# Nonlinear Model Reduction for Uncertainty Quantification in Large-Scale Inverse Problems: Application to Nonlinear Convection-Diffusion-Reaction Equation

by

David Galbally

Ingeniero Industrial, Universidad Pontificia Comillas, Spain (2001)

Submitted to the Department of Aeronautics and Astronautics
in partial fulfillment of the requirements for the degree of

Master of Science in Aeronautics and Astronautics

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 2008

Author ........................................................
Department of Aeronautics and Astronautics
January 30, 2008

Certified by ........................................................
Karen E. Willcox
Associate Professor of Aeronautics and Astronautics
Thesis Supervisor

Accepted by ........................................................
David L. Darmofal
Associate Professor of Aeronautics and Astronautics
Chair, Committee on Graduate Students

# Nonlinear Model Reduction for Uncertainty Quantification in Large-Scale Inverse Problems: Application to Nonlinear Convection-Diffusion-Reaction Equation

by

David Galbally

Submitted to the Department of Aeronautics and Astronautics
on January 30, 2008, in partial fulfillment of the
requirements for the degree of
Master of Science in Aeronautics and Astronautics

## Abstract

There are multiple instances in science and engineering where quantities of interest are evaluated by solving one or several nonlinear partial differential equations (PDEs) that are parametrized in terms of a set of inputs. Even though well-established numerical techniques exist for solving these problems, their computational cost often precludes their use in cases where the outputs of interest must be evaluated repeatedly for different values of the input parameters such as probabilistic analysis applications.

In this thesis we present a model reduction methodology that combines efficient representation of the nonlinearities in the governing PDE with an efficient model-constrained, greedy algorithm for sampling the input parameter space. The nonlinearities in the PDE are represented using a coefficient-function approximation that enables the development of an efficient offline-online computational procedure where the online computational cost is independent of the size of the original high-fidelity model. The input space sampling algorithm used for generating the reduced space basis adaptively improves the quality of the reduced order approximation by solving a PDE-constrained continuous optimization problem that targets the output error between the reduced and full order models in order to determine the optimal sampling point at every greedy cycle. The resulting model reduction methodology is applied to a highly nonlinear combustion problem governed by a convection-diffusion-reaction PDE with up to 3 input parameters. The reduced basis approximation developed for this problem is up to $50,000$ times faster to solve than the original high-fidelity finite element model with an average relative error in prediction of outputs of interest of $2.5 \cdot 10^{-6}$ over the input parameter space.

The reduced order model developed in this thesis is used in a novel probabilistic methodology for solving inverse problems. The extreme computational cost of the Bayesian framework approach for inferring the values of the inputs that generated a given set of empirically measured ouputs often precludes its use in practical applications. In this thesis we show that using a reduced order model for running the Markov

Chain Monte Carlo simulations required by the Bayesian approach yields the same results as the high-fidelity model while reducing the computational cost by several orders of magnitude.

Thesis Supervisor: Karen E. Willcox
Title: Associate Professor of Aeronautics and Astronautics

# Acknowledgments

I would like to express my most sincere thanks to my advisor, Professor Karen Willcox, for giving me the opportunity to work with her and guiding and encouraging me throughout the course of this thesis work.

I am most grateful to Cuong Nguyen for always answering my multiple reduced order modeling questions with a smile on his face and never being too busy to help me out when I was stuck. I would also like to thank Thomas Richter for suggesting the combustion problem used in this thesis and Krzysztof Fidkowski for reading the first chapters of my thesis and offering valuable comments and suggestions.

I would especially like to thank Sonja for all my non-PDE-related life at MIT: the movies, the cooking, the trips, the card games, the good manners tips and, of course, the German lessons... Ich mag deinen {beep} sehr... I also need to thank Manuel and Macarena for feeding me from time to time and helping me stay in touch with the real world where people get married, have babies and don't care about convergence rates and bugs in their code.

I must also thank the faculty and friends at ICAI and my GE friends for the help and encouragement to apply to grad school. In particular, I would like to thank Prof. Alberto Carnicero and Prof. Ángela Jiménez Casas for their patience with my requests for letters of recommendation. To Fred Moody — thanks for your support with the application process and keeping in touch during this time at MIT. Thanks also to Dan, Mark, Matt, Teddy and all the GE Edison crew for making me feel at home when I first came to the US.

I must specially thank Zu for standing by me, loving me and giving me the freedom to pursue my dreams during all this time. Finally, I must thank my family: Mom for her support and not always requested (although always welcomed) advice and opinions, Dad for giving me so many opportunities (including coming to the US), Javier for entertaining me with his blog posts and for all the LaTeX tips, Carlos for always being unpredictable and living life on the edge (which makes for a lot of funny stories), Elena for always being happy and having something new to share over the phone and Ela for always caring about her favorite grandson.

This work was supported in part by the Singapore-MIT Alliance Computational Engineering Programme, the Air Force Office of Sponsored Research under grant FA9550-06-0271, program director Dr. F. Fahroo, and a Graduate Fellowship from the Caja Madrid Foundation.

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

## 1.1 Motivation

There are multiple instances in science and engineering where outputs of interest are evaluated by solving one or several partial differential equations (PDEs) that are parametrized in terms of a set of inputs. Recent years have seen significant improvements in well-established, "classical" numerical techniques for solving PDEs such as the finite element method (FEM), finite difference method (FDM) or boundary element method (BEM). However, simulation of complex systems using PDEs usually leads to very large numerical models that are computationally expensive to solve. This poses significant challenges when the outputs of interest need to be evaluated repeatedly for different values of the input parameters such as in optimal design or probabilistic analysis applications (referred to as the "multiple-query" context in [47]). Classical numerical techniques also face challenges that are often impossible to overcome when the outputs of interest have to be evaluated as soon as the inputs are available such as in control system applications (referred to as the "real-time" context in [47]).

Model order reduction is a powerful technique that permits the construction of simplified, cost-efficient representations of complex, large-scale systems while preserving their input-output behavior. Applying model order reduction methods to the large systems of equations that result from the discretization of PDEs using classical techniques has proven to be a viable way of tackling the challenges posed by the multiple-query and real-time applications mentioned earlier. Several model order reduction methods exist, for example, modal truncation [60], proper orthogonal decomposition (POD) [28, 54], balanced truncation [39], Krylov-subspace methods [15, 18, 26], reduced basis methods [45], and a quasi-convex optimization approach [56]. These methods have been applied successfully to numerous disciplines in science and engineering. However, there are still two main challenges that have not been fully addressed: the systematic sampling of high-dimensional input spaces and the efficient representation of nonlinearities in the underlying PDEs.

The savings in computational cost that can be achieved when applying conventional model order reduction techniques to nonlinear systems is often modest at best because the operation count required for evaluating the nonlinear terms is a function of the size of the full order model. At the same time, the task of reproducing the input-output behavior of the high-fidelity, large-scale system over a wide range of input parameter values is a challenging one because it requires an efficient sampling algorithm that scales well with the number of parameters (the exponential scaling associated with standard grid-based sampling methods is unacceptable for sampling high-dimensional input spaces). In this thesis we present a methodology that combines systematic sampling of the input parameter space with efficient representation of the nonlinearities in the governing PDEs. This combination enables the application of model order reduction to a class of problems that could not be previously tackled. The development of efficient model order reduction techniques for this class of high-dimensional, nonlinear problems in turn enables the application of the computationally intensive probabilistic framework for solving inverse problems that are intractable when using high-fidelity models.

The probabilistic approach to inverse problems has been applied successfully in numerous settings including tomography [2, 29], geophysics [16, 51], error propagation [20] and contaminant transport models [35]. However, the extreme computational cost required for exploring the input parameter domain using stochastic sampling methods is an open issue that still remains unanswered. Recently, the use of lower-fidelity models has been proposed in order to decrease the cost of the probabilistic methodology [2, 30]. However, these low-fidelity models are not obtained via systematic model reduction methods; instead, they are built using traditional numerical methods — i.e., FEM, FDM, BEM — but use meshes that are coarser than usual so that the resulting problem becomes tractable. The limitations of this approach are evident since the level of mesh coarsening required for decreasing the computational cost of these problems to acceptable levels typically results in large errors that are often hard to quantify or even yield unstable numerical schemes. In this thesis we show that efficient reduced order models can be used for solving inverse problems governed by nonlinear PDEs, yielding the same accuracy as high-fidelity models at a computational cost that is several orders of magnitudes less than that of traditional numerical techniques.

## 1.2 Application Considered in this Thesis: Nonlinear Convection-Diffusion-Reaction PDE

The methodology presented in this thesis is general and, therefore, valid for performing model order reduction of a wide range of nonlinear problems over high-dimensional parametric input spaces. However, we have chosen to illustrate its performance by applying it to a particular example instead of describing it in a more abstract manner. The problem chosen in this thesis is a highly nonlinear combustion problem governed by a convection-diffusion-reaction PDE which presents all the challenges discussed in Section 1.1 for applying conventional model order reduction techniques (i.e., highly nonlinear behavior and outputs of interest that must be evaluated over a wide range of input parameters).

Modeling and simulation of reactive flows is an important issue in scientific computation. In particular, modeling of combustion reactions presents several challenges due to the extreme stiffness of these problems, which usually requires very fine spatial and temporal discretizations. In general, obtaining a solution of a combustion problem requires significant computational effort, which means that the real-time and many-query applications discussed in Section 1.1 often become intractable when using "classical" numerical techniques (e.g., FEM/BEM/FDM). Furthermore, the exponential nonlinearities that appear in the underlying partial differential equations have posed significant difficulties for applying any of the reduced order modeling techniques mentioned in Section 1.1.

The model order reduction technique chosen in this thesis is the Reduced Basis (RB) method. This method leads itself to an efficient offline-online implementation that has been successfully used for obtaining rapid and reliable predictions of PDE-induced input-output relationships in applications where the governing equations are linear [25, 33, 47, 49] or, at most, quadratically nonlinear functions of the solution [44, 61, 62]. It has not been until very recently that Patera and co-workers have developed a framework that allows the use of the RB methodology in problems with arbitrary nonlinearities [24, 41, 43].

The estimation of reaction rates or Arrhenius parameters and the estimation of diffusion coefficients in combustion reactions based on experimental measurements of system conditions requires extensive exploration of parameter space (especially when using the Bayesian framework for solving the inverse problem) and can therefore be used as a typical example of the many-query context. For an overview of parameter estimation problems in chemistry, we refer to the book by Englezos and Kalogerakis [14], which gives multiple applications of parameter identification in the framework of ordinary differential equations. Parameter estimation problems for reactive flows in one space dimension are treated, for instance, by Bock *et al.* [64]. Parameter estimation for multidimensional computation of flames using finite element methods has been performed by Becker *et al.* [4].

An example of the real-time context is found in optimal control problems of combustion reactions where timeliness and reliability are critical for the efficient control of the process. Model-based control of combustion dynamics is discussed for example by Ghoniem *et al.* [63].

As discussed above, both the real-time and many-query contexts in combustion modeling present significant challenges to classical numerical techniques due to the fact that these techniques use dense approximation subspaces for finding the solution to the underlying PDE. Even though Green and co-workers have developed the "adaptive chemistry" methodology for reducing the complexity of combustion computations [22, 46], performing fast and reliable reacting flow calculations remains a very challenging task.

In this thesis we show that the reduced basis framework for nonlinear parametrized PDEs that was presented in [24, 41, 43] can be used in conjunction with an efficient sampling methodology based on the algorithm developed in [8, 9] in order to obtain a very efficient tool for satisfying the requirements of the real-time and many-query contexts in combustion modeling while preserving the accuracy of classical (and computationally expensive) numerical techniques. In particular, the efficient model order reduction methodology presented in this thesis is applied to the solution and uncertainty quantification of inverse combustion problems using a Bayesian approach. This is a typical example of a many-query application that would be intractable if classical techniques were used. Once again we emphasize that, even though the work presented in this thesis deals with model order reduction and uncertainty quantification in inverse problems of reactive flows, the methodology discussed herein is general and can be applied to any nonlinear problem governed by a parametrized PDE.

## 1.3 The Reduced Basis Method

As discussed in Section 1.1, there are multiple examples in science and engineering of systems that are modeled using *input-parametrized* PDEs. In general, the output of interest is not the full field that is the solution of the PDE, but rather certain outputs, which can be evaluated as functionals of this field. The input parameters characterize the different system configurations. In order to compute the value of the output that is generated by a particular value of the inputs one must compute the field that relates inputs and outputs by solving the underlying PDE.

The solution to the PDE that governs the system behavior is usually expensive to compute. However, as discussed in Section 1.1, there are many situations that require rapid, yet reliable, evaluation of the input-output relationships governed by the underlying PDE. The reduced basis method is one of several possible model order reduction tools that can be used for satisfying these requirements.

The reduced basis method takes advantage of the fact that, even though the spaces used for computing the solution to the underlying PDE are generally of very high dimension, the actual solution field resides in a parametrically induced manifold which is typically of low dimension. In the reduced basis method, the low dimension space where the solution to the PDE actually resides is approximated as the span of precomputed solutions or "snapshots" as originally referred to by Sirovich [54]. Therefore, the reduced basis method tries to represent any solution to the PDE of interest as a linear combination of solutions that have been precomputed using any classical numerical technique. Efficient computation of snapshots for high-dimensional input spaces is a challenging problem that is successfully addressed in this thesis.

The reader is referred to [47] for a historical perspective of the reduced basis method and a detailed description of the methodology. As discussed in Section 1.2, the traditional reduced basis method has been successfully applied to numerous problems governed by linear PDEs that are affine in the parameter or nonlinear PDEs that are at most quadratically nonlinear in the field variable. However, when applied to highly nonlinear problems, such as combustion equations, the traditional reduced basis methodology becomes inefficient and the achievable time savings do not justify its use versus classical techniques.

In this thesis we use the extension of the traditional reduced basis method to nonlinear problems proposed in [24, 41, 43]. The basic ingredient of this methodology is the replacement of the nonlinear term by a *coefficient-function approximation*, which can be built using the techniques described in [3] or [42]. This coefficient function approximation allows to recover the low computational complexity associated with the evaluation of the terms required during the online phase of the simulation.

## 1.4 Thesis Scope

In this thesis, the governing PDE is a convection-diffusion-reaction equation that includes an exponential Arrhenius term which is typically used for modeling combustion processes. The field variable that is computed by solving the governing PDE represents the molar concentration of fuel at every point inside the domain of interest, typically a combustion chamber.

The Arrhenius term in the convection-diffusion-reaction equation comes from an empirical law and the value of the parameters that appear in it are generally not known *a priori*. Therefore, the underlying PDE is naturally parametrized in terms of the Arrhenius parameters. Given a particular value for the Arrhenius parameters, the molar concentration of fuel at every point inside the combustion chamber is computed. The outputs of interest are the average concentrations of fuel along vertical lines located throughout the computational domain, which model the location of laser measurement sensors in the physical combustion chamber.

The reduced basis method, online-offline computational procedure and efficient sampling algorithm with associated *a posteriori* error estimator for the PDE of interest are developed in this thesis.

Finally, given a set of experimental measurements representing fuel concentrations along lines in the combustion chamber, an inverse problem for determining the value of the Arrhenius parameters associated with the combustion reaction that generated those measurements is formulated. The inverse problem is formulated using both a deterministic approach and the probabilistic Bayesian framework. The multiple forward problem solutions required by the probabilistic methodology are computed using the reduced basis approximation developed in earlier chapters, showing that the use of a reduced order model for solving the inverse problem yields the same results as a high-fidelity model at a fraction of the cost.

## 1.5 Thesis Outline

In Chapter 2, the weak form of the convection-diffusion-reaction equation is presented and the Streamline Upwind / Petrov-Galerkin method used for stabilizing the formulation is derived. The finite element implementation and results are presented at the end of the chapter. In Chapter 3, the reduced basis methodology and online-offline computational algorithm are developed. In Chapter 4, a greedy algorithm for selecting the reduced basis in an efficient manner is proposed and compared against uniform grid sampling for problems with 2 and 3 parameters. The inverse problem is formulated and solved in Chapter 5. Finally, Chapter 6 concludes the thesis with suggestions for future work.

# Chapter 2

# Mathematical Model and Finite Element Solution

## 2.1 Governing Equation

In this thesis we consider the following stationary nonlinear convection-diffusion-reaction equation for the field variable $u$ in a domain $\Omega \subset \mathbb{R}^2$ with a divergence-free velocity field $\boldsymbol{U}$ and a diffusion coefficient $\kappa$:

$$\boldsymbol{U} \cdot \nabla u - \nabla(\kappa \nabla u) + s(u; \boldsymbol{\mu}) = f, \tag{2.1}$$

where $f$ is a linear source term and, as is usual in combustion problems, the nonlinear reaction term $s(u; \boldsymbol{\mu})$ is of Arrhenius type,

$$s(u; \boldsymbol{\mu}) = Au(c - u)e^{\frac{-E}{d-u}}. \tag{2.2}$$

where $c$, $d$ are known constants and the system parameters $\boldsymbol{\mu} = (\ln(A), E)$ can vary within the system parameters domain $\mathcal{D} \equiv [5.00, 7.25] \times [0.05, 0.15] \subset \mathbb{R}^2$

We introduce a Dirichlet boundary condition that specifies the value of the field variable $u = u_D$ at the inflow boundary $\Gamma_{\text{in}} \subset \partial\Omega$, where $\partial\Omega$ denotes the boundary of the domain $\Omega$:

$$u = u_D \quad \text{on} \quad \Gamma_{\text{in}}. \tag{2.3}$$

Finally we use a Neumann boundary condition to specify the diffusive flux on all the other boundaries of the domain:

$$\nabla u \cdot \hat{\mathbf{n}} = 0 \quad \text{on} \quad \partial\Omega \setminus \Gamma_{\text{in}}, \tag{2.4}$$

In the reaction term, $u$ represents the mole fraction of fuel, while the mole fraction of the oxidizer is $(c - u)$. Since the Arrhenius law is a heuristic law and cannot be derived from first principles, parameters $A$ and $E$ are *a priori* unknown and have to be calibrated. This parameter fitting is usually done by comparison of experimental data and simulation results. Therefore, this example is well suited for the many-query context described in Section 1.1. In general, the outputs of interest of our model will be a set of computed quantities $\boldsymbol{o}(u) \in \mathbb{R}^{N_s}$ that we will compare against an equivalent set of empirical measurements $\bar{\boldsymbol{o}} \in \mathbb{R}^{N_s}$ in order to estimate the values of $A$ and $E$. Typically, these empirical quantities are obtained by performing laser measurements of mean concentrations along fixed lines inside the combustion chamber as shown in Figure 2-1. Here, $N_s$ is the number of sensor locations in the combustion chamber.

## 2.2 Weak Form Without Stabilization

In what follows, we will use some notation borrowed from Patera and co-workers [23, 24, 41, 47]: the superindex "e" refers to the "exact" solution to the mathematical model given by equations (2.1) – (2.4). In Section 2.3 we will introduce a "truth" approximation that will refer to the solution of the discretized weak form of equations (2.1) – (2.4). The "truth" approximation will bear no superscript. The "exact" weak form of the combustion model from the previous section can be stated as follows: for any $\boldsymbol{\mu} \in \mathcal{D} \subset \mathbb{R}^2$, find $o_i^e(\boldsymbol{\mu}) = \ell_i^O(u^e(\boldsymbol{\mu}))$, $i = 1, \ldots, N_s$ where $u^e(\boldsymbol{\mu}) \in X^{D,e} = \{ u \in H^1(\Omega) \mid u|_{\Gamma_{\text{in}}} = u_D \}$ satisfies the weak form of the $\boldsymbol{\mu}$-

parametrized nonlinear PDE (2.1)

$$a(u^e(\boldsymbol{\mu}), v) + \int_\Omega s(u^e(\boldsymbol{\mu}); \boldsymbol{\mu})v = \ell(v), \quad \forall v \in X^e. \tag{2.5}$$

In the weak form above, $X^e = \{v \in H^1(\Omega) \mid v|_{\Gamma_{in}} = 0\}$ and $a(\cdot,\cdot)$ and $\ell(\cdot)$, $\ell^O(\cdot)$ are $X^e$-continuous bounded bilinear and linear functionals respectively. In our model problem we will assume that the diffusivity $\kappa$ is constant throughout the entire domain $\Omega$. Therefore, the bilinear and linear functionals for our problem are given by

$$a(w, v) = \int_\Omega v\boldsymbol{U} \cdot \nabla w \, d\Omega + \kappa \int_\Omega \nabla w \cdot \nabla v \, d\Omega, \tag{2.6}$$

$$\ell(v) = \int_\Omega vf \, d\Omega \tag{2.7}$$

$$\ell_i^O(v) = \int_{\Gamma_i} v \, ds, \quad i = 1, \ldots, N_s. \tag{2.8}$$

In this problem there are $N_s$ different outputs of interest that represent the average value of the solution $u$ over each of the measurement lines $\Gamma_i, i = 1, \ldots, N_s$ shown in Figure 2-1.

Also note that, since all Neumann boundary conditions (2.4) are homogeneous, they do not contribute to the linear form $\ell(v)$.

## 2.3 Weak Form Using Streamline Upwind / Petrov-Galerkin Stabilization

### 2.3.1 "Truth" Approximation

After arriving at the weak form (2.5) one could think of discretizing it using a sufficiently fine mesh in order to obtain a suitably accurate "truth" approximation to the exact solution. As will be discussed later, a direct discretization of (2.5) is unstable for low values of $\kappa$ due to the convective nature of the governing PDE. However, for now we shall proceed with a naive discretization in order to introduce the finite element triangulation and interpolation spaces that will be used later for obtaining the stable version of the "truth" formulation.

The first step for obtaining the discretized version of (2.5) is to discretize the domain $\Omega$ into element domains. Let $\mathcal{T}_h(\Omega)$ be a regular partition, also called *triangulation*, of $\Omega$ into $N_{\text{elem}}$ subdomains $T_h^k$, $k = 1, \ldots, N_{\text{elem}}$, such that each subdomain has a piecewise smooth boundary $\Gamma_h = \partial T_h$, and $h$ is a characteristic mesh size (diam$(T_h) \leq h$ for all elements).

We can then introduce our fine reference finite element interpolation spaces, $X \subset X^e$ and $X^D \subset X^{D,e}$ of large dimension $\mathcal{N}$. In general, richer interpolation spaces will produce more accurate "truth" approximations (i.e., closer to the exact solution introduced in Section 2.2) but will also increase the cost of computing the solution. Therefore, a compromise is needed between the accuracy of the "truth" approximation and the computational resources required to obtain it. In Chapter 3 the RB approximation will be built upon the "truth" solution and all errors will be measured with respect to this "truth" so $\mathcal{N}$ should be large enough that the "truth" solution meets the accuracy requirements of our specific application. When using the "truth" formulation for building a RB approximation it is common to err on the safe side and use a conservatively large value of $\mathcal{N}$ since the online cost of computing the RB approximation is independent of the dimension of the finite element interpolation space and, therefore, the value of $\mathcal{N}$ only increases the cost of the offline phase where the reduced bases are constructed.

The finite element interpolation spaces $X$ and $X^D$ used for computing the "truth" approximation are defined as

$$X = \{\, v \in H^1(\Omega) \mid v|_{T_h} \in \mathbb{P}_m(T_h),\ \forall T_h \in \mathcal{T}_h \text{ and } v = 0 \text{ on } \Gamma_{\text{in}} \,\} \tag{2.9}$$

$$X^D = \{\, u \in H^1(\Omega) \mid u|_{T_h} \in \mathbb{P}_m(T_h),\ \forall T_h \in \mathcal{T}_h \text{ and } u = u_D \text{ on } \Gamma_{\text{in}} \,\} \tag{2.10}$$

where $\mathbb{P}_m$ is the *finite element interpolation space* (usually the set of polynomials of total degree $\leq m$). In our problem we will use piecewise-linear finite element interpolation spaces, thus $m = 1$. The "truth" approximation can therefore be stated as follows: given any $\boldsymbol{\mu} \in \mathcal{D}$, evaluate

$$o_i(\boldsymbol{\mu}) = \ell_i^O(u(\boldsymbol{\mu})), \quad i = 1, \dots, N_s, \tag{2.11}$$

where $u(\boldsymbol{\mu}) \in X^D$ is the solution of

$$a(u(\boldsymbol{\mu}), v) + \int_\Omega s(u(\boldsymbol{\mu}); \boldsymbol{\mu})\, v\, d\Omega = \ell(v), \quad \forall v \in X. \tag{2.12}$$

Note that the linear and bilinear forms and the nonlinear term used in the "truth" approximation have the same names as those used in the exact formulation (2.5). Being rigorous, this implicitly assumes that all quadratures are exact: otherwise we would need $a^e$, $\ell^e$, $s^e$ in Section 2.2 and $a$, $\ell$, $s$ in the current section. In practice, we will use approximate quadrature rules for discretizing all the terms that appear in the "truth" approximation so there will be additional errors between the exact solution and the "truth" used to build the RB approximation. In this thesis we use the term "truth" to refer to a discrete approximation of the exact solution that satisfies the accuracy requirements of our specific application. We do not imply any level of accuracy by using the word "truth" and, in fact, the "truth" approximation could be far from the exact solution as long as it is "good enough" for our needs.

### 2.3.2 Streamline Upwind / Petrov-Galerkin Stabilization

As mentioned at the beginning of the current section, if we tried to solve the discretized weak form (2.12) we would find that the solution is corrupted by non-physical oscillations when the problem is dominated by convection (high flow velocities or low diffusivities). As discussed in [13], this oscillatory behavior is caused by the negative numerical diffusion inherent in the Galerkin finite element method. This phenomenon is analogous to the spurious oscillations found in solutions to convection-dominated problems when using central differences instead of upwinding schemes.

To characterize the relative importance of convective and diffusive terms in a given flow we introduce the mesh Péclet number

$$Pe = \frac{\|\boldsymbol{U}\|h}{2\kappa} \tag{2.13}$$

where $\|\cdot\|$ denotes the $\ell^2$-norm, $\boldsymbol{U}$ is the convective flow velocity, $h$ is the element size and $\kappa$ is the diffusivity as defined in previous sections. Note that, in general, the flow velocity field and the element sizes will not be constant throughout the entire mesh. Therefore, each element will have a different elemental Péclet number. In this thesis we will refer to the mesh Péclet number as the most unfavorable of all the elemental Péclet numbers. As shown in [13], the Galerkin solution is corrupted by spurious oscillations when the Péclet number is larger than one. Therefore, we will define the mesh Péclet number as the largest of the elemental Péclet numbers.

Several methods exist for counterbalancing the negative numerical diffusion introduced by the Galerkin approximation when the Péclet number is greater than one. Herein we use the Streamline-Upwind Petrov-Galerkin (SUPG) technique proposed by Brooks and Hughes [6] where an extra term over the element interiors is added to the Galerkin weak form. It is worth noting that the additional term is a function of the full residual of the differential equation (not only the convective term) in order to ensure consistency. The stabilized version of the weak form (2.12) is shown below

$$a(u(\boldsymbol{\mu}), v) + \int_\Omega s(u(\boldsymbol{\mu}); \boldsymbol{\mu})v + \underbrace{\sum_{k=1}^{N_{\mathrm{elem}}} \int_{T_h^k} \mathcal{P}(v)\,\tau\,\mathcal{R}(u)}_{\text{stabilization term}} = l(v), \quad \forall v \in X, \tag{2.14}$$

where $\mathcal{R}(u)$ is the residual of the differential equation as shown below

$$\mathcal{R}(u) = \boldsymbol{U}\cdot\nabla u - \kappa\nabla^2 u + s(u; \boldsymbol{\mu}) - f \tag{2.15}$$

Since the solution $u(\boldsymbol{\mu})$ in equation (2.14) already resides in our finite element approximation space $X^D \subset X^{D,e}$, the residual $\mathcal{R}(u)$ is computed only for each element interior. $\mathcal{P}(\cdot)$ is a certain operator applied to the the test function and $\tau$ is the stabilization parameter. Equation (2.14) provides a general form for all consistent stabilization techniques. Each technique is characterized by a different definition of the operator $\mathcal{P}(\cdot)$. As discussed before, in this thesis we shall use the SUPG method, which is defined by taking

$$\mathcal{P}(v) = \boldsymbol{U} \cdot \nabla v \tag{2.16}$$

For a general treatment of consistent stabilization methods of the form (2.14) we refer to [10]. For the weak formulation given by (2.14), it can be shown that the stabilization term has the effect of inducing numerical diffusion in the streamline direction, which is equivalent to using an upwinding scheme or using modified weighting functions such that the element upstream of a node is weighted more heavily than the element downstream of a node. For a rigorous variational interpretation of the SUPG method we refer to [55].

### 2.3.3 Stabilization Parameter

The weak form (2.14) contains a stabilization parameter, $\tau$, that is critical for the proper convergence of the SUPG method. The literature contains several definitions of $\tau$ that can be proven to be optimal for simple, linear convection-diffusion or convection-diffusion-reaction equations. However, no definition of the stabilization parameter is readily available for the nonlinear PDE (2.1). Since none of the different definitions of $\tau$ available in the literature have been validated for our specific combustion equation, several of them are presented below in order to assess their performance. The solver written to solve equation (2.1) is capable of using any of the four implementations discussed in this section.

#### Definition by Brooks and Hughes [6]

This definition was originally derived based on the one-dimensional convection-diffusion equation, for which it can be shown to achieve superconvergence in the form of nodally exact results. Even though it was originally intended for the convection-diffusion equation, it has been tested successfully on more complex problems such as the Navier-Stokes equations. In this case, the stabilization parameter is defined as follows,

$$\tau = \frac{\tilde{\kappa}}{\|\boldsymbol{U}\|^2} \tag{2.17}$$

$$\tilde{\kappa} = \frac{1}{2}\tilde{\xi}\,\|\boldsymbol{U}\|\,h \tag{2.18}$$

$$\tilde{\xi} = \coth(Pe) - \frac{1}{Pe} \tag{2.19}$$

It is worth noting once again that, in the most general case, each element of the triangulation will have a different stabilization parameter. Generally, the flow velocity used for computing the element-level Péclet number used in this definition is computed at the barycenter of the element.

### Definition by Tezduyar and Osawa [58]

Once again, this definition was derived in the context of the convection-diffusion equation although an analogous definition was tested successfully on the unsteady, incompressible Navier-Stokes equations.

The main advantage of this definition is that $\tau$ is computed from element-level matrices without separately computing local length scales (i.e. element sizes). For the steady case, $\tau$ is given by

$$\tau = \left(\frac{1}{\tau_{S1}^r} + \frac{1}{\tau_{S2}^r}\right)^{-\frac{1}{r}} \tag{2.20}$$

$$\tau_{S1} = \frac{\|\boldsymbol{K}\|}{\|\boldsymbol{K}_{SUPG}\|} \tag{2.21}$$

$$\tau_{S2} = Re \cdot \tau_{S1} \tag{2.22}$$

$$Re = \frac{\|\boldsymbol{U}\|^2}{\kappa}\frac{\|\boldsymbol{K}\|}{\|\boldsymbol{K}_{SUPG}\|} \tag{2.23}$$

where $r$ is a positive integer and $\|\cdot\|$ indicates any matrix norm. As shown by Donea and Huerta [13], $r = 2$ has been tested successfully in different applications.

For each element in the mesh, the matrix $\boldsymbol{K}$ is the elemental matrix associated with the convective term, $\int_{T_h} v\,\boldsymbol{U}\cdot\nabla u\,d\Omega$ and $\boldsymbol{K}_{SUPG}$ is the elemental matrix associated with the SUPG stabilization of the convective term, $\int_{T_h} \boldsymbol{U} \cdot \nabla v \ \boldsymbol{U} \cdot \nabla u \, d\Omega$.

### Definition by Mizukami [38]

Mizukami provides a definition of the stabilization parameter that is specifically intended to be used with triangular linear elements. In this case, $\tau$ is a three-element vector where each component corresponds to the stabilization parameter associated with one node of the element. The definition of the stabilization parameter in this case is given by

$$\tau_i = \frac{1}{2}\left(\frac{M_{ii}}{\max_j |K^1_{ij}| + K^2_{ij}}\right), \quad i,j = 1,2,3 \tag{2.24}$$

where $\boldsymbol{M}$ is the element-level mass matrix and $\boldsymbol{K}^1$ and $\boldsymbol{K}^2$ are the convective and diffusive parts of the elemental stiffness matrix corresponding to the element-level integration of the terms $\int_{T_h} v \ \boldsymbol{U} \cdot \nabla u \, d\Omega$ and $\int_{T_h} \kappa \nabla u \cdot \nabla v \, d\Omega$ respectively.

### Definition by Codina [11] and Shakib [52]

The SUPG stabilization parameters proposed by Hughes, Tezduyar and Mizukami were derived for the pure convection-diffusion case without a reaction term. The obvious question that arises in our problem is how the presence of a reaction term may affect the SUPG stabilization parameter. Codina [11] and Shakib and co-workers [52] have proposed expressions for $\tau$ that include the effect of linear reaction terms. The linear convection-diffusion-reaction equation for which the expressions of $\tau$ derived by Codina and Shakib are strictly applicable is shown below

$$\boldsymbol{U} \cdot \nabla u - \kappa \nabla^2 u + \sigma u = f \tag{2.25}$$

where $\sigma u$ is the linear reaction term. For equation (2.25), Codina proposes the following expression for computing $\tau$

$$\tau = \frac{h}{2U}\left(1 + \frac{1}{Pe} + \frac{h\sigma}{2U}\right)^{-1}, \tag{2.26}$$

whereas Shakib and co-workers suggest using

$$\tau = \frac{h}{2U}\left(1 + \frac{9}{Pe^2} + \left(\frac{h\sigma}{2U}\right)^2\right)^{-\frac{1}{2}}. \tag{2.27}$$

In the previous equations $U$ denotes the $\ell^2$-norm of the velocity vector $\boldsymbol{U}$. Based on equations (2.26) and (2.27) it is easy to see that, for a given ratio $U/h$, the maximum stabilization is required for very large Péclet numbers (highly convective flows) and negative reaction rates. Large positive reaction rates tend to stabilize the weak formulation whereas negative reaction rates require increased values of $\tau$ in order to achieve a stable scheme.

Recall that the nonlinear reaction term in our problem of interest is given by equation (2.2). A piecewise linear representation of $s(u)$ can be obtained by performing a Taylor series expansion about $u = 0$ and $u = c$. This piecewise linear representation, denoted by $\tilde{s}(u)$ is given by

$$\tilde{s}(u) = \begin{cases} \sigma_1 u, & \text{if } 0 \le u < u_c; \\ \sigma_2 (u - c) & \text{if } u_c \le u < c, \end{cases} \tag{2.28}$$

where

$$\sigma_1 = Ace^{-\frac{E}{d}} \tag{2.29}$$

$$\sigma_2 = -Ace^{-\frac{E}{d-c}} \tag{2.30}$$

$$u_c = \frac{c}{1 + e^{\frac{Ec}{d(d-c)}}} \tag{2.31}$$

Note that the reaction rates $\sigma_1$ and $\sigma_2$ given in equations (2.29) and (2.30) can now be used in the definitions of $\tau$ proposed by Codina and Shakib. The only difficulty is that the stabilization parameters for all the elements must be calculated *before* a solution is obtained, which means that one cannot evaluate the expression (2.28) in order to determine which reaction rate corresponds to each element. An iterative process would be required for computing $\tau$. Since the exact value of the stabilization parameter is not critical for obtaining a correct solution, the implementation of the SUPG solver used in this thesis follows a conservative approach and uses the value of $\sigma_2$ as the reaction rate. This may add additional stabilization in some regions but its effect is negligible in the final solution.

We wish to emphasize that the linearization of the Arrhenius term is only performed in order to account for reaction effects on the stabilization parameter when using the definitions proposed by Codina and Shakib. The actual problem being solved always contains the fully nonlinear exponential combustion term $s(u)$ regardless of the stabilization method used for solving it.

### Conclusions Regarding Stabilization Parameter

No suggestions were found in the literature for defining the SUPG stabilization parameter for our particular problem. Most of the expressions for $\tau$ that are available in the literature have been derived and tested on linear or, at most, quadratically nonlinear problems so it was not clear how they would perform for stabilizing equation (2.1). Therefore, four different definitions were implemented in order to assess their performance when used for solving a nonlinear combustion PDE.

After performing several tests, it was observed that the definitions by Brooks and Hughes (see equation (2.17)) and Codina and Shakib (see equations (2.26) and (2.27)) provide the best stabilization for our particular problem. All the numerical results included in this thesis have been computed using the definition by Brooks and Hughes unless stated otherwise.

## 2.4 Finite Element Implementation and Solution Method

### 2.4.1 Galerkin Projection

As discussed in section 2.3, in this thesis we use piecewise-linear finite element approximation spaces $X$ and $X^D$. The typical "hat" basis functions that span these spaces are defined as

$$\varphi_i \in X, \quad \varphi_i(x_j) = \delta_{ij}, \quad 1 \le i, j \le \mathcal{N}. \tag{2.32}$$

For this particular case, the second order derivatives in the residual of the PDE (see equation (2.15)) vanish so the "truth" approximation can be formulated as follows: given any $\boldsymbol{\mu} \in \mathcal{D}$, evaluate

$$o_i(\boldsymbol{\mu}) = \ell_i^O(u(\boldsymbol{\mu})), \quad i = 1, \ldots, N_s, \tag{2.33}$$

where $u(\boldsymbol{\mu}) \in X^D$ is the solution of

$$\begin{aligned} &a(u(\boldsymbol{\mu}), v) + \int_\Omega s(u(\boldsymbol{\mu});\, \boldsymbol{\mu})\, v \, d\Omega \\ &+ \sum_{k=1}^{N_{\text{elem}}} \int_{T_h^k} (\boldsymbol{U} \cdot \nabla v)\ \tau_k\ [\boldsymbol{U} \cdot \nabla u(\boldsymbol{\mu}) + s(u(\boldsymbol{\mu});\, \boldsymbol{\mu}) - f]\ d\Omega = \ell(v), \quad \forall v \in X, \end{aligned} \tag{2.34}$$

where the bilinear and linear forms $a(\cdot,\cdot)$, $\ell(\cdot)$ and $\ell^O(\cdot)$ were defined in equations (2.6)–(2.8).

### 2.4.2 Nonlinear System of Algebraic Equations

The weak form (2.34) yields the following nonlinear system of equations

$$\boldsymbol{A}\boldsymbol{u}(\boldsymbol{\mu}) + \boldsymbol{S}\left(\boldsymbol{u}(\boldsymbol{\mu}); \boldsymbol{\mu}\right) = \boldsymbol{F} \tag{2.35}$$

where the vector $\boldsymbol{u} \in \mathbb{R}^{\mathcal{N}}$ contains the nodal values of the solution $u(x; \boldsymbol{\mu})$. The stiffness matrix, $\boldsymbol{A} \in \mathbb{R}^{\mathcal{N}\times\mathcal{N}}$, load vector $\boldsymbol{F} \in \mathbb{R}^{\mathcal{N}}$ and nonlinear reaction vector, $\boldsymbol{S}\left(\boldsymbol{u}(\boldsymbol{\mu}); \boldsymbol{\mu}\right) \in \mathbb{R}^{\mathcal{N}}$ are given by

$$A_{ij} = a(\varphi_j, \varphi_i) + \sum_{k=1}^{N_{\text{elem}}} \int_{T_h^k} (\boldsymbol{U} \cdot \nabla\varphi_i)\ \tau_k\ (\boldsymbol{U} \cdot \nabla\varphi_j)\ d\Omega \quad 1 \le i, j \le \mathcal{N} \tag{2.36}$$

$$F_i = \ell(\varphi_i) + \sum_{k=1}^{N_{\text{elem}}} \int_{T_h^k} (\boldsymbol{U} \cdot \nabla\varphi_i)\ \tau_k\, f\, d\Omega \quad 1 \le i \le \mathcal{N} \tag{2.37}$$

$$\begin{aligned} S_i\left(\boldsymbol{u}(\boldsymbol{\mu}); \boldsymbol{\mu}\right) = &\int_\Omega s\left(\sum_{j=1}^{\mathcal{N}} u_j\varphi_j;\ \boldsymbol{\mu}\right)\ \varphi_i\, d\Omega \\ &+ \sum_{k=1}^{N_{\text{elem}}} \int_{T_h^k} (\boldsymbol{U} \cdot \nabla\varphi_i)\ \tau_k\ s\left(\sum_{j=1}^{\mathcal{N}} u_j\varphi_j;\ \boldsymbol{\mu}\right)\, d\Omega \quad 1 \le i \le \mathcal{N} \end{aligned} \tag{2.38}$$

The nonlinear system of equations (2.35) is solved using a Newton iterative scheme: given a current iterate $\boldsymbol{u}^p$ we find an increment $\Delta\boldsymbol{u}^p$ such that

$$\left.\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{u}}\right|_{\boldsymbol{u}^p} \Delta\boldsymbol{u}^p = -\boldsymbol{R}\left(\boldsymbol{u}^p(\boldsymbol{\mu}); \boldsymbol{\mu}\right), \tag{2.39}$$

where $\boldsymbol{R}\left(\boldsymbol{u}^p(\boldsymbol{\mu}); \boldsymbol{\mu}\right)$ is the residual of the algebraic system of equations, defined as

$$\boldsymbol{R}\left(\boldsymbol{u}^p(\boldsymbol{\mu}); \boldsymbol{\mu}\right) = \boldsymbol{A}\boldsymbol{u}^p + \boldsymbol{S}\left(\boldsymbol{u}^p(\boldsymbol{\mu}); \boldsymbol{\mu}\right) - \boldsymbol{F}, \tag{2.40}$$

and $\left.\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{u}}\right|_{\boldsymbol{u}^p} = \boldsymbol{J}\left(\boldsymbol{u}^p(\boldsymbol{\mu}); \boldsymbol{\mu}\right) \in \mathbb{R}^{\mathcal{N}\times\mathcal{N}}$ is the Jacobian of the residual evaluated at the current iteration,

$$\boldsymbol{J}\left(\boldsymbol{u}^p(\boldsymbol{\mu}); \boldsymbol{\mu}\right) = \boldsymbol{A} + \left.\frac{\partial \boldsymbol{S}}{\partial \boldsymbol{u}}\right|_{\boldsymbol{u}^p}, \tag{2.41}$$

where the term $\left.\frac{\partial \boldsymbol{S}}{\partial \boldsymbol{u}}\right|_{\boldsymbol{u}^p}$ can be evaluated using a suitable quadrature rule as follows

$$\begin{aligned} \left.\frac{\partial S_i}{\partial u_j}\right|_{\boldsymbol{u}^p} = &\sum_{k=1}^{N_{\text{elem}}} \sum_{q=1}^{N_Q} w_q\, \varphi_j(\xi_1^q, \xi_2^q)\, J_{CT}^k(\xi_1^q, \xi_2^q)\ \left(W_i^k(\xi_1^q, \xi_2^q) + \varphi_i(\xi_1^q, \xi_2^q)\right) \\ &\cdot \left\{ A\, e^{-\frac{E}{d - u_{Q^q}^p}} \left( c - 2u_{Q^q}^p - \frac{E(c - u_{Q^q}^p)u_{Q^q}^p}{(u_{Q^q}^p - d)^2} \right) \right\}, \quad 1 \le i, j \le \mathcal{N} \end{aligned} \tag{2.42}$$

where $N_Q$ is the number of quadrature points in each element, $(\xi_1^q, \xi_2^q)$ are the coordinates of quadrature point $q$ in the master element, in the computational domain; $w_q$ is the weight associated with quadrature point $q$ and $J_{CT}^k(\xi_1^q, \xi_2^q)$ is determinant of the Jacobian of the coordinate transformation between element $k$ in physical space and the master element in the computational domain, evaluated at the quadrature point. The expression $u^p_{Q^q}$ represents the value of the solution at Newton's iteration $p$ evaluated at the quadrature point $q$, which is given by

$$u^p_{Q^q} = \sum_{i=1}^{\mathcal{N}} u^p_i \varphi_i(\xi^q_1, \xi^q_2) \tag{2.43}$$

Finally, $\boldsymbol{W}^k \in \mathbb{R}^3$ contains part of the contribution from the SUPG stabilization term and is given by

$$W^k_i = \sum_{j=1}^{2} \sum_{n=1}^{2} U_j \frac{\partial \mathcal{H}^k_i}{\partial \xi_n} \frac{\partial \xi_n}{\partial x_j}, \quad 1 \leq i \leq 3, \; 1 \leq k \leq N_{\text{elem}}. \tag{2.44}$$

where $\mathcal{H}^k_i$ represents the basis function associated to the local node $i$ of element $k$ restricted to $T^k_h$: $\mathcal{H}^k_i = \varphi_{\text{node}\, i}|_{T^k_h}$. Note that here $i$ refers to the local nodes of element $k$ $(i = 1, 2, 3)$.

