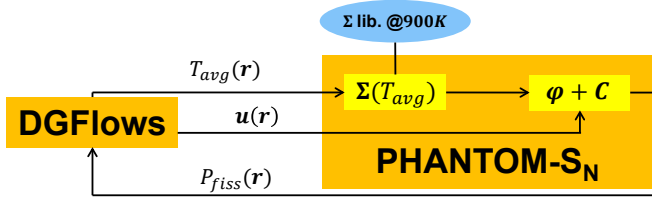
Figure 4.4: Computational scheme of the multi-physics tool representing the high-fidelity model. The CFD code, DGFlows, exchanges data with the radiation transport code, PHANTOM-S$_N$, at each iteration due to the coupling between the physics characterizing the molten salt nuclear system.
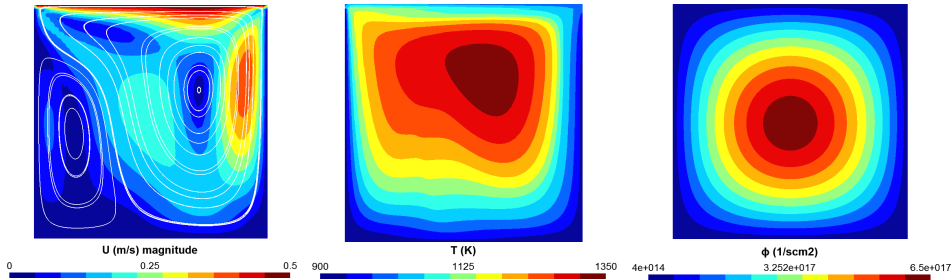
Figure 4.5: Velocity magnitude, temperature, and total flux fields representing the steady state solution of the simplified MSFR problem for nominal values of the input parameters.

to Equation 4.32. Then, the neutronics problem is solved taking the velocity field (**u**) from DGFlows as another input for the delayed neutron precursors equation. Finally, the fission power density (P_{fiss}) is transferred to the CFD code. The steady state solution is sought by iterating DGFlows and PHANTOM-S$_N$ until convergence. More details on the multi-physics tool can be found in [17].

Simulations of the molten salt system were performed choosing a 50 × 50 uniform structured mesh, with a second-order polynomial discretization for the velocity and a first-order one for all the other quantities. An S_2 discretization was chosen for the angular variable. Figure 4.5 shows the steady state fields (velocity magnitude, temperature, and total flux) obtained for the nominal values of the input parameters. The nominal multiplication factor in this configuration is $k_{eff} = 0.99295$. The upper bounds for each of the six energy groups are shown in Table 4.1 along with the space averaged flux (Φ_{avg}) for each group in the nominal case.

4.6. CONSTRUCTION OF THE REDUCED-ORDER MODEL

A ROM model was built for the molten salt system by considering 27 input parameters. We assumed a uniform distribution for all of them. The parameters and the corresponding percentage variation from the nominal values are summarized in Table 4.2, where P_0 is the initial power, β_{th} is the thermal expansion coefficient, $\Sigma_{f,g}$ is the fission cross section for group g, β_i is the delayed neutron fraction for precursors family i, λ_i is the decay constant for precursors family i, v_{lid} is the lid velocity, ν is the viscosity, and h is the heat

Table 4.1: Average group flux in the nominal case along with the upper energy bound for each group.

Energy group	1	2	3	4	5	6
Upper bound [keV]	20000	2231	497.9	2.479	5.531	0.7485
Φ_{avg} [cm^{-2}s^{-1}] $\times 10^{16}$	1.22	4.92	9.19	5.94	4.74	1.43

Table 4.2: Nominal values and the corresponding variation for the considered parameters.

Parameter	Nominal value	Percentage variation	Parameter	Nominal value	Percentage variation
P_0 [W]	10^9	±20%	β_7	6.05×10^{-4}	±10%
β_{th} [K^{-1}]	2×10^{-4}	±10%	β_8	1.66×10^{-4}	±10%
$\Sigma_{f,1}$ [cm^{-1}]	1.11×10^{-3}	±10%	λ_1 [s^{-1}]	1.25×10^{-2}	±10%
$\Sigma_{f,2}$ [cm^{-1}]	1.08×10^{-3}	±10%	λ_2 [s^{-1}]	2.83×10^{-2}	±10%
$\Sigma_{f,3}$ [cm^{-1}]	1.52×10^{-3}	±10%	λ_3 [s^{-1}]	4.25×10^{-2}	±10%
$\Sigma_{f,4}$ [cm^{-1}]	2.58×10^{-3}	±10%	λ_4 [s^{-1}]	1.33×10^{-1}	±10%
$\Sigma_{f,5}$ [cm^{-1}]	5.36×10^{-3}	±10%	λ_5 [s^{-1}]	2.92×10^{-1}	±10%
$\Sigma_{f,6}$ [cm^{-1}]	1.44×10^{-2}	±10%	λ_6 [s^{-1}]	6.66×10^{-1}	±10%
β_1	2.33×10^{-4}	±10%	λ_7 [s^{-1}]	1.63	±10%
β_2	1.03×10^{-3}	±10%	λ_8 [s^{-1}]	3.55	±10%
β_3	6.81×10^{-4}	±10%	v$_{lid}$ [m/s]	0.5	±20%
β_4	1.37×10^{-3}	±10%	ν [m^2/s]	0.025	±10%
β_5	2.14×10^{-3}	±10%	h [W/m^{2}K]	10^6	±20%
β_6	6.41×10^{-4}	±10%			

transfer coefficient. Since we aim at using the reduced model for uncertainty and sensitivity analysis, we assigned a variation of ±10% for parameters with typical experimental uncertainties whereas we vary design parameters (P_0, v$_{lid}$ and h) by ±20%. Our interest is in the effect of these parameters on the spatial distribution of the total flux $\Phi(\mathbf{r})$, the temperature $T(\mathbf{r})$, and the value of the effective multiplication factor k_{eff}. Therefore, the reference model has 27 inputs and returns a value for the k_{eff} and two field vectors each of length 7500 corresponding to the coefficients of the discontinuous Galerkin expansion for the total flux $\boldsymbol{\Phi}$ and temperature $\boldsymbol{T}$. In this work, we compare the stacking of the outputs approach described in Subsection 4.4.1 with the single-output approach. For the multiple-outputs approach, the snapshot matrix for the outputs evaluated at points $[\mathbf{x}_1, \ldots, \mathbf{x}_p]$ is computed as $[(\boldsymbol{\Phi}_1^T, \boldsymbol{T}_1^T, k_{eff,1})^T, \ldots, (\boldsymbol{\Phi}_p^T, \boldsymbol{T}_p^T, k_{eff,p})^T]$.

The global relative tolerances ζ_{rel} for $\boldsymbol{\Phi}$ and $\boldsymbol{T}$ were set to be 10^{-2}, which means we require the error in the L_2 norm for these fields to be less than 1%. For k_{eff}, we require the error to be less than 50 pcm, so we set ζ_{rel} for k_{eff} to be 50×10^{-5}. The interpolation threshold (γ_{int}) was chosen to be one order of magnitude less than the set relative tolerances. Therefore, γ_{int} was 10^{-3} for both $\boldsymbol{\Phi}$ and $\boldsymbol{T}$ and was set to be 5×10^{-5} for k_{eff}.

We first built a reduced model using a greediness value $\mu = 1$. For the multiple-outputs approach, the algorithm required 4495 reference model evaluation to converge.

However, only 142 points were included in the important set. The small number of selected important points is an indication of oversampling. The algorithm was then run again with $\mu = 0$. In this case, the algorithm sampled 472 points with 105 important points included in the snapshot matrix, which is a reduction by about a factor of 10 in the number of evaluations compared with the the case of $\mu = 1$. Each reference model evaluation takes about 1.5 hours to run (performed on a Linux cluster using 1 CPU operating at 2.60 GHz). Therefore, this reduction in number of evaluations is massive in computational time. In order to test the model, 1000 Latin Hypercube Sampling (LHS) points were generated. LHS is a method to generate unbiased random points in higher dimensional spaces by partitioning the hypercube first. Then, drawing one sample from each partition. These generated points were not part of the snapshot matrix. Note that the reduced model was trained only on the important set. The rest of the model evaluations served as trial points but were not included in the snapshot matrix. In machine learning terminology, the important set is the training set and the rest of the evaluations served the function of the validation set [23]. Therefore, the generated 1000 unbiased random points in the test set represent 10 times more testing points than training points. Running the reduced model on the 1000 testing points needed only about one second on a personal computer.

Table 4.3 summarizes the maximum L_2 norm error found for each output. It is evident that all tested points resulted in errors well below the set tolerances. We also compare the results of the single-output approach to the multiple-outputs approach in the same table. While both approaches satisfied the required tolerances, the number of full model evaluations required in the offline stage was different. The single-output approach required fewer evaluations compared to the multiple-outputs approach. This is due to the fact that the POD modes in the single-output approach are tailored to that output field. The algorithm in this case, samples points to construct a specific reduced model satisfying the desired tolerance for that output. In the multiple-outputs approach, on the other hand, the algorithm uses POD modes containing information for all output fields, which require more points to satisfy the desired tolerances for every output fields. However, because the reduced model is enriched with every additional sampling point, the multiple-outputs model has a slightly less error in the online phase compared to the single-output approach.

Figure 4.6 shows the distribution of the L_2 norm error for the tested 1000 random points for each output in the reduced model of the multiple-outputs approach and $\mu = 0$. A comparison between the temperature distributions of the reduced model and the reference full order model at the point that resulted in the maximum error is shown in Figure 4.7. The L_2 norm error for this case was 0.2% while the maximum absolute difference locally was 13.9 K, which is about 1% of the maximum local temperature (about 1482.6 K). Both cases of $\mu = 1$ and $\mu = 0$ converged with 3 iterations (k=3). To highlight the cost effectiveness of the present adaptive approach, for such 27-dimensional problem, the classical (non-adaptive) sparse grid approach would require 27829 points after 3 iterations, which is extremely expensive to run.

Table 4.4 summarizes the number of unique nodes per dimension, which was found to be the same for both the single and multiple-outputs approaches. This number is indicative of the linearity/non-linearity of the reference model. During the construction stage, the algorithm captures the degree of linearity of the output of the reference model

Table 4.3: Maximum L_2 error in each output with respect to the reference model after testing the reduced model on 1000 random points. The total number of full model evaluations in the offline stage for each ROM construction is also shown.

		Φ	T	k_{eff}	Total number of evaluations
Multiple outputs	$\mu = 1$	0.18%	0.14%	23 pcm	4495
	$\mu = 0$	0.22%	0.20%	22 pcm	472
Single output	$\mu = 1$	0.35%	0.14%	23 pcm	3548
	$\mu = 0$	0.35%	0.25%	33 pcm	348

with respect to each dimension within the defined range. A value of 3 means that the algorithm considered that dimension to be constant because after building a constant interpolant at the root 0.5, the error in the model was found to be within the defined tolerances at the children points {0, 1}. The algorithm then stopped further refinements along that dimension. A value of 5 indicates that the model is piecewise linear in the segments (0,0.5) and (0.5,1) with respect to that dimension because the refinement is stopped after testing the piecewise linear interpolant using the first 3 points {0.5,0,1} at the children {0.25,0.75}. A value higher than 5 indicates that the model is nonlinear along that dimension.

It is evident from the number of unique nodes that the algorithm found the outputs of the model to be constant (within the set tolerances) with respect to β_i and λ_i, which means varying these parameters within the 10% range does not significantly affect the defined outputs. Additionally, the model was found to be piecewise linear with respect to the power, velocity, thermal expansion coefficient, viscosity, and the fission cross section for the groups 1–4. However, for the lowest energy groups (group 5 and 6), the model was nonlinear. This can be explained by the fact that the flux distributions for all groups were not changing significantly due to the homogeneity of the changes to the system. In addition, the group fluxes were found to have the same order of magnitude as shown in Table 4.1 for the nominal case. However, the nominal values of the fission cross section for $\Sigma_{f,5}$ and $\Sigma_{f,6}$ are higher compared to the other fast groups, which weigh more in the
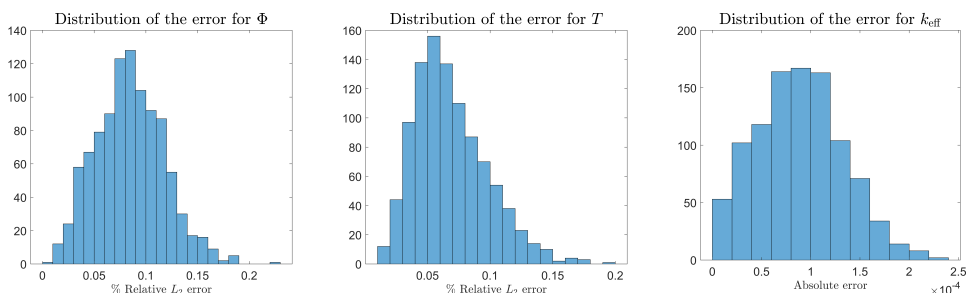
Figure 4.6: Histogram showing the error in each of the outputs resulting from testing the reduced model on 1000 random points.
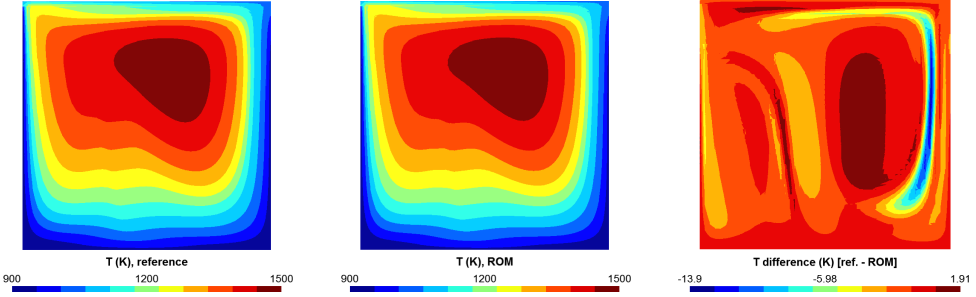
Figure 4.7: Temperature distribution at the point of maximum error showing the reference model (left), the ROM model (center), and the distribution of the difference (right). Note the change of the colour bar scale in the difference plot (right).

Table 4.4: Number of unique nodes per dimension.

Parameter	number of unique nodes	Parameter	number of unique nodes
P_0	5	β_7	3
β_{th}	5	β_8	3
$\Sigma_{f,1}$	5	λ_1	3
$\Sigma_{f,2}$	5	λ_2	3
$\Sigma_{f,3}$	5	λ_3	3
$\Sigma_{f,4}$	5	λ_4	3
$\Sigma_{f,5}$	9	λ_5	3
$\Sigma_{f,6}$	9	λ_6	3
β_1	3	λ_7	3
β_2	3	λ_8	3
β_3	3	v_{lid}	5
β_4	3	ν	5
β_5	3	h	9
β_6	3		

calculation of k_{eff}. By examining the cause for the additional unique points along $\Sigma_{f,5}$ and $\Sigma_{f,6}$, we found that they were triggered purely by k_{eff} and not by $\boldsymbol{\Phi}$ or $\boldsymbol{T}$. The model was also nonlinear in the heat transfer coefficient. The negligible effect of β_i and λ_i explains the reason for the massive reduction in number of evaluations with the setting $\mu = 0$. The algorithm in this case recognized that β_i and λ_i have no effect within the defined range and stopped sampling points along these dimensions. Since β_i and λ_i amount to 16 out of the 27 dimensions, the reduction in number of points was massive.

4.7. UNCERTAINTY AND SENSITIVITY ANALYSIS

In this section, we demonstrate the potential of the built ROM model in an application of uncertainty quantification and sensitivity analysis. We study the effect of the selected input parameters on the maximum temperature and the multiplication factor k_{eff}. The resulting ROM can be sampled cheaply at any point within the specified range. The ROM model from the multiple-outputs approach and $\mu = 0$ is employed for the study in this section. However, we do not expect differences in the results if any of the other 3 ROM models developed in Section 4.6 were used instead. We use Latin Hypercube Sampling to sample the reduced model with 100,000 random points. The density histograms approximating the Probability Distribution Function (PDF) are shown in Figure 4.8. For comparison, the densities resulting from running the reference model on the 1000 testing points are also shown in the figure. The density histogram shows a distribution close to a normal distribution, which can be explained by the fact that all input parameters are assumed to have uniform distribution and the model is linear or almost linear in these parameters. Therefore, the sum of these uniform distribution approaches the normal distribution. The normal probability plot in Figure 4.9 confirms that the distribution is normal within the middle range while the deviation from the normal is seen at the tails of the distribution. The mean of the maximum temperature was found to be at 1336.5 K with standard deviation equal to 61.1 while the mean of k_{eff} was 0.99229 with standard deviation equal to 0.016.

Local and global sensitivity analyses were also performed using the built ROM. For the local sensitivities, Table 4.5 presents the averaged derivatives computed from several points within a distance of 10^{-14} (measured in the unit hypercube $[0, 1]^d$) from the input's nominal values. In order to provide a better comparison of the effect of the parameters, the computed derivatives in the table are normalized by the ratio $R_0 / x_{p,0}$, where R_0 is the desired response (maximum temperature or k_{eff}) computed at the nominal values of the input parameters $x_{p,0}$.

The results show that the maximum temperature is mainly affected by the initial power P_0 and the heat transfer coefficient h. This is expected because these two parameters directly control the amount of energy present in the system. Higher initial power increases the amount of energy in the system which directly raises the temperature. The heat transfer coefficient, on the other hand, is negatively correlated with T_{max} because lower h decreases the amount of energy being extracted from the system causing the temperature to rise.

The thermal expansion coefficient is related to the natural convection phenomenon. Forced and natural convection play a competing role in terms of mixing of the salt in the cavity. There are two vortexes in the cavity as shown by the streamlines in Figure 4.5
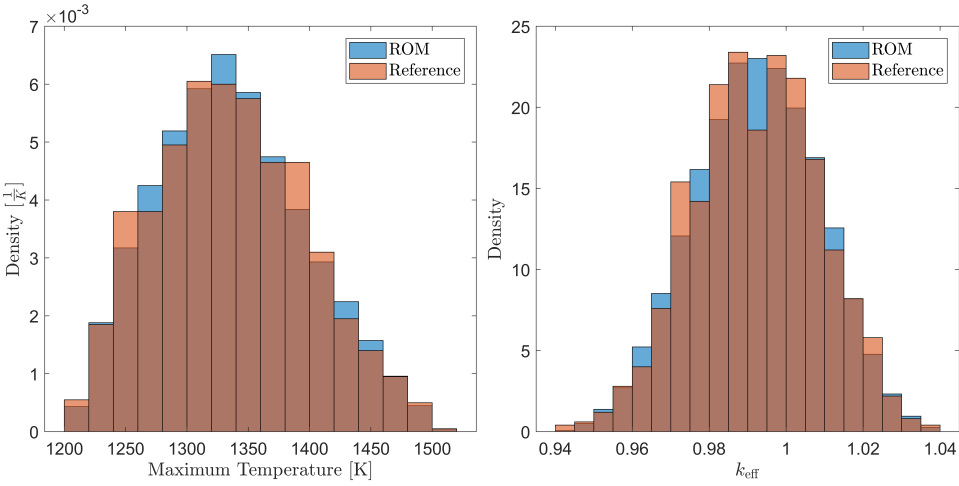
Figure 4.8: Density histograms of the maximum temperature (left) and the multiplication factor k_{eff} (right) by sampling the reduced model with 100,000 points. The distributions of same variables from sampling the reference model with the 1000 testing points are also shown. Note that the histogram is normalized such that the sum of the areas of the bars equals to 1.

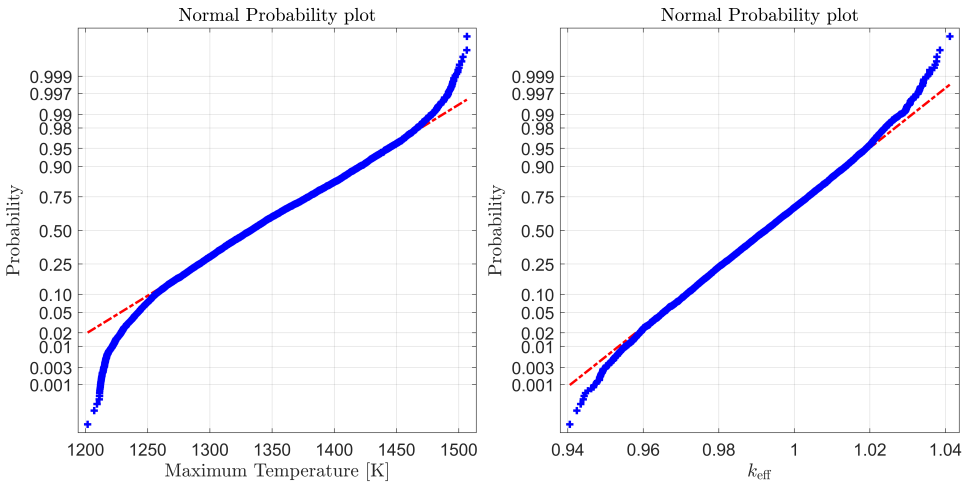

Figure 4.9: Normal probability plots for the maximum temperature (left) and the multiplication factor k_{eff} (right) showing the distribution to be normal within the middle parts but deviating from the normal distribution at the tales.

Table 4.5: Normalized local sensitivities of the maximum temperature (T_{max}) and k_{eff} with respect to the parameters around the nominal values. The derivatives are normalized by the ratio of output nominal value ($T_{max,0}$ and $k_{eff,0}$) to the nominal values of the input parameters $x_{p,0}$.

x_p	$\dfrac{\partial T_{max}}{\partial x_p}\bigg/\dfrac{T_{max,0}}{x_{p,0}}$	$\dfrac{\partial k_{eff}}{\partial x_p}\bigg/\dfrac{k_{eff,0}}{x_{p,0}}$	x_p	$\dfrac{\partial T_{max}}{\partial x_p}\bigg/\dfrac{T_{max,0}}{x_{p,0}}$	$\dfrac{\partial k_{eff}}{\partial x_p}\bigg/\dfrac{k_{eff,0}}{x_{p,0}}$
P_0	0.289	-0.012	β_7	0	0
β_{th}	-0.036	-0.012	β_8	0	0
$\Sigma_{f,1}$	-2×10^{-5}	0.012	λ_1	0	0
$\Sigma_{f,2}$	-8×10^{-5}	0.041	λ_2	0	0
$\Sigma_{f,3}$	-6×10^{-5}	0.101	λ_3	0	0
$\Sigma_{f,4}$	-2×10^{-5}	0.11	λ_4	0	0
$\Sigma_{f,5}$	3×10^{-5}	0.182	λ_5	0	0
$\Sigma_{f,6}$	9×10^{-5}	0.145	λ_6	0	0
β_1	0	0	λ_7	0	0
β_2	0	0	λ_8	0	0
β_3	0	0	v_{lid}	0.0003	10^{-4}
β_4	0	0	ν	0.023	-10^{-4}
β_5	0	0	h	-0.258	0.011
β_6	0	0			

(left) for the nominal case. When forced convection increases, the larger vortex grows causing the vortex centre to move towards the cavity centre. In this case, salt in the central region of the cavity would always circulate around the centre where the fission power is maximum. On the other hand, when natural convection increases, the smaller vortex in the bottom left corner becomes larger causing the salt to pass through the centre then transported close to the boundaries of the cavity where the thermal energy is minimum. Hence, in the range of variations considered in this work, natural convection tends to redistribute the heat in the cavity, whereas forced convection has the opposite effect. Higher β_{th} causes natural convection to be more prevalent over forced convection. This causes the temperature to be more uniform. For this reason, β_{th} is negatively correlated with T_{max}. The viscosity, on the other hand, has the opposite effect. Increasing the viscosity reduces the mixing of the liquid, which creates more concentrated hot spots that increase the maximum temperature. The lid velocity is also positively correlated with the maximum temperature because it increases the forced convection. However, this correlation is shown to be weak because the range in which the velocity changes ($\pm 20\%$) is very small. The fission cross sections have negligible effect on T_{max}. The delayed neutron fractions and the precursors decay constants have zero derivatives because our reduced model assumes them to be constants at any point.

The multiplication factor is mainly affected by the fission cross sections as expected. The fission cross sections of the two lowest energy groups are the most important. This is because of their higher weight (higher nominal values with similar flux magnitudes compared to the fast groups) in computing k_{eff}. Although $\Sigma_{f,5}$ has a nominal value of about half $\Sigma_{f,6}$, the Sobol index of $\Sigma_{f,5}$ is about 50% higher than $\Sigma_{f,6}$. This can be explained by the higher flux magnitude of group 5 compared to group 6 as can be seen
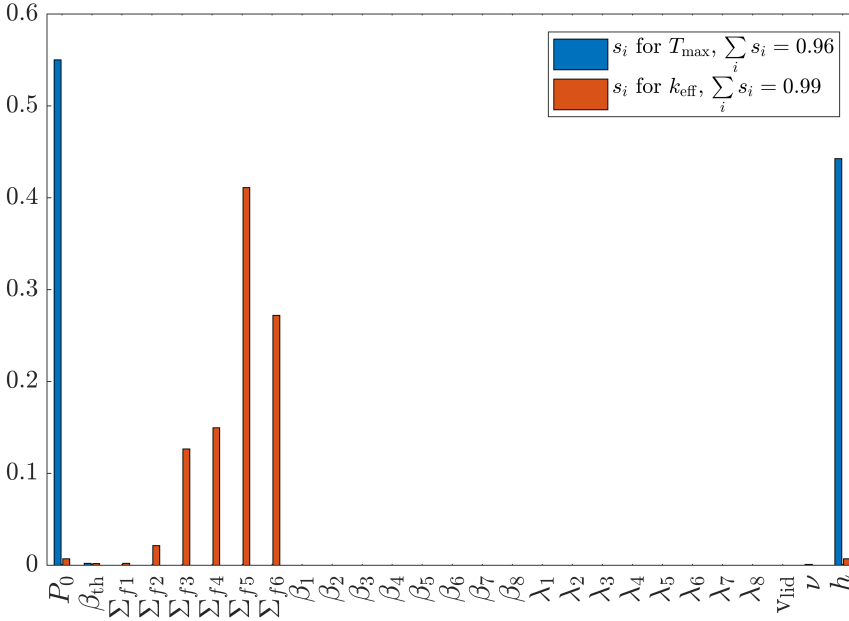
Figure 4.10: First order Sobol indices showing the first order sensitivities of $T_{\max}$ and k_{eff} to each input parameter. The sum of the first order sensitivities for each output is also shown in the legend.

from the average flux value reported in Table 4.1 for the nominal case. The thermal expansion coefficient is negatively correlated with k_{eff} because by increasing β_{th}, the liquid is mixed more, which in turn causes more precursors to move from regions of higher importance to regions of lower importance near the boundaries. The initial power is negatively correlated with k_{eff} due to the negative temperature feedback coefficient of the system. For the same reason, the heat transfer coefficient is positively correlated with k_{eff}. The lid velocity and viscosity have negligible effect on the multiplication factor.

