Model-Constrained Optimization Methods for Reduction of Parameterized Large-Scale Systems

by

Tan Bui-Thanh
B.Eng., Ho Chi Minh City University of Technology, Vietnam (2001)
M.Eng., Singapore-MIT Alliance, Singapore (2003)

Submitted to the Department of Aeronautics and Astronautics
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy in Aeronautics and Astronautics
at the
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
May 2007

Author .....
Department of Aeronautics and Astronautics
May 24, 2007

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

Certified by .....
Jacob White
Professor of Electrical Engineering and Computer Science
Thesis Committee

Certified by .....
Jaime Peraire
Professor of Aeronautics and Astronautics
Thesis Committee

Accepted by .....
Professor Jaime Peraire
Chairman, Department Committee on Graduate Students

Model-Constrained Optimization Methods for Reduction of Parameterized Large-Scale Systems

by

Tan Bui-Thanh

Submitted to the Department of Aeronautics and Astronautics
on May 24, 2007, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Aeronautics and Astronautics

Abstract

Most model reduction techniques employ a projection framework that utilizes a reduced-space basis. The basis is usually formed as the span of a set of solutions of the large-scale system, which are computed for selected values (samples) of input parameters and forcing inputs. In existing model reduction techniques, choosing where and how many samples to generate has been, in general, an ad-hoc process. A key challenge is therefore how to systematically sample the input space, which is of high dimension for many applications of interest.

This thesis proposes and analyzes a model-constrained greedy-based adaptive sampling approach in which the parametric input sampling problem is formulated as an optimization problem that targets an error estimation of reduced model output prediction. The method solves the optimization problem to find a locally-optimal point in parameter space where the error estimator is largest, updates the reduced basis with information at this optimal sample location, forms a new reduced model, and repeats the process. Therefore, we use a systematic, adaptive error metric based on the ability of the reduced-order model to capture the outputs of interest in order to choose the snapshot locations that are locally the worst case scenarios. The state-of-the-art subspace trust-region interior-reflective inexact Newton conjugate-gradient optimization solver is employed to solve the resulting greedy partial differential equation-constrained optimization problem, giving a reduction methodology that is efficient for large-scale systems and scales well to high-dimensional input spaces.

The model-constrained adaptive sampling approach is applied to a steady thermal fin optimal design problem and to probabilistic analysis of geometric mistuning in turbomachinery. The method leads to reduced models that accurately represent the full large-scale systems over a wide range of parameter values in parametric spaces up to dimension 21.

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

Acknowledgments

It is time for me to acknowledge and recognize the following people who have made this thesis possible.

I am indebted to my thesis advisor, Prof. Karen Willcox, for her constant guidance, encouragement, understanding and patience. I would not have had a chance to work on this wonderful topic without her advice and support before I applied to MIT. Preparing for the GREs was really hard for a non-native English speaker like me, but it was even harder to get admitted to MIT. During this dark period, she was one of my saviors who continuously gave me strength to overcome all the hardships. I am more than lucky to have such a very nice advisor who has been willing to forgive all the mistakes that I have made during my stay at MIT. Now I understand what she meant by saying: "I don't have children BUT I have students". I think a dozen pages is not enough for me to write down what I have owed her and what she has done for me, both in academic and in real life, since I was her master student. With that, I would like to thank my other master thesis advisor, Prof. Murali Damodaran of the Nanyang Technological University, for introducing Prof. Karen Willcox to me among all MIT Professors. He was the first person in this world who introduced CFD to me and strongly encouraged me to apply to MIT.

I would also like to thank my doctoral committee, including Prof. Jaime Peraire and Prof. Jacob White, for their valuable guidance and criticism. Their constructive comments have helped to improve my research both in quality and quantity. A special thanks goes to Prof. Anthony Patera for his invaluable suggestions during my research. Thanks to Prof. Omar Ghattas of the University of Texas at Austin for answering countless questions on optimization—especially for teaching me what I know about PDE-constrained optimization and about computing the Hessian-vector product on-the-fly using one forward and one adjoint solves—and for serving as a thesis reader. I would like to give a special thanks to Prof. David Darmofal at MIT for reading the thesis.

A big thanks goes to Prof. Tim Davis of the University of Florida for many

questions on his direct sparse solver software UMFPACK. His generosity in giving the UMFPACK software to public is fully appreciated. Without UMFPACK, I would have been still far way from completing the thesis. I would also like to thank Dr. Per-Olof Persson of MIT for countless questions on sparse matrices.

Back to six years ago when I just got the B.Eng degree in aeronautics from the Ho Chi Minh City University of Technology in Vietnam. I was almost hopeless to find a suitable job and I thought it was the end of the world. It was Prof. Nguyen Thien Tong who recognized my potential in research during my undergrad years, and it was him who introduced the Singapore-MIT Alliance (SMA) grad school to me and supported me to apply to this program as a master student in High Performance Computation for Engineered Systems. SMA connected me to many top researchers at MIT and one of which became my PhD advisor. Prof. Nguyen Thien Tong was another one who strongly encouraged me to apply to MIT. Prof. Nguyen Thien Tong, I consider you as a father who has led his son to a brighter future.

To the ACDLers: thanks to Cuong Nguyen for the friendship. I must thank Todd Oliver and Krzysztof Fidkowski for countless questions on the ProjectX code during my first year at MIT; thanks to Garrett Barter for so many questions on Linux and English; a student life is probably less colorful without beer and I would like to give a special thanks to Mike Park and Dan King for all the beers; the nicest person in ACDL is probably Jean Sofronas and I would like to thank her for all her help in organization and quietly working behind the scenes.

To all Vietnamese friends in Boston: I would like to thank you all for your encouragement. Through Chi Linh I knew Anh Hai, and it was Anh Hai who introduced me to the rest of my friends in Boston, including Hoa-Minh, Ngoc-Hanh, Van-Phuong, Thien-Dinh, Co Chu The, etc. In particular, I would like to thank Anh Hai for making my life in Boston colorful with his stereo karaoke system. Anh Hai, thanks for helping me move my furnitures many times, and for all the beers too!

To all Vietnamese friends in Springfield: I would like to thank you all for your encouragement (again it was Anh Hai who introduced me to these true friends). I am fortunate enough to meet such a diverse set of people whom I am proud to call

friends. In particular, I would like to thank Chu Thim Hieu for countless advices on my newborn baby and for delicious Vietnamese food. Thanks to Chu Thim Ba Van for their hospitality that makes me feel at home whenever I visit them. My trips to Springfield would have been very boring if Anh Tung (+ Thao) and Anh Danh (+ Thuy) had not turn on their super stereo karaoke system. Anh Danh and Anh Tung, thanks for the friendship and the beers!

To other friends in the US: thanks to everyone for your encouragement. Specifically, I would like to acknowledge Hop Pho, Kim Tu and Anh Vu for motivating me to study in the US. Who could ever forget the trip to Rochester, playing bowling, eating Buffalo chicken wings (probably the best I have ever had) and being waken up in the middle of the night because Anh Hai was snoring too loud!

To all friends back in Vietnam: thanks to everyone for all the funny jokes through our 12A1_ChauThanh_BR_1996 yahoo group. In particular, I would like to give a special thanks to Ngoc Xuan, Quang Minh and To Loan for frequently visiting my Mom when I am abroad.

My little family owes a special thanks to my Uncle, Uncle Larry Ng, for contributing so much to our life here in Boston. God really gave us such a very nice Uncle! Without his help, encouragement and protection, we could not have survived until today. Whenever we have difficulties, he is there to provide help. We have no relatives in Boston except Uncle Larry "because he is half Chinese and half Vietnamese" as he usually says. It is impossible to count all what he has done for us. Uncle Larry, your contribution in this thesis is substantial! We wonder how we could repay everything you have done for us!

My aunt's family—Chu Trung, Co Gai, and Ti Nho—have helped me so much by taking care of my Mom when I am abroad. Their love and encouragement have been continuously guiding me throughout my research. Ti Nho, I hope that you will forgive me for my strictness when we were together in Ho Chi Minh City. After all, I consider you as my only blood brother rather than a cousin.

I would also like to thank my parent-, my brother- and my sister-in-law for their encouragement. In particular, I appreciate my father-in-law's help in building my

new house back in Vietnam. A special thanks must go to Chi Hoa for taking care of my Mom when I am not in Vietnam.

If Prof. Karen Willcox is the savior in my academic life, my wife, Cong Minh Pham, is the savior in my real life. I guess saying thank-you is not enough but I could not find a better word. I really would like to thank my wife for sharing happiness and sadness during the course of my research. Whenever I am in bad mood because the code is not working, she is there to comfort and encourage me to go forward. I am sure that I would not have gotten this far without her support, patience, understanding and love. On the way home after every hard working day, I am looking forward to enjoying the delicious food which she has cooked for me with all her love. More importantly, I would like to thank my wife for giving me a handsome little prince, William Bui. For the past few months, without my wife and my little son I would have either become crazy or stopped working due to the intensive stress in finishing the thesis. It is amazing how a little three-month-old son can help his Dad release stress. Whenever I feel too stressful, I come and see my son; with a smile on his face, all my stresses go away!

Last but not least, I would like to deeply thank my Mom, an alone illiterate Mom who has managed to bring her son up all the way to his PhD. She has devoted all her life to me. Mom, you have instilled the love of learning in me by telling me the disadvantages that an illiterate has to face up to in his life. Since my son (your grandson) was born, my wife and I have realized how much hard it is to take care of him, and I could not imagine how you alone could bring me up; but I could imagine all the hardships that you have dealt with. Mom, you are my idol and I love you so much. This thesis is for you.

This work was supported in part by the Singapore-MIT Alliance Computational Engineering Programme, Universal Technology Corporation under contract number 04-S530-0022-07-C1, technical contract monitor Dr. Cross of Air Force Research Laboratory, and the Air Force Office of Sponsored Research grant FA9550-06-0271, program director Dr. Fahroo.

DEDICATION TO

my Mom, Ha Thi Man,

my wife, Cong Minh Pham,

my son, William Bui,

and my brother, Ho Bao Han

Contents

List of Figures

List of Tables

Nomenclature

G

Δ_k Trust region radius at the k^{th} Newton step, see equation (3.20), page 56

Δt Time step, see equation (2.21), page 45

$\emptyset$ Empty set, see equation (3.29), page 63

ϵ Desired level of accuracy, see equation (3.13), page 53

γ Full adjoint vector associated with the full initial condition, see equation (3.57), page 71

$\Gamma(\mathbf{A})$ *Numerical range* of the matrix $\mathbf{A}$, see equation (2.20), page 45

γ_r Reduced adjoint vector associated with the reduced initial condition, see equation (3.57), page 71

κ_i i^{th} thermal conductivity, see equation (4.0), page 77

λ Full adjoint vector associated with the full model equations, see equation (3.37), page 68

λ_r Reduced adjoint vector associated with the reduced model equations , see equation (3.37), page 68

ω Frequency, see equation (3.65), page 74

$\mathbf{\Phi}$ A basis of the reduced trial space, see equation (2.1), page 38

$\mathbf{\Psi}$ A basis of the reduced test space, see equation (2.1), page 38

ρ Density, see equation (5.2), page 108

σ_i Standard deviation, see equation (5.14), page 116

$\sigma_{min}^{\mathbf{A}}$ Smallest singular value of the system matrix $\mathbf{A}$, see equation (2.17), page 43

τ Some given tolerance, see equation (3.22), page 57

φ_k Merit function in the trust region subproblem at the k^{th} Newton step, see equation (3.20), page 56

R

$\mathbf{A}$ System matrix of the full model, see equation (2.5), page 40

$\mathbf{A}_r$ System matrix of the reduced model, see equation (2.9), page 40

$\mathbf{B}$ Input matrix of the full model, see equation (2.5), page 40

$\mathbf{B}_r$ Input matrix of the reduced model, see equation (2.9), page 40

Bi Biot number, see equation (4.0), page 78

$\mathbf{C}$ Output matrix of the full model, see equation (2.5), page 40

$\mathbf{C}_r$ Output matrix of the reduced model, see equation (2.9), page 40

C_L Lift coefficient, see equation (5.23), page 122

C_M Moment coefficient, see equation (5.23), page 122

$\mathbf{D}$ Diagonal scaling matrix, see equation (3.16), page 55

$\mathbf{D}^{CL}$ Coleman-Li diagonal scaling matrix, see equation (3.16), page 55

d Number of parameters, see equation (2.1), page 38

$\mathbf{E}$ Mass matrix of the full model, see equation (2.5), page 40

$\mathbf{e}$ Error between the full solution and its approximation, see equation (2.12), page 41

$\mathbf{E}_r$ Mass matrix of the reduced model, see equation (2.9), page 40

$\mathbf{F}^x$ Euler flux in x-direction, see equation (5.3), page 108

$\mathbf{F}^y$ Euler flux in y-direction, see equation (5.3), page 108

$\bar{g}$ Average geometric variation, see equation (5.14), page 116

$\mathcal{G}$ Cost functional or the objective function, see equation (3.1), page 51

g General blade geometry, see equation (5.14), page 116

g_n Nominal blade geometry, see equation (5.14), page 116

ℓ Linear functional, see equation (4.6), page 79

LU_f^F Number of flop counts incurred by the LU factorization for the full model, see equation (3.23), page 59

LU_s^F Number of flop counts for each triangular solve for the full model, see equation (3.23), page 59

m Reduced model size, see equation (2.1), page 38

$\hat{\mathbf{n}}$ Normal vector, see equation (4.2), page 79

n Full model size, see equation (2.1), page 38

n_G Number of greedy cycles, see equation (3.23), page 60

n_s Number of geometric mode shapes or the number of snapshots, see equation (5.14), page 116

n_t Number of time steps, see equation (2.21), page 45

n_{Hv} Number of Hessian-vector products, see equation (3.23), page 59

$\mathcal{P}$ Output operator, see equation (2.1), page 38

P Pressure, see equation (5.3), page 108

p Number of inputs, see equation (2.1), page 38

$\mathcal{Q}$ Error indicator, see equation (3.9), page 52

q Number of outputs, see equation (2.1), page 38

$\mathbf{R}$ Residual operator resulting from a numerical discretization of a set of PDEs, see equation (2.1), page 38

$\mathbf{R}^k$ Residual vector at the k^{th} time step, see equation (2.22), page 45

$\mathbb{R}$ Set of real numbers, see equation (2.1), page 38

$\mathbf{s}$ (Inexact) Newton step, see equation (3.18), page 55

$\mathcal{S}_k$ Trust region subspace at the k^{th} Newton step, see equation (3.20), page 56

t Time, see equation (2.1), page 38

t_f Time horizon of interest, see equation (3.13), page 53

$\mathbf{u}$ Time-dependent input vector, see equation (2.1), page 38

$\mathbf{u}^k$ Input vector at the k^{th} time step, see equation (2.21), page 45