After solving the linear system of equations (2.39) we can update the solution

$$\boldsymbol{u}^{p+1} = \boldsymbol{u}^p + \Delta \boldsymbol{u}^p, \tag{2.45}$$

and we iterate until the value of the residual is suitably small. Finally, we evaluate the vector of outputs, $\boldsymbol{o}(\boldsymbol{\mu}) \in \mathbb{R}^N_m$ as

$$\boldsymbol{o}(\boldsymbol{\mu}) = \boldsymbol{L}^O \boldsymbol{u}(\boldsymbol{\mu}), \tag{2.46}$$

where $\boldsymbol{L}^O \in \mathbb{R}^{N_s \times \mathcal{N}}$ is given by

$$L^O_{ij} = \int_{\Gamma_i} \varphi_j \, d\Gamma, \quad 1 \leq i \leq N_s, \; 1 \leq j \leq \mathcal{N} \tag{2.47}$$

### 2.4.3 Implementation Remarks for Newton Solver

The PDE solved in this thesis is extremely stiff so a naive updating scheme like the one presented in equation (2.45) will never converge. Different Newton solver implementations can be used to overcome this difficulty. Two of them are discussed in this section and used in this thesis.

The first solver is a damped Newton solver where the size of the solution step in every update is selected such that the norm of the residual decreases at every iteration. The algorithm is summarized below

```
$u = u^0$
while $\|\Delta u\| > \varepsilon_u$ and $\|R(u;\mu)\| > \varepsilon_R$
    $R = R(u;\mu)$
    $J = \left.\frac{\partial R(u;\mu)}{\partial u}\right|_u$
    $\Delta u = -J^{-1} R$
    $\alpha = 1.0$
    while $\|R\| \geq \|R(u+\Delta u;\mu)\|$
        $\alpha = 0.5\,\alpha$
        $R = R(u+\alpha\Delta u;\mu)$
    end
    $u = u + \alpha\Delta u$
end
```

The second Newton solver incorporates a load stepping algorithm that limits the nonlinearity in the case of nonconvergence. The philosophy is similar to the damped Newton algorithm but, instead of limiting the size of the step used for updating the solution, it limits the value of the nonlinear parameter vector $\mu$ such that the problem becomes closer to linear (for a purely linear problem any Newton solver obviously converges in just one step).

Both solvers offer similar performance. The damped solver was used for computing all the results included in this thesis unless noted otherwise.

## 2.5 Numerical Results

The domain of interest for our problem represents the interior of a combustion chamber. The full combustion chamber is shown in Figure 2-1. The fuel is injected into the chamber through the central pipe whereas the oxidizer is injected via the upper and lower tubes. Once inside the chamber, both fuel and oxidizer diffuse and the combustion reaction occurs.

Note that the computational domain only includes the part of the combustion chamber downstream of the injection pipes (region inside the dotted rectangle in Figure 2-1). Since both reactants are isolated from each other while flowing through the injection pipes, this region of the domain is not necessary for modeling the combustion reaction.

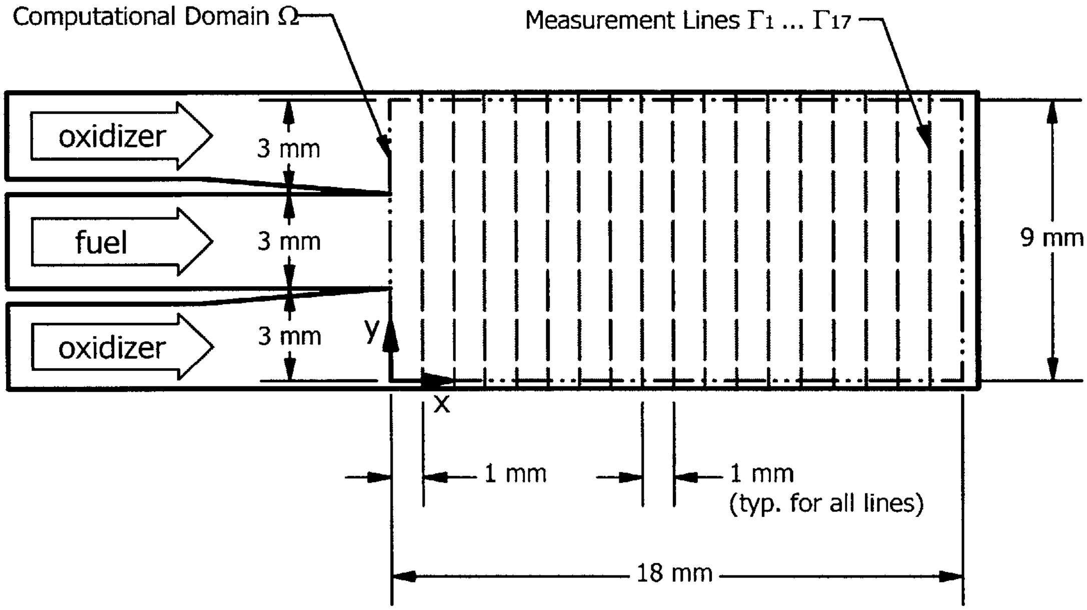

Figure 2-1: Reaction chamber configuration. Computational domain is enclosed inside dash-dotted lines. Dashed vertical lines indicate the lines where measurements are modeled.

Figure 2-1 also shows 17 vertical dashed lines spaced at uniform intervals of 1 mm. These are the lines along which laser measurements are taken in the physical chamber in order to estimate the value of the Arrhenius parameters. Therefore, in the computational model, the outputs of interest are the average value of the fuel concentration along these lines.

The computational domain has an overall length of $18\,mm$ and height of $9\,mm$. The fixed parameters in the Arrhenius law (2.2) are $c = 0.2$ and $d = 0.24$. The diffusivity has a value of $5 \times 10^{-6}\ m^2/s$. A uniform convective velocity field given by $\boldsymbol{U} = (0.2,\, 0)\, m/s$ is assumed inside the combustion chamber. It is worth noting that this simplistic flow velocity field does not limit the generality of the reduced basis method proposed in this thesis. A more realistic velocity field obtained from solving the Navier-Stokes equations could be used without modifying the methodology.

Finally, the source term $f$ in equation (2.1) is set to $f \equiv 0$ and the Dirichlet boundary conditions at the left vertical boundary of the domain are given by

$$u_D = \begin{cases} 0, & \text{if } 0 \le y < 3\,mm; \\ c, & \text{if } 3\,mm \le y < 6\,mm; \\ 0, & \text{if } 6\,mm \le y \le 9\,mm; \end{cases} \tag{2.48}$$

All other boundaries have homogeneous Neumann boundary conditions as described in Section 2.1.

The mesh for obtaining the "truth" solution was generated using a customized version of Distmesh [48]. The mesh contains 6696 triangular elements and is refined in the regions where the combustion fronts are expected to occur as shown in Figure 2-2. The largest elements in the mesh have a Péclet number of approximately 20, which clearly requires some sort of stabilization in order to prevent oscillations.

Figures 2-3, 2-4 and 2-5 show the solution $u$ computed at three corners of parameter space. Figure 2-3 corresponds to a combustion process with very low reaction rates so the resulting solution is dominated by convection-diffusion. Figure 2-4 corresponds to the highest reaction rates considered in this study. In this case, a sharp reaction front occurs and the resulting flame is confined to the region near the fuel injector. This is the situation that corresponds to a fast-burning fuel. Finally, Figure 2-5 represents an intermediate situation. From these figures it is obvious that the reaction parameters $A$ and $E$ have a very significant effect on the field variable $u$ so the problem shows a high dependence on the nonlinear Arrhenius term.

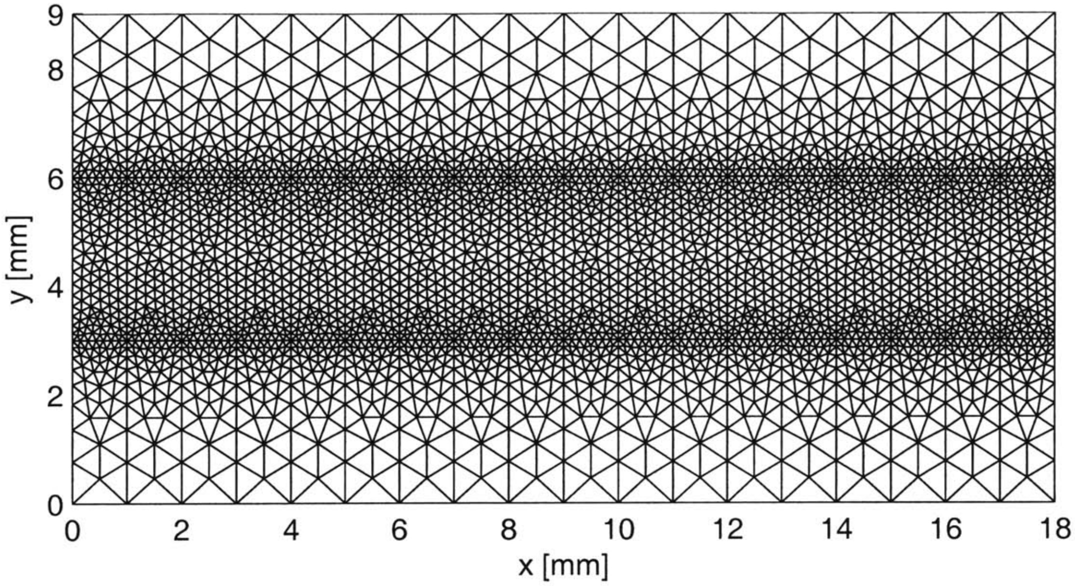

Figure 2-2: Mesh used for computing "truth" solution.

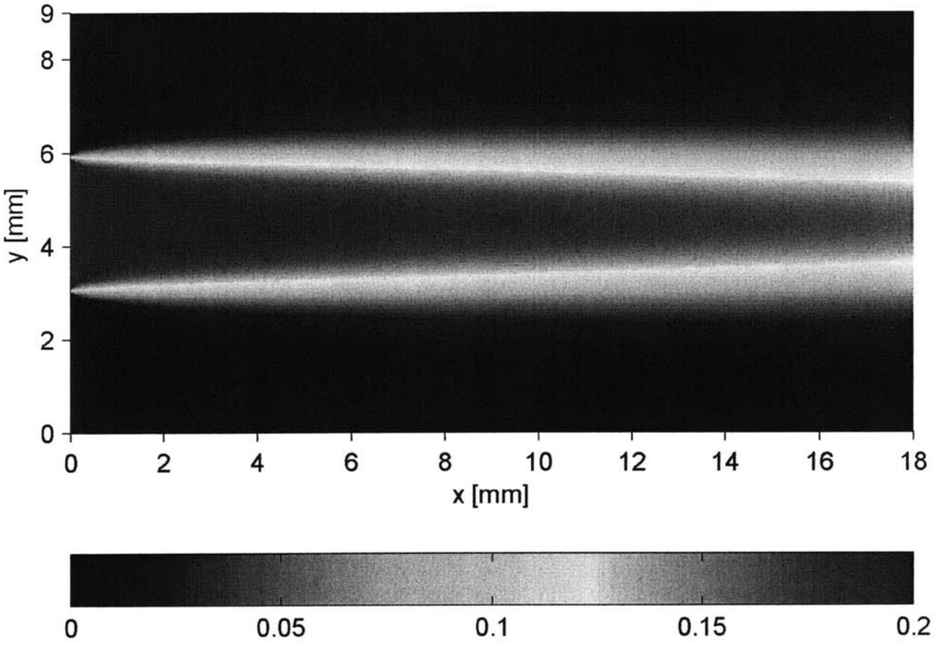

Figure 2-3: Molar fraction of fuel for $(A, E) = (e^5, 0.15)$.

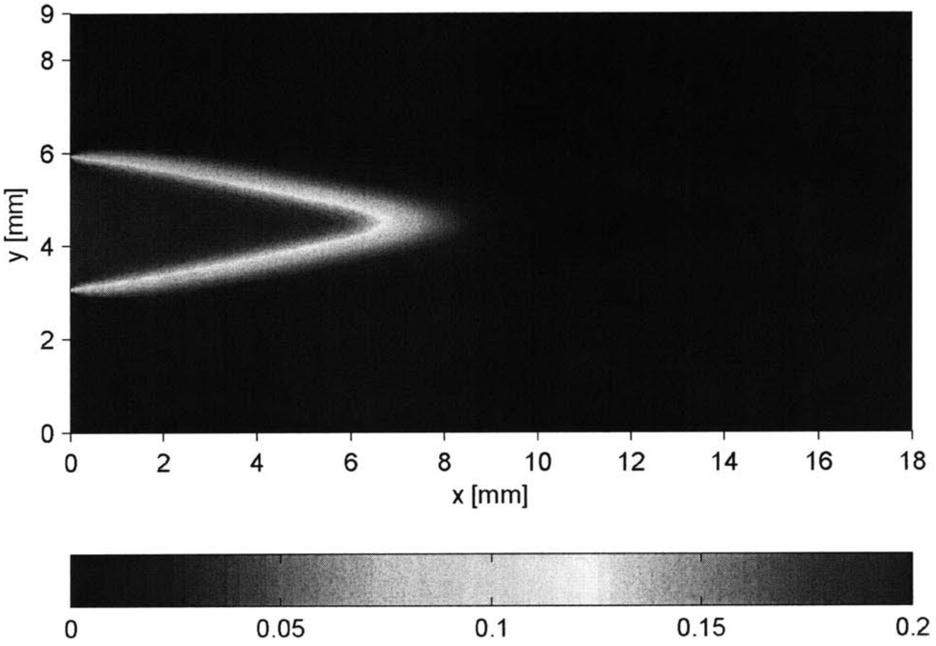

Figure 2-4: Molar fraction of fuel for $(A, E) = (e^{7.25}, 0.05)$.

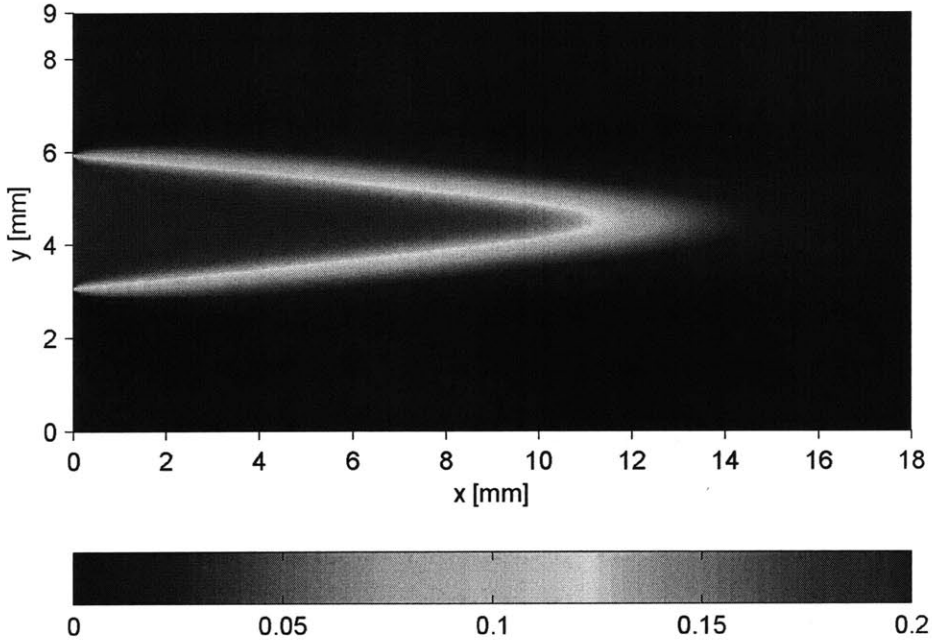

Figure 2-5: Molar fraction of fuel for $(A, E) = (e^{7.25}, 0.15)$.

# Chapter 3

# Reduced Basis Approximation

## 3.1 Introduction

In this chapter we develop the reduced basis approximation for the nonlinear convection-diffusion-reaction equation presented in Section 2.1. For a more detailed explanation of the reduced basis methodology and nomenclature used in this chapter, the reader is referred to the work by Patera and co-workers, which is presented in detail in [47].

We start the chapter by introducing the proper orthogonal decomposition (POD) procedure used to construct the low-dimensional approximation spaces required for building the reduced basis approximation. Then, we continue by developing a naive reduced basis approximation following the standard Galerkin reduced-order approach and explain why the resulting approximation cannot be considered a true reduced-order model since the operation count for evaluating the nonlinear term scales as a function of $\mathcal{N}$ (size of the FE model developed in Chapter 2). Next, we devote a section to developing the methodology for treating the inhomogeneous boundary conditions found in our problem of interest.

In Section 3.5, we present the critical building blocks that are required for constructing an efficient reduced basis approximation with an online operation count that is independent of $\mathcal{N}$, namely, the empirical interpolation method (EIM) and best points interpolation method (BPIM) used for developing the coefficient-function approximation of the nonlinear Arrhenius term.

Finally, using all the building blocks presented throughout the chapter, we develop an efficient offline-online algorithm for computing the outputs of interest given any input $\boldsymbol{\mu} \in \mathcal{D}$. The results obtained from the resulting reduced order model are shown at the end of the chapter.

## 3.2 Proper Orthogonal Decomposition

In this chapter we use the proper orthogonal decomposition (POD) procedure, presented by Karhunen [31] and Loève [32], for constructing a set of orthonormal basis functions $\{\zeta_n\}_{n=1}^N$ given a set of snapshots $\{\xi_k\}_{k=1}^K$. The set of basis functions gives the best representation of the set of snapshots in the sense that each basis function has the largest mean square projection on the set of snapshots. That is, for a given $N < K$, the POD procedure consists in finding $\zeta_n$, $1 \leq n \leq N$ such that

$$\zeta_n = \arg \max_{\substack{(\zeta_n, \zeta_{n'})_X = \delta_{nn'} \\ 1 \leq n, n' \leq N}} \sum_{n=1}^{N} \left( \frac{1}{K} \sum_{k=1}^{K} (\zeta_n, \xi_k)_X^2 \right), \quad 1 \leq n \leq N, \tag{3.1}$$

where $(\cdot,\cdot)_X$ denotes the inner product associated to $X$. It is worth noting that this inner product should induce a norm equivalent to the $H^1(\Omega)$ norm in order to be consistent with the exact infinite-dimensional formulation derived in Chapter 2. Since the bilinear form $a(w,v)$ defined in equation (2.6) is coercive (positive definite) but not symmetric we select the following definition of inner product

$$(w,v)_X \equiv a_S(w,v), \quad \forall w, v \in X \tag{3.2}$$

and hence, the induced norm is given by

$$\|w\|_X = \sqrt{a_S(w,w)}, \quad \forall w \in X, \tag{3.3}$$

where $a_S(w,v) = \frac{1}{2}\left(a(w,v) + a(v,w)\right)$ is the symmetric part of $a$. It is easy to prove that equations (3.2) and (3.2) define indeed a valid inner product and norm. The reason for selecting this inner product and norm instead of the maybe more common $L^2$-norm or its discrete $\ell^2$ version resides in the fact that, even though all norms are equivalent for a finite-dimensional space, the equivalence constants are dependent on the dimension of the space and not necessarily bounded as $\mathcal{N} \to \infty$. Therefore, a norm that is inconsistent with the infinite-dimensional formulation will likely result in ill-conditioned systems when the dimension of the space $X$ is very large. This being said, for the problem considered herein, one can replace the $(\cdot,\cdot)_X$ inner product and induced $X-$norm by the standard Euclidean inner product and norm without any noticeable impact on the numerical results.

Going back to the POD procedure, after expressing each basis function as a linear combination of snapshots,

$$\zeta_n(\boldsymbol{x}) = \sum_{k=1}^{K} a_k^n \xi_k(\boldsymbol{x}), \quad 1 \leq n \leq N, \tag{3.4}$$

it can be shown that problem (3.1) is equivalent to solving the following eigenvalue problem

$$\boldsymbol{C}\boldsymbol{a} = \lambda \boldsymbol{a}, \tag{3.5}$$

where $\boldsymbol{C} \in \mathbb{R}^{K \times K}$ is given by

$$C_{ij} = \frac{1}{K} \left(\xi_i, \xi_j\right)_X, \quad 1 \leq i, j \leq K, \tag{3.6}$$

The eigenproblem (3.5) is then solved for the first $N$ eigenvectors from which the POD basis functions $\zeta_n$, $1 \leq n \leq N$ are constructed using (3.4).

Since the POD basis functions are obtained from solving the maximization problem (3.1), they provide the most accurate representation of the snapshot set $\{\xi_k\}_{k=1}^K$ in the mean square error sense.

## 3.3 Standard Galerkin Reduced Basis Approximation

Let us recall the formulation of our problem of interest in the finite element approximation space that was introduced in Section 2.3: given any $\boldsymbol{\mu} \in \mathcal{D}$, evaluate

$$o_i(\boldsymbol{\mu}) = \ell_i^O(u(\boldsymbol{\mu})), \quad i = 1, \ldots, N_s, \tag{3.7}$$

where $u(\boldsymbol{\mu}) \in X^D$ is the solution of

$$\begin{aligned} &a(u(\boldsymbol{\mu}), v) + \int_\Omega s(u(\boldsymbol{\mu});\, \boldsymbol{\mu})\, v \, d\Omega \\ &+ \sum_{k=1}^{N_{\text{elem}}} \int_{T_h^k} (\boldsymbol{U} \cdot \nabla v)\ \tau_k\ [\boldsymbol{U} \cdot \nabla u(\boldsymbol{\mu}) + s(u(\boldsymbol{\mu});\, \boldsymbol{\mu}) - f]\; d\Omega = \ell(v), \quad \forall v \in X. \end{aligned} \tag{3.8}$$

where the bilinear and linear forms $a(\cdot,\cdot)$, $\ell(\cdot)$ and $\ell^O(\cdot)$ were defined in equations (2.6)–(2.8).

The objective of the reduced basis approximation is to obtain an approximation to the output of interest $\boldsymbol{o}_N(\boldsymbol{\mu})$ at a much cheaper computational cost. Towards this end we first introduce a sample set $S_K = \{\boldsymbol{\mu}_1 \in \mathcal{D}, \ldots, \boldsymbol{\mu}_K \in \mathcal{D}\}$ with an associated set of snapshots $\mathcal{S}_K^u \equiv \{\xi_k^u(\boldsymbol{x}) = u(\boldsymbol{\mu}_k),\ \boldsymbol{\mu}_k \in S_K,\ 1 \le k \le K\}$, where $u(\boldsymbol{\mu}_k)$ is the solution of (3.8) at $\boldsymbol{\mu} = \boldsymbol{\mu}_k$.

The solutions of the reduced order model reside in the reduced basis space $W_N^D$ given by $W_N^D = \text{span}\,\{\zeta_n,\ 1 \le n \le N\}$ where the functions $\zeta_n$, $1 \le n \le N$, are computed by applying the POD procedure upon the set of snapshots $\mathcal{S}_K^u$. The standard Galerkin reduced order model is then obtained by a standard Galerkin projection: given any $\boldsymbol{\mu} \in \mathcal{D}$, evaluate

$$o_{N\,i}^{\text{SG}}(\boldsymbol{\mu}) = \ell_i^O(u_N^{\text{SG}}(\boldsymbol{\mu})), \quad i = 1, \ldots, N_s, \tag{3.9}$$

where $u_N^{\mathrm{SG}}(\boldsymbol{\mu}) \in W_N^D$ is the solution of

$$\begin{aligned} a(u_N^{\mathrm{SG}}(\boldsymbol{\mu}), v) &+ \int_\Omega s(u_N^{\mathrm{SG}}(\boldsymbol{\mu});\, \boldsymbol{\mu})\, v\, d\Omega \\ &+ \sum_{k=1}^{N_{\mathrm{elem}}} \int_{T_h^k} (\boldsymbol{U} \cdot \nabla v)\ \tau_k\ \left[\boldsymbol{U} \cdot \nabla u_N^{\mathrm{SG}}(\boldsymbol{\mu}) + s(u_N^{\mathrm{SG}}(\boldsymbol{\mu});\, \boldsymbol{\mu}) - f\right]\, d\Omega = \ell(v), \\ &\qquad\qquad\qquad \forall v \in W_N, \end{aligned} \tag{3.10}$$

where

$$u_N^{\mathrm{SG}}(\boldsymbol{\mu}) = \sum_{j=1}^{N} u_{N\,i}^{\mathrm{SG}}(\boldsymbol{\mu})\, \zeta_i \tag{3.11}$$

Note that in the series expansion (3.11), the function $u_N^{\mathrm{SG}}(\boldsymbol{\mu})$ actually stands for $u_N^{\mathrm{SG}}(\boldsymbol{x}; \boldsymbol{\mu})$ whereas the unknown coefficients $u_{N\,i}^{\mathrm{SG}}(\boldsymbol{\mu})$, $1 \le i \le N$ only depend on the parameter $\boldsymbol{\mu}$. The dependence on the spatial variable $\boldsymbol{x}$ is introduced through the basis functions $\zeta_i(\boldsymbol{x})$, $1 \le i \le N$. Note also that postulating that the reduced order solution $u_N^{\mathrm{SG}}(\boldsymbol{\mu})$ is a linear combination of functions $\{\zeta_i\}_{i=1}^N$ that live in $W_N^D$ is a daring statement: in general, for problems with inhomogeneous boundary conditions ($u_D \neq 0$), the sum of two elements of $W_N^D$ is not an element of $W_N^D$, which means that the solution $u_N^{\mathrm{SG}}(\boldsymbol{\mu})$ as written in (3.11) is only guaranteed to satisfy the boundary conditions of the problem for the homogeneous case. We address this issue in Section 3.4. First we shall discuss some more immediate problems associated with the standard Galerkin formulation (3.10).

In the case of a linear, or weakly nonlinear, system it is possible to substitute (3.11) into (3.10) in order to implement an efficient offline-online decomposition such that the online stage requires the solution of an algebraic system of equations with an operation count that is only a function of $N$.

Unfortunately, since in our case $s$ is an exponential nonlinearity of $u$, it is not possible to develop an efficient offline-online computational procedure following the same methodology used for example in [25, 33, 44, 47, 49, 61, 62]. Using a standard Galerkin projection, the evaluation of the nonlinear terms $\int_\Omega s(u_N^{\mathrm{SG}}(\boldsymbol{\mu});\, \boldsymbol{\mu})\, v\, d\Omega$ and $\int_\Omega (\boldsymbol{U} \cdot \nabla v)\ \tau\, s(u_N^{\mathrm{SG}}(\boldsymbol{\mu});\, \boldsymbol{\mu})\, d\Omega$ will always scale as some power of $\mathcal{N}$. Therefore, the reduction in computational cost achieved with the standard Galerkin projection on highly nonlinear problems such as (3.8) is generally very modest and considering (3.10) a reduced order model is somewhat optimistic.

A true reduced order model should have an online evaluation cost that depends only on the dimension of the reduced-basis approximation spaces and *not* on $\mathcal{N}$. As a side comment, it is worth noting that, in general, the online operation count of a reduced order model also depends on the parametric complexity of the problem, that is, the number of operations that are required to compute the $\boldsymbol{\mu}$-dependent coefficients in the reduced order model. However, for typical problems like the one considered in this thesis, these coefficients can be computed by evaluating simple algebraic expressions so the computational cost is negligible compared to the cost of solving the model and will not be taken into consideration when discussing operation counts.

In Section 3.6 we derive an efficient reduced basis approximation that overcomes the limitations of the standard Galerkin projection proposed in this section. However, first we need to address the issue of dealing with inhomogeneous Dirichlet boundary conditions in a rigorous and efficient manner.

## 3.4 Treatment of Inhomogeneous Dirichlet Boundary Conditions

As briefly discussed in the previous section, the fact that our problem of interest has a portion of the inlet boundary with inhomogeneous boundary conditions requires some attention. In this thesis we follow the methodology used in [50] for treating inhomogeneous boundary conditions in the POD context.

Recall that in the finite element formulation we used two different interpolation spaces: the first one, denoted by $X$, is composed of test functions and consists of all piecewise linear functions over the computational domain $\Omega$, that vanish on the

Dirichlet portion, $\Gamma_{\text{in}}$ of the boundary. This space was defined as follows:

$$X = \{\, v \in H^1(\Omega) \;\mid\; v|_{T_h} \in \mathbb{P}_1(T_h),\, \forall T_h \in \mathcal{T}_h \text{ and } v = 0 \text{ on } \Gamma_{\text{in}} \,\} \tag{3.12}$$

The second space, $X^D$ contains the trial solutions, which are also piecewise linear functions over $\Omega$ but, as opposed to the test functions, they are required to satisfy the actual Dirichlet conditions on $\Gamma_{\text{in}}$. Mathematically, this space is defined as

$$X^D = \{\, u \in H^1(\Omega) \;\mid\; u|_{T_h} \in \mathbb{P}_1(T_h),\, \forall T_h \in \mathcal{T}_h \text{ and } u = u_D \text{ on } \Gamma_{\text{in}} \,\} \tag{3.13}$$

We note that $X^D \equiv X + \bar{u}_D$ where $\bar{u}_D$ is any function in $H^1(\Omega)$ such that $\bar{u}_D|_{T_h} \in \mathbb{P}_1(T_h)$, $\forall T_h \in \mathcal{T}_h$ *and*, $\bar{u}_D = u_D$ on $\Gamma_{\text{in}}$. Clearly, since $X^D$ is actually a translation of $X$, it is an affine space. These two spaces only coincide for problems with homogeneous boundary conditions.

In the reduced basis context we define the reduced basis approximation space as $W_N^D = \text{span}\,\{\zeta_n,\ 1 \le n \le N\}$. Since this space is constructed using solutions of the full-order problem for specific values of the parameters, it is clear that $W_N^D$ is an $N$-dimensional subspace of $X_N^D$. In general, it is not true that any linear combination of basis functions of $W_N^D$ is also a member of $W_N^D$. This is only true if the coefficients satisfy an additional constraint. For example, if we use snapshots of the solution directly as basis functions, $\zeta_n = u(\boldsymbol{\mu}_k)$, $k = 1, \ldots, N$ then a general function $w_N$ defined as $w_N = \sum_{n=1}^{N} w_{N\,n}\, \zeta_n$ would be a member of $W_N^D$ *if and only if* $\sum_{n=1}^{N} w_{N\,n} = 1$. Otherwise, $w_N \notin W_N^D$. For basis functions computed by applying Gram-Schmidt orthonormalization or POD to a set of snapshots it is straightforward to derive an equivalent condition that the coefficients $w_{N\,n}$ must satisfy such that $w_N \in W_N^D$.

The condition that the truth solution of the discrete full-order problem must belong to $X^D$ is not imposed in the weak formulation. Instead, it is imposed strongly while solving the system of equations (2.35). In order to impose that the reduced basis approximation to our nonlinear problem $u_N(\boldsymbol{\mu})$ must reside in $W_N^D$ following an analogous strong formulation we would need to impose an additional equation on the sum of the coefficients $u_{N\,n}(\boldsymbol{\mu})$ as discussed previously. Adding an additional equation to an already determined system poses some obvious difficulties.

In this thesis we solve the problem by taking advantage of the fact that there exists a subspace of $X$, denoted by $W_N$, such that $W_N^D$ is a translation of $W_N$; that is, $W_N^D \equiv W_N + \bar{u}_D$, where $\bar{u}_D$ is the same function that was already used for defining the affine space $X^D$. We now define our basis functions as members of $W_N$ to exploit the fact that any linear combination of functions that reside in $W_N$ also belongs to $W_N$. The process for building this approximation space is as follows: we first introduce a sample set $S_K = \{\boldsymbol{\mu}_1 \in \mathcal{D}, \ldots, \boldsymbol{\mu}_K \in \mathcal{D}\}$ with an associated set of snapshots $\mathcal{S}_K^u \equiv \{\xi_k^u(\boldsymbol{x}) = u(\boldsymbol{\mu}_k) - \bar{u},\ \boldsymbol{\mu}_k \in S_K,\ 1 \le k \le K\}$, where $u(\boldsymbol{\mu}_k)$ is the solution of (3.8) at $\boldsymbol{\mu} = \boldsymbol{\mu}_k$ and $\bar{u}$ is given by

$$\bar{u} = \frac{1}{K} \sum_{k=1}^{K} u(\boldsymbol{\mu}_k) \tag{3.14}$$

Note that we could use any other definition of $\bar{u}$ with the only condition that $\bar{u}$ must be a member of $W_N^D$. The reduced basis approximation is then given by

$$u_N^{\mathrm{SG}}(\boldsymbol{\mu}) = \bar{u} + \sum_{j=1}^{N} u_{N\,i}^{\mathrm{SG}}(\boldsymbol{\mu})\, \zeta_i, \tag{3.15}$$

which is guaranteed to live in $W_N^D$ even though the basis functions $\zeta_n$, $1 \le n \le N$ are now members of the space $W_N$ defined as $W_N = \operatorname{span}\{\zeta_n,\ 1 \le n \le N\}$. Using expression (3.15) for performing the Galerkin projection guarantees that the inhomogeneous boundary conditions are satisfied while eliminating the need for modifying the discretized reduced order equations prior to solving them.

## 3.5 Coefficient-Function Approximation

In Section 3.3 we showed that a reduced basis approximation built using the standard Galerkin reduced order approach is inefficient due to the $\mathcal{N}$-dependent operation count required for evaluating the nonlinear term during the online RB stage. In this section we present two alternative methods for overcoming this limitation by developing a coefficient-function approximation of the Arrhenius term prior to performing the Galerkin projection on the reduced basis space.

### 3.5.1 Empirical Interpolation Method

The empirical interpolation method (EIM) was first introduced in [3] and applied to the solution of nonaffine and nonlinear PDEs in [24]. Given a $\boldsymbol{\mu}$-parametrized function $s(\boldsymbol{x}; \boldsymbol{\mu}) \in L^{\infty}(\Omega) \cap C^{0}(\Omega)$, $\forall \boldsymbol{\mu} \in \mathcal{D}$ the EIM consists of a "greedy" selection process for constructing an approximation space and a set of interpolation points that can be used for building a coefficient-function expansion $s_M(\boldsymbol{x}; \boldsymbol{\mu})$ that approximates the original function for the entire domain of interest in parameter space, $\mathcal{D}$. Here, $\boldsymbol{x} = (x_1, x_2)$ is a point of the physical domain $\Omega \in \mathbb{R}^2$, and $\mathcal{D} \in \mathbb{R}^P$ is the parameter space in which our $P$-tuple parameter vector $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_P)$ resides (for our specific application, $P = 2$).

In this thesis, the approximation space is spanned by $M$ orthonormal basis functions, $\Phi_M = \operatorname{span}\{\phi_1, \ldots, \phi_M\}$, with $(\phi_i, \phi_j) = \delta_{ij}$, $1 \leq i, j \leq M$; which are obtained by applying the POD procedure described in Section 3.2 to a set of snapshots.

First, we introduce a sample in parameter space $S_K = \{\boldsymbol{\mu}_1 \in \mathcal{D}, \ldots, \boldsymbol{\mu}_K \in \mathcal{D}\}$ and an associated set of snapshots $\mathcal{S}_K \equiv \{\xi_k = s(\boldsymbol{x}, \boldsymbol{\mu}_k), 1 \leq k \leq K\}$. We then apply the POD procedure to the snapshot set $\mathcal{S}_K$ to construct $M$ basis functions $\{\phi_m\}_{m=1}^{M}$ and define an associated approximation space $\Phi_M = \operatorname{span}\{\phi_1, \ldots, \phi_M\}$.

We point out that in [3] and [24], the approximation space is built by applying a greedy algorithm directly on the set of snapshots $\mathcal{S}_K$ and then orthonormalizing the selected snapshots using the Gram-Schmidt algorithm. Here we use POD instead of a greedy algorithm for constructing the approximation space.

After obtaining the basis set $\{\phi_m\}_{m=1}^{M}$, the empirical interpolation points $\{\boldsymbol{z}_m \in \Omega\}_{m=1}^{M}$ associated with this set are computed as follows:

```
$\boldsymbol{z}_1 = \arg \ \text{ess} \sup_{\boldsymbol{x} \in \Omega} |\phi_1(\boldsymbol{x})|$
for $L = 2, \ldots, M$
    solve $\sum_{j=1}^{L-1} \phi_j(\boldsymbol{z}_i)\sigma_j^{L-1} = \phi_L(\boldsymbol{z}_i), \quad 1 \leq i \leq L-1$
    $r_L(\boldsymbol{x}) = \phi_L(\boldsymbol{x}) - \sum_{j=1}^{L-1} \sigma_j^{L-1} \phi_j(\boldsymbol{x})$
    $\boldsymbol{z}_L = \arg \ \text{ess} \sup_{\boldsymbol{x} \in \Omega} |r_L(\boldsymbol{x})|$
end
```

Given the sets of orthonormal basis functions $\{\phi_m\}_{m=1}^M$ and interpolation points $\{\boldsymbol{z}_m\}_{m=1}^M$ we define the coefficient-function approximation as

$$s_M(\boldsymbol{x}; \boldsymbol{\mu}) = \sum_{m=1}^{M} s(\boldsymbol{z}_m; \boldsymbol{\mu}) \psi_m(\boldsymbol{x}), \tag{3.16}$$

where the cardinal functions $\{\psi_m\}_{m=1}^M$ are defined by $\psi_j(\boldsymbol{z}_i) = \delta_{ij}$, and hence given by

$$\phi_i(\boldsymbol{x}) = \sum_{j=1}^{M} \phi_i(\boldsymbol{z}_j) \psi_j(\boldsymbol{x}), \quad 1 \leq i \leq M. \tag{3.17}$$

The sample in parameter space $S_K$ should obviously cover all the domain of interest $\mathcal{D}$ and be suitably fine for this approximation to provide accurate results. For a detailed derivation of the methodology including error estimates we refer the reader to [3, 24].

### 3.5.2 Best Points Interpolation Method

The "best points" interpolation method (BPIM) was introduced in [42] as an efficient method for approximating parametrized functions. The BPIM replaces the greedy algorithm used in the EIM by an optimization problem which provides higher accuracy at the cost of greater computational complexity. Its application to the reduced basis approximation of nonaffine and nonlinear PDEs was described in [43]. Here, we provide a brief description of the method; for a more detailed explanation of the methodology we refer the reader to [42].