For the global sensitivities, we computed the first order Sobol indices using quasi Monte Carlo method with Sobol sequence sampling [24]. We selected the size of our sampling matrices to be 10^5, which generates 2 matrices each of dimension $10^5 \times 27$. The first order Sobol indices were then estimated using the estimators recommended by Saltelli et al. [25]. The computed indices for both $T_{\max}$ and k_{eff} are shown in Figure 4.10. The Sobol indices show agreement with the conclusions of the local sensitivities. The maximum temperature is predominantly sensitive to P_0 and h while β_{th} and ν have a slight effect on $T_{\max}$. The multiplication factor, on the other hand, is mainly sensitive to the fission cross sections with the lowest energy groups having the most importance. P_0 and h have a reduced effect while β_{th} has a minimal effect on k_{eff}. The agreement between the local and global sensitivities show that the system is only weakly nonlinear. Additionally, the sum of the computed first order Sobol indices was found to be very close to one, which indicates that second and higher order interactions between the parameters are almost negligible. This confirms the weak nonlinearity of the model.

In total, 3×10^6 model evaluations were performed to complete the uncertainty and

sensitivity analysis study. The time to perform these simulations using the reduced model was about 45 minutes on a personal computer, which is about half the time to perform a single simulation of the full model on the computer cluster. Using 472 snapshots computed in the offline phase, we obtained a gain of about a factor of 5×10^6 in the online computations with respect to the reference model. This demonstrates the advantage of ROM for such applications.

4.8. CONCLUSIONS

The developed ROM algorithm (aPOD) based on POD and the adaptive sparse grids method was applied to a coupled model of a test case for the Molten Salt Fast Reactor. 27 input parameters were chosen to model their effect on the distribution of the flux and temperature, and the value of the multiplication factor. In a completely nonintrusive manner, aPOD was able to build a representative (1% accurate) ROM model with 4495 model evaluations. This number was effectively reduced by a factor of 10 with the setting $\mu = 0$. This great reduction was successfully achieved due to the ability of the algorithm to automatically recognize that the 16 dimensions corresponding to β_i and λ_i have no significant effect within the defined range. It was also observed that the initial power, thermal expansion coefficient, fission cross section of the fast 4 groups, lid velocity, and viscosity all have piecewise linear effect on the outputs. On the other hand, the fission cross section for the 2 lowest energy groups and the heat transfer coefficient have slight nonlinear effect. As a test of the model, 1000 Latin Hypercube Sampling points were tested and compared with respect to the reference model. The errors were found to be well within the defined tolerances for all outputs.

The multiple-outputs approach was found to require more sampling points to satisfy the desired tolerances compared to a single separate run for each output. This can be explained by the fact that with the single-output ROM model, the POD modes are tailored to that output field and the algorithm only needs to sample points to satisfy the tolerance for that field. The multiple-outputs approach requires the composite POD modes to represent all output fields, which leads to more sampling points to satisfy the tolerances. However, because of the additional sampling in the construction of the reduced model, the error was found to be lower for the multiple approach compared to the single approach.

For an application of uncertainty and sensitivity analysis, we studied the effect of the 27 input parameters on the maximum temperature and the multiplication factor. The density histograms showed a normal distributions of these variables, which can be explained by the uniform distribution assumption of the selected parameters and the weak nonlinearity of the model with respect to the input parameters within the defined ranges. The maximum temperature was shown to be sensitive to the initial power and the heat transfer coefficient while the multiplication factor was mainly sensitive to the fission cross sections as expected. The uncertainty and sensitivity study was performed using a total of 3 million random points, which were completed in about half the time to run a single simulation of the reference model. The nonintrusive approach of the algorithm provides great potential for studies of complex coupled nuclear systems such as the molten salt reactor, particularity in applications of uncertainty quantification, sensitivity analysis, fuel management, design optimization, and control.

REFERENCES

[1] 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 **141**, 107321 (2020).

[2] A. C. Antoulas, D. C. Sorensen, and S. Gugercin, *A survey of model reduction methods for large-scale systems,* Contemporary Mathematics **280**, 193 (2001).

[3] P. Benner, S. Gugercin, and K. Willcox, *A Survey of Projection-Based Model Reduction Methods for Parametric Dynamical Systems,* SIAM review **57**, 483 (2015).

[4] A. G. Buchan, C. C. Pain, F. Fang, and I. M. Navon, *A POD reduced-order model for eigenvalue problems with application to reactor physics,* International Journal for Numerical Methods in Engineering **95**, 1011 (2013).

[5] 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 Rreactor Spatial Kinetics,* Annals of Nuclear Energy **71**, 217 (2014).

[6] 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 **46**, 263 (2017).

[7] R. Manthey, A. Knospe, C. Lange, D. Hennig, and A. Hurtado, *Reduced order modeling of a natural circulation system by proper orthogonal decomposition,* Progress in Nuclear Energy **114**, 191 (2019).

[8] P. German and J. C. Ragusa, *Reduced-order modeling of parameterized multi-group diffusion k-eigenvalue problems,* Annals of Nuclear Energy **134**, 144 (2019).

[9] H. V. Ly and H. T. Tran, *Modeling and Control of Physical Processes using Proper Orthogonal Decomposition,* Mathematical and Computer Modelling **33**, 223 (2001).

[10] V. Buljak, *Inverse Analyses with Model Reduction: Proper Orthogonal Decomposition in Structural Mechanics* (Springer, Berlin, 2012).

[11] J. Hesthaven and S. Ubbiali, *Non-intrusive reduced order modeling of nonlinear problems using neural networks,* Journal of Computational Physics **363**, 55 (2018).

[12] N. Nguyen and J. Peraire, *Gaussian functional regression for output prediction: Model assimilation and experimental design,* Journal of Computational Physics **309**, 52 (2016).

[13] C. Audouze, F. D. Vuyst, and P. B. Nair, *Reduced-order modeling of parameterized PDEs using time-space-parameter principal component analysis,* International Journal for Numerical Methods in Engineering **80**, 1025 (2009).

[14] H.-J. Bungartz and M. Griebel, *Sparse grids,* Acta Numerica **13**, 147 (2004).

4

[15] F. Alsayyari, Z. Perkó, D. Lathouwers, and J. L. Kloosterman, *A nonintrusive redu-ced order modelling approach using Proper Orthogonal Decomposition and locally adaptive sparse grids*, Journal of Computational Physics **399**, 108912 (2019).

[16] M. Allibert, M. Aufiero, M. Brovchenko, S. Delpech, V. Ghetta, D. Heuer, A. Laureau, and E. Merle-Lucotte, *Molten salt fast reactors*, in *Handbook of Generation IV Nuclear Reactors*, edited by I. L. Pioro (Woodhead Publishing, 2016) pp. 157–188.

[17] M. Tiberga, D. Lathouwers, and J. L. Kloosterman, *A discontinuous Galerkin FEM multiphysics solver for the Molten Salt Fast Reactor*, in *International Conference on Mathematics and Computational Methods applied to Nuclear Science and Engineer-ing (M&C 2019)* (Portland, OR, USA, 2019).

[18] M. Griebel, *Adaptive sparse grid multilevel methods for elliptic PDEs based on finite differences*, Computing **61**, 151 (1998).

[19] V. Barthelmann, E. Novak, and K. Ritter, *High Dimensional Polynomial Interpolation on Sparse Grids*, Advances in Computational Mathematics **12**, 273 (2000).

[20] A. Klimke, *Uncertainty Modeling using Fuzzy Arithmetic and Sparse Grids*, Ph.D. thesis, Universität Stuttgart, Stuttgart (2006).

[21] M. Aufiero and P. Rubiolo, *Testing and Verification of Multiphysics Tools for Fast-Spectrum MSRs: The CNRS Benchmark. ,* in *Transactions of the 2018 ANS Annual Meeting* (Phialdelphia, PA, USA, 2018).

[22] A. Laureau, *Développement de modèles neutroniques pour le couplage thermohydrau-lique du MSFR et le calcul de paramètres cinétiques effectifs*, Ph.D. thesis, Grenoble Alpes University, France (2015).

[23] B. D. Ripley, *Pattern Recognition and Neural Networks* (Cambridge University Press, 1996).

[24] I. M. Sobol, *Global sensitivity indices for nonlinear mathematical models and their Monte Carlo estimates*, Mathematics and Computers in Simulation **55**, 271 (2001).

[25] A. Saltelli, P. Annoni, I. Azzini, F. Campolongo, M. Ratto, and S. Tarantola, *Variance based sensitivity analysis of model output. design and estimator for the total sensitivity index*, Computer Physics Communications **181**, 259 (2010).

5

GENERALIZING THE ADAPTIVE ALGORITHM TO DYNAMICAL SYSTEMS

We present an approach to build a reduced-order model for nonlinear, time-dependent, parametrized partial differential equations in a nonintrusive manner. The approach is based on combining proper orthogonal decomposition (POD) with a Smolyak hierarchical interpolation model for the POD coefficients. The sampling of the high-fidelity model to generate the snapshots is based on a locally adaptive sparse grid method. The novelty of the work is in the adaptive sampling of time, which is treated as an additional parameter. The goal is to have a robust and efficient sampling strategy that minimizes the risk of overlooking important dynamics of the system while disregarding snapshots at times when the dynamics are not contributing to the construction of the reduced model. The developed algorithm was tested on three numerical tests. The first was an advection problem parametrized with a five-dimensional space. The second was a lid-driven cavity test, and the last was a neutron diffusion problem in a subcritical nuclear reactor with 11 parameters. In all tests, the algorithm was able to detect and include more snapshots in important transient windows, which produced accurate and efficient representations of the high-fidelity models.

5.1. INTRODUCTION

I N science and engineering applications, dynamic models can be described by time-dependent mathematical models. Often, these models are written as parametrized partial (integro-) differential equations (PDE). Applications such as uncertainty and sensitivity analysis and design optimization require solving the equations repeatedly for different values of the PDE parameters. For complex, large-scale problems, applications of repeated evaluations demand excessive computational power and memory resources. In such cases, *model reduction* techniques are used to overcome the computational burden. Model reduction methods aim to replace the high-fidelity model with an efficient, low-dimensional reduced-order model (ROM) capturing the main dynamics of the system with a controlled level of accuracy. Model reduction methods can be classified into intrusive and nonintrusive methods. Intrusive methods are mainly projection-based, where the high-dimensional model is approximated by projecting the original equations onto a reduced subspace. For an overview of different projection-based approaches, see [2, 3].

Intrusive approaches require access to the operator of the high-fidelity model, which can be limiting for applications where the numerical solver is closed-source or legacy, coupled multi-physics code. In these cases, nonintrusive methods are applicable where the reduced model is built using only data generated from the high-fidelity model. For this reason, they are also called data-driven methods. A class of nonintrusive methods aims at recovering part of the problem's physical structure by inferring an assumed operator from the data. In the Loewner framework [4], a reduced model is built by interpolating measurements of the transfer function in the frequency domain. This approach was extended to construct a reduced model from time-domain data [5]. However, reduced models in the Loewner framework are applied to linear, time-invariant systems (or linear PDEs). For non-parametrized PDEs, dynamic mode decomposition (DMD) [6] learns a linear operator by fitting a sequence of time snapshots data in an optimal least square sense. However, this approach cannot be directly applied to parametrized problems.

A different line of research attempts to construct a nonintrusive reduced model for general (nonlinear) parametrized PDEs without an operator inference. The PDE solver is considered as a black-box. This class of methods is closer to machine learning techniques. They use generated data to fit a model mapping a defined input space to the desired output space. The main difference is that classical machine learning methods are trained on an abundance of data while, in both numerical and experimental computational science and engineering applications, data are typically expensive to generate. Therefore, an important challenge to overcome for nonintrusive ROM methods is to build an accurate model using minimum data size. One effective black-box ROM method adapts the projection-based proper orthogonal decomposition (POD) method to be a nonintrusive approach. This nonintrusive version starts in a similar way to the projection-based version by constructing a reduced basis space. However, instead of projecting the high-fidelity model equations onto the reduced basis space to solve for the POD coefficients, data-fit surrogate models for the POD expansion coefficients are used. Different routes can be followed to construct the models for the expansion coefficients. One can interpolate with splines [7] or, more commonly, use radial basis function (RBF) [8, 9]. Additionally, neural networks can be used to learn a surrogate model for the

coefficients [10]. Gaussian process regression (or kriging) is another option to build the surrogate model [11, 12]. We have presented an approach using locally adaptive sparse grid and hierarchical interpolation [13], which was then applied to perform analysis of the uncertainties in a coupled multi-physics model of a nuclear reactor system [14].

Most of the work on nonintrusive ROM methods has been developed either for parametrized time-independent (steady-state solutions) problems or time-dependent non-parametrized problems. Generalizing a ROM method to address both spatiotemporal discretization as well as the parameter space is not trivial. As a direct approach, one could build a separate ROM model for each time instance of interest using any of the (steady-state) nonintrusive ROM methods. However, such an approach is computationally unfeasible for the entire discretized time series. The challenge is even more complicated if the boundary and initial conditions are also parameter and time-dependent or if the parameter space is of a high dimension. Audouze et al. (2013) [9] suggested a nonintrusive ROM approach for time-dependent PDE problems using a two-level RBF-POD technique. This approach constructs two reduced basis spaces, one for spatial basis and a second for temporal basis. The authors use a coarse grid discretization of the spatial coordinates, time, and parameter spaces to sample the high fidelity model and generate the snapshots for the POD. Chen et al. (2018) [15] extended this work to include adaptive sampling for the parameter space using an RBF error estimator based on the distance between the RBF coefficients. The adaptivity in this approach cannot easily be extended to higher dimensional parameter spaces. Xiao et al. (2017) [16] presented an approach to tackle the high dimensional parameter space challenge using (non-adaptive) sparse grid to generate the sampling points. Their approach is also based on RBF-POD, but only one reduced basis space is constructed offline while a two-level RBF interpolation is used online; the first layer generates interpolated coefficients in parameter space then a second RBF layer propagate these coefficients in time. Peherstorfer and Willcox (2016) [17] proposed a nonintrusive operator inference ROM approach that can be applied to linear systems or systems with nonlinear terms of low order polynomials. As an extension of this work, Qian et al. (2020) [18] proposed first to lift the generated data from the high-fidelity model to a quadratic form using auxiliary variables. Then apply the operator inference approach to the lifted system. However, defining the lifting maps is problem specific and requires characterization of the nonlinear term, which is an intrusive step. Guo et al. (2019) [19] proposed an approach based on Gaussian process regression models for the POD coefficients. Time is treated as a parameter, and the snapshots for the POD basis construction are generated based on parameter and time tensorization. Recently, several studies investigated ROM approaches based on Artificial Neural Networks (ANN) [20–26]. Swischuk et al. [27] compared between different machine learning methods for POD-based ROM modelling. They found ANN to be underperforming in cases where data is scarce. Their finding is in line with ANN literature, where it has been established that successful training of the ANN model requires a minimum data size that is a multiple of the complexity of ANN structure [28, 29].

In all of the studied ROM methods, one must select the time snapshots of the high-fidelity model a priori. If the snapshots are too close to each other in time, the computational burden is unnecessarily increased. On the other hand, defining coarse time intervals risks overlooking important system dynamics. This issue has been identified

by the projection-based community [30–33], where an adaptive selection of the time snapshots has been proposed to improve the projection-based POD modelling. However, the selection of the snapshot is imposed based on criteria that require knowledge of the governing equations, which is unfeasible when the system's precise dynamics are unknown, such as our nonintrusive setting.

In the present work, we aim to develop a general nonintrusive approach to identify and select the snapshots for any parametrized, time-dependent system. We build on our previous work for steady-state systems where we presented an adaptive sparse grid approach combined with POD [13]. We extend the adaptivity in parameter space to the time domain. We consider time as a parameter and use our adaptive technique to choose the important snapshots both in time and parameter spaces. This approach assumes a bounded time window of interest (i.e., $t \in [0, T]$). Therefore, the reduced model has no predictive capabilities beyond the defined end time T. However, the reduced model is able to simulate the spatiotemporal evolution of the system as a function of system parameters up to the end time T. We present three numerical tests for our adaptive approach. The first is a two-dimensional linear unsteady advection problem (Molenkamp test) that has an exact solution. This problem has five input parameters to investigate. In this test, we compare between the direct (fixed time grid) method and our time-adaptive approach. The second test is a lid-driven cavity problem, which was solved as a non-parametrized model (i.e., only time was considered as a parameter). The third is a two-dimensional time-dependent neutron diffusion problem in a subcritical reactor, which was parametrized with an 11-dimensional space. This problem is challenging due to the higher dimensionality of the parameter space and the abrupt response of the system during the transient.

The remainder of this chapter is organized as follows: Problem formulation is introduced in Section 5.2 along with a summary of the adaptive-POD algorithm and the time treatment approach. The numerical tests are presented in Section 5.3. Our conclusions are discussed in Section 5.4.

5.2. ADAPTIVE-POD APPROACH

5.2.1. PROBLEM FORMULATION

We are interested in building a reduced model for a general parametrized time-dependent problem. Due to our nonintrusive approach, the governing equations are unknown. Therefore, a general form for the problem under an unknown nonlinear operator $\mathcal{N}(\cdot)$ can be written as

$$\mathcal{N}(y(\boldsymbol{x}, t, \boldsymbol{\alpha}), \boldsymbol{x}, t, \boldsymbol{\alpha}) = s(\boldsymbol{x}, t, \boldsymbol{\alpha}), \tag{5.1}$$

where $y(\boldsymbol{x}, t, \boldsymbol{\alpha})$ is the solution of the system, $\boldsymbol{x}$ is the state independent variable (e.g., spatial coordinates, energy, or angular direction), t is time, $\boldsymbol{\alpha} \in \mathbb{R}^d$ is a vector of d parameters representing properties of the system (e.g., material, geometry, or boundary conditions), and $s(\boldsymbol{x}, t, \boldsymbol{\alpha})$ is a source function. We aim at building a reduced model to capture the dynamics of the solution $y(\boldsymbol{x}, t, \boldsymbol{\alpha})$ within a defined range of the parameter $\boldsymbol{\alpha}$. We assume the availability of a numerical solver for the discretized version of the problem. That is

$$\mathcal{N}(\boldsymbol{y}(t,\boldsymbol{\alpha}),t,\boldsymbol{\alpha}) = \boldsymbol{s}(t,\boldsymbol{\alpha}), \tag{5.2}$$

where $\boldsymbol{y} \in \mathbb{R}^n$ is a vector with n state variables. The computational burden usually scales with the dimension of the state vector n. Our approach is based on POD method, where we seek to approximate $\boldsymbol{y}(\boldsymbol{\alpha},t)$ using an expansion of the form

$$\boldsymbol{y}(\boldsymbol{\alpha},t) \approx \sum_{j=1}^{r} c_j(\boldsymbol{\alpha},t)\boldsymbol{v}_j, \tag{5.3}$$

where $\boldsymbol{v}_j \in \mathbb{R}^n$ is the basis vector (or POD mode) and $c_j(\boldsymbol{\alpha},t)$ is its coefficient that depends on parameter $\boldsymbol{\alpha}$ and the time (t). The POD method extracts a reduced basis space for the system using the left singular vectors of the singular value decomposition (SVD) applied to the snapshot matrix, which is a matrix containing an ensemble of solutions at different states of the system. The basis space can be truncated at the first r left singular vectors such that the truncation error is below a cut-off threshold γ_{tr}. That is

$$\frac{\sum_{j=r+1}^{n} \sigma_j^2}{\sum_{j=1}^{n} \sigma_j^2} < \gamma_{\text{tr}}, \tag{5.4}$$

where σ_j is the singular value of the left singular vector $\boldsymbol{v}_j$.

We have proposed in [13] an iterative algorithm to build a reduced model by adaptively selecting important points from the parameter space and updating the snapshot matrix. In this work, we propose to deal with time as a parameter. That is, we consider time to be an additional input parameter such that the solution $\boldsymbol{y}(\boldsymbol{\alpha},t) = \boldsymbol{y}(\boldsymbol{\alpha}^*)$, where $\boldsymbol{\alpha}^* = [\boldsymbol{\alpha}^T, t]^T$. The dimension of the parameter space becomes $d^* = d+1$ and Equation 5.3 becomes

$$\boldsymbol{y}(\boldsymbol{\alpha}^*) \approx \sum_{j=1}^{r} c_j(\boldsymbol{\alpha}^*)\boldsymbol{v}_j. \tag{5.5}$$

Formulating the problem in this way allows us to directly use the previously developed adaptive tool. Once the orthonormal basis is known, the coefficient values at the sampled point $\boldsymbol{\alpha}_q^*$ can be computed as

$$c_j(\boldsymbol{\alpha}_q^*) = <\boldsymbol{v}_j, \boldsymbol{y}(\boldsymbol{\alpha}_q^*)>, \tag{5.6}$$

where $< .,. >$ indicates the scalar product.

5.2.2. SMOLYAK INTERPOLATION
To compute the coefficient at any non-sampled point, we use the Smolyak iterative interpolant developed in [13]. Here, we only present a summary of the adaptive algorithm. At iteration k, the d^*-dimensional interpolant $A_{k,d^*}(c)(\boldsymbol{\alpha}^*)$ is given by

$$A_{k,d^*}(c)(\boldsymbol{\alpha}^*) = A_{k-1,d^*}(c)(\boldsymbol{\alpha}^*) + \Delta A_{k,d^*}(c)(\boldsymbol{\alpha}^*), \tag{5.7}$$

with $A_{0,d^*}(c)(\boldsymbol{\alpha}^*) = 0$, and

$$\Delta A_{k,d^*}(c)(\boldsymbol{\alpha}^*) = \sum_{n=1}^{m_k^\Delta} w_n^k \Theta_n(\boldsymbol{\alpha}^*), \tag{5.8}$$

where m_k^{Δ} is the cardinality of the so-called important set $\mathcal{Z}^k$. The important set contains the parameter points $\boldsymbol{\alpha}^*$ at which the interpolant was found to have an error greater than a pre-defined threshold γ_{int}. In the next iteration, the algorithm refines the sampling scheme in the neighbourhood of the points in the important set. The d^*-variate basis function $\Theta_n(\boldsymbol{\alpha}^*)$ is defined for every point $\boldsymbol{\alpha}_n^* = (\alpha_{n,1}^{i_1}, \ldots, \alpha_{n,d}^{i_d}) \in \mathcal{Z}^k$ as

$$\Theta_n(\boldsymbol{\alpha}^*) = \prod_{p=1}^{d^*} a_{\alpha_{n,p}^{i_p}}^{i_p}(\alpha_p), \tag{5.9}$$

where $\boldsymbol{\alpha}^*$ has support nodes $= (\alpha_1, \ldots, \alpha_{d^*})$, and i_p is the level (tree depth) index along dimension p. The unidimensional interpolant $a_{\alpha_{n,p}^{i_p}}^{i_p}(\alpha_p)$ is defined as

$$a_{\alpha_n^i}^i(\alpha) = 1 \quad \text{if } i = 1,$$

$$a_{\alpha_n^i}^i(\alpha) = \begin{cases} 1 - (m^i - 1) \cdot |\alpha - \alpha_n^i|, & \text{if } |\alpha - \alpha_n^i| < \dfrac{1}{m^i - 1}, \\ 0, & \text{otherwise}, \end{cases} \tag{5.10}$$

and

$$m^i = \begin{cases} 1 & \text{if } i = 1, \\ 2^{i-1} + 1 & \text{if } i > 1, \end{cases} \tag{5.11}$$

where the dependence on the dimension p is dropped for notational convenience.

The surplus w_n^k is defined as the difference between the interpolated value and the true value of the coefficient at $\boldsymbol{\alpha}_n^*$. That is

$$w_n^k = c(\boldsymbol{\alpha}_n^*) - A_{k-1,d^*}(c)(\boldsymbol{\alpha}_n^*). \tag{5.12}$$

The reduced model is then built by using the interpolant of Equation 5.7 as a surrogate model for the POD coefficients in Equation 5.5, yielding

$$\boldsymbol{y}(\boldsymbol{\alpha}^*) \approx \sum_{j=1}^{r} A_{k,d^*}(c_j)(\boldsymbol{\alpha}^*) \boldsymbol{v}_j. \tag{5.13}$$

5.2.3. ADAPTIVE SAMPLING STRATEGY

The adaptive sparse grid algorithm is based on arranging the nodes along each dimension in a tree structure, as shown in Figure 5.1. Each node has two children and one father with an exception at the boundary nodes in level 2 where each has one child only. The nodes are then tensorized to form points in parameter space. To maximize the separation of points in parameter space, we choose the equidistant rule for the unidimensional nodes. Each point in parameter space has *forward* and *backward* points. The forward points for a point $\boldsymbol{\alpha} = (\alpha_1, \alpha_2 \ldots, \alpha_{d^*})$ is generated by tensorizing the children of each node with the rest of the nodes. That is, the first forward point of $\boldsymbol{\alpha}$ is $(b_1(\alpha_1), \alpha_2 \ldots, \alpha_{d^*})$, where $b_1(\alpha)$ is a function that returns the first child from the tree. The second forward point is $(b_2(\alpha_1), \alpha_2 \ldots, \alpha_{d^*})$, where $b_2(\alpha)$ is a function that returns the second child. By applying $b_1(\alpha)$ and $b_2(\alpha)$ to α_2, the third and fourth forward points are generated and so forth for the rest of the dimensions. Therefore, for any point in parameter space we can generate

up to $2d$ forward points. The backward points are generated in the same manner but by using a function that returns the father of a node instead of the child function. Hence, each point has at most d backward points. By generating the backward points recursively, the set of *ancestors* are created. Note that a forward point can be shared between two different backward points. Therefore, points in parameter space do not form a classical tree structure but rather are connected as a network.
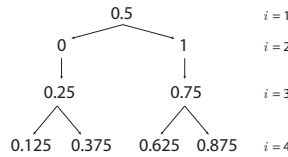

Figure 5.1: Illustration of the first 4 levels of the tree structure where 0.5 is the root of the tree and nodes are added at half the distance between the previous nodes. Each node has 2 children except the nodes at level 2 where each has one child only.