$\mathcal{V}^m$ Reduced trial space, see equation (2.1), page 38

v_i i^{th} geometric mode shapes, see equation (5.14), page 116

$\mathcal{W}^m$ Reduced test space, see equation (2.1), page 38

$\tilde{\mathbf{x}}$ Approximate solution vector computed from the reduced model, see equation (2.1), page 38

$\mathbf{x}$ Solution (state) vector of the full model, see equation (2.1), page 38

$\mathbf{x}^0$ Specified initial state of the full model, see equation (2.1), page 38

$\mathbf{x}^k$ Full state vector at the k^{th} time step, see equation (2.21), page 45

$\mathbf{x}_r$ Solution (state) vector of the reduced model, see equation (2.2), page 38

$\mathbf{x}_r(0)$ Initial condition of the reduced model, see equation (2.3), page 39

$\mathbf{x}_r^k$ Reduced state vector at the k^{th} time step, see equation (2.22), page 45

$\tilde{\mathbf{x}}^k$ Approximation of the full solution at the k^{th} time step, see equation (2.22), page 45

x x-coordinate, see equation (5.3), page 108

$\mathbf{y}$ Output vector of the full model, see equation (2.1), page 38

$\mathbf{y}_r$ Output vector of the reduced model, see equation (2.3), page 39

y y-coordinate, see equation (5.3), page 108

$\mathbf{z}$ Vector containing the parameters of interest, see equation (2.1), page 38

$\mathbf{z}^*$ Maximizer of the greedy optimization problem, see equation (3.13), page 53

$\mathbf{z}^0$ Initial guess for the parameter vector $\mathbf{z}$ in the bound-constrained optimization solver, see equation (3.13), page 53

$\mathbf{z}_{max}$ Upper bound of the parameter vector $\mathbf{z}$, see equation (3.9), page 52

$\mathbf{z}_{min}$ Lower bound of the parameter vector $\mathbf{z}$, see equation (3.9), page 52

Chapter 1

Introduction

1.1 Motivation

Recent years have seen considerable progress in solution and optimization methods for partial differential equations (PDEs), leading to advances across a broad range of engineering applications, including computational fluid dynamics (CFD), structural dynamics, aeroelasticity, and large-scale optimization, to name a few. Improvements in both methodology and computing power have been substantial; however, a number of challenges remain to be addressed. In many cases, computational models for PDEs lead to large-scale systems of equations that are computationally expensive to solve, e.g. for applications such as optimal design or probabilistic analyses.

Model order reduction is a powerful tool that permits the systematic generation of cost-efficient representations of large-scale systems that result from discretization of PDEs. Several reduction methods have been developed, for example, modal truncation [2, 3], proper orthogonal decomposition (POD) [4, 5], balanced truncation [6], Krylov-subspace methods [7–9], reduced basis methods [10], and a quasi-convex optimization approach [11]. These methods have been applied in many different settings with considerable success, including controls [12, 13], fluid dynamics [4, 5], structural dynamics [14–19], and circuit design [20–22]. However, a number of open issues remain with these methods, including efficient model reduction techniques for systems with large input spaces, and for nonlinear systems.

Optimal design, optimal control [23, 24], probabilistic analysis [25] and inverse problem applications [26] present additional challenges for model reduction methods. In such cases—where the physical system must be simulated repeatedly—the availability of reduced models can greatly facilitate solution of the optimization problem, particularly for large-scale applications. To be useful for these applications, the reduced model must provide an accurate representation of the high-fidelity model over a wide range of parameters. In particular, discretization produces high-dimensional input spaces when the input parameters represent continuous fields (such as initial conditions, boundary conditions, distributed source terms, and geometric variability). Model reduction for high-dimensional input spaces remains a challenging problem. Approaches developed for dynamical systems, such as POD and Krylov-based methods, have been applied in an optimization context [21, 27, 28]; however, the number of parameters in the optimization application was small.

Nonlinearity of the underlying physics of problems at hand presents another challenge for projection-based model order reduction. The difficulty here is how to obtain an efficient reduced model. Even though the size of the reduced model is much smaller than that of the full model, in the nonlinear case it is not necessarily true that solving the reduced model is cheaper than solving the full model. For example, if the reduced matrices depend on the full model size and if forming (or evaluating) the full matrices is the dominant cost, then solving the reduced model may be more expensive than solving the full one because one has to first evaluate the full matrices before evaluating the reduced ones. If, on the other hand, the reduced matrices do not depend on the full model size, they can be evaluated once in the offline stage, and in the online stage, the cost of solving the reduced model is negligible. Therefore a reduced model must be not only valid for a range of parameters but also independent of the full model size.

1.2 Literature Review

1.2.1 Projection-Based Model Reduction Techniques

Most reduction techniques for large-scale systems employ a projection framework that utilizes reduced-space bases. The key challenge in projection-based model reduction is how to find a reduced basis such that the reduced system provides an accurate representation of the large-scale system over the desired range of inputs. Algorithms such as optimal Hankel model reduction [29–31] and balanced truncation [6] have been used widely throughout the controls community to generate reduced models with strong guarantees of quality. These algorithms can be carried out in polynomial time; however, the computational requirements make them impractical for application to large systems such as those arising from the discretization of PDEs, for which system orders typically exceed 10^4.

While considerable effort has been applied in recent years towards development of algorithms that extend balanced truncation to large-scale linear time-invariant (LTI) systems [32–35], efficient algorithms for very large systems remain a challenge. In addition, application of balanced truncation methods to systems that are linear time-varying or have parametric variation has been limited to small systems [36–38]. The Krylov-subspace methods [7–9] have been shown to be an alternative efficient model reduction method for large-scale LTI systems. Meanwhile, the POD method [4, 5, 39, 40] has emerged as a popular alternative for reduction of very large dynamical systems. POD has been used widely throughout CFD applications such as aeroelasticity [41, 42] and flow control [27, 28]. However, both Krylov-subspace and POD methods lack the quality guarantees of methods such as balanced truncation. As opposed to the balanced truncation method, computing a reduced basis in Krylov-subspace and POD methods is straightforward; the reduced basis is formed as the span of a set of state solutions, commonly referred to as snapshots. These snapshots are computed by solving the full system for selected values of the parameters and selected forcing inputs (possibly selected frequencies if a Krylov-subspace method is used). The quality of the resulting reduced-order model is very dependent on the choice of parameters and

inputs over which snapshots are computed. This is because the span of the snapshot determines the span of the reduced basis, which in turn determines the quality of the resulting reduced-order model. A key issue that needs to be addressed is therefore sampling; that is, how to choose the parameters and inputs over which to compute the basis.

1.2.2 Sampling for Model Reduction in Multi-Dimensional Parameter Spaces

In previous model reduction techniques via POD or Krylov-based methods, choosing where and how many samples to generate has been, in general, an ad-hoc process. Standard schemes such as uniform sampling (uniform gridding of the parameter space) or random sampling are not optimal. More importantly, if the dimension of the parameter space is large, uniform sampling will quickly become too computationally expensive due to the combinatorial explosion of samples needed to cover the parameter space. Random sampling, on the other hand, might fail to recognize where the important parameters are in the parameter space. One sampling strategy that compromises between the uniformity and the size of the sample is the stratified sampling family of which the popular Latin hypercube sampling (LHS) method is one example [43]. The LHS method is more efficient than uniform sampling and often more accurate than random sampling. Recently, the centroidal voronoi tessellation (CVT) sampling method [44–46] has arisen as a promising method. Initial evaluations [45–47] show that compared to LHS, the random sampling Monte Carlo methods, and Hammersley quasi Monte Carlo sequence methods, on balance the CVT sampling performs best at least for statistical sampling and function integration. Nonetheless, no attempt has compared these sampling methods in the context of efficient and accurate snapshot generation for model reduction of large-scale engineering problems.

One can use knowledge of the application at hand to determine representative inputs. In particular, empirical knowledge of the problem has been used to create the training parameter set for the quasi-convex optimization relaxation method [11],

and to sample a parameter space to generate a POD or Krylov basis for cases where the number of input parameters is small, for example optimal control applications [27, 28, 48, 49], aerodynamic applications [50, 51], physical processes [52], parametrized design of interconnect circuits [20–22], and in the case of multiple parameters describing inhomogeneous boundary conditions for parabolic PDEs [53]. However, this empirical approach breaks down for cases with many parameters. The recently developed iterative rational Krylov algorithm [54] proposes a systematic method for selecting interpolation points for multipoint rational Krylov approximations based on the rigorous $\mathcal{H}_2$-norm optimality criterion. This method has been applied to reduction of large-scale LTI systems, although its extension to parametrized LTI systems remains an open question.

Intuitively, one should not select parameter points where the error between the full and the reduced models is small since the reduced model is already a good approximation at these points. As a result, sampling the small error region increases the cost but no new information is added to the snapshot set. Instead, large error parameter points should be selected [55–57]. Recently, a greedy algorithm has been proposed to address the challenge of sampling a high-dimensional parameter space to build a reduced basis [55, 56, 58–62]. The greedy algorithm adaptively selects snapshots by finding the location in a training parameter set where an output error bound is maximal, updating the reduced basis with the solution at this sample location, forming a new reduced model, and repeating the process. The method has been successfully applied to many applications with small number of parameters such as fluid dynamics [56, 63], structure [56, 61], heat transfer [55, 61] and inverse problems [61, 62]; however, like the Krylov-subspace and POD methods, the quality of the training parameter set remains an open question.

1.2.3 Nonlinearity Treatments in Model Reduction

Recent efforts towards efficient nonlinear reduced models for the Euler and Navier-Stokes equations have been successful [12, 13, 64], by exploiting the special structure of the nonlinear terms of the governing equations so that reduced models are completely

independent of the full model size. For general nonlinear problems, this approach is no longer applicable. If the problems under consideration are weakly nonlinear, efficient reduced models could be obtained by retaining low order terms in the Taylor expansion of nonlinear terms [65]. Another approach is to use the trajectory piecewise-linear scheme [66] in which a weighted combination of various linear models is employed, hence a better approximation to the nonlinear behavior compared with using a single model; however, this method was found to be very sensitive and not robust to tuning parameters for CFD problems [67]. Refs. 61,62,68,69 propose an empirical interpolation method in which the nonlinear terms are approximated as a linear combination of empirical basis functions. Theoretical and numerical results show that this method is a promising approach for a general nonlinear problem.

1.2.4 Model Reduction Applications

Despite the fact that there are still open issues remaining to be addressed, model order reduction has been successfully applied to many areas of engineering. In particular, the reduced basis approach has been applied in the heat transfer context to design a thermal fin to minimize the material cost and the power required to effect the desired cooling [70]. In the context of active flow control, the POD method has been used to control the wake unsteadiness downstream of flow past a cylinder [13], and to control the driven velocity in a driven cavity flow and the vorticity in a channel flow over a backward-facing step [12]. In order to address the fact that the POD basis must be updated for the reduced model to be still a good representation of the full model when the control input changes, the trust-region POD (TRPOD) [71,72] and the POD for optimality system (OS-POD) [73] have been proposed and applied successfully for flow control problems. For applications in interconnect circuit analysis and MEMS, the quasi-convex optimization approach [11], the Krylov-based methods [20–22], the trajectory piecewise-linear scheme [66], and truncated balanced realization [74] have been applied to reduce full systems with thousands of states to reduced systems with a handful number of states, while accurately capturing the full system behavior. In the context of mistuning analysis of bladed-disks, both structural model reduction

[14–19] and aerodynamic model reduction [75] are able to accurately represent the large-scale full models with a handful number of reduced states. For aeroelasticity applications, both flutter and limit cycle oscillation predictions have been successful with the POD method in reducing more than three quarters of a million degrees of freedom to a few dozen degrees of freedom [76, 77]. Recently, the POD method has been applied to the CFD-based reduced-order aeroelastic modeling of a complete F-16 fighter configuration [78]. The results show that the reduced model aeroelastic predictions are in very good agreement with the full nonlinear model results and the experimental data.

1.3 Thesis Objectives

The objective of this thesis is to derive a general framework for efficient evaluation of the effects of parametric inputs (for example, shape variations, PDE coefficients, initial conditions, etc.) in the design and probabilistic analysis of large-scale systems, using model reduction methods.

In particular, this thesis aims to:

1. Develop a systematic technique for sampling parametric input spaces of high dimension in order to create a reduced basis.

2. Create reduced models that span the parametric input space for general large-scale systems—resulting from discretization of PDEs, for example—and quantify the ability of the reduced models to predict the outputs of interest over the parametric input space.

3. Demonstrate the proposed parametrized model reduction technique for two problems: optimal design of a steady heat conduction problem, and probabilistic analysis of the effects of blade shape variations on unsteady forced response of compressor blades.

1.4 Thesis Contributions

The main contributions of the thesis are to:

1. Develop a greedy-based adaptive model order reduction approach for general parameter-dependent problems such that:
    - The reduced basis takes into account the outputs of interest and the governing equations.
    - The reduced basis spans both parametric and temporal spaces.
    - The cost of constructing the reduced basis scales well to large dimensional parameter spaces.
2. Propose a reduction method for probabilistic analysis of the geometric mistuning problem in turbomachinery

1.5 Thesis Outline

In Chapter 2, methodologies to obtain reduced models for both steady and unsteady problems are discussed. In particular, components of a general projection-based model order reduction method will be developed. Components of the model order reduction method that could make the approximate solution unbounded are identified and investigated. Then, approaches to overcome the instability are discussed. In Chapter 3, a model-constrained greedy-based adaptive sampling method is proposed for reduction of large-scale problems that depend on a large number of parameters. First, the model-constrained adaptive sampling concepts, mathematical formulation and solution methodology of the greedy optimization problem—which is one of the key components of the model-constrained adaptive sampling approach—are presented. An analysis of the proposed adaptive sampling approach is then carried out. Chapter 4 applies the model-constrained adaptive sampling approach and the projection-based model reduction methods on the steady thermal fin optimal design problem.

A number of numerical simulations are presented to validate the theoretical developments in Chapter 3 and to compare the model-constrained sampling approach with other existing sampling methods. The application of the model-constrained adaptive model reduction method—which is the combination of the model-constrained adaptive sampling approach proposed in Chapter 3 and the projection-based model reduction techniques discussed in Chapter 2—in the context of optimal design is then discussed. To prepare for the numerical results in Chapter 6, Chapter 5 presents an unsteady linearized CFD model based on the Discontinuous Galerkin finite element method. A linearized CFD model for incorporating geometric variation effects into the unsteady simulation is then developed. Numerical results to validate the linearized CFD model are also discussed. Chapter 6 begins with a numerical result to demonstrate the stability and the convergence of the Petrov-Galerkin-projection-based model reduction discussed in Chapter 2. The application of the model-constrained adaptive model reduction approach to find reduced models for an unsteady problem in turbomachinery application is then presented. In particular, the application of the resulting reduced models in the context of probabilistic analysis with a large number of geometric variability parameters is investigated. Finally, Chapter 7 concludes the thesis with recommendations for extensions and future work.