The starting point for the BPIM is identical to that used for the EIM, namely, given an approximation space spanned by $M$ orthonormal basis functions, $\Phi_M = \operatorname{span}\{\phi_1, \ldots, \phi_M\}$ (in our case this space is constructed by applying POD upon a suitably fine set of snapshots), we seek a set of interpolation points $\{\boldsymbol{z}_m\}_{m=1}^M$ such that we can obtain an approximation to the original parametrized function as a linear combination of pre-computed basis functions $\psi_m(\boldsymbol{x})$, $1 \leq i \leq M$ with coefficients that are given by the values of the original function at the interpolation points,

$$s_M(\boldsymbol{x}; \boldsymbol{\mu}) = \sum_{m=1}^{M} s(\boldsymbol{z}_m; \boldsymbol{\mu}) \psi_m(\boldsymbol{x}), \tag{3.18}$$

where the basis functions $\{\psi_m\}_{m=1}^M$ are defined by (3.17).

In order to find the set of best interpolation points we introduce the best approximation to the elements in the snapshot set $\mathcal{S}_K$ that resides in our approximation space $\Phi_M$ as

$$s_M^*(\,\cdot\,; \boldsymbol{\mu}_k) = \arg \min_{w_M \in \Phi_M} \|s(\,\cdot\,; \boldsymbol{\mu}_k) - w_M\|, \quad 1 \leq k \leq K. \tag{3.19}$$

Then, we postulate that the interpolation points $\{\boldsymbol{z}_m\}_{m=1}^M$ are optimal if they minimize the sum of the squared errors between the best approximation $s_M^*(\,\cdot\,; \boldsymbol{\mu}_k)$ and the coefficient-function approximation $s_M(\,\cdot\,; \boldsymbol{\mu})$. That is, the best interpolation points are the minimizer of the following problem,

$$\min_{\boldsymbol{z}_1 \in \Omega, \ldots, \boldsymbol{z}_M \in \Omega,} \sum_{k=1}^{K} \|s_M^*(\,\cdot\,; \boldsymbol{\mu}_k) - s_M(\,\cdot\,; \boldsymbol{\mu}_k)\|^2 . \tag{3.20}$$

From the orthonormality of the basis functions $\phi_m(\boldsymbol{x})$, $1 \leq i \leq M$ and from the definition of the coefficient-function approximation, it is clear that the best approximation and the coefficient-function approximation in (3.20) can be expressed as

$$s^*_M(\,\cdot\,;\boldsymbol{\mu}_k) = \sum_{m=1}^{M} \alpha_{M\,m}(\boldsymbol{\mu}_k)\phi_m(\boldsymbol{x}), \quad 1 \le k \le K, \tag{3.21}$$

$$s_M(\,\cdot\,;\boldsymbol{\mu}_k) = \sum_{m=1}^{M} \beta_{M\,m}(\boldsymbol{\mu}_k)\phi_m(\boldsymbol{x}), \quad 1 \le k \le K, \tag{3.22}$$

where the coefficients of the series expansion for the best approximation are given by

$$\alpha_{M\,m}(\boldsymbol{\mu}_k) = (\phi_m, s(\,\cdot\,;\boldsymbol{\mu}_k))\,, \quad 1 \le m \le M,\ 1 \le k \le K, \tag{3.23}$$

and, by definition of the coefficient-function approximation, the coefficients of its series expansion are computed as the solution of

$$\sum_{m=1}^{M} \phi_m(\boldsymbol{z}_i)\beta_{M\,m}(\boldsymbol{\mu}_k) = s(\boldsymbol{z}_i;\boldsymbol{\mu}_k), \quad 1 \le i \le M,\ 1 \le k \le K, \tag{3.24}$$

We now substitute equations (3.21) through (3.24) into (3.20) and invoke orthonormality of $\{\phi_m\}_{m=1}^M$ to obtain the following least-squares minimization problem

$$\min_{\boldsymbol{z}_1\in\Omega,\ldots,\boldsymbol{z}_M\in\Omega} \sum_{k=1}^{K}\sum_{m=1}^{M} \left(\alpha_{M\,m}(\boldsymbol{\mu}_k) - \beta_{M\,m}(\boldsymbol{z}_1,\ldots,\boldsymbol{z}_M;\boldsymbol{\mu}_k)\right)^2 \tag{3.25}$$

$$\sum_{n=1}^{M} \phi_n(\boldsymbol{z}_m)\beta_{M\,n}(\boldsymbol{z}_1,\ldots\boldsymbol{z}_M;\boldsymbol{\mu}_k) = s(\boldsymbol{z}_m;\boldsymbol{\mu}_k), \quad 1 \le m \le M,\ 1 \le k \le K.$$

The solution to problem (3.25) is computed using the Levenberg-Marquardt algorithm [34]. Let $\boldsymbol{t} = (\boldsymbol{z}_1, \ldots, \boldsymbol{z}_M)$, we write the objective in (3.25) as

$$F(\boldsymbol{t}) = \frac{1}{2}\sum_{q=1}^{Q} r_q^2(\boldsymbol{t}), \tag{3.26}$$

where $r_q(\boldsymbol{t})$, $1 \le q \le Q = KM$, are given by

$$r_q(\boldsymbol{t}) = \alpha_{M\,m}(\boldsymbol{\mu}_k) - \beta_{M\,m}(\boldsymbol{t};\boldsymbol{\mu}_k), \quad 1 \le m \le M,\ 1 \le k \le K. \tag{3.27}$$

The gradient of the objective function can therefore be computed as

$$\nabla F(\boldsymbol{t}) = \sum_{q=1}^{Q} r_q(\boldsymbol{t}) \nabla r_q(\boldsymbol{t}) = (\boldsymbol{J}(\boldsymbol{t}))^T \boldsymbol{r}(\boldsymbol{t}) \tag{3.28}$$

where the Jacobian matrix $\boldsymbol{J}(\boldsymbol{t})$ is computed by differentiating both sides of the constraint in the minimization problem (3.25) with respect to $\boldsymbol{t}$ and $\boldsymbol{r}(\boldsymbol{t}) \in \mathbb{R}^Q$ is given by (3.27).

The Hessian can be approximated in terms of the Jacobian matrix $\boldsymbol{J}(\boldsymbol{t})$ as

$$\nabla^2 F(\boldsymbol{t}) \approx (\boldsymbol{J}(\boldsymbol{t}))^T \boldsymbol{J}(\boldsymbol{t}) \tag{3.29}$$

Finally, it is necessary to mention that, as with any optimization algorithm, an initial guess is required in order to run the Levenberg-Marquardt (LM) solver. In this case we use the empirical interpolation points as initial guess for computing the best points. Since running the optimization algorithm is relatively inexpensive, another option would be to generate several sets of random initial guesses and keep the best points that result from the initial guess with minimum value of the objective function after running the LM algorithm.

### 3.5.3 Numerical Examples

In this section we use the EIM and BPIM described in Sections 3.5.1 and 3.5.2 to interpolate the nonlinear reaction term in the convection-diffusion-reaction equation.

First, we define $\mu_1^{\min} \left(\equiv \ln A^{\min}\right) = 5$, $\mu_1^{\max} \left(\equiv \ln A^{\max}\right) = 7.25$, $\mu_2^{\min} \left(\equiv E^{\min}\right) = 0.05$, $\mu_2^{\max} \left(\equiv E^{\max}\right) = 0.15$, and then $\mathcal{I}^1 \equiv \left[\mu_1^{\min}, \mu_1^{\max}\right] \left(= \mathcal{I}^A \equiv [\ln A^{\min}, \ln A^{\max}]\right)$ and $\mathcal{I}^2 \equiv \left[\mu_2^{\min}, \mu_2^{\max}\right] \left(= \mathcal{I}^E \equiv [E^{\min}, E^{\max}]\right)$ such that we can express our parameter domain as $\mathcal{D} \equiv \mathcal{I}^1 \times \mathcal{I}^2$.

We then introduce the set of $L$ equi-spaced points between $p^{\min}$ and $p^{\max}$,

$$G_L[p^{\min}, p^{\max}] = \{p_L^1, p_L^2, \ldots, p_L^L\} \tag{3.30}$$
$$p_L^j = p^{\min} + \frac{j-1}{L-1}\left(p^{\max} - p^{\min}\right), \quad 1 \le j \le L.$$

These points are used to define a tensor product grid $\Xi_K$ over $\mathcal{D}$, $\Xi_K \subset \mathcal{D} \subset \mathbb{R}^2$ as

$$\Xi_K = G_L[\mu_1^{\min}, \mu_1^{\max}] \times G_L[\mu_2^{\min}, \mu_2^{\max}], \tag{3.31}$$

where $K = L^2$. For our numerical example we will consider $L = 14$ to generate a regular grid $\Xi_{196}$ over $\mathcal{D}$. We then choose the sample set $S_K = \{\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K\}$ as the nodes of this grid and we generate our sample of solution snapshots,

$$\mathcal{S}_K^u \equiv \{\xi_k^u(\boldsymbol{x}) = u(\boldsymbol{\mu}_k), \quad \boldsymbol{\mu}_k \in S_K,\ 1 \le k \le K\} \tag{3.32}$$

by solving the convection-diffusion-reaction PDE (2.1) at every point on the grid. The set of snapshots of the nonlinear Arrhenius term is then computed as

$$\mathcal{S}_K^s \equiv \{\xi_k^s(\boldsymbol{x}) = s(u(\boldsymbol{x}; \boldsymbol{\mu}_k); \boldsymbol{\mu}_k), \quad \boldsymbol{\mu}_k \in S_K,\ 1 \le k \le K\} \tag{3.33}$$

We then use the POD procedure outlined in Section 3.2 to compute the basis set $\{\phi_m\}_{m=1}^M$. Figure 3-1 shows the location of the best points and the EIM points for $M = 15$. We observe that the interpolation points are distributed around the regions where the reaction fronts occur in the solution (regions with highest reaction rates).

Figures 3-2 and 3-3 show the shape of the nonlinear Arrhenius term at the four corners of the parameter space $\mathcal{D}$. The EIM points ($\circ$) and BPIM points ($\diamond$) for $M = 10$ are overlaid on top of the reaction fields. Note that $\left\{\boldsymbol{z}_m^{\text{EIM}}\right\}_{m=1}^{10} \subset \left\{\boldsymbol{z}_m^{\text{EIM}}\right\}_{m=1}^{15}$ whereas $\left\{\boldsymbol{z}_m^{\text{BPIM}}\right\}_{m=1}^{10} \not\subset \left\{\boldsymbol{z}_m^{\text{BPIM}}\right\}_{m=1}^{15}$. These figures clearly show that, as expected, the interpolation points are located in the regions where the reaction occurs. It is worth noting that the nonlinear reaction field changes not only its shape but also its magnitude throughout the parameter space. To illustrate this, Figure 3-2 uses the

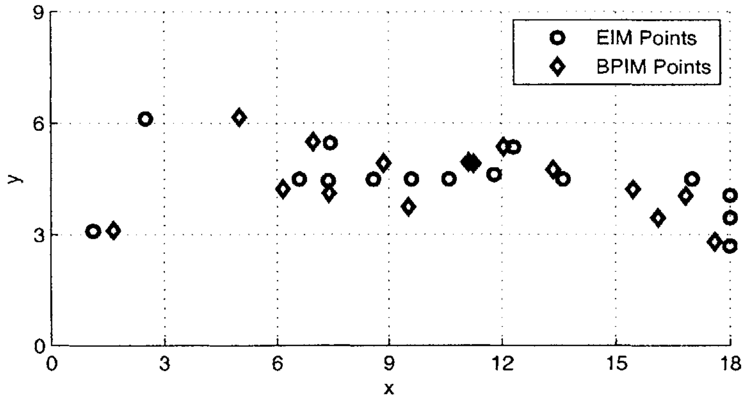

Figure 3-1: Distribution of the empirical interpolation points and best points on the physical domain for $M = 15$.

same color scale to plot the reaction fields at all four corners of parameter space. Note that the reaction term at point $\boldsymbol{\mu} = (5.0, 0.15)$ is hardly visible when plotted using the same scale used for plotting the reaction field at $\boldsymbol{\mu} = (7.25, 0.05)$. Figure 3-3 plots all four fields using different color scales for each of them in order to better show the changes in shape.

In order to test the performance of the interpolation methods we introduce a $23 \times 23$ grid, $\Xi_{529}^{\text{test}}$, which is finer and mostly noncoincident with the grid $\Xi_{196}$ used to compute the approximation space $\Phi_M = \text{span}\{\phi_1, \ldots, \phi_M\}$, EIM points $\left\{\boldsymbol{z}_m^{\text{EIM}}\right\}_{m=1}^M$ and BPIM points $\left\{\boldsymbol{z}_m^{\text{BP}}\right\}_{m=1}^M$. We then define the maximum relative error, $\varepsilon_{M,\max,\text{rel}}$ as

$$\varepsilon_{M,\max,\text{rel}} = \max_{\boldsymbol{\mu}\in\Xi_{529}^{\text{test}}} \frac{\varepsilon_M(\boldsymbol{\mu})}{\|s(\cdot,\boldsymbol{\mu})\|}, \tag{3.34}$$

where $\varepsilon_M(\boldsymbol{\mu}) = \|s(\cdot,\boldsymbol{\mu}) - s_M(\cdot,\boldsymbol{\mu})\|$. We wish to emphasize that $\|\cdot\|$ represents the continuous $L^2$-norm as opposed to the discrete $\ell^2$-norm or vector norm. Therefore, the error $\varepsilon_M$ is given by,

$$\varepsilon_M(\boldsymbol{\mu}) = \left(\int_\Omega (s(u(\boldsymbol{x};\boldsymbol{\mu});\boldsymbol{\mu}) - s_M(u(\boldsymbol{x};\boldsymbol{\mu});\boldsymbol{\mu}))^2\, d\Omega\right)^{\frac{1}{2}} \tag{3.35}$$

For the results shown herein, the integrals used to compute the error norms have

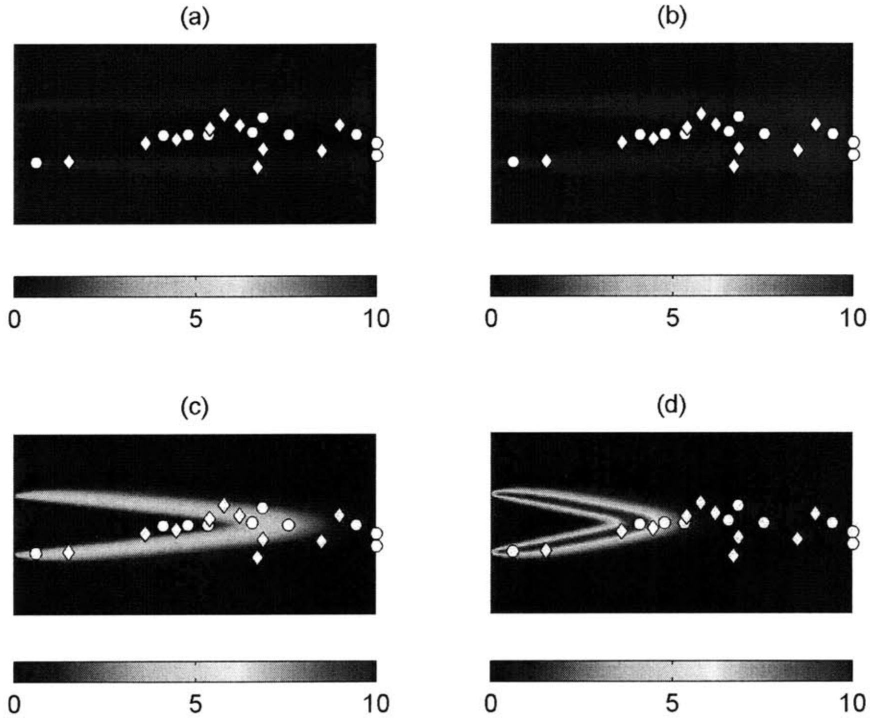

Figure 3-2: Nonlinear reaction field at the four corners of the grid $\Xi_{196}$: (a) $\boldsymbol{\mu}_{14} = (5.0, 0.15)$, (b) $\boldsymbol{\mu}_1 = (5.0, 0.05)$, (c) $\boldsymbol{\mu}_{196} = (7.25, 0.15)$ and (d) $\boldsymbol{\mu}_{183} = (7.25, 0.05)$. Same color scale is used in all cases to illustrate the change in magnitude of the reaction term. EIM points ($\circ$) and BPIM points ($\diamond$) for $M = 10$ are overlaid on top of the reaction fields.

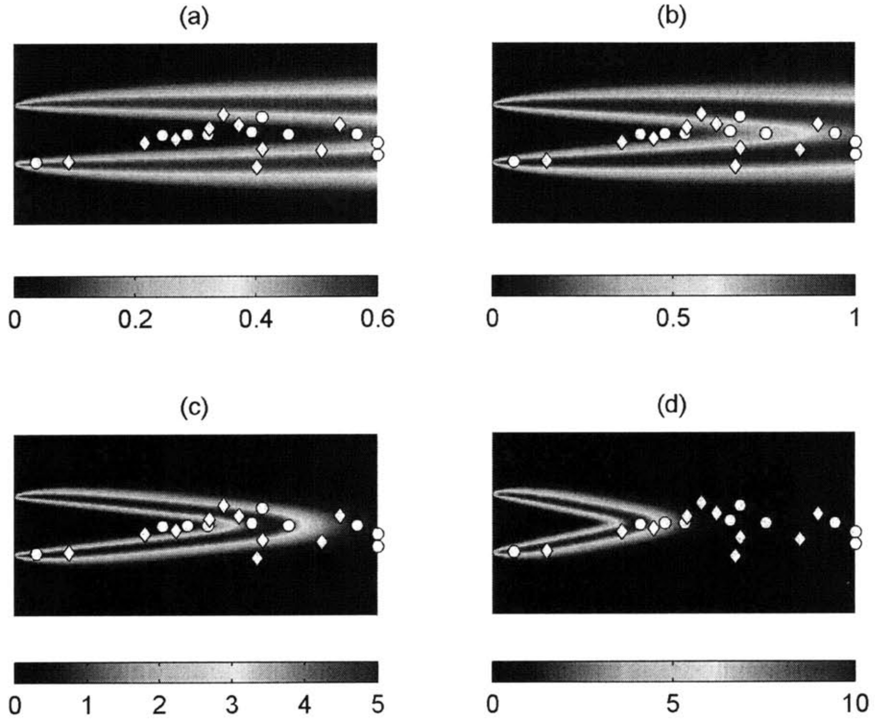

Figure 3-3: Nonlinear reaction field at the four corners of the grid $\Xi_{196}$: (a) $\boldsymbol{\mu}_{14} = (5.0, 0.15)$, (b) $\boldsymbol{\mu}_1 = (5.0, 0.05)$, (c) $\boldsymbol{\mu}_{196} = (7.25, 0.15)$ and (d) $\boldsymbol{\mu}_{183} = (7.25, 0.05)$. Each reaction field is plotted using a different color scale to illustrate the change in shape. EIM points ($\circ$) and BPIM points ($\diamond$) for $M = 10$ are overlaid on top of the reaction fields.

been computed using 4-point quadrature rules as shown below

$$\varepsilon_M(\boldsymbol{\mu}) = \left( \sum_{k=1}^{N_{\mathrm{elem}}} \sum_{q=1}^{N_Q} w_q J_{CT}^k \left(\xi_1^q, \xi_2^q\right) \left(s(u(\xi_1^q, \xi_2^q; \boldsymbol{\mu}); \boldsymbol{\mu}) - s_M(u(\xi_1^q, \xi_2^q; \boldsymbol{\mu}); \boldsymbol{\mu})\right)^2 \right)^{\frac{1}{2}} \tag{3.36}$$

where $(\xi_1^q, \xi_2^q)$ are the coordinates of quadrature point $q$ inside the master element in the computational domain, $J_{CT}^k(\xi_1^q, \xi_2^q)$ is the determinant of the Jacobian of the coordinate transformation between element $k$ in physical space and the master element, $w_q$ gives the weight associated with quadrature point $q$ and $N_Q$ is the total number of quadrature points in the master element (4 in our specific application).

The maximum relative errors for the “best points” and “empirical” interpolation methods as a function of the number of interpolation points are shown in Figure 3-4. As expected, the maximum relative error decreases monotonically with $M$ and the error obtained using the BPIM is always less than the error of the EIM approximation. For some values of $M$, the BPIM can achieve the same accuracy as the EIM with up to 6 fewer basis functions, which leads to a more economical reduced-order model. Since the reduced order model is typically several orders of magnitude cheaper to evaluate than the full-order model, a reduction of a few basis functions in the reduced-order model may not seem significant. However, for time-critical applications where real time input-output evaluations are required, the BPIM offers an important advantage over the EIM.

Finally, it is worth noting that there are some values of $M$ for which the improvement of the BPIM with respect to the EIM is not very significant. For these points, the EIM gave an initial solution that was close to a local optimum so the Levenberg-Marquardt algorithm used in the BPIM could not decrease the value of the objective function significantly. It is possible that the BPIM could have obtained better interpolation points by using a different initial guess instead of the EIM points.

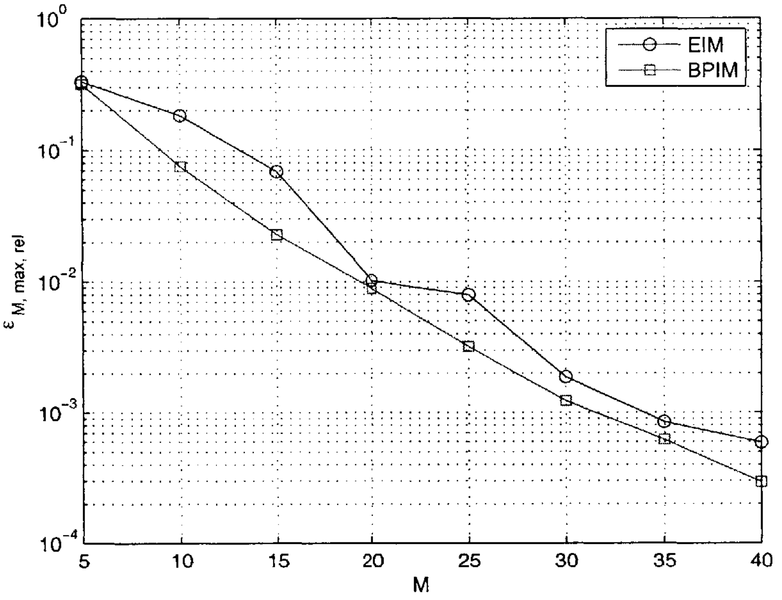

Figure 3-4: Maximum relative error norm $\varepsilon_{M,\max,\mathrm{rel}}$ as a function of $M$ for the EIM and BPIM.

## 3.6 Efficient Reduced Basis Approximation

### 3.6.1 Galerkin Projection

As discussed in Section 3.3, a true reduced order model should have an online evaluation cost that is independent of $\mathcal{N}$. In order to construct such a model we use the already existing set of snapshots of the solution $\mathcal{S}_K^u$ to compute a set of snapshots of the nonlinear reaction term,

$$\mathcal{S}_K^s \equiv \{\xi_k^s(\boldsymbol{x}) = s(u(\boldsymbol{x};\boldsymbol{\mu}_k);\boldsymbol{\mu}_k), \quad \boldsymbol{\mu}_k \in S_K,\ 1 \leq k \leq K\} \tag{3.37}$$

We now apply the POD procedure to compute an approximation space associated to this set of snapshots, $\Phi_M = \mathrm{span}\,\{\phi_1,\ldots,\phi_M\}$ and construct a set of interpolation basis functions $\{\psi_m\}_{m=1}^M$ and interpolation points $\{z_m\}_{m=1}^M$ using either the EIM or BPIM as outlined in Sections 3.5.1 and 3.5.2. Then, for any given function $w \in X$, we can approximate $s(w;\boldsymbol{\mu})$ by

$$s_M^w(\boldsymbol{x};\boldsymbol{\mu}) = \sum_{m=1}^{M} s(w(\boldsymbol{z}_m);\boldsymbol{\mu})\ \psi_m(\boldsymbol{x}) \tag{3.38}$$

In particular, we may now replace the nonlinear function $s\big(u_N^{\rm SG}(\boldsymbol{x};\boldsymbol{\mu});\boldsymbol{\mu}\big)$ in (3.10) with $s_M^{u_{N,M}}(\boldsymbol{x};\boldsymbol{\mu})$. Then, the reduced basis approximation becomes: given any $\boldsymbol{\mu} \in \mathcal{D}$, evaluate

$$o_{N,M\,i}(\boldsymbol{\mu}) = \ell_i^O(u_{N,M}(\boldsymbol{\mu})), \quad i = 1,\ldots,N_s, \tag{3.39}$$

where $u_{N,M}(\boldsymbol{\mu}) \in W_N^D$ is the solution of

$$\begin{aligned} a(\bar{u},v) + a(u_{N,M}(\boldsymbol{\mu}),v) + \int_\Omega s_M^{u_{N,M}}(\boldsymbol{x};\boldsymbol{\mu})\, v\, d\Omega \\ + \sum_{k=1}^{N_{\rm elem}} \int_{T_h^k} (\boldsymbol{U}\cdot\nabla v)\ \tau_k\ \big(\boldsymbol{U}\cdot\nabla u_{N,M}(\boldsymbol{\mu}) + \boldsymbol{U}\cdot\nabla\bar{u} + s_M^{u_{N,M}}(\boldsymbol{x};\boldsymbol{\mu}) - f\big)\ d\Omega \\ = \ell(v), \quad \forall v \in W_N. \end{aligned} \tag{3.40}$$

where the series expansion of the reduced basis approximation is given by

$$u_{N,M}(\boldsymbol{x};\boldsymbol{\mu}) = \bar{u}(\boldsymbol{x}) + \sum_{n=1}^{N} u_{N,M\,n}(\boldsymbol{\mu})\,\zeta_n(\boldsymbol{x}) \tag{3.41}$$

and the coefficient-function approximation is given by

$$\begin{aligned} s_M^{u_{N,M}}(\boldsymbol{x};\boldsymbol{\mu}) &= \sum_{m=1}^{M} s(u_{N,M}(\boldsymbol{z}_m;\boldsymbol{\mu});\boldsymbol{\mu})\ \psi_m(\boldsymbol{x}) \\ &= \sum_{m=1}^{M} s\left(\bar{u}(\boldsymbol{z}_m) + \sum_{n=1}^{N} u_{N,M\,n}(\boldsymbol{\mu})\,\zeta_n(\boldsymbol{z}_m);\boldsymbol{\mu}\right)\ \psi_m(\boldsymbol{x}) \end{aligned} \tag{3.42}$$

### 3.6.2 Reduced System of Algebraic Equations

Choosing $v = \zeta_n$, $1 \le n \le N$ and substituting (3.41) and (3.42) into (3.40) yields

$$\boldsymbol{A}_0 + \boldsymbol{A}_N\,\boldsymbol{u}_{N,M} + (\boldsymbol{E}_{N,M} + \boldsymbol{G}_{N,M})\ s(\bar{\boldsymbol{u}}_M + \boldsymbol{D}_{M,N}\,\boldsymbol{u}_{N,M};\ \boldsymbol{\mu}) = \boldsymbol{F}_N, \tag{3.43}$$

where $\boldsymbol{A}_0 \in \mathbb{R}^N$, $\boldsymbol{A}_N \in \mathbb{R}^{N\times N}$, $\boldsymbol{E}_{N,M} \in \mathbb{R}^{N\times M}$, $\boldsymbol{G}_{N,M} \in \mathbb{R}^{N\times M}$, $\bar{\boldsymbol{u}}_M \in \mathbb{R}^M$, $\boldsymbol{D}_{M,N} \in \mathbb{R}^{M\times N}$ and $\boldsymbol{F}_N \in \mathbb{R}^N$ are given below.

$$\begin{aligned}
A_{0\,n} &= a(\bar{u},\zeta_n) + \sum_{k=1}^{N_{\text{elem}}} \int_{T_h^k} (\boldsymbol{U}\cdot\nabla\zeta_n)\ \tau_k\ (\boldsymbol{U}\cdot\nabla\bar{u})\ d\Omega \\
&= a\left(\sum_{j=1}^{\mathcal{N}} \bar{u}_j\varphi_j, \sum_{i=1}^{\mathcal{N}} \zeta_i^n\varphi_i\right) + \sum_{k=1}^{N_{\text{elem}}} \int_{T_h^k} \left(\boldsymbol{U}\cdot\nabla\sum_{i=1}^{\mathcal{N}} \zeta_i^n\varphi_i\right)\ \tau_k\ \left(\boldsymbol{U}\cdot\nabla\sum_{j=1}^{\mathcal{N}} \bar{u}_j\varphi_j\right)\ d\Omega \\
&= \sum_{i=1}^{\mathcal{N}}\sum_{j=1}^{\mathcal{N}} \bar{u}_j\zeta_i^n a(\varphi_j,\varphi_i) + \sum_{i=1}^{\mathcal{N}}\sum_{j=1}^{\mathcal{N}} \bar{u}_j\zeta_i^n \sum_{k=1}^{N_{\text{elem}}} \int_{T_h^k} (\boldsymbol{U}\cdot\nabla\varphi_i)\ \tau_k\ (\boldsymbol{U}\cdot\nabla\varphi_j)\ d\Omega \\
&= (\boldsymbol{\zeta}^n)^T \boldsymbol{A}\,(\bar{\boldsymbol{u}}), \quad 1 \le n \le N,
\end{aligned} \tag{3.44}$$

Note that in the expression above, $\zeta_i^n$, $1 \le i \le \mathcal{N}$ refers to the nodal values of the basis function $\zeta_n(\boldsymbol{x})$, which can also be written in vector form as $\boldsymbol{\zeta}^n$. The same naming convention is used for $\bar{u}_j$, $1 \le j \le \mathcal{N}$. We now introduce the "basis" matrix $\boldsymbol{Z} \in \mathbb{R}^{\mathcal{N}\times N}$, which contains the $N$ basis vectors $\{\zeta_n\}_{n=1}^N$ stored in columns,

$$Z_{in} = \zeta_i^n, \quad 1 \le i \le \mathcal{N},\ 1 \le n \le N, \tag{3.45}$$

so that $\boldsymbol{A}_0$ can be written succinctly as

$$\boldsymbol{A}_0 = \boldsymbol{Z}^T \boldsymbol{A}\,\bar{\boldsymbol{u}}. \tag{3.46}$$

We now give the expressions for the other matrices in (3.43)

$$\begin{aligned}
A_{N\,nn'} &= a(\zeta_{n'},\zeta_n) + \sum_{k=1}^{N_{\text{elem}}} \int_{T_h^k} (\boldsymbol{U}\cdot\nabla\zeta_n)\ \tau_k\ (\boldsymbol{U}\cdot\nabla\zeta_{n'})\ d\Omega \\
&= a\left(\sum_{j=1}^{\mathcal{N}} \zeta_j^{n'}\varphi_j, \sum_{i=1}^{\mathcal{N}} \zeta_i^n\varphi_i\right) + \sum_{k=1}^{N_{\text{elem}}} \int_{T_h^k} \left(\boldsymbol{U}\cdot\nabla\sum_{i=1}^{\mathcal{N}} \zeta_i^n\varphi_i\right)\ \tau_k\ \left(\boldsymbol{U}\cdot\nabla\sum_{j=1}^{\mathcal{N}} \zeta_j^{n'}\varphi_j\right)\ d\Omega \\
&= \sum_{i=1}^{\mathcal{N}}\sum_{j=1}^{\mathcal{N}} \zeta_j^{n'}\zeta_i^n a(\varphi_j,\varphi_i) + \sum_{i=1}^{\mathcal{N}}\sum_{j=1}^{\mathcal{N}} \zeta_j^{n'}\zeta_i^n \sum_{k=1}^{N_{\text{elem}}} \int_{T_h^k} (\boldsymbol{U}\cdot\nabla\varphi_i)\ \tau_k\ (\boldsymbol{U}\cdot\nabla\varphi_j)\ d\Omega \\
&= (\boldsymbol{\zeta}^n)^T \boldsymbol{A}\left(\boldsymbol{\zeta}^{n'}\right), \quad 1 \le n, n' \le N,
\end{aligned} \tag{3.47}$$

which can also be written in terms of the basis matrix $\boldsymbol{Z}$ as

$$\boldsymbol{A}_N = \boldsymbol{Z}^T \boldsymbol{A}\, \boldsymbol{Z}. \tag{3.48}$$

The expression for $\boldsymbol{F}_N$ can be easily derived using the same process used for arriving at $\boldsymbol{A}_0$ and $\boldsymbol{A}_N$ so we will simply give its final form:

$$F_{N\,i} = \ell(\zeta_i) \quad 1 \leq i \leq N, \tag{3.49}$$

which can be written as

$$\boldsymbol{F}_N = \boldsymbol{Z}^T \boldsymbol{F}. \tag{3.50}$$

Lastly, we define the matrices that appear in the nonlinear term of the reduced equation (3.43),

$$E_{N,M\,nm} = \int_\Omega \zeta_n\, \psi_m\, d\Omega, \quad 1 \leq n \leq N,\ 1 \leq m \leq M, \tag{3.51}$$

$$G_{N,M\,nm} = \sum_{k=1}^{N_{\text{elem}}} \int_{T_h^k} \tau_k \boldsymbol{U} \cdot \nabla\zeta_n\, \psi_m\, d\Omega, \quad 1 \leq n \leq N,\ 1 \leq m \leq M, \tag{3.52}$$

$$D_{M,N\,mn} = \zeta_n(\boldsymbol{z}_m), \quad 1 \leq n \leq N,\ 1 \leq m \leq M, \tag{3.53}$$

$$\bar{u}_{M\,m} = \bar{u}(\boldsymbol{z}_m), \quad 1 \leq m \leq M. \tag{3.54}$$

After solving the nonlinear system of equations (3.43) it is possible to evaluate the reduced order output as

$$\boldsymbol{o}_{N,M}(\boldsymbol{\mu}) = \boldsymbol{L}_N^O\, \boldsymbol{u}_{N,M}(\boldsymbol{\mu}) + \boldsymbol{L}_0^O, \tag{3.55}$$

where $\boldsymbol{L}_N^O \in \mathbb{R}^{N_s \times N}$ and $\boldsymbol{L}_0^O \in \mathbb{R}^{N_s}$ are given by

$$L_{N\,in}^O = \int_{\Gamma_i} \zeta_n\, d\Gamma, \quad 1 \leq i \leq N_s,\ 1 \leq n \leq N \tag{3.56}$$

$$L_{0\,i}^O = \int_{\Gamma_i} \bar{u}\, d\Gamma, \quad 1 \leq i \leq N_s, \tag{3.57}$$

which, in matrix form, reduce to

$$\boldsymbol{L}_N^O = \boldsymbol{L}^O \, \boldsymbol{Z} \tag{3.58}$$

$$\boldsymbol{L}_0^O = \boldsymbol{L}^O \, \bar{\boldsymbol{u}}. \tag{3.59}$$

### 3.6.3 Solution Method

The nonlinear system of equations (3.43) is solved using the same damped Newton algorithm implemented for solving the full-order problem obtained from the truth finite element discretization: given a current iterate $\boldsymbol{u}_{N,M}^p$ we find an increment $\Delta\boldsymbol{u}_{N,M}^p$ such that

$$\left.\frac{\partial \boldsymbol{R}_N}{\partial \boldsymbol{u}_{N,M}}\right|_{\boldsymbol{u}_{N,M}^p} \Delta\boldsymbol{u}_{N,M}^p = -\boldsymbol{R}_N\big(\boldsymbol{u}_{N,M}^p(\boldsymbol{\mu});\boldsymbol{\mu}\big)\,, \tag{3.60}$$

where $\boldsymbol{R}_N\big(\boldsymbol{u}_{N,M}^p(\boldsymbol{\mu});\boldsymbol{\mu}\big)$ is the residual of the nonlinear system of equations (3.43), defined as

$$\begin{aligned}\boldsymbol{R}_N\big(\boldsymbol{u}_{N,M}^p(\boldsymbol{\mu});\boldsymbol{\mu}\big) &= \boldsymbol{A}_0 + \boldsymbol{A}_N\boldsymbol{u}_{N,M}^p \\ &\quad + (\boldsymbol{E}_{N,M} + \boldsymbol{G}_{N,M})\, s\big(\bar{\boldsymbol{u}}_M + \boldsymbol{D}_{M,N}\boldsymbol{u}_{N,M}^p(\boldsymbol{\mu});\boldsymbol{\mu}\big) - \boldsymbol{F}_N,\end{aligned} \tag{3.61}$$

and $\left.\frac{\partial \boldsymbol{R}_N}{\partial \boldsymbol{u}_{N,M}}\right|_{\boldsymbol{u}_{N,M}^p} = \boldsymbol{J}_N\left(\boldsymbol{u}_{N,M}^p(\boldsymbol{\mu});\boldsymbol{\mu}\right) \in \mathbb{R}^{N\times N}$ is the Jacobian of the residual evaluated at the current iteration. The computation of this Jacobian must be performed at every iteration using the expression below and has a cost of $O(MN^2)$.

$$\boldsymbol{J}_N\left(\boldsymbol{u}_{N,M}^p(\boldsymbol{\mu});\boldsymbol{\mu}\right) = \boldsymbol{A}_N + (\boldsymbol{E}_{N,M} + \boldsymbol{G}_{N,M}) \left.\frac{\partial \boldsymbol{S}_M}{\partial \boldsymbol{u}_{N,M}}\right|_{\boldsymbol{u}_{N,M}^p}, \tag{3.62}$$

where $\left.\frac{\partial \boldsymbol{S}_M}{\partial \boldsymbol{u}_{N,M}}\right|_{\boldsymbol{u}_{N,M}^p}$ is given by

$$\left.\frac{\partial S_{M\,m}}{\partial u_{N,M\,n}}\right|_{\boldsymbol{u}^p_{N,M}} = \zeta_n(\boldsymbol{z}_m) A e^{-\frac{E}{d-u_{N,M}(\boldsymbol{z}_m)}} \left\{ c - 2u_{N,M}(\boldsymbol{z}_m) \right. \\ \left. - \frac{E\,(c - u_{N,M}(\boldsymbol{z}_m))\,u_{N,M}(\boldsymbol{z}_m)}{\left(u_{N,M}(\boldsymbol{z}_m) - d\right)^2} \right\}, \\ 1 \le m \le M,\ 1 \le n \le N, \qquad (3.63)$$

where

$$u_{N,M}(\boldsymbol{z}_m) = \bar{u}(\boldsymbol{z}_m) + \sum_{n=1}^{N} u_{N,M\,n}\, \zeta_n(\boldsymbol{z}_m). \qquad (3.64)$$

After solving the linear system of equations (3.60) at cost $O(N^3)$, the solution is updated,

$$\boldsymbol{u}^{p+1}_{N,M} = \boldsymbol{u}^p_{N,M} + \alpha \Delta \boldsymbol{u}^p_{N,M}, \qquad (3.65)$$

where the damping coefficient $\alpha \in [0, 1]$ is selected using the same algorithm described in the finite element implementation.

### 3.6.4 Offline-Online Algorithm

Figures 3-5 and 3-6 show the offline-online decomposition of the reduced basis methodology proposed in this chapter.

In the *offline* stage we form the *parameter-independent* matrices and vectors that will be used during the *online* stage. Note that the operation count for the offline phase is $\mathcal{N}$-dependent — and hence very expensive. The online phase, however, has a computational cost of only $O(MN^2 + N^3)$ per Newton iteration. As will be shown in the next section, $N$ and $M$ are usually low (typically 5-50) and the Newton solver converges in a few iterations so the online phase is very inexpensive. We wish to emphasize that the operation count during the online phase is independent of the number of degrees of freedom of the original finite element model.

**Offline Stage**

1. Compute the snapshot set $\mathcal{S}_K^u \equiv \{\xi_k^u(\boldsymbol{x}) = u(\boldsymbol{\mu}_k),\ 1 \le k \le K\}$;
2. Compute the reduced basis set $\{\zeta_n\}_{n=1}^N$ using the POD procedure;
3. Compute the snapshot set $\mathcal{S}_K^s \equiv \{\xi_k^s(\boldsymbol{x}) = s(u(\boldsymbol{\mu}_k); \boldsymbol{\mu}_k),\ 1 \le k \le K\}$;
4. Compute the set of interpolation basis functions $\{\psi_m\}_{m=1}^M$ using the POD procedure as described in Section 3.5.1;
5. Compute the set of interpolation points $\{\boldsymbol{z}_m\}_{m=1}^M$ using either EIM or BPIM.
6. Form and store the *parameter-independent* matrices $\boldsymbol{A}_0$, $\boldsymbol{A}_N$, $\boldsymbol{E}_{N,M}$, $\boldsymbol{G}_{N,M}$, $\boldsymbol{D}_{M,N}$, $\bar{\boldsymbol{u}}_M$, $\boldsymbol{F}_N$ and $\boldsymbol{L}_N^O$ following Section 3.6.

Figure 3-5: Offline phase: construction of the parameter-independent matrices.

**Online Stage**

1. Select the value of the parameter vector $\boldsymbol{\mu}$ for which the output will be computed;
2. Select initial guess $u^0_{N,M}(\boldsymbol{\mu})$, for example by solving the linear convection-diffusion problem without a reaction term.
3. Iterate using Newton's method until convergence is reached. At every Newton step it is necessary to:
   - 3.1. Assemble Jacobian from (3.62) at cost $O(MN^2)$ and
   - 3.2. Solve linear system (3.60) at cost $O(N^3)$.
4. Calculate output $\boldsymbol{o}_{N,M}(\boldsymbol{\mu}) = \boldsymbol{L}_N^O\,\boldsymbol{u}_{N,M}(\boldsymbol{\mu})$ at cost $O(NN_s)$

Figure 3-6: Online phase: compute the value of the output for every value of the parameter vector $\boldsymbol{\mu}$.