The parameter space is bounded by the defined upper and lower values for each dimension in $\boldsymbol{\alpha}^*$. This space is mapped to a unitary hypercube with dimension d^*, where 1 is mapped to the upper value and 0 represents the lower value of the range. In the initialization step ($k = 0$), the algorithm selects the central point in the hypercube and adds it to the important set $\mathcal{Z}^0$. The high-fidelity model is then sampled at that point. Then, a reduced model is built using Equation 5.13. In the first iteration ($k = 1$), a trial set is generated from the forward points of the points in $\mathcal{Z}^0$. The algorithm then samples the high-fidelity model and computes the error of the reduced model at each point in the trial set. Points with an error above a pre-defined threshold (γ_{int}) are considered important points. Then, in any iteration k, the trial set is generated from the forward points of $\mathcal{Z}^{k-1}$. For each point in the trial set, if the error is found to be above γ_{int}, this point is marked as a candidate point. The algorithm then considers the ancestors of each candidate point. If all ancestors were included in $\bigcup_{l=0}^{k-1} \mathcal{Z}^l$, that candidate point is added to the important set. On the other hand, if the candidate point has an ancestor that was not included in $\bigcup_{l=0}^{k-1} \mathcal{Z}^l$, that ancestor is marked important and the candidate point is stored and tested again in the next iteration. Points that are not marked important are added to the inactive set.

To control the efficiency of the sampling scheme, a generated forward point is excluded from the trial set if it has a fraction of inactive backward points above a predefined parameter $\mu \in [0,1]$. For $\mu = 1$, all forward points are sampled and the algorithm is more exploratory whereas for $\mu = 0$, the algorithm is more efficient by only sampling points which have all their backward points in the important set. Figure 5.2 summarizes the algorithm with a flow chart. For a detailed description of the algorithm, we refer the reader to [13].

Clearly, time is not an input parameter and special attention has to be taken with such an approach. This is because numerical solvers are discretized in time. The algorithm could request a snapshot at a certain time t_l, which could be a time instance in-between the solver's default time steps. This can be addressed either by solving up to the last default time step before t_l then modifying the time step to reach t_l or interpolating between two
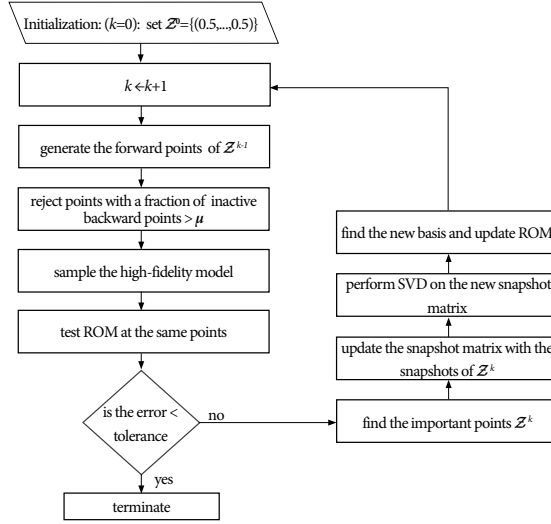
Figure 5.2: A flow chart illustrating the adaptive-POD algorithm.

time steps before and after t_l. Additionally, for every request of t_l, the high-fidelity model will solve for all time instances from $t = 0$ up to t_l. Management of the interface with the solver is important to avoid redundant simulations. If at one iteration, the algorithm requests $\boldsymbol{\alpha}_q$ with t_l, we can store all generated snapshots for $t < t_l$ in a library. In this manner, the algorithm can recall from this library instead of rerunning the high-fidelity solver with each $\boldsymbol{\alpha}_q$ call. Likewise, If the algorithm requests $t > t_l$ with $\boldsymbol{\alpha}_q$, the solver needs only to be restarted from t_l instead of the initial conditions $t = 0$. This strategy saves computational resources.

However, in cases where memory is limited, one might opt to store only a certain percentage of the generated snapshots. Then, restart a requested simulation from the closest stored snapshot. Note that storing all generated snapshots is not an integral part of the algorithm. The only snapshots that need to be stored are the ones marked as important and included in the snapshot matrix. Stored snapshots that are not used during the construction stage can serve as testing points for the reduced model once the algorithm is terminated. The snapshot matrix is the only memory consuming step in the algorithm. One can reduce the memory burden of the snapshot matrix with the use of an SVD updating algorithm [34] instead of the full SVD at each iteration. In our implementation for this work, however, we have used full SVD at each iteration.

We use the ℓ^2 norm to compute the relative error for any point $\boldsymbol{\alpha}_q^*$ as

$$e_q^k = \frac{\left\| \boldsymbol{y}(\boldsymbol{\alpha}_q^*) - \sum_{j=1}^r A_{k,d^*}(c_j)(\boldsymbol{\alpha}_q^*)\boldsymbol{v}_j \right\|_{\ell^2}}{\left\| \boldsymbol{y}(\boldsymbol{\alpha}_q^*) \right\|_{\ell^2} + \epsilon}, \tag{5.14}$$

where ϵ is introduced as an offset for cases when $\left\| \boldsymbol{y}(\boldsymbol{\alpha}_q^*) \right\|_{\ell^2}$ has near zero magnitude. A point $\boldsymbol{\alpha}_q^*$ is marked as a candidate point when e_q^k is above the threshold γ_{int}. The iterative

algorithm is terminated when e_q^k for all points in the trial set of iteration k is below a global tolerance ζ.

5.3. APPLICATIONS

Our proposed algorithm is tested on three time-dependent problems. The first is a two-dimensional linear unsteady advection problem that has an exact solution. This problem is also called the Molenkamp test [35]. We parametrize this problem on a five-dimensional space. The second is a lid-driven cavity test. This problem is not parametrized but tests the ability of the algorithm to detect the important transient window. The third is a challenging 11-dimensional transient nuclear reactor problem. This problem simulates a subcritical reactor with an external source.

5.3.1. MOLENKAMP TEST

The original Molenkamp test [35] is a two-dimensional advection problem which has an exact solution as a Gaussian cloud of material being transported in a circular path without changing its shape. However, in order to create a more challenging setting for the adaptive-POD algorithm, we modified this problem to include an additional reaction term, which in effect causes the amplitude of the solution to decay over time. The dimensionless advection-reaction equation is

$$\frac{\partial q(x, y, t)}{\partial t} + u \frac{\partial q(x, y, t)}{\partial x} + v \frac{\partial q(x, y, t)}{\partial y} + \lambda_3 q(x, y, t) = 0, \qquad (x, y) \in [-1, 1], \qquad (5.15)$$

where the velocity field describes a solid body rotation $u = -2\pi y$ and $v = 2\pi x$. The initial condition is

$$q(x, y, 0) = \lambda_1 0.01^{\lambda_2 h(x, y, 0)^2}, \qquad h(x, y, 0) = \sqrt{(x - \lambda_4 + \frac{1}{2})^2 + (y - \lambda_5)^2}. \qquad (5.16)$$

The exact solution is imposed on the inflow boundary condition as

$$q(x, y, t) = \lambda_1 0.01^{\lambda_2 h(x, y, t)^2} e^{-\lambda_3 t}, \quad h(x, y, t) = \sqrt{(x - \lambda_4 + \frac{1}{2} \cos 2\pi t)^2 + (y - \lambda_5 + \frac{1}{2} \sin 2\pi t)^2}. \qquad (5.17)$$

The exact solution is evaluated on a Cartesian uniform 100×100 grid. Therefore, the model has 10,000 degrees of freedom. Note that in this problem, evaluating the solution is computationally efficient and a reduced model is not necessary. However, this problem is selected to test the ability of the developed algorithm in capturing such dynamics.

The problem is parametrized with a 5 dimensional space λ_i for $i = 1, \ldots, 5$. Figure 5.3 and Figure 5.4 show selected time snapshots of the solution for different values of λ_2. The snapshots show the Gaussian cloud initially centred at $x = -0.5$ and $y = 0$. Over time, the cloud is transported in a circle which completes a full rotation at $t = 1$. The cloud also decays to reach a near-zero magnitude after a full rotation. The parameter λ_1 is a linear scaling factor that controls the magnitude of the initial cloud, λ_4 and λ_5 control the initial coordinates of the center of the cloud with respect to the domain center. The parameter λ_3 is the decay constant of the cloud that controls the speed of the decay. The parameter
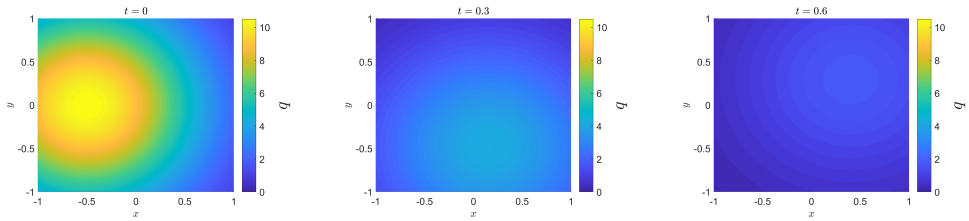
Figure 5.3: Snapshots of the solution for the smooth Molenkamp problem ($\lambda_2 = 0.15$) at selected time steps ($t \in \{0, 0.3, 0.6\}$).
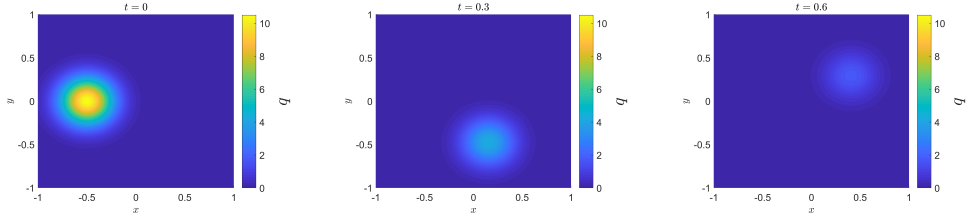

Figure 5.4: Snapshots of the solution for the steep Molenkamp problem ($\lambda_2 = 3$) at selected time steps ($t \in \{0, 0.3, 0.6\}$).

λ_2 controls the size of the cloud. For smaller values of λ_2, the cloud size is bigger and the solution is smoother over the domain as shown in Figure 5.3 whereas for higher values, the solution has a steeper gradient (spike-like) as shown in Figure 5.4.

We test two different settings of the problem. The first is a smooth solution by varying λ_2 between 0.1 and 0.2 and the second is a steep solution with values of λ_2 between 2 and 4. The steep solution is more challenging to capture for a POD-based ROM because as the solution becomes steeper (or closer to being orthogonal), the required basis space becomes larger. That is, the rank of the snapshot matrix is higher for snapshots that are already orthogonal or near orthogonal, which entails more POD modes for an accurate representative model.

Table 5.1 summarizes the range of variation for each parameter. We are interested in building a reduced model that can reproduce the solution $q(x, y, t)$ over the spatial domain and time $t \in [0, 1]$ for any values of the parameters within the defined range.

Table 5.1: Range of variations for each parameter in the Molenkamp test.

Parameter	Lower bound	Upper bound
λ_1	1	20
λ_2	0.1 (2 for steep setting)	0.2 (4 for steep setting)
λ_3	1	5
λ_4	-0.1	0.1
λ_5	-0.1	0.1

We compare two approaches for this test. The first is the developed time-adaptive approach as described in Section 5.2. The second is the more direct approach by defining

a fixed time grid then building a separate reduced model for each time instance in the grid. To reproduce the solution at any time t, the solution of the ROM models on the time grid is interpolated. However, to select the snapshots in parameter space, we still use the adaptive algorithm for each model on the grid. For this approach, a single basis space is constructed for all ROM models on the grid. A point in parameter space is selected to be included in the important set if any of the ROM models marked that point as important. Thus, the basis space for all ROM models on the fixed grid is updated with any point marked important by at least one of the ROM models. We initially defined the fixed time grid with 11 times points uniformly separated in the time window of interest (i.e., $t \in [0, 1]$).

For both approaches, we choose a greediness value of $\mu = 0$ and require the reduced model to have a maximum of 1% ℓ^2 norm error. Therefore, we set the global tolerance (ζ) to be 1% and the adaptive threshold (γ_{int}) to 0.1%. The POD truncation threshold (γ_{tr}) was set to 10^{-12}. The results are summarized in Table 5.2. The table presents a comparison between the two approaches for both the smooth and the steep solution settings in the number of calls to run the high-fidelity model, the total number of snapshots resulted from these runs, the number of POD modes after truncation, and the maximum relative error resulted from testing the model on 1000 randomly generated points using latin hypercube sampling (LHS) (i.e., snapshots generated by random point in the space formed by the parameters λ_i and time t, which were not part of the snapshot matrix). In Table 5.2, we report the maximum error results for the model of the fixed grid approach in two separate occasions: The maximum error at the predefined fixed grid instances and the maximum error at interpolated points in-between these instances. The maximum of the two values is the more relevant result to be compared to the error results of the adaptive approach. The interpolated values were obtained using splines interpolation.

For the smooth Molenkamp setting, the time-adaptive approach needed 775 high-fidelity model runs and computed 6369 snapshots. Out of the total number of snapshots, 4692 were marked important and included in the snapshot matrix. The number of POD modes after truncation was 33. On the other hand, the fixed time grid approach needed 1379 model runs resulting in a total of 15169 snapshots, out of which 9889 snapshots were important. The result of the test on the 1000 random points showed the time-adaptive model having a maximum error of 0.5%, which is less than the set tolerance of 1%. The fixed grid model resulted in a maximum error of 0.17% at the grid points. However, at interpolated points (time instances in-between the defined grid points), the maximum error was 1.1%. The adaptive model has a clear advantage in this test as the error was lower and the model was more efficient in the number of high-fidelity model calls. Note that for the time-adaptive approach, not all high-fidelity model calls are simulated up to the end time T (where in this case $T = 1$). This is because the time-adaptive algorithm requests some high-fidelity model runs with a time t_l that is less than T. Therefore, the efficiency in the time-adaptive model is not only in the reduced number of high-fidelity model runs but also in the reduced computational burden of each model run.

For the steep solution test, we notice that both the time-adaptive and fixed grid approaches needed an increased number of model runs and a larger POD basis space compared to the smooth solution setting. For the time-adaptive approach, a total of 2944 high-fidelity model runs were requested with 78035 snapshots sampled and 64379 of

Table 5.2: Results for the Molenkamp problem for the smooth and steep setting comparing time-adaptive and fixed grid approaches. The number of model runs indicates the number of calls to run the high-fidelity model. The number of snapshots indicates the total number of snapshots resulted from the high-fidelity model runs. The number of POD modes indicates the number of basis vectors selected after truncation. The maximum relative error reports the maximum computed error from testing the model on 1000 randomly generated points that are not part of the snapshots. For the fixed grid, the errors at the defined grid points and at interpolated time instances are reported separately.

Problem Setting	Approach	Number of model runs	Number of snaps-hots	Number of POD modes	Maximum relative ℓ^2 error	
Smooth Molenkamp	Time-adaptive	775	6369	33	0.5%	
	Fixed grid (11 points)	1379	1379 × 11	33	At grid points	0.17%
					Interpolated	1.1%
Steep Molenkamp	Time-adaptive	2944	78035	238	1.4%	
	Fixed grid (11 points)	5093	5093 × 11	223	At grid points	0.33%
					Interpolated	76%
	Fixed grid (101 points)	5093	5093 × 101	234	At grid points	0.33%
					Interpolated	0.34%

them marked as important. The number of POD modes was 238. A conclusion similar to the smooth setting can be drawn in this case about the time-adaptive model being more efficient and more accurate. In fact, the fixed grid model captured the dynamics of the solution at the grid points but the error was as high as 76% at the interpolated points. In order to produce a more accurate model, we built another fixed grid model with 101 time points uniformly distributed in time $t \in [0, 1]$). This model reduced the maximum error at the interpolated points to about 0.34%. However, this was achieved with about twice as much model runs compared to the time-adaptive approach and about 6 times more snapshots.

Table 5.3 summarizes the number of projected important points on each dimension. This number represents the linearity of the output with respect to each dimension. A value of 1 signals that the algorithm considered that dimension to be constant. In other words, varying the value of that parameter has a negligible effect on the output of the model with respect to the defined tolerance. A value of 3 means that the algorithm considered the output of the model to be linear or piecewise linear with respect to that dimension. A higher value implies a nonlinear parameter, and the degree of the non-linearity scales with the value. It can be seen that the algorithm recognized λ_1 as a linear scaling parameter in both settings. The decay constant λ_3 was the most sampled parameter and was not affected by the change in the shape of the solution controlled by λ_2. This can be confirmed from the exact solution in Equation 5.17. The paramters λ_4 and λ_5, on the other hand, were affected by the shape as they control the location of the cloud. For this reason, the algorithm added more points along the dimensions of these parameters in the steep

Table 5.3: Number of projected important point on each dimension for the Molenkamp test.

parameter	Number of points in the Smooth setting	Number of points in the Steep setting
λ_1	3	3
λ_2	9	13
λ_3	33	33
λ_4	5	17
λ_5	5	17
t	129	513

setting, where the cloud size is smaller compared to the smooth setting.

5.3.2. LID-DRIVEN CAVITY TEST
In this test, the incompressible Navier–Stokes equations are solved in a two-dimensional lid-driven cavity. Zero-velocity boundary conditions are assumed around the cavity except at the top lid, where a velocity equal to v_{lid} is imposed. The model equations read

$$\frac{\partial \boldsymbol{u}(x,y,t)}{\partial t} + \nabla \cdot (\boldsymbol{u}(x,y,t) \otimes \boldsymbol{u}(x,y,t)) - \nabla \cdot \left[v(\nabla \boldsymbol{u}(x,y,t) + \nabla (\boldsymbol{u}(x,y,t))^T) \right] = -\nabla p,$$

$$\text{in } \Omega = [0,1]^2,$$

$$\nabla \cdot \boldsymbol{u}(x,y,t) = 0 \qquad \text{in } \quad \Omega,$$

$$\boldsymbol{u}(x,y,t) = \boldsymbol{0} \qquad \text{on } \quad \Gamma_1,$$

$$\boldsymbol{u}(x,y,t) = (v_{\text{lid}},0)^T \qquad \text{on } \quad \Gamma_2,$$

$$\boldsymbol{u}(x,y,0) = \boldsymbol{0}, \tag{5.18}$$

where $\boldsymbol{u}(x,y,t)$ is the flow velocity, v is the viscosity, p is the pressure, and v_{lid} is the velocity of the top cavity wall, which was imposed as a ramp according to

$$v_{\text{lid}} = \begin{cases} -t & \text{if } 0 \le t < 1, \\ -1 & \text{if } t \ge 1. \end{cases} \tag{5.19}$$

The domain is illustrated in Figure 5.5. We consider a laminar flow with Reynolds number of 1000 and are interested in the velocity field within a time range $t \in [0,100]$. An in-house Navier–Stokes solver was used as the high-fidelity model [36]. The system of equations is solved with a pressure-correction method, discretizing the equations in space with a discontinuous Galerkin finite element method and in time with the implicit Euler scheme. A fixed time-step size of 10^{-3} was chosen. The domain was discretized on a structured non-uniform (finer near the walls) mesh of 60×60 elements. The velocity field was discretized using a second-order polynomial, which leads to a total of 43200 degrees of freedom for the high-fidelity model. A single high-fidelity simulation to $t = 100$ requires about 35 CPU-hours.
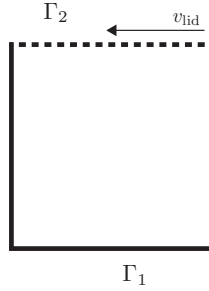
Figure 5.5: Illustration of the domain for the lid-driven cavity problem. The boundaries around the cavity are marked by Γ_1 except at the top lid where Γ_2 labels the boundary there.

Lorenzi et al. (2016) [37] has presented an approach to build a reduced model for this benchmark using a projection-based POD approach. To select the snapshots, the authors sampled the velocity field using a fixed grid of 1000 equally spaced time points. As pointed out in their work, this test is challenging for projection-based POD methods due to the potential instability of the reduced model induced by truncating modes that have small energy magnitudes but are important for dissipating the energy of the system. Nonintrusive approaches do not face such an issue.

We aim to build a non-parametrized reduced model that captures the velocity evolution with time as a response to v_{lid}. We require a 0.5% maximum ℓ^2 norm error and set a POD truncation threshold γ_{tr} to 10^{-12}. The algorithm selected 463 snapshots and marked 232 of them as important. The number of POD modes was 229. The selected points are shown in Figure 5.6. The algorithm was successful in identifying the first part of the transient to be more important than the last. This is because most of the changes to the velocity field occur within the first few seconds and then gradually stabilize until a steady-state is reached. In fact, the algorithm recognizes that the flow is in a steady-state by $t = 50$ and no snapshots were marked important between $t = 50$ and $t = 100$.
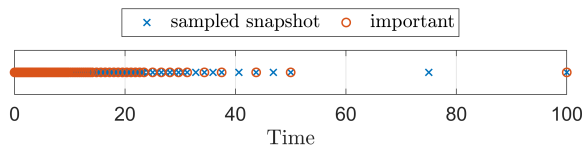
Figure 5.6: Time instances selected by the time-adaptive algorithm for the lid-driven cavity test. Points added to the important set are marked with a red circle.

To test the reduced model, Figure 5.7 shows the computed relative ℓ^2 norm error between the reduced model and the high-fidelity model at 10,000 randomly generated points in time. The tested points were not part of the snapshots matrix. It can be seen that all tested points resulted in an error below the set tolerance of 0.5%. The figure shows the error to oscillate between 10^{-7} and the tolerance (5×10^{-3}), with the oscillation frequency being higher in the first part of the time domain. These oscillations are due to the fact that some of the tested points are very close to points that were marked important and

included in the snapshot matrix. The error for reconstructing a point in the snapshot matrix can be estimated with the POD truncation threshold $\gamma_{\mathrm{tr}} = 10^{-12}$. Therefore, γ_{tr} can be considered as a lower bound of the error in the ROM model. When a point is tested near a point included in the snapshot matrix, the error can be expected to approach γ_{tr}. On the other hand, when the tested point is further from any point in the snapshot matrix, the error increases to the tolerance. This is evident by considering Figure 5.6 where the frequency of the selected important points correlated well with the frequency of the oscillation in the error shown in Figure 5.7.

The figure shows the maximum error to be 0.38%. In fact, this maximum is at the first time step of the high-fidelity model. This high error is attributed to the discontinuity in the velocity field between the initial conditions (null velocity everywhere) and the first time step (velocity almost zero except at the very top of the cavity where v_{lid} is introduced). The relative error is magnified by the near zero ℓ^2 norm of the solution at this first step. The maximum absolute difference between the reduced model and the high-fidelity model at this point was about 6×10^{-6}, while the magnitude of the maximum velocity at the top of the cavity was 8×10^{-4}. Beyond $t = 1$ (when the input ramp ends), the highest error is observed to be 0.054% at $t = 68.157$ s. A comparison between the high-fidelity model and the reduced model at this point is shown in Figure 5.8. We also plot the velocity components along the horizontal and vertical central lines in Figure 5.9 at $t = 68.157$. The figures show that the reduced model produced an accurate representation of the high-fidelity model despite the fact that no snapshots were selected in the important set between $t = 50$ and $t = 100$. In addition, Figure 5.9 compares the results with steady-state benchmark data from Botella and Peryret (1997) [38] to verify that the algorithm was successful in recognizing the flow to be in a steady-state beyond $t = 50$. Simulating the flow for the 10,000 tested points required about 10 seconds on a personal computer with the reduced model compared to the 35 CPU-hours needed by the high-fidelity model to simulate the flow to $t = 100$.

5.3.3. SUBCRITICAL REACTOR TEST

Nuclear reactors are complex systems with multiple interacting physical phenomena. A standard model to describe the neutron flux dynamics inside a reactor is the time-dependent diffusion equation [39]

$$\frac{1}{v}\frac{\partial \phi(\boldsymbol{x}, t)}{\partial t} - \nabla \cdot D(\boldsymbol{x})\nabla\phi(\boldsymbol{x}, t) + \Sigma_a(\boldsymbol{x})\phi(\boldsymbol{x}, t) = S(\boldsymbol{x}, t), \qquad \boldsymbol{x} \in \Omega \qquad (5.20)$$

where $\phi(\boldsymbol{x}, t)$ is the one-speed neutron flux with speed $v = 300,000$ cm/s, $D(\boldsymbol{x})$ is the diffusion coefficient, and Σ_a is the absorption (removal) cross section. The source term $S(\boldsymbol{x}, t)$ is defined as

$$S(\boldsymbol{x}, t) = (1 - \beta)v\Sigma_f(\boldsymbol{x})\phi(\boldsymbol{x}, t) + \lambda C(\boldsymbol{x}, t) + q(\boldsymbol{x}, t), \qquad (5.21)$$

where β is the delayed neutron fraction, v is the number of neutrons emitted per fission, Σ_f is the fission cross section, $q(\boldsymbol{x}, t)$ is the external neutron source, and λ is the decay constant of the precursors $C(\boldsymbol{x}, t)$. The dynamics of the precursors is governed by

$$\frac{\partial C(\boldsymbol{x}, t)}{\partial t} = -\lambda C(\boldsymbol{x}, t) + \beta v\Sigma_f(\boldsymbol{x})\phi(\boldsymbol{x}, t). \qquad (5.22)$$
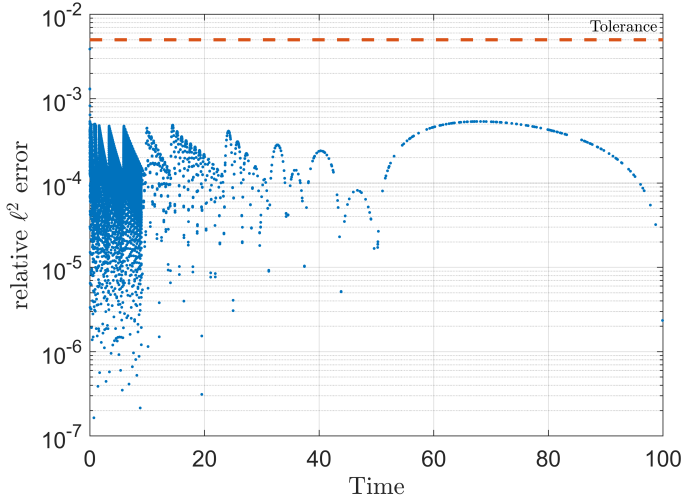
Figure 5.7: Relative ℓ^2 error in the reduced model for the lid-driven cavity problem tested on 10,000 points that were not part of the snapshot matrix. The global tolerance was set at 0.5%.