Chapter 2

Projection-Based Model Order Reduction

In this chapter we discuss methodologies to obtain reduced models for both steady and unsteady problems. In particular, the chapter begins by developing components of a general projection-based model order reduction method. We first investigate why a projection-based reduced model could be unstable. That is, we identify components of the model order reduction method that could make the approximate solution unbounded. Then we discuss approaches to overcome the instability. One of the main ingredients is the minimum-residual (least-squares) projection that is widely used in linear algebra [79], in the finite element community [80–82], and recently in the reduced basis context [83–85].

2.1 General Projection-Based Model Order Reduction

Most large-scale model reduction frameworks are based on a projection approach, which can be described in general terms as follows. Given a large-scale dynamical problem

Find $\mathbf{x} \in I\!R^n$ *such that*

$$\mathbf{R}(\dot{\mathbf{x}}, \mathbf{x}, \mathbf{z}, \mathbf{u}, t) = \mathbf{0}, \quad \mathbf{x}(0) = \mathbf{x}^0, \quad \mathbf{y} = \mathcal{P}(\mathbf{x}, \mathbf{z}, \mathbf{u}, t), \tag{2.1}$$

where $\mathbf{x} = \mathbf{x}(\mathbf{z}, \mathbf{u}, t) \in \mathbb{R}^n$ is the full solution (state) vector, $\mathbf{z} \in \mathbb{R}^d$ is the vector containing the parameters of interest, t denotes time, $\mathbf{x}^0$ is the specified initial state, $\mathbf{u} = \mathbf{u}(t) \in \mathbb{R}^p$ is some time-dependent input vector, $\mathbf{R}$ is some discrete operator (residual operator resulting from a numerical discretization of a set of PDEs, for example), and $\mathbf{y} \in \mathbb{R}^q$ is a vector containing q outputs of interest computed by some output operator $\mathcal{P}$. The dynamical system (2.1) could be, for example, the finite element discretization of a set of PDEs, or governing equations in molecular dynamics simulation, or a dynamical system from circuit simulation, etc.

We will develop projection-based model order reduction techniques using linear algebra tools [79, 86, 87]. The functional analysis point of view, i.e. using variational tools, can be found in Refs. 56, 61, 62, 84. Since the size of the system, n, is typically very large, e.g. $n > 10^5$, in the context of design and optimization—in which the large-scale problem (2.1) needs to be solved repeatedly for many different design parameters $\mathbf{z}$—it is too computationally expensive to use the original full problem (2.1). Instead, we seek an approximate solution $\tilde{\mathbf{x}}$ within a reduced space $\mathcal{V}^m$. That is, $\tilde{\mathbf{x}} \in \mathcal{V}^m \subset \mathbb{R}^n$ where $\mathcal{V}^m$ is defined to be the span of some m independent vectors of $\mathbb{R}^n$, and is called the trial space. We also introduce the test space $\mathcal{W}^m \subset \mathbb{R}^n$. Let $\mathbf{\Phi}, \mathbf{\Psi} \in \mathbb{R}^{n \times m}$ be a basis of $\mathcal{V}^m$ and $\mathcal{W}^m$, respectively, and assume $\mathbf{\Psi}^T \mathbf{\Phi} = \mathbf{I}$, where $\mathbf{I}$ is the identity matrix. For example, $\mathbf{\Phi}$ contains as columns the basis vectors ϕ_i, i.e., $\mathbf{\Phi} = [\phi_1 \ \phi_2 \ \cdots \ \phi_m]$, and is assumed to span the full solution space over a range of interest of parameters $\mathbf{z}$ and unsteady inputs $\mathbf{u}$, i.e. $\mathbf{x}(\mathbf{z}, \mathbf{u}, t) \in \text{span}\{\mathbf{\Phi}\}, \forall \mathbf{z} \in \mathcal{S}^{\mathbf{z}}, \forall \mathbf{u} \in \mathcal{S}^{\mathbf{u}}$ where $\mathcal{S}^{\mathbf{z}}$ and $\mathcal{S}^{\mathbf{u}}$ are some subspaces of interest of the parametric and forcing spaces, respectively. The approximate solution $\tilde{\mathbf{x}}$ can be then expressed as

$$\tilde{\mathbf{x}} = \mathbf{\Phi} \mathbf{x}_r, \tag{2.2}$$

and the reduced model can be written in matrix form

Find $\mathbf{x}_r \in \mathbb{R}^m$ *such that*

$$\boldsymbol{\Psi}^T \mathbf{R}(\boldsymbol{\Phi}\dot{\mathbf{x}}_r, \boldsymbol{\Phi}\mathbf{x}_r, \mathbf{z}, \mathbf{u}, t) = 0, \quad \mathbf{x}_r(0) = \boldsymbol{\Psi}^T \mathbf{x}^0, \quad \mathbf{y}_r = \mathcal{P}(\boldsymbol{\Phi}\mathbf{x}_r, \mathbf{z}, \mathbf{u}, t), \tag{2.3}$$

where $\mathbf{x}_r = \mathbf{x}_r(\mathbf{z}, \mathbf{u}, t) \in \mathbb{R}^m$ is the vector of the coordinates of the approximate solution $\tilde{\mathbf{x}}$ in the reduced basis $\boldsymbol{\Phi}$, and $\mathbf{y}_r \in \mathbb{R}^q$ is the approximate output. We have used the reduced transform (2.2) and the assumption $\boldsymbol{\Psi}^T\boldsymbol{\Phi} = \mathbf{I}$ to obtain the initial condition for the reduced state $\mathbf{x}_r(0)$. If the test space is the same as the trial space, i.e. $\boldsymbol{\Psi} = \boldsymbol{\Phi}$, the reduced system (2.3) is obtained via a Galerkin projection. If the test space is different from the trial space, the reduced system (2.3) is obtained via a Petrov-Galerkin projection.

Figure 2-1 summarizes the above projection-based reduction method. One starts with a dynamical system in the full space. The first step is to identify the reduced basis pair $\boldsymbol{\Phi}, \boldsymbol{\Psi}$, and hence the reduced spaces $\mathcal{V}^m$ and $\mathcal{W}^m$. The second step is to employ the reduced transform (2.2) and to perform the projection to obtain the reduced dynamical system. Once the reduced state $\mathbf{x}_r$ is found, the approximate solution is then reconstructed using the reduced transform (2.2).

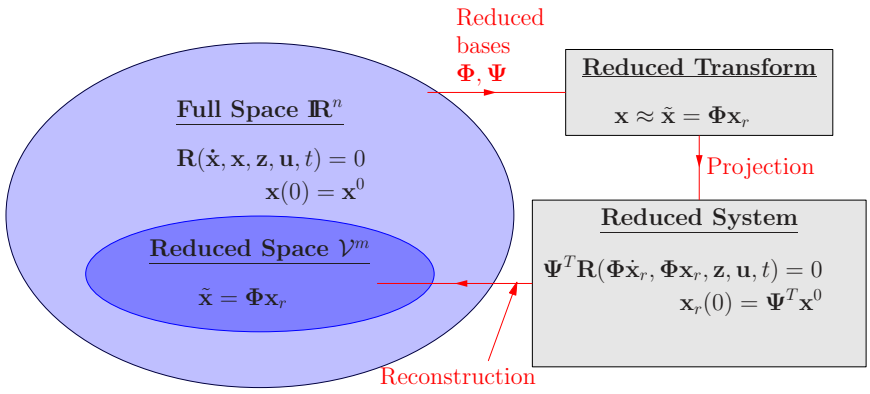


Figure 2-1: A general projection-based model order reduction.

One of the important tasks of a projection-based model reduction technique is therefore to find a reduced basis pair $\boldsymbol{\Psi}$ and $\boldsymbol{\Phi}$ so that the reduced system (2.3) provides an accurate representation of the large-scale system (2.1) over the desired range of inputs and (possibly) parameters.

Next let us apply the general projection framework to find a reduced model of the general parametrized LTI dynamical system

$$\mathbf{E}(\mathbf{z})\dot{\mathbf{x}} = \mathbf{A}(\mathbf{z})\mathbf{x} + \mathbf{B}(\mathbf{z})\mathbf{u}, \qquad \mathbf{y} = \mathbf{C}(\mathbf{z})\mathbf{x}, \tag{2.4}$$

with initial condition

$$\mathbf{x}(0) = \mathbf{x}^0, \tag{2.5}$$

where the matrices $\mathbf{E}(\mathbf{z}), \mathbf{A}(\mathbf{z}) \in \mathbb{R}^{n\times n}$, $\mathbf{B}(\mathbf{z}) \in \mathbb{R}^{n\times p}$, and $\mathbf{C}(\mathbf{z}) \in \mathbb{R}^{q\times n}$ in (2.4) may depend (possibly nonlinearly) on a set of parameters $\mathbf{z}$; the parameters $\mathbf{z}$ could be coefficients of the PDEs, for example heat conductivities, or shape parameters. The input vector $\mathbf{u}(t) \in \mathbb{R}^p$ could describe prescribed unsteady boundary motion. Systems of the form (2.4) that result from spatial discretization of a set of PDEs can be found in Chapter 5. In this case, the dimension of the system, n, is very large and the matrices $\mathbf{E}(\mathbf{z}), \mathbf{A}(\mathbf{z}), \mathbf{B}(\mathbf{z})$ and $\mathbf{C}(\mathbf{z})$ result from the chosen spatial discretization method.

Now denoting the residual as

$$\mathbf{R}(\mathbf{\Phi}\dot{\mathbf{x}}_r, \mathbf{\Phi}\mathbf{x}_r, \mathbf{z}, \mathbf{u}, t) = \mathbf{E}(\mathbf{z})\mathbf{\Phi}\dot{\mathbf{x}}_r - \mathbf{A}(\mathbf{z})\mathbf{\Phi}\mathbf{x}_r - \mathbf{B}(\mathbf{z})\mathbf{u}, \tag{2.6}$$

we apply the above general projection-based model order reduction technique. This yields the parametrized LTI reduced-order model with state $\mathbf{x}_r(t)$ and output $\mathbf{y}_r(t)$

$$\begin{aligned} \mathbf{E}_r(\mathbf{z})\dot{\mathbf{x}}_r &= \mathbf{A}_r(\mathbf{z})\mathbf{x}_r + \mathbf{B}_r(\mathbf{z})\mathbf{u}, & (2.7)\\ \mathbf{y}_r &= \mathbf{C}_r(\mathbf{z})\mathbf{x}_r, & (2.8)\\ \mathbf{x}_r(0) &= \mathbf{\Psi}^T\mathbf{x}^0, & (2.9) \end{aligned}$$

where $\mathbf{E}_r(\mathbf{z}) = \mathbf{\Psi}^T\mathbf{E}(\mathbf{z})\mathbf{\Phi}$, $\mathbf{A}_r(\mathbf{z}) = \mathbf{\Psi}^T\mathbf{A}(\mathbf{z})\mathbf{\Phi}$, $\mathbf{B}_r(\mathbf{z}) = \mathbf{\Psi}^T\mathbf{B}(\mathbf{z})$, $\mathbf{C}_r(\mathbf{z}) = \mathbf{C}(\mathbf{z})\mathbf{\Phi}$. Note that we have performed the projection without discretizing the time derivative terms, but one could also discretize the time derivative terms first, then perform the projection on the fully discrete system.

If, on the other hand, one starts with a full steady system

$$\mathbf{A}(\mathbf{z})\mathbf{x} = \mathbf{B}(\mathbf{z}), \quad \mathbf{y} = \mathbf{C}(\mathbf{z})\mathbf{x}, \tag{2.10}$$

and defines the residual as

$$\mathbf{R}(\mathbf{\Phi}\mathbf{x}_r, \mathbf{z}) = \mathbf{B}(\mathbf{z}) - \mathbf{A}(\mathbf{z})\mathbf{\Phi}\mathbf{x}_r, \tag{2.11}$$

the above projection-based model order reduction technique will yield the reduced system of the form

$$\mathbf{A}_r(\mathbf{z})\mathbf{x}_r = \mathbf{B}_r(\mathbf{z}), \quad \mathbf{y}_r = \mathbf{C}_r(\mathbf{z})\mathbf{x}_r, \tag{2.12}$$

where again $\mathbf{A}_r(\mathbf{z}) = \mathbf{\Psi}^T\mathbf{A}(\mathbf{z})\mathbf{\Phi}$, $\mathbf{B}_r(\mathbf{z}) = \mathbf{\Psi}^T\mathbf{B}(\mathbf{z})$, $\mathbf{C}_r(\mathbf{z}) = \mathbf{C}(\mathbf{z})\mathbf{\Phi}$. Systems of the form (2.10) that result from spatial discretization of a set of PDEs can be found in Chapter 4.

In the following we discuss methodologies to construct the reduced basis $\mathbf{\Psi}$ to obtain a guaranteed stable reduced model for both steady and unsteady problems that are linear in state vector as in (2.4)–(2.5) and (2.10). Extensions to problems that are nonlinear in state vector are not addressed in this thesis. Construction of the basis $\mathbf{\Phi}$ will be discussed in Chapter 3.

2.2 Construction of the Reduced Test Basis $\mathbf{\Psi}$ for Steady Problems

For steady problems, we first denote $\mathbf{e} = \mathbf{x} - \tilde{\mathbf{x}}$, the state error between the full solution and the approximation. The $\mathbf{A}^T\mathbf{A}-$norm is defined as $\|\mathbf{v}\|_{\mathbf{A}^T\mathbf{A}} = \mathbf{v}^T\mathbf{A}^T\mathbf{A}\mathbf{v}, \forall \mathbf{v} \in \mathbb{R}^n$. From the residual-error relation $\mathbf{R}(\mathbf{\Phi}\mathbf{x}_r, \mathbf{z}) = \mathbf{A}\mathbf{e}$, where $\mathbf{R}$ is defined in (2.11), it is easy to see that the following identity is true

$$\|\mathbf{R}\|_2 = \|\mathbf{e}\|_{\mathbf{A}^T\mathbf{A}}. \tag{2.13}$$

Now if the Galerkin projection, i.e. $\boldsymbol{\Psi} = \boldsymbol{\Phi}$, is employed to find the reduced state $\mathbf{x}_r$, it provides no guarantee on the stability of the reduced model if $\mathbf{A}$ is not symmetric positive definite. (If $\mathbf{A}$ is, however, symmetric positive definite one could prove that the Galerkin projection yields an optimal reduced model in the $\mathbf{A}-$norm, that is, the state error is minimized in the $\mathbf{A}-$norm.) This is because the reduced equation (2.12) is obtained by enforcing the residual to be orthogonal to the reduced space, and hence the residual could be mathematically arbitrarily large while being orthogonal to the reduced space.