### 3.6.5 Implementation Remarks

There are a few important details that must be considered while assembling the matrices $\boldsymbol{E}_{N,M}$ and $\boldsymbol{G}_{N,M}$ in order to guarantee that the reduced basis approximation $u_{N,M}(\boldsymbol{\mu})$ will converge to the truth finite element solution $u(\boldsymbol{\mu})$ as $N, M \to \infty$.

First, as discussed in Section 2.4, the Newton iterative scheme used to solve the truth finite element solution computes the Jacobian of the nonlinear residual at every iteration using a Gauss quadrature rule with $N_Q$ quadrature points in every element. For consistency, matrices $\boldsymbol{E}_{N,M}$ and $\boldsymbol{G}_{N,M}$ should be computed using the same number of quadrature points that was used to compute the finite element solution:

$$
\begin{aligned}
E_{N,M\,nm} &= \int_{\Omega} \zeta_n \, \psi_m \, d\Omega \\
&\approx \sum_{k=1}^{N_{\text{elem}}} \sum_{q=1}^{N_Q} w_q \, J^k_{CT}(\xi_1^q, \xi_2^q) \, \zeta_n(\xi_1^q, \xi_2^q) \, \psi_m(\xi_1^q, \xi_2^q) \\
&\qquad\qquad\qquad\qquad 1 \le n \le N, \; 1 \le m \le M
\end{aligned}
\tag{3.66}
$$

and

$$
\begin{aligned}
G_{N,M\,nm} &= \sum_{k=1}^{N_{\text{elem}}} \int_{T_h^k} \tau_k \, \boldsymbol{U} \, \nabla\zeta_n \, \psi_m \, d\Omega, \\
&\approx \sum_{k=1}^{N_{\text{elem}}} \sum_{q=1}^{N_Q} w_q \, J^k_{CT}(\xi_1^q, \xi_2^q) \, \tau_k \, \boldsymbol{U}(\xi_1^q, \xi_2^q) \, \nabla\zeta_n(\xi_1^q, \xi_2^q) \, \psi_m(\xi_1^q, \xi_2^q) \\
&\qquad\qquad\qquad\qquad 1 \le n \le N, \; 1 \le m \le M
\end{aligned}
\tag{3.67}
$$

We wish to emphasize that, since the interpolation basis functions $\psi_m(\boldsymbol{x})$, $1 \le m \le M$ are nonlinear functions of $u(\boldsymbol{x}; \boldsymbol{\mu})$, it is not possible to compute their values at quadrature points by linearly interpolating in between nodal values. Instead, we use the set of snapshots of the solution, $\mathcal{S}_K^u \equiv \{\xi_k^u(\boldsymbol{x}) = u(\boldsymbol{\mu}_k),\ 1 \le k \le K\}$ to obtain a set of snapshots of the nonlinear reaction term evaluated at the quadrature points, $\boldsymbol{x}_j^{QP}$, $1 \le j \le N_{\text{elem}} N_Q$, namely,

$$
\mathcal{S}_K^{s,QP} \equiv \left\{ \boldsymbol{\xi}_k^{s,QP} = s(u(\boldsymbol{x}_j^{QP}; \boldsymbol{\mu}_k); \boldsymbol{\mu}_k),\ 1 \le k \le K,\ 1 \le j \le N_{\text{elem}} N_Q \right\}. \tag{3.68}
$$

Then, the set of basis functions $\{\phi_m\}_{m=1}^M$ that span the interpolation space, $\Phi_M = \operatorname{span}\{\phi_1, \ldots, \phi_M\}$, are computed at the quadrature points using the POD coefficients $a_k^m$, $1 \le k \le K$, $1 \le m \le M$ obtained from solving the eigenvalue problem (3.5),

$$
\boldsymbol{\phi}_m^{QP} = \sum_{k=1}^{K} a_k^m \boldsymbol{\xi}_k^{s,QP}, \quad 1 \le m \le M, \tag{3.69}
$$

where $\boldsymbol{\phi}_m^{QP}$, $1 \le m \le M$ and $\boldsymbol{\xi}_k^{s,QP}$, $1 \le k \le K$ are all vectors of length $N_{\text{elem}} N_Q$.

Finally, the values of the cardinal functions $\{\psi_m\}_{m=1}^{M}$ at the Gauss quadrature points are computed by solving the following system of equations,

$$\boldsymbol{\phi}_i^{QP} = \sum_{j=1}^{M} \phi_i(\boldsymbol{z}_j)\boldsymbol{\psi}_j^{QP}, \quad 1 \leq i \leq M. \tag{3.70}$$

The elements in each vector $\boldsymbol{\psi}_j^{QP}$, $1 \leq j \leq M$ give the values of the functions $\psi_j(\boldsymbol{x})$ at each of the quadrature points. These are the values that are used for evaluating the quadrature sums (3.66) and (3.67).

### 3.6.6 Numerical Results

In order to test the performance of the efficient reduced basis approximation we introduce a regular test grid of 23 × 23 points over domain $\mathcal{D}$, $\Xi_{529}^{\text{test}}$. This is the same grid that was used in Section 3.5.3 for evaluating the convergence of the EIM and BPIM interpolation methods.

In Figure 3-7 we show a comparison between the solution field computed using the efficient reduced basis approximation and the truth finite element solution for a random point in $\Xi_{529}^{\text{test}}$. The reconstructed field is virtually indistinguishable from the original finite element solution. Note that reconstructing the solution field requires computing the series expansion $u_{N,M}(\boldsymbol{x};\boldsymbol{\mu}) = \bar{u}(\boldsymbol{x}) + \sum_{n=1}^{N} u_{N,M\,n}(\boldsymbol{\mu})\,\zeta_n(\boldsymbol{x})$, which has an $\mathcal{N}$-dependent operation count. Computing the approximate solution field $u_{N,M}(\boldsymbol{x})$ is not required for evaluating the output of interest, $\boldsymbol{o}_{N,M}(\boldsymbol{\mu})$ and, therefore, this step is never performed during the online phase of the reduced basis algorithm. The approximate field has been reconstructed here simply for visualization purposes but we emphasize that this step should never be performed online because it requires $O(\mathcal{N})$ operations.

Figure 3-7 provides a nice visual comparison between the reduced basis approximation and the finite element solution at one test point in parameter space. However, this is not by any means a rigorous measure of accuracy of the reduced basis approximation. For the purpose of evaluating the accuracy of our reduced order model we

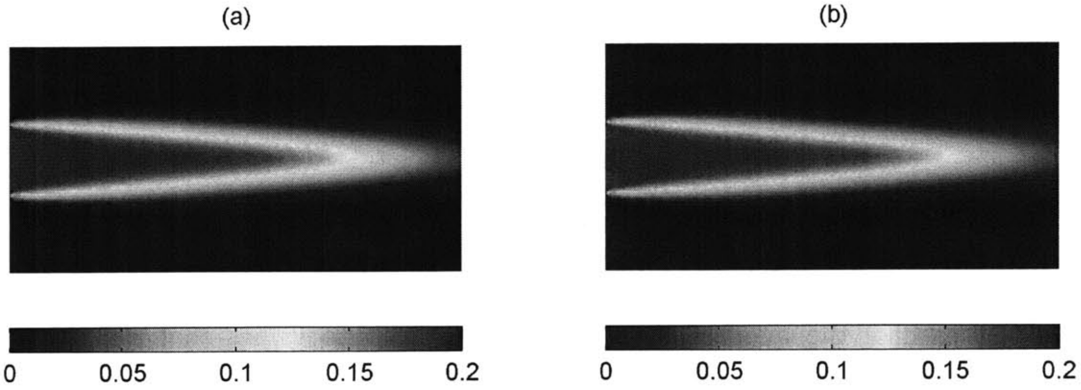

Figure 3-7: Comparison of truth finite element solution (a) and reconstructed solution using the efficient reduced basis approximation (b) with $N = 40$, $M = 50$ and EIM interpolation. Solution corresponds to point 332 in $\Xi^{\text{test}}_{529}$, which is given by $\boldsymbol{\mu}_{332} = (6.4318, 0.1091)$.

define the average relative error norm of the output as

$$\varepsilon^{o}_{N,M\,\text{ave,rel}} = \operatorname*{mean}_{\boldsymbol{\mu}\in\Xi^{\text{test}}_{529}} \frac{\|\boldsymbol{o}(\boldsymbol{\mu}) - \boldsymbol{o}_{N,M}(\boldsymbol{\mu})\|}{\|\boldsymbol{o}(\boldsymbol{\mu})\|} \tag{3.71}$$

Figure 3-8 shows $\varepsilon^{o}_{N,M\,\text{ave,rel}}$ as a function of $N$ and $M$ for a reduced basis approximation that uses the EIM for interpolating the nonlinear reaction term. For a given value of $M$, the error decreases as $N$ is increased until a point is reached where the convergence stalls: the error remains constant regardless of the number of basis functions added to the approximation space. Convergence stalls when the error in the solution is dominated by the error due to the approximation of the nonlinear reaction term. At the point where the error curve levels off, it is necessary to increase the number of interpolation functions $M$ used in the coefficient-function approximation in order to reduce the error further. That is, for a given value of $N$ it is always possible to find a sufficiently large value of $M$ such that the error introduced by the coefficient-function approximation does not have an appreciable effect on the total output error.

Using EIM or BPIM for the coefficient-function approximation only has an effect on the point where convergence stalls. That is, for a given value of $M$, the minimum

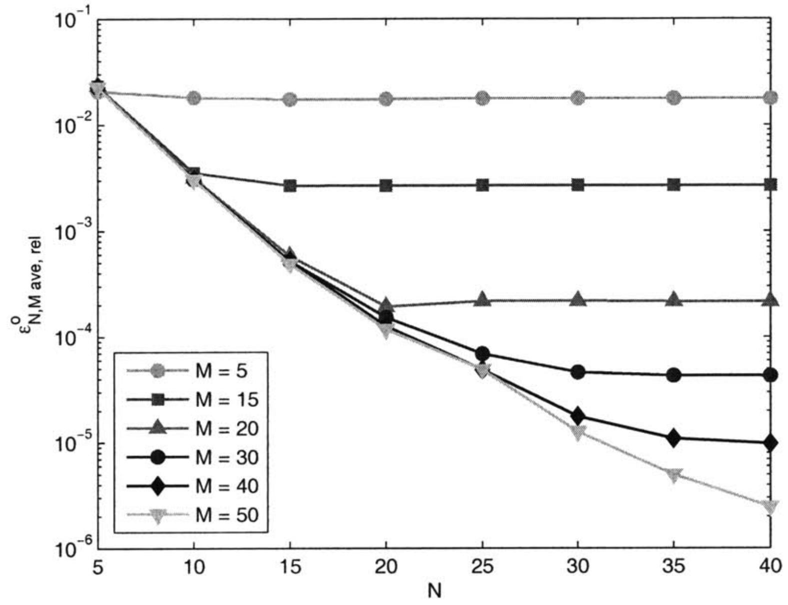

Figure 3-8: Average relative error norm of the output $\varepsilon^o_{N,M\,\text{ave,rel}}$ as a function of $N$ and $M$. The Empirical Interpolation Method was used for developing the coefficient-function approximation for the nonlinear reaction term.

achievable error (for sufficiently large $N$) when using BPIM is less than when using EIM. This is illustrated in Figure 3-9. Note that before the error curves level off both methods provide almost identical results. The difference is that the error curves for EIM level off sooner than those corresponding to BPIM for the same value of $M$.

Finally, in Table 3.1 we present a summary of the numerical results obtained with the efficient reduced basis approximation presented in this chapter. In addition to showing the maximum and average relative errors in the output over the test grid $\Xi^{\text{test}}_{529}$, Table 3.1 also includes a comparison of the online computational time of the reduced basis approximation versus the computational time required by the finite element solver to compute the truth solution at $\boldsymbol{\mu}_{529} = (7.25, 0.05)$. Note that the reduced basis approximation, $\boldsymbol{o}_{N,M}(\boldsymbol{\mu})$, can be computed more than $50,000$ times faster than the truth solution $\boldsymbol{o}(\boldsymbol{\mu})$ for all values of $N$ and $M$ shown in this table. Furthermore, the convergence of the reduced basis approximation is excellent and errors decrease rapidly as $N$ is increased. As a concluding remark we note that the offline stage required 196 "truth" finite element solutions for building the efficient

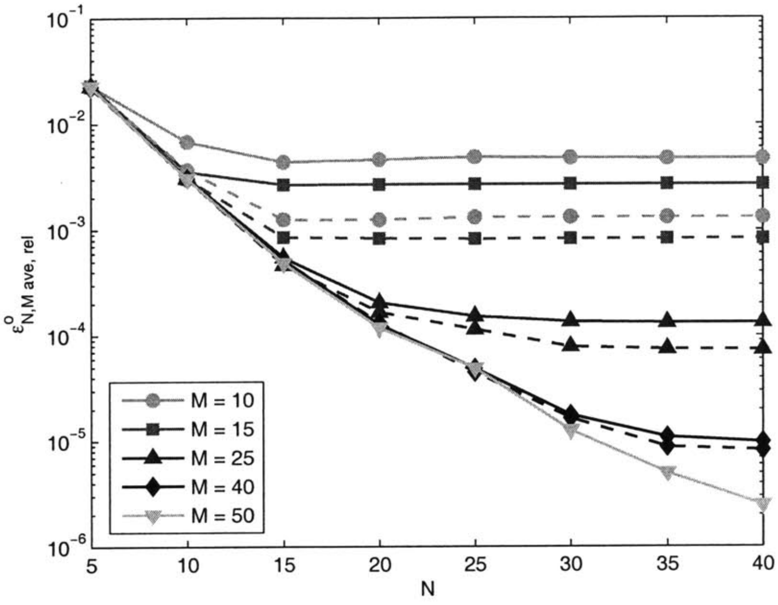

Figure 3-9: Average relative error norm of the output $\varepsilon^{o}_{N,M\,\text{ave,rel}}$ as a function of $N$ and $M$ when coefficient-function approximation is built using EIM (solid lines) versus BPIM (dashed lines).

reduced basis approximation developed in this chapter. The online cost of computing a reduced basis solution after the reduced order model has been built is negligible so one would need to compute at least 196 reduced order solutions in order to amortize the computational cost of the offline stage. In practice, building a reduced basis approximation of a complex nonlinear problem is not a completely straightforward task (the reader has probably been able to appreciate this fact after going through this chapter) so the upfront investment is only justified if the reduced order model is used multiple times (significantly more than 196 times). In the real-time context, however, building a reduced order model may be the only viable alternative for computing results in the required amount of time so the cost of building a reduced basis approximation can be justified without the need for calculating a break-even number of reduced order computations.

|  |  | ERBA |  |  | FEM |
|---|---|---|---|---|---|
| N | M | Avg. rel. error | Max. rel. error | Online time | Comp. time |
| 5 | 50 | $2.25\,\mathrm{E}-02$ | $9.73\,\mathrm{E}-02$ | $1.59\,\mathrm{E}-05$ | 1 |
| 10 | 50 | $3.03\,\mathrm{E}-03$ | $2.78\,\mathrm{E}-02$ | $1.61\,\mathrm{E}-05$ | 1 |
| 20 | 50 | $1.18\,\mathrm{E}-04$ | $2.00\,\mathrm{E}-03$ | $1.63\,\mathrm{E}-05$ | 1 |
| 30 | 50 | $1.26\,\mathrm{E}-05$ | $4.48\,\mathrm{E}-04$ | $1.71\,\mathrm{E}-05$ | 1 |
| 40 | 50 | $2.47\,\mathrm{E}-06$ | $1.34\,\mathrm{E}-04$ | $2.00\,\mathrm{E}-05$ | 1 |

Table 3.1: Maximum and average relative errors in the output of the efficient reduced basis approximation (ERBA) over the test grid $\Xi_{529}^{\mathrm{test}}$ and online computational time required by the reduced order solver as a function of $N$ for $M = 50$. Computational times are normalized with respect to the time required by the finite element solver to compute the truth solution at $\boldsymbol{\mu}_{529} = (7.25, 0.05)$.

# Chapter 4

# Model-Constrained Greedy Adaptive Sampling

## 4.1 Introduction

In Chapter 3 we presented the mathematical formulation for obtaining an efficient reduced basis approximation of the nonlinear convection-diffusion-reaction problem that was introduced in Chapter 2. It has been shown that, given an adequate set of snapshots, it is possible to obtain an approximate reduced basis solution that is arbitrarily close to the "truth" approximation. Furthermore, the online computation of the reduced basis approximation is several orders of magnitude cheaper than the corresponding finite element solution. However, there is a caveat that still needs to be addressed: as we increase the number of basis functions and nonlinear interpolation functions, the reduced basis approximation described in Chapter 3 is only guaranteed to converge to the "truth" approximation for all values of the parameter vector $\boldsymbol{\mu} \in \mathcal{D}$ if an adequate set of snapshots is used to build the reduced bases. In this chapter we address the issue of how to obtain the required set of snapshots by sampling the solution space in an efficient manner, especially when the dimension of the parameter space becomes large.

The word "adequate" can be somewhat vague so we will start by defining what constitutes an adequate set of snapshots. There are two main requirements that a set of snapshots should satisfy in order to be considered "adequate": first, the snapshots should span the entire solution space such that any "truth" solution computed for a particular value of the parameter vector $\boldsymbol{\mu} \in \mathcal{D}$ can be represented as a linear combination of snapshots. Second, the set of snapshots must be of reasonable size such that the offline phase of the reduced basis method does not become computationally intractable.

When the parameter domain $\mathcal{D} \subset \mathbb{R}^P$ is of low dimension one can easily generate an adequate set of snapshots by defining a regular grid $\Xi_K^{\text{train}}$ over $\mathcal{D}$ and choosing the sample set $S_K = \{\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K\}$ as the nodes of this grid. The set of solution snapshots $\mathcal{S}_K^u \equiv \{\xi_k^u(\boldsymbol{x}) = u(\boldsymbol{\mu}_k),\ \boldsymbol{\mu}_k \in S_K,\ 1 \leq k \leq K\}$ is computed by solving the governing PDE at every grid point. The set of snapshots of the nonlinear term is then obtained by evaluating the nonlinear field $s(\cdot; \boldsymbol{\mu})$ at every grid point: $\mathcal{S}_K^s \equiv \{\xi_k^s(\boldsymbol{x}) = s(u(\boldsymbol{x}; \boldsymbol{\mu}_k); \boldsymbol{\mu}_k),\ \boldsymbol{\mu}_k \in S_K,\ 1 \leq k \leq K\}$. Recall that this is the procedure that was used in Chapter 3. It is clear that this sampling method satisfies the first requirement for obtaining an adequate set of snapshots because it covers the entire parameter domain of interest. For the particular case studied in Chapter 3 this method also satisfied the second requirement because the entire domain $\mathcal{D}$ was covered with just a couple of hundred samples. However, the number of grid points required for this sampling method grows exponentially as the dimension of the parameter domain, $P$, increases. Therefore, for $P > 3$ overlaying a uniform grid over the parameter domain of interest becomes intractable.

For linear problems it is possible to compute a cheap and rigorous output error bound following the same offline-online decomposition that is used for computing the output of interest. This fact can be exploited in order to design a greedy algorithm that adaptively selects snapshots by finding the location in a training parameter set where the error bound is maximal. At every greedy cycle the algorithm evaluates the error bound at every point in the training set, finds the location where the error bound is maximal and updates the set of snapshots with the solution at this sample point. A new reduced order model is then built using the updated set of snapshots and the process is repeated until the desired accuracy is reached. For details regarding this greedy algorithm the reader is referred to [23, 24, 25, 41, 44, 47]. We emphasize that this sampling method is still a grid-based method where the critical ingredient is an error bound that can be computed without knowing the "truth" approximation at every grid point. For general nonlinear problems, however, it is not possible to compute an error bound without actually computing the "truth" finite element solution. Therefore, determining the maximum error point in the training set requires evaluating the very expensive finite element solution at every point in the grid and this method becomes intractable as $P$ increases.

Recently, a general model-constrained greedy algorithm has been proposed to address the challenges of sampling a high-dimensional parameter space without having to compute any error bounds or full-order solutions at every point in a discrete training parameter set defined *a priori* [8, 9]. This gridless method determines the location in parameter space where the error between the 'truth" approximation and the reduced basis approximation is maximum by solving a continuous optimization problem without the need for discretizing the parameter domain $\mathcal{D}$. The greedy optimization problem was formulated and tested in [8, 9] for steady and unsteady problems that are linear in the state vector. Herein we extend this methodology to nonlinear problems.

In this chapter we first present the model-constrained greedy adaptive sampling methodology. Next, the mathematical formulation is derived for the same 2-parameter problem that was presented in Chapter 2 and the numerical results are compared against those obtained using grid sampling. Finally, the diffusivity $\kappa$ is added as a third parameter to the nonlinear convection-diffusion-reaction PDE and we analyze the performance of the model-constrained greedy sampling methodology for the case where $\mathcal{D} \subset \mathbb{R}^3$.

## 4.2 Model-Constrained Greedy Sampling Methodology

In this section we describe the model-constrained greedy sampling methodology and derive the formulation for sampling the parameter space $\mathcal{D}$ where the parameters of the combustion PDE (2.1) reside. The reduced order model built using this greedy sampling algorithm is then compared against the reduced basis approximation that was obtained in Chapter 2 using uniform grid sampling.

### 4.2.1 General Greedy Adaptive Sampling Algorithm

The greedy algorithm described herein seeks to improve the accuracy of an existing reduced order model by adaptively adding a new snapshot at every iteration. Every new snapshot is sampled at the location in parameter space where the error between the "truth" and reduced basis outputs is maximal. The main steps of the algorithm are described below.

1. The set of sample points is initialized by arbitrarily choosing one point in parameter space $S_1 = \{\boldsymbol{\mu}_1 \in \mathcal{D}\}$. One of the corners of the domain $\mathcal{D}$ is usually a good option but any point can be chosen. We then generate the snapshot sets $S_1^u \equiv \{\xi_1^u(\boldsymbol{x}) = u(\boldsymbol{\mu}_1)\}$ and $S_1^s \equiv \{\xi_1^s(\boldsymbol{x}) = s(u(\boldsymbol{\mu}_1); \boldsymbol{\mu}_1)\}$. Note that the initial sets of snapshots can contain several snapshots instead of just one for example by selecting all the corners of the parameter domain $\mathcal{D}$ or by overlaying a very coarse initial grid over $\mathcal{D}$. This will result in a more accurate initial reduced order model so fewer greedy cycles will be required in order to obtain the desired level of accuracy.

2. The offline stage outlined in Figure 3-5 is performed in order to obtain the parameter-independent matrices $\boldsymbol{A}_0$, $\boldsymbol{A}_N$, $\boldsymbol{E}_{N,M}$, $\boldsymbol{G}_{N,M}$, $\boldsymbol{D}_{M,N}$, $\bar{\boldsymbol{u}}_M$, $\boldsymbol{F}_N$ and $\boldsymbol{L}_N^O$ for $N = 1$ and $M = 1$ (or $N = N_{\text{init}}$ and $M = M_{\text{init}}$ if the initial sample set contains more than one point).

3. While the error between the reduced basis approximation and the truth output is greater than a user-specified tolerance, the reduced order model is iteratively improved by performing Steps 3a through 3d below:

   (a) At iteration $K$ we find the location in parameter space $\mathcal{D}$ where the error between the "truth" and reduced basis outputs outputs is maximal: $\boldsymbol{\mu}_K^u = \arg\max_{\boldsymbol{\mu}\in\mathcal{D}} \|\boldsymbol{o}(\boldsymbol{\mu}) - \boldsymbol{o}_{N,M}(\boldsymbol{\mu})\|$. Note that, in general, it is possible to optimize the reduced basis space used to approximate the solution and the interpolation space used to compute the nonlinear coefficient-function approximation separately. This results in different sample points for computing the snapshot of the solution $\xi_K^u(\boldsymbol{x})$ and the snapshot of the nonlinear term $\xi_K^s(\boldsymbol{x})$ at every iteration $K$. However, in this thesis we will always use the output error as the only indicator for determining the optimal sampling point. Therefore, the same point is used for sampling both the solution space and the nonlinear reaction term space: $\boldsymbol{\mu}_K^u \equiv \boldsymbol{\mu}_K^s$. Since both sample points are identical at every iteration, they will bear no superscript.

   (b) The sample set is then updated by adding the new point found in Step 3a: $S_K = S_{K-1} \cup \boldsymbol{\mu}_K^u$.

   (c) The new solution and reaction snapshots are given by $\xi_K^u(\boldsymbol{x}) = u(\boldsymbol{\mu}_K)$ and $\xi_K^s(\boldsymbol{x}) = s(u(\boldsymbol{\mu}_K); \boldsymbol{\mu}_K)$ respectively and the sets of snapshots are updated by adding these new snapshots to the existing ones: $\mathcal{S}_K^u = \mathcal{S}_{K-1}^u \cup \xi_K^u(\boldsymbol{x})$ and $\mathcal{S}_K^s = \mathcal{S}_{K-1}^s \cup \xi_K^s(\boldsymbol{x})$.

   (d) The parameter-independent matrices for the new reduced order model are computed using the offline stage steps outlined in Figure 3-5.

The procedure outlined above is very general and can be implemented in different ways, yielding different greedy algorithms that are efficient for treating various kinds problems. The main differences between the different implementations reside in the methodology used to determine the optimal sampling point at every cycle (Step 3a).

Patera and co-workers have extensively used a discrete version of the greedy adaptive sampling algorithm presented in this section for obtaining reduced basis approximations to linear or, at most, quadratically nonlinear problems in a very efficient manner [23, 24, 25, 41, 44, 47]. For this version of the algorithm, a discrete training parameter set $\Xi^{\text{train}}$ is generated by defining a suitably fine grid over the parameter domain $\mathcal{D}$. Note that this parameter set is defined *a priori*. The error between the "truth" and reduced basis outputs is then maximized over the training set $\Xi^{\text{train}}$ at every greedy cycle. The critical ingredient of this algorithm is an efficient error bound $\Delta_N(\boldsymbol{\mu})$ for the reduced basis error $\|\boldsymbol{o}(\boldsymbol{\mu}) - \boldsymbol{o}_N(\boldsymbol{\mu})\|$. The error bound must be *"efficient"* in the sense that, in the limit of many evaluations, the cost to evaluate $\Delta_N(\boldsymbol{\mu})$ is independent of $\mathcal{N}$ (it is not necessary to compute the "truth" output at every point in $\Xi^{\text{train}}$ in order to evaluate the error bound). This efficiency is accomplished by deriving an offline-online decomposition that is analogous to the one used to compute the reduced basis output $\boldsymbol{o}_N(\boldsymbol{\mu})$. The error bounds developed by Patera and co-workers for multiple types of problems are also *rigorous* in the sense that $\Delta_N(\boldsymbol{\mu})$ is never less than the true error and *sharp*, meaning that $\Delta_N(\boldsymbol{\mu})$ is not too much greater than the true error. In this version of the algorithm, the optimization problem in Step 3a, $\boldsymbol{\mu}_K = \arg\max_{\boldsymbol{\mu}\in\mathcal{D}} \|\boldsymbol{o}(\boldsymbol{\mu}) - \boldsymbol{o}_N(\boldsymbol{\mu})\|$, is replaced by $\boldsymbol{\mu}_K = \arg\max_{\boldsymbol{\mu}\in\Xi_{\text{train}}} \Delta_N(\boldsymbol{\mu})$.

The reader has probably noticed that all the reduced basis quantities used while discussing the discrete version of the greedy adaptive sampling algorithm only bear the subscript $N$ instead of the traditional subscript "$N, M$" that has been used throughout this thesis to indicate that, in our problem, the reduced basis approximation is not only a function of the number of basis functions used to span the reduced solution space but also of the number of interpolation functions used to compute the coefficient-function approximation of the nonlinear term. The omission of the subindex $M$ has been intentional in order to emphasize that the error bound $\Delta_N(\boldsymbol{\mu})$ only maintains its efficiency when the problem is linear, or at most quadratic, in the state variable and affine in the parameters. For general highly nonlinear or nonaffine problems the error bound loses its efficiency — the computation of $\Delta_N(\boldsymbol{\mu})$ is no longer independent of $\mathcal{N}$. Thus, the discrete version of the greedy adaptive sampling algorithm applied to the nonlinear combustion problem analyzed in this thesis requires evaluating the "truth" output at every point in $\Xi^{\text{train}}$ at every greedy cycle, which clearly becomes intractable when sampling high dimensional parameter spaces (by "high" we mean $P > 3$).

A recently proposed approach to address the shortcomings of the discrete greedy algorithm when dealing with inefficient ($\mathcal{N}$-dependent) error bounds is the model-constrained greedy-based adaptive sampling presented in [8, 9]. In this methodology, Step 3a of the greedy algorithm is formulated as an optimization problem that searches for optimal parameter points in the continuous parameter space $\mathcal{D}$. The resulting PDE-constrained optimization problem is solved using standard optimization techniques. Note that, in general, the objective function used to formulate the optimization problem may have several local maxima so the optimization solver is not guaranteed to converge to the global maximum. This means that the value of the objective function at sample point $\boldsymbol{\mu}_K$ computed during iteration $K$ can no longer be used as a *rigorous* error bound for the reduced basis approximation built using $K - 1$ snapshots. Therefore, the criterion for stopping the greedy algorithm in Step 3 is not as clearly defined as in the discrete version where the parameter points at every iteration were guaranteed to be globally optimal over $\Xi^{\text{train}}$. However, as discussed above, the discrete version of the greedy algorithm becomes intractable for high-dimensional nonlinear problems so we will willingly pay the price of less rigorous error indicators in exchange for a more efficient method.

In [8, 9] the continuous model-constrained greedy-based sampling algorithm was tested for steady and unsteady problems that were linear in the state variable. Here, we apply it to the nonlinear combustion problem that was formulated in Chapter 2.

### 4.2.2 PDE-Constrained Optimization Problem for Greedy Algorithm

In this section we formulate the problem to find a local maximum of the error between the "truth" and reduced outputs at every greedy cycle. In this thesis we formulate the maximization problem in terms of the true output error although we could also use some type of error bound or error indicator if one was available.

The PDE-constrained optimization problem is formulated in discrete form; that is, we take advantage of the discretization of the full and reduced governing equations that was performed in Chapters 2 and 3 prior to forming the Lagrangian function and deriving the optimality conditions. Note that this is not the only way to formulate the problem: it would also be possible to form the infinite-dimensional Lagrangian and then write the infinite-dimensional optimality conditions in terms of the continuous weak forms of the full and reduced governing PDEs of the problem of interest. As discussed in [1], when these infinite-dimensional conditions are discretized, they may result in different optimality conditions than those obtained by first discretizing the PDE constraints and then differentiating the Lagrangian function. Since in our case we already have the discrete version of the governing PDEs as well as efficient solvers for the resulting nonlinear systems of equations we will follow the path of first discretizing the constraints and then differentiating to form the optimality conditions.

We define the cost functional in terms of the $\ell^2$-norm of the true output error as follows:

$$\mathcal{J}(\boldsymbol{u}, \boldsymbol{u}_{N,M}, \boldsymbol{\mu}) = \frac{1}{2}\|\boldsymbol{o} - \boldsymbol{o}_{N,M}\|_2^2 \tag{4.1}$$

Given the reduced basis approximation at the current greedy cycle we find the point in parameter space where the true output error is maximal by solving the optimization problem

$$\begin{aligned} \max_{\boldsymbol{u}, \boldsymbol{u}_{N,M}, \boldsymbol{\mu}} \mathcal{J}(\boldsymbol{u}, \boldsymbol{u}_{N,M}, \boldsymbol{\mu}) &\overset{def}{=} \frac{1}{2}\|\boldsymbol{o} - \boldsymbol{o}_{N,M}\|_2^2 \\ \text{subject to } \boldsymbol{R}_N(\boldsymbol{u}_{N,M}; \boldsymbol{\mu}) &= 0 \\ \boldsymbol{o}_{N,M} &= \boldsymbol{L}_N^O \boldsymbol{u}_{N,M} + \boldsymbol{L}_0^O \\ \boldsymbol{R}(\boldsymbol{u}; \boldsymbol{\mu}) &= 0 \\ \boldsymbol{o} &= \boldsymbol{L}^O \boldsymbol{u} \\ \boldsymbol{\mu}_{\min} \leq \boldsymbol{\mu} &\leq \boldsymbol{\mu}_{\max} \end{aligned} \tag{4.2}$$

where $\boldsymbol{u} \in \mathbb{R}^{\mathcal{N}}$, $\boldsymbol{u}_{N,M} \in \mathbb{R}^N$ are the "truth" and reduced state variables, $\boldsymbol{\mu} \in \mathbb{R}^P$ is the parameter vector or decision variable, $\mathcal{J} \in \mathbb{R}$ is the objective function, $\boldsymbol{R} \in \mathbb{R}^{\mathcal{N}}$,

$\boldsymbol{R}_N \in \mathbb{R}^N$ are the full and reduced discretized equations respectively and $\boldsymbol{o} \in \mathbb{R}^{N_s}$, $\boldsymbol{o}_{N,M} \in \mathbb{R}^{N_s}$ are the "truth" and reduced linear outputs of interest, which are computed using the matrices $\boldsymbol{L}^O \in \mathbb{R}^{N_s \times \mathcal{N}}$, $\boldsymbol{L}_N^O \in \mathbb{R}^{N_s \times N}$, and $\boldsymbol{L}_0^O \in \mathbb{R}^{N_s}$ defined in equations (2.47) and (3.56). The upper and lower bounds for the parameter vector are given by $\boldsymbol{\mu}_{\max}$ and $\boldsymbol{\mu}_{\min}$ respectively. Recall that the parameter vector is formed by the two parameters ($P = 2$) that control the combustion reaction; i.e., $\boldsymbol{\mu} = (\log(A), E)$.

Substituting the output constraints into the objective function and transforming the maximization problem into a minimization problem by changing the sign of the objective function yields

$$\begin{aligned} \min_{\boldsymbol{u}, \boldsymbol{u}_{N,M}, \boldsymbol{\mu}} \mathcal{J}(\boldsymbol{u}, \boldsymbol{u}_{N,M}) &\overset{def}{=} -\frac{1}{2} \|\boldsymbol{L}^O \boldsymbol{u} - \boldsymbol{L}_N^O \boldsymbol{u}_{N,M} - \boldsymbol{L}_0^O\|_2^2 \qquad (4.3) \\ \text{subject to } \boldsymbol{R}_N(\boldsymbol{u}_{N,M}; \boldsymbol{\mu}) &= 0 \\ \boldsymbol{R}(\boldsymbol{u}; \boldsymbol{\mu}) &= 0 \\ \boldsymbol{\mu}_{\min} \leq \boldsymbol{\mu} &\leq \boldsymbol{\mu}_{\max} \end{aligned}$$

where the full state discretized equations, $\boldsymbol{R} \in \mathbb{R}^{\mathcal{N}}$, were defined in (2.40) and the reduced state discretized equations, $\boldsymbol{R}_N \in \mathbb{R}^N$ were defined in (3.61).

### 4.2.3 Solution of the PDE-Constrained Optimization Problem

The optimality conditions for the constrained optimization problem (4.3) are derived by introducing the full and reduced adjoint variables, $\boldsymbol{\lambda} \in \mathbb{R}^{\mathcal{N}}$ and $\boldsymbol{\lambda}_N \in \mathbb{R}^N$ and defining the Lagrangian function,

$$\mathcal{L}(\boldsymbol{u}, \boldsymbol{u}_{N,M}, \boldsymbol{\mu}, \boldsymbol{\lambda}, \boldsymbol{\lambda}_N) \overset{\text{def}}{=} \mathcal{J}(\boldsymbol{u}, \boldsymbol{u}_{N,M}) + \boldsymbol{\lambda}^T \boldsymbol{R}(\boldsymbol{u}; \boldsymbol{\mu}) + \boldsymbol{\lambda}_N^T \boldsymbol{R}_N(\boldsymbol{u}_{N,M}; \boldsymbol{\mu}) \qquad (4.4)$$

Note that the bound constraints are not included in the Lagrangian since they are treated separately inside the optimization solver. The first order Karush-Kuhn-Tucker optimality conditions require stationarity of the Lagrangian — i.e., the gradient of the Lagrangian must vanish:

$$\begin{bmatrix} \partial_{\boldsymbol{u}}\mathcal{L} \\ \partial_{\boldsymbol{u}_{N,M}}\mathcal{L} \\ \partial_{\boldsymbol{\mu}}\mathcal{L} \\ \partial_{\boldsymbol{\lambda}}\mathcal{L} \\ \partial_{\boldsymbol{\lambda}_N}\mathcal{L} \end{bmatrix} = \begin{bmatrix} \boldsymbol{g}_{\boldsymbol{u}} + \boldsymbol{J}_{\boldsymbol{u}}^T\boldsymbol{\lambda} \\ \boldsymbol{g}_{\boldsymbol{u}_{N,M}} + \boldsymbol{J}_{N\boldsymbol{u}_{N,M}}^T\boldsymbol{\lambda}_N \\ \boldsymbol{J}_{\boldsymbol{\mu}}^T\boldsymbol{\lambda} + \boldsymbol{J}_{N\boldsymbol{\mu}}^T\boldsymbol{\lambda}_N \\ \boldsymbol{R} \\ \boldsymbol{R}_N \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} \tag{4.5}$$

where $\boldsymbol{g}_{\boldsymbol{u}} \in \mathbb{R}^{\mathcal{N}}$ and $\boldsymbol{g}_{\boldsymbol{u}_{N,M}} \in \mathbb{R}^N$ are the gradients of $\mathcal{J}$ with respect to the full and reduced state variables respectively:

$$\boldsymbol{g}_{\boldsymbol{u}} = -(\boldsymbol{L}^O)^T(\boldsymbol{L}^O\boldsymbol{u} - \boldsymbol{L}_N^O\boldsymbol{u}_{N,M} - \boldsymbol{L}_0^O) \tag{4.6}$$

$$\boldsymbol{g}_{\boldsymbol{u}_{N,M}} = (\boldsymbol{L}_N^O)^T(\boldsymbol{L}^O\boldsymbol{u} - \boldsymbol{L}_N^O\boldsymbol{u}_{N,M} - \boldsymbol{L}_0^O), \tag{4.7}$$

$\boldsymbol{J}_{\boldsymbol{u}} \in \mathbb{R}^{\mathcal{N}\times\mathcal{N}}$ is the Jacobian of the full-order nonlinear problem as defined in (2.41) and $\boldsymbol{J}_{N\boldsymbol{u}_{N,M}} \in \mathbb{R}^{N\times N}$ is the Jacobian of the reduced basis equations which can be computed using (3.62). Finally, $\boldsymbol{J}_{\boldsymbol{\mu}} \in \mathbb{R}^{\mathcal{N}\times P}$ and $\boldsymbol{J}_{N\boldsymbol{\mu}} \in \mathbb{R}^{N\times P}$ are the Jacobians of the full and reduced state equations with respect to the decision variables. Since the last two Jacobians were not used in the problem formulation that was derived in previous chapters, their definitions are included below for completeness.

For the two-parameter problem considered in this section the two columns of the Jacobian of the full state equations with respect to the decision variables are given by

$$\begin{aligned} J_{\boldsymbol{\mu}\, i1} = \frac{\partial R_i}{\partial \mu_1} = \sum_{k=1}^{N_{\text{elem}}} \sum_{q=1}^{N_Q} w_q J_{CT}^k(\xi_1^q, \xi_2^q) \left( W_i^k(\xi_1^q, \xi_2^q) + \varphi_i(\xi_1^q, \xi_2^q) \right) \\ \cdot \left\{ u_Q(c - u_Q) e^{-\frac{E}{d - u_Q}} \right\}, \quad 1 \leq i \leq \mathcal{N} \end{aligned} \tag{4.8}$$

and

$$\begin{aligned} J_{\boldsymbol{\mu} i1} = \frac{\partial R_i}{\partial \mu_2} = \sum_{k=1}^{N_{\text{elem}}} \sum_{q=1}^{N_Q} w_q J_{CT}^k(\xi_1^q, \xi_2^q) \left(W_i^k(\xi_1^q, \xi_2^q) + \varphi_i(\xi_1^q, \xi_2^q)\right) \\ \cdot \left\{ u_Q \frac{-c + u_Q}{d - u_Q} A e^{-\frac{E}{d - u_Q}} \right\}, \quad 1 \le i \le \mathcal{N}; \end{aligned} \tag{4.9}$$

where $N_Q$ is the number of quadrature points in each element; $(\xi_1^q, \xi_2^q)$ are the coordinates of quadrature point $q$ in the master element, $w_q$ is the weight associated with quadrature point $q$ and $J_{CT}^k(\xi_1^q, \xi_2^q)$ is the determinant of the Jacobian of the coordinate transformation between element $k$ in physical space and the master element in the computational domain, evaluated at the quadrature point $q$. The term $W_i^k(\xi_1^q, \xi_2^q)$ is given by equation (2.44) and $u_Q$ is the value of the "truth" solution at quadrature point $q$, which is given by

$$u_Q = \sum_{i=1}^{\mathcal{N}} u_i \, \varphi_i(\xi_1^q, \xi_2^q) \tag{4.10}$$

The two columns of the Jacobian of the reduced state equations with respect to the decision variables, $J_{N\boldsymbol{\mu} n1}$ and $J_{N\boldsymbol{\mu} n2}$ with $1 \le n \le N$, are given by

$$\begin{aligned} J_{N\boldsymbol{\mu} n1} &= \frac{\partial R_{Nn}}{\partial \mu_1} \\ &= \sum_{m=1}^{M} B_{N,M\,nm} \left( u_{N,M}(\boldsymbol{z}_m) \left(c - u_{N,M}(\boldsymbol{z}_m)\right) e^{-\frac{E}{d - u_{N,M}(\boldsymbol{z}_m)}} \right) \end{aligned} \tag{4.11}$$

and

$$\begin{aligned} J_{N\boldsymbol{\mu} n2} &= \frac{\partial R_{Nn}}{\partial \mu_2} \\ &= \sum_{m=1}^{M} B_{N,M\,nm} \left( A \, u_{N,M}(\boldsymbol{z}_m) \frac{-c + u_{N,M}(\boldsymbol{z}_m)}{d - u_{N,M}(\boldsymbol{z}_m)} e^{-\frac{E}{d - u_{N,M}(\boldsymbol{z}_m)}} \right) \end{aligned} \tag{4.12}$$

Here, $u_{N,M}(\boldsymbol{z}_m)$ was defined in Section 3.6.3 and is given by

$$u_{N,M}(\boldsymbol{z}_m) = \bar{u}(\boldsymbol{z}_m) + \sum_{n=1}^{N} u_{N,M\,n} \, \zeta_n(\boldsymbol{z}_m). \tag{4.13}$$

and $\boldsymbol{B}_{N,M} = \boldsymbol{E}_{N,M} + \boldsymbol{G}_{N,M}$, where $\boldsymbol{E}_{N,M} \in \mathbb{R}^{N \times M}$, $\boldsymbol{G}_{N,M} \in \mathbb{R}^{N \times M}$ are given by (3.66) and (3.67) respectively.