We consider a two-dimensional domain (i.e., $\boldsymbol{x} = (x, y)$) divided into 4 regions as shown in Figure 5.10. The dimensions of the reactor (including the extrapolated length) were set to $x, y \in [-109.36, 109.36]$, which were chosen such that the flux $\phi(\boldsymbol{x}, t)$ is zero at the boundary Γ. Each region has uniform material properties such that $D(\boldsymbol{x}), \Sigma_a(\boldsymbol{x}), \Sigma_f(\boldsymbol{x}) \rightarrow \boldsymbol{D}, \boldsymbol{\Sigma}_a, \boldsymbol{\Sigma}_f \in \mathbb{R}^4$. The external source $q(\boldsymbol{x}, t)$ is assumed to be present only in the lower left corner of the domain,

$$q(\boldsymbol{x}, t) = \begin{cases} q_{\text{ext}}(t) & \forall \boldsymbol{x} \in \text{Region 1}, \\ 0 & \text{elsewhere.} \end{cases} \tag{5.23}$$

The multiplication factor of a reactor (k_{eff}) is the ratio of the neutrons produced from fission in one generation to the neutrons lost in the previous generation. For $k_{\text{eff}} < 1$, a fission chain reaction cannot be sustained and the reactor is said to be subcritical, while for $k_{\text{eff}} > 1$, the reactor is supercritical since the neutron population is multiplying over time. For $k_{\text{eff}} = 1$, the reactor is critical and the neutron population is constant in time. In our test, the reactor is assumed to be in a subcritical condition with a multiplication factor $k_{\text{eff}} = 0.94$ in the nominal state. The neutron population is kept in a steady-state due to the external source $q_{\text{ext}}(t) = 1$. At time $t = 100$ s, the source intensity is perturbed. This is equivalent to a step-change in the source at time $t = 100$ s,

$$q_{\text{ext}}(t) = \begin{cases} 1 & \text{for } 0 \leq t < 100, \\ q_1 & \text{for } t \geq 100, \end{cases} \tag{5.24}$$

where $q_1 \in [0, 5]$. The neutron flux response is then observed as a function of time and space. We aim to build a reduced model that captures the dynamics of flux under different

(a) Velocity magnitude of HFM.

(b) Velocity magnitude of ROM.

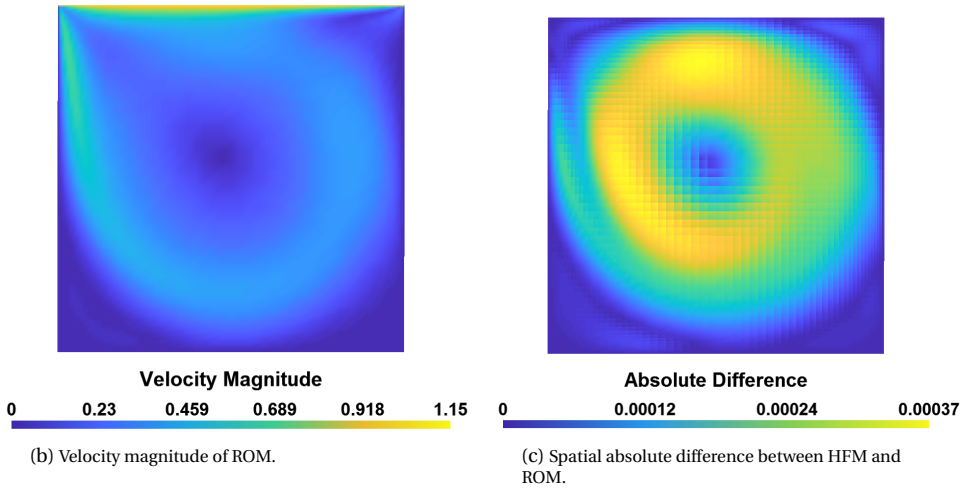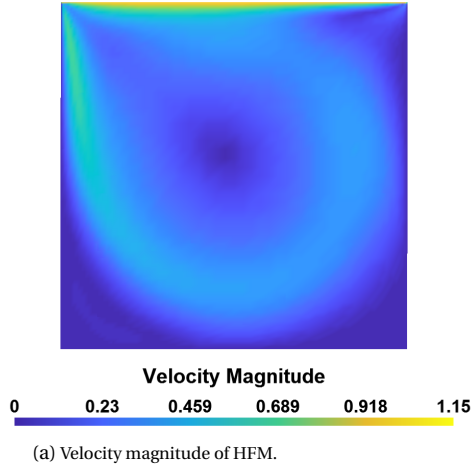(c) Spatial absolute difference between HFM and ROM.

Figure 5.8: Comparison between the reduced model (ROM) and the high-fidelity model (HFM) for the lid-driven cavity problem at $t = 68.157$ where the relative ℓ^2 error was found to be 0.054%.

(a) u_x profile along vertical centerline.

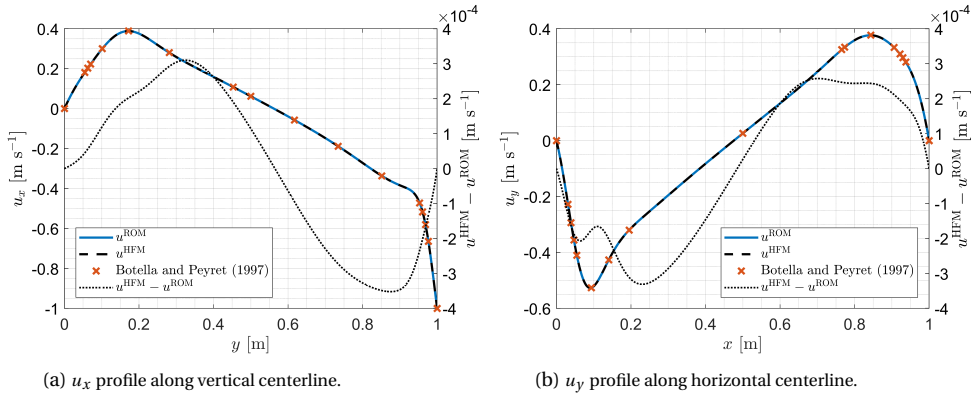(b) u_y profile along horizontal centerline.

Figure 5.9: Velocity components profile along the central line for both the high-fidelity (HFM) and the reduced model (ROM) for lid-driven cavity problem at the time $t = 68.157$ s. Benchmark data from Botella and Peryret (1997) [38] are also marked. The right axes of the figures show the difference between HFM and ROM ($u^{\mathrm{HFM}} - u^{\mathrm{ROM}}$).
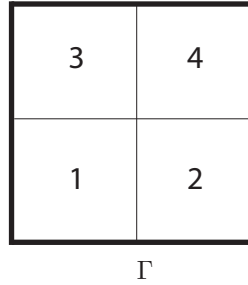

Figure 5.10: Domain of the subcritical reactor test showing the boundary Γ and the 4 regions. The neutron source is present only in Region 1.

conditions of material properties $D, \Sigma_a, \lambda, \beta$ and source intensity q_1. Therefore, the model is parametrized with an 11-dimensional space. The nominal values and range of variations of each parameter are summarized in Table 5.4.

The parameters range of variations was chosen such that the reactor is kept in a subcritical condition ($k_{\mathrm{eff}} < 1$) in all cases. The level of the subcritical condition is controlled with Σ_a and D, which also set the initial flux value. As the reactor gets closer to criticality, the response following a perturbation becomes steeper and the transient becomes longer. Therefore, the time until reaching a new steady state is a function of the material properties. The flux response to the perturbations can be described by two main parts. First, an initial abrupt response due to the prompt neutrons, which has a magnitude controlled by β, Σ_a and D. This prompt response has a duration in the order of $1/v\Sigma_a = 2 \times 10^{-5}$ s. The second is the response due to the delayed neutrons emitted from the decay of the precursors, which is governed by a time in the order of $1/\lambda = 12$ s. The final steady-state value scales linearly with the external source q_1. Therefore, this test
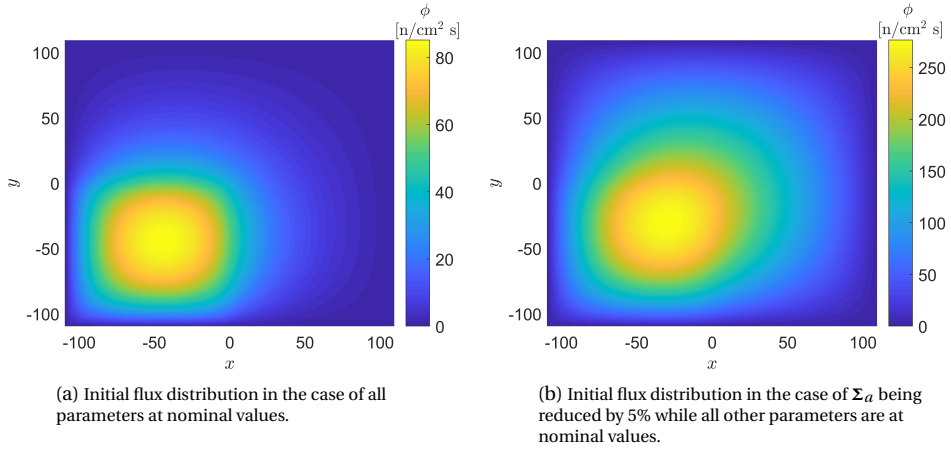
Table 5.4: Nominal values and range of variations of each parameter in the subcritical reactor test.

Parameter	Nominal Value	% variation
$\boldsymbol{D} = [D_1, D_2, D_3, D_4]^{\mathrm{T}}$	9.21 cm	$\pm 5\%$
$\boldsymbol{\Sigma}_a = [\Sigma_{a1}, \Sigma_{a2}, \Sigma_{a3}, \Sigma_{a4}]^{\mathrm{T}}$	0.153 cm^{-1}	$\pm 5\%$
β	0.00686	$\pm 20\%$
λ	0.08 s^{-1}	$\pm 20\%$
q_1	2.5 n/cm^3 s	$\pm 100\%$

poses a challenge for any nonintrusive approach because different parameters affect the dynamics at different timescales.

The model was solved using a finite element method with an unstructured mesh descretizing the spatial domain such that the model has 1084 degrees of freedom. An implicit Euler discretization for time was employed with variable step sizes. The step size following the perturbation was taken as 10^{-5} s for the first 0.1 seconds to resolve steep variations. Then, a step size of 10^{-2} s was employed for the remainder of the transient. A single simulation of the transient takes about 30 seconds on a personal computer. Therefore, this model is not computationally demanding. However, we considered this test to challenge the algorithm in capturing the effect of the 11 parameters on the complete transient. The initial flux distribution before perturbing the source ($t = 0$) for the nominal case is shown in Figure 5.11a while Figure 5.11b shows the initial flux distribution when reducing $\boldsymbol{\Sigma}_a$ by 5%. It can be seen that reducing $\boldsymbol{\Sigma}_a$ caused the flux intensity to increase and the shape to broaden over the spatial domain, which is expected since fewer neutrons are being absorbed in this case. Figure 5.12 shows the transient tracking the flux at the center of the reactor following three different source perturbations $q_1 \in \{0, 2.5, 5\}$ and compares the case of all parameters at the nominal values with the case of only reducing $\boldsymbol{\Sigma}_a$ by 5% of the nominal value. The figure shows that by reducing $\boldsymbol{\Sigma}_a$, the reactor is closer to criticality, which not only has an effect on the initial flux value but also resulted in a slower response to reach a new steady-state following a perturbation.

We built a reduced model with a global tolerance of 1% and a POD truncation threshold of 10^{-12}. The algorithm ran 3295 high-fidelity simulations and selected 155270 snapshots, where 52710 were marked important. The number of POD modes was 294 after truncation. The projection of the important points onto each dimension is given in Table 5.5. The table shows that the algorithm considered the diffusion coefficient to be linear within the defined range of $\pm 5\%$ while the absorption cross section was the most nonlinear parameter. This is expected because the absorption cross section has a direct effect on the subcriticality level of the reactor. In addition, it is shown that Σ_{a1} was considered the most nonlinear parameter and was sampled more densely because it belongs to the region that contains the external source. On the other hand, Σ_{a4} belongs to the region furthest from the source and was sampled the least. The parameters λ and β had 3 unique nodes each, which implies that within the defined ranges of $\pm 20\%$, the effect of these parameters on the dynamics of the reactor is linear. The external source intensity was correctly identified as a linear scaling factor. The time parameter was considered important at 110 time instances.

(a) Initial flux distribution in the case of all parameters at nominal values.

(b) Initial flux distribution in the case of Σ_a being reduced by 5% while all other parameters are at nominal values.

Figure 5.11: Comparison of the distribution of the initial neutron flux ($t = 0$) for the subcritical reactor test showing the difference in the flux intensity and shape between the case of all parameters at nominal values and a case of Σ_a reduced by 5% of the nominal value while all other parameters are kept at nominal values.

(a) selected transients of the flux following different source perturbations at $t = 100$ s in the case of all parameters at nominal values.

(b) selected transients of the flux following different source perturbations at $t = 100$ s in the case of Σ_a being reduced by 5% while all other parameters are at nominal values.
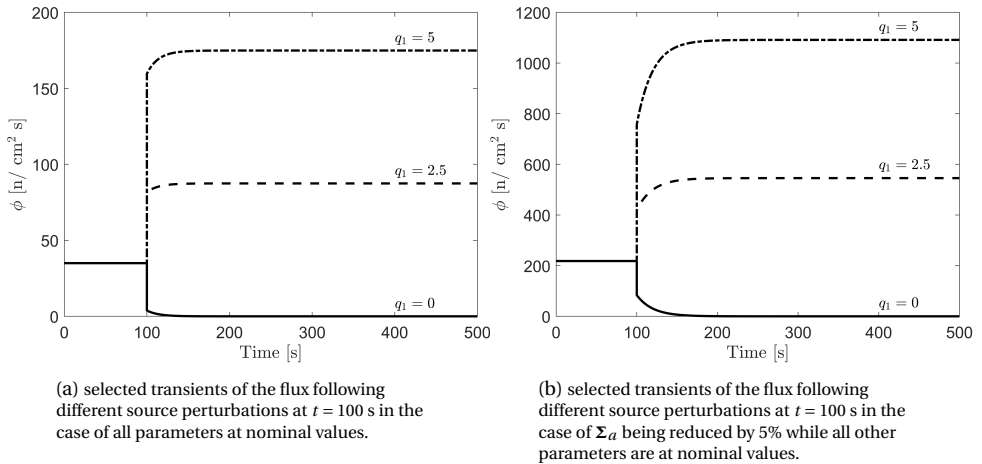
Figure 5.12: Transients for the subcritical reactor test at a point in the center of the reactor following selected perturbation of $q_1 \in \{0, 2.5, 5\}$ showing the difference in response between the case of all parameters at nominal values and a case of reducing Σ_a by 5% of the nominal value.

Table 5.5: Nominal values of the parameter in the subcritical reactor test.

Parameter	Number of unique nodes	Parameter	Number of unique nodes
D_1	3	Σ_{a3}	12
D_2	3	Σ_{a4}	8
D_3	3	β	3
D_4	3	λ	3
Σ_{a1}	22	q_1	3
Σ_{a2}	11	t	110

Figure 5.13 shows the projection of the sampled points onto the (Σ_{a1}, t) plane. It can be seen that the algorithm sampled most of the points during the period from $t = 100$ s to $t = 250$ s, which is the transient time following the source perturbation. Along the Σ_{a1} dimension, most of the sampled points were in the lower range of the domain. This is expected, because for lower values of the absorption cross section, the reactor is closer to criticality and the response becomes more nonlinear.
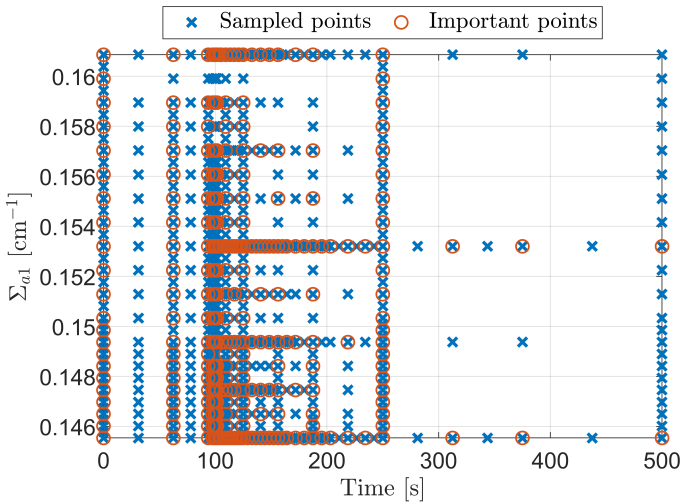

Figure 5.13: Projection of the sampled point onto the (Σ_{a1}, t) plane for the subcritical reactor test. Points included in the important set are marked with a circle.

The model was tested on 1000 randomly generated points using LHS method. The histogram of the relative errors in Figure 5.14 shows that 99.5% of the points were below the tolerance. The maximum relative error was found to be 3.2%. The point with the maximum error corresponds to a case where the source value $q_1 = 1.1 \times 10^{-3}$ n/cm^3 s and time $t = 203$ s. The solutions of the reduced and high-fidelity models are compared for this case in Figure 5.15. It can be seen from the figure that the flux at this case is almost zero at $t = 203$ s. The maximum absolute difference between the reduced and

high-fidelity models was found to be 7×10^{-3} n/cm^2 s. The complete transient for this case is also shown in Figure 5.15d. The figure shows that the ROM model was able to track the reference solution with great accuracy at the initial and final steady-state while most of the discrepancy was contained in the transient. The second highest error was 1.5%, which was also a point with q_1 near zero ($q_1 = 7 \times 10^{-4}$ n/cm^3 s). The third highest error was 1.2%, which was found at $q_1 = 1.76$ n/cm^3 s and time $t = 470$ s. This case is shown in Figure 5.16, which shows that the error in this case was in the steady-state value rather than the transient. Simulating the 1000 points needed 10 seconds with the reduced model while the high-fidelity model required about 6 hours for the same points.
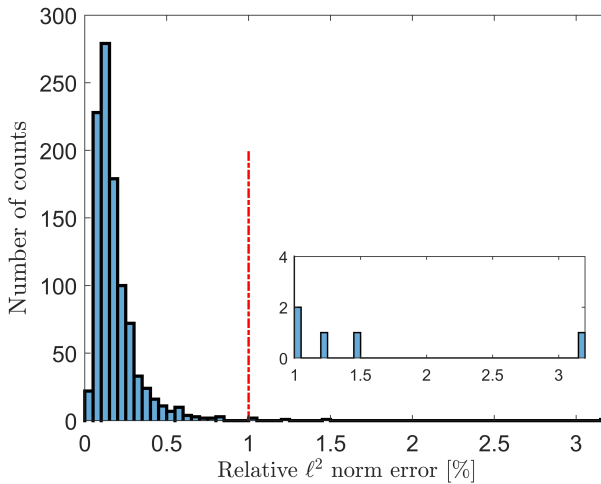

Figure 5.14: Histogram of the relative error resulting from testing the reduced model for the subcritical reactor on 1000 random points. The maximum error was 3.2%. 99.5% of the points resulted in errors below the tolerance of 1%. A close up of the histogram for the values above the tolerance is shown in the box.

5.4. Conclusions

This work presented an approach for time and parameter adaptivity to build a nonintrusive reduced-order model. The approach is an extension of our sparse grid adaptive-POD algorithm to time-dependent parametrized problems. Time was considered as an additional parameter, which enabled the locally adaptive sparse grid algorithm to be applied directly. The adaptivity provided a tool to include more snapshots from important time windows, which reduces the probability of overlooking crucial dynamics in the POD snapshot matrix. Moreover, the efficiency of the construction phase (offline phase) is improved by sampling the high-fidelity model less in time periods of steady-state or slow (smooth) changes.

Three numerical problems were presented to test the proposed approach. The first was a Molenkamp problem with five parameters, which was solved in two settings: a smooth solution and a more challenging steep solution. In this test, we compared the time-adaptive approach with an a-priori fixed sampling approach of the time domain.
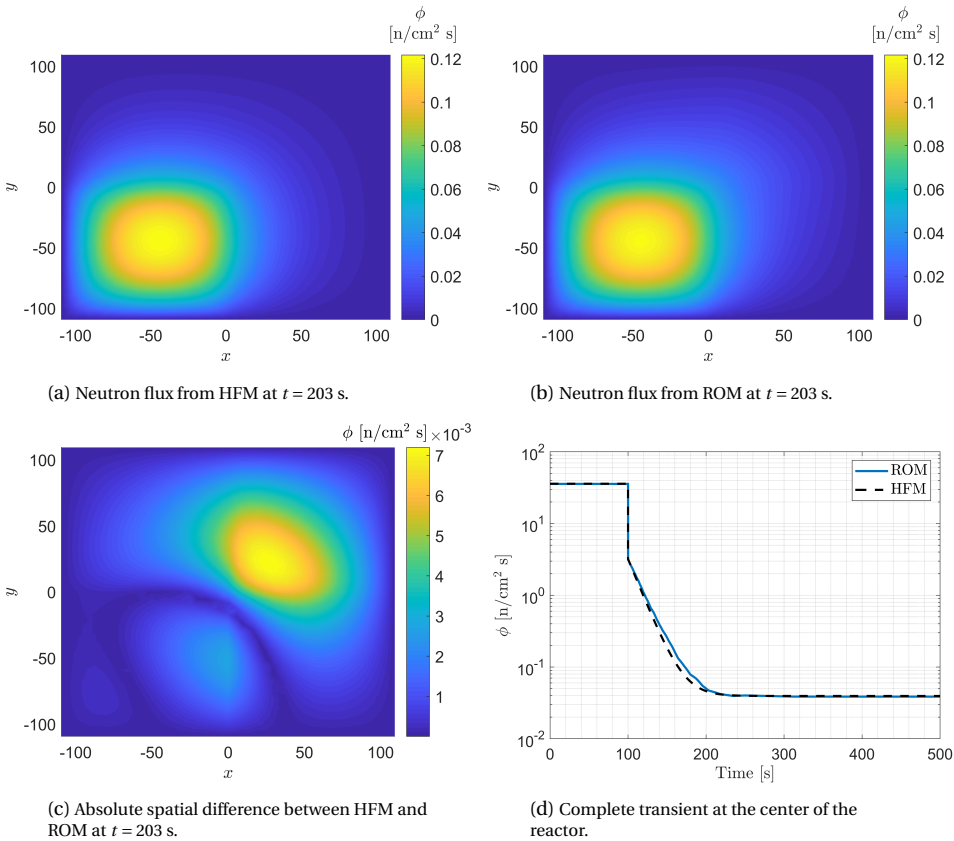
(a) Neutron flux from HFM at $t = 203$ s.

(b) Neutron flux from ROM at $t = 203$ s.

(c) Absolute spatial difference between HFM and ROM at $t = 203$ s.

(d) Complete transient at the center of the reactor.

Figure 5.15: Comparison between ROM and HFM for the subcritical reactor test at the point with the maximum error (3.2%), $q_1 = 1.1 \times 10^{-3}$ n/cm^3 s and time $t = 203$ s.

The results in both settings showed that the time-adaptive approach was more efficient without compromising the accuracy. Additionally, the algorithm was able to identify the linearity of the response with respect to each parameter. The second test was a standard lid-driven cavity problem. For this problem, only time was considered as a parameter. The adaptive algorithm was able to identify that the important time period was the first few seconds of the transient when the flow is still developing.

Moreover, the algorithm recognized that after about t=50, the flow was fully developed and no important snapshots were selected between $t = 50$ and $t = 100$. The reduced model was able to simulate the flow in 10 seconds compared to the 35 CPU-hours needed by the high-fidelity model. The last subcritical reactor test was challenging not only due to the higher dimensionality of the parameter space but also due to the abrupt dynamics at small timescales. The algorithm correctly recognized the time of the important transient following the source perturbation. In addition, the algorithm revealed the region of importance of each parameter and correspondingly concentrated the sampling of the points in these discovered regions. This improved the efficiency of the approach compared to
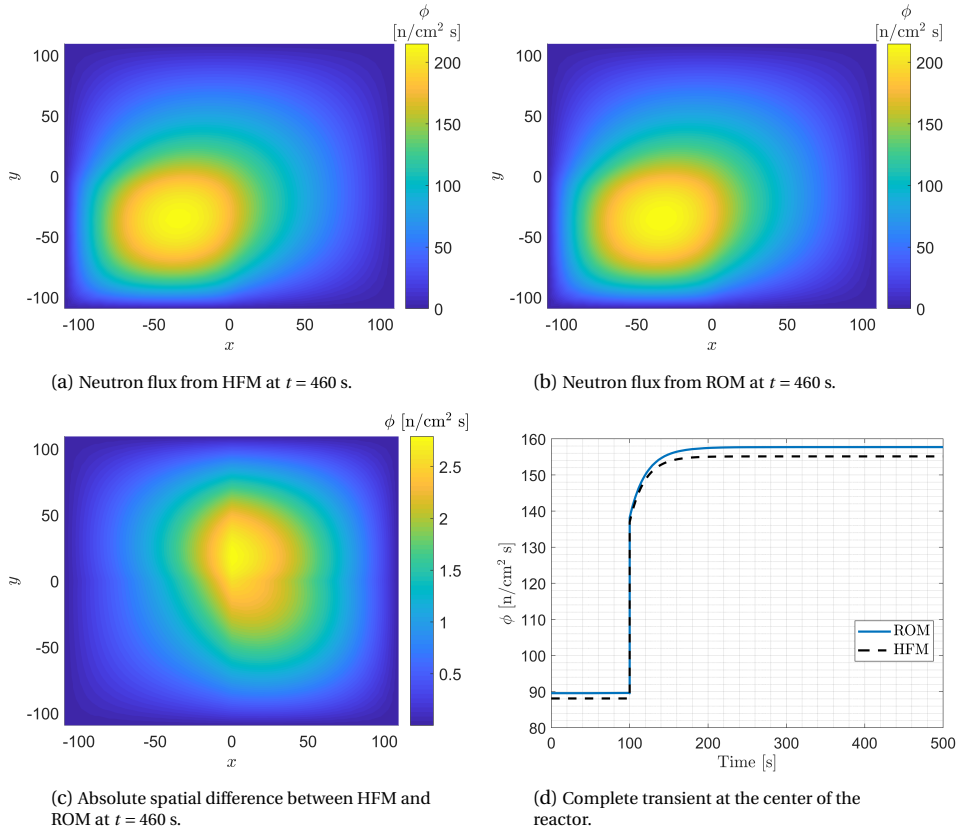
(a) Neutron flux from HFM at $t = 460$ s.