The above discussion suggests that the approximate solution should minimize the residual in (2.11). In other words, we find the reduced state $\mathbf{x}_r$ to minimize the residual. This is a form of the minimum-residual statement in linear algebra and the finite element contexts [79–81]. Recently, the minimum-residual approach has been successfully used in the reduced-basis context [84, 85, 88]. Mathematically, the minimum-residual statement can be expressed in terms of the following least square minimization

$$\mathbf{x}_r = \arg \min_{\bar{\mathbf{x}}_r \in \mathbb{R}^m} \|\mathbf{R}(\boldsymbol{\Phi}\mathbf{x}_r, \mathbf{z})\|_2^2 = \|\mathbf{B} - \mathbf{A}\boldsymbol{\Phi}\bar{\mathbf{x}}_r\|_2^2 \tag{2.14}$$

whose optimality condition, which is the reduced model, is given by

$$(\mathbf{A}\boldsymbol{\Phi})^T(\mathbf{A}\boldsymbol{\Phi})\mathbf{x}_r = (\mathbf{A}\boldsymbol{\Phi})^T\mathbf{B}. \tag{2.15}$$

Next, if we choose the reduced test basis $\boldsymbol{\Psi}$ to be

$$\boldsymbol{\Psi} = \mathbf{A}\boldsymbol{\Phi}, \tag{2.16}$$

then the minimum-residual approach is equivalent to a Petrov-Galerkin projection with the test space given in (2.16). This particular Petrov-Galerkin projection is equivalent to Galerkin projection on the normal equation. Moreover, the fact in (2.13) that the residual *2-norm* is exactly the state error in the $\mathbf{A}^T\mathbf{A}-$norm makes this particular Petrov-Galerkin projection the best reduction approach in the sense that the resulting reduced model minimizes the state error in the $\mathbf{A}^T\mathbf{A}-$norm. It

should be pointed out that the minimum-residual approach yields the test reduced basis and the reduced model at the same time. We are now in position to prove the reduced model (2.15) obtained from the minimum-residual statement, or equivalently from the Petrov-Galerkin projection (2.16), is guaranteed to be stable.

Theorem 2.1 *Assume* $\mathbf{A}$ *has a bounded condition number, then the reduced model* (2.15) *is stable in the sense that the state error is bounded. In particular, the following bound holds*

$$\|\mathbf{e}\|_2 \leq \frac{1}{\sigma_{min}^{\mathbf{A}}} \|\mathbf{R}(\mathbf{\Phi}\mathbf{x}_r, \mathbf{z})\|_2 \tag{2.17}$$

where $\sigma_{min}^{\mathbf{A}}$ *is the smallest singular value of* $\mathbf{A}$*, the reduced state* $\mathbf{x}_r$ *is computed from equation* (2.15)*, and the residual* $\mathbf{R}$ *is defined in* (2.11)*.*

Proof: Making use of the inequality for compatible matrix and vector norms on the error-residual relation $\mathbf{e} = \mathbf{A}^{-1}\mathbf{R}$ and using the definition of the singular values of a matrix yield the bound (2.17). Since the reduced state found from equation (2.15) minimizes the residual, the residual is finite (because $\|\mathbf{R}(\mathbf{x}_r, \mathbf{z})\|_2 = \min_{\bar{\mathbf{x}}_r \in \mathbb{R}^m} \|\mathbf{R}(\mathbf{\Phi}\bar{\mathbf{x}}_r, \mathbf{z})\|_2 \leq \|\mathbf{R}(\mathbf{\Phi}\mathbf{0}, \mathbf{z})\|_2 = \|\mathbf{B}(\mathbf{z})\|_2$). In addition, $1/\sigma_{min}^{\mathbf{A}}$ is finite due to the bounded conditioned number assumption on $\mathbf{A}$. We therefore conclude that the state error in the *2-norm* is bounded. □

A priori convergence results are standard in the context of the finite element method [2,89], and recently in the reduced basis approach context [56,58–62]. In this light, we present an *a priori* convergence result for the above steady reduced model.

Theorem 2.2 *As the reduced basis* $\mathbf{\Phi}$ *is enriched, i.e. more basis vectors are added, the approximate solution, hence the reduced model, is improved in the sense that the state error in the* $\mathbf{A}^T\mathbf{A}-$*norm is a non-increasing function of the number of reduced basis vectors. In particular, there exists* $m \leq n$ *at which the state error in the* $\mathbf{A}^T\mathbf{A}-$*norm is strictly monotone decreasing (a linear algebra version of the a priori convergence result in [56, 58–62]).*

Proof: Assume $\mathbf{\Phi} \in \mathbb{R}^{n \times m}$ is the current reduced basis. Now adding a new basis

vector ϕ_{m+1}, the new reduced basis $\tilde{\mathbf{\Phi}}$ is given as

$$\tilde{\mathbf{\Phi}} = [\mathbf{\Phi}, \phi_{m+1}] \in \mathbb{R}^{n\times(m+1)}. \tag{2.18}$$

The minimum-residual statement of the new reduced model is given as

$$\mathbf{x}_r^* = \arg \min_{\tilde{\bar{\mathbf{x}}}_r \in \mathbb{R}^{m+1}} \|\mathbf{B} - \mathbf{A}[\mathbf{\Phi}, \phi_{m+1}]\tilde{\bar{\mathbf{x}}}_r\|_2^2. \tag{2.19}$$

Comparing the minimization statements (2.14) and (2.19), and using the fact that $span\{\mathbf{\Phi}\} \subset span\{\tilde{\mathbf{\Phi}}\}$ and $\mathbb{R}^m \subset \mathbb{R}^{m+1}$, it can be seen that $\bar{\mathbf{x}}_r$ is a special case of $\tilde{\bar{\mathbf{x}}}_r$ whose last element is zero. That is, in (2.19), the residual norm is minimized in a larger space, and hence the residual should be no larger than that of (2.14). Equivalently, the state error in the $\mathbf{A}^T\mathbf{A}-$norm is no larger when the reduced space is richer. Since the approximate solution is exact if $m = n$, there exists $m \leq n$ such that when more basis vectors are added the state error is smaller. □

Note that from the residual-error relation $\mathbf{R} = \mathbf{A}\mathbf{e}$ and the output error relation $\mathbf{y} - \mathbf{y}_r = \mathbf{C}\mathbf{e}$, we conclude that as the reduced basis is enriched, the residual and the output error are also improved. This *a priori* convergence result is important for the theoretical development of our adaptive sampling method discussed later in Chapter 3.

2.3 Construction of the Reduced Test Basis $\mathbf{\Psi}$ for Unsteady Problems

For the unsteady problem, we first review mathematically why a reduced model could be unstable even if the full model is stable. The details of the origin of the instability can be found in [86, 90] and references therein. To begin, assume $\mathbf{E} = \mathbf{I}$ and denote the *numerical range* of the matrix $\mathbf{A}$ as

$$\Gamma(\mathbf{A}) = \{\mathbf{x}^T\mathbf{A}\mathbf{x} : \|\mathbf{x}\|_2 = 1\}. \tag{2.20}$$

If $\mathbf{A}$ is normal, i.e. $\mathbf{A}\mathbf{A}^T = \mathbf{A}^T\mathbf{A}$, the numerical range is equal to the convex hull of the spectrum of $\mathbf{A}$. As a consequence, the reduced matrix $\mathbf{\Phi}^T\mathbf{A}\mathbf{\Phi}$ can never be unstable, e.g. the reduced solution $\tilde{\mathbf{x}}_r$ does not increase exponentially in time, if $\mathbf{A}$ is stable. However, if $\mathbf{A}$ is not normal, the numerical range can extend into the right half plane (e.g. the reduced matrix has some eigenvalues with positive real parts) depending on the reduced basis $\mathbf{\Phi}$, and hence the reduced matrix could be unstable. Unfortunately, in most CFD applications, the matrix $\mathbf{A}$ is not normal and that is the reason why the Galerkin projection method can yield an unstable reduced model.

It is well known that there two approaches in optimization. That is, one can use either the differentiate-then-discretize or discretize-then-differentiate approaches. Each of these methods has its own advantages and disadvantages; a detailed discussion of these two approaches in the optimal control context can be found in [91]. In the following, we discuss the application of the discretize-then-differentiate approach together with the minimum-residual statement in Section 2.2 to find a test reduced basis $\mathbf{\Psi}$ and at the same time a reduced model for time-dependent problems.

In the discretize-then-differentiate approach, one first discretizes the time derivative of the full model. A suitable optimization problem is then constructed and differentiated to find the optimality condition. The optimality condition is solved to find an approximate optimizer.

Now, without loss of generality, assume that the Backward-Euler method is used to discretize the time-dependent terms of equation (2.4), the full model is now given as

$$(\mathbf{E} - \Delta t\mathbf{A})\mathbf{x}^k = \mathbf{E}\mathbf{x}^{k-1} + \Delta t\mathbf{B}\mathbf{u}^k, \quad k = 1, \ldots, n_t, \tag{2.21}$$

where k denotes the time level, Δt is the time step, and n_t is the number of time steps. Next, denote the residual at time step k as

$$\mathbf{R}^k = (\mathbf{E} - \Delta t\mathbf{A})\tilde{\mathbf{x}}^k - \mathbf{E}\tilde{\mathbf{x}}^{k-1} - \Delta t\mathbf{B}\mathbf{u}^k, \tag{2.22}$$

where $\tilde{\mathbf{x}}^k = \mathbf{\Phi}\mathbf{x}_r^k$ is the approximation at the k^{th} time step, and $\mathbf{x}_r^k$ is the corresponding reduced state vector.

Similar to the minimum-residual statement in Section 2.2, all the residuals are minimized simultaneously as

$$\min_{\mathbf{x}_r^1,\ldots,\mathbf{x}_r^{n+1}\in\mathbb{R}^m} \sum_{k=1}^{n_t} \|\mathbf{R}^k\|_2^2 = \sum_{k=1}^{n_t} \|(\mathbf{E}-\Delta t\mathbf{A})\mathbf{\Phi}\mathbf{x}_r^k - \mathbf{E}\mathbf{\Phi}\mathbf{x}_r^{k-1} - \Delta t\mathbf{B}u^k\|_2^2. \tag{2.23}$$

This is equivalent to stacking up the residuals at all time steps into a single long residual vector. Symbolically, the residual equations can be written as

$$\underbrace{\begin{bmatrix}\mathbf{R}^1\\ \mathbf{R}^2\\ \vdots\\ \mathbf{R}^{n_t}\end{bmatrix}}_{\bar{\mathbf{R}}} = \underbrace{\begin{bmatrix}-\Delta t\mathbf{B}\mathbf{u}^1 - \mathbf{E}\mathbf{\Phi}\mathbf{x}_r^0\\ -\Delta t\mathbf{B}\mathbf{u}^2\\ \vdots\\ -\Delta t\mathbf{B}\mathbf{u}^{n_t}\end{bmatrix}}_{\bar{\mathbf{B}}} - \underbrace{\begin{bmatrix}\mathbf{Q} & & & \\ \mathbf{E} & \mathbf{Q} & & \\ & \ddots & \ddots & \\ & & \mathbf{E} & \mathbf{Q}\end{bmatrix}}_{\bar{\mathbf{A}}} \underbrace{\begin{bmatrix}\mathbf{\Phi} & & & \\ & \mathbf{\Phi} & & \\ & & \ddots & \\ & & & \mathbf{\Phi}\end{bmatrix}}_{\bar{\mathbf{\Phi}}} \underbrace{\begin{bmatrix}\mathbf{x}_r^1\\ \mathbf{x}_r^2\\ \vdots\\ \mathbf{x}_r^{n_t}\end{bmatrix}}_{\bar{\mathbf{x}}_\mathbf{r}} \tag{2.24}$$

where $\mathbf{Q} = -(\mathbf{E}-\Delta t\mathbf{A})$, and blank entries in matrices $\bar{\mathbf{A}}$ and $\bar{\mathbf{\Phi}}$ mean zero blocks. The minimum-residual problem (2.23) is equivalent to the following minimum-residual problem

$$\min_{\bar{\mathbf{x}}_r} \|\bar{\mathbf{R}}\|_2^2 = \|\bar{\mathbf{B}} - \bar{\mathbf{A}}\bar{\mathbf{\Phi}}\bar{\mathbf{x}}_r\|_2^2, \tag{2.25}$$

where $\bar{\mathbf{R}}, \bar{\mathbf{B}}, \bar{\mathbf{A}}$ and $\bar{\mathbf{\Phi}}$ are defined in (2.24). Next, setting the first derivative of the residual norm $\|\bar{\mathbf{R}}\|_2^2$ with respect to $\bar{\mathbf{x}}_r^k$ to zero, we obtain the reduced equations

$$\begin{aligned}\left[(\mathbf{E}-\Delta t\mathbf{A})\mathbf{\Phi}\right]^T\mathbf{R}^k - (\mathbf{E}\mathbf{\Phi})^T\mathbf{R}^{k+1} &= 0, \quad k=1,\ldots,n_t-1, &(2.26)\\ \left[(\mathbf{E}-\Delta t\mathbf{A})\mathbf{\Phi}\right]^T\mathbf{R}^{n_t} &= 0, &(2.27)\end{aligned}$$

which can be viewed as a weighted Petrov-Galerkin projection in which two different test reduced spaces $(\mathbf{E}-\Delta t\mathbf{A})\mathbf{\Phi}$ and $\mathbf{E}\mathbf{\Phi}$ are used for the residual at two successive time steps. In matrix form, the reduced equations have the following symmetric block

tri-diagonal form

$$\begin{bmatrix} \mathbf{H} & \mathbf{D}^T & 0 & \dots & \dots & \dots & \dots \\ \mathbf{D} & \mathbf{H} & \mathbf{D}^T & 0 & \dots & \dots & \dots \\ 0 & \ddots & \ddots & \ddots & \ddots & \dots & \dots \\ \dots & 0 & \mathbf{D} & \mathbf{H} & \mathbf{D}^T & 0 & \dots \\ \vdots & \vdots & \ddots & \ddots & \ddots & \ddots & \ddots \\ \dots & \dots & \dots & 0 & \mathbf{D} & \mathbf{H} & \mathbf{D}^T \\ \dots & \dots & \dots & \dots & 0 & \mathbf{D} & \bar{\mathbf{H}} \end{bmatrix} \begin{bmatrix} \mathbf{x}_r^1 \\ \mathbf{x}_r^2 \\ \vdots \\ \mathbf{x}_r^k \\ \vdots \\ \mathbf{x}_r^{n_t-1} \\ \mathbf{x}_r^{n_t} \end{bmatrix} = \begin{bmatrix} \mathbf{F}^1 \\ \mathbf{F}^2 \\ \vdots \\ \mathbf{F}^k \\ \vdots \\ \mathbf{F}^{n_t-1} \\ \mathbf{F}^{n_t} \end{bmatrix} \tag{2.28}$$

where

$$\mathbf{H} = [(\mathbf{E} - \Delta t\mathbf{A})\mathbf{\Phi}]^T [(\mathbf{E} - \Delta t\mathbf{A})\mathbf{\Phi}] + \mathbf{\Phi}^T\mathbf{E}^T\mathbf{E}\mathbf{\Phi}, \tag{2.29}$$

$$\bar{\mathbf{H}} = [(\mathbf{E} - \Delta t\mathbf{A})\mathbf{\Phi}]^T [(\mathbf{E} - \Delta t\mathbf{A})\mathbf{\Phi}], \tag{2.30}$$

$$\mathbf{D} = -[(\mathbf{E} - \Delta t\mathbf{A})\mathbf{\Phi}]^T (\mathbf{E}\mathbf{\Phi}), \tag{2.31}$$

$$\mathbf{F}^1 = [(\mathbf{E} - \Delta t\mathbf{A})\mathbf{\Phi}]^T (\mathbf{E}\mathbf{\Phi}\mathbf{x}_r^0 + \Delta t\mathbf{B}\mathbf{u}^1) - (\mathbf{E}\mathbf{\Phi})^T(\Delta t\mathbf{B}\mathbf{u}^2), \tag{2.32}$$

$$\mathbf{F}^k = [(\mathbf{E} - \Delta t\mathbf{A})\mathbf{\Phi}]^T (\Delta t\mathbf{B}\mathbf{u}^k) - (\mathbf{E}\mathbf{\Phi})^T(\Delta t\mathbf{B}\mathbf{u}^{k+1}), k = 2, ..., n_t - 1, \tag{2.33}$$

$$\mathbf{F}^{n_t} = [(\mathbf{E} - \Delta t\mathbf{A})\mathbf{\Phi}]^T (\Delta t\mathbf{B}\mathbf{u}^{n_t}). \tag{2.34}$$