Now that we have defined all the terms that appear in the optimality conditions we can solve the nonlinear system of equations (4.5) using a Newton method. At every Newton iteration we linearize the system of optimality conditions by formulating the following linear system of equations

$$\nabla^2 \mathcal{L}(\boldsymbol{u}, \boldsymbol{u}_{N,M}, \boldsymbol{\mu}, \boldsymbol{\lambda}, \boldsymbol{\lambda}_N) \cdot \boldsymbol{\Delta} = -\nabla \mathcal{L}(\boldsymbol{u}, \boldsymbol{u}_{N,M}, \boldsymbol{\mu}, \boldsymbol{\lambda}, \boldsymbol{\lambda}_N) \tag{4.14}$$

where $\nabla^2 \mathcal{L}(\boldsymbol{u}, \boldsymbol{u}_{N,M}, \boldsymbol{\mu}, \boldsymbol{\lambda}, \boldsymbol{\lambda}_N) \in \mathbb{R}^{(\mathcal{N}+N+P)\times(\mathcal{N}+N+P)}$ is the Hessian matrix of the Lagrangian and can be block-partitioned according to the state, adjoint and decision variables as follows:

$$\nabla^2 \mathcal{L} = \begin{bmatrix} \boldsymbol{W}_{\boldsymbol{u}\boldsymbol{u}} & \boldsymbol{W}_{\boldsymbol{u}\boldsymbol{u}_{N,M}} & \boldsymbol{W}_{\boldsymbol{u}\boldsymbol{\mu}} & \boldsymbol{J}_{\boldsymbol{u}}^T & 0 \\ \boldsymbol{W}_{\boldsymbol{u}_{N,M}\boldsymbol{u}} & \boldsymbol{W}_{\boldsymbol{u}_{N,M}\boldsymbol{u}_{N,M}} & \boldsymbol{W}_{\boldsymbol{u}_{N,M}\boldsymbol{\mu}} & 0 & \boldsymbol{J}_{N\boldsymbol{u}_{N,M}}^T \\ \boldsymbol{W}_{\boldsymbol{\mu}\boldsymbol{u}} & \boldsymbol{W}_{\boldsymbol{\mu}\boldsymbol{u}_{N,M}} & \boldsymbol{W}_{\boldsymbol{\mu}\boldsymbol{\mu}} & \boldsymbol{J}_{\boldsymbol{\mu}}^T & \boldsymbol{J}_{N\boldsymbol{\mu}}^T \\ \boldsymbol{J}_{\boldsymbol{u}} & 0 & \boldsymbol{J}_{\boldsymbol{\mu}} & 0 & 0 \\ 0 & \boldsymbol{J}_{N\boldsymbol{u}_{N,M}} & \boldsymbol{J}_{N\boldsymbol{\mu}} & 0 & 0 \end{bmatrix} \tag{4.15}$$

$\boldsymbol{\Delta} \in \mathbb{R}^{\mathcal{N}+N+P}$ gives the new search direction in the variables of the problem,

$$\boldsymbol{\Delta} = \begin{bmatrix} \Delta \boldsymbol{u} \\ \Delta \boldsymbol{u}_{N,M} \\ \Delta \boldsymbol{\mu} \\ \Delta \boldsymbol{\lambda} \\ \Delta \boldsymbol{\lambda}_N \end{bmatrix} \tag{4.16}$$

and $\nabla \mathcal{L}(\boldsymbol{u}, \boldsymbol{u}_{N,M}, \boldsymbol{\mu}, \boldsymbol{\lambda}, \boldsymbol{\lambda}_N) \in \mathbb{R}^{\mathcal{N}+N+P}$ is the gradient of the Lagrangian as shown in (4.5). Refer to the next section of this chapter for the definitions of the nine blocks of the Hessian matrix that have not been defined yet.

The linear system of equations (4.14) is commonly referred to as the Karush-Kuhn-Tucker (KKT) system and its coefficient matrix (4.15) as the KKT matrix.

The KKT system is generally very large and often very ill-conditioned so solving it directly can be a daunting task. In this thesis we use the *reduced space* methodology that can be found in [1, 5]. The basic idea of this methodology is to take advantage of the existing PDE solvers developed in Chapters 2 and 3 to eliminate the state and adjoint equations and variables from (4.14). Then the reduced Hessian system is solved in the remaining decision space. This method, known as *reduced Newton* method, is derived by performing block elimination on the KKT system (4.15). The steps are detailed below.

**Step 1.** Given a value of the decision vector $\boldsymbol{\mu}$ at iteration $k$ we can compute the reduced gradient as follows:

1. Solve the nonlinear state equations

$$\boldsymbol{R}(\boldsymbol{u};\boldsymbol{\mu}) = 0, \tag{4.17}$$

$$\boldsymbol{R}_N(\boldsymbol{u}_{N,M};\boldsymbol{\mu}) = 0, \tag{4.18}$$

using the same solvers that were developed in Chapters 2 and 3.

2. Use the state variables computed in the previous step to solve the adjoint equations,

$$\boldsymbol{\lambda} = -\boldsymbol{J}_{\boldsymbol{u}}^{-T}\boldsymbol{g}_{\boldsymbol{u}}, \tag{4.19}$$

$$\boldsymbol{\lambda}_N = -\boldsymbol{J}_{N\boldsymbol{u}_{N,M}}^{-T}\boldsymbol{g}_{\boldsymbol{u}_{N,M}}, \tag{4.20}$$

where $(\cdot)^{-T}$ is an abbreviation for $((\cdot)^T)^{-1}$.

3. Substitute $\boldsymbol{u}$, $\boldsymbol{u}_{N,M}$, $\boldsymbol{\lambda}$ and $\boldsymbol{\lambda}_N$ from equations (4.17) through (4.20) into the decision equation. The residual of this decision equation is the *reduced gradient*, $\nabla\mathcal{J}$; i.e., the gradient of the cost function while maintaining the state satisfied,

$$\nabla\mathcal{J} = -\boldsymbol{J}_{\boldsymbol{\mu}}^T\boldsymbol{J}_{\boldsymbol{u}}^{-T}\boldsymbol{g}_{\boldsymbol{u}} - \boldsymbol{J}_{N\boldsymbol{\mu}}^T\boldsymbol{J}_{N\boldsymbol{u}_{N,M}}^{-T}\boldsymbol{g}_{\boldsymbol{u}_{N,M}} \tag{4.21}$$

**Step 2.** Incorporate the results from Step 1 into the right-hand side of the Newton step (4.14); that is, substitute the results from Step 1 into the gradient of the Lagrangian so that (4.14) reduces to

$$\begin{bmatrix} \boldsymbol{W}_{\boldsymbol{u}\boldsymbol{u}} & \boldsymbol{W}_{\boldsymbol{u}\boldsymbol{u}_{N,M}} & \boldsymbol{W}_{\boldsymbol{u}\boldsymbol{\mu}} & \boldsymbol{J}_{\boldsymbol{u}}^{T} & 0 \\ \boldsymbol{W}_{\boldsymbol{u}_{N,M}\boldsymbol{u}} & \boldsymbol{W}_{\boldsymbol{u}_{N,M}\boldsymbol{u}_{N,M}} & \boldsymbol{W}_{\boldsymbol{u}_{N,M}\boldsymbol{\mu}} & 0 & \boldsymbol{J}_{N\boldsymbol{u}_{N,M}}^{T} \\ \boldsymbol{W}_{\boldsymbol{\mu}\boldsymbol{u}} & \boldsymbol{W}_{\boldsymbol{\mu}\boldsymbol{u}_{N,M}} & \boldsymbol{W}_{\boldsymbol{\mu}\boldsymbol{\mu}} & \boldsymbol{J}_{\boldsymbol{\mu}}^{T} & \boldsymbol{J}_{N\boldsymbol{\mu}}^{T} \\ \boldsymbol{J}_{\boldsymbol{u}} & 0 & \boldsymbol{J}_{\boldsymbol{\mu}} & 0 & 0 \\ 0 & \boldsymbol{J}_{N\boldsymbol{u}_{N,M}} & \boldsymbol{J}_{N\boldsymbol{\mu}} & 0 & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{\Delta u} \\ \boldsymbol{\Delta u}_{N,M} \\ \boldsymbol{\Delta \mu} \\ \boldsymbol{\Delta \lambda} \\ \boldsymbol{\Delta \lambda}_{N} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -\nabla \mathcal{J} \\ 0 \\ 0 \end{bmatrix} \tag{4.22}$$

**Step 3.** Eliminate $\boldsymbol{\Delta u}$ from the fourth block of equations,

$$\boldsymbol{\Delta u} = -\boldsymbol{J}_{\boldsymbol{u}}^{-1}\boldsymbol{J}_{\boldsymbol{\mu}}\,\boldsymbol{\Delta \mu}. \tag{4.23}$$

**Step 4.** Eliminate $\boldsymbol{\Delta u}_{N,M}$ from the last block of equations,

$$\boldsymbol{\Delta u}_{N,M} = -\boldsymbol{J}_{N\boldsymbol{u}_{N,M}}^{-1}\boldsymbol{J}_{N\boldsymbol{\mu}}\,\boldsymbol{\Delta \mu}. \tag{4.24}$$

**Step 5.** Eliminate $\boldsymbol{\Delta \lambda}$ from the first block of equations,

$$\boldsymbol{\Delta \lambda} = \left(\boldsymbol{J}_{\boldsymbol{u}}^{-T}\boldsymbol{W}_{\boldsymbol{u}\boldsymbol{u}}\boldsymbol{J}_{\boldsymbol{u}}^{-1}\boldsymbol{J}_{\boldsymbol{\mu}} + \boldsymbol{J}_{\boldsymbol{u}}^{-T}\boldsymbol{W}_{\boldsymbol{u}\boldsymbol{u}_{N,M}}\boldsymbol{J}_{N\boldsymbol{u}_{N,M}}^{-1}\boldsymbol{J}_{N\boldsymbol{\mu}} - \boldsymbol{J}_{\boldsymbol{u}}^{-T}\boldsymbol{W}_{\boldsymbol{u}\boldsymbol{\mu}}\right)\boldsymbol{\Delta \mu}. \tag{4.25}$$

**Step 6.** Eliminate $\boldsymbol{\Delta \lambda}_N$ from the second block of equations,

$$\begin{aligned} \boldsymbol{\Delta \lambda}_N = \Big(&\boldsymbol{J}_{N\boldsymbol{u}_{N,M}}^{-T}\boldsymbol{W}_{\boldsymbol{u}_{N,M}\boldsymbol{u}}\boldsymbol{J}_{\boldsymbol{u}}^{-1}\boldsymbol{J}_{\boldsymbol{\mu}} + \boldsymbol{J}_{N\boldsymbol{u}_{N,M}}^{-T}\boldsymbol{W}_{\boldsymbol{u}_{N,M}\boldsymbol{u}_{N,M}}\boldsymbol{J}_{N\boldsymbol{u}_{N,M}}^{-1}\boldsymbol{J}_{N\boldsymbol{\mu}} \\ &- \boldsymbol{J}_{N\boldsymbol{u}_{N,M}}^{-T}\boldsymbol{W}_{\boldsymbol{u}_{N,M}\boldsymbol{\mu}}\Big)\boldsymbol{\Delta \mu}. \end{aligned} \tag{4.26}$$

**Step 7.** Substitute the results from Steps 3 through 6 into the third block of (4.22) and solve for $\boldsymbol{\Delta \mu}$ to get,

$$\nabla^2\mathcal{J}\,\boldsymbol{\Delta \mu} = -\nabla\mathcal{J} \tag{4.27}$$

where $\nabla^2 \mathcal{J} \in \mathbb{R}^{P\times P}$ is a Schur complement of the linearized KKT conditions with respect to the decision variables, referred to as the *reduced Hessian*. In our problem, the reduced Hessian is given by

$$\begin{aligned}
\nabla^2 \mathcal{J} = &- \boldsymbol{W}_{\boldsymbol{\mu u}} \boldsymbol{J}_{\boldsymbol{u}}^{-1} \boldsymbol{J}_{\boldsymbol{\mu}} - \boldsymbol{W}_{\boldsymbol{\mu u}_{N,M}} \boldsymbol{J}_{N\boldsymbol{u}_{N,M}}^{-1} \boldsymbol{J}_{N\boldsymbol{\mu}} \\
&+ \boldsymbol{W}_{\boldsymbol{\mu\mu}} + \boldsymbol{J}_{\boldsymbol{\mu}}^{T} \boldsymbol{J}_{\boldsymbol{u}}^{-T} \boldsymbol{W}_{\boldsymbol{uu}} \boldsymbol{J}_{\boldsymbol{u}}^{-1} \boldsymbol{J}_{\boldsymbol{\mu}} \\
&+ \boldsymbol{J}_{\boldsymbol{\mu}}^{T} \boldsymbol{J}_{\boldsymbol{u}}^{-T} \boldsymbol{W}_{\boldsymbol{uu}_{N,M}} \boldsymbol{J}_{N\boldsymbol{u}_{N,M}}^{-1} \boldsymbol{J}_{N\boldsymbol{\mu}} \\
&- \boldsymbol{J}_{\boldsymbol{\mu}}^{T} \boldsymbol{J}_{\boldsymbol{u}}^{-T} \boldsymbol{W}_{\boldsymbol{u\mu}} + \boldsymbol{J}_{N\boldsymbol{\mu}}^{T} \boldsymbol{J}_{N\boldsymbol{u}_{N,M}}^{-T} \boldsymbol{W}_{\boldsymbol{u}_{N,M}\boldsymbol{u}} \boldsymbol{J}_{\boldsymbol{u}}^{-1} \boldsymbol{J}_{\boldsymbol{\mu}} \\
&+ \boldsymbol{J}_{N\boldsymbol{\mu}}^{T} \boldsymbol{J}_{N\boldsymbol{u}_{N,M}}^{-T} \boldsymbol{W}_{\boldsymbol{u}_{N,M}\boldsymbol{u}_{N,M}} \boldsymbol{J}_{N\boldsymbol{u}_{N,M}}^{-1} \boldsymbol{J}_{N\boldsymbol{\mu}} \\
&- \boldsymbol{J}_{N\boldsymbol{\mu}}^{T} \boldsymbol{J}_{N\boldsymbol{u}_{N,M}}^{-T} \boldsymbol{W}_{\boldsymbol{u}_{N,M}\boldsymbol{\mu}}
\end{aligned} \tag{4.28}$$

Using the reduced space method outlined in Steps 1 through 7 we are able to reduce the very large, ill-conditioned problem (4.14) and transform it into the small, well-behaved Newton step (4.27). Note that assembling the reduced Hessian using expression (4.28) is still a very expensive task. Instead of forming the reduced Hessian explicitly, it is possible to solve the reduced Newton step (4.27) by a Krylov conjugate gradients method (CG). Within every CG iteration we can then evaluate the action of the reduced Hessian on a parameter space vector in a matrix-free manner; that is, without ever assembling the reduced Hessian matrix. The cost of evaluating this matrix-free Hessian-vector product is dominated by four full-order linear system solves (four matrix factorizations if the systems of equations are solved to machine precision). This method is commonly referred to as a *reduced Newton conjugate gradient* (RNCG) method.

### 4.2.4 Computation of Hessian Blocks for PDE-Constrained Optimization Problem

In this section we present the expressions for computing the nine Hessian blocks $\boldsymbol{W}_{\boldsymbol{uu}} \in \mathbb{R}^{N\times N}$, $\boldsymbol{W}_{\boldsymbol{uu}_{N,M}} \in \mathbb{R}^{N\times N}$, $\boldsymbol{W}_{\boldsymbol{u\mu}} \in \mathbb{R}^{N\times P}$, $\boldsymbol{W}_{\boldsymbol{u}_{N,M}\boldsymbol{u}} \in \mathbb{R}^{N\times N}$, $\boldsymbol{W}_{\boldsymbol{u}_{N,M}\boldsymbol{u}_{N,M}} \in \mathbb{R}^{N\times N}$, $\boldsymbol{W}_{\boldsymbol{u}_{N,M}\boldsymbol{\mu}} \in \mathbb{R}^{N\times P}$, $\boldsymbol{W}_{\boldsymbol{\mu u}} \in \mathbb{R}^{P\times N}$, $\boldsymbol{W}_{\boldsymbol{\mu u}_{N,M}} \in \mathbb{R}^{P\times N}$ and $\boldsymbol{W}_{\boldsymbol{\mu\mu}} \in \mathbb{R}^{P\times P}$ that appear in (4.28).

$$W_{\boldsymbol{u}\boldsymbol{u}\,ij} = -L_i^O L_j^O + \sum_{n=1}^{\mathcal{N}} \frac{\partial^2 R_n}{\partial u_i \partial u_j}\lambda_n, \quad 1 \leq i,j \leq \mathcal{N} \tag{4.29}$$

$$W_{\boldsymbol{u}\boldsymbol{u}_{N,M}\,ij} = L_i^O L_{N\,j}^O \quad 1 \leq i \leq \mathcal{N}, \quad 1 \leq j \leq N \tag{4.30}$$

$$W_{\boldsymbol{u}\boldsymbol{\mu}\,ij} = \sum_{n=1}^{\mathcal{N}} \frac{\partial^2 R_n}{\partial u_i \partial \mu_j}\lambda_n \quad 1 \leq i \leq \mathcal{N}, \quad 1 \leq j \leq P \tag{4.31}$$

$$W_{\boldsymbol{u}_{N,M}\boldsymbol{u}\,ij} = W_{\boldsymbol{u}\boldsymbol{u}_{N,M}\,ji} \quad 1 \leq i \leq N, \quad 1 \leq j \leq \mathcal{N} \tag{4.32}$$

$$W_{\boldsymbol{u}_{N,M}\boldsymbol{u}_{N,M}\,ij} = -L_{N\,i}^O L_{N\,j}^O + \sum_{n=1}^{N} \frac{\partial^2 R_{Nn}}{\partial u_{N,M\,i}\, \partial u_{N,M\,j}}\lambda_{Nn} \quad 1 \leq i,j \leq N \tag{4.33}$$

$$W_{\boldsymbol{u}_{N,M}\boldsymbol{\mu}\,ij} = \sum_{n=1}^{N}\sum_{m=1}^{M} B_{N,M\,nm} \frac{\partial^2 S_{Mm}}{\partial u_{N,M\,i}\, \partial \mu_j}\lambda_{Nn} \quad 1 \leq i \leq N, \quad 1 \leq j \leq P \tag{4.34}$$

$$W_{\boldsymbol{\mu}\boldsymbol{u}\,ij} = W_{\boldsymbol{u}\boldsymbol{\mu}\,ji} \qquad 1 \leq i \leq P, \quad 1 \leq i \leq \mathcal{N} \tag{4.35}$$

$$W_{\boldsymbol{\mu}\boldsymbol{u}_{N,M}\,ij} = W_{\boldsymbol{u}_{N,M}\boldsymbol{\mu}\,ji} \quad 1 \leq i \leq P, \quad 1 \leq j \leq N \tag{4.36}$$

$$W_{\boldsymbol{\mu}\boldsymbol{\mu}\,ij} = \sum_{n=1}^{\mathcal{N}} \frac{\partial^2 R_n}{\partial \mu_i \partial \mu_j}\lambda_n + \sum_{n=1}^{N}\sum_{m=1}^{M} B_{N,M\,nm} \frac{\partial^2 S_{Mm}}{\partial \mu_i\, \partial \mu_j}\lambda_{Nn}, \quad 1 \leq i,j \leq P. \tag{4.37}$$

Here, all the terms that appear in equations (4.29) through (4.37) have already been defined in Chapters 2 and 3. The only exception is matrix $\boldsymbol{B}_{N,M} \in \mathbb{R}^{N \times M}$, which is given by $\boldsymbol{B}_{N,M} = \boldsymbol{E}_{N,M} + \boldsymbol{G}_{N,M}$, where $\boldsymbol{E}_{N,M} \in \mathbb{R}^{N \times M}$ and $\boldsymbol{G}_{N,M} \in \mathbb{R}^{N \times M}$ were defined in (3.66) and (3.67) respectively.

### 4.2.5 Different Implementations of the Solution Method for the PDE-Constrained Optimization Problem

In this section we describe three different implementations of the *reduced space* methodology described in Section 4.2.3 for solving the PDE-constrained optimization problem (4.3). The first two are based on MATLAB's Optimization Toolbox [36] whereas the third one uses the subspace trust region interior reflective Newton-CG method for bound constrained optimization problems developed in [8].

The first implementation of the optimization solver used to solve the PDE-constrained minimization problem (4.3) utilizes a sequential quadratic programming (SQP) method where a quadratic programming subproblem is solved at each iteration. An estimate of the reduced Hessian is constructed at every iteration using the formula of Broyden [7], Fletcher [17], Goldfarb [21] and Shanno [53] (BFGS). The reduced gradient information required by the BFGS formula is computed analytically using (4.21) by following the procedure outlined in Step 1 of Section 4.2.3. Since the reduced Hessian is computed approximately, only the reduced gradient is required by the optimization solver so the Steps 3 through 7 described in Section 4.2.3 are not used.

The second implementation uses a large-scale version of a subspace trust region method based on the interior-reflective Newton method described in [12], where the resulting linear system at every iteration is solved using the method of preconditioned conjugate gradients. The basic idea of a trust region method is to approximate the objective function to be minimized with a simpler function that reflects the behavior of the original function in a neighborhood of the current point in parameter space. This neighborhood is the so-called *trust region* where the approximated function is minimized. In the trust region method used in our solver, a quadratic approximation of the objective function is defined by the first two terms of its Taylor series expansion at the current point in parameter space. The trust region subproblem is then given by

$$\min_{\boldsymbol{\Delta\mu}\in\mathbb{R}^P} \left\{ \frac{1}{2}(\boldsymbol{\Delta\mu})^T \cdot \nabla^2 \mathcal{J} \cdot \boldsymbol{\Delta\mu} + (\boldsymbol{\Delta\mu})^T \nabla \mathcal{J} \text{ such that } \|\boldsymbol{D}\boldsymbol{\Delta\mu}\| \leq \boldsymbol{\Upsilon} \right\} \tag{4.38}$$

where $\boldsymbol{\Delta\mu}$ is the updating step for the next iteration; $\nabla\mathcal{J}$ is the reduced gradient at the current point $\boldsymbol{\mu}$ computed using (4.21); $\nabla^2\mathcal{J}$ is the reduced Hessian given by (4.28), $\boldsymbol{D}$ is a diagonal scaling matrix and $\boldsymbol{\Upsilon}$ is a positive scalar that gives the current radius of the trust region. In order to avoid the inversion of the reduced Hessian for solving (4.38), the trust region subproblem is restricted to a two-dimensional subspace determined by using a conjugate gradient process. Therefore, only Hessian-vector products are needed at every iteration instead of the full Hessian. Computing the action of the reduced Hessian on a vector has the dominant cost of only four linearized full order solves as described in Section 4.2.3, which is clearly much cheaper than computing the actual Hessian according to (4.28).

The third and final optimization solver tested in this thesis uses the subspace trust region Newton solver STIRNCG described in [8].

Finally, it should be noted that all the methods described in this section require an initial guess for finding the next sample point at every greedy cycle. Since the objective function of the greedy problem — i.e., the output error between the reduced and "truth" approximations — usually develops several local maxima as the greedy algorithm progresses, the location of the initial guess usually determines which of the local optima will be found by the optimization solver. Furthermore, if the initial guess is close to an existing sample point, the value of the objective function and its curvature will be very small and the solver may not be able to make any progress towards finding an optimum. Therefore, the quality of the initial guesses is critical for finding a good sampling point for generating each new snapshot. Developing an efficient algorithm for computing initial guesses is a challenging task in its own right and can be very problem-dependent.

The method used in this thesis for obtaining a suitable initial guess at every greedy cycle is to generate random points until one is found that is sufficiently far from existing sample points and where the gradient is significantly greater than the tolerance specified for the optimization solver. Random initial guesses are preferably located on the boundary of the parameter space $\mathcal{D}$, especially during the first greedy cycles. Since interpolating is generally more efficient than extrapolating, most adaptive sampling algorithms commonly used for generating efficient reduced basis tend to sample the corners and boundaries of the parameter space. Therefore, it is usually a reasonable idea to exploit this fact for generating initial guesses. Finally, we note that identical initial guesses were used for testing the three optimization solvers described in this section. Therefore, the differences in performance shown in Section 4.2.6 are only due to the algorithms themselves and not to the quality of the initial guesses.

### 4.2.6 Comparison of Greedy Sampling Versus Grid Sampling

In this section we present the reduced basis approximation built using the continuous version of the greedy sampling algorithm introduced in Section 4.2.1 applied to the 2-parameter combustion problem that was presented in Chapter 2. The critical step of finding the next sample location at every greedy cycle is formulated using the PDE-constrained minimization problem (4.3) and solved using the three optimization solvers discussed in Section 4.2.5.

The results of the greedy sampling method are compared against the results provided by a typical grid sampling approach based on the Latin hypercube sampling methodology [37]. In this case we use the 14 × 14 regular grid $\Xi_{196}$ over $\mathcal{D}$ that was introduced in Section 3.5.3. The snapshots are added sequentially by running 6 Latin hypercube cycles on $\Xi_{196}$. At every Latin hypercube cycle 14 random samples are added such that there is only one new sample in each row and each column of $\Xi_{196}$. After the first 6 Latin hypercube cycles, the remaining 112 sample points in $\Xi_{196}$ are added using purely random sampling. Figure 4-1 shows the location of the first 56 samples which correspond to the first 4 Latin hypercube cycles.

The performance of the four different sampling methods tested in this section is shown in Figure 4-2. The performance is measured in terms of the maximum relative error of the reduced basis approximation, $\varepsilon^o_{N,M\,\max,\mathrm{rel}}$, as a function of the number of nonlinear full-order solves at each greedy cycle. Note that one nonlinear full-order solve involves solving the nonlinear system of equations (4.17). The maximum relative error at any greedy cycle is given by

$$\varepsilon^o_{N,M\,\max,\mathrm{rel}} = \max_{\boldsymbol{\mu}\in\Xi^{\mathrm{test}}_{529}} \frac{\|o(\boldsymbol{\mu}) - o_{N,M}(\boldsymbol{\mu})\|}{\|o(\boldsymbol{\mu})\|} \tag{4.39}$$

Note that the maximum relative error is computed over a 23 × 23 uniform grid, $\Xi^{\mathrm{test}}_{529}$, which is finer and mostly non-coincident with the grid used for the grid-based sampling. The performance of the different algorithms is measured in terms of the number of full-order solves required to find the sample point at each greedy cycle because this is the dominant computational cost of the greedy algorithm. Note that the

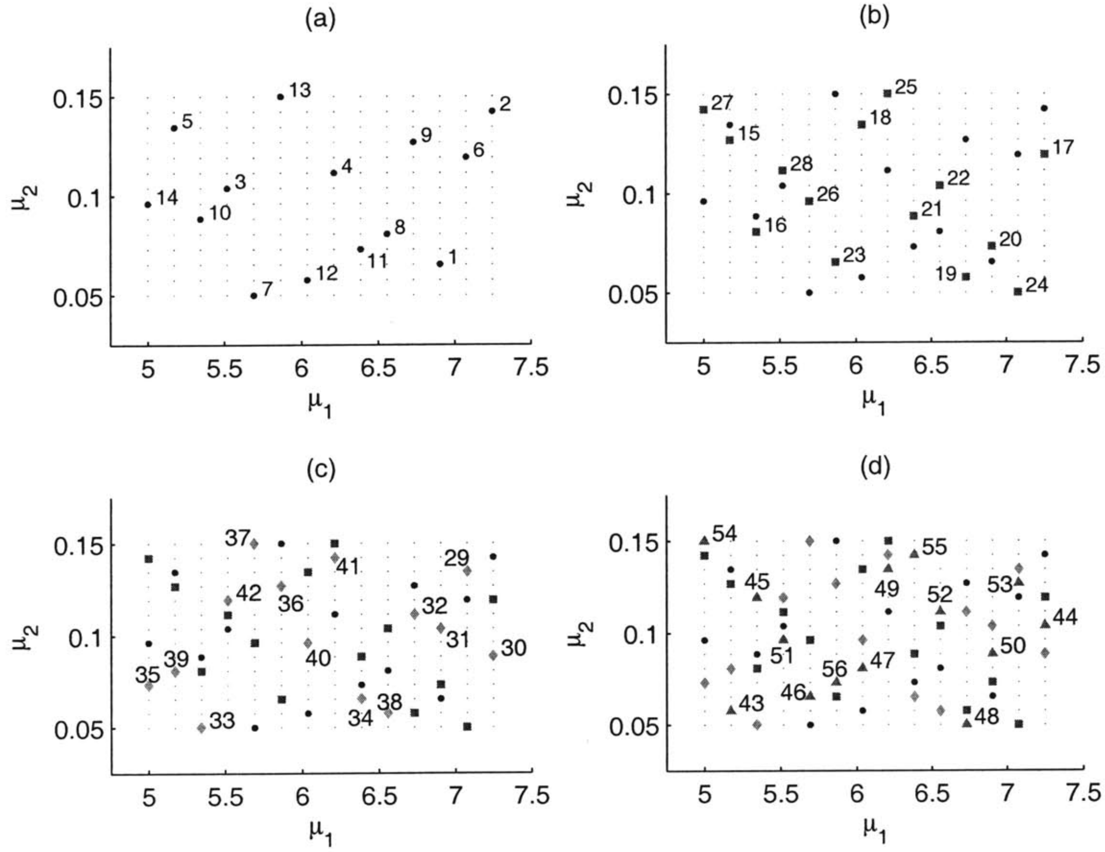

Figure 4-1: Location of the first 56 sample points using Latin hypercube sampling on a regular 14 × 14 grid. Each of the four plots corresponds to a different Latin hypercube cycle where 14 sample points are added such that there is only one new point in each row and each column of the grid. In this figure the parameter vector is given by $(\mu_1, \mu_2) = (\log(A), E)$.

cost of assembling the Hessian blocks and computing the Hessian-vector products required by the subspace trust region Newton solvers is significantly less than the cost of performing each nonlinear full-order solve so it has not been taken into account when evaluating the computational performance of the large-scale algorithms. Similarly, the cost of approximating the Hessian using the BFGS formula in the medium-scale SPQ algorithm has also been neglected.

Since the grid-based sampling selects the sample points randomly, it only requires one full-order solve per sampling iteration. Therefore, the full grid is covered in 196 solves. The three PDE-constrained optimization algorithms require several solves at each greedy cycle in order to arrive at a local optimum. The subspace trust region method based on [36] (labeled "STIRNCG-1" in Figure 4-2) shows the most efficient performance of the three optimization algorithms, requiring a total of 676 solves in order to achieve the same maximum relative error as the grid sampling over $\Xi_{196}$. As expected, the SQP algorithm (labeled "SQP" in Figure 4-2) yields the worst performance since it only uses an approximation of the reduced Hessian instead of the exact reduced Hessian for solving the optimization problem (4.3). The performance of the subspace trust region algorithm described in [8] (labeled "STIRNCG-2" in Figure 4-2) is in between the previous two. This is probably due to the fact that this algorithm was originally designed for tackling linear problems so its efficiency drops when used for solving nonlinear problems where it is not possible to reuse the matrix factorizations required for performing the different full-order solves.

We must point out that the grid-based sampling achieved the desired error by sampling 196 grid points whereas the PDE-constrained optimization methods only required 50 snapshots in order to obtain the same level of accuracy in the reduced basis approximation. This, however, is not a major advantage since the snapshots are compressed using POD prior to generating the reduced basis vectors, therefore yielding reduced order models of very similar sizes.

The locations of the 50 sample points computed using the first subspace trust region Newton method are shown in Figure 4-3. As expected, most of the sample points are located on the boundaries of the parameter space $\mathcal{D}$, with a higher concentration

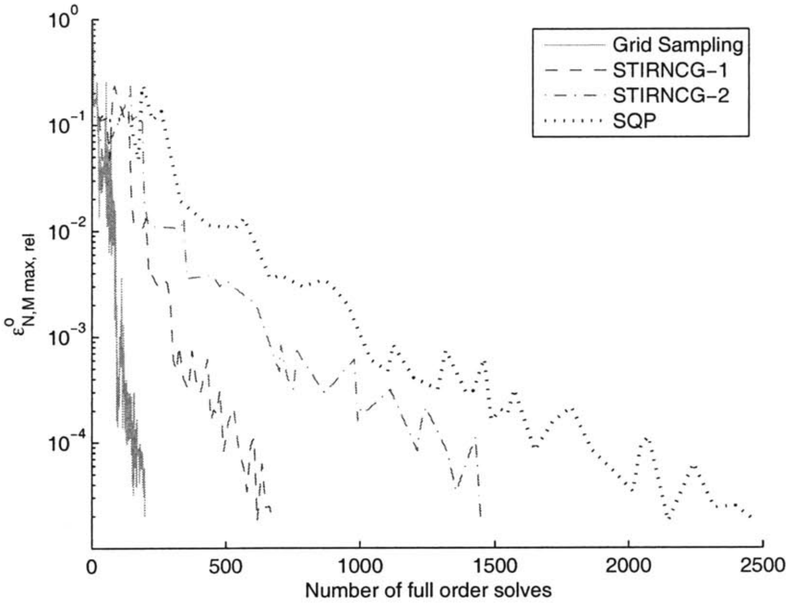

Figure 4-2: Performance of the four different sampling methods in terms of the number of nonlinear full-order solves required to find the next sampling point at each greedy cycle.

in the region of higher nonlinearity (large values of $A$ and small values of $E$).

### 4.2.7 *A Posteriori* Error Estimation

For a problem with only two parameters such as the one considered in this section, it is relatively inexpensive to cover the entire parameter space using a grid of sample points. Therefore, grid-based sampling methods are computationally cheaper than PDE-constrained optimization methods as shown in Section 4.2.6. However, even for low-dimensional parametric problems, the PDE-constrained optimization approach offers a significant advantage over random-sampling, grid-based methods due to the fact that it gives an error estimator for the reduced basis approximation at every greedy cycle.

It has been shown throughout this thesis that the accuracy of the reduced basis approximation depends greatly on the number of basis functions and interpolation functions used to approximate the solution space and the nonlinear term respectively.

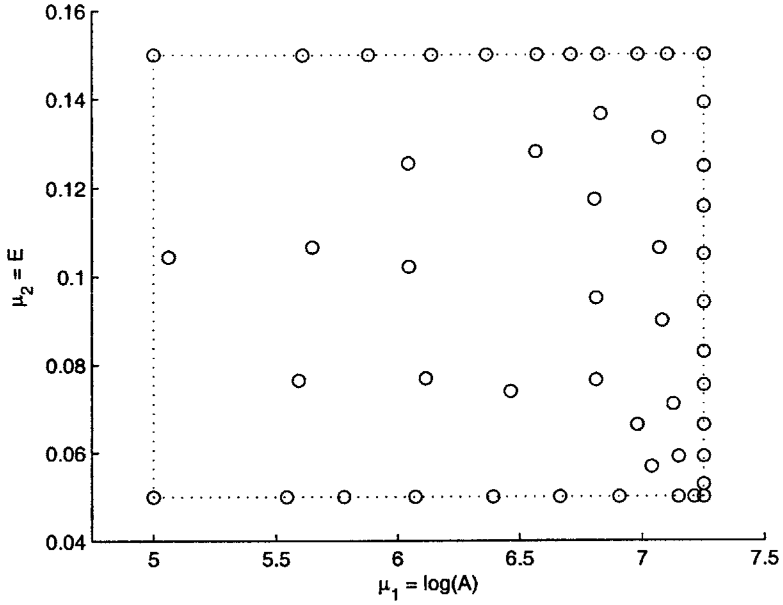

Figure 4-3: Location of the 50 sample points computed using the first subspace trust region Newton solver (STIRNCG-1).

Therefore, for any given application where a predetermined level of accuracy is required, it is critical to be able to compute an error indicator that will tell us when the desired accuracy has been reached. Otherwise, it is impossible to determine whether the dimension of the reduced basis spaces $N$ and $M$ are adequate for our needs. If $N$ and $M$ are too small, the reduced basis error may be too large, which could potentially have catastrophic effects when using the results of the reduced basis approximation in a critical application. On the other hand, if $N$ and $M$ are too large, the reduced basis results will be unnecessarily expensive to compute, therefore compromising the real-time deployment of the the reduced order model.

It is clear that an error estimator is a critical ingredient for obtaining an efficient reduced basis approximation that meets our accuracy requirements while minimizing the size of the approximation spaces. Random grid-based sampling methods like the Latin hypercube method introduced in Section 4.2.6 do not provide any kind of error estimator so the stopping criterion for adding snapshots can only be based on intuition, heuristics or simply on the computational resources available for generating snapshots. Clearly, this is not a satisfactory state of affairs when trying to construct an efficient reduced order model.

The grid-based greedy sampling methods developed by Patera and co-workers [23, 24, 25, 41, 44, 47] do include a rigorous error bound such that the ratio of this error bound over the actual error is always greater than unity for all $\boldsymbol{\mu}$ in $\mathcal{D}$. However, it has already been mentioned that it is not possible to derive an $\mathcal{N}$-independent error bound for general nonlinear problems so this method becomes intractable for high-dimensional nonlinear problems and is not considered here.

Here we propose the use of the error value computed by the PDE-constrained optimization solver as a viable error estimator for estimating the accuracy of the reduced order model at each greedy cycle. Note that this error estimator is not a rigorous error bound due to the fact that the optimizer is not guaranteed to converge to the global maximum. Figure 4-4 shows an example of how the optimization solver can converge to a point that is not a global maximum of the objective function. Since the objective function is non-convex, the maximum error found by the PDE-constrained minimization algorithm is only guaranteed to be locally optimal but not necessarily a global optimum over the entire parameter domain. Therefore, it is not possible to certify that the reduced basis output error will be less than the error estimator for all $\boldsymbol{\mu}$ in $\mathcal{D}$. However, despite the lower standards of certainty associated with this error estimator as opposed to a rigorous error bound, we believe it is still a useful tool for obtaining an accurate order of magnitude of the reduced basis error at each greedy cycle.