(b) Neutron flux from ROM at $t = 460$ s.

(c) Absolute spatial difference between HFM and ROM at $t = 460$ s.

(d) Complete transient at the center of the reactor.

Figure 5.16: Comparison between ROM and HFM for the subcritical reactor test at the point with error 1.2%, $q_1 = 1.76$ n/cm^3 s and time $t = 460$ s.

non-adaptive techniques. The model was tested on 1000 randomly generated points which were simulated in 10 seconds while the reference model needed about 6 hours to simulate the same points. In all tests, the reduced models built with the time-adaptive approach captured the dynamics of the model with an accuracy that fell within the defined tolerances.

Our approach was nonintrusive which can be applied to a wide range of problems. Despite the fact that nonintrusive approaches do not preserve the physical structure of the system, using adaptive approaches, such as the one presented in this work, provides an insight into the physics of the system by ranking the importance of the parameters or exploring linearity. A challenge for any adaptive method is to scale efficiently to higher dimensional spaces. This issue was addressed in our approach by using the locally adaptive sparse grid approach. However, for the Molenkamp and subcritical reactor tests, the algorithm required a high number of snapshots compared to the number of POD modes selected after truncation. This is an indication that most of the sampled snapshots were needed for the construction of the surrogate model of the POD coefficient more than

revealing additional dynamics of the system. Therefore, an area to study in future work is the use of higher order interpolation models for the POD coefficients with the aim to reduce the number of snapshots and further improve the efficiency. Another interesting area of research to achieve this goal is investigating a space-time decomposition of the basis space or the construction of several local basis spaces tailored to different dynamics instead of a single global basis space.

5

REFERENCES

[1] F. Alsayyari, Z. Perkó, M. Tiberga, J. L. Kloosterman, and D. Lathouwers, *A fully adaptive nonintrusive reduced order modeling approach for parametrized time-dependent problems,* Computer Methods in Applied Mechanics and Engineering (2020), under review.

[2] A. C. Antoulas, D. C. Sorensen, and S. Gugercin, *A survey of model reduction methods for large-scale systems,* Contemporary Mathematics **280**, 193 (2001).

[3] P. Benner, S. Gugercin, and K. Willcox, *A Survey of Projection-Based Model Reduction Methods for Parametric Dynamical Systems,* SIAM review **57**, 483 (2015).

[4] A. Antoulas, A. Ionita, and S. Lefteriu, *On two-variable rational interpolation,* Linear Algebra and its Applications **436**, 2889 (2012).

[5] B. Peherstorfer, S. Gugercin, and K. Willcox, *Data-driven reduced model construction with time-domain loewner models,* SIAM Journal on Scientific Computing **39**, A2152 (2017).

[6] P. J. Schmid, *Dynamic mode decomposition of numerical and experimental data,* Journal of Fluid Mechanics **656**, 5 (2010).

[7] H. V. Ly and H. T. Tran, *Modeling and Control of Physical Processes using Proper Orthogonal Decomposition,* Mathematical and Computer Modelling **33**, 223 (2001).

[8] V. Buljak, *Inverse Analyses with Model Reduction: Proper Orthogonal Decomposition in Structural Mechanics* (Springer, Berlin, 2012).

[9] C. Audouze, F. D. Vuyst, and P. B. Nair, *Nonintrusive reduced-order modeling of parametrized time-dependent partial differential equations,* Numerical Methods for Partial Differential Equations **29**, 1587 (2013).

[10] J. Hesthaven and S. Ubbiali, *Non-intrusive reduced order modeling of nonlinear problems using neural networks,* Journal of Computational Physics **363**, 55 (2018).

[11] N. Nguyen and J. Peraire, *Gaussian functional regression for output prediction: Model assimilation and experimental design,* Journal of Computational Physics **309**, 52 (2016).

[12] M. Xiao, P. Breitkopf, R. F. Coelho, C. Knopf-Lenoir, M. Sidorkiewicz, and P. Villon, *Model reduction by CPOD and Kriging,* Structural and Multidisciplinary Optimization **41**, 555 (2009).

[13] F. Alsayyari, Z. Perkó, D. Lathouwers, and J. L. Kloosterman, *A nonintrusive reduced order modelling approach using Proper Orthogonal Decomposition and locally adaptive sparse grids,* Journal of Computational Physics **399**, 108912 (2019).

[14] 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 **141**, 107321 (2020).

[15] W. Chen, J. S. Hesthaven, B. Junqiang, Y. Qiu, Z. Yang, and Y. Tihao, *Greedy nonintrusive reduced order model for fluid dynamics,* AIAA Journal **56**, 4927 (2018).

[16] D. Xiao, F. Fang, C. Pain, and I. Navon, *A parameterized non-intrusive reduced order model and error analysis for general time-dependent nonlinear partial differential equations and its applications,* Computer Methods in Applied Mechanics and Engineering **317**, 868 (2017).

[17] B. Peherstorfer and K. Willcox, *Data-driven operator inference for nonintrusive projection-based model reduction,* Computer Methods in Applied Mechanics and Engineering **306**, 196 (2016).

[18] E. Qian, B. Kramer, B. Peherstorfer, and K. Willcox, *Lift & learn: Physics-informed machine learning for large-scale nonlinear dynamical systems,* Physica D: Nonlinear Phenomena **406**, 132401 (2020).

[19] M. Guo and J. S. Hesthaven, *Data-driven reduced order modeling for time-dependent problems,* Computer Methods in Applied Mechanics and Engineering **345**, 75 (2019).

[20] A. T. Mohan and D. V. Gaitonde, *A Deep Learning based Approach to Reduced Order Modeling for Turbulent Flow Control using LSTM Neural Networks,* (2018), arXiv:1804.09269 .

[21] F. Regazzoni, L. Dedè, and A. Quarteroni, *Machine learning for fast and reliable solution of time-dependent differential equations,* Journal of Computational Physics **397**, 108852 (2019).

[22] R. Hu, F. Fang, C. Pain, and I. Navon, *Rapid spatio-temporal flood prediction and uncertainty quantification using a deep learning method,* Journal of Hydrology **575**, 911 (2019).

[23] O. San, R. Maulik, and M. Ahmed, *An artificial neural network framework for reduced order modeling of transient flows,* Communications in Nonlinear Science and Numerical Simulation **77**, 271 (2019).

[24] Z. Deng, Y. Chen, Y. Liu, and K. C. Kim, *Time-resolved turbulent velocity field reconstruction using a long short-term memory LSTM-based artificial intelligence framework,* Physics of Fluids **31**, 075108 (2019).

[25] S. Pawar, S. M. Rahman, H. Vaddireddy, O. San, A. Rasheed, and P. Vedula, *A deep learning enabler for nonintrusive reduced order modeling of fluid flows,* Physics of Fluids **31**, 085101 (2019).

[26] H. F. S. Lui and W. R. Wolf, *Construction of reduced-order models for fluid flows using deep feedforward neural networks,* Journal of Fluid Mechanics **872**, 963 (2019).

[27] R. Swischuk, L. Mainini, B. Peherstorfer, and K. Willcox, *Projection-based model reduction: Formulations for physics-based machine learning,* Computers & Fluids **179**, 704 (2019).

5

[28] A. Alwosheel, S. van Cranenburgh, and C. G. Chorus, *Is your dataset big enough? Sample size requirements when using artificial neural networks for discrete choice analysis,* Journal of Choice Modelling **28**, 167 (2018).

[29] S. O. Haykin, *Neural Networks and Learning Machines (3rd Edition)* (Pearson, 2008).

[30] K. Kunisch and S. Volkwein, *Optimal snapshot location for computing POD basis functions,* ESAIM: Mathematical Modelling and Numerical Analysis **44**, 509 (2010).

[31] O. Lass and S. Volkwein, *Adaptive POD basis computation for parametrized nonlinear systems using optimal snapshot location,* Computational Optimization and Applications **58**, 645 (2014).

[32] G. M. Oxberry, T. Kostova-Vassilevska, W. Arrighi, and K. Chand, *Limited-memory adaptive snapshot selection for proper orthogonal decomposition,* International Journal for Numerical Methods in Engineering **109**, 198 (2016).

[33] A. Alla, C. Gräßle, and M. Hinze, *A residual based snapshot location strategy for POD in distributed optimal control of linear parabolic equations,* IFAC-PapersOnLine **49**, 13 (2016).

[34] M. Brand, *Fast low-rank modifications of the thin singular value decomposition,* Linear Algebra and its Applications **415**, 20 (2006).

[35] C. Vreugdenhil and B. Koren, eds., *Numerical methods for advection-diffusion problems* (Vieweg, Germany, 1993).

[36] M. Tiberga, D. Lathouwers, and J. L. Kloosterman, *A multi-physics solver for liquid-fueled fast systems based on the discontinuous Galerkin FEM discretization,* Progress in Nuclear Energy **127**, 103427 (2020).

[37] 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 **311**, 151 (2016).

[38] O. Botella and R. Peyret, *Benchmark spectral results on the lid-driven cavity flow,* Computers & Fluids **27**, 421 (1998).

[39] J. J. Duderstadt and L. J. Hamilton, *Nuclear Reactor Analysis* (John Wiley & Sons, Inc., 1976).

5

6

ANALYSIS OF THE MOLTEN SALT FAST REACTOR USING REDUCED-ORDER MODELS

In this chapter, we present a reduced-order modelling approach to study the Molten Salt Fast Reactor (MSFR). Our approach is nonintrusive and based on the proper orthogonal decomposition method. We include adaptivity in selecting the sampling points both in time and parameter space. Steady-state and transient analysis were both performed using the developed models. In the steady-state analysis, we capture the effect of 30 model parameters on the spatial distributions of fission power and temperature, and on the multiplication factor. The reduced model was then used for uncertainty and sensitivity study of the maximum temperature in the reactor and the multiplication factor. In the transient analysis, the reduced model captured the effect of perturbations in the flow rate of salt in the intermediate circuit on the fission power density and temperature. The reduced models were successfully tested on a set of points that were not part of the snapshots used during the construction stage.

6.1. INTRODUCTION

MOLTEN salt reactors have gained interest due to their potential safety, reliability, and sustainability [2]. Different designs of this concept have been proposed in the literature [3]. In this work, we consider the Molten Salt Fast Reactor (MSFR) [4]. A key design feature of this unmoderated reactor is the use of a liquid salt fuel, which also plays the role of the coolant. This design introduces a unique modelling challenge because of the tightly coupled neutronics and thermal-hydraulics phenomena (e.g., transport of delayed neutron precursors, distributed thermal energy deposition directly in the coolant, a strong negative temperature feedback coefficient). To address these challenges, high-fidelity coupled models are used to provide an insight into the behaviour of the reactor. For safety assessment applications, an accurate and explicit quantification of the propagation of uncertainties through these complex models is required [5]. Quantifying uncertainties and analysing sensitivities in reactor physics can be accomplished using adjoint methods as in [6, 7]. However, adjoint-based methods require the availability of an adjoint solver, which might not always be feasible for coupled problems. Another approach is using forward-based methods, which requires repeated evaluation of the high-fidelity model for different parameter configurations [8]. However, coupled high-fidelity models are computationally demanding, which renders their use to be expensive for repeated evaluations. For such applications, *reduced-order modelling* (ROM) techniques can be used to simplify the high-fidelity model and produce an efficient, cheap, and accurate model of the system.

While different ROM approaches can be found in literature, proper orthogonal decomposition (POD) is the method most suited for nonlinear systems [9–11]. In the POD approach, a reduced basis space for the system is built using snapshots of the high-fidelity model. In nuclear reactor application, POD has been applied to solve criticality eigenvalue problems [12–15], for fuel pin reactor core calculations [16], in fuel burnup calculations [17], in thermal hydraulics modeling [18], in stability analysis [19, 20], in spent fuel pool modeling [21], and to model the lead cooled fast reactor [22]. In all of these applications, the original set of model equations were projected onto the constructed reduced basis. Projection-based approaches are intrusive because of the need to access the operator of the original high-fidelity model in order to employ the projection. In cases where the high-fidelity model is a legacy coupled solver or closed-source, intrusive approaches are not applicable. Moreover, the stability of the high-fidelity model is not preserved in the reduced model with the use of projection-based POD methods [23].

To overcome these issues, nonintrusive approaches have been proposed in the literature, which are techniques that can be wrapped around the high-fidelity solver and avoid the need to access the original set of equations. Nonintrusive spectral projection methods such as polynomial chaos expansion (PCE) is one approach for uncertainty quantification applications, which has been applied to a molten salt fuelled system [24]. However, such methods do not reduce the dimensionality of the high-fidelity model. Dynamic Mode Decomposition (DMD) is a nonintrusive model order reduction method that has been employed to model the MSFR [25]. DMD constructs a linear operator from a sequence of snapshots of the system. However, while DMD is able to produce a reduced model for control applications and transients analysis, it is not applied for uncertainty and sensitivity analysis because the constructed operator is not parametrized [26]. An alternative

ROM approach is through a nonintrusive POD implementation, where a surrogate model is trained to compute the POD coefficients instead of projecting the model equation onto the reduced basis. In nuclear reactor applications, a Range Finding Algorithm (RFA) has been used in [27] to build the reduced basis (referred to as active subspace) combined with a simple polynomial surrogate for the POD coefficients. In addition, we have proposed a nonintrusive adaptive POD algorithm in [28]. A key difference in our developed algorithm is the adaptive sampling of the high-fidelity model as opposed to random or a priori uniform sampling employed by the RFA. Our algorithm is based on adaptive sparse grids technique, which is suited for problems with higher dimensional input spaces. An uncertainty and sensitivity application of the algorithm on a simplified two-dimensional molten salt fuelled system was presented in [29]. We have presented an extension of the algorithm in [30] to deal with time-dependent parametrized problems. In this work, we investigate the MSFR reactor using the developed algorithm, which demonstrates the capability of the algorithm on a large-scale full three-dimensional MSFR model. We consider two applications. The first is a steady-state reduced model for uncertainty and sensitivity analysis of 30 model parameters. The second is a transient reduced model, which can be used for transient analysis and control applications.

The remainder of the chapter is organized as follows: Section 6.2 presents an introduction to the problem formulation and the POD method. A summary of the adaptive sampling algorithm is presented in section 6.3. Then, a description of the MSFR model is given in Section 6.4. The results for the steady-state analysis along with the uncertainty and sensitivity study are given in Section 6.5. We present the results of the transient study in Section 6.6. Finally, conclusions are discussed in Section 6.7.

6.2. PROPER ORTHOGONAL DECOMPOSITION

Consider a general high-fidelity model that is described by a function $f : \mathbb{R}^d \to \mathbb{R}^n$, which maps input parameter $\boldsymbol{p} \in \mathbb{R}^d$ to a state (field) vector $\boldsymbol{\phi} \in \mathbb{R}^n$, where d is the size of the input space (representing, for example, material properties, and boundary conditions), and n is the size of the state space (representing, for example, space, angle, and energy). We consider f as a black-box model, which can either describe single physics or coupled multi-physics code, that is,

$$\boldsymbol{\phi} = f(\boldsymbol{p}). \tag{6.1}$$

The high-fidelity model f can be evaluated at any given $\boldsymbol{p}_q$ and produce an output $\boldsymbol{\phi}_q$ (i.e., $\boldsymbol{\phi}_q = f(\boldsymbol{p}_q)$). We are interested in finding an approximation for $\boldsymbol{\phi}_q$ without the use of the computationally intensive f. For this task, we build a reduced model that produces an approximation $\tilde{\boldsymbol{\phi}}_q$, which takes the form of

$$\tilde{\boldsymbol{\phi}}_q = \sum_{g=1}^{r} c_g(\boldsymbol{p}_q)\boldsymbol{u}_g, \tag{6.2}$$

where $\boldsymbol{u}_g$ is the g-th basis vector in the reduced basis space, $c_g(\boldsymbol{p})$ is the corresponding coefficient, which is a function of the parameter $\boldsymbol{p}$, and r is the size of the reduced basis space. The POD approach constructs a reduced basis space from snapshots of the state

vectors $\boldsymbol{\phi}$ such that the error is minimized in a least square sense, that is, it solves the minimization problem

$$\min_{\boldsymbol{u}_g} \left\| \boldsymbol{\phi} - \sum_{g=1}^{r} c_g(\boldsymbol{p})\boldsymbol{u}_g \right\|_{\ell^2}. \tag{6.3}$$

The basis vectors are defined to be orthonormal (i.e., $< \boldsymbol{u}_g, \boldsymbol{u}_h >= \delta_{gh}$, where $< .,. >$ indicates the scalar product). The reduced basis space can be found using the method of snapshots. Let M be a matrix collecting snapshots of the solution $\boldsymbol{\phi}$ at some selected values of $\boldsymbol{p}$, that is,

$$\mathbf{M} = [\boldsymbol{\phi}_1, \boldsymbol{\phi}_2, \ldots, \boldsymbol{\phi}_s] \quad \in \mathbb{R}^{n \times s}, \tag{6.4}$$

where $\boldsymbol{\phi}_q$ is the solution of the high-fidelity model (f) at parameter value $\boldsymbol{p}_q$, and s is the number of selected snapshots. Note that for transient problems where f computes a solution $\boldsymbol{\phi}$ that is time-dependent, we consider the time to be a parameter, rather than an independent variable – that is, $\boldsymbol{p}$ includes time as one of the parameters. Such a formulation allows for time adaptive sampling as described in [30].

The basis vectors are then found to be the left singular vectors of the singular value decomposition (SVD) applied to the matrix M, that is, let M be decomposed using SVD as $M = USV$, then $\boldsymbol{u}_g$ are the first r column vectors of U. The size of the reduced space is determined by r, which can be chosen to truncate U such that the sum of the squared singular values (σ) corresponding to the neglected singular vectors is below a predefined threshold γ_{tr}.

$$\frac{\sum_{g=r+1}^{n} \sigma_g^2}{\sum_{g=1}^{n} \sigma_g^2} < \gamma_{\text{tr}}. \tag{6.5}$$

Once the basis vectors are known, we can use the orthogonality of the space to compute the coefficients at the sampled point ($\boldsymbol{p}_q$) as

$$c_g(\boldsymbol{p}_q) =< \boldsymbol{u}_g, \boldsymbol{\phi}_q >. \tag{6.6}$$

6.3. ADAPTIVE SAMPLING

Using the values of the coefficient $c_g(\boldsymbol{p})$ at the sampled points, one can train a surrogate model to compute the solution of the system at any point. Different surrogate models can be employed for such a task. To deal with higher dimensional input spaces, we choose the Smolyak sparse grid interpolation technique. The hierarchical structure of the interpolant allows for the desired adaptive strategy, which can further reduce the burden of dealing with high dimensional spaces. This section is devoted to summarize the developed adaptive algorithm, which is detailed in [28].

Our implementation of the sparse grid technique uses localized adaptive sampling. Without loss of generality, the d-dimensional space of the input parameter $\boldsymbol{p}$ is mapped to a unitary hypercube $[0, 1]^d$. The sparse grid technique first generates unidimensional nodes along each dimension. Then, points in the parameter space are formed by a specific combination of the generated unidimensional nodes. Different choices for the

unidimensional node generation rule are possible. However, we choose the equidistant rule to increase the separation between points. The unidimensional nodes are arranged in a tree structure as shown in Figure 6.1. Each level is assigned an index i and contains several nodes. Nodes are added at each level at half the distances between the nodes from the previous levels. Each node is connected to two children at the next level $(i + 1)$ and one father from the previous level $(i - 1)$. The root node is considered a father for itself. There is, however, an exception at level $i = 2$, where each node has one child because these nodes mark the boundaries of the unit hypercube.
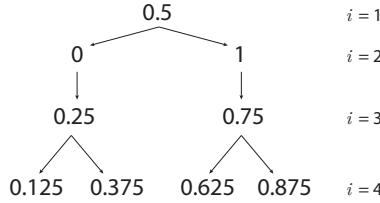

Figure 6.1: Illustration of the first 4 levels of the tree structure, where 0.5 is the root of the tree and nodes are added at half the distances between the previous nodes. Each node has 2 children except the nodes at level 2 where each has one child only.

Each node in the tree can be uniquely identified with the level index i and an index j. For each node from the tree p^i_j along dimension l, we can build a basis function $a^i_{p^i_j}(p)$ as

$$a^1_{p^1} = 1 \quad \text{if } i = 1,$$

$$a^i_{p^i_j}(p) = \begin{cases} 1 - (m^i - 1) \cdot |p - p^i_j|, & \text{if } |p - p^i_j| < \dfrac{1}{m^i - 1}, \\ 0, & \text{otherwise}, \end{cases}$$

(6.7)

where the dependence on the dimension l is dropped for notational convenience. The level parameter m^i is defined as follows:

$$m^i = \begin{cases} 1 & \text{if } i = 1, \\ 2^{i-1} + 1 & \text{if } i > 1. \end{cases}$$

(6.8)

A point $\boldsymbol{p}$ in parameter space is formed by combining nodes from all dimensions (i.e., $\boldsymbol{p} = (p^{i_1}_{j_1}, \dots p^{i_d}_{j_d})$). Extending the tree structure to points in parameter space allows us to define *forward* points. The first forward points for a point $\boldsymbol{p}$ along the first dimension are $(b(p^{i_1}_{j_1}), \dots p^{i_d}_{j_d})$, where $b(p)$ is a function that returns the children of the node p. In general, the forward points along the l-th dimension are $(p^{i_1}_{j_1}, \dots, b(p^{i_l}_{j_l}), \dots p^{i_d}_{j_d})$. Because each node has at most two children, each point has at most $2d$ forward points. We can also define *backward* points in the same manner by applying a function $b^*(p)$, which returns the father of the node. Therefore, each point has at most d backward points. Recursively generating the forward points creates the set of *descendant* points. On the other hand, the *ancestor* points for a point $\boldsymbol{p}$ are formed by recursively generating backward points until the root point $(0.5, \dots, 0.5)$ is reached.

The adaptive algorithm is iterative. In every iteration, a subset of points is selected and marked *important* from a set of trial points. Let the iteration number be k and a set that collects the selected important points at each iteration k be $\mathcal{Z}^k$. The union of the important points sets ($\mathcal{Z}^k$) from all iterations up to k forms the set of selected grid points $\mathcal{X}^k$, that is,

$$\mathcal{X}^k = \bigcup_{h=0}^{k} \mathcal{Z}^h. \tag{6.9}$$

A point $\boldsymbol{p}_q$ is marked as important if it satisfies one of two conditions. The first is to have an approximation error e_q above a defined threshold γ_{int} with all ancestors of that points included in $\mathcal{X}^k$. The second is to have a descendant point with an error above γ_{int}. Points that do not meet any of the conditions are added to a set of inactive points. The error is computed in the ℓ^2 norm as

$$e_q^k = \frac{\left\| \boldsymbol{\phi}_q - \tilde{\boldsymbol{\phi}}_q^k \right\|_{\ell^2}}{\left\| \boldsymbol{\phi}_q \right\|_{\ell^2} + \epsilon}, \tag{6.10}$$

where $\boldsymbol{\phi}_q$ is the solution returned by the high-fidelity code (f) at point $\boldsymbol{p}_q$, $\tilde{\boldsymbol{\phi}}_q^k$ is the approximation produced by the reduced model at the point $\boldsymbol{p}_q$ and iteration k, and ϵ is an offset introduced for cases when the norm of the solution is near-zero.

At iteration k, a surrogate model for the POD coefficient $c_g(\boldsymbol{p})$ is built as

$$A_{k,d}(c)(\boldsymbol{p}) = A_{k-1,d}(c)(\boldsymbol{p}) + \Delta A_{k,d}(c)(\boldsymbol{p}) , \tag{6.11}$$

where $A_{k,d}(c)(\boldsymbol{p})$ is the Smolyak interpolation operator applied to $c(\boldsymbol{p})$ that depends on the iteration k and dimension of the input space d. For the initialization ($k = 0$), we enforce $A_{0,d}(c)(\boldsymbol{p})$ to be zero. The term $\Delta A_{k,d}(c)(\boldsymbol{p})$ is defined as

$$\Delta A_{k,d}(c)(\boldsymbol{p}) = \sum_{n=1}^{m_k^{\Delta}} w_n^k \Theta_n(\boldsymbol{p}), \tag{6.12}$$

where m_k^{Δ} is the cardinality of the set $\mathcal{Z}^k$. The function $\Theta_n(\boldsymbol{p})$ is the d-variate basis function for the point $\boldsymbol{p}_n \in \mathcal{Z}^k$,

$$\Theta_n(\boldsymbol{p}) = \prod_{l=1}^{d} a_{p_{n,l}^{i_l}}^{i_l}(p_l), \tag{6.13}$$

The surplus w_n^k is defined as the difference between the interpolated value and the true value of the coefficient at $\boldsymbol{p}_n$,

$$w_n^k = c(\boldsymbol{p}_n) - A_{k-1,d}(c)(\boldsymbol{p}_n). \tag{6.14}$$

ALGORITHM
In the initialization step ($k = 0$), we select the root point at the center of the hypercube $(0.5, 0.5, \ldots, 0.5)$ and evaluate the high-fidelity model f. That point is added to the important set $\mathcal{Z}^0$, and the snapshot is added to the matrix M. Then, set $k = 1$ and do

1. Generate the forward points of the points in $\mathcal{Z}^{k-1}$;

2. Add points that have all their backward points in $\mathcal{X}^{k-1}$ (defined as in Equation 6.9) to the trial set;

3. Evaluate the high-fidelity model f at the points in the trial set;

4. Compute the coefficients $c_g(\boldsymbol{p})$ at the points in the trial set using $A_{k-1,d}(c)(\boldsymbol{p})$;

5. Compute the error at every point in the trial set using Equation 6.10;

6. If the maximum found error was below a prescribed global tolerance ζ, terminate; Otherwise, continue;

7. Find the important points $\mathcal{Z}^k$;

8. Add the snapshots corresponding to the important points in $\mathcal{Z}^k$ to the matrix M;

9. Perform SVD on the matrix M to extract the reduced basis;

10. Use Equation 6.11 - 6.14 to construct $A_{k,d}(c)(\boldsymbol{p})$;

11. Set $k \leftarrow k+1$ and go to Step 1.

For a detailed description of the algorithm, see [28].