It can be seen that the residual equation (2.24) and the minimum-residual statement (2.25) have exactly the same form as the steady residual equation (2.11) and the minimum-residual statement (2.14), respectively. As a result, all the results in Section 2.2 hold for reduced models of unsteady problems using the method in this section as well. In particular, the stability of the form (2.17) and *a priori* convergence result as in Theorem 2.2 hold.

However, even though the system (2.28) is sparse, its size, $m \times n_t$, can be large, because even if the number of reduced states m is moderate, the number of time steps n_t could be very large. In that case, an efficient algorithm must be derived to solve this symmetric block tri-diagonal system to find the reduced state vectors. Therefore, while this method is of theoretical interest (because it provides an *a priori*

convergence result which is useful for the theoretical developments of our adaptive sampling approach in Chapter 3), in practice we minimize the residuals sequentially. That is, the residual at the first time step is minimized to determine the reduced state at the first time step. Once the reduced state at the first time step is computed, the residual at the second time step is minimized to determine the reduced state at the second time step. This process is repeated until the reduced state at the final time step is computed. In particular, minimizing the residual at the k^{th} time step yields the corresponding reduced equation at that time step as

$$[(\mathbf{E} - \Delta t\mathbf{A})\mathbf{\Phi}]^T \mathbf{R}^k = 0 \tag{2.35}$$

which can be seen to be obtained via a Petrov-Galerkin projection with the test reduced basis as $\mathbf{\Psi} = (\mathbf{E} - \Delta t\mathbf{A})\mathbf{\Phi}$. The main advantage of this approach is that the reduced equations at each time step are decoupled, and hence are easy to solve. We do not yet have *a priori* convergence result as in Theorem 2.2, though numerical results, as we will present in Section 6.1, justify this convergence.

Chapter 3

Model-Constrained Greedy-Based Adaptive Sampling Approaches for Parameter-Dependent Problems

We have discussed methodologies to compute a good test reduced basis in Chapter 2. In this chapter we propose approaches to find a good trial reduced basis. In particular, we develop a model-constrained greedy-based adaptive sampling method for model reduction of large-scale problems that depend on a large number of parameters. First, the model-constrained adaptive sampling concepts, and corresponding mathematical formulations are presented. A solution methodology for the greedy optimization problem, which is one of the key components of the model-constrained adaptive sampling approach, is also discussed. An analysis of the proposed adaptive sampling approach is then carried out. Finally, the greedy optimization problem and its first order optimality conditions are derived for steady and unsteady problems that are linear in the state vector.

3.1 Model-Constrained Greedy-Based Sampling Approach

Recall that our model reduction task is one of determining an appropriate reduced basis that spans both the parametric input space $\mathbf{z}$ and the space of unsteady inputs $\mathbf{u}(t)$. In the case of the dynamical system (2.1) with no dependence on parameters $\mathbf{z}$, a number of model reduction techniques can be used, such as Krylov-based methods and POD. In these methods, the reduced basis is formed as the span of a set of state solutions, commonly referred to as snapshots. These snapshots are computed by solving the full system for selected values of the parameters and selected forcing inputs (possibly selected frequencies if a Krylov-subspace method is used). In order to extend these techniques to the general case where the system matrices depend on the parameters $\mathbf{z}$, we require a systematic method of sampling the parametric input space, and the forcing input space as well.

3.1.1 Greedy Adaptive Sampling Concept

A recently proposed approach to address the challenge of sampling a high-dimensional parameter space to build a reduced basis is the greedy algorithm [56, 58, 62, 63]. The greedy algorithm adaptively selects snapshots by finding the location in a predetermined discrete parameter set (training parameter set) where an output error bound, i.e. an upper bound of the error between the full and the reduced outputs, is maximal, updating the basis with information gathered from this sample location, forming a new reduced model, and repeating the process.

Here, we formulate the greedy approach as an optimization problem that targets an error estimation (which could be an output error indicator, an output error bound, or the true output error) of reduced model output prediction. The optimization problem is defined by introducing as constraints the systems of equations representing the reduced model (and possibly the full model if the true output error is targeted). The optimization formulation treats the parameter space as continuous; that is, we do not

require *a priori* selection of a discrete training parameter set. As a result, instead of finding globally-optimal parameter points in the training set, our method seeks to find locally-optimal parameter points in the continuous parameter space. Further, since any error estimation can be used as our selection criteria, our approach is applicable in cases for which output error bounds are unavailable. We use state-of-the-art optimization techniques to solve the resulting greedy PDE-constrained optimization problem.

3.1.2 Greedy Optimization Problem

In each cycle of the greedy algorithm, the key step is to determine the location in parameter space where the error in the reduced model is maximal. For the sake of clarity, we first discuss sampling methods in the parameter space, and then address sampling approaches for the unsteady forcing input space. We define the cost functional as a function of the output error norm

$$\mathcal{G}(\mathbf{x}, \mathbf{x}_r, \mathbf{z}) = \frac{1}{2}\|\mathbf{y} - \mathbf{y}_r\|_O^2, \tag{3.1}$$

where the appropriate definition of the output error norm $\|\cdot\|_O$ for steady and unsteady problems will be discussed later. Given a current basis $\mathbf{\Phi}$, we find the location in parameter space of maximum output error by solving the optimization problem

$$\max_{\mathbf{x},\mathbf{x}_r,\mathbf{z}} \mathcal{G} = \frac{1}{2}\|\mathbf{y} - \mathbf{y}_r\|_O^2 \tag{3.2}$$

subject to

$$\begin{aligned}
\mathbf{R}(\dot{\mathbf{x}}, \mathbf{x}, \mathbf{z}, \mathbf{u}(t), t) &= 0, && (3.3)\\
\mathbf{x}(0) &= \mathbf{x}^0, && (3.4)\\
\mathbf{y} &= \mathcal{P}(\mathbf{x}, \mathbf{z}, \mathbf{u}(t), t), && (3.5)\\
\mathbf{\Psi}^T\mathbf{R}(\mathbf{\Phi}\dot{\mathbf{x}}_r, \mathbf{\Phi}\mathbf{x}_r, \mathbf{z}, \mathbf{u}(t), t) &= 0, && (3.6)\\
\mathbf{x}_r(0) &= \mathbf{\Psi}^T\mathbf{x}^0, && (3.7)\\
\mathbf{y}_r &= \mathcal{P}(\mathbf{\Phi}\mathbf{x}_r, \mathbf{z}, \mathbf{u}(t), t), && (3.8)\\
\mathbf{z}_{min} \leq \quad \mathbf{z} \quad &\leq \mathbf{z}_{max}, && (3.9)
\end{aligned}$$

where $\mathbf{z}_{min}$ and $\mathbf{z}_{max}$ are respectively lower and upper bounds on the parameter vector $\mathbf{z}$. Since the true output error—the error between the full and the reduced outputs—has been used as the cost functional, both the full model (3.3)–(3.5) and the reduced model (3.6)–(3.8) are constraints of the optimization problem.

To solve the optimization problem (3.2)–(3.9), each optimization iteration may be expensive, because the constraints include the full model. If an output error bound [60–62, 84] exists, it could be used as the cost functional instead of the true output error. In that case, the constraints only comprise the reduced model and the bound constraints. As a result, solving the optimization problem in this case is much less expensive since it involves no full system solves. However, for a general problem, an error bound may not be available. Alternatively, an error indicator, for example the square of the *2-norm* of the residual, $\|\mathbf{R}(\mathbf{\Phi}\dot{\mathbf{x}}_r, \mathbf{\Phi}\mathbf{x}_r, \mathbf{z}, \mathbf{u}, t)\|_2^2$, could be employed (note that for problems that result from spatial discretization of a set of PDEs one can use the dual norm as in Section 4.3.1 or any weighted-residual forms). In such cases, denote the output error bound or the norm of the residual as $\mathcal{Q}(\mathbf{x}_r, \mathbf{z}, \mathbf{u}, t)$; the optimization problem now reads

$$\max_{\mathbf{x}, \mathbf{x}_r, \mathbf{z}} \mathcal{G} = \mathcal{Q}(\mathbf{x}_r, \mathbf{z}, \mathbf{u}, t) \qquad (3.10)$$

subject to

$$\begin{aligned} \boldsymbol{\Psi}^T \mathbf{R}(\boldsymbol{\Phi}\dot{\mathbf{x}}_r, \boldsymbol{\Phi}\mathbf{x}_r, \mathbf{z}, \mathbf{u}(t), t) &= 0, && (3.11) \\ \mathbf{x}_r(0) &= \boldsymbol{\Psi}^T \mathbf{x}^0, && (3.12) \\ \mathbf{z}_{min} \leq \quad \mathbf{z} \quad &\leq \mathbf{z}_{max}. && (3.13) \end{aligned}$$

From now on to the end of this chapter, theoretical results will be developed based on the optimization problem using the true output error, (3.2)–(3.9). However, by removing the full model constraint and using $\mathcal{Q}(\mathbf{x}_r, \mathbf{z}, \mathbf{u}, t)$ instead of $\frac{1}{2}\|\mathbf{y} - \mathbf{y}_r\|_O^2$ in the cost functional, the results also hold for the optimization problem (3.10)–(3.13).

We denote the parameter vector that solves the maximization problem (3.2)–(3.9) by $\mathbf{z}^*$. Next, we compute the solution $\mathbf{x}(\mathbf{z}^*, t)$ of the full system at the worst-case parameter value $\mathbf{z}^*$. This solution information is added to the basis $\boldsymbol{\Phi}$, for example using the POD (note that once the sample location has been found, other model reduction methods could also be employed). The procedure is then repeated by solving the optimization problem (3.2)–(3.9) with the updated basis $\boldsymbol{\Phi}$. Thus, we are using a systematic, adaptive error metric based on the ability of the reduced-order model to capture the outputs of interest in order to choose the snapshot locations that are locally the worst case scenarios. This adaptive sampling approach is summarized in the following algorithm.

Algorithm 3.1 *Model-Constrained Adaptive Sampling Procedure*

1. *Given a reduced basis* $\boldsymbol{\Phi}$ *and initial guess* $\mathbf{z}^0$*, solve the optimization problem* (3.2)–(3.9) *to find the location in parameter space at which the error is maximized, i.e. find* $\mathbf{z}^* = \arg\max \mathcal{G}(\mathbf{z})$.

2. *If* $\mathcal{G}(\mathbf{z}^*) < \epsilon$*, where* ϵ *is the desired level of accuracy, then terminate the algorithm. If not, go to the next step.*

3. *With* $\mathbf{z} = \mathbf{z}^*$*, solve the full system* (3.3) *to compute the state solutions* $\mathbf{x}(\mathbf{z}^*, t)$*,* $t = (0, t_f)$*,where* t_f *is some time horizon of interest. Use the span of these state solutions to update the basis* $\boldsymbol{\Phi}$ *such that* $\mathcal{G}(\mathbf{z}^*) < \epsilon$*. Go to Step 1.*

3.1.3 Solution of the Greedy Optimization Problem

In order to reduce the offline cost of deriving the reduced model, we would like to minimize the cost of solving the greedy optimization problem, especially when large-scale system equations appear as constraints in (3.3). Therefore, it is important to use an efficient optimization algorithm that allows us to exploit the structure of the system. In order to solve the constrained optimization problem (3.2)–(3.9), we choose to solve an equivalent bound-constrained optimization problem in the $\mathbf{z}$ variables by eliminating the state variables $\mathbf{x}$ and $\mathbf{x}_r$. That is, we replace $\max_{\mathbf{x},\mathbf{x}_r,\mathbf{z}} \mathcal{G}(\mathbf{x}, \mathbf{x}_r, \mathbf{z})$ with $\max_{\mathbf{z}} \mathcal{G}(\mathbf{x}(\mathbf{z}), \mathbf{x}_r(\mathbf{z}), \mathbf{z}) = \max_{\mathbf{z}} \mathcal{G}(\mathbf{z})$, where the dependence of $\mathbf{x}$ and $\mathbf{x}_r$ on $\mathbf{z}$ is implicit through the full equation (3.3) and reduced state equation (3.6) (of course we assume that the full and the reduced equations are well defined in the sense that given a parameter vector $\mathbf{z}$ we can solve for the full and the reduced states). Explicitly, the bound constrained optimization reads

$$\max_{\mathbf{z}} \mathcal{G}(\mathbf{z}) \tag{3.14}$$

subject to

$$\mathbf{z}_{min} \leq \mathbf{z} \leq \mathbf{z}_{max}. \tag{3.15}$$

While the bound-constrained optimization problem (3.14)–(3.15) is small, in the sense that the number of optimization variables is small relative to the size of the state, it may still be expensive to solve. This is because solution of the large-scale system (3.3)–(3.5) is still required at each iteration of the optimization solver.