To quantify the merit of the error indicator proposed in this section we define its effectivity as follows,

$$\eta_k(\boldsymbol{\mu}) = \frac{\Lambda_k(\boldsymbol{\mu}^*)}{\max\limits_{\boldsymbol{\mu}\in\Xi^{\text{test}}} \|\boldsymbol{o}(\boldsymbol{\mu}) - \boldsymbol{o}_{N,M}^{k-1}(\boldsymbol{\mu})\|} \tag{4.40}$$

where $k$ gives the current greedy cycle, $\boldsymbol{o}_{N,M}^{k-1}(\boldsymbol{\mu})$ is the reduced basis output at point $\boldsymbol{\mu}$ using the reduced order model built in cycle $k-1$, $\boldsymbol{\mu}^*$ is the value of the sample

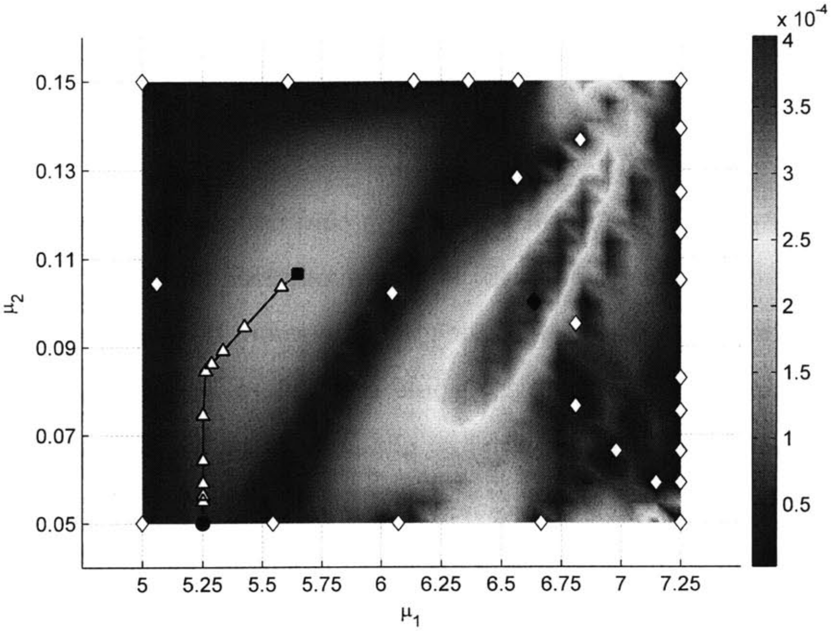

Figure 4-4: Example of optimization algorithm converging to a local optimum instead of the global optimum. Plot corresponds to greedy cycle $k = 28$ using the first subspace trust region Newton solver (STIRNCG-1). The 27 snapshots added prior to the current cycle are indicated by diamonds ($\Diamond$). The location of the initial guess ($\bullet$), local maximum found by optimizer (■) and maximum error over $\Xi_{529}^{\text{test}}$ (♦) are also shown on this plot. The intermediate steps computed by the optimizer in order to get from the initial guess to the local optimum are indicated by triangles ($\triangle$). Note that the colormap gives the value of the relative output error at every point in $\Xi_{529}^{\text{test}}$, not the value of the objective function $\mathcal{J}$.

point computed by the PDE-constrained optimization solver; i.e.,

$$\begin{aligned} \boldsymbol{\mu}^* = \arg\min_{\boldsymbol{\mu}} & \mathcal{J}(\boldsymbol{u}(\boldsymbol{\mu}), \boldsymbol{u}_{N,M}(\boldsymbol{\mu})) \\ \text{subject to } & \boldsymbol{R}_N(\boldsymbol{u}_{N,M}; \boldsymbol{\mu}) = 0 \\ & \boldsymbol{R}(\boldsymbol{u}; \boldsymbol{\mu}) = 0 \\ & \boldsymbol{\mu}_{\min} \leq \boldsymbol{\mu} \leq \boldsymbol{\mu}_{\max}, \end{aligned} \tag{4.41}$$

and $\Lambda_k(\boldsymbol{\mu}^*)$ is the value of the output error between the "truth" solution and reduced basis approximation computed by the optimization solver at greedy cycle $k$ (this is in fact our proposed error estimator). $\Lambda_k(\boldsymbol{\mu}^*)$ is trivially related to the value of the objective function at sample point $k$ by the following equation:

$$\Lambda_k(\boldsymbol{\mu}^*) = \sqrt{2\,\mathcal{J}(\boldsymbol{u}(\boldsymbol{\mu}^*), \boldsymbol{u}_{N,M}(\boldsymbol{\mu}^*))}. \tag{4.42}$$

Note that $\Lambda_k(\boldsymbol{\mu}^*)$ is the result of solving a continuous optimization problem over $\mathcal{D}$ whereas the denominator of (4.40) is only maximized over $\Xi^{\text{test}}$. Therefore, when the optimization solver converges to the global optimum the effectivity will be greater than or equal to one (the equal case would correspond to the unlikely case where a point in $\Xi^{\text{test}}$ coincides exactly with the global maximum of the continuous output error function).

From (4.40), it is apparent that a necessary condition for having a rigorous error indicator at greedy iteration $k$ is given by $\eta_k(\boldsymbol{\mu}) \geq 1$, $\forall \boldsymbol{\mu} \in \Xi^{\text{test}}$ (the sufficient condition is given by $\eta_k(\boldsymbol{\mu}) \geq 1$, $\forall \boldsymbol{\mu} \in \mathcal{D}$). Since the objective function is non-convex neither the necessary nor the sufficient conditions can be achieved in general. Figure 4-4 provides a clear example of a greedy iteration where the effectivity is less than unity. In particular, for the case shown in Figure 4-4 we have that $\Lambda_{28}(\boldsymbol{\mu}^*) = 1.74 \cdot 10^{-4}$ whereas $\max_{\boldsymbol{\mu}\in\Xi_{529}^{\text{test}}} \|\boldsymbol{o}(\boldsymbol{\mu}) - \boldsymbol{o}_{N,M}^{27}(\boldsymbol{\mu})\| = 4.06 \cdot 10^{-4}$, which yields an efficiency of $\eta_{27}(\boldsymbol{\mu}) = 0.43 < 1$.

Figure 4-5 shows the effectivity at every greedy iteration when using the subspace trust region Newton solver STIRNCG-1. Since the test grid used in this case, $\Xi_{529}^{\text{test}}$, is sufficiently fine, the necessary condition for rigor of the error indicator is also sufficient so all points with effectivities greater than or equal to unity correspond to greedy iterations where the optimization solver converged to the global maximum. Conversely, effectivities below one correspond to greedy cycles where the sample point did not match the global optimum of the objective function. Note that in all cases the effectivity is greater than 0.2, which means that using the proposed error indicator for estimating the reduced basis error would "only" underestimate the actual error by a factor of approximately 5. Furthermore, in most cases the effectivity is significantly closer to, or even greater than, unity which supports the use of the quantity $\Lambda(\boldsymbol{\mu}^*)$ as a valid error indicator.

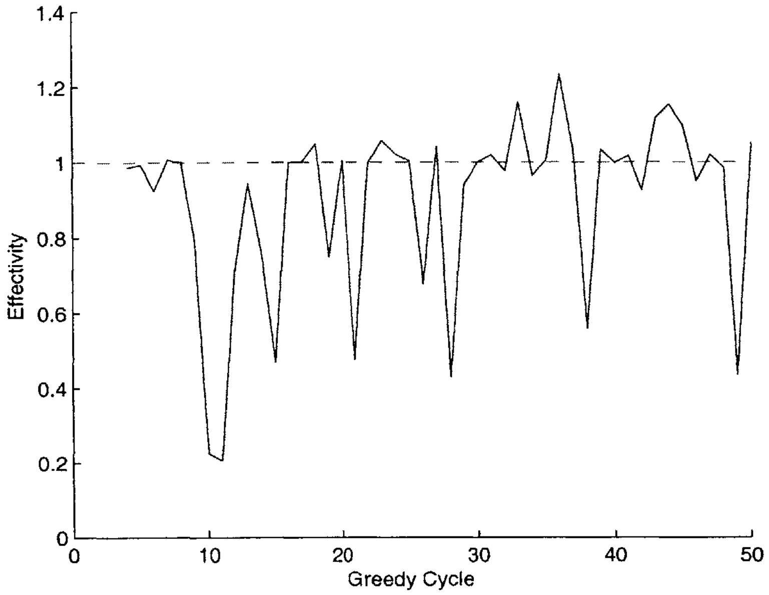

Figure 4-5: Effectivity of error indicator at each greedy iteration when using the subspace trust region Newton solver STIRNCG-1. All effectivity values equal to or greater than unity indicate that the optimum found by the solver is the global maximum. Effectivities less than unity indicate convergence to a local maximum.

## 4.3 Application of Greedy Sampling to 3-Parameter Case

In this section we apply the PDE-constrained greedy sampling methodology developed earlier to a combustion problem that is parametrized in terms of 3 parameters instead of the usual 2 — $A$ and $E$ — that have been used throughout this thesis. We now introduce the diffusivity $\kappa$ as a third parameter in our model such that the parameter vector $\boldsymbol{\mu} \in \mathbb{R}^3$ is given by $(\mu_1, \mu_2, \mu_3) = (\ln(A), E, \kappa)$. The system parameters can vary within the domain $\mathcal{D} \equiv [5.00, 0.05, 5.5 \cdot 10^{-6}] \times [6.90, 0.15, 2.0 \cdot 10^{-5}] \subset \mathbb{R}^3$. The governing PDE for the combustion reaction is the same equation (2.1) that was introduced in Chapter 2.

In this section we show that even for a relatively small number of parameters ($P = 3$), the PDE-constrained optimization greedy sampling offers comparable performance to grid-based sampling methods while maintaining the advantage of providing a valid error indicator for estimating the accuracy of the resulting reduced basis approximation at every greedy cycle.

### 4.3.1 Problem Formulation

The "truth" approximation in this case is formulated as follows: given any $\boldsymbol{\mu} \in \mathcal{D}$, evaluate

$$o_i(\boldsymbol{\mu}) = \ell_i^O(u(\boldsymbol{\mu})), \quad i = 1, \ldots, N_s, \tag{4.43}$$

where $u(\boldsymbol{\mu}) \in X^D$ is the solution of

$$\begin{aligned}&\sum_{q=1}^{2} \Theta_q(\boldsymbol{\mu}) a_q(u(\boldsymbol{\mu}), v) + \int_\Omega s(u(\boldsymbol{\mu}); \boldsymbol{\mu})\, v \, d\Omega \\ &+ \Theta_3(\boldsymbol{\mu}) \sum_{k=1}^{N_{\text{elem}}} \int_{T_h^k} (\boldsymbol{U} \cdot \nabla v) \ (\boldsymbol{U} \cdot \nabla u(\boldsymbol{\mu}) + s(u(\boldsymbol{\mu}); \boldsymbol{\mu}) - f) \ d\Omega = \ell(v), \quad \forall v \in X.\end{aligned} \tag{4.44}$$

where $\Theta_q : \mathcal{D} \to \mathbb{R},\ 1 \le q \le 3$ are parameter-dependent functions given by

$$\Theta_1(\boldsymbol{\mu}) = 1, \tag{4.45}$$

$$\Theta_2(\boldsymbol{\mu}) = \kappa, \tag{4.46}$$

$$\Theta_3(\boldsymbol{\mu}) = \frac{h}{2\|\boldsymbol{U}\|} \left( \coth\left( \frac{\|\boldsymbol{U}\| h}{2\kappa} \right) - \frac{2\kappa}{\|\boldsymbol{U}\| h} \right) \tag{4.47}$$

and

$$a_1(w, v) = \int_\Omega v\, \boldsymbol{U} \cdot \nabla w \, d\Omega \tag{4.48}$$

$$a_2(w, v) = \int_\Omega \nabla w \cdot \nabla v \, d\Omega, \tag{4.49}$$

$$\ell(v) = \int_\Omega v f \, d\Omega \tag{4.50}$$

$$\ell_i^O(v) = \int_{\Gamma_i} v \, d\Gamma, \quad i = 1, \ldots, N_s. \tag{4.51}$$

are parameter-independent bilinear and linear forms.

It is worth noting that the stabilization term in (4.44) is not general for two reasons: First, the definition by Brooks and Hughes has been used for computing the stabilization parameter (refer to Section 2.3.3 for details). Second, and more importantly, the stabilization parameter has been assumed constant throughout the entire domain $\Omega$. The latter is a severe limitation, since the SUPG stabilization parameter $\tau$ is, in general, different for every element in the mesh. The assumption of constant $\tau$ is only strictly valid for meshes of identical element sizes and problems with constant flow velocity $\boldsymbol{U}$. This is not exactly our case but we still make this assumption in order to simplify the "truth" formulation and the subsequent reduced basis approximation. If a general, element-dependent, formulation of the SUPG stabilization parameter is used, then the stabilization term in (4.44) becomes *nonaffine* in $\kappa$; that is, the parametrized differential operator can no longer be expressed as a sum of products of parameter-dependent functions and parameter-independent operators.

The nonaffine dependence of the stabilization term on $\kappa$ can be treated by developing a coefficient-function approximation following an analogous procedure to the one used in Chapter 3 for treating the nonlinear term $s(u(\boldsymbol{\mu});\, \boldsymbol{\mu})$. However, the purpose of this section is not to develop a perfect "truth" approximation but to illustrate the application of our PDE-constrained greedy algorithm to a high-dimensional problem. Therefore, we shall assume that the use of a constant stabilization parameter that allows the affine decomposition shown in (4.44) yields a "truth" approximation that is adequate for our accuracy requirements (we are admittedly using the word "truth" in a very flexible way here).

The weak form (4.44) yields the following nonlinear system of equations:

$$\boldsymbol{A}\boldsymbol{u}(\boldsymbol{\mu}) + \boldsymbol{S}\left(\boldsymbol{u}(\boldsymbol{\mu}); \boldsymbol{\mu}\right) = \boldsymbol{F}, \tag{4.52}$$

which is identical to the system of equations (2.35) derived in Chapter 2 except for the fact that in this case the stiffness matrix, $\boldsymbol{A} \in \mathbb{R}^{\mathcal{N}\times\mathcal{N}}$ admits the following affine decomposition for $Q = 3$:

$$\boldsymbol{A} = \sum_{q=1}^{Q} \Theta_q(\boldsymbol{\mu}) \boldsymbol{A}_q, \tag{4.53}$$

where

$$A_{1\,ij} = a_1(\varphi_j, \varphi_i), \qquad 1 \le i, j \le \mathcal{N} \tag{4.54}$$

$$A_{2\,ij} = a_2(\varphi_j, \varphi_i), \qquad 1 \le i, j \le \mathcal{N} \tag{4.55}$$

$$A_{3\,ij} = \sum_{k=1}^{N_{\text{elem}}} \int_{T_h^k} (\boldsymbol{U} \cdot \nabla \varphi_i)\ (\boldsymbol{U} \cdot \nabla \varphi_j)\ d\Omega \qquad 1 \le i, j \le \mathcal{N} \tag{4.56}$$

The load vector $\boldsymbol{F} \in \mathbb{R}^{\mathcal{N}}$ and nonlinear reaction vector, $\boldsymbol{S}\left(\boldsymbol{u}(\boldsymbol{\mu}); \boldsymbol{\mu}\right) \in \mathbb{R}^{\mathcal{N}}$ are identical to those defined in Section 2.4 with the only difference that the SUPG stabilization parameter is now given by $\Theta_3(\boldsymbol{\mu})$, which eliminates the element-dependence and, therefore, the nonaffine dependence on $\mu_3$.

The method for solving (4.52) is identical to that described in Chapter 2 for the 2-parameter case. Figure 4-6 shows the "truth" solutions computed at the eight corners of the parameter space $\mathcal{D}$. As in the 2-parameter case, it can be observed that the parameter vector $\boldsymbol{\mu}$ has a very significant effect on the behavior of the solution.

### 4.3.2 Reduced Basis Approximation

The method for constructing the reduced basis approximation for the 3-parameter problem is analogous to that used in Chapter 3 for the 2-parameter case. Using the same coefficient-function approximation $s_M^{u_{N,M}}(\boldsymbol{x}; \boldsymbol{\mu})$ for expanding the nonlinear term in series of the interpolation functions allows us to formulate the reduced basis approximation as follows: given any $\boldsymbol{\mu} \in \mathcal{D}$, evaluate

$$o_{N,M\,i}(\boldsymbol{\mu}) = \ell_i^O(u_{N,M}(\boldsymbol{\mu})), \quad i = 1, \ldots, N_s, \tag{4.57}$$

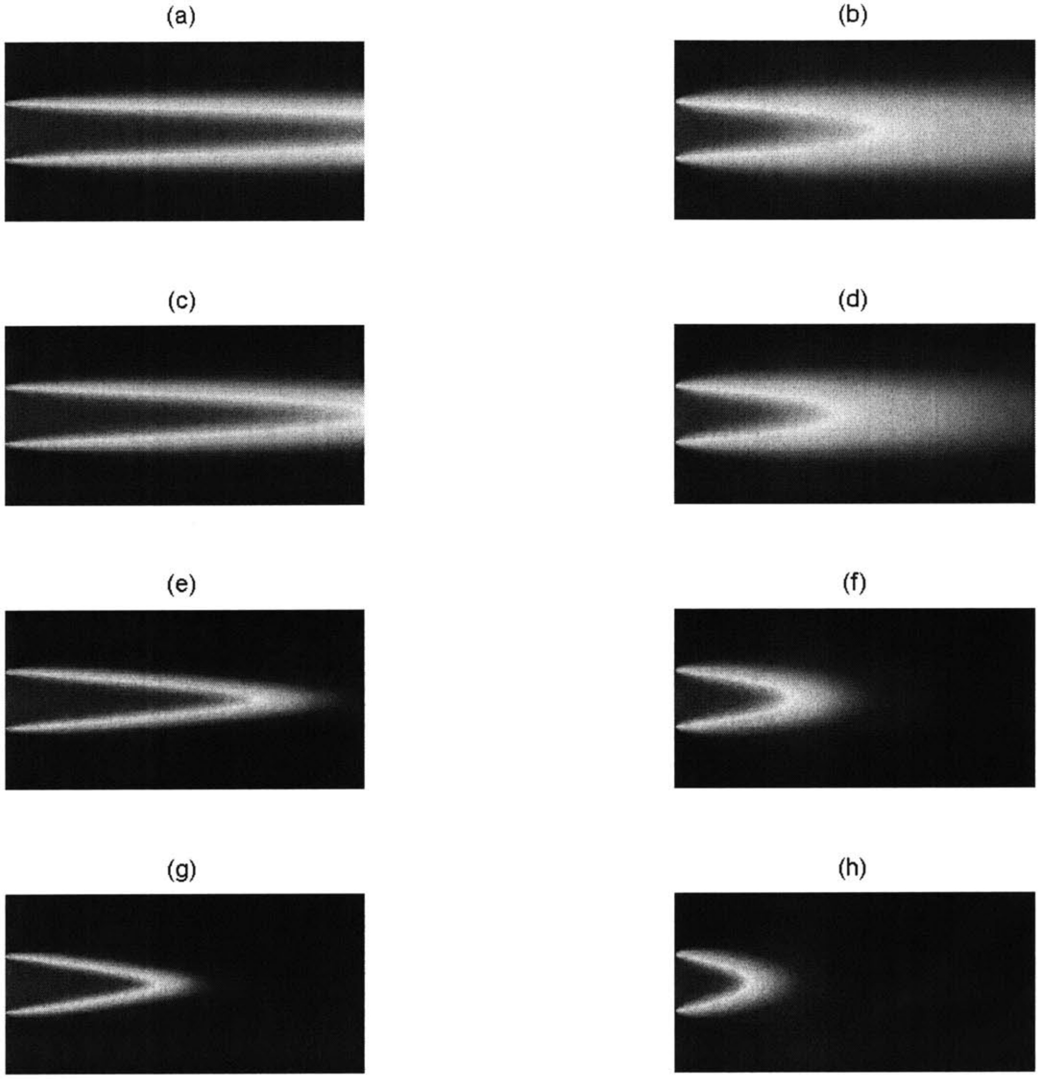

Figure 4-6: "Truth" solutions computed at the eight corners of the parameter space $\mathcal{D}$. As usual, blue (outside region) corresponds to $u = 0$ whereas red (center region) corresponds to $u = c$. The values of the parameter vector at each of the 8 corners are: $\boldsymbol{\mu}_a = (5.0, 0.15, 5.5 \cdot 10^{-6})$, $\boldsymbol{\mu}_b = (5.0, 0.15, 2.0 \cdot 10^{-5})$, $\boldsymbol{\mu}_c = (5.0, 0.05, 5.5 \cdot 10^{-6})$, $\boldsymbol{\mu}_d = (5.0, 0.05, 2.0 \cdot 10^{-5})$, $\boldsymbol{\mu}_e = (6.9, 0.15, 5.5 \cdot 10^{-6})$, $\boldsymbol{\mu}_f = (6.9, 0.15, 2.0 \cdot 10^{-5})$, $\boldsymbol{\mu}_g = (6.9, 0.05, 5.5 \cdot 10^{-6})$, $\boldsymbol{\mu}_h = (6.9, 0.05, 2.0 \cdot 10^{-5})$.

where $u_{N,M}(\boldsymbol{\mu}) \in W_N^D$ is the solution of

$$\begin{aligned}
&\sum_{q=1}^{2}\Theta_q(\boldsymbol{\mu})a_q(\bar{u},v) + \sum_{q=1}^{2}\Theta_q(\boldsymbol{\mu})a_q(u_{N,M}(\boldsymbol{\mu}),v) + \int_\Omega s_M^{u_{N,M}}(\boldsymbol{x};\boldsymbol{\mu})\, v\, d\Omega \\
&+\Theta_3(\boldsymbol{\mu}) \sum_{k=1}^{N_\text{elem}} \int_{T_h^k} (\boldsymbol{U}\cdot\nabla v)\ (\boldsymbol{U}\cdot\nabla u_{N,M}(\boldsymbol{\mu}) + \boldsymbol{U}\cdot\nabla\bar{u} - f)\ d\Omega \\
&+\Theta_3(\boldsymbol{\mu}) \sum_{k=1}^{N_\text{elem}} \int_{T_h^k} (\boldsymbol{U}\cdot\nabla v)\ s_M^{u_{N,M}}(\boldsymbol{x};\boldsymbol{\mu})\, d\Omega = \ell(v), \quad \forall v \in W_N.
\end{aligned} \tag{4.58}$$

where the series expansion of the reduced basis approximation is given by

$$u_{N,M}(\boldsymbol{x};\boldsymbol{\mu}) = \bar{u}(\boldsymbol{x}) + \sum_{n=1}^{N} u_{N,M\,n}(\boldsymbol{\mu})\, \zeta_n(\boldsymbol{x}) \tag{4.59}$$

and the coefficient-function approximation is given by

$$s_M^{u_{N,M}}(\boldsymbol{x};\boldsymbol{\mu}) = \sum_{m=1}^{M} s\left(\bar{u}(\boldsymbol{z}_m) + \sum_{n=1}^{N} u_{N,M\,n}(\boldsymbol{\mu})\, \zeta_n(\boldsymbol{z}_m); \boldsymbol{\mu}\right)\ \psi_m(\boldsymbol{x}) \tag{4.60}$$

Choosing $v = \zeta_n$, $1 \le n \le N$ and substituting (4.59) and (4.60) into (4.58) yields

$$\boldsymbol{A}_0 + \boldsymbol{A}_N\, \boldsymbol{u}_{N,M} + (\boldsymbol{E}_{N,M} + \Theta_3(\boldsymbol{\mu})\, \boldsymbol{G}_{N,M})\ s(\bar{\boldsymbol{u}}_M + \boldsymbol{D}_{M,N}\, \boldsymbol{u}_{N,M};\ \boldsymbol{\mu}) = \boldsymbol{F}_N \tag{4.61}$$

where $\boldsymbol{E}_{N,M} \in \mathbb{R}^{N\times M}$, $\boldsymbol{G}_{N,M} \in \mathbb{R}^{N\times M}$, $\bar{\boldsymbol{u}}_M \in \mathbb{R}^M$, $\boldsymbol{D}_{M,N} \in \mathbb{R}^{M\times N}$ and $\boldsymbol{F}_N \in \mathbb{R}^N$ were defined in Section 3.6 (with the only difference that in this case $\tau_k \equiv 1$, $\forall\, k = 1,\ldots,N_\text{elem}$ because the stabilization parameter is defined globally in terms of $\Theta_3(\boldsymbol{\mu})$). Here $\boldsymbol{A}_0 \in \mathbb{R}^N$ and $\boldsymbol{A}_N \in \mathbb{R}^{N\times N}$, are given by

$$\boldsymbol{A}_0 = \sum_{q=1}^{Q} \Theta_q(\boldsymbol{\mu})\boldsymbol{A}_{0q}, \tag{4.62}$$

$$\boldsymbol{A}_N = \sum_{q=1}^{Q} \Theta_q(\boldsymbol{\mu})\boldsymbol{A}_{Nq}, \tag{4.63}$$

where

$$\boldsymbol{A}_{0q} = \boldsymbol{Z}^T \boldsymbol{A}_q \bar{\boldsymbol{u}}, \quad 1 \leq q \leq Q \tag{4.64}$$

$$\boldsymbol{A}_{Nq} = \boldsymbol{Z}^T \boldsymbol{A}_q \boldsymbol{Z}, \quad 1 \leq q \leq Q. \tag{4.65}$$

The offline-online decomposition for the 3-parameter case is the same as the one shown in Figures 3-5 and 3-6 but in this case the parameter-independent matrices that must be computed in the offline phase are $\boldsymbol{A}_{0q}$, $\boldsymbol{A}_{Nq}$, $\boldsymbol{E}_{N,M}$, $\boldsymbol{G}_{N,M}$, $\boldsymbol{D}_{M,N}$, $\bar{\boldsymbol{u}}_M$, $\boldsymbol{F}_N$ and $\boldsymbol{L}_N^0$, $1 \leq q \leq Q$ whereas $\boldsymbol{A}_0$, $\boldsymbol{A}_N$ are formed in the online stage using (4.62) and (4.63).

### 4.3.3 PDE-Constrained Greedy Sampling

The greedy sampling method for the 3-parameter case is identical to that presented in Section 4.2 for the 2-parameter case. In this case we use the the first subspace trust region Newton solver presented in 4.2.5 as the optimization solver of choice since it provided the best results of the three solvers that were tested for the 2-parameter case.

The samples selected by the PDE-constrained greedy sampling algorithm after 75 greedy iterations are shown in Figure 4-7. Note that most of the sample points are located on the boundaries of the parameter domain, especially in the region of highest nonlinearity (low diffusivity, high $A$ and low $E$).

For the Latin hypercube grid-based sampling we define a uniform 12 × 12 × 12 grid $\Xi_{1728}^{\text{train}}$ over $\mathcal{D}$. Initially, sample points are selected using 6 Latin hypercube cycles over $\Xi_{1728}^{\text{train}}$ and the remaining points are sampled using purely random sampling following the same method used for the 2-parameter case.

The reduced order models built using the Latin hypercube grid-based sampling and the PDE-constrained greedy sampling are tested on a uniform 17 × 17 × 17 grid $\Xi_{4913}^{\text{test}}$ which is finer and mostly non-coincident with $\Xi_{1728}^{\text{train}}$. Figure 4-8 shows the maximum relative error of both reduced order models as a function of the number of full-order solves required to build the models. If errors below $10^{-4}$ are required in

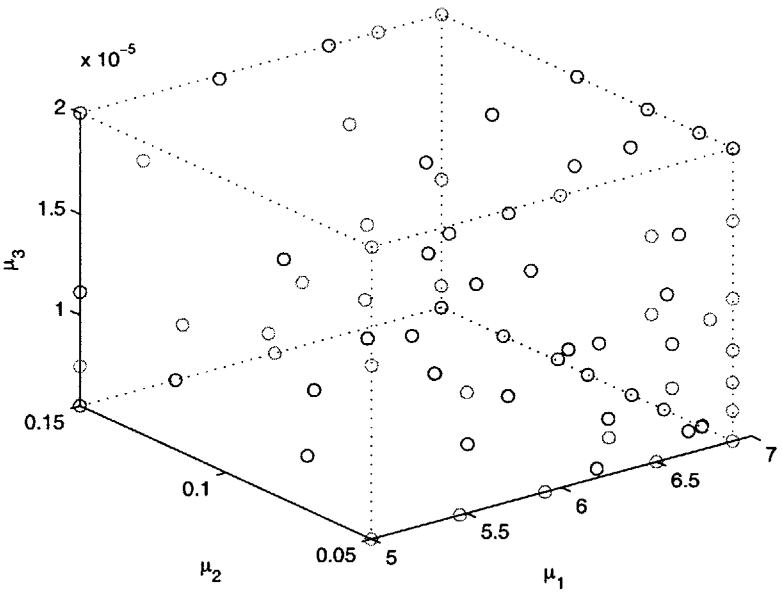

Figure 4-7: Location of the first 75 sample points computed using the PDE-constrained greedy sampling algorithm in the 3-parameter case. Points are colored based on their location: black is used for interior points, red indicates that point is located on the $A = A_{\max}$ or $A = A_{\min}$ surfaces, blue is used for points on the surfaces given by $\kappa = \kappa_{\max}$ or $\kappa = \kappa_{\min}$ and green is used for points where $E = E_{\max}$ or $E = E_{\min}$.

our application, then both the grid-based sampling and the PDE-constrained greedy sampling algorithms offer similar performance in terms of full-order solves. If the maximum allowable error is greater than $10^{-4}$ then grid-based sampling is cheaper. Note that at the point where both algorithms start offering similar performance, the greedy sampling approach has only added 75 snapshots to the reduced basis approximation while the reduced order model built using the grid-based approach contains over 1200 snapshots. Storing such a large number of snapshots poses potential memory management problems and raises concerns regarding the size of the grid-based reduced order model when snapshots are orthonormalized using Gram-Schimdt. However, these problems can be easily overcome if snapshots are compressed using POD after every sample is added.

As discussed in the 2-parameter case, one problem of the Latin hypercube grid-based sampling that is not easy to overcome is the fact that it is not possible to get

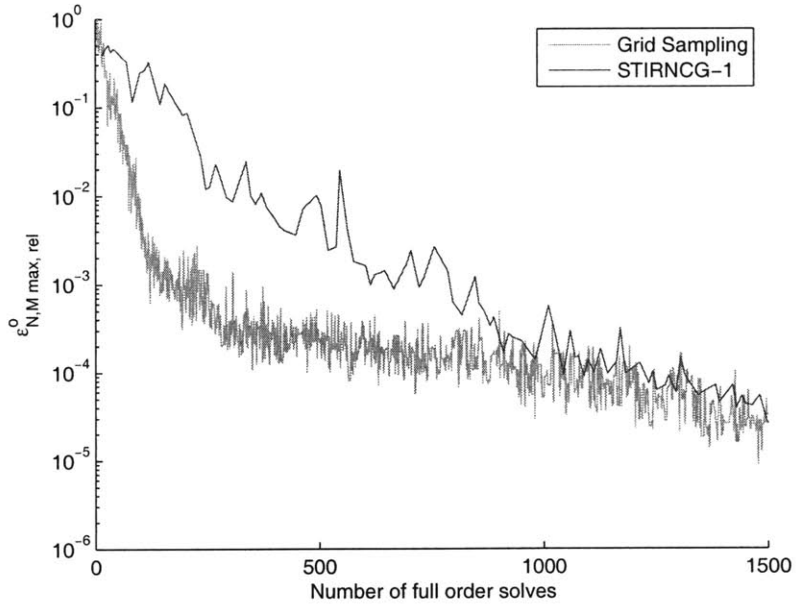

Figure 4-8: Maximum relative error over $\Xi^{\rm test}_{4913}$ for the reduced order model built using grid-based sampling (green) and the reduced order model built using PDE-constrained greedy sampling (red).

an indication of the accuracy of the reduced order model after each new snapshot is added. Therefore, it is not possible to establish a reliable criterion for determining when the desired accuracy has been reached. The greedy sampling methodology offers the error indicator $\Lambda_k(\boldsymbol{\mu}^*)$ defined in Section 4.2.7. The effectivity of this error indicator at every greedy cycle for the 3-parameter case is shown in Figure 4-9. Once again, $\Lambda_k(\boldsymbol{\mu}^*)$ gives a good order of magnitude of the maximum relative output error $\varepsilon^o_{N,M\,\max,\mathrm{rel}}$ at every greedy iteration. In this case, however, there are more points where the effectivity is significantly less than one, indicating that the optimization solver converged to a local maximum of the objective function. This is probably due to the fact that the objective function tends to become more multimodal as the number of parameter increases.

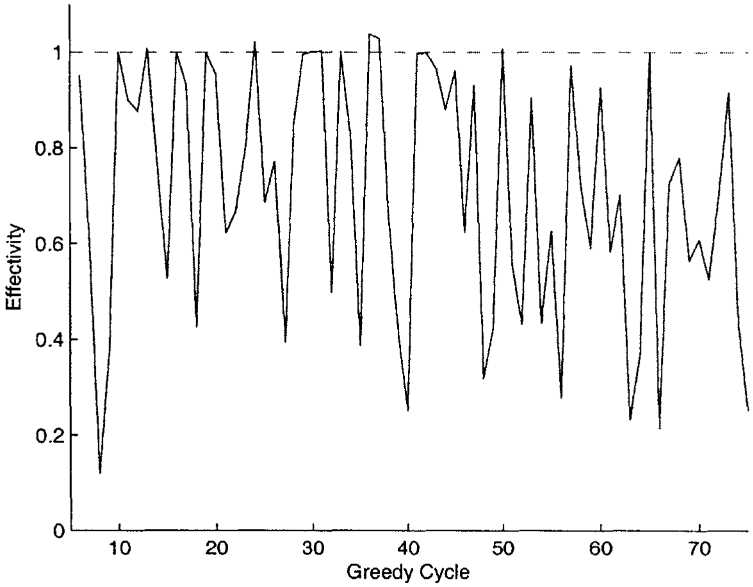

Figure 4-9: Effectivity of error indicator $\Lambda_k(\boldsymbol{\mu}^*)$ for the 3-parameter problem.

# Chapter 5

# Inverse Problem

## 5.1 Introduction

In previous chapters of this thesis we have extensively analyzed the *forward problem* associated with the nonlinear combustion PDE that was introduced in Chapter 2: given a known input parameter vector $\boldsymbol{\mu}^* \in \mathcal{D}$ we seek a reduced basis approximation to the output of interest, $\boldsymbol{o}_{N,M}(\boldsymbol{\mu}^*)$, that is arbitrarily close to the "truth" approximation $\boldsymbol{o}(\boldsymbol{\mu}^*)$. In this chapter we tackle the *inverse problem* of determining the unknown value of the parameter vector $\boldsymbol{\mu}^* \in \mathcal{D}$ given a set of empirically measured outputs $\bar{\boldsymbol{o}} \in \mathbb{R}^{N_s}$.

Two different approaches for determining the value of the unknown inputs given a set of empirical measurements are presented in this chapter. First, we develop a deterministic methodology that determines the "best" value of the parameter vector by minimizing the discrepancy between the empirical measurements and the numerical predictions provided by the reduced order model that was built in chapters 3 and 4. The deterministic approach builds upon the model-constrained optimization methodology that was presented in Chapter 4 for sampling the parameter space $\mathcal{D}$. It is therefore relatively straightforward to implement and computationally efficient. However, in the presence of experimental error and/or incomplete knowledge of the problem due to sparse measurements, the results provided by this methodology are usually insufficient and can in fact raise more questions than they answer: what is the uncertainty associated with the "best" estimate of $\boldsymbol{\mu}^*$ predicted by the optimization algorithm?. How do the experimental errors in the measured outputs propagate through the numerical model used to solve the inverse problem?. These are important questions that the deterministic approach is not capable of addressing and provide motivation for the second approach presented in this chapter.

In a realistic setting where experimental errors are not negligible it is not possible to determine the unknown parameter vector $\boldsymbol{\mu}^*$ exactly. Therefore, a single value or "best" estimate of $\boldsymbol{\mu}^*$ is of no real use if it is not accompanied by an uncertainty that gives the range of possible values of $\boldsymbol{\mu}^*$ that are consistent with the provided measurements. The second method presented in this chapter eliminates the shortcomings of the deterministic approach by using Bayesian statistical inference in order to relate the empirical measurements $\bar{\boldsymbol{o}} \in \mathbb{R}^{N_s}$ with the theoretical predictions $\boldsymbol{o}_{N,M}(\boldsymbol{\mu}^*)$ and thus generate a probability density function for the parameter $\boldsymbol{\mu}^*$.

For a general overview of this framework we refer the reader to [40, 57]. The probabilistic approach to inverse problems has been applied successfully in numerous settings including tomography [2, 29], geophysics [16, 51] and contaminant transport models [35]. However, the extreme computational cost required for exploring the input parameter domain using stochastic sampling methods is an open issue that still remains unanswered.

The probabilistic approach presented herein uses Markov Chain Monte Carlo (MCMC) simulations in order to sample the probability distributions of the input parameters and requires tens of thousands of forward problem solutions. This is a perfect example of the "multiple-query" context discussed in Section 1.1 where using a traditional, high-fidelity, numerical technique such as the finite element method for solving the forward problem would be computationally intractable. Recently, the use of lower-fidelity models has been proposed in order to decrease the cost of the probabilistic methodology [2, 30]. However, these low-fidelity models are not obtained via systematic model reduction methods; instead, they are built using traditional numerical methods — i.e., FEM, FDM, BEM — but use meshes that are coarser than usual so that the resulting problem becomes tractable. The limitations of this approach are evident since the level of mesh coarsening required for decreasing the computational cost of these problems to acceptable levels typically results in large errors that are often hard to quantify or even yield unstable numerical schemes.

In this thesis we use the reduced basis approximation developed in chapters 3 and 4 as a critical ingredient that enables the successful implementation of the Bayesian statistical inference approach for estimating the probability distribution of $\boldsymbol{\mu}^*$ given a set of empirical measurements. To the author's knowledge, this is the first published work where systematic model reduction is used in order to decrease the computational cost of the probabilistic inverse problem approach by several orders of magnitude while maintaining the accuracy and stability of the multiple forward solutions required by the MCMC simulations.

## 5.2 Deterministic Approach

In this section we formulate the inverse problem in the deterministic sense by minimizing the discrepancy between a measured set of outputs, $\bar{\boldsymbol{o}} \in \mathbb{R}^{N_s}$, and the outputs predicted using the reduced basis approximation built in chapters 3 and 4, $\boldsymbol{o}_{N,M}(\boldsymbol{\mu}^*) \in \mathbb{R}^{N_s}$. The outputs of interest are average fuel concentrations measured along vertical lines in the combustion chamber as shown in Figure 2-1. Instead of using experimental data, we choose the input parameters to be $\boldsymbol{\mu} = \boldsymbol{\mu}_e$ and replace the measurements with computations using the "truth" finite element solution: $\bar{\boldsymbol{o}} = \boldsymbol{o}(\boldsymbol{\mu}_e)$. As a consequence, the "measurements" almost exactly match the numerical predictions of the reduced basis approximation (the only difference between the numerical results and the artificial measurements is the error of the reduced basis approximation with respect to the "truth" solution, which has been shown to be negligible for reasonable values of $N$ and $M$). Since the deterministic approach is not well suited for handling experimental uncertainties, no empirical errors have been added to the synthetic measurements. The reader is referred to the probabilistic approach to inverse problems presented in Section 5.3 for a rigorous derivation on how to incorporate non-negligible empirical errors in the inverse problem formulation.

### 5.2.1 Formulation Using Model-Constrained Optimization

Given a reduced order model and a measured output of interest we solve the inverse problem by finding the point in parameter space where the $\ell^2$-norm of the difference between the output predicted by the reduced order model and the actual measured output is minimal,

$$\min_{\boldsymbol{u}_{N,M},\boldsymbol{\mu}} \mathcal{J}(\boldsymbol{u}_{N,M},\boldsymbol{\mu}) \overset{def}{=} \frac{1}{2}\|\boldsymbol{o}_{N,M} - \bar{\boldsymbol{o}}\|_2^2 \tag{5.1}$$

$$\text{subject to } \boldsymbol{R}_N(\boldsymbol{u}_{N,M};\boldsymbol{\mu}) = 0 \tag{5.1a}$$

$$\boldsymbol{o}_{N,M} = \boldsymbol{L}_N^O \boldsymbol{u}_{N,M} + \boldsymbol{L}_0^O \tag{5.1b}$$

$$\boldsymbol{\mu}_{\min} \leq \boldsymbol{\mu} \leq \boldsymbol{\mu}_{\max} \tag{5.1c}$$

where $\boldsymbol{u}_{N,M} \in \mathbb{R}^N$ is the reduced state variable, $\boldsymbol{\mu} \in \mathbb{R}^P$ is the parameter vector or decision variable, $\mathcal{J} \in \mathbb{R}$ is the objective function, $\boldsymbol{R}_N \in \mathbb{R}^N$ are the reduced discretized equations as defined in (3.61), and $\bar{\boldsymbol{o}} \in \mathbb{R}^{N_s}$, $\boldsymbol{o}_{N,M} \in \mathbb{R}^{N_s}$ are the "measured" and reduced linear outputs of interest. The reduced basis output is computed using the matrices $\boldsymbol{L}_N^O \in \mathbb{R}^{N_s \times N}$ and $\boldsymbol{L}_0^O \in \mathbb{R}^{N_s}$ defined in equations (3.56) and (3.57). The upper and lower bounds for the parameter vector are given by $\boldsymbol{\mu}_{\max}$ and $\boldsymbol{\mu}_{\min}$ respectively. Recall that the parameter vector is formed by the two parameters ($P = 2$) that control the combustion reaction; i.e., $\boldsymbol{\mu} = (\log(A), E)$.