6.4. MSFR MODEL

This section presents the MSFR main design features and modeling approach. A more detailed description of the model can be found in [31] and the references within.

A schematic illustration of the fuel circuit is shown in Figure 6.2. The reactor core is a toroidal cavity where the liquid fuel salt (a mixture of lithium, thorium, and fissile nuclides fluorides) can flow freely without any moderator or control rod. Sixteen identical sectors branch out from the central cavity. Each sector contains a pump, a heat exchanger, and a unit for the separation and treatment of the helium gas dispersed in the fuel salt to control reactivity and remove metallic fission products [32]. A blanket with fertile salt surrounds the cavity while reflectors are placed at the top and bottom of the core. The heat exchangers transfer thermal energy to the intermediate circuit filled with inert molten salt, which in turn, delivers the heat to the energy conversion system consisting
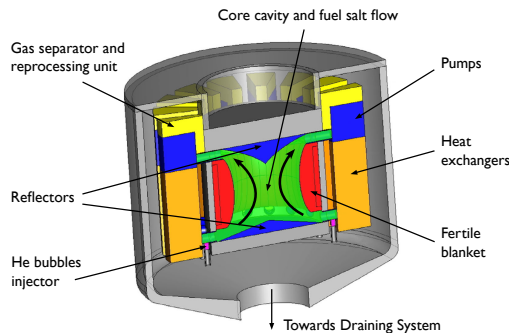

Figure 6.2: A schematic illustration of the main design features of the MSFR fuel circuit.

Table 6.1: Main design parameters for MSFR model in the nominal case [4, 33].

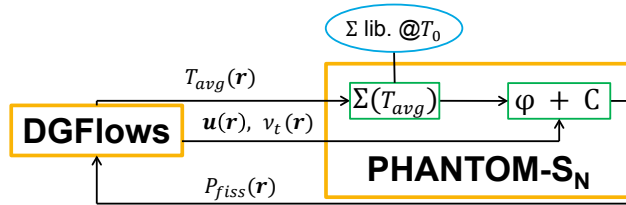| Parameter	Value
Total thermal power [MW]	3000
Total fuel salt volume [m^3]	18
Fuel salt circulation time [s]	4
Average fuel salt temperature [K]	973.15
Minimum fuel salt temperature [K]	923.15
Average intermediate salt temperature [K]	908.15
Pressure drop across heat exchanger [bar]	4
Volumetric heat transfer coefficient with intermediate salt $\left[\text{MW/m}^3\text{K}\right]$	19.95

Figure 6.3: Computational scheme of the multi-physics tool constituting the high-fidelity MSFR model. DGFlows is the CFD code, while PHANTOM-S$_N$ is the neutronics code. Data are exchanged at each iteration between the two solvers to model the coupled physics phenomena characterizing the MSFR [31].

of a conventional Joule-Brayton cycle. Table 6.1 summarizes the main MSFR design parameters.

The reactor was modelled using an in-house multi-physics tool, which couples an incompressible Reynolds-averaged Navier-Stokes solver (DGFlows) with an S_N multi-group neutronics code (PHANTOM-S$_N$). The latter is equipped with transport equations for the delayed neutron precursors to model their movement. Both codes employ the discontinuous Galerkin finite element method for space discretization and implicit backward differentiation formulae (BDF) time schemes. Figure 6.3 displays the structure of the multi-physics tool and the data exchanged between the codes. PHANTOM-S$_N$ receives the average temperature on each element (T_{avg}) and corrects the cross sections accordingly with respect to a library at the reference temperature T_0. The corrections take into account the effects of density feedback (which has a linear dependence on temperature) and Doppler feedback (which has a logarithmic dependence). The velocity and turbulent viscosity fields (**u** and v_t) are also taken from DGFlows as another input to solve the delayed neutron precursors equation. Then, the fission power density (P_{fiss}) is transferred to DGFlows as it constitutes the right-hand side of the energy equation. The steady-state solution is sought by iterating the solvers until convergence (four iterations are typically sufficient). On the other hand, transient simulations are performed with a loose-coupling strategy, first computing a time-step with DGFlows and then calling PHANTOM-S$_N$ to solve the neutronics problem. we refer the reader to [31, 34] for a more comprehensive description of the coupled code.
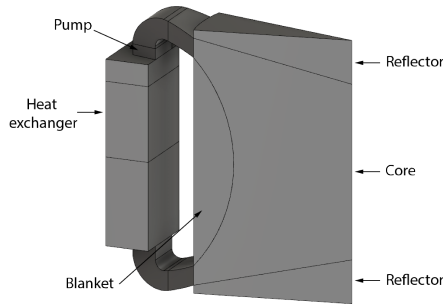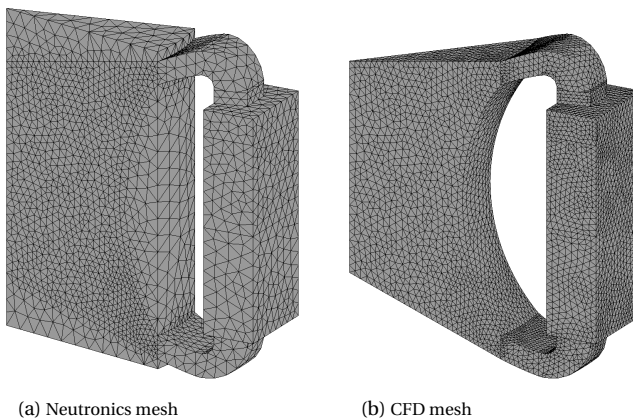
Figure 6.4: Geometry of the MSFR recirculation loop used for simulations, showing the main regions considered in the model [31].

(a) Neutronics mesh   (b) CFD mesh

Figure 6.5: Mesh adopted for the MSFR model. The neutronics mesh (left), corresponding to the master-mesh, consists of 26072 tetrahedra (21489 in the fuel salt domain), while the CFD mesh (right) has 55243 elements. The latter is derived by refining the former once uniformly in the outer-core region.

Because of the symmetry in the reactor design, we modelled only one recirculation loop. The geometry considered is reported in Figure 6.4. While the neutron flux is calculated in the reflectors and the blanket, the CFD code (DGFlows) neglects heat transfer in these regions. Figure 6.5 shows the meshes used in this study. Neutronics calculations were performed on an unstructured mesh consisting of 26072 tetrahedra (of which 21489 are in the fuel salt domain). This mesh is finer in the core region, where neutron importance is high, and coarser in the external sector. This master mesh was then refined once uniformly in the outer-core region to obtain the CFD mesh, which consists of 55243 elements. A second-order polynomial was used for the velocity discretization, while a first order polynomial was used for all other quantities.

The fuel salt composition is reported in Table 6.2, along with some physical properties that were fixed and selected for the uncertainty analysis study of Section 6.5. The neutron energy range was condensed into six groups, with boundaries shown in Table 6.3. Delayed neutron precursors were grouped into eight families. All neutronics data were evaluated

Table 6.2: Properties of the fuel salt mixture [37].

Property	Value or equation of state
Composition [% mol]	LiF(77.5)-ThF$_4$(6.6)-^{enr}UF$_4$(12.3)-(Pu-MA)F$_3$(3.6)
Density [kg/m^3]	4306.7
Dynamic viscosity [Pa s]	$6.187 \times 10^{-4} \exp\{(772.2/(T(\text{K}) - 765.2))\}$
Thermal expansion coefficient [K^{-1}]	1.9119×10^{-4}
Melting point [K]	854.15

Table 6.3: Upper energy bound for each group and average flux of each group for the nominal case.

Energy group	1	2	3	4	5	6
Upper energy bound [keV]	20000	2231	497.9	2.479	5.531	0.7485
Average group flux [cm^{-2}s^{-1}] $\times 10^{14}$	0.51	2.05	5.14	3.51	2.20	0.46

at temperature $T_0 = 900\,\text{K}$ with Serpent [35] using the JEFF-3.1.1 data library [36].

For the CFD calculations, symmetry boundary conditions were imposed at the wedge sides, and standard wall-functions with adiabatic conditions were assumed at all walls. For the neutronics calculations, reflective boundary conditions were assumed at the sides of the wedge, while vacuum conditions were imposed elsewhere. For the transport of the precursors within the fuel salt, homogeneous Neumann and no-inflow conditions were imposed at all walls.

Lacking detailed design specifications for the primary heat exchanger, salt cooling was modelled via a volumetric heat sink term equal to $h_{\text{int},0}(T_{\text{int},0} - T)$, where $h_{\text{int},0}$ and $T_{\text{int},0}$ are the nominal volumetric heat transfer coefficient and average temperature of the intermediate salt whose values are reported in Table 6.1. The fuel pump was modeled with a momentum source term, and buoyancy was taken into account through the Boussinesq approximation (the reference density and thermal expansion coefficient are reported in Table 6.2). We considered the flow field to be fixed at the nominal state because natural convection has a negligible contribution to the total nominal flow rate, and the pump specifications were fixed throughout the analyses in this work. Moreover, since we are interested in reactor steady-state and operational transients, decay heat was not taken into account. For the transient calculations described in Section 6.6, the time-dependent equations were discretized with the second-order BDF scheme with a fixed time-step size of 0.1 s.

6.5. STEADY-STATE ANALYSIS

In this section, we study the effect of selected parameters of interest on three outputs: fission power density (P_{fiss}), temperature distribution (T), and the multiplication factor, k_{eff}. The fission power density P_{fiss} is computed on the neutronics mesh with 104288 degrees of freedom (DOF) while the temperature T is computed on the CFD mesh with

220972 DOF. This analysis is performed on the steady-state model of the MSFR with a fixed reactor power of 3000 MW. However, since we are modeling 1/16th of the reactor, the observed nominal power is 187.5 MW. The selected parameters to be investigated were specific heat capacity of the fuel salt (c_p), heat conductivity of the fuel salt (κ), fission cross sections for the six energy groups ($\Sigma_{f,g}$), capture cross sections for the six energy groups ($\Sigma_{c,g}$), delayed neutron fractions for eight families of precursors (β_i), and corresponding decay constants (λ_i). The nominal values of the selected parameters are reported in Table 6.4. These nominal values are theoretical as the actual salt properties and the associated uncertainties are still under investigation. However, for the propose of this work, we assume an uncertainty following a Gaussian distribution for all parameters with a mean (μ) equal to the nominal value and a standard deviation (σ) taken as 5% of the mean.

6.5.1. CONSTRUCTION OF THE REDUCED-ORDER MODEL

Our implementation of the adaptive approach was developed for bounded input domains. For this reason, we consider the Gaussian distribution to be truncated. Truncating the Gaussian distribution, in this case, is a valid approximation because the parameter uncertainties we are considering are epistemic. Therefore, we are not altering a random process but rather limiting the scope of analysis to the region with the highest probability of having the true value. Moreover, truncating the distribution prevents unphysical values of the parameter, such as negative β. The truncation was selected to be at 3σ, retaining 99.7% of the probability range. This implies that the range of variation for all parameters is set to be $\pm 15\%$ of the nominal value. We first constructed the reduced model to be uniformly accurate within the defined range, then use the reduced model for the uncertainty and sensitivity analysis employing the corresponding probability distribution of each parameter. A separate reduced model was built for each output. The global relative tolerance ζ was set to be 1% for P_{fiss}, 0.1% for T, and 50 pcm for k_{eff}. The interpolation threshold γ_{int} was 1×10^{-3} P_{fiss}, 1×10^{-2} for T, and 5×10^{-5} for k_{eff}. The POD truncation threshold γ_{tr} was 10^{-12} for all outputs.

After construction, each reduced model is tested on 1000 independent points that were not part of the snapshots generated during the constructions. Latin Hypercube Sampling (LHS) was used to draw the random testing points from the input space. Table 6.5 summarizes the test results for each model. It can be seen that all models resulted in a maximum relative error that was below the set tolerance ζ. These results certify that the reduced models are an accurate representation of the high-fidelity MSFR model with in the desired tolerances. Figure 6.6 compares the reduced model for P_{fiss} with the reference model at the point with maximum error. The difference is seen to be maximum in the central region of the reactor core, where the flux is maximum. Figure 6.7 shows the comparison at the maximum error for T. The maximum difference for this case is observed at the bottom of the core, where the relative error locally is around 0.3%. A single high-fidelity computation requires about 4 CPU-hours (performed on a Linux cluster) whereas the reduced model produces the results in less than a second.

Table 6.5 reports also the number of POD modes after truncation, representing the number of DOF in the reduced space. The dimensionality of P_{fiss} was reduced from 104288 DOF in the physical space to 10 DOF in the reduced space whereas T has 3 DOF in

Table 6.4: Nominal values of the selected parameters.

Parameter	Nominal value	Parameter	Nominal value
c_p [J/kgK]	1.59×10^3	β_2	7.14×10^{-4}
κ [W/mK]	1.7×10^0	β_3	3.59×10^{-4}
$\Sigma_{f,1}$ [cm^{-1}]	4.45×10^{-3}	β_4	7.94×10^{-4}
$\Sigma_{f,2}$ [cm^{-1}]	2.52×10^{-3}	β_5	1.47×10^{-3}
$\Sigma_{f,3}$ [cm^{-1}]	1.80×10^{-3}	β_6	5.14×10^{-4}
$\Sigma_{f,4}$ [cm^{-1}]	2.62×10^{-3}	β_7	4.65×10^{-4}
$\Sigma_{f,5}$ [cm^{-1}]	5.20×10^{-3}	β_8	1.51×10^{-4}
$\Sigma_{f,6}$ [cm^{-1}]	1.39×10^{-2}	λ_1 [s^{-1}]	1.25×10^{-2}
$\Sigma_{c,1}$ [cm^{-1}]	1.99×10^{-3}	λ_2 [s^{-1}]	2.83×10^{-2}
$\Sigma_{c,2}$ [cm^{-1}]	7.41×10^{-4}	λ_3 [s^{-1}]	4.25×10^{-2}
$\Sigma_{c,3}$ [cm^{-1}]	2.15×10^{-3}	λ_4 [s^{-1}]	1.33×10^{-1}
$\Sigma_{c,4}$ [cm^{-1}]	5.10×10^{-3}	λ_5 [s^{-1}]	2.92×10^{-1}
$\Sigma_{c,5}$ [cm^{-1}]	1.02×10^{-2}	λ_6 [s^{-1}]	6.66×10^{-1}
$\Sigma_{c,6}$ [cm^{-1}]	2.48×10^{-2}	λ_7 [s^{-1}]	1.63×10^0
β_1	1.23×10^{-4}	λ_8 [s^{-1}]	3.55×10^0

6

Table 6.5: Results for the steady-state reduced models for each output showing the maximum relative ℓ^2 error after a test on 1000 independent points, the number of POD modes after truncation, the number of evaluations to construct each model, and the final number of points in $\mathcal{X}^k$.

	P_{fiss}	T	k_{eff}
Maximum relative ℓ^2 error	0.24%	0.02%	37 pcm
Required tolerance ζ	1%	0.1%	50 pcm
Number of POD modes	10	3	–
Total number of evaluations	61	63	1639
Number of selected points in $\mathcal{X}^k$	15	5	227

the reduced space compared to 220972 DOF in the high-fidelity model. Note that k_{eff} is a single-valued response, and reducing the dimensionality is not applicable for this model.

The total number of evaluations requested by the algorithm during the construction stage is also reported in Table 6.5. The models for P_{fiss} and T needed 61, and 63 evaluations respectively, while the k_{eff} model needed 1639 evaluations. The increased number of evaluations reflects the strong nonlinearity of k_{eff} with respect to the input parameters compared to P_{fiss} and T. This finding is supported by the number of unique nodes per dimension for each output, which is reported in Table 6.6. This number is the result of projecting the final grid points in $\mathcal{X}^k$ onto each dimension. It is a measure of the nonlinearity in the output with respect to the dimension as captured by the reduced model. A value of 1 indicates that the reduced model considered that dimension to have a negligible effect on the output. A value of 3 means that the output is linear or piecewise linear with respect to that input parameter because the reduced model used 3 nodes (the root node
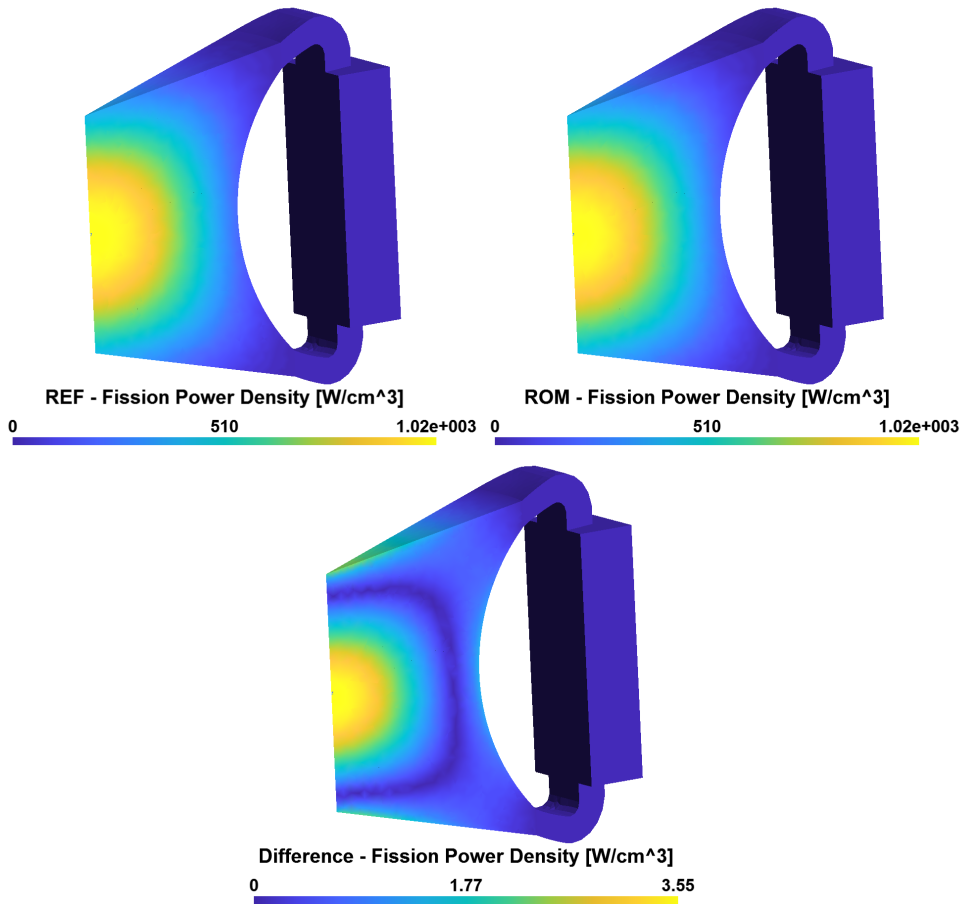
Figure 6.6: Fission power density distribution at the point of maximum error showing the reference model (top left), the ROM model (top right), and the distribution of the absolute difference (bottom). The relative ℓ^2 error was 0.24%.

and the first two children) to construct a linear or a piecewise linear interpolant and no further refinement was required along that dimension. A value of 5 or more indicates that the interpolant along that dimension constructed a nonlinear interpolant between the nodes. In general, the nonlinearity in the interpolant is proportional to this number.

Table 6.6 shows that the fission power density was found to be linear with respect to the specific heat capacity c_p, which is explained by the temperature feedback effect. The fission cross sections of groups 2 to 6 also have a linear effect on the fission power density, which is expected since the fission power is proportional to $\Sigma_{f,g}$. The fission cross section of the first group, however, was observed to have a negligible effect within the tolerances on $\boldsymbol{P}_{\text{fiss}}$. This can be explained by the lower magnitude of the first-group flux compared to the rest of the groups, as shown in the average flux value per group in Table 6.3. This has an effect of a lower weight in the calculation of the fission power
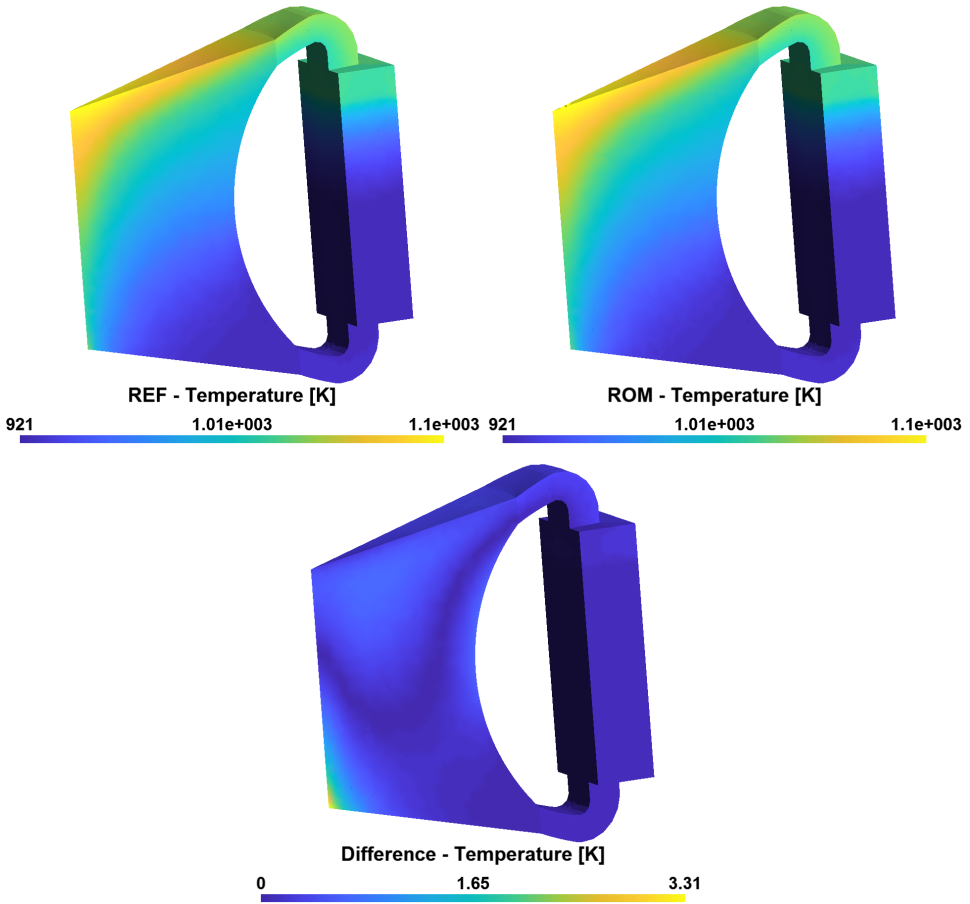
Figure 6.7: Temperature distribution at the point of maximum error showing the reference model (top left), the ROM model (top right), and the distribution of the absolute difference (bottom). The relative ℓ^2 error was 0.02%.

density. The capture cross section of the most thermal group is the only capture cross section that has a linear effect on P_{fiss}. This is due to its larger nominal value resulting in a larger range of variations compared to the other groups. The rest of the parameters had a negligible effect on P_{fiss} and were considered as constants. The temperature is shown to be nonlinear with respect to c_p and unaffected by the rest of the parameters within the defined tolerances. The multiplication factor is nonlinear with respect to the cross section of groups 3 to 6 both fission and capture while having a piecewise linear interpolant with respect to the two most energetic groups. The specific heat capacity is also seen to have a linear effect on k_{eff} through the temperature feedback. The delayed neutron fractions of families 2, 4, and 5 were the only families with a significant effect on k_{eff}. This, however, is due to the larger nominal value of the theses parameters, which results in a larger range of variations compared to the rest of the families. The decay constant is shown to be taken as

a constant for all families with 1 node each, which indicates that within the set tolerance of 50 pcm and a range of variation of $\pm 15\%$, λ_i had no effect on k_{eff}.

Table 6.6: Number of unique nodes per dimension for each output. A value of 1 indicates that the output is constant with respect to the parameter. A value of 3 signals that the output is piecewise linear in the parameter. A higher value indicates that output is nonlinear in the parameter.

Parameter	P_{fiss}	T	k_{eff}	Parameter	P_{fiss}	T	k_{eff}
c_p	3	5	3	β_2	1	1	3
κ	1	1	1	β_3	1	1	1
$\Sigma_{f,1}$	1	1	3	β_4	1	1	3
$\Sigma_{f,2}$	3	1	3	β_5	1	1	3
$\Sigma_{f,3}$	3	1	5	β_6	1	1	1
$\Sigma_{f,4}$	3	1	5	β_7	1	1	1
$\Sigma_{f,5}$	3	1	7	β_8	1	1	1
$\Sigma_{f,6}$	3	1	5	λ_1	1	1	1
$\Sigma_{c,1}$	1	1	3	λ_2	1	1	1
$\Sigma_{c,2}$	1	1	3	λ_3	1	1	1
$\Sigma_{c,3}$	1	1	5	λ_4	1	1	1
$\Sigma_{c,4}$	1	1	5	λ_5	1	1	1
$\Sigma_{c,5}$	1	1	9	λ_6	1	1	1
$\Sigma_{c,6}$	3	1	7	λ_7	1	1	1
β_1	1	1	1	λ_8	1	1	1

6.5.2. PROPAGATING UNCERTAINTIES

We used the reduced models to propagate uncertainties in the parameters to the responses of interest. We consider two model responses; The first is the maximum temperature of the system, which has a value in the nominal state of 1084.8 K, and the second is the multiplication factor with a nominal value of 1.00999. To extract the probability density function (PDF) of the response, we run the reduced model at randomly sampled points drawn from the distribution of the input parameters. A histogram of the response is an approximation of the PDF. Figure 6.8 shows the normalized density histograms for the maximum temperature and multiplication factor by using 100,000 random points. The maximum temperature has a mean of 1085 K with a standard deviation of 6.9 K. The distribution is close to a normal distribution with a slight skew to lower temperatures. Figure 6.9 shows the normality plot of the data, which is a measure of the degree of deviation of the data from the normal distribution. The multiplication factor has a mean of 1.01009, with a standard deviation of 0.01899. The distribution of k_{eff} follows a perfect normal distribution, which is confirmed by the normal probability plot in Figure 6.9.