In the literature, there are several methods to solve the above bound-constrained optimization problem [92–101]. In particular, the method of Coleman-Li [98, 99], and its extensions [95–97, 100, 101] are used here. Since our main goal is to make the cost of solving the optimization problem as small as possible, in the following we will combine the modified Coleman-Li scaling developed by Heinkenschloss *et al.* [95] and the subspace trust region interior reflective method in [101].

The first order necessary optimality system for the bound constrained problem (3.14)–(3.15) can be written in the following form

$$\mathbf{D}(\mathbf{z})\nabla\mathcal{G}(\mathbf{z}) = 0, \tag{3.16}$$

where the diagonal elements of the Coleman-Li diagonal scaling matrix $\mathbf{D}$ are given by

$$D_{ii}(\mathbf{z}) = D_{ii}^{CL}(\mathbf{z}) = \begin{cases} z_i - z_{min}^i & \text{if } \nabla\mathcal{G}_i > 0 \\ z_{max}^i - z_i & \text{if } \nabla\mathcal{G}_i < 0 \\ \min\{z_i - z_{min}^i, z_{max}^i - z_i\} & \text{otherwise} \end{cases}, \tag{3.17}$$

and $\nabla\mathcal{G}_i$ denotes the gradient of $\mathcal{G}$ with respect to z_i.

Next the Newton step for (3.16) at a current optimization point, $\mathbf{z}^k$, can be written as

$$\mathbf{M}(\mathbf{z}^k)\mathbf{s} = -\mathbf{D}(\mathbf{z}^k)\nabla\mathcal{G}(\mathbf{z}^k), \tag{3.18}$$

where $\mathbf{s}$ is the Newton step, and $\mathbf{M} = \mathbf{D}\nabla^2\mathcal{G} + \text{diag}(|\nabla\mathcal{G}_i|)$. Heinkenschloss *et al.* [95] show that the Coleman-Li scaling yields linear convergence for degenerate cases, i.e. $|\nabla\mathcal{G}_i| = 0$ if $z_i^* \in \{z_{min}^i, z_{max}^i\}$. To overcome this, they propose modified Coleman-Li scalings given as

$$D_{ii}(\mathbf{z}) = \begin{cases} D_{ii}^{CL}(\mathbf{z}) & \text{if } |\nabla\mathcal{G}_i| < \min\{z_i - z_{min}^i, z_{max}^i - z_i\}^p \text{ or} \\ & \text{if } \min\{z_i - z_{min}^i, z_{max}^i - z_i\} < |\nabla\mathcal{G}_i|^p \\ 1 & \text{otherwise} \end{cases} \tag{3.19}$$

for some $p > 1$.

The Coleman-Li scaling approach enables us to use the subspace trust region interior reflective Newton framework, proposed in Branch *et al.* [101], to solve the resulting bound-constrained optimization problem efficiently. The subspace trust region subproblem we need to solve is given as

$$\min_{\mathbf{s}\in\mathbb{R}^d}\{\varphi_k(\mathbf{s}) : \|\mathbf{D}\mathbf{s}\|_2 \leq \Delta_k, \mathbf{s} \in \mathcal{S}_k\}, \tag{3.20}$$

where d is the number of parameters, Δ_k is the current trust region radius whose updating rule can be found in [101], and the merit function $\varphi_k(\mathbf{s})$ is given as

$$\varphi_k(\mathbf{s}) = \mathbf{s}^T \nabla \mathcal{G} + \frac{1}{2}\mathbf{s}^T \mathbf{D}^{-1}\mathbf{M}\mathbf{s}. \tag{3.21}$$

We use the conjugate gradient (CG) method to determine the subspace $\mathcal{S}_k$ in which the inexact Newton step $\mathbf{s}$ is found. We terminate the CG subroutine when either of the two following conditions is satisfied: (1) a negative curvature direction is encountered; or (2) the norm of the residual of the Newton system is brought down to a sufficiently small value relative to the norm of the gradient. The subspace $\mathcal{S}_k$ is then constructed from the gradient and the output of the CG solver. Finally, the subspace trust region subproblem (3.20) is solved. This method combines the rapid locally-quadratic convergence rate properties of Newton's method, the effectiveness of trust region globalization for treating ill-conditioned problems, and the Eisenstat-Walker idea of preventing oversolving [102]. In addition, this method has been numerically demonstrated [101] to lead to faster convergence since it better captures the negative curvature information than does the conventional inexact Newton-CG method [103, 104].

The gradient of $\mathcal{G}$ with respect to $\mathbf{z}$, as required by the trust region subproblem, can be computed efficiently by an adjoint method which will be developed in detail for the steady and unsteady problems in the following subsections. The Hessian-vector product as required by CG is computed on-the-fly; because it is a directional derivative of the gradient its computation similarly involves solution of state-like and adjoint-like equations. Therefore, the optimization algorithm requires solution of a pair of state and adjoint systems at each CG iteration. Below is the summary of the optimization algorithm.

Algorithm 3.2 *Bound-Constrained Optimization Solver*

1. *At the current Newton step $\mathbf{z}^k$, compute the gradient $\nabla\mathcal{G}(\mathbf{z}^k)$ (requires a pair of full forward and full adjoint solves, and one pair of reduced forward and reduced adjoint solves).*

2. *Compute the subspace $\mathcal{S}_k$ (each CG iteration requires a pair of full forward- and adjoint-like solves, and a pair of reduced forward- and adjoint-like solves).*

3. *Solve the trust region subproblem* (3.20).

4. *Compute the following quantity*

$$\rho = \frac{\mathcal{G}(\mathbf{z}^k + \mathbf{s}) - \mathcal{G}(\mathbf{z}^k)}{\varphi_k(\mathbf{s})}. \tag{3.22}$$

5. *If $\rho > \nu$, where $\nu < 1$, then set $\mathbf{z}^{k+1} = \mathbf{z}^k + \mathbf{s}$ and $\Delta_{k+1} = 2\Delta_k$. Otherwise, set $\mathbf{z}^{k+1} = \mathbf{z}^k$ and $\Delta_{k+1} = 0.5\Delta_k$. Go to Step 2.*

Note that a very simple updating rule for $\mathbf{z}^k$ and Δ_k is given in Step 5 of Algorithm 3.2 for simplicity of the exposition. In practice, a sophisticated updating rule similar to that of Branch *et al.* [101] is used. Algorithm 3.2 is repeated until some stopping criterion is met, for example, when the norm of the scaled gradient $\|\mathbf{D}\nabla\mathcal{G}\|_2$ is less than some given tolerance τ.

Since the system dependence on the parameter $\mathbf{z}$ is nonlinear, in the general case the optimization problem (3.2)–(3.9) is non-convex. In particular, as the greedy algorithm progresses we expect the cost functional to become increasingly multi-modal, since the error function will be close to zero (below the tolerance ϵ) at each of the previous parameter sample locations. It should be noted that, while finding the global maximum is obviously preferred, convergence to a local maximum is not necessarily an adverse result. Solving the greedy optimization problem is a heuristic to systematically find "good" sample points; at a local maximum the error is (locally) large. The stopping criterion applied in Step 2 of Algorithm 3.1 monitors $\mathcal{G}(\mathbf{z}^*)$, the reduced model error associated with the optimal solution $\mathbf{z}^*$. It is important to note that if $\mathcal{G}(\mathbf{z}^*)$ falls below the desired error level, this guarantees only that the local error between full and reduced model is sufficiently small. Due to the non-convexity of the optimization problem, it is possible that larger errors may exist elsewhere in the parameter space.

3.1.4 Computational Complexity

In this section, we will approximate the cost, in terms of flop counts, of the model-constrained adaptive sampling procedure, Algorithm 3.1. In order to compute the cost, it is helpful to point out that the adaptive sampling procedure involves two main loops. The outer loop runs through the number of greedy cycles, for which Algorithm 3.1 shows one cycle. The inner loop runs through the number of Newton steps, one instance of which is given by Algorithm 3.2. The CG loop, which is inside the inner loop occurring in Step 2 of Algorithm 3.2, is an additional loop which is not shown here. The cost estimation will be only given for steady problems of the form (2.10), but the extensions to problems that are nonlinear in the state vector, and to unsteady problems are straightforward. We further assume that only the matrix $\mathbf{A}$ depends on the parameters $\mathbf{z}$ and is given by the following affine decomposition

$$\mathbf{A}(\mathbf{z}) = \sum_{i=1}^{n_\Theta} \Theta_i(\mathbf{z}) \mathbf{A}_i, \tag{3.23}$$

where $\mathbf{A}_i$ does not depend on $\mathbf{z}$ and Θ_i is some scalar function of $\mathbf{z}$. Finally, all the trial iterates, $\mathbf{z}^k + \mathbf{s}$, are assumed to satisfy $\rho > \nu$ in Step 5 of Algorithm 3.2. (In the case that this condition is not satisfied, the cost will be slightly increased due to additional Hessian-vector products in Step 2 and additional solves in Step 4, because one has to restart Step 2 of Algorithm 3.2 with smaller trust region radius.)

The order of approximating the cost is as follows. The cost of each Newton step is first approximated, which is then summed over all Newton steps to form the cost for each greedy cycle. The total offline cost is then the sum of the cost of all greedy cycles.

In order to solve the full model, we use the state-of-the-art direct sparse solver UMFPACK [105–108]. Since we consider problems that are linear in the state vector, we can perform the LU factorization of the full matrix at the beginning of Step 1 of Algorithm 3.2, and then use the full LU factors for the full forward and full adjoint solves and all subsequent full forward- and adjoint-like solves required by the CG method within that Newton step. Similarly, the LU factorization for the reduced

model needs to be computed once at the beginning of Step 1 of Algorithm 3.2, and the LU factors can be then used for the reduced forward and reduced adjoint solves and all subsequent reduced forward- and adjoint-like solves required by the CG method.

Table 3.1 shows the cost in terms of flop counts for each instance of Algorithm 3.2 (each Newton step). At the beginning of Step 1 of Algorithm 3.2, we need one full LU factorization, and its cost is $\mathcal{O}(LU_f^F)$, where LU_f^F is the number of multiplicative flops (* and /) [109]. The number of triangular solves for each forward (or each adjoint solve) is one (it is n_t, the number of timesteps, for unsteady problems). At the end of Step 1, we need two (one forward and one adjoint) triangular solves. Denote the number of Hessian-vector products in Step 2 as n_{Hv} (which is at most d for the CG method in exact arithmetic) which is assumed to be the same for all instances of Algorithm 3.2. Since each CG iteration requires one pair of full forward- and adjoint-like solves, the number of triangular solves for the CG solver is $2n_{Hv}$. As a result, we have a total of $2(n_{Hv}+1)$ triangular solves for one instance of Algorithm 3.2; the total cost for all triangular solves is therefore $2(n_{Hv}+1)\mathcal{O}(LU_s^F)$, where LU_s^F is the number of flop counts for each triangular solve. Similarly, the flop counts related to the reduced model are shown in Table 3.1. Since the reduced model is dense, the explicit flop counts for LU factorization and triangular solve are available in terms of the reduced model size m_j, where j denotes the j^{th} greedy cycle.

Table 3.1: Cost in terms of flop counts for the bound-constrained optimization solver for each Newton step for steady problems. The true error is used as the cost functional.

	LU-factorization		Triangular solves	
	Number	Cost	Number	Cost
Full	1	$\mathcal{O}(LU_f^F)$	$2(n_{Hv}+1)$	$\mathcal{O}(LU_s^F)$
Reduced	1	$\mathcal{O}(\frac{2}{3}m_j^3)$	$2(n_{Hv}+1)$	$\mathcal{O}(m_j^2)$

The numerical experiments in Sections 4.3.2 and 6.2 suggest that the number of Newton steps, and hence the number of instances of Algorithm 3.2, scales linearly with the number of parameters; we therefore assume the number of Newton steps to be $\mathcal{O}(d)$. As a result, the offline cost (flop counts) of computing a reduced basis with

m basis vectors after n_G greedy cycles, excluding the cost of forming the reduced model and the cost of computing the reduced basis at each greedy cycle, scales as

$$\text{Cost}^{TE} \sim \mathcal{O}\left(\underbrace{d\sum_{j=1}^{n_G}\frac{2}{3}m_j^3 + 2(1+n_{Hv})m_j^2}_{\text{cost}^R} + \underbrace{dn_G[LU_f^F + 2(1+n_{Hv})LU_s^F]}_{\text{cost}^F}\right), \quad (3.24)$$

where cost^R and cost^F denote the cost incurred by solving the intermediate reduced model and by solving the full model, respectively, during the adaptive sampling process.

The offline cost in terms of flop counts in (3.24) is for the model-constrained adaptive sampling method with the greedy optimization problem of the form (3.2)–(3.9), that is, the true error is used as the sampling selection criteria. For the adaptive sampling method with an error indicator, such as the residual norm, the offline cost is given by

$$\text{Cost}^{IE} \sim \mathcal{O}\left(\underbrace{d\sum_{j=1}^{n_G}\frac{2}{3}j^3 + 2(1+n_{Hv})j^2}_{\text{cost}^R} + \underbrace{n_G(LU_f^F + LU_s^F)}_{\text{cost}^F}\right), \quad (3.25)$$

which involves only n_G full system solves at n_G worst-case sampling points. Since only one snapshot at the optimal point is computed in this case, only one basis vector is added to the current reduced basis. As a result, m_j is equal to j at the end of the j^{th} greedy cycle. It can be seen that if the costs of the full LU factorization and full triangular solves dominate the other costs, the error-indicator approach is much less expensive than the true-error approach per greedy cycle.

If a good preconditioner for the CG solver is available so that the number of Hessian-vector products n_{Hv} does not depend on the number of parameters d, the offline cost of constructing reduced basis vectors in each greedy cycle scales linearly with the dimension of the parameter space, d. This is clearly important as the dimension of the parameter space increases. If, in addition, the size of the reduced basis is

constrained to be fixed, and hence the number of greedy cycles, n_G, is fixed, it can be also proved that the total offline cost of constructing the reduced basis scales linearly with the dimension of the parameter space. However, if n_G is allowed to increase until Algorithm 3.1 terminates, it will likely depend on the dimension of the parameter space, i.e. it might be expected that larger input spaces require more sample points, and therefore larger reduced basis size, as well on complexity of dependence of outputs on parameters. In that case, the total offline cost of constructing the reduced basis no longer scales linearly with the dimension of the parameter space.