Note that the optimization problem (5.1) is formulated in terms of the reduced basis output. If the "truth" FE model were used, the formulation would look the same. One would simply need to replace constraint (5.1a) with the full discretized equations, $\boldsymbol{R}(\boldsymbol{u};\boldsymbol{\mu}) = 0$, and (5.1b) with the FE output, $\boldsymbol{o} = \boldsymbol{L}^O\boldsymbol{u}$. The solution method would be identical to that described in this section for the reduced basis case. For the remainder of this section we will limit ourselves to the formulation using the reduced order model since it is computationally more efficient and yields practically the same accuracy. Substituting the output constraints into the objective function yields

$$\begin{aligned} \min_{\boldsymbol{u}_{N,M},\boldsymbol{\mu}} \mathcal{J}(\boldsymbol{u}_{N,M},\boldsymbol{\mu}) &\overset{def}{=} \frac{1}{2}\|\boldsymbol{L}_N^O\boldsymbol{u}_{N,M} + \boldsymbol{L}_0^O - \bar{\boldsymbol{o}}\|_2^2 \\ \text{subject to } \boldsymbol{R}_N(\boldsymbol{u}_{N,M};\boldsymbol{\mu}) &= 0 \\ \boldsymbol{\mu}_{\min} \leq \ \boldsymbol{\mu} &\leq \boldsymbol{\mu}_{\max} \end{aligned} \tag{5.2}$$

The optimality conditions for the model-constrained optimization problem (5.2) are derived by defining the Lagrangian functional,

$$\mathcal{L}(\boldsymbol{u}_{N,M},\boldsymbol{\mu},\boldsymbol{\lambda}_N) \overset{\text{def}}{=} \mathcal{J}(\boldsymbol{u}_{N,M}) + \boldsymbol{\lambda}_N^T\boldsymbol{R}_N(\boldsymbol{u}_{N,M};\boldsymbol{\mu}) \tag{5.3}$$

where $\boldsymbol{\lambda}_N \in \mathbb{R}^N$ is the adjoint variable that enforces the reduced model equations. As was done in Chapter 4, the bound constraints are treated separately inside the optimization solver and, consequently, are not explicitly included in the Lagrangian. The first order Karush-Kuhn-Tucker optimality conditions for this problem are given by

$$\begin{bmatrix} \partial_{\boldsymbol{u}_{N,M}}\mathcal{L} \\ \partial_{\boldsymbol{\mu}}\mathcal{L} \\ \partial_{\boldsymbol{\lambda}_N}\mathcal{L} \end{bmatrix} = \begin{bmatrix} \boldsymbol{g}_{\boldsymbol{u}_{N,M}} + \boldsymbol{J}_{N\boldsymbol{u}_{N,M}}^T\boldsymbol{\lambda}_N \\ \boldsymbol{J}_{N\boldsymbol{\mu}}^T\boldsymbol{\lambda}_N \\ \boldsymbol{R}_N \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \tag{5.4}$$

where $\boldsymbol{g}_{\boldsymbol{u}_{N,M}} \in \mathbb{R}^N$ is the gradient of $\mathcal{J}$ with respect to the reduced state variable:

$$\boldsymbol{g}_{\boldsymbol{u}_{N,M}} = (\boldsymbol{L}_N^O)^T(\boldsymbol{L}_N^O\boldsymbol{u}_{N,M} + \boldsymbol{L}_0^O - \bar{\boldsymbol{o}}) \tag{5.5}$$

$\boldsymbol{J}_{N\boldsymbol{u}_{N,M}} \in \mathbb{R}^{N\times N}$ is the Jacobian of the reduced basis equations which can be computed using (3.62) and $\boldsymbol{J}_{N\boldsymbol{\mu}} \in \mathbb{R}^{N\times P}$ is the Jacobian of the reduced state equations with respect to the decision variables, which is given by (4.11) and (4.12).

The nonlinear system of equations (5.4) is solved using the reduced space framework described in Section 4.2.3. Since the minimization problem (5.1) only involves the reduced model equations, any reasonable optimization algorithm will converge to the optimum with a very low computational cost. Therefore, instead of using the subspace trust region interior-reflective Newton method that requires computing the reduced Hessian at every iteration, we use the simpler sequential quadratic programming method that was also described in Section 4.2.5. Recall that this method constructs an estimate of the reduced Hessian at every iteration using the BFGS formula so only the reduced gradient is computed analytically.

The reduced gradient is computed at every iteration by first solving the reduced state equations to find the reduced state variable $\boldsymbol{u}_{N,M}$, then solving the adjoint equations to compute reduced adjoint variable $\boldsymbol{\lambda}_N$ and finally substituting the state and adjoint variables into the decision equation. The residual of the decision equation is the reduced gradient, $\nabla\mathcal{J}$:

$$\nabla\mathcal{J} = -\boldsymbol{J}_{N\boldsymbol{\mu}}^{T}\boldsymbol{J}_{N\boldsymbol{u}_{N,M}}^{-T}\boldsymbol{g}_{\boldsymbol{u}_{N,M}} \tag{5.6}$$

### 5.2.2 Numerical Results Using Deterministic Approach

In this section we present some numerical results for the inference of the Arrhenius parameters $\boldsymbol{\mu} = (\log(A), E)$ given a set of experimental measurements using the deterministic framework described in Section 5.2.1. As discussed in the introduction, the "measurements" are generated numerically by choosing a fixed value for the input parameters, $\boldsymbol{\mu}_e = (\log(A_e), E_e)$ and solving the forward problem using the "truth" finite element model that was developed in Chapter 2. The inverse problem is solved using a reduced basis approximation with $N = 40$ and $M = 50$ so the difference between the "measurements" and the numerical outputs of interest is negligible (on the order of $10^{-6}$).

For the numerical experiment presented in this section, the "measurements" are generated using an input parameter vector given by $(\log(A_e), E_e) = (7.147727, 0.054545)$, which corresponds to a reaction-dominated, highly nonlinear solution. The initial guess for the sequential quadratic programming optimization algorithm is chosen as $(\log(A)_0, E_0) = (5.00, 0.15)$, which corresponds to a convection-dominated, weakly nonlinear solution. Both the initial guess and the actual solution used to generate the "measurements" are shown in Figure 5-1. The measurement lines are represented

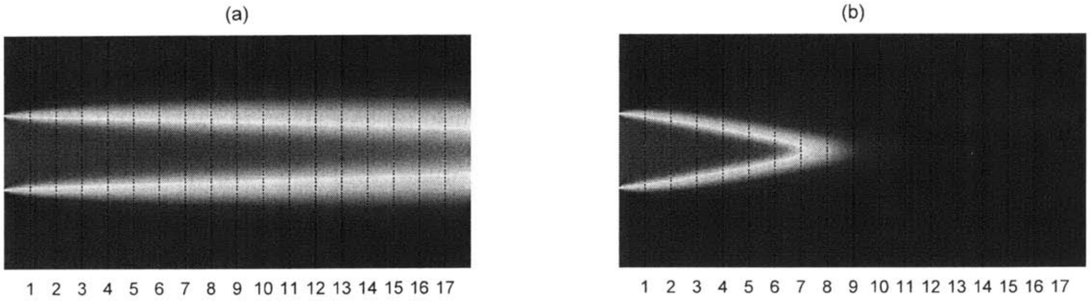

Figure 5-1: Initial guess for optimization algorithm (a) ($\boldsymbol{\mu}_0 = (5.00, 0.15)$) and solution used to generate "measurements" (b) ($\boldsymbol{\mu}_e = (7.147727, 0.054545)$). Measurements are average fuel concentrations along dashed vertical lines. Line indices give the x-coordinate of each measurement line in millimeters.

by vertical lines at equally spaced intervals along the combustion chamber. Note that the initial guess differs significantly from the "truth" solution used to generate the measurements ($\boldsymbol{\mu}_0$ and $\boldsymbol{\mu}_e$ are almost in opposite corners of parameter space) so the example chosen to illustrate this methodology is clearly a challenging one.

The number and location of the measurement lines chosen to infer the Arrhenius parameters that generated a given set of outputs plays a critical role in the well-posedness of the inverse problem. For example, in the case presented in this section, the measurements along lines $\Gamma_{10}$ through $\Gamma_{17}$ are practically zero due to the fact that the flame does not reach these lines. Therefore, if only these lines were used to generate the measured outputs, the resulting inverse problem would be clearly ill-posed. Using all 17 measurement lines is a safe option for obtaining a well-posed problem since they cover the entire combustion chamber. When all lines are used, the inverse problem is well-posed and, for this particular case, the resulting optimization problem is convex so the same optimum is found regardless of the initial guess used in the optimizer. Figure 5-2 shows the performance of the optimization solver for the numerical example considered in this section using all 17 measurement lines available in the model. The optimum is located at $\boldsymbol{\mu}^* = (7.147728, 0.054551)$ and the optimization solver requires 15 iterations and 61 reduced order solves in order to find it. Note that the solves required by the optimization algorithm only involve the reduced order model so the computational cost required to find the optimum is negligible. Also note that the $\ell^2$-norm of the error between the inferred parameter vector $\boldsymbol{\mu}^*$ and the actual input vector $\boldsymbol{\mu}_e$ used to generate the synthetic measurements is $6.65 \cdot 10^{-6}$, which is of the same order of magnitude as the error between the reduced basis approximation used to solve the inverse problem and the "truth" finite element model used to compute the outputs.

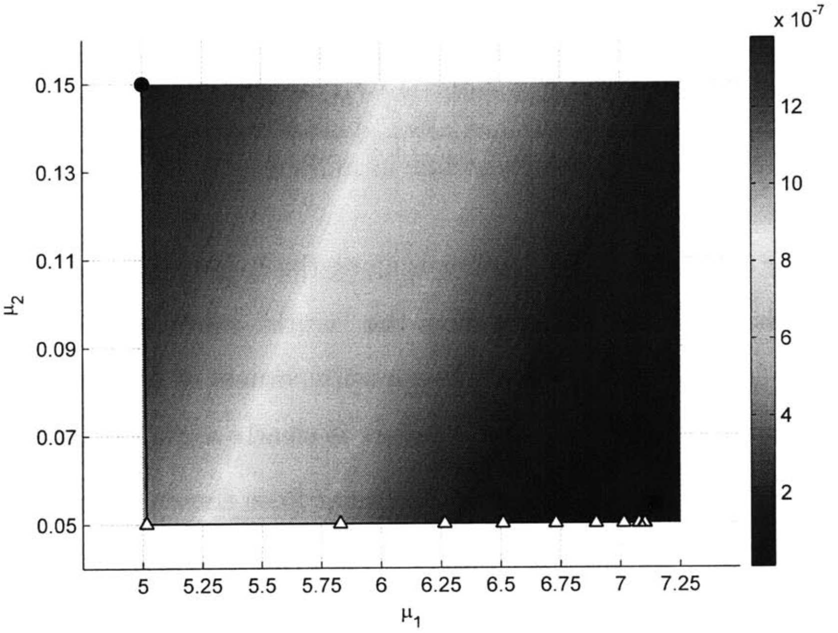

Figure 5-2: Solution of inverse problem using sequential quadratic programming and 17 measurements. Colormap shows the value of the objective function $\mathcal{J}$ over the entire parameter domain of interest $\mathcal{D}$. Figure shows location of initial guess (●), optimum (■), and intermediate steps after every iteration of the optimization solver (△).

As discussed previously, selecting all 17 available measurements is a guarantee of obtaining a well-posed inverse problem. However, it is also possible to select fewer measurement lines and still be able to infer the value of the Arrhenius parameters without a significant impact on the error. The advantage of selecting fewer measurement locations for solving the inverse problem is twofold: first, in a realistic setting where actual measurements are collected, it is always desirable to limit the number of required measurements to a minimum in order to decrease the experimental cost of the process. Second, the size of a reduced order model is generally related to

| Measurement lines used | Inverted Solution $\mu_1^*$ | $\mu_2^*$ | Error $\|\|\boldsymbol{\mu}_e - \boldsymbol{\mu}^*\|\|_2$ | Number of reduced order solves |
|---|---|---|---|---|
| All | 7.147728 | 0.054551 | $6.65\,\text{E}-06$ | 61 |
| 1-6, 8, 10, 13 | 7.147894 | 0.054567 | $1.68\,\text{E}-04$ | 69 |
| 1, 3, 5, 8, 13 | 7.147978 | 0.054576 | $2.53\,\text{E}-04$ | 65 |
| 1, 3, 10 | 7.147257 | 0.054494 | $4.72\,\text{E}-04$ | 76 |
| 1, 3 | 7.146442 | 0.054403 | $1.29\,\text{E}-03$ | 53 |

Table 5.1: Inverse problem solutions using sequential quadratic programming optimization algorithm. First column contains indices of measurement lines used in each case (refer to Figure 5-1 for line locations). In all cases the synthetic measurements were computed using the "truth" finite element model with $\boldsymbol{\mu}_e = (7.147727, 0.054545)$ and a starting initial guess for the optimization algorithm given by $\boldsymbol{\mu}_0 = (5.00, 0.15)$. Error between the input parameter used to compute the "measurements", $\boldsymbol{\mu}_e$, and inverse problem solution, $\boldsymbol{\mu}^*$, is given by error $= \|\boldsymbol{\mu}_e - \boldsymbol{\mu}^*\|_2$. The last column shows the number of reduced basis solutions computed by the optimization solver in order to reach the optimum $\boldsymbol{\mu}^*$. The reduced order model used by the optimizer contained 40 basis functions ($N = 40$) and 50 interpolation functions ($M = 50$).

the number of outputs that the model must compute accurately; fewer outputs of interest usually require smaller reduced order models, which, in turn, decreases the computational cost of both the offline and online phases of the reduced basis approximation. Table 5.1 shows the effect of selecting fewer measurement lines for solving the inverse problem. We note once again that care should be exercised in order to select lines that provide measurements that are representative of the case of interest — i.e., measurements where the concentration of fuel is non-zero and which cover the flame region as thoroughly as possible.

## 5.3 Probabilistic Approach

In this section we present a Bayesian statistics framework for inferring the input parameters of interest, $\boldsymbol{\mu}^* \in \mathbb{R}^2$, given a set of sparse measurements, $\bar{\boldsymbol{o}} \in \mathbb{R}^{N_s}$, with non-negligible experimental error. We refer the reader to [40, 57] for a detailed discussion on the probabilistic approach to inverse problems used herein.

### 5.3.1 Bayesian Formulation

Given a known probability distribution for the experimental errors and a known input parameter vector $\boldsymbol{\mu}^*$, it is possible to solve the forward problem using the reduced basis approximation from chapters 3 and 4 in order to compute $p(\bar{\boldsymbol{o}}|\boldsymbol{\mu}^*)$, the conditional probability of the observed measurements given the input. This is called the *forward probability* density. When solving the inverse problem we are interested in $p(\boldsymbol{\mu}^*|\bar{\boldsymbol{o}})$, which is the conditional probability of the possible inputs given that a certain set of measurements has been collected. This is the *inverse probability*, which defines our knowledge of the inputs after measuring the outputs.

Forward and inverse probabilities are related by Bayes' theorem as follows,

$$p(\boldsymbol{\mu}^*|\bar{\boldsymbol{o}}) = \frac{1}{p(\bar{\boldsymbol{o}})} p(\bar{\boldsymbol{o}}|\boldsymbol{\mu}^*) p(\boldsymbol{\mu}^*) \tag{5.7}$$

where $p(\boldsymbol{\mu}^*)$ is the probability function that contains our knowledge of the parameter $\boldsymbol{\mu}$ before collecting any measurements. This is known as the *prior probability*. The inverse probability $p(\boldsymbol{\mu}^*|\bar{\boldsymbol{o}})$, often called *posterior probability*, defines our knowledge of the inputs given a set of measurements. Determining this probability distribution is the goal of the probabilistic inverse problem framework. The forward probability function, $p(\bar{\boldsymbol{o}}|\boldsymbol{\mu}^*)$, introduced before can be determined from the experimental errors and our numerical reduced order model. Finally, the remaining factor, $p(\bar{o})^{-1}$, is determined by normalization since the integral of the posterior probability distribution over all possible inputs must be equal to one.

### 5.3.2 Problem Formulation Using Stochastic Simulation

Once again, the "measurements" in our problem are generated by setting the input parameters to $\boldsymbol{\mu} = \boldsymbol{\mu}_e$ and replacing the measurements with computations using the "truth" finite element solution: $\bar{\boldsymbol{o}} = \boldsymbol{o}(\boldsymbol{\mu}_e)$. However, in this case we also introduce a normally distributed measurement error such that the final vector of measurements, $\bar{\boldsymbol{o}} \in \mathbb{R}^{N_s}$, is given by

$$\bar{o}_i = o_i(\boldsymbol{\mu}_e) + \varepsilon_i, \quad 1 \leq i \leq N_s, \tag{5.8}$$

where the measurement errors $\varepsilon_i$, $1 \le i \le N_s$ are assumed to be normally distributed with standard deviation $\sigma$ and zero mean. We note that this framework also allows considering additional errors that may exist in any of the steps of the process used to compute the outputs of interest given a set of inputs. In our case, for example, we could consider incorporating the error between the "truth" finite element solution and the reduced basis approximation used to solve the inverse problem into the error vector $\boldsymbol{\varepsilon} \in \mathbb{R}^{N_s}$. The question that arises when doing this is the type of probability distribution associated with the reduced basis error. Note that the reduced basis error is non-uniform across the parameter domain $\mathcal{D}$ and clearly non-normal. In our case, we use a very accurate reduced basis approximation with $N = 40$ and $M = 50$ so the reduced basis error is negligible compared to the experimental error and is therefore excluded from the formulation. However, incorporating the numerical error of the reduced basis approximation in the probabilistic formulation is clearly an interesting topic that should be considered when using small reduced order models (with large numerical errors).

The only prior knowledge about the input parameter before making the measurements is that it must reside in the input parameter domain $\mathcal{D}$. Therefore, we assume that the prior distribution of $\boldsymbol{\mu}^*$ is uniform on $\boldsymbol{\mu}^* \in \mathcal{D}$,

$$p(\boldsymbol{\mu}) \propto \begin{cases} 1, & \text{if } \boldsymbol{\mu} \in \mathcal{D} \\ 0, & \text{otherwise.} \end{cases} \tag{5.9}$$

Since the measurement errors are normally distributed, the forward probability is given by

$$\begin{aligned} p(\bar{o}_i|\boldsymbol{\mu}) = & \prod_{i=1}^{N_s} p(\varepsilon_i = \bar{o}_i - o_i(\boldsymbol{\mu})) \\ \propto & \exp\left[-\frac{1}{2}\sum_{i=1}^{N_s}\left(\frac{\bar{o}_i - o_i(\boldsymbol{\mu})}{\sigma}\right)^2\right], \quad 1 \le i \le N_s. \end{aligned} \tag{5.10}$$

We now invoke Bayes' theorem to obtain the following posterior probability density for $1 \leq i \leq N_s$,

$$p(\boldsymbol{\mu}|\bar{o}_i) \propto \begin{cases} \exp\left[-\frac{1}{2}\sum_{i=1}^{N_s}\left(\frac{\bar{o}_i - o_i(\boldsymbol{\mu})}{\sigma}\right)^2\right], & \text{if } \boldsymbol{\mu} \in \mathcal{D} \\ 0, & \text{otherwise.} \end{cases} \tag{5.11}$$

In order to sample from the posterior probability density function we use the Metropolis-Hastings Markov Chain Monte Carlo (MCMC) algorithm [27]. Using this algorithm we can generate a Markov chain $\{Y_n\}_{n=1}^{N_m}$ with equilibrium distribution (5.11). The algorithm is as follows:

Let $Y_n = (\mu_1, \mu_2)$, $Y_{n+1}$ is found by performing the following two steps,

1. **Generation Step:** Let $r_1$ and $r_2$ be stochastic variables drawn from a uniform distribution on $[0,1]$ and $w_1$, $w_2$ be positive constants. Set

$$\mu_1' = \mu_1 + w_1(2r_1 - 1) \tag{5.12}$$

$$\mu_2' = \mu_2 + w_2(2r_2 - 1), \tag{5.13}$$

which means that the proposed density function for the generation step is given by

$$g(\mu_1', \mu_2'|\mu_1, \mu_2) = \begin{cases} \frac{1}{4w_1w_2} & \text{if } |\mu_1' - \mu_1| < w_1 \text{ and } |\mu_2' - \mu_2| < w_2, \\ 0 & \text{otherwise.} \end{cases} \tag{5.14}$$

Note that $g(\mu_1', \mu_2'|\mu_1, \mu_2) = g(\mu_1, \mu_2|\mu_1', \mu_2')$.

2. **Acceptance Step:** With probability

$$\begin{aligned} \alpha(\mu_1', \mu_2'|\mu_1, \mu_2) &= \min\left\{1, \frac{p(\mu_1', \mu_2'|\bar{o}_1, \ldots, \bar{o}_{N_s})}{p(\mu_1, \mu_2|\bar{o}_1, \ldots, \bar{o}_{N_s})} \frac{g(\mu_1', \mu_2'|\mu_1, \mu_2)}{g(\mu_1, \mu_2|\mu_1', \mu_2')}\right\} \\ &= \min\left\{1, \exp\left[-\frac{1}{2}\sum_{i=1}^{N_s}\left[\left(\frac{\bar{o}_i - o_i(\mu_1', \mu_2')}{\sigma}\right)^2 - \frac{\bar{o}_i - o_i(\mu_1, \mu_2)}{\sigma}\right)^2\right]\right]\right\} \end{aligned} \tag{5.15}$$

set $Y_{n+1} = (\mu'_1, \mu'_2)$ (i.e., accept $\boldsymbol{\mu}'$), otherwise set $Y_{n+1} = (\mu_1, \mu_2)$ (i.e., reject $\boldsymbol{\mu}'$).

The samples in the Markov chain $\{Y_n\}_{n=1}^{N_m}$ are drawn from the posterior probability distribution so the distribution of $\{Y_n\}_{n=1}^{N_m}$ converges to the actual posterior probability distribution of $\boldsymbol{\mu}^*$ as $N_m \to \infty$. However, it is clear that for practical purposes we need to choose a finite value for $N_m$ (and hopefully not so large that generating the Markov chain will become a computationally intractable task). In general, we are interested in using the Markov chain for computing a statistic of interest $f(Y)$ so we need to select $N_m$ such that our estimate of this statistic using a specific realization of the chain is representative of the actual value of the statistic. Let $Y_0 = y^{(0)}, Y_1 = y^{(1)} \dots Y_{N_m} = y^{(N_m)}$ be a realization of a homogeneous and reversible Markov chain. In our case, $\{y^{(n)}\}_{n=1}^{N_m}$ is the output of the MCMC algorithm outlined previously in this section. Let $f(Y)$ be the statistic of interest; we can estimate the expected value of this statistic, referred to as $\langle f(Y) \rangle$, using the output sample set $\{y^{(n)}\}_{n=1}^{N_m}$ by computing the quantity

$$\bar{f}_{N_m}(\{y^{(n)}\}_{n=1}^{N_m}) \equiv \frac{1}{N_m} \sum_{n=1}^{N_m} f(y^{(n)}) \tag{5.16}$$

where $\bar{f}_{N_m}(\{y^{(n)}\}_{n=1}^{N_m})$ is called an estimator of $\langle f(Y) \rangle$. Since the samples $\{y^{(n)}\}_{n=1}^{N_m}$ are realizations of random variables $\{Y_n\}_{n=1}^{N_m}$, our estimate, $\bar{f}_{N_m}(\{y^{(n)}\}_{n=1}^{N_m})$, is itself a random variable and, therefore, a different estimate will be obtained each time we compute a realization $\{y^{(n)}\}_{n=1}^{N_m}$ of the random variable $\{Y_n\}_{n=1}^{N_m}$.

In order to answer the question of how good of an estimator for $\langle f(Y) \rangle$ is $\bar{f}_{N_m}$ we note that when $N_m$ is large, $\bar{f}_{N_m}$ is normally distributed with mean $\langle f \rangle$ and some variance, $\mathrm{var}(\bar{f}_{N_m})$. Therefore, using the central limit theorem we can say that the limit of $\sqrt{N_m}(\bar{f}_{N_m} - \langle f \rangle)$ as $N_m \to \infty$ tends to a normal distribution with zero mean and variance $c$ independent of $N_m$ as discussed in [19]. Therefore, if $\{y^{(n)}\}_{n=1}^{N_m}$ were independent samples and $\bar{f}_{N_m} \equiv \frac{1}{N_m}\sum_{n=1}^{N_m} f(y^{(n)})$ then

$$\mathrm{var}(\bar{f}_{N_m}) = \frac{\mathrm{var}(f)}{N_m} \tag{5.17}$$

However, since the samples $\{y^{(n)}\}_{n=1}^{N_m}$ are not independent, we should expect the variance to decrease more slowly due to the presence of correlated samples. The variance of our estimate is actually given by

$$\mathrm{var}(\bar{f}_{N_m}) = \frac{\tau_f \,\mathrm{var}(f)}{N_m} \tag{5.18}$$

where $\tau_f > 0$ is the integrated autocorrelation time that accounts for the existence of correlated samples in the Markov chain. Note that if we are able to compute $\tau_f$ given a particular realization of a Markov chain, then we can estimate the accuracy of our estimate $\bar{f}_{N_m}$ by using (5.18) to compute its standard deviation ($\sigma_{\bar{f}_{N_m}} = \sqrt{\mathrm{var}(\bar{f}_{N_m})}$) and therefore give a confidence interval $\bar{f}_{N_m} \pm q\sigma_{\bar{f}_{N_m}}$, $q \in \mathbb{N}$, for the statistic of interest.

The definition of $\tau_f$ is given by

$$\tau_f \equiv 1 + 2\sum_{s=1}^{\infty} \rho_{ff}(s) \tag{5.19}$$

where $\rho_{ff}(s)$ is the normalized autocovariance function at lag $s$, defined as follows

$$\rho_{ff}(s) = \frac{1}{\mathrm{var}(f(Y_n))}\mathrm{cov}(f(Y_n), f(Y_{n+s})) \tag{5.20}$$

The normalized autocovariance function $\rho_{ff}(s)$ approaches zero as the lag $s$ increases (the random variables in the Markov chain become more uncorrelated as the distance between them increases). However, since $N_m$ is finite, the estimates of $\rho_{ff}(s)$ are noisy so there is a value $M_m$ of $s$ such that for $s \geq M_m$ the values of $\rho_{ff}(s)$ are pure noise. Therefore, if we were to compute our estimate $\bar{\tau}_f$ for $\tau_f$ by summing over all $s$ as shown in (5.19) the resulting estimate would be corrupted by noise. In practice, we truncate the sum over $\rho_{ff}(s)$ at $s = M_m$ where noise begins to dominate.

For our particular problem we will define the mean, $\langle Y \rangle$, and the variance, $\langle (Y - \langle Y \rangle)^2 \rangle$, of the posterior probability distribution of $\boldsymbol{\mu}^*$ as our statistics of interest. The estimator for the mean is given by

$$\bar{f}_{1\,N_m}(\{y^{(n)}\}_{n=1}^{N_m}) \equiv \frac{1}{N_m}\sum_{n=1}^{N_m} f_1(y^{(n)}), \tag{5.21}$$

where $f_1(y) = y$. The estimator of the variance is given by

$$\bar{f}_{2\,N_m}(\{y^{(n)}\}_{n=1}^{N_m}) \equiv \frac{1}{N_m}\sum_{n=1}^{N_m} f_2(y^{(n)}), \tag{5.22}$$

with $f_1(y) = (y - \bar{y})^2$ and $\bar{y} = \frac{1}{N_m}\sum_{n=1}^{N_m} y^{(n)}$.

The uncertainty associated with estimators $\bar{f}_{1\,N_m}$ and $\bar{f}_{2\,N_m}$ is obtained by first computing a Markov chain of length $N_m$, dropping the states from the start of the run, and then estimating $\tau_{f_1}$ and $\tau_{f_2}$ for the statistics $f_1(Y)$ and $f_2(Y)$ according to (5.19). The variance associated with each estimator is computed using equation (5.18). The standard deviation can then be computed as $\sigma_{\bar{f}_{i\,N_m}} = \sqrt{\mathrm{var}(\bar{f}_{i\,N_m})}$, $i = 1, 2$. Finally, a 95% confidence interval is reported for each statistic of interest by using the value of the estimate and its standard deviation: $\left[\bar{f}_{i\,N_m} - 2\sigma_{\bar{f}_{i\,N_m}}, \bar{f}_{i\,N_m} + 2\sigma_{\bar{f}_{i\,N_m}}\right]$, $i = 1, 2$.

As a final remark, we note that the confidence intervals for the statistics of interest developed above are only valid if the Markov chain has converged to its equilibrium distribution. Since convergence of a Markov chain cannot be demonstrated in general, it may appear that the methodology presented in this section is difficult to validate. However, in practice, the following steps are sufficient for assessing the convergence of the MCMC outputs of interest: first, it is necessary to check that $N_m \gg \max\{\tau_{f_i}\}$, $i = 1, 2$, in order to guarantee that the chain is sufficiently long. Second, one should verify that the realization of the chain, $\{y^{(n)}\}_{n=1}^{N_m}$, does not show any obvious trend. The final step to check convergence requires plotting the normalized autocovariance functions of the statistics of interest against the lag $s$ in order to check that they fall off smoothly to zero and then oscillate randomly about the x-axis due to noise.

| Measurement line index | FE output $o(\boldsymbol{\mu}_e)$ | Set 1 $\bar{o}_{0.5\%}(\boldsymbol{\mu}_e)$ | Set 2 $\bar{o}_{1.5\%}(\boldsymbol{\mu}_e)$ | Set 3 $\bar{o}_{2.5\%}(\boldsymbol{\mu}_e)$ |
|---|---|---|---|---|
| 1 | 0.5570 | 0.5565 | 0.5592 | 0.5646 |
| 2 | 0.5288 | 0.5269 | 0.5306 | 0.5328 |
| 3 | 0.4936 | 0.4938 | 0.4996 | 0.5006 |
| 4 | 0.4534 | 0.4538 | 0.4555 | 0.4463 |
| 5 | 0.4095 | 0.4082 | 0.4073 | 0.4094 |
| 6 | 0.3628 | 0.3642 | 0.3641 | 0.3618 |
| 7 | 0.3137 | 0.3151 | 0.3101 | 0.3042 |
| 8 | 0.2625 | 0.2625 | 0.2624 | 0.2640 |
| 9 | 0.2095 | 0.2099 | 0.2093 | 0.2033 |
| 10 | 0.1564 | 0.1566 | 0.1564 | 0.1647 |
| 11 | 0.1075 | 0.1073 | 0.1064 | 0.1028 |
| 12 | 0.0682 | 0.0690 | 0.0721 | 0.0713 |
| 13 | 0.0406 | 0.0399 | 0.0340 | 0.0419 |
| 14 | 0.0232 | 0.0257 | 0.0247 | 0.0177 |
| 15 | 0.0129 | 0.0127 | 0.0161 | 0.0001 |
| 16 | 0.0071 | 0.0072 | 0.0097 | 0.0067 |
| 17 | 0.0039 | 0.0051 | 0.0059 | -0.0021 |

Table 5.2: Sets of measurements used to test performance of probabilistic approach. First column contains the line index for each measurement (refer to Figure 5-1 for line locations). The second column contains the exact output at each line, $o(\boldsymbol{\mu}_e)$, computed using the FE model. The last three columns contain the three sets of measurements generated by adding normally distributed measurement errors to the exact outputs shown in the second column.

### 5.3.3 Numerical Examples

In this section we present numerical results for the posterior distribution of $\boldsymbol{\mu}^*$ for three different sets of measurements. These sets of measurements were generated using expression (5.8) with $\boldsymbol{\mu}_e = (6.7386, 0.0727)$ and normally distributed measurement errors with standard deviations given by $\sigma_{0.5\%} = 1.18 \cdot 10^{-3}$, $\sigma_{1.5\%} = 3.54 \cdot 10^{-3}$ and $\sigma_{2.5\%} = 5.90 \cdot 10^{-3}$. These standard deviations correspond to approximately 0.5%, 1.5% and 2.5% of the average fuel concentration over the seventeen output measurements considered in this problem. Table 5.2 shows the exact outputs $o(\boldsymbol{\mu}_e)$ computed using the "truth" finite element model as well as the three sets of artificial measurements obtained by drawing a sample from each distribution corresponding to the actual solution plus the normally distributed experimental errors.

|  | IAC for $\mu_1^* = \log(A^*)$ |  | IAC for $\mu_2^* = E^*$ |  |
|---|---|---|---|---|
| Meas. Set | $\tau_{f_1}$ (mean) | $\tau_{f_2}$ (variance) | $\tau_{f_1}$ (mean) | $\tau_{f_2}$ (variance) |
| $\bar{o}_{0.5\%}(\boldsymbol{\mu}_e)$ | 223 | 304 | 222 | 299 |
| $\bar{o}_{1.5\%}(\boldsymbol{\mu}_e)$ | 263 | 147 | 266 | 150 |
| $\bar{o}_{2.5\%}(\boldsymbol{\mu}_e)$ | 246 | 168 | 246 | 161 |

Table 5.3: Integrated autocorrelation time (IAC), $\tau_f$, for the two statistics of interest (mean and variance) considered for each of the two unknown input parameters of the problem, $\mu_1^* = \log(A^*)$, and $\mu_2^* = E^*$.

As described in Section 5.3.2, the statistics of interest are the mean and the variance of the probability distributions of $\mu_1^* = \log(A^*)$ and $\mu_2^* = E^*$ for each set of measurements. The posterior distributions were sampled using the MCMC algorithm outlined in Section 5.3.2 with 50,000 samples for each Markov chain ($N_m = 50,000$). Table 5.3 shows the values of the integrated autocorrelation time for both statistics of interest for the three sets of measurements. Note that in all cases $N_m$ is much greater than $\tau_f$, which is the first practical check discussed in Section 5.3.2 for assessing convergence of the Markov chain.

Figures 5-3 through 5-5 show the Markov chain samples for each of the three sets of measurements. No obvious trend is observed in any of the sequences, which is a valid indication of the fact that the chains are in equilibrium as discussed in Section 5.3.2.

As a final check in order to assess the convergence of the Markov chains used to compute the outputs presented in this section, Figures 5-6 and 5-7 show the normalized autocovariance functions for the mean, $\rho_{f_1 f_1}(s)$, and the variance, $\rho_{f_2 f_2}(s)$, against the lag $s$ for both input parameters and all three sets of measurements. In all cases, the normalized autocovariance functions approach the x-axis smoothly and then oscillate randomly due to noise.

After checking that $N_m \gg \max\{\tau_{f_i}\}, i = 1, 2$ (see Table 5.3), verifying that the realization of the chains, $\{y^{(n)}\}_{n=1}^{N_m}$, do not show any obvious trend (see Figures 5-3 through 5-5) and plotting the normalized autocovariance functions of the statistics of interest against the lag $s$ in order to check that they fall off smoothly to zero and then oscillate randomly about the x-axis due to noise (see Figures 5-6 and 5-7), we can

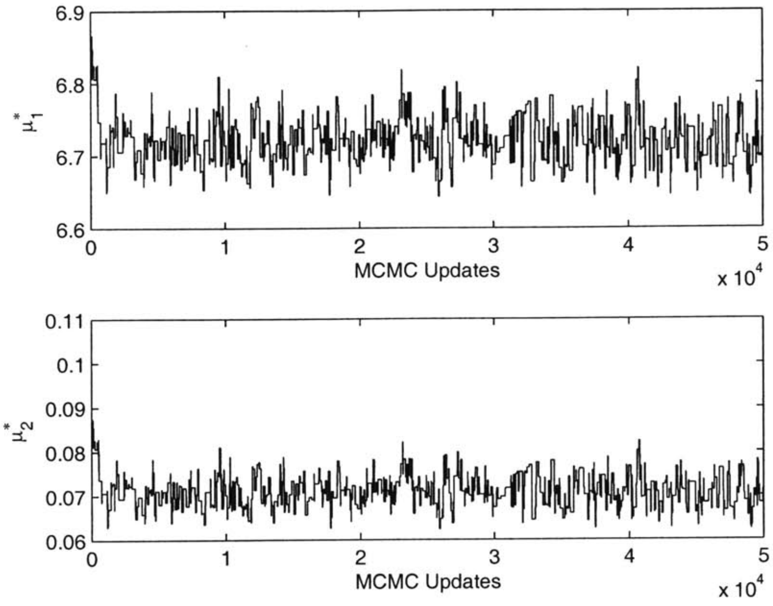

Figure 5-3: Markov Chain samples from posterior probability distribution for the Arrhenius parameters obtained from measurements $\bar{o}_{0.5\%}(\boldsymbol{\mu}_e)$ (refer to Table 5.2 for measurement values).

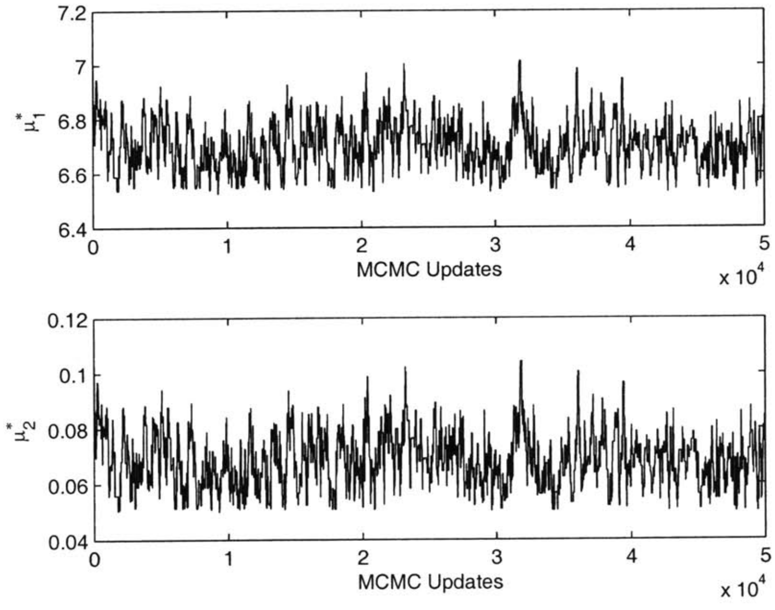

Figure 5-4: Markov Chain samples from posterior probability distribution for the Arrhenius parameters obtained from measurements $\bar{o}_{1.5\%}(\boldsymbol{\mu}_e)$ (refer to Table 5.2 for measurement values).

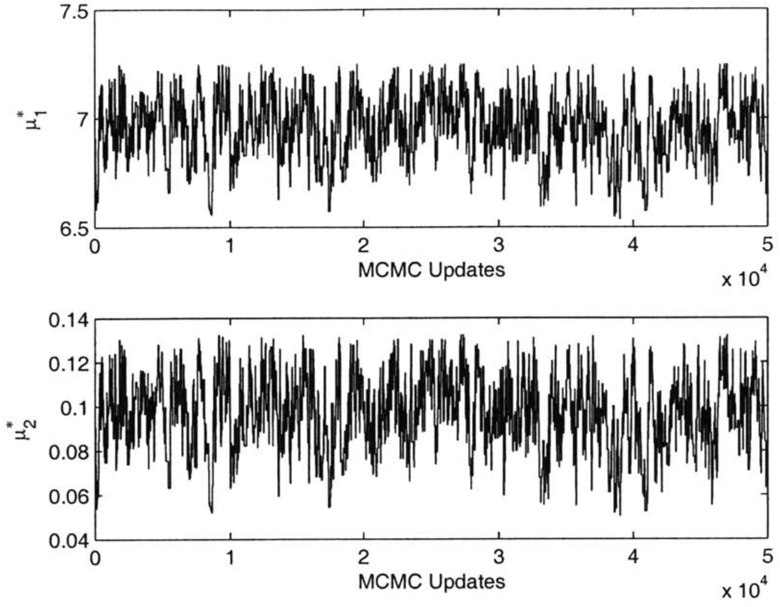

Figure 5-5: Markov Chain samples from posterior probability distribution for the Arrhenius parameters obtained from measurements $\bar{\boldsymbol{o}}_{2.5\%}(\boldsymbol{\mu}_e)$ (refer to Table 5.2 for measurement values).

conclude that the Markov chains computed in this section are adequate for obtaining reliable posterior probability distributions for the unknown input parameters of the inverse problem.

Table 5.4 shows the 95% confidence intervals for the mean and standard deviation of the input parameter $\boldsymbol{\mu}^*$ for the three sets of measurements shown in Table 5.2. The confidence intervals have been computed using the estimators $\bar{f}_{i\,N_m}$, $i = 1, 2$ and methodology described in Section 5.3.2.

The marginal posterior histograms for $\mu_1^* = \log(A^*)$ and $\mu_2^* = E^*$ for each set of measurements are shown in Figures 5-8 through 5-10. The increase in spread of the three histograms shows that the uncertainty associated with $\boldsymbol{\mu}^*$ increases as the measurements become more inaccurate. It is also worth noting that the mean values associated with the distribution of $\boldsymbol{\mu}^*$ deviate from the original values of $\boldsymbol{\mu}_e$ used to generate the exact outputs as the measurement errors are increased. Note that if we selected one measurement accuracy (say $\sigma_{1.5\%} = 3.54 \cdot 10^{-3}$) and generated $K$ sets of measurements (with $K >> 1$) using this empirical error, the average of the resulting

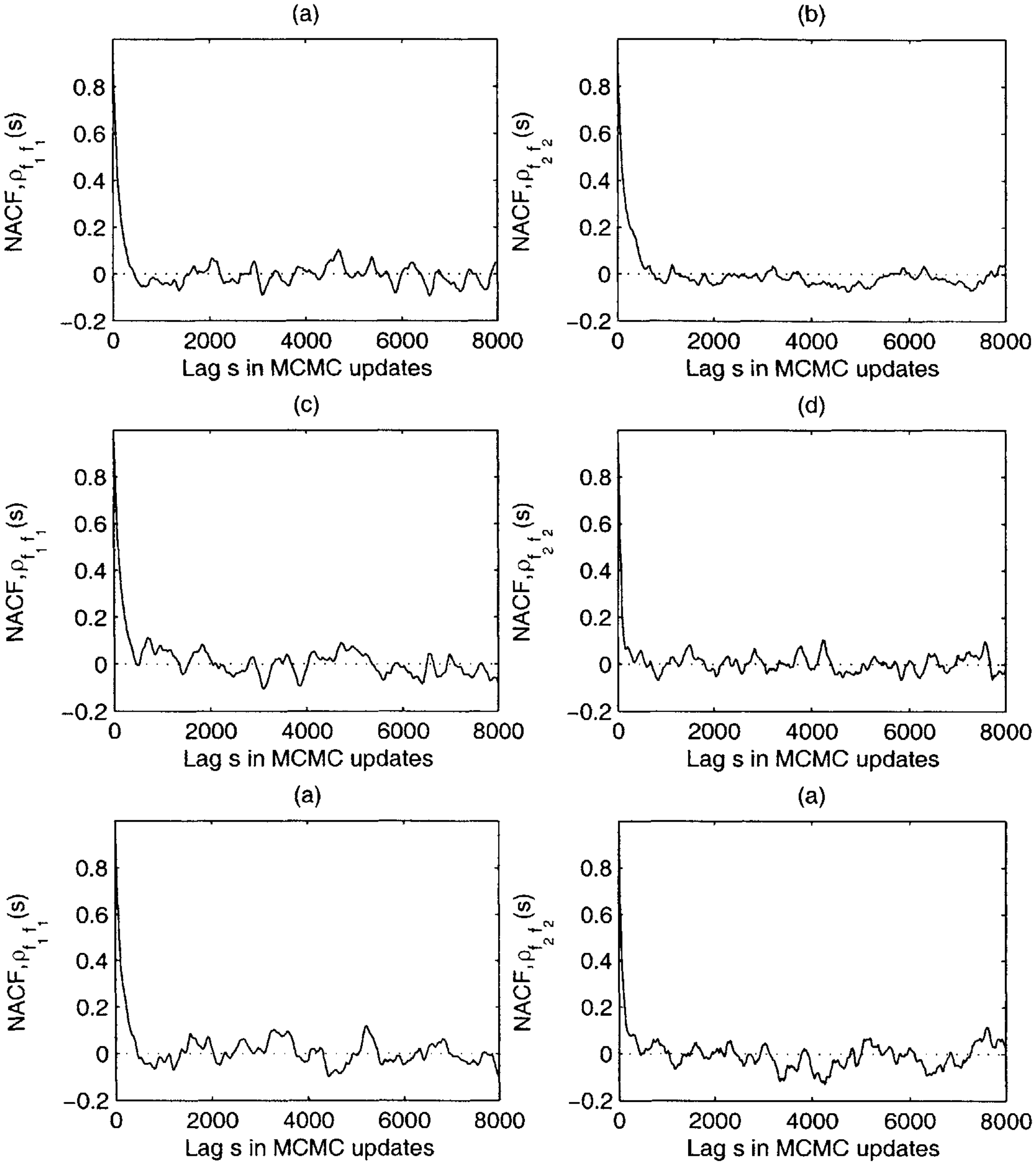

Figure 5-6: Normalized autocovariance functions (NACFs) for input parameter $\mu_1^* = \log(A^*)$. Plots (a) and (b) show the NACFs for statistics $\langle f_1(Y)\rangle$ (mean) and $\langle f_2(Y)\rangle$ (variance) respectively computed using the Markov chain obtained for the first set of empirical measurements, $\bar{o}_{0.5\%}(\boldsymbol{\mu}_e)$. Similarly, plots (c) and (d) correspond to the NACFs of the mean and variance for the second set of measurements, $\bar{o}_{1.5\%}(\boldsymbol{\mu}_e)$, and plots (e) and (f) show the NACFs of the mean and variance for the third set of measurements, $\bar{o}_{2.5\%}(\boldsymbol{\mu}_e)$.