In order to study the sensitivity of the response to the input parameters, the first order Sobol indices are computed from the reduced models. However, from the number of unique nodes per dimension given in Table 6.6, the reduced model for temperature was shown to be sensitive only to one parameter (c_p). Therefore, we only compute Sobol indices for k_{eff}. We used a quasi-random sampling Sobol sequence [38]. The
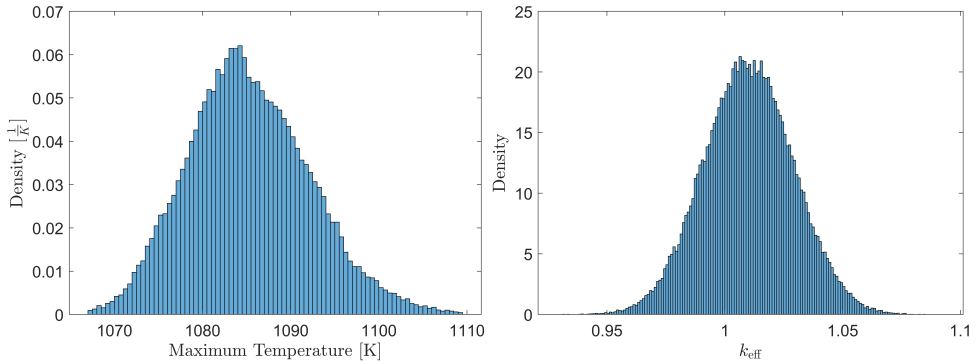
Figure 6.8: Normalized density histograms for the maximum temperature (left) and the multiplication factor k_{eff} (right). The data was generated by sampling the reduced model with 100,000 random points drawn from the distribution of the input parameters.
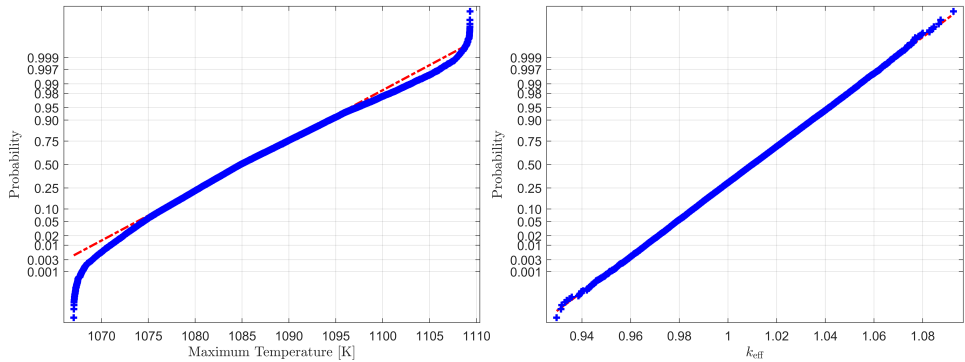

Figure 6.9: Normal probability plots for the maximum temperature (left) and the multiplication factor k_{eff} (right).

sequence was generated on the unit hypercube, then mapped to the distribution of each parameter using inverse sampling of the cumulative distribution function (CDF). Two sets of sampling points each of size 10^5 were used to compute the Sobol indices as given by Saltelli et al. [39]. The Sobol indices in Figure 6.10 show k_{eff} to be sensitive only to the cross sections. The indices ranked the fission and capture cross sections of group 5 to be the most significant, followed by groups 3 and 4. The fast groups (groups 1 and 2) and the most thermal groups have relatively lower impact on k_{eff}. This can be explained by the fact that the reactor operates with an epithermal spectrum. This is confirmed from the average flux value per group in Table 6.3. The indices also show that, in general, fission cross sections have higher importance on k_{eff} compared to the capture cross sections. This is expected because fission cross sections have a direct impact on the multiplication factor while the capture cross section impacts k_{eff} through the loss term which also includes the system leakages.
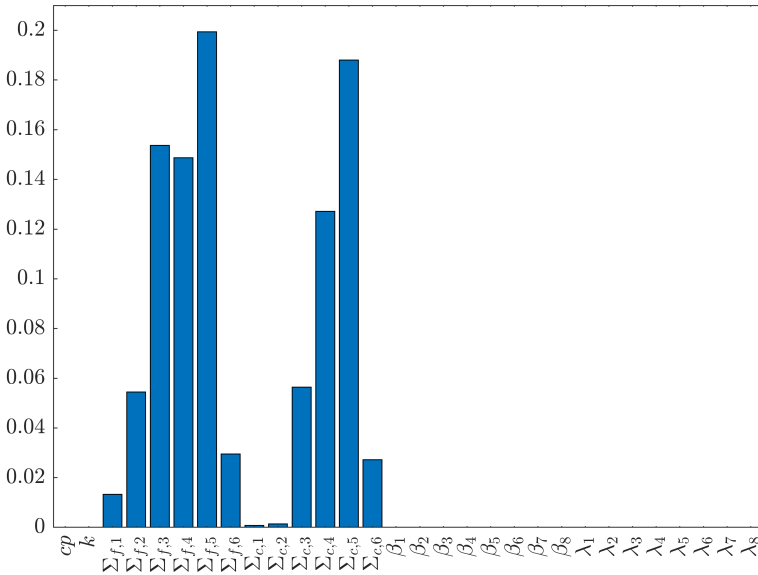
Figure 6.10: First order Sobol indices showing the first order sensitivities of k_{eff} to each input parameter.

6.6. TRANSIENT ANALYSIS

In this section, we consider the transient model of the MSFR where the reactor power is no longer fixed. The reactor is kept at steady-state by dividing the fission operator by nominal value of k_{eff}. Therefore, the initial conditions for the transients are the nominal values. We are interested in a reduced model for control and simulation purposes, capturing the dynamics of the fission power density and temperature with respect to perturbations in the salt flow rate of the intermediate circuit. This case is simulating an operational scenario where the reactor power is controlled through adjustments in the flow rate of the salt in the intermediate circuit. The intermediate circuit extracting heat from the heat exchangers was not modelled explicitly. However, for the transient analysis, we simulate the effect of controlling the reactor power through the salt flow rate in the intermediate circuit by adjusting the average intermediate salt temperature and heat transfer coefficient. Since we are interested in operational conditions, we employ a simple linear empirical model relating changes in the flow rate of the salt in the intermediate circuit to changes in the average intermediate salt temperature and heat transfer coefficients as

$$\Delta T_{\text{int}} = -0.375 \, \Delta q, \tag{6.15}$$

$$\Delta h_{\text{int}} = 74825 \Delta q, \tag{6.16}$$

where ΔT_{int} is the change in intermediate salt temperature in Kelvin, Δh_{int} is the change in the heat transfer coefficients in units of MW/m^{3}K, and Δq is the percentage change in the flow rate of the salt in the intermediate circuit. In order to approximate the dynamics of controlling the salt flow rate, these perturbations are introduced in the model exponentially with a time constant of 10 s. Therefore, the models for the average intermediate salt

temperature and heat transfer coefficient are

$$T_{\text{int}}(t) = T_{\text{int},0} + \Delta T_{\text{int}}\left(1 - \exp\left(\frac{-t}{10}\right)\right), \tag{6.17}$$

$$h_{\text{int}}(t) = h_{\text{int},0} + \Delta h_{\text{int}}\left(1 - \exp\left(\frac{-t}{10}\right)\right), \tag{6.18}$$

where $T_{\text{int}}(t)$ and $h_{\text{int}}(t)$ are respectively the intermediate salt temperature and heat transfer coefficient as functions of time t, $T_{\text{int},0}$ is the nominal value of the intermediate salt temperature with a value of 908.15 K (as reported in Table 6.1), and $h_{\text{int},0}$ is the nominal value of the heat transfer coefficient with a value of 19.95 MW/m^{3}K (Table 6.1).

Because we considered time as an input parameter in our formulation, this model has two input parameters, the change in the intermediate circuit flow rate (Δq) and time (t). The outputs are the fission power density, which is computed on the neutronics mesh with 104288 DOF, and the temperature, which is computed on the CFD mesh with 220972 DOF. We considered the flow rate to range between -40% to +15% of the nominal values. The time range was taken $t \in [0, 180]$ s – that is, we are interested in constructing a reduced model for the first 180 s after a perturbation in the intermediate flow rate Δq.

A transient of increasing the flow rate by 15% is shown in Figure 6.11, along with snapshots of the final solutions at $t = 180$ s. Increasing the flow rate of the intermediate circuit causes the temperature of salt flowing back into the cavity from the heat exchanger to drop, which is observed as a decrease of the minimum and average temperatures. The lower temperature in the reactor core introduces positive reactivity because of the strong negative temperature feedback of this reactor. For this reason, the trends show an increase in the reactor power and the maximum temperature registered at the top of the reactor cavity. The average temperature then gradually adjust the downward trend, and a new steady-state is reached.

To build the reduced model, we set the tolerances $\zeta = 1\%$ and $\gamma_{\text{int}} = 0.1\%$ for the fission power density, and $\zeta = 0.1\%$ and $\gamma_{\text{int}} = 0.01\%$ for the temperature. The POD truncation tolerance γ_{tr} was set to be 10^{-12} for both models. A summary of the results is given in Table 6.7. The algorithm required 9 simulations (corresponding to 9 values of Δq) for the fission power density and selected a total of 78 snapshots to build the reduced model. For the temperature, the algorithm required 5 simulations and selected 62 snapshots. The selected points for both models are shown in Figure 6.12. It can be seen that the algorithm selected more snapshots in the transient region (the first 80 seconds) and fewer points towards the steady-state region. Note that for the fission power density model, not all simulations were run full transient from $t = 0$ to 180 s. Only 5 full simulations were required while 4 simulations ran only to $t = 90$ s. Given that a full simulation to $t = 180$ s requires about 150 CPU-hours, such adaptivity significantly improves the efficiency of constructing the reduced model compared to an a priori snapshot selection approach, which does not consider the actual dynamics of the system.

To test the constructed reduced models, we ran 8 high-fidelity test simulations with values of Δq at half the distances between the points selected during the construction phase. In this manner, we maximize the distance (along Δq dimension) between the test simulations and the simulations used for the construction of the reduced models. We

then select 1000 random snapshots at different times from the testing simulations, which are used to test the reduced model. A histogram of the relative ℓ^2 norm error is shown in Figure 6.13 for both models. The error in the temperature model was found to be well below the set tolerance, with the maximum error being 0.06%. For the fission power density model, most of the points resulted in an error below the set tolerance. However, 6 out of the 1000 points were above the tolerance, with the maximum being 1.8%. The point that resulted in the maximum error is compared with the high-fidelity solution in Figure 6.14 for the fission power density and Figure 6.15 for the temperature. In this case, the fission power density comparison shows the maximum difference to be at the center of the cavity, where the flux is maximum. The local relative error of at that point is 1.8%. The maximum temperature difference, on the other hand, is observed at the inlet of the reactor core where the temperature is minimum, with a local relative error of 0.1%. A full simulation to $t = 180\,\text{s}$ is completed with the reduced model in under 5 seconds, while the high-fidelity model needed 150 CPU hours on a high performance computing unit.

Table 6.7: Summary of results for the transient reduced models corresponding to the fission power density $\boldsymbol{P}_{\text{fiss}}$ and temperature $\boldsymbol{T}$ showing the number of required simulations to construct each model, number of snapshots selected, the final number of points in $\mathcal{X}^k$, the number of POD modes after truncation, and the maximum relative ℓ^2 error after a test on 1000 random points.

	$\boldsymbol{P}_{\text{fiss}}$	$\boldsymbol{T}$
Number of simulations	9	5
Number of snapshots	78	62
Number of selected points in $\mathcal{X}^k$	38	31
Number of POD modes	5	6
Maximum relative ℓ^2 error	1.8%	0.06%

6.7. CONCLUSIONS

We have applied an adaptive POD approach to a large-scale, three-dimensional model of the MSFR. The developed algorithm was able to construct reduced models for both steady-state and transient analysis of the reactor. The steady-state analysis considered higher dimensional input space with 30 parameters. Three reduced models were constructed to capture the effects of those parameters on the fission power density and temperature distributions, and on the multiplication factor. Each model was tested on 1000 random points that were not part of the snapshot selections. The model for fission power density required 61 high-fidelity evaluations and resulted in a maximum relative ℓ^2 error of 0.24%. The model for the temperature required 63 evaluations and resulted in a maximum error of 0.02%. The multiplication factor model needed 1639 evaluations and the maximum error was 37 pcm. Evaluating the reduced models was completed in less than a second while the high-fidelity model needed 4 CPU-hours.

The reduced models were then used to propagate uncertainties in the input parameters to the maximum temperature of the reactor and the multiplication factor. The constructed PDF showed the multiplication factor to follow a normal distribution with a mean 1.01009 and standard deviation of 0.01899. The maximum temperature had a mean

of 1085 K, a standard deviation of 6.9 K, and showed a distribution close to normal with a slight skew to lower temperatures. The sensitivity study concluded that the maximum temperature of the MSFR is only sensitive to the specific heat capacity within the defined tolerances. Heat conductivity and neutronics data had no significant effect on the temperature. Therefore, since the salt properties of the MSFR are still under investigation, we recommend prioritizing studies to reduce uncertainties in the specific heat capacity over heat conductivity. The multiplication factor was shown to be only sensitive to the cross sections. A standard deviation of 5% in the decay constant and delayed neutron fraction is sufficient to characterize the multiplication factor within 50 pcm error. However, a standard deviation of 5% in the cross sections resulted in a distribution of the multiplication factor with a standard deviation of 1899 pcm, which is significant. Theretofore, uncertainties in cross section data should be below 5% standard deviation for this reactor.

The transient analysis studied the effect of controlling the flow of salt in the intermediate circuit on the fission power density and temperature distributions. Our approach considers time to be a parameter in input space in order to allow for an adaptive selection of the snapshots. The model for the fission power density needed 9 simulations and selected 78 snapshots. The test on 1000 independent points showed the maximum relative ℓ^2 error to be 1.8%. The model for the temperature required 5 simulations and selected 62 snapshots. The maximum error was found to be 0.06% for this model. The selected points showed that the algorithm sampled the high-fidelity model in regions of the beginning of the transient more than the steady-state. This allowed the algorithm to be efficient by simulating some of the points only to half the transient. A full simulation to the end of the transient required about 150 CPU-hours. Therefore, reducing the transient time for a point is a massive saving in computational resources. The constructed reduced models were able to produce a full simulation in less than 5 seconds. As follow-up work, this reduced model can be used to design a controller for the reactor.

In all models, the number of points included in the final grid set $\mathcal{X}^k$ was a small fraction of the total snapshots generated by the algorithm. This fraction was about 50% for the transient models while for the steady-state models, it was found to be as low as 10%. This is an indication that the algorithm is still oversampling. The number of POD modes was found to be even a smaller fraction of the snapshots, which is a signal that most of the points were generated to train the surrogate model rather than discover new dynamics of the system. Therefore, the algorithm could be improved further with more advanced surrogate models to reduce the number of sampling points. The use of higher-order basis functions is a potential area of study for this propose. Moreover, our approach to the transient problem assumes the initial conditions to be parametrized. The approach is not yet applicable for problems with a time-dependent input signal, which cannot be parametrized. These areas of research are the subjects of future work.
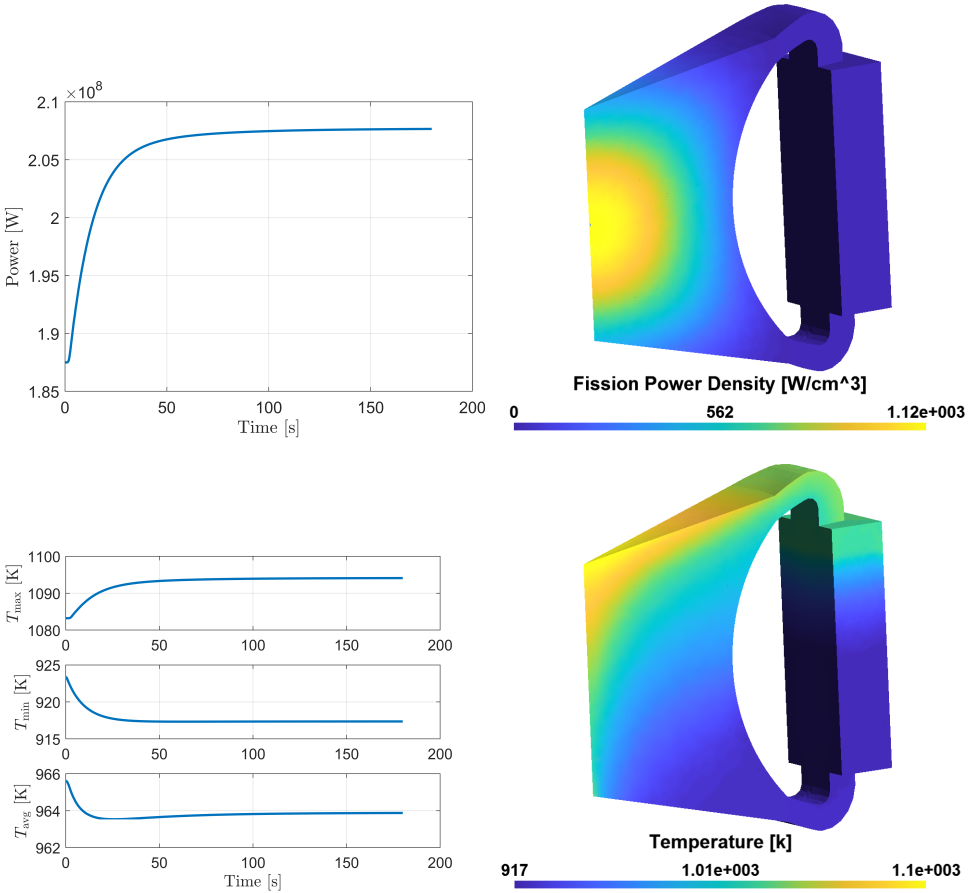
Figure 6.11: A selected transient resulting from $\Delta q = 15\%$ showing the reactor power (top left), fission power density distribution at $t = 180\,\text{s}$ (top right), temperature trends for the maximum temperature T_{max}, minimum temperature T_{min}, and average temperature T_{avg} (bottom left), and temperature distribution at $t = 180\,\text{s}$ (bottom right).
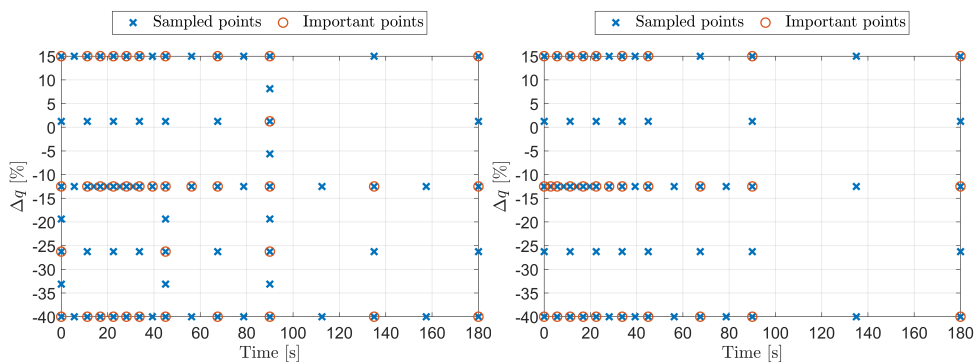
Figure 6.12: The generated points for the transient model of the fission power density (left) and temperature (right). The important points included in the final $\mathscr{X}^k$ are marked with a red circle.
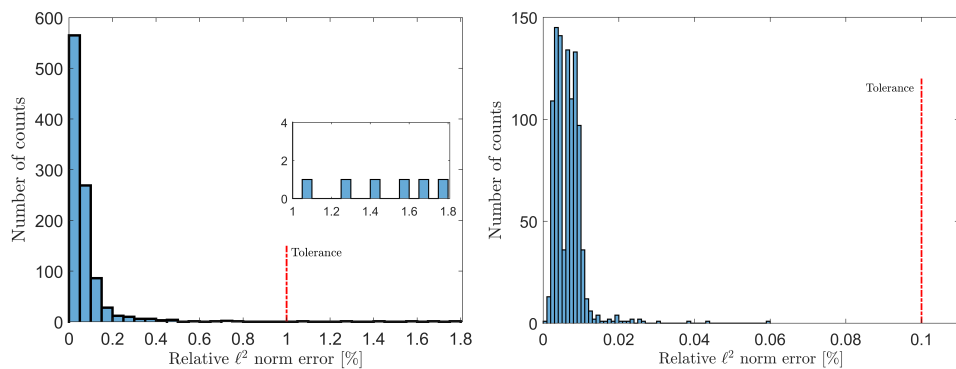

Figure 6.13: Histogram of the relative ℓ^2 error in the fission power density model (left) and temperature model (right). A close up on the region above 1% is shown for the histogram of the fission power density model.

REF - Fission Power Density [W/cm^3]

0   503   1.01e+003

ROM - Fission Power Density [W/cm^3]

0   503   1.01e+003

Difference - Fission Power Density [W/cm^3]

0   8.84   17.7
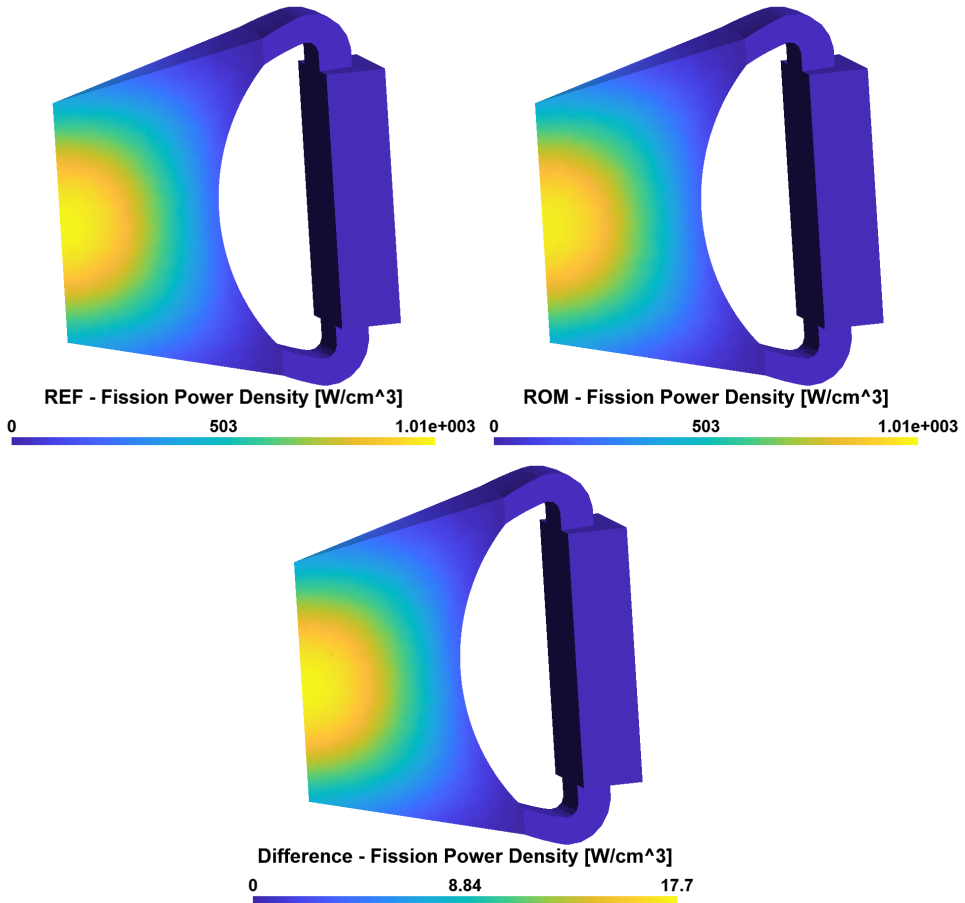
Figure 6.14: Fission power density distribution at the point of maximum error showing the reference model (top left), the ROM model (top right), and the distribution of the absolute difference (bottom). This point correspond to $\Delta q = -36.5$ and at time $t = 2.4$ s. The relative ℓ^2 error was 0.24%.
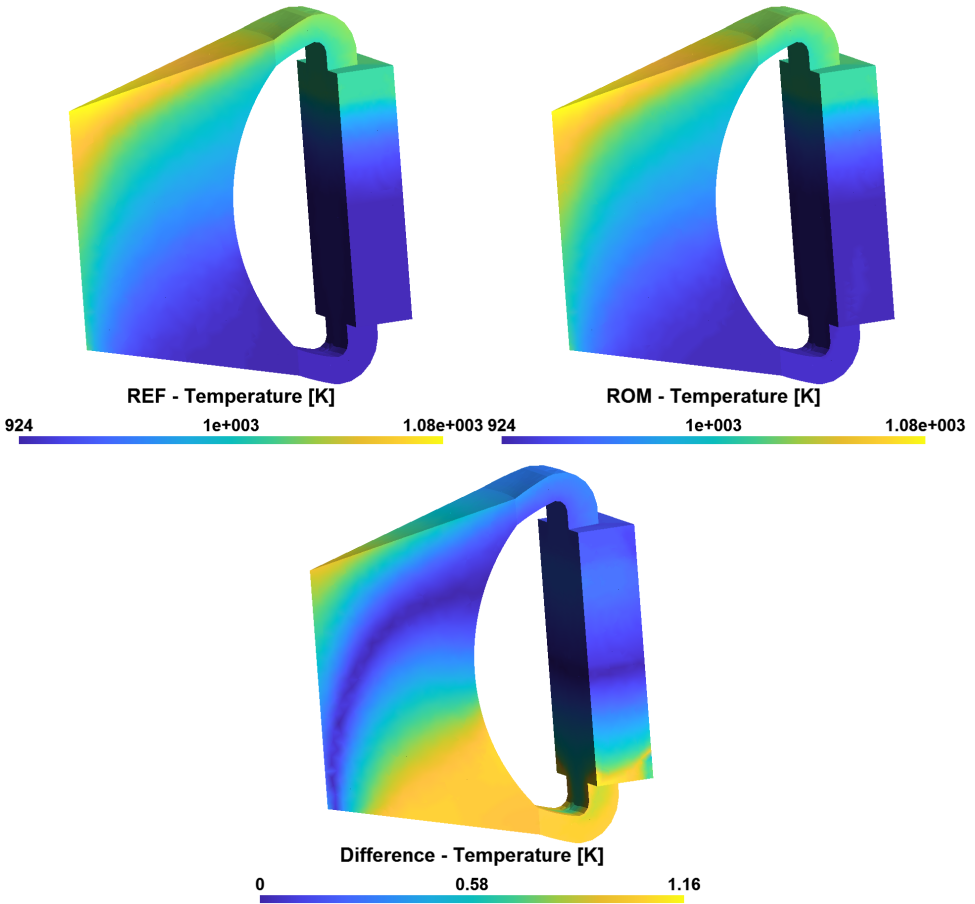
Figure 6.15: Fission power density distribution at the point of maximum error showing the reference model (top left), the ROM model (top right), and the distribution of the absolute difference (bottom). This point correspond to $\Delta q = -36.5$ and at time $t = 1.5\,\mathrm{s}$. The relative ℓ^2 error was 0.06%.