It should be pointed out that in general there exists no good explicit approximation for $\mathcal{O}(LU_f^F)$ (or $\mathcal{O}(LU_s^F)$) (of course a very conservative upper bound is $\mathcal{O}(\frac{2}{3}n^3)$ for $\mathcal{O}(LU_f^F)$ and $\mathcal{O}(2n^2)$ for $\mathcal{O}(LU_s^F)$ from dense matrix theory). However, for any well-shaped finite element mesh [109, 110], the flop counts can be approximated as $\mathcal{O}(LU_f^F) + \mathcal{O}(LU_s^F) = \mathcal{O}(n^{3/2})$ for two-dimensional problems and $\mathcal{O}(LU_f^F) + \mathcal{O}(LU_s^F) = \mathcal{O}(n^2)$ for three-dimensional problems. Furthermore, the storage requirement for the LU factors is $\mathcal{O}(n\log n)$ for two-dimensional problems and $\mathcal{O}(n^{4/3})$ for three-dimensional problems. Therefore, for three-dimensional problems, a direct solver may not be possible, and an iterative sparse solver is an alternative [111].

To estimate the cost of forming the reduced model after each adaptive cycle, we first consider the error-indicator approach for simplicity. The discussion for the cost estimation of the true-error will then follow. Since the matrix $\mathbf{A}$ is assumed to be of the form (3.23), we compute the cost of forming the reduced model explicitly. From the reduced equation (2.15), the reduced matrix is given as

$$\mathbf{A}_r = \sum_{k=1}^{n_\Theta} \sum_{l=1}^{n_\Theta} \Theta_k(\mathbf{z}) \Theta_l(\mathbf{z}) \mathbf{\Phi}^T \mathbf{A}_k^T \mathbf{A}_l \mathbf{\Phi}, \tag{3.26}$$

which suggests that we need to compute and store the matrices

$$\mathbf{A}_r^{kl} = \mathbf{\Phi}^T \mathbf{A}_k^T \mathbf{A}_l \mathbf{\Phi}. \tag{3.27}$$

It should be noted that the dimension of the reduced matrix $\mathbf{A}_r^{kl}$ computed at the

end of the $(j+1)^{th}$ cycle is $(j+1) \times (j+1)$, but its $j \times j$ principle submatrix is the reduced matrix at the end of the j^{th} cycle. Therefore, one only needs to update the $(j+1)^{th}$ row and $(j+1)^{th}$ column of $\mathbf{A}_r^{kl}$ after the $(j+1)^{th}$ greedy cycle. Next, denote nnz to be the number of nonzeros of $\mathbf{A}_i$, and assume it is the same for all $\mathbf{A}_i$. To compute the first j elements of the $(j+1)^{th}$ column, we need to compute

$$\mathbf{\Phi}^T \mathbf{A}_k^T \mathbf{A}_l \Phi_{j+1}, \tag{3.28}$$

where Φ_{j+1} is the new basis vector computed at the end of the $(j+1)^{th}$ cycle. Multiplying from right to left, the cost is approximately given as $2jn + 2nnz$, which is also the cost of computing the first j elements of the $(j+1)^{th}$ row. Finally, the cost of computing the $(j+1)^{th}$ element of the $(j+1)^{th}$ column is given as $2n + 2nnz$. The total cost of updating the reduced matrix $\mathbf{A}_r^{kl}$ is given as $2(2j+1)n+6nnz$. Therefore the total cost to compute all the reduced matrices at the end of the $(j+1)^{th}$ greedy cycle is given as

$$n_\Theta^2[2(2j+1)n + 6nnz], \tag{3.29}$$

and the corresponding storage requirement is $n_\Theta^2(j+1)^2$. It can be seen that, depending on n_Θ, j, n, and nnz, the cost of computing the intermediate reduced matrices can form a considerable portion of the total offline cost.

For the true-error case, the cost of forming the reduced model after each greedy cycle is the same as that of the error-indicator case if only the snapshot at the optimal parameter point is used to update the reduced basis. However, as will be discussed in Section 3.1.6, the size of the reduced model depends on how we update the reduced basis, and hence the cost could be slightly increased. Nonetheless, the procedure for estimating the cost of forming the reduced basis is similar.

It must be also mentioned that computing reduced basis vectors and using them to update the reduced basis can take a considerable amount of time (due to dense linear algebra operations on these dimension n reduced basis vectors) and storage, depending on how many basis vectors are generated and on the size of the full model, n.

3.1.5 Initialization

Initially, there are no basis vectors in the reduced basis; it is therefore natural to choose the initial basis as the empty set, $\mathbf{\Phi} = \emptyset$, and the reduced model is a zero-order approximation of the full model.

It should be also pointed out that the Algorithm 3.2 starts with an initial guess $\mathbf{z}^0$ in the parameter space and moves iteratively towards a local maximizer. To avoid convergence to a local maximum close to a previous sample location, and hence to explore the parameter space better, a random initialization of the optimization variables $\mathbf{z}$ is used. An initial guess is accepted only if it is "far enough" away from the previous sample locations and its corresponding cost functional is larger than ϵ. In particular, the smallest allowable distance between an initial guess and all the existing sample locations is chosen, for example, to be $0.5 \min_i \{z^i_{max} - z^i_{min}\}$.

3.1.6 Updating the Reduced Basis

Recall that the purpose of Algorithm 3.1 is to sample the (locally) optimal parameter points in the parameter space. Therefore, the snapshots at these optimal sample locations are important, and it is natural to use these snapshots to update the reduced basis as proposed in Step 3 of Algorithm 3.1. However, if the true output error is used as the cost functional, besides the snapshots computed at the optimal solutions, the solution snapshots and the adjoint solutions at each Newton step are also available as part of the optimization process; although these snapshots may not add much information as those at the optimal parameters, they can be used to improve the reduced basis. Given all these snapshots, some approaches to update the reduced basis are given as follows

1. Use only the snapshots at the optimal parameter points, $\mathbf{z}^*$, to update the reduced basis, for example, using the Gram-Schmidt procedure.

2. Store all snapshots, and then perform the POD method on the complete snapshot set, which comprises those of the current greedy cycle and all previous ones. For large-scale problems, this approach is, however, expensive both in terms of

storage and in terms of computing time. A less expensive approach is first to perform the POD method on just the snapshots of the current greedy cycle in order to extract dominant POD basis vectors. These dominant POD vectors are then added to the current reduced basis using Gram-Schmidt orthogonalization. Another approach is to perform a second POD computation on the snapshot set comprising the current reduced basis and the newly computed POD vectors. In that case, since the POD basis vectors are of unit length, an appropriate scaling, for example using the corresponding singular values, needs to be done before the second POD computation. Otherwise, the important information compressed in the dominant POD vectors of the current reduced basis and of the POD basis of the current greedy cycle, may be lost, resulting in a reduced basis that is poor in quality.

3. Add all snapshots of the current greedy cycle to the current reduced basis using Gram-Schmidt orthogonalization. However, care must be taken since this method potentially makes both the offline and the online stages expensive due to a large number of vectors in the reduced basis. Yet, the reduced model may not be accurate because not all the snapshots are important.

4. Solve an (inner) optimization problem to find the basis that minimizes the output error at the sample points at which the snapshots are computed [112].

If the error indicator is used, only snapshots at the optimal parameter points are computed. As a result, if the number of optimal parameter points, and hence the number of snapshots, is small, one can simply use the first updating approach, as discussed above. However, if the number of snapshots is large, all the above approaches can be employed.

3.2 An Analysis of the Adaptive Sampling Approach

We have proposed an adaptive sampling method for parameter-dependent problems in Section 3.1. Recall that the optimization problem in each adaptive sampling procedure is a PDE-constrained optimization problem. As a result, the question arises under what conditions the adaptive sampling procedure works. In particular, we need to answer the following questions: 1) Does the PDE-constrained optimization problem in each adaptive cycle have a solution? 2) If it does, is the solution unique? That is, the existence and uniqueness of the solution of the optimization problem need to be addressed.

3.2.1 Existence and Uniqueness of a Maximizer in Each Adaptive Cycle

In this section, we prove that a solution of the optimization problem (3.14)–(3.15), and hence the optimization problem (3.2)–(3.9) or (3.10)–(3.13), exists, and discuss the uniqueness of that solution. To begin, let us recall one of the fundamental theorems about continuous functions.

Theorem 3.1 *If $\mathcal{G}(z_1, \ldots, z_d) : \Omega \subset I\!R^d \rightarrow I\!R$ is continuous and Ω is a compact subset of $I\!R^d$, then there is a maximum point $\bar{\mathbf{z}} \in \Omega$ such that $\mathcal{G}(\mathbf{z}) \leq \mathcal{G}(\bar{\mathbf{z}}), \forall \mathbf{z} \in \Omega$.*

Proof: The proof can be found, for example, in [113]. □

This theorem shows that the greedy optimization problem in each adaptive cycle has at least one solution by the following corollary.

Corollary 3.1 *In each adaptive cycle, assume that the cost functional is a continuous function in the parameter $\mathbf{z}$, then there exists at least one solution for the optimization problem* (3.14)–(3.15).

Proof: Denote Ω to be the parameter set defined by the bound constraints (this is called a d-cell in mathematics [113]). Then, it can be proved that a d-cell is compact

(the proof requires a few other theorems in set topology study and therefore we simply use this result).

On the other hand, the states are eliminated so that the cost function $\mathcal{G}(\mathbf{z})$ is only a function of the parameter $\mathbf{z}$. Therefore, if cost functional $\mathcal{G}(\mathbf{z})$ is a continuous function on Ω, Theorem 3.1 applies and Corollary 3.1 is proved. That is, there exists a solution (a global maximizer according to the theorem) to the optimization problem (3.14)–(3.15) in each adaptive cycle. □

Clearly uniqueness is not guaranteed in the general case since there could be many global maximizers.

3.2.2 Properties of the Adaptive Sampling Approach

Next, some important properties of the adaptive sampling approach will be discussed.

Theorem 3.2 *Assume that the full model is linear in state* $\mathbf{x}$*. Then, in the* k^{th} *adaptive cycle, the cost functional is less than* ϵ *at all the maximizers found in the previous cycles* $\bar{k} < k$*.*

Proof: Recall that in Step 3 of Algorithm 3.1 the span of the state solutions at the local maximizers found in previous cycles $\bar{k} < k$ are used as basis vectors such that the cost functional at these local maximizers is less than ϵ. Furthermore, we proved in Chapter 2 that, for problems that are linear in state $\mathbf{x}$, as the reduced basis is enriched, the reduced model error cannot increase. As a result, the cost functional in the k^{th} adaptive cycle is less than ϵ at all the maximizers found in the previous cycles $\bar{k} < k$. □

As a consequence of the above theorem, the below corollary is an important result for the adaptive sampling approach.

Corollary 3.2 *Assume that the full model is linear in state* $\mathbf{x}$*. Then, the adaptive sampling approach will never sample at the previous sampled points in the parameter space.*

Proof: By definition in (3.2), the cost functional is non-negative. To prove the corollary, it is sufficient to show that in the k^{th} adaptive cycle the maximizer must be different from the maximizers found in the previous cycles. First, recall that the cost functional in the current adaptive cycle is smaller than ϵ at all previous maximizers, as proved in Theorem 3.2. Second, we only start at an initial guess where the cost function is greater than ϵ. Third, the optimization solver only accepts an iterate if the cost functional is larger than that at the previous iterate. Using these three facts we conclude that the cost functional at a new maximizer must be larger than ϵ. Therefore, the maximizer found in the k^{th} adaptive cycle must be different from the previous maximizers. □

Even though we do not yet have proofs that the results in this section are true for general problems that are nonlinear in both state and parameters, this is not a severe limitation if, for example, one uses the empirical interpolation method [61, 62, 68,69,114,115] to pre-process the nonlinear terms by a linear combination of empirical interpolation basis functions. In that case, the results in this section still apply for the pre-processed model. Of course, if one can also prove that, for a problem at hand that is nonlinear in state $\mathbf{x}$, the reduced model is improved as the reduced basis is enriched, both Theorem 3.2 and Corollary 3.2 also hold for that nonlinear problem.

We next derive the optimization problem in each adaptive cycle and its optimality system for a class of steady and unsteady problems that are linear in the state vector, but depend nonlinearly on the parameters.

3.3 Steady Problems

Consider a general large-scale parameter-dependent steady problem that is linear in the state vector $\mathbf{x}$

$$\mathbf{A}(\mathbf{z})\mathbf{x} = \mathbf{B}(\mathbf{z}); \quad \mathbf{y} = \mathbf{C}(\mathbf{z})\mathbf{x}, \quad \mathbf{z}_{min} \leq \mathbf{z} \leq \mathbf{z}_{max}. \tag{3.30}$$

As discussed in Section 2.1, a projection-based model order reduction technique yields the reduced system of the form

$$\mathbf{A}_r(\mathbf{z})\mathbf{x}_r = \mathbf{B}_r(\mathbf{z}); \quad \mathbf{y}_r = \mathbf{C}_r(\mathbf{z})\mathbf{x}_r, \quad \mathbf{z}_{min} \leq \mathbf{z} \leq \mathbf{z}_{max}, \tag{3.31}$$

where $\mathbf{A}_r(\mathbf{z}) = \mathbf{\Psi}^T\mathbf{A}(\mathbf{z})\mathbf{\Phi}$, $\mathbf{B}_r(\mathbf{z}) = \mathbf{\Psi}^T\mathbf{B}(\mathbf{z})$ and $\mathbf{C}_r(\mathbf{z}) = \mathbf{C}(\mathbf{z})\mathbf{\Phi}$. Note that the full and the reduced models were given in (2.10) and (2.12); they are repeated here for clarity. Next, define the cost functional

$$\mathcal{G}(\mathbf{x}, \mathbf{x}_r, \mathbf{z}) = \frac{1}{2}\|\mathbf{y} - \mathbf{y}_r\|_O^2 = \frac{1}{2}\|\mathbf{y} - \mathbf{y}_r\|_2^2, \tag{3.32}$$

which describes the output error between the full and the reduced models. Here, $\|\cdot\|_2$ is the Euclidean *2-norm* (any weighted combination of the outputs could be used instead). The greedy optimization problem in each adaptive cycle then reads

$$\max_{\mathbf{x},\mathbf{x}_r,z} \mathcal{G} = \frac{1}{2}\|\mathbf{C}(\mathbf{z})\mathbf{x} - \mathbf{C}_r(\mathbf{z})\mathbf{x}_r\|_2^2, \tag{3.33}$$

subject to

$$\begin{aligned} \mathbf{A}(\mathbf{z})\mathbf{x} &= \mathbf{B}(\mathbf{z}), & (3.34)\\ \mathbf{A}_r(\mathbf{z})\mathbf{x}_r &= \mathbf{B}_r(\mathbf{z}), & (3.35)\\ \mathbf{z}_{min} \leq \ \mathbf{z} &\leq \mathbf{z}_{max}. & (3.36) \end{aligned}$$

The optimality conditions for the constrained optimization problem (3.33)–(3.36) can be derived by defining the Lagrangian functional

$$\begin{aligned} \mathcal{L}(\mathbf{x}, \mathbf{x}_r, \mathbf{z}, \lambda, \lambda_r) &= \mathcal{G}(\mathbf{x}, \mathbf{x}_r, \mathbf{z}) + \lambda^T\left[\mathbf{A}(\mathbf{z})\mathbf{x} - \mathbf{B}(\mathbf{z})\right] \\ &+ \lambda_r^T\left[\mathbf{A}_r(\mathbf{z})\mathbf{x}_r - \mathbf{B}_r(\mathbf{z})\right], \end{aligned} \tag{3.37}$$

where $\lambda \in \mathbb{R}^n$ and $\lambda_r \in \mathbb{R}^m$ are the full and the reduced adjoint variables that respectively enforce the full and the reduced equations. Note that the bound constraints are

excluded and treated separately as in Section 3.1.3. The first-order Karush-Kuhn-Tucker optimality system can be derived by taking derivatives of the Lagrangian with respect to the adjoints, states, and parameter as follows:

- Setting the derivative of the Lagrangian with respect to λ to zero yields the full equations (3.34).