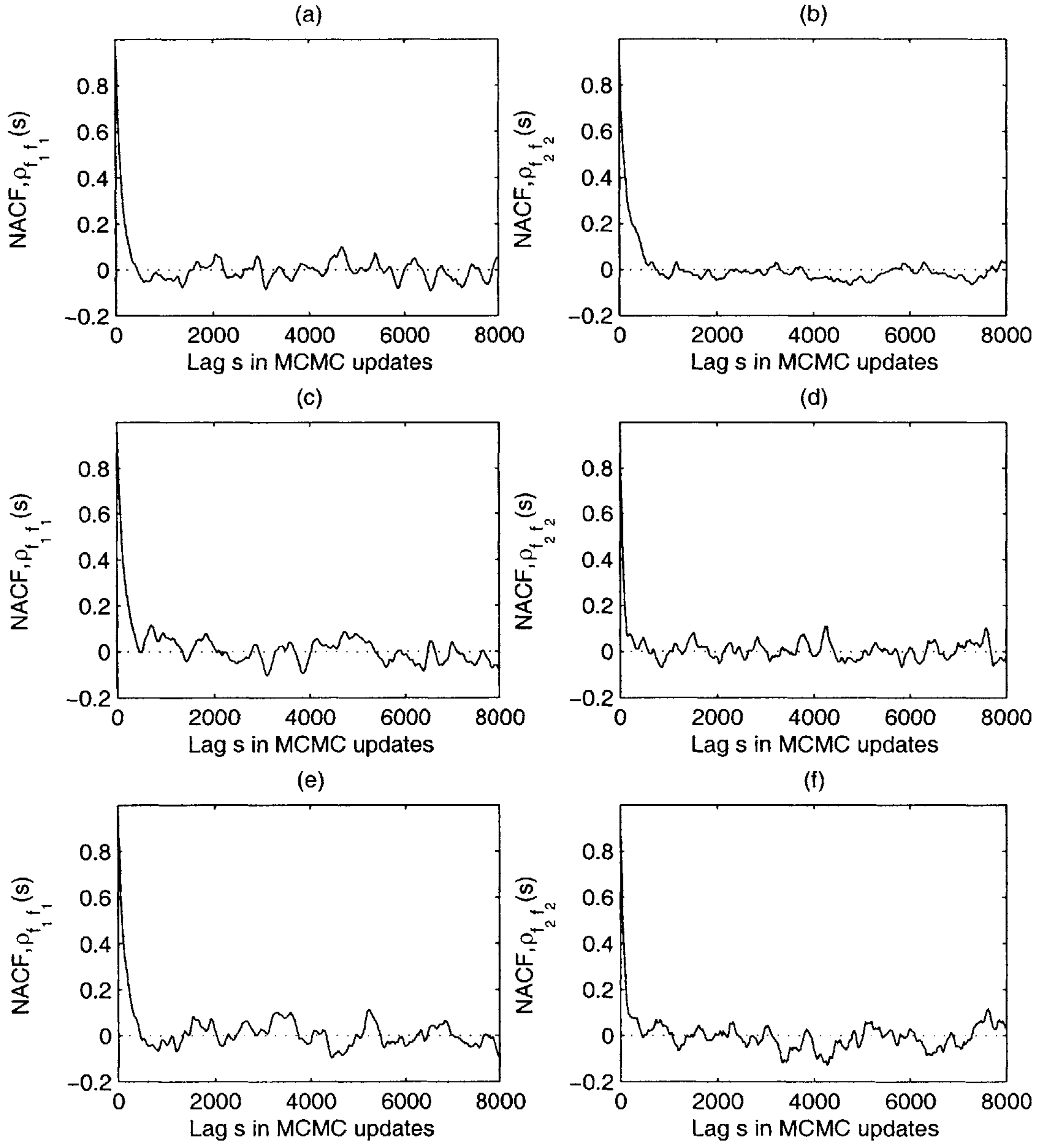

Figure 5-7: Normalized autocovariance functions (NACFs) for input parameter $\mu_2^* = E^*$. Plots (a) and (b) show the NACFs for statistics $\langle f_1(Y) \rangle$ (mean) and $\langle f_2(Y) \rangle$ (variance) respectively computed using the Markov chain obtained for the first set of empirical measurements, $\bar{o}_{0.5\%}(\boldsymbol{\mu}_e)$. Similarly, plots (c) and (d) correspond to the NACFs of the mean and variance for the second set of measurements, $\bar{o}_{1.5\%}(\boldsymbol{\mu}_e)$, and plots (e) and (f) show the NACFs of the mean and variance for the third set of measurements, $\bar{o}_{2.5\%}(\boldsymbol{\mu}_e)$.

|  | Mean |  | Standard Deviation |  |
|---|---|---|---|---|
| Meas. Set | mean($\mu_1^*$) | mean($\mu_2^*$) | std($\mu_1^*$) | std($\mu_2^*$) |
| $\bar{o}_{0.5\%}(\boldsymbol{\mu}_e)$ | $6.720 \pm 0.004$ | $0.0708 \pm 0.0005$ | $0.030 \pm 0.003$ | $0.0034 \pm 0.0004$ |
| $\bar{o}_{1.5\%}(\boldsymbol{\mu}_e)$ | $6.70 \pm 0.013$ | $0.068 \pm 0.0015$ | $0.086 \pm 0.007$ | $0.0095 \pm 0.0007$ |
| $\bar{o}_{2.5\%}(\boldsymbol{\mu}_e)$ | $6.95 \pm 0.02$ | $0.097 \pm 0.0025$ | $0.15 \pm 0.01$ | $0.017 \pm 0.0012$ |

Table 5.4: 95% confidence intervals for the mean and standard deviation of the posterior probability distribution of the input parameter $\boldsymbol{\mu}^*$ for the three sets of measurements shown in Table 5.2. Exact outputs (without measurement errors) were computed using $\boldsymbol{\mu}_e = (6.7386, 0.0727)$. In all cases, the Markov Chain used to infer the posterior probability distribution of $\boldsymbol{\mu}^*$ contained 50,000 samples.

predictions given by $\frac{1}{K}\sum_{k=1}^{K} \boldsymbol{\mu}_k^*$ should coincide with $\boldsymbol{\mu}_e$ (since the measurement errors are normally distributed with zero mean). However, in this study we are only considering a single set of measurements for each value of the empirical error so it is not possible to estimate how close to $\boldsymbol{\mu}_e$ the predicted values of $\boldsymbol{\mu}^*$ should be (although it seems reasonable to expect that large empirical errors will result in predicted values that are further away from the original input $\boldsymbol{\mu}_e$ used to generate the outputs).

As discussed previously, the Markov chains used to infer the posterior probability distribution for $\boldsymbol{\mu}^*$ contained 50,000 samples. Therefore, generating the Markov chains for each set of measurements required computing 50,000 forward solves. It is apparent that a reduced basis approximation is a critical ingredient of the probabilistic approach presented in this section since computing 50,000 "truth" FE forward solves for every set of measurements is a daunting task. However, the use of a reduced order model for computing the MCMC updates may raise some concerns regarding the effect that the error between the reduced basis approximation and the "truth" FE solution may have on the computed Markov chain and, consequently, on the statistics of interest inferred from such chain. Herein we address this issue by using the "truth" FE model in order to generate the Markov chain for the second set of measurements, $\bar{o}_{1.5\%}(\boldsymbol{\mu}_e)$, and comparing the resulting confidence intervals for the mean and standard deviation of $\boldsymbol{\mu}^*$ with those predicted using the reduced order model. Since the FE model is several orders of magnitude more expensive to run, the Markov chain in this case is limited to 10,000 samples instead of the 50,000 samples computed for the

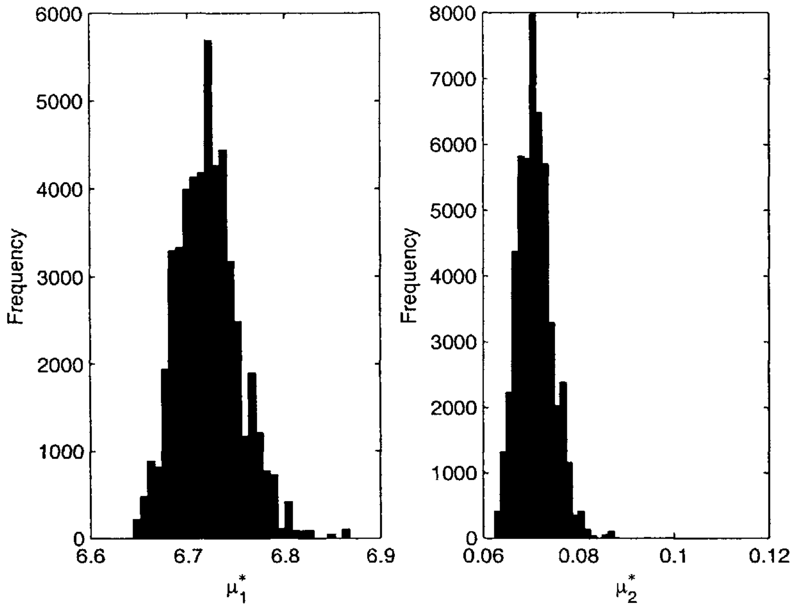

Figure 5-8: Marginal posterior histograms for $\mu_1^* = \log(A^*)$ and $\mu_2^* = E^*$ obtained from measurements $\bar{o}_{0.5\%}(\boldsymbol{\mu}_e)$ (refer to Table 5.2 for measurement values). Note that $\boldsymbol{\mu}^* = (\mu_1^*, \mu_2^*)$ can vary within the system parameters domain $\mathcal{D} \equiv [5.00, 7.25] \times [0.05, 0.15] \subset \mathbb{R}^2$

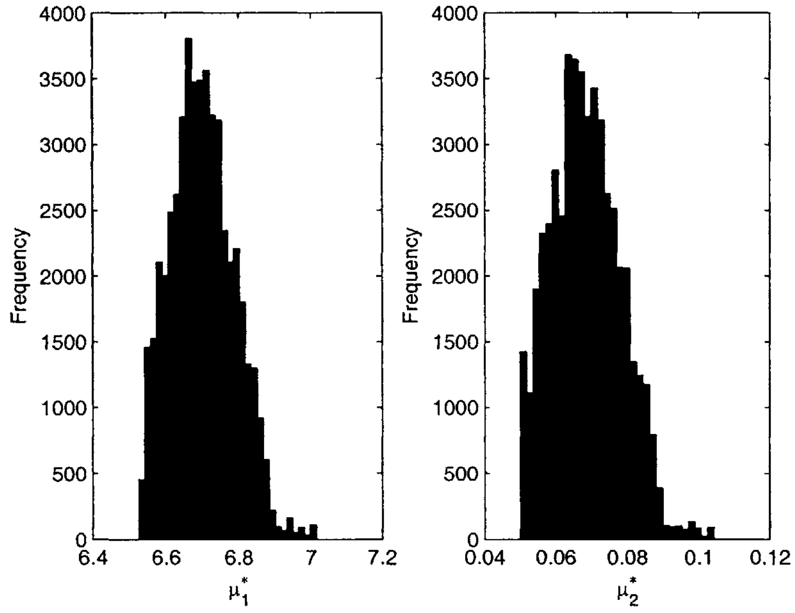

Figure 5-9: Marginal posterior histograms for $\mu_1^* = \log(A^*)$ and $\mu_2^* = E^*$ obtained from measurements $\bar{o}_{1.5\%}(\boldsymbol{\mu}_e)$ (refer to Table 5.2 for measurement values). Note that $\boldsymbol{\mu}^* = (\mu_1^*, \mu_2^*)$ can vary within the system parameters domain $\mathcal{D} \equiv [5.00, 7.25] \times [0.05, 0.15] \subset \mathbb{R}^2$

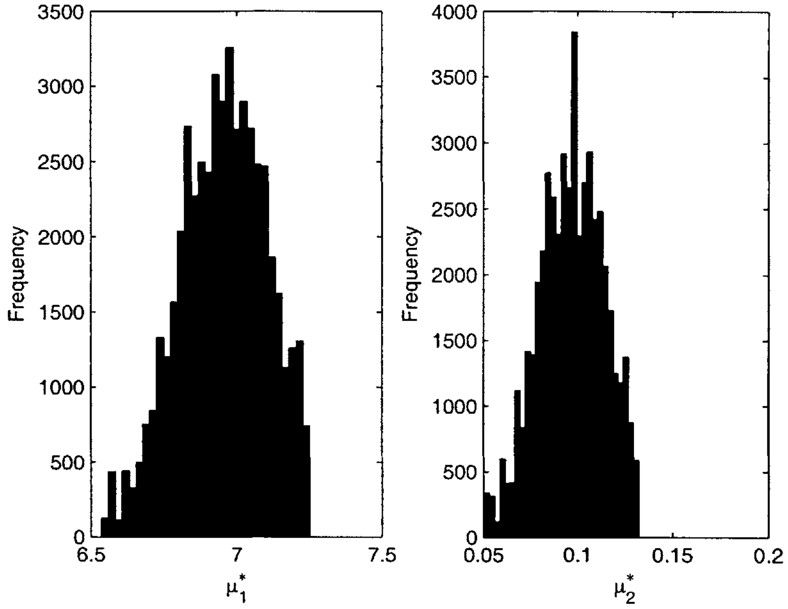

Figure 5-10: Marginal posterior histograms for $\mu_1^* = \log(A^*)$ and $\mu_2^* = E^*$ obtained from measurements $\bar{o}_{2.5\%}(\boldsymbol{\mu}_e)$ (refer to Table 5.2 for measurement values). Note that $\boldsymbol{\mu}^* = (\mu_1^*, \mu_2^*)$ can vary within the system parameters domain $\mathcal{D} \equiv [5.00, 7.25] \times [0.05, 0.15] \subset \mathbb{R}^2$

reduced basis chains shown earlier in this section. Even after reducing the number of samples, the FE chain took $4 \cdot 10^5$ seconds to compute whereas computing the RB chain with $50,000$ samples only required $5 \cdot 10^2$ seconds on the same machine. All the convergence criteria discussed in Section 5.3.2 were applied to the FE chain in order to guarantee its convergence to its equilibrium distribution prior to computing the statistics of interest. The main effect of using only $10,000$ samples instead of $50,000$ is that the resulting confidence intervals for the mean and standard deviation of the posterior distribution for $\boldsymbol{\mu}^*$ are wider than those obtained using the reduced basis chains. However, the results are still valid for the purposes of assessing the validity of the estimates obtained using the reduced basis Markov chains.

Table 5.5 shows the 95% confidence intervals for the mean and standard deviation of the distribution of $\boldsymbol{\mu}^*$ using the reduced basis chain and the "truth" FE chain. As expected, the confidence intervals computed using the FE chain are wider than those obtained from the RB chain due to the different number of samples used. However, the

|  | Mean |  | Standard Deviation |  |
|---|---|---|---|---|
| Chain | mean($\mu_1^*$) | mean($\mu_2^*$) | std($\mu_1^*$) | std($\mu_2^*$) |
| FE | $6.69 \pm 0.036$ | $0.067 \pm 0.004$ | $0.083 \pm 0.015$ | $0.0092 \pm 0.0016$ |
| RB | $6.70 \pm 0.013$ | $0.068 \pm 0.0015$ | $0.086 \pm 0.007$ | $0.0095 \pm 0.0007$ |

Table 5.5: 95% confidence intervals for the mean and standard deviation of the posterior probability distribution of the input parameter $\boldsymbol{\mu}^*$ for the second set of measurements shown in Table 5.2. First row corresponds to the confidence intervals computed using a Markov chain with 10,000 samples generated using the "truth" FE model whereas second row contains the results from a Markov chain with 50,000 samples generated using the reduced basis approximation. Note that computing the FE chain required $4 \cdot 10^5$ seconds versus only $5 \cdot 10^2$ seconds for the RB chain with 5 times more samples.

confidence intervals obtained from the RB chain are completely contained within the FE intervals, indicating that the results from both MCMC simulations are statistically identical. The results shown in Table 5.5 are plotted using error bars in Figures 5-11 and 5-12 for visualization purposes.

From Table 5.5 and Figures 5-11 and 5-12 we can conclude that the results obtained using the reduced order model for running the MCMC simulations are statistically identical to those obtained using the "truth" FE model. Furthermore, the reduced order model developed in this thesis is capable of achieving the same inverse problem results as the FE model approximately 4000 times faster. This addresses the issue of the high computational cost of high-fidelity models when implementing the probabilistic framework for solving inverse problems and, therefore, opens the door for dealing with a class of problems that could not be tackled until now.

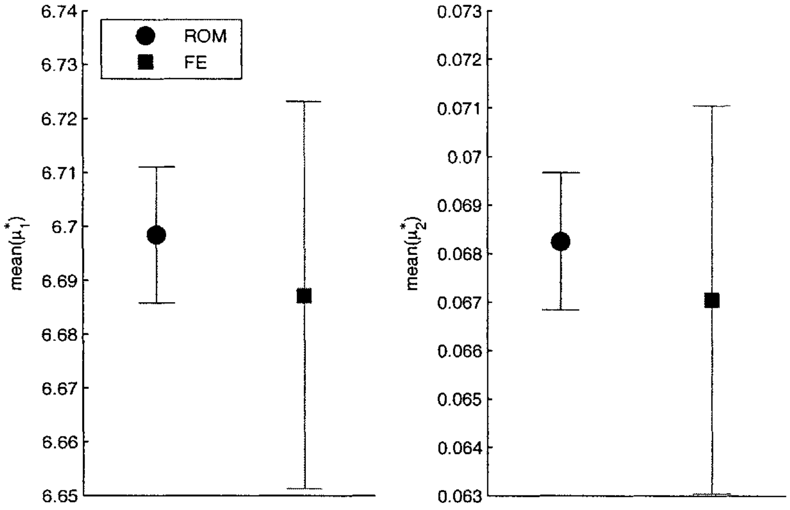

Figure 5-11: 95% confidence intervals for the mean of $\mu_1^* = \log(A^*)$ and $\mu_2^* = E^*$ for the second set of measurements shown in Table 5.2. Large intervals were computed using a Markov chain with 10,000 samples generated using the "truth" FE model. Smaller intervals were computed from a Markov chain with 50,000 samples generated using the reduced basis approximation presented in Chapters 3 and 4 with $N = 40$ and $M = 50$.

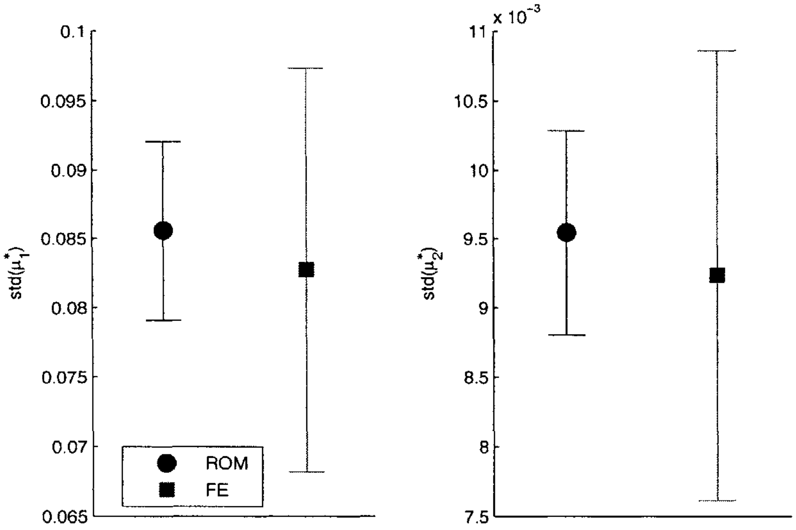

Figure 5-12: 95% confidence intervals for the standard deviation of the posterior distributions of $\mu_1^* = \log(A^*)$ and $\mu_2^* = E^*$ for the second set of measurements shown in Table 5.2. Large intervals were computed using a Markov chain with 10,000 samples generated using the "truth" FE model. Smaller intervals were computed from a Markov chain with 50,000 samples generated using the reduced basis approximation presented in Chapters 3 and 4 with $N = 40$ and $M = 50$.

# Chapter 6

# Conclusions and Future Work

## 6.1 Summary

The main goal of this thesis is to develop an efficient model order reduction methodology for highly nonlinear parametrized partial differential equations (PDEs) and demonstrate its application to the stochastic solution of inverse problems that are computationally intractable when using traditional numerical techniques such as the finite element method (FEM), finite difference method (FDM) or boundary element method (BEM).

The model order reduction technique chosen in this thesis is the reduced basis method. By using a coefficient-function approximation that represents the nonlinear terms of the PDE, this method is well suited for obtaining an efficient offline-online implementation where the operation count of the online stage is independent of the number of degrees of freedom of the original high-fidelity model. The reduced basis method seeks to represent any solution to the PDE of interest as a linear combination of solutions, often referred to as "snapshots", that have been precomputed using any classical numerical technique. Efficient computation of snapshots such that the resulting reduced basis approximation reproduces the input-output behavior of the original high-fidelity, large-scale system over a wide range of input parameter values without incurring in excessive offline cost is a challenging problem that is successfully addressed in this thesis. The input parameter space is sampled using a greedy algorithm that seeks to improve the accuracy of an existing reduced order model by adaptively adding a new snapshot at every iteration. Every new snapshot is sampled at the location in parameter space where the error between the "truth" and reduced basis outputs is maximal. This location is found by solving a PDE-constrained optimization problem in the continuous input parameter space. The optimization problem is formulated using a reduced Newton method that scales well with the number of input parameters.

The application used to demonstrate the performance of the efficient reduced basis approximation and sampling algorithm is a combustion problem governed by a two-dimensional convection-diffusion-reaction PDE which presents the challenges of highly nonlinear behavior and outputs of interest that must be evaluated over a wide range of input parameters.

The exponential Arrhenius term in the convection-diffusion-reaction equation considered in this thesis comes from an empirical law and the value of the parameters that appear in it are generally not known *a priori*. Therefore, the underlying PDE is naturally parametrized in terms of the Arrhenius parameters. Given a particular value for the Arrhenius parameters, the field of molar concentrations of fuel inside the combustion chamber is obtained by solving the governing PDE. The outputs of interest are the average concentrations of fuel along vertical lines located throughout the computational domain, which model the location of laser measurement sensors in the physical combustion chamber.

The forward problem, which is tackled in the first part of this thesis, consists in computing the outputs of interest given a know input vector of Arrhenius parameters. Chapter 6 of this thesis considers the more interesting inverse problem of inferring the value of the Arrhenius parameters given a set of empirical measurements obtained from a particular combustion reaction. The inverse problem is first formulated using a deterministic approach which gives the most likely value of the unknown input parameters by solving an optimization problem that minimizes the discrepancy between the empirical observations and the numerical outputs predicted by an analytical model. In a realistic setting where the knowledge of the solution field is incomplete due to sparsity of the measurements and presence of experimental errors, the deterministic approach is not capable of estimating the uncertainty associated with the inverse problem solution, which clearly hampers the use of this methodology in real applications. In order to overcome these limitations the inverse problem is formulated using the probabilistic Bayesian framework. This methodology samples the posterior probability distribution of the unknown input parameters using Markov chain Monte Carlo simulations. Even though the probabilistic approach is several orders of magnitude more expensive than the deterministic methodology, it is capable of incorporating experimental errors into the solution process, therefore giving insight into the uncertainty associated with the inverse problem results. In this thesis, the probabilistic approach is implemented using a reduced basis approximation for computing the forward solves required by the algorithm, therefore dramatically decreasing its computational complexity.

## 6.2 Conclusions

Several conclusions can be drawn from the forward problem results obtained in this thesis. First, the use of a coefficient-function approximation yields a reduced order model that is efficient even for high nonlinearities such as the ones found in combustion problems. The reduced basis approximation developed in this thesis is capable of computing results up to $50,000$ times faster than the original high-fidelity finite element model with average relative errors of magnitude $\mathcal{O}(10^{-6})$. Second, the model-constrained greedy adaptive sampling algorithm used for sampling the input parameter space scales well as the number of parameters increases. Even though the model-constrained greedy sampling method is more expensive than conventional grid-based sampling for the two-parameter case, the performances of both methods become comparable for the three-parameter case and the greedy sampling algorithm is expected to be faster than grid-based methods for higher dimensional input spaces. Furthermore, the model-constrained greedy sampling method offers the significant advantage of providing an error estimate for the reduced order model at every greedy cycle whereas conventional grid-based sampling offers no error indication to adequately size the reduced order model such that it meets our accuracy requirements without incurring in excessive computational online cost. Third, even though in general it is not possible to compute a rigorous error bound for nonlinear PDEs, the error estimator proposed in this thesis provides an acceptable indication of the accuracy of the reduced basis approximation at every greedy cycle.

Regarding the inverse problem results, the use of the model order reduction methodology discussed in this thesis in order to compute the samples of the Markov chains needed in the Bayesian framework enables the application of the probabilistic approach to a class of inverse problems that could not be previously tackled. The results obtained using the reduced order model for running the Markov chain Monte Carlo simulations are $4,000$ faster to compute and statistically identical to those obtained by running the simulations using a high-fidelity finite element model.

## 6.3 Future Work

The list below includes suggestions for improvements and extensions of the work presented in this thesis to new applications.

1. Extend the reduced basis approximation and sampling algorithm used in this thesis to systems of unsteady PDEs. Simulation of complex reactions with multiple species is a challenging task in combustion modeling which could be addressed by applying the same methodology used for solving the two-species, steady PDE considered in this thesis.

2. Test the model-constrained greedy adaptive algorithm used in this thesis on input spaces with more than three parameters. A particularly interesting application would be the case where the nonlinear PDE contains an input parameter defined as a continuous (infinite-dimensional) field.

3. Develop an efficient algorithm for generating initial guesses to be used by the model-constrained greedy adaptive sampling algorithm. In this thesis, the sam-

pling algorithm was initialized using an ad-hoc, problem-dependent process. Since the initial guess used by the sampling algorithm has a significant impact on the convergence of the optimization solver to the location of maximum output error, it is important to develop a robust and efficient method for computing adequate initial guesses.

4. Explore the use of an error estimator instead of the true output error in order to reduce the computational cost of the model-constrained greedy adaptive sampling algorithm. Defining an error estimator that is independent of the high-fidelity finite element output will eliminate the need to compute full-order solves at every greedy cycle, therefore increasing the speed of the algorithm.

5. Perform a thorough analysis to quantify the effect of the reduced basis errors on the probabilistic inverse problem solution. Since an accurate reduced order model was used in this thesis, the reduced basis errors were neglected in the Bayesian formulation without affecting the final results. However, incorporating this error in the formulation would be of paramount importance in situations that required using a faster, and therefore less accurate, reduced order model.

# Bibliography

[1] V. Akcelik, G. Biros, O. Ghattas, J. Hill, D. Keyes, and B. van Bloemen Waanders. Parallel algorithms for PDE-constrained optimization. In M. Heroux, P. Raghaven, and H. Simon, editors, *Frontiers of Parallel Computing.* SIAM, 2006.

[2] S. R. Arridge, J. P. Kaipio, V. Kolehmainen, M. Schweiger, E. Somersalo, T. Tarvainen, and M. Vauhkonen. Approximation errors and model reduction with an application in optical diffusion tomography. *Inverse Problems*, 22:175–195, 2006.

[3] M. Barrault, Y. Maday, N. C. Nguyen, and A. T. Patera. An "empirical interpolation" method: Application to efficient reduced-basis discretization of partial differential equations. *C. R. Acad. Sci. Paris, Série I*, 339(9):667–672, November 2004.

[4] R. Becker, M. Braack, and B. Vexler. Numerical parameter estimation for chemical models in multidimensional reactive flows. *Combustion Theory and Modelling*, 8:661–682, 2004.

[5] G. Biros and O. Ghattas. Parallel Lagrange-Newton-Krylov-Schur methods for PDE-constrained optimization. Part I: The Krylov-Schur solver. *SIAM J. Sci. Comput.*, 27(2):687–713, 2005.

[6] A. N. Brooks and T. J. R. Hughes. Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations. *Comput. Meths. Appl. Mech. Engrg.*, 32:199–259, 1982.

[7] C. G. Broyden. The convergence of a class of double-rank minimization algorithms. *J. Inst. Maths. Applics.*, 6:76–90, 1970.

[8] T. Bui-Thanh. *Model-Constrained Optimization Methods for Reduction of Parameterized Large-Scale Systems.* PhD thesis, Massachusetts Institute of Technology, May 2007.

[9] T. Bui-Thanh, K. Willcox, and O. Ghattas. Model reduction for large-scale systems with high-dimensional parameteric input space. *SIAM J. Sci. Comp.*, 2007. Submitted.

[10] R. Codina. Comparison of some finite element methods for solving the diffusion-convection-reaction equation. *Comput. Meths. Appl. Mech. Engrg.*, 156(1-4):185–210, 1998.

[11] R. Codina. On stabilized finite element methods for linear systems of convection-diffusion-reaction equations. *Comput. Meths. Appl. Mech. Engrg.*, 188(1-3):61–82, 2000.

[12] T. F. Coleman and Y. Li. An interior trust region approach for nonlinear minimization subject to bounds. *SIAM Journal on Optimization*, 6:418–445, 1996.

[13] J. Donea and A. Huerta. *Finite Element Methods for Flow Problems.* John Wiley & Sons Ltd, 2003.

[14] P. Englezos and N. Kalogerakis. *Applied Parameter Estimation for Chemical Engineers.* CRC, 2000.

[15] P. Feldmann and R. W. Freund. Efficient linear circuit analysis by Padé approximation via the Lanczos process. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 14:639–649, 1995.

[16] M. N. Fienen, P. K. Kitanidis, D. Watson, and P. Jardine. An application of Bayesian inverse methods to vertical deconvolution of hydraulic conductivity in a heterogeneous aquifer at Oak Ridge National Laboratory. *Mathematical Geology*, 36(1):101–126, 2004.

[17] R. Fletcher. A new approach to variable metric algorithms. *Computer Journal*, 13:317–322, 1970.

[18] K. Gallivan, E. Grimme, and P. Van Dooren. Padé approximation of large-scale dynamic systems with Lanczos methods. In *Proceedings of the 33rd IEEE Conference on Decision and Control*, 1994.

[19] C. J. Geyer. Practical Markov chain Monte Carlo. *Statistical Science*, 7(4):473–511, 1992.

[20] R. G. Ghanem and A. Doostan. On the construction and analysis of stochastic models: Characterization and propagation of the errors associated with limited data. *Journal of Computational Physics*, 217:63–81, 2006.

[21] D. Goldfarb. A family of variable metric updates derived by variational means. *Mathematics of Computing*, 24:23–26, 1970.

[22] W. H. Green and D. A. Schwer. Adaptive chemistry. In K. J. Bathe, editor, *Computational Fluid and Solid Mechanics: Proceedings of the First MIT Conference on Computational Fluid and Solid Mechanics, June 12-15, 2001.* Elsevier, 2001.

[23] M. A. Grepl. *Reduced-Basis Approximations and A Posteriori Error Estimation for Parabolic Partial Differential Equations.* PhD thesis, Massachusetts Institute of Technology, May 2005.

[24] M. A. Grepl, Y. Maday, N. C. Nguyen, and A. T. Patera. Efficient reduced-basis treatment of nonaffine and nonlinear partial differential equations. *Mathematical Modelling and Numerical Analysis*, 41(3):575+, May-June 2007.

[25] M. A. Grepl and A. T. Patera. *A posteriori* error bounds for reduced-basis approximations of parametrized parabolic partial differential equations. *Mathematical Modelling and Numerical Analysis*, 39(1):157–181, 2005.

[26] E. Grimme. *Krylov Projection Methods for Model Reduction.* PhD thesis, Coordinated-Science Laboratory, University of Illinois at Urbana-Champaign, 1997.

[27] W. K. Hastings. Monte Carlo sampling methods using Markov Chains and their applications. *Biometrika*, 57(1):97–109, 1970.

[28] P. Holmes, J. L. Lumley, and G. Berkooz. *Turbulence, Coherent Structures, Dynamical Systems and Symmetry.* Cambridge University Press, Cambridge, UK, 1996.

[29] J. P. Kaipio, V. Kolehmainen, E. Somersalo, and M. Vauhkonen. Statistical inversion and Monte Carlo sampling methods in electrical impedance tomography. *Inverse Problems*, 16(5):1487–1522, 2000.

[30] J. P. Kaipio and E. Somersalo. Statistical inverse problems: Discretization, model reduction and inverse crimes. *Journal of Computational and Applied Mathematics*, 198:493–504, 2007.

[31] K. Karhunen. Zur spektral theorie stochasticher prozesse. *Ann. Acad. Sci. Fennicae, Ser. A1 Math. Phys.*, 34:1–7, 1946.

[32] M. Loève. Functiona aleatoire de second ordre. *Revue Science*, 84:195–206, 1946.

[33] Y. Maday, A. T. Patera, and G. Turinici. Global *a priori* convergence theory for reduced-basis approximation of single-parameter symmetric coercive elliptic partial differential equations. *C. R. Acad. Sci. Paris, Série I*, 331(2):153–158, July 2002.

[34] D. W. Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *SIAM J. Appl. Math.*, 11(2):431–444, 1963.

[35] Y. M. Marzouk, H. N. Najm, and L. A. Rahn. Stochastic spectral methods for efficient Bayesian solution of inverse problems. *Journal of Computational Physics*, 224:560–586, 2007.

[36] The MathWorks, Natick, MA. *MATLAB Optimization Toolbox 3. User's Guide*, 2007.

[37] M. D. McKay, R. J. Beckman, and W. J. Conover. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 21(2):239–245, May 1979.

[38] A. Mizukami. An implementation of the streamline upwind/Petrov-Galerkin method for linear triangular elements. *Comput. Meths. Appl. Mech. Engrg.*, 49:357–364, 1985.

[39] B. C. Moore. Principal component analysis in linear systems: Controllability, observability, and model reduction. *IEEE Transactions on Automatic Control*, (1):17–31, August 1981.

[40] K. Mosegaard and A. Tarantola. Probabilistic approach to inverse problems. In *International Handbook of Earthquake & Engineering Seismology (Part A)*, pages 237–265. Academic Press, 2002.

[41] N. C. Nguyen. *Reduced-Basis Approximation and A Posteriori Error Bounds for Nonaffine and Nonlinear Partial Differential Equations: Application to Inverse Analysis.* PhD thesis, National University of Singapore, Singapore-MIT Alliance, July 2005.

[42] N. C. Nguyen, A. T. Patera, and J. Peraire. A "best points" interpolation method for efficient approximation of parametrized functions. *Int. J. Numer. Meth. Engng.*, 2007. Accepted.

[43] N. C. Nguyen and J. Peraire. An efficient reduced-order modeling approach for nonlinear parametrized partial differential equations. *Int. J. Numer. Meth. Engng.* To appear.

[44] N. C. Nguyen, K. Veroy, and A. T. Patera. Certified real-time solution of parametrized partial differential equations. In S. Yip, editor, *Handbook of Materials Modeling*, pages 1523–1558. Springer, 2005.

[45] A. K. Noor and J. M. Peters. Reduced basis technique for nonlinear analysis of structures. *AIAA Journal*, 18(4):455–462, 1980.

[46] O. O. Oluwole, B. Bhattacharjee, J. E. Tolsma, B. I. Barton, and W. H. Green. Rigorous valid ranges for optimally-reduced kinetic models. *Combustion and Flame*, 146:348–365, 2006.

[47] A. T. Patera and G. Rozza. *Reduced Basis Approximation and A Posteriori Error Estimation for Parametrized Partial Differential Equations.* Version 1.0, Copyright MIT 2006. To appear in (tentative rubric) MIT Pappalardo Graduate Monographs in Mechanical Engineering, 2006.

[48] P. O. Persson and G. Strang. A simple mesh generator in matlab. *SIAM Review*, 46(2), 2004.

[49] C. Prud'homme, D. Rovas, K. Veroy, Y. Maday, A. T. Patera, and G. Turinici. Reliable real-time solution of parametrized partial differential equations: Reduced-basis output bound methods. *Journal of Fluids Engineering*, 124(1):70–80, March 2002.

[50] S. S. Ravindran. A reduced-order approach for optimal control of fluids using proper orthogonal decomposition. *Int. J. Numer. Meth. Fluids*, 34:425–448, 2000.

[51] M. Sambridge and K. Mosegaard. Monte Carlo methods in geophysical inverse problems. *Reviews of Geophysics*, 40:(3–1)–(3–29), 2002.

[52] F. Shakib, T. J. R. Hughes, and Z. Johan. A new finite element formulation for computational fluid dynamics: X. The compressible Euler and Navier-Stokes equations. *Comput. Meths. Appl. Mech. Engrg.*, 89(1-3):141–219, 1991.

[53] D. F. Shanno. Conditioning of quasi-newton methods for function minimization. *Mathematics of Computing*, 24:647–656, 1970.

[54] L. Sirovich. Turbulence and the dynamics of coherent structures. Part I: Coherent structures. *Quarterly of Applied Mathematics*, 45(3):561–571, October 1987.

[55] P. Smolinski. A variational formulation for the generalized Galerkin method for the convection-diffusion equation. *Comput. Meths. Appl. Mech. Engrg.*, 73:93–98, 1989.

[56] K. C. Sou, A. Megretski, and L. Daniel. A quasi-convex optimization approach to parameterized model-order reduction. In *IEEE/ACM Design Automation Conference*, June 2005.

[57] A. Tarantola. *Inverse Problem Theory and Methods for Model Parameter Estimation.* SIAM, Philadelphia, 2005.

[58] T. E. Tezduyar and Y. Osawa. Finite element stabilization parameters computed from element matrices and vectors. *Comput. Meths. Appl. Mech. Engrg.*, 190:411–430, 2000.

[59] T. E. Tezduyar and Y. J. Park. Discontinuity-capturing finite element formulations for nonlinear convection- diffusion-reaction equations. *Comput. Meths. Appl. Mech. Engrg.*, 59:307–325, 1986.

[60] A. Varga. Enhanced modal approach for model reduction. *Mathematical Modelling of Systems*, 1:91–105, 1995.

[61] K. Veroy and A. T. Patera. Certified real-time solution of the parametrized steady incompressible Navier-Stokes equations: Rigorous reduced-basis a posteriori error bounds. *International Journal for Numerical Methods in Fluids*, 47:773–788, January 2005.

[62] K. Veroy, C. Prud'homme, D. V. Rovas, and A. T. Patera. A posteriori error bounds for reduced-basis approximation of parametrized noncoercive and nonlinear elliptic partial differential equations (AIAA Paper 2003-3847). In *Proceedings of the 16th AIAA Computational Fluid Dynamics Conference*, June 2003.

[63] D. Wee, S. Park, T. Yi, A. M. Annaswamy, and A. F. Ghoniem. Model-based optimal active control of liquid-fueled combustion systems. In G. D. Roy, editor, *Combustion Processes in Propulsion*, pages 201–210, Burlington, 2005. Butterworth-Heinemann.

[64] M. W. Ziesse, H. G. Bock, J. V. Gallitzendörfer, and J. P. Schlöder. Parameter estimation in multispecies transport reaction systems using parallel algorithms. In J. Gottlieb and P. DuChateau, editors, *Parameter Identification and Inverse Problems in Hydrology, Geology and Ecology.* Springer, 1996.