References

[1] F. Alsayyari, M. Tiberga, Z. Perkó, J. L. Kloosterman, and D. Lathouwers, *Analysis of the molten salt fast reactor using reduced-order models,* Progress in Nuclear Energy (2020), under review.

[2] Generation IV International Forum, *A Technology Roadmap for Generation IV Nuclear Energy Systems,* Tech. Rep. GIF-002–00 (U.S. DOE Nuclear Energy Research Advisory Committee and the Generation IV International Forum, 2002).

[3] T. Dolan, *Molten Salt Reactors and Thorium Energy,* Series in Energy (Woodhead Publishing, Elsevier Ltd., Cambridge, MA, United States, 2017).

[4] M. Allibert, M. Aufiero, M. Brovchenko, S. Delpech, V. Ghetta, D. Heuer, A. Laureau, and E. Merle-Lucotte, *Molten salt fast reactors,* in *Handbook of Generation IV Nuclear Reactors,* edited by I. L. Pioro (Woodhead Publishing, 2016) pp. 157–188.

[5] *Deterministic Safety Analysis for Nuclear Power Plants,* Specific Safety Guides No. SSG-2 (Rev.1) (INTERNATIONAL ATOMIC ENERGY AGENCY, Vienna, 2019).

[6] L. Gilli, D. Lathouwers, J. Kloosterman, and T. van der Hagen, *Applying second-order adjoint perturbation theory to time-dependent problems,* Annals of Nuclear Energy **53**, 9 (2013).

[7] Z. Perkó, D. Lathouwers, J. L. Kloosterman, and T. van der Hagen, *Adjoint-based sensitivity analysis of coupled criticality problems,* Nuclear Science and Engineering **173**, 118 (2013).

[8] D. Cacuci, *Sensitivity & Uncertainty Analysis,,* Vol. 1 (Chapman and Hall/CRC, 2003).

[9] A. C. Antoulas, D. C. Sorensen, and S. Gugercin, *A survey of model reduction methods for large-scale systems,* Contemporary Mathematics **280**, 193 (2001).

[10] W. H. A. Schilders, H. A. van der Vorst, and J. Rommes, eds., *Model Order Reduction: Theory, Research Aspects and Applications* (Springer Berlin Heidelberg, 2008).

[11] P. Benner, S. Gugercin, and K. Willcox, *A Survey of Projection-Based Model Reduction Methods for Parametric Dynamical Systems,* SIAM review **57**, 483 (2015).

[12] A. G. Buchan, C. C. Pain, F. Fang, and I. M. Navon, *A POD reduced-order model for eigenvalue problems with application to reactor physics,* International Journal for Numerical Methods in Engineering **95**, 1011 (2013).

[13] J. P. Senecal and W. Ji, *Characterization of the proper generalized decomposition method for fixed-source diffusion problems,* Annals of Nuclear Energy **126**, 68 (2019).

[14] P. German and J. C. Ragusa, *Reduced-order modeling of parameterized multi-group diffusion k-eigenvalue problems,* Annals of Nuclear Energy **134**, 144 (2019).

[15] Z. M. Prince and J. C. Ragusa, *Application of proper generalized decomposition to multigroup neutron diffusion eigenvalue calculations,* Progress in Nuclear Energy **121**, 103232 (2020).

6

[16] A. Cherezov, R. Sanchez, and H. G. Joo, *A reduced-basis element method for pin-by-pin reactor core calculations in diffusion and SP₃ approximations,* Annals of Nuclear Energy **116**, 195 (2018).

[17] C. Castagna, M. Aufiero, S. Lorenzi, G. Lomonaco, and A. Cammi, *Development of a reduced order model for fuel burnup analysis,* Energies **13**, 890 (2020).

[18] L. Vergari, A. Cammi, and S. Lorenzi, *Reduced order modeling approach for parametrized thermal-hydraulics problems: inclusion of the energy equation in the POD-FV-ROM method,* Progress in Nuclear Energy **118**, 103071 (2020).

[19] D. Prill and A. Class, *Semi-automated proper orthogonal decomposition reduced order model non-linear analysis for future BWR stability,* Annals of Nuclear Energy **67**, 70 (2014).

[20] R. Manthey, A. Knospe, C. Lange, D. Hennig, and A. Hurtado, *Reduced order modeling of a natural circulation system by proper orthogonal decomposition,* Progress in Nuclear Energy **114**, 191 (2019).

[21] J. Y. Escanciano and A. G. Class, *POD-Galerkin modeling of a heated pool,* Progress in Nuclear Energy **113**, 196 (2019).

[22] 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 **87**, 198 (2016).

[23] D. Amsallem and C. Farhat, *Stabilization of projection-based reduced-order models,* International Journal for Numerical Methods in Engineering **91**, 358 (2012).

[24] M. Santanoceto, M. Tiberga, Z. Perko, S. Dulla, and D. Lathouwers, *Uncertainty quantification in steady state simulations of a molten salt system using polynomial chaos expansion analysis,* in *International Conference On Physics Of Reactors (PHYSOR 2020)* (Cambridge, United Kingdom, 2020).

[25] A. D. Ronco, C. Introini, E. Cervi, S. Lorenzi, Y. S. Jeong, S. B. Seo, I. C. Bang, F. Giacobbo, and A. Cammi, *Dynamic mode decomposition for the stability analysis of the Molten Salt Fast Reactor core,* Nuclear Engineering and Design **362**, 110529 (2020).

[26] J. H. Tu, , C. W. Rowley, D. M. Luchtenburg, S. L. Brunton, and J. N. Kutz, *On dynamic mode decomposition: Theory and applications,* Journal of Computational Dynamics **1**, 391 (2014).

[27] D. Huang, H. Abdel-Khalik, C. Rabiti, and F. Gleicher, *Dimensionality reducibility for multi-physics reduced order modeling,* Annals of Nuclear Energy **110**, 526 (2017).

[28] F. Alsayyari, Z. Perkó, D. Lathouwers, and J. L. Kloosterman, *A nonintrusive reduced order modelling approach using Proper Orthogonal Decomposition and locally adaptive sparse grids,* Journal of Computational Physics **399**, 108912 (2019).

[29] 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 **141**, 107321 (2020).

[30] F. Alsayyari, Z. Perkó, M. Tiberga, J. L. Kloosterman, and D. Lathouwers, *A fully adaptive nonintrusive reduced order modeling approach for parametrized time-dependent problems,* Computer Methods in Applied Mechanics and Engineering (2020), under review.

[31] M. Tiberga, D. Lathouwers, and J. L. Kloosterman, *A multi-physics solver for liquid-fueled fast systems based on the discontinuous Galerkin FEM discretization,* Progress in Nuclear Energy **127**, 103427 (2020).

[32] S. Delpech, E. Merle-Lucotte, D. Heuer, M. Allibert, V. Ghetta, C. Le-Brun, X. Doligez, and G. Picard, *Reactor physic and reprocessing scheme for innovative molten salt reactor system,* Journal of Fluorine Chemistry **130**, 11 (2009).

[33] D. Gérardin, M. Allibert, D. Heuer, A. Laureau, E. Merle-Lucotte, and C. Seuvre, *Design evolutions of Molten Salt Fast Reactor,* in *International Conference on Fast Reactors and Related Fuel Cycles: Next Generation Nuclear Systemes for Sustainable Development (FR17)* (Yekaterinburg, Russia, 2017).

[34] M. Tiberga, D. Lathouwers, and J. L. Kloosterman, *A discontinuous Galerkin FEM multiphysics solver for the Molten Salt Fast Reactor,* in *International Conference on Mathematics and Computational Methods applied to Nuclear Science and Engineering (M&C 2019)* (Portland, OR, USA, 2019).

[35] J. Leppänen, M. Pusa, T. Viitanen, V. Valtavirta, and T. Kaltiaisenaho, *The Serpent Monte Carlo code: Status, development and applications in 2013,* Annals of Nuclear Energy **82**, 142 (2015).

[36] A. Santamarina, D. Bernard, P. Blaise, M. Coste, A. Courcelle, T. D. Huynh, C. Jouanne, P. Leconte, O. Litaize, J.-M. Ruggiéri, O. Sérot, J. Tommasi, C. Vaglio, and J.-F. Vidal, *The JEFF-3.1.1 Nuclear Data Library,* JEFF Report 22 (NEA–OECD, 2009).

[37] O. Beneš, M. Salanne, M. Levesque, and R. J. M. Konings, *Physico-Chemical properties of the MSFR fuel salt — Deliverable D3.2,* Tech. Rep. (EVOL Project, 2013).

[38] I. M. Sobol, *Global sensitivity indices for nonlinear mathematical models and their Monte Carlo estimates,* Mathematics and Computers in Simulation **55**, 271 (2001).

[39] A. Saltelli, P. Annoni, I. Azzini, F. Campolongo, M. Ratto, and S. Tarantola, *Variance based sensitivity analysis of model output. design and estimator for the total sensitivity index,* Computer Physics Communications **181**, 259 (2010).

6

7

CONCLUSIONS AND OUTLOOK

THE main goal of this work was to develop a methodology for constructing reduced-order models for nonlinear, large-scale nuclear reactor models. We have opted for a nonintrusive approach because these high-fidelity models are often implemented in closed-source or intricate coupled codes, rendering attempts to apply intrusive approaches intractable. In addition, the high-dimensional parameter space characterizing nuclear reactor systems was a fundamental issue motivating this work. In the intrusive setting, the success of the reduced basis (RB) approach was driven by the greedy sampling strategy formed by the seamless fusion between a cheap a posteriori error estimator and the knowledge of the physics contained in the projected original equations. While some attempts to employ adaptive strategies in nonintrusive settings exist in the literature, their applicability has been limited to parameter spaces of small dimensional sizes, as discussed in *Section* 3.1 for steady-state systems and *Section* 5.1 for time-dependent systems.

Our work attempted to address the lack of tools for adaptive sampling strategies in the nonintrusive setting. Such tools are indispensable for parameter spaces that have high dimensions. Ideally, the sampling technique and the reduced-order modelling (ROM) method have synergistic effects where the reduced-order model requires only few sampling points, and the sampling technique is smart enough to place those points in the most relevant regions. To that end, we developed an algorithm that utilizes the locally adaptive sparse grid technique as a sampling strategy for the proper orthogonal decomposition (POD) method in nonintrusive settings. We present the main results and conclusions of our work in the following section.

7.1. MAIN RESULTS AND CONCLUSIONS

The first chapter presents a survey of different ROM approaches and categorized the methods into intrusive and nonintrusive based on their requirement to access the governing equations of the high-fidelity solver. The main conclusions of the survey are

- POD is the method most suited for general nonlinear systems;

- A nonintrusive black-box POD approach can be implemented by learning surrogate models for the POD coefficients. Different approaches for the surrogate models exist in the literature;

- To address high-dimensional parameter spaces, sparse grid interpolations can be used to build a surrogate multivariate interpolant for the POD coefficients.

Following the survey conclusions, we compare two POD-based black-box approaches in **Chapter** 2: Classical (non-adaptive) sparse grids and radial basis functions (RBF). The two approaches were tested on two neutron diffusion eigenvalue problems. The first problem was parametrized on a two-dimensional parameter space and the second was parametrized with 5 dimensions. The main finding of the chapter are

- While RBF resulted in a faster reduced-order model in the online stage, the sparse grid model provided a superior accuracy. Admittedly, such a study is not conclusive with regards to the superiority of sparse grids approach to RBF for all problems because aspects such as selecting the RBF kernel and tuning the shape parameter were not investigated. However, the combination of the sparse grids sampling method with the Smolyak interpolant provides an interesting framework for high-dimensional parametrized problems;

- Although the sparse grid approach was more efficient in generating the sampling points compared to the full grid tensorization, the total number of points, which is set by the sparse grid level, still depends on the dimension of the parameter space (as illustrated by Table 2.1).

The dependence of the number of sampling points on the dimension size renders the classical sparse grids approach to be impractical for high-dimensional spaces. For this reason, we investigated including adaptivity to the sparse grids approach. In **Chapter** 3, we present our adaptive sparse grids algorithm for a POD-based ROM method. We added a greediness parameter to the developed algorithm to control the efficiency of the sampling strategy in the offline stage. We also provide an effective approach to update the surpluses of the Smolyak interpolants with each update of the POD basis space. Three numerical test cases are presented in this chapter. The main conclusions are

- The local adaptivity improves the effectiveness of the sampling strategy compared to classical sparse grid approach because of the algorithm's ability to recognize regions of high importance;

- The added greediness parameter allowed for even higher dimensional applications by identifying and disregarding irrelevant dimensions after only few iterations;

- The structure of the developed adaptive sparse grid technique provides a tool for characterizing the nonlinearities of the model with respect to each parameter. That is, counting the number of the projected nodes on each dimension revealed information about the underlying physics without any knowledge about the governing equations.

- In spite of the non-intrusiveness of our approach, the presented adaptive sparse grid technique can also advance sampling strategies for intrusive approaches.

In order to test the ability of the algorithm in constructing an effective reduced-order model for a large-scale complex system, **Chapter** 4 presents a test on a two-dimensional molten salt nuclear reactor system. The model has multiple outputs and 27 parameters to be investigated. The constructed reduced-order model was used for uncertainty quantification and sensitivity analysis application. The main findings of the chapter are

- The constructed reduced-order model achieved massive reduction in computational time for such a large-scale problem. The reduced-model was able to reproduce the solution for any point within the defined range faster than the high-fidelity model by a factor of 5×10^6;

- The developed reduced-order model was efficient enough to allow for exhaustive sampling in applications of uncertainty quantification and global sensitivities analysis (Sobol indices);

- The derivatives of each output with respect to the parameters can be expressed analytically, which allows for computing local sensitivities;

- For multiple output problems, constructing a separate reduced-order model for each output was more efficient in terms of total number of sampling points compared to stacking the outputs in a single column approach.

In **Chapter** 5, we turn to time-dependent problems and extend the adaptivity of our algorithm to the time domain. Three numerical test cases of various degrees of nonlinearities, complexity, scale, and dimensionality are presented to demonstrate the effectiveness of the developed time-adaptive approach. The main chapter conclusions are

- The developed algorithm was successful in identifying important regions in the space formed by time and the input parameters;

- Concentrating the sampling points in the discovered important regions resulted in an efficient offline stage because the algorithm correctly identified that not all points need to be run up to the end of the simulation time. Thus, neglecting snapshots in transient regions of near steady-state;

- An adaptive approach in selecting the time snapshots for parametrized dynamical systems is more effective than a fixed sampling scheme, which is blind to the dynamics of the system.

In **Chapter** 6, we present reduced-order models for the molten salt fast reactor (MSFR) for both steady-state and transient analysis. Preliminary analysis of the steady-state considered uncertainties in 30 model parameters while a dynamic reduced-order model was constructed for the transient analysis to relate changes in the salt flow rate of the intermediate circuit to the distributions of the reactor fission power and temperature. The main results and conclusions of the study are

- Within the defined tolerances, the maximum temperature of the fuel salt was shown to be more sensitive to the specific heat capacity than heat conductivity for the same normally distributed uncertainty. For this reason, it is recommended that thermodynamic studies of the salt mixture are prioritized to reduce uncertainties in the specific heat capacity;

- Neutronics data had no effects on the temperature within the defined tolerances;

- The multiplication factor was sensitive to the cross section data, with no significant effect from the precursors decay constants or delayed neutron fractions;

- A standard deviation of 5% in the normal distribution of the cross sections resulted in a normally distributed multiplication factor with a standard deviation of 1800 pcm, which is relatively high. This suggests that cross section data studies should aim to reduce the uncertainty in the cross section data to lower than 5% in the standard deviation.

- For the transient analysis, the algorithm was successful in constructing a reduced-order model for the dynamic, large-scale, multi-physics reactor model. As follow-up work, the developed reduced-order model can be used for control design purposes.

- The high-fidelity model requires about 150 CPU-hours to simulate a transient of 180 s while the constructed reduced-order model produces the same results in about 5 seconds.

7.2. RECOMMENDATIONS

This research can be extended in several directions. The developed algorithm provided a tool to control the accuracy by setting the required tolerance on the error in the constructed reduced-order model, which is a great advantage for any ROM approach. The numerical tests in our work showed that, in most cases, the tolerance was a good estimator for the maximum error in the model. However, this error estimator is not rigorous and can only be taken as an indicator. In some of the tests, the maximum error was observed to be slightly higher than the tolerance. For this reason, we developed the validation algorithm presented in **Chapter** 3, which can restart the offline phase to enrich the initially constructed reduced-order model in regions of unsatisfactory error. Defining a rigorous upper bound for the error remains an open challenge for any black-box nonintrusive approach because the inherent advantage of not relying on a physical model is a weakness in having an authority to certify the constructed reduced-order model. Moving towards grey-box modeling by incorporating some knowledge about the system can be a path to define a robust error estimator that is specific to the problem.

Moreover, the numerical tests in this work showed that the algorithm marked only about 10 to 30% of the total sampled snapshots as important. It was also observed that the final reduced basis size was an even smaller fraction. Such a low percentage indicates that most of the points were sampled to improve the interpolants rather than discover new dynamics for the POD basis space. The inactive points (i.e., points that were sampled but not included in the snapshots matrix) were not wasted because they served the function of testing the model during the construction stage. However, the high percentage of

these inactive points indicates an oversampling issue, which is an area to explore for further improving the algorithm. Such improvement is especially crucial when applying the algorithm to problems with very high-dimensional parameter spaces (e.g., in the hundreds). In these problems, the algorithm may require a prohibitively large number of model evaluations. Reducing the number of evaluations for such problems can be addressed by utilizing knowledge of the physics of the system to screen the parameters first or limit their range of variations in order to narrow the search in the parameter space. For example, if we know that the sensitivity of the model to variations at the boundaries is small, we can choose a tree structure for the unidimensional rule without the boundary points. Furthermore, investigating the use of different basis functions for the surrogate models, such as higher-order interpolants or regression models instead of interpolation, is among the possibilities to address the oversampling issue. Constructing different local basis spaces for different dynamics of the system is another approach to tackle oversampling.

Additionally, we have not covered coupling the reduced-order model to a high-fidelity model, which can be in the form of providing an initial estimate for the solution so that the high-fidelity solver converges faster or providing a solution to a different coupled phenomenon for increased accuracy. Such concepts are interesting lines of research, which open questions of adapting the algorithm to input signals and initial conditions that are not parametrized (i.e., cannot be bounded by a predefined range). Furthermore, we have not addressed iterative updating of the singular value decomposition (SVD) instead of the full SVD in each iteration, which can help in reducing the algorithm's memory requirement in large-scale applications. However, SVD updating algorithms are fairly straightforward since they are successfully used for the RB approach. Moreover, defining different error measures to select the important points, for example, by weighing the error with the adjoint solution, is an intriguing potential development for the algorithm. Using the developed algorithm to construct reduced-order models from noisy or experimental data is another research path with vast potential applications.

Albeit the focus of this research is on nuclear applications, methodologies developed during this research are expected to serve many fields. Reducing the order of large-scale problems with strong nonlinearity is an ever-present challenge in various sciences and engineering applications. Such objectives are of great economic benefits for different industries in areas of optimization, control, and uncertainty quantification. The research can even have a direct societal impact in areas of emerging technologies such as digital twins, which is an innovative concept whereby a product or system is mirrored into the digital world with great detail in physics, structure, functionality, economics, and visualization. Having a reduced-order model that can provide a high-fidelity solution for such complex integration on a personal computer or a smartphone is an exciting potential application for the nonintrusive adaptive POD algorithm.

7

ACKNOWLEDGEMENTS

This thesis is the product of research activities I performed at the Reactor Physics and Nuclear Material group in TU Delft. I am indebted to many great people who have provided the support and encouragement I needed to grow both as a researcher and as a person during my incredible journey.

First, I would like to thank my promotor Prof. Jan Leen Kloosterman, for his tremendous support, insightful comments, and valuable contributions both to the thesis and my doctoral education. His patience and understanding were evident from the first day we met when I forgot my passport and could not enter the reactor building for our first meeting. We walked all the way to the Aerospace building where we had our meeting at a table in the canteen. He then graciously accepted me into the group and provided me with the opportunity to accomplish this work.

I owe my deepest gratitude to my second promotor Dr. Danny Lathouwers who has been a great mentor with his continuous guidance, constant optimism, and endless support through every step of my Ph.D. journey. I appreciate that he was always available for a motivating discussion or explaining complicated matters in a simple way. I would also like to convey my appreciation to Dr. Zoltán Perkó for his inspiring ideas and encouragement. Zoltán is also the reason behind every typo you do not find in this thesis; His attention to detail and swift responses make him the best proofreader I have ever worked with. I feel incredibly lucky to have worked with such an outstanding supervisory team.

This research has been possible with funding from King Abdulaziz City for Science and Technology (KACST). My gratitude goes to all people there who have provided me with the support, technical or otherwise. In particular, Dr. Turki bin Saud has given me his incredible support and encouragement, which opened the doors for my graduate education. I am also grateful to Dr. Khalid Aleissa for his generous support, mentorship, and enlightening discussions on nuclear matters, energy, and education. His guidance has shaped my career from day one at KACST. I also thank Yousef for being the best companion and friend, both in Argentina and in Europe. I also have to mention Ahmad, Abdullah, and Mohammed for being great friends during this journey. I would like also to give special thanks to the Saudi Cultural Mission in the Netherlands for all their support, making my life easier in the Netherlands. I thank especially Dr. Alwasel and my attentive advisor Lieke Gieles who was always supportive and helpful with all administrative issues.

I met many exceptional people during my work in the Reactor Institute Delft (RID), which made my stay in Delft a remarkable experience. I particularly thank Fulvio and Marco for being great office mates and more incredible friends outside the office. Our technical discussions and funny conversations are both appreciated at the same level. I also owe deep gratitude to the rest of my little-Italy community who have taken me in as one of their own and were my Italian family, Sara, Elisa, Laura, Allegra, Matteo, (quasi-Italian) Guilhem, and Enrica. I will always remember us enjoying beautiful moments, cooking and sharing delicious meals, playing beach volleyball, bouldering in Delfts Bleau,

161

and enjoying unforgettable trips. I also thank my non-Italian friends, Koos, Aldo, and Evgenii, for all the great memories, such as our trip to Georgia. I wish good luck to Jaen, Oscar, Bouke, Daniel, and Thomas in their own Ph.D. endeavors. I am grateful also to Anna for being a wonderful person who provided generous support and helped me with the Dutch translations. Anna, jij weet niks van zwerkbal.

Last but not least, I would have never been able to achieve this milestone of my career without the love and support of a wonderful family of which I am proud. To my brothers, Tariq, Saud, Yazeed, and (soon to be Dr.) Mohammed, I am forever indebted to your unwavering support and infinite backing through every part of my life. To my kind sister, Tarfa, you are not only the best sister I could have asked for but also the most creative designer I have met. I know that my words are not enough to compensate for all the tedious comments I gave you while working on the thesis cover, but know that I enjoyed every part of working with you. Above all, I owe everything I have achieved in my life to my role models and best teachers, my father, Sulaiman, and my wonderfully sweet mother, Norah.

Delft, September 2020

CURRICULUM VITÆ

Fahad ALSAYYARI

The author was born in Riyadh, Saudi Arabia. He attended Riyadh Schools from kindergarten to high school. In autumn of 2004, he enrolled in King Fahd University of Petroleum and Minerals in Dhahran, Saudi Arabia, majoring in Control and Instrumentation's systems engineering. He graduated in the summer of 2009 with an awarded Second Honours. He then joined the Atomic Energy Research Institute (AERI) at King Abdulaziz City for Science and Technology (KACST) as a researcher. After six months, he obtained a scholarship funded by KACST to study a master's degree in Instituto Balseiro, Argentina. He spent four years in San Carlos de Bariloche, Argentina, where he learned Spanish and completed a master's degree in engineering, specializing in Control of Nuclear Reactors. His master thesis was titled "Dynamic modeling of nuclear reactors using the simple nodal method." During these years, he also participated in the Low Power Research Reactor project. Then, he went back to Saudi Arabia, where he was appointed as head of the Nuclear Systems group within the newly established National Center for Nuclear Technology at KACST. In October of 2015, he moved to the Netherlands with another scholarship to become a Ph.D. candidate at the Radiation Science and Technology (RST) department of Delft University of Technology, where he accomplished this thesis.

LIST OF PUBLICATIONS

JOURNALS

1. **F. Alsayyari**, M. Tiberga, Z. Perkó, J.L. Kloosterman, D. Lathouwers, *Analysis of the Molten Salt Fast Reactor using Reduced-Order Models*, Progress in Nuclear Energy (2020). *Under review.*

2. **F. Alsayyari**, Z. Perkó, M. Tiberga, J.L. Kloosterman, D. Lathouwers, *A fully adaptive nonintrusive reduced order modeling approach for parametrized time-dependent problems*, Computer Methods in Applied Mechanics and Engineering (2020). *Under review.*

3. **F. Alsayyari**, M. Tiberga, Z. Perkó, D. Lathouwers, J.L. Kloosterman, *A nonintrusive adaptive reduced order modeling approach for a molten salt reactor system*, Annals of Nuclear Energy **141**, 107321 (2020).

4. **F. Alsayyari**, Z. Perkó, D. Lathouwers, J.L. Kloosterman, *A nonintrusive reduced order modelling approach using proper orthogonal decomposition and locally adaptive sparse grids*, Journal of Computational Physics **399**, 108912 (2019).

CONFERENCES

1. **F. Alsayyari**, M. Tiberga, D. Lathouwers, J.L. Kloosterman, *A Nonintrusive POD-Adaptive Reduced Order Model for Coupled Nuclear Reactor applications*, in European Numerical Mathematics and Advanced Applications Conference, Egmond aan Zee, The Netherlands, September 30 – October 4, 2019.

2. **F. Alsayyari**, Z. Perkó, D. Lathouwers, J.L. Kloosterman, *A Reduced Order Modeling Approach for Reactor Physics Problems Using Locally Adaptive Sparse Grids*, in Model Reduction of Parametrized Systems IV Conference, Nantes, France, April 10-13, 2018.

3. **F. Alsayyari**, D. Lathouwers, J.L. Kloosterman, *A Nonintrusive POD Approach for High Dimensional Problems using Sparse Grids*, in International Conference on Mathematics and Computational Methods applied to Nuclear Science and Engineering (M&C), Jeju, South Korea, April 16-20, 2017.