- Setting the derivative of the Lagrangian with respect to λ_r to zero yields the reduced equations (3.35).

- Setting the derivative of the Lagrangian with respect to $\mathbf{x}$ to zero yields the full adjoint equations

$$\mathbf{A}^T(\mathbf{z})\lambda = \mathbf{C}^T(\mathbf{z})\left[\mathbf{C}_r(\mathbf{z})\mathbf{x}_r - \mathbf{C}(\mathbf{z})\mathbf{x}\right]. \tag{3.38}$$

- Setting the derivative of the Lagrangian with respect to $\mathbf{x}_r$ to zero yields the reduced adjoint equations

$$\mathbf{A}_r^T(\mathbf{z})\lambda_r = \mathbf{C}_r^T(\mathbf{z})\left[\mathbf{C}(\mathbf{z})\mathbf{x} - \mathbf{C}_r(\mathbf{z})\mathbf{x}_r\right]. \tag{3.39}$$

- Setting the derivative of the Lagrangian with respect to z_i to zero yields the optimality conditions

$$\begin{aligned} & \left[\frac{\partial \mathbf{C}}{\partial z_i}(\mathbf{z})\mathbf{x} - \frac{\partial \mathbf{C}_r}{\partial z_i}(\mathbf{z})\mathbf{x}_r\right]^T \left[\mathbf{C}(\mathbf{z})\mathbf{x} - \mathbf{C}_r(\mathbf{z})\mathbf{x}_r\right] + \lambda^T \left[\frac{\partial \mathbf{A}}{\partial z_i}(\mathbf{z})\mathbf{x} - \frac{\partial \mathbf{B}}{\partial z_i}(\mathbf{z})\right] \\ + \quad & \lambda_r^T \left[\frac{\partial \mathbf{A}_r}{\partial z_i}(\mathbf{z})\mathbf{x}_r - \frac{\partial \mathbf{B}_r}{\partial z_i}(\mathbf{z})\right] = 0. \end{aligned} \tag{3.40}$$

The reduced gradient of the cost functional is then given by

$$\begin{aligned} \frac{\partial \mathcal{G}}{\partial z_i} \quad = \quad & \left[\frac{\partial \mathbf{C}}{\partial z_i}(\mathbf{z})\mathbf{x} - \frac{\partial \mathbf{C}_r}{\partial z_i}(\mathbf{z})\mathbf{x}_r\right]^T \left[\mathbf{C}(\mathbf{z})\mathbf{x} - \mathbf{C}_r(\mathbf{z})\mathbf{x}_r\right] + \lambda^T \left[\frac{\partial \mathbf{A}}{\partial z_i}(\mathbf{z})\mathbf{x} - \frac{\partial \mathbf{B}}{\partial z_i}(\mathbf{z})\right] \\ + \quad & \lambda_r^T \left[\frac{\partial \mathbf{A}_r}{\partial z_i}(\mathbf{z})\mathbf{x}_r - \frac{\partial \mathbf{B}_r}{\partial z_i}(\mathbf{z})\right], \end{aligned} \tag{3.41}$$

where the full state $\mathbf{x}$, reduced state $\mathbf{x}_r$, full adjoint variable λ and reduced adjoint variable λ_r respectively satisfy the full forward equation (3.34), the reduced forward equation (3.35), the full adjoint equation (3.38), and the reduced adjoint equation (3.39).

3.4 Unsteady Problems

For clarity, we recall the results in Section 2.1 that given a general parametrized LTI dynamical system

$$\begin{aligned} \mathbf{E}(\mathbf{z})\dot{\mathbf{x}} &= \mathbf{A}(\mathbf{z})\mathbf{x} + \mathbf{B}(\mathbf{z})\mathbf{u}, && (3.42)\\ \mathbf{y} &= \mathbf{C}(\mathbf{z})\mathbf{x}, && (3.43)\\ \mathbf{x}(0) &= \mathbf{x}^0, && (3.44)\\ \mathbf{z}_{min} \leq\ & \mathbf{z} \leq \mathbf{z}_{max}, && (3.45) \end{aligned}$$

a projection-based model order reduction technique yields the reduced system of the form

$$\begin{aligned} \mathbf{E}_r(\mathbf{z})\dot{\mathbf{x}}_r &= \mathbf{A}_r(\mathbf{z})\mathbf{x}_r + \mathbf{B}_r(\mathbf{z})\mathbf{u}, && (3.46)\\ \mathbf{y}_r &= \mathbf{C}_r(\mathbf{z})\mathbf{x}_r, && (3.47)\\ \mathbf{x}_r(0) &= \mathbf{\Psi}^T\mathbf{x}^0, && (3.48)\\ \mathbf{z}_{min} \leq\ & \mathbf{z} \leq \mathbf{z}_{max}, && (3.49) \end{aligned}$$

where $\mathbf{E}_r(\mathbf{z}) = \mathbf{\Psi}^T\mathbf{E}(\mathbf{z})\mathbf{\Phi}$, $\mathbf{A}_r(\mathbf{z}) = \mathbf{\Psi}^T\mathbf{A}(\mathbf{z})\mathbf{\Phi}$, $\mathbf{B}_r(\mathbf{z}) = \mathbf{\Psi}^T\mathbf{B}(\mathbf{z})$, $\mathbf{C}_r(\mathbf{z}) = \mathbf{C}(\mathbf{z})\mathbf{\Phi}$. We define the cost functional

$$\mathcal{G}(\mathbf{z}) = \frac{1}{2}\|\mathbf{y} - \mathbf{y}_r\|_O^2 = \frac{1}{2}\int_0^{t_f} \|\mathbf{y} - \mathbf{y}_r\|_2^2\, dt = \frac{1}{2}\int_0^{t_f} \|\mathbf{C}(\mathbf{z})\mathbf{x} - \mathbf{C}_r(\mathbf{z})\mathbf{x}_r\|_2^2\, dt, \quad (3.50)$$

which describes the error between the full and reduced models over the parameter space $\mathbf{z}$, integrated over some time horizon of interest t_f. The greedy optimization

problem in each adaptive cycle then reads

$$\max_{\mathbf{x},\mathbf{x}_r,\mathbf{z}} \mathcal{G} = \frac{1}{2}\int_0^{t_f} \|\mathbf{C}(\mathbf{z})\mathbf{x} - \mathbf{C}_r(\mathbf{z})\mathbf{x}_r\|_2^2 \, dt, \tag{3.51}$$

subject to

$$\begin{aligned}
\mathbf{E}(\mathbf{z})\dot{\mathbf{x}} &= \mathbf{A}(\mathbf{z})\mathbf{x} + \mathbf{B}(\mathbf{z})\mathbf{u}, && (3.52)\\
\mathbf{x}(0) &= \mathbf{x}^0, && (3.53)\\
\mathbf{E}_r(\mathbf{z})\dot{\mathbf{x}}_r &= \mathbf{A}_r(\mathbf{z})\mathbf{x}_r + \mathbf{B}_r(\mathbf{z})\mathbf{u}, && (3.54)\\
\mathbf{x}_r(0) &= \mathbf{\Psi}^T\mathbf{x}^0, && (3.55)\\
\mathbf{z}_{min} \leq \ \mathbf{z} &\leq \mathbf{z}_{max}. && (3.56)
\end{aligned}$$

The optimality conditions for the constrained optimization problem (3.51)–(3.56) can be derived by defining the Lagrangian functional

$$\begin{aligned}
&\mathcal{L}(\mathbf{z},\mathbf{x},\mathbf{x}_r,\lambda,\lambda_r,\gamma,\gamma_r) = \frac{1}{2}\int_0^{t_f} \|\mathbf{C}(\mathbf{z})\mathbf{x} - \mathbf{C}_r(\mathbf{z})\mathbf{x}_r\|_2^2 \, dt\\
&+ \int_0^{t_f} \lambda^T \left[\mathbf{E}(\mathbf{z})\dot{\mathbf{x}} - \mathbf{A}(\mathbf{z})\mathbf{x} - \mathbf{B}(\mathbf{z})\mathbf{u}\right] \, dt + \int_0^{t_f} \lambda_r^T \left[\mathbf{E}_r(\mathbf{z})\dot{\mathbf{x}}_r - \mathbf{A}_r(\mathbf{z})\mathbf{x}_r - \mathbf{B}_r(\mathbf{z})\mathbf{u}\right] \, dt\\
&+\gamma^T \left[\mathbf{x}(0) - \mathbf{x}^0\right] + \gamma_r^T \left[\mathbf{x}_r(0) - \mathbf{\Psi}^T\mathbf{x}^0\right],
\end{aligned} \tag{3.57}$$

where $\lambda \in \mathbb{R}^n$ and $\lambda_r \in \mathbb{R}^m$ are the full and the reduced adjoint variables that respectively enforce the full and the reduced equations. Two additional adjoint variables $\gamma \in \mathbb{R}^n$ and $\gamma_r \in \mathbb{R}^m$ are used to enforce the initial conditions for the full and the reduced models, respectively. Note that the bound constraints are excluded and treated separately as in Section 3.1.3. The first-order Karush-Kuhn-Tucker optimality system can be derived by taking variations of the Lagrangian with respect to the adjoints, states, and parameter as follows:

- Setting the first variation of the Lagrangian with respect to λ to zero, and arguing that the variation of λ is arbitrary in $(0, t_f)$, yield the full equation (3.52).

- Setting the first variation of the Lagrangian with respect to λ_r to zero, and arguing that the variation of λ_r is arbitrary in $(0, t_f)$, yield the reduced equation (3.54).

- Setting the derivative of the Lagrangian with respect to γ to zero yields the full initial condition equation (3.53).

- Setting the derivative of the Lagrangian with respect to γ_r to zero yields the reduced initial condition equation (3.55).

- Setting the first variation of the Lagrangian with respect to $\mathbf{x}$ to zero, and arguing that the variation of $\mathbf{x}$ is arbitrary in $(0, t_f)$, at $t = 0$ and at $t = t_f$, yield the full adjoint equation, final condition and definition of γ

$$\begin{aligned} \mathbf{E}^T(\mathbf{z})\dot{\lambda} + \mathbf{A}^T(\mathbf{z})\lambda &= \mathbf{C}^T(\mathbf{z})\left[\mathbf{C}(\mathbf{z})\mathbf{x} - \mathbf{C}_r(\mathbf{z})\mathbf{x}_r\right], && (3.58) \\ \lambda(T) &= 0, && (3.59) \\ \gamma &= \mathbf{E}^T(\mathbf{z})\lambda(0). && (3.60) \end{aligned}$$

- Setting the first variation of the Lagrangian with respect to $\mathbf{x}_r$ to zero, and arguing that the variation of $\mathbf{x}_r$ is arbitrary in $(0, t_f)$, at $t = 0$ and at $t = t_f$, yield the reduced adjoint equation, final condition and definition of γ_r

$$\begin{aligned} \mathbf{E}_r^T(\mathbf{z})\dot{\lambda}_r + \mathbf{A}_r^T(\mathbf{z})\lambda_r &= \mathbf{C}_r^T(\mathbf{z})\left[\mathbf{C}_r(\mathbf{z})\mathbf{x}_r - \mathbf{C}(\mathbf{z})\mathbf{x}\right], && (3.61) \\ \lambda_r(T) &= 0, && (3.62) \\ \gamma_r &= \mathbf{E}_r^T(\mathbf{z})\lambda_r(0). && (3.63) \end{aligned}$$

- Setting the derivative of the Lagrangian with respect to z_i to zero yields the

optimality condition

$$\begin{aligned}
&\int_0^{t_f} \left[\frac{\partial \mathbf{C}}{\partial z_i}(\mathbf{z})\mathbf{x} - \frac{\partial \mathbf{C}_r}{\partial z_i}(\mathbf{z})\mathbf{x}_r\right]^T [\mathbf{C}(\mathbf{z})\mathbf{x} - \mathbf{C}_r(\mathbf{z})\mathbf{x}_r]\, dt \\
&+ \int_0^{t_f} \lambda^T \left[\frac{\partial \mathbf{E}}{\partial z_i}(\mathbf{z})\dot{\mathbf{x}} - \frac{\partial \mathbf{A}}{\partial z_i}(\mathbf{z})\mathbf{x} - \frac{\partial \mathbf{B}}{\partial z_i}(\mathbf{z})\mathbf{u}\right] dt \\
&+ \int_0^{t_f} \lambda_r^T \left[\frac{\partial \mathbf{E}_r}{\partial z_i}(\mathbf{z})\dot{\mathbf{x}}_r - \frac{\partial \mathbf{A}_r}{\partial z_i}(\mathbf{z})\mathbf{x}_r - \frac{\partial \mathbf{B}_r}{\partial z_i}(\mathbf{z})\mathbf{u}\right] dt \\
&-(\gamma^T + \gamma_r^T \mathbf{\Psi}^T)\frac{\partial \mathbf{x}^0}{\partial z_i} = 0.
\end{aligned} \tag{3.64}$$

The reduced gradients of the cost functional with respect to $\mathbf{z}$ are then given by

$$\begin{aligned}
\frac{\partial \mathcal{G}}{\partial z_i} \;=\; &\int_0^{t_f} \left[\frac{\partial \mathbf{C}}{\partial z_i}(\mathbf{z})\mathbf{x} - \frac{\partial \mathbf{C}_r}{\partial z_i}(\mathbf{z})\mathbf{x}_r\right]^T [\mathbf{C}(\mathbf{z})\mathbf{x} - \mathbf{C}_r(\mathbf{z})\mathbf{x}_r]\, dt \\
&+ \int_0^{t_f} \lambda^T \left[\frac{\partial \mathbf{E}}{\partial z_i}(\mathbf{z})\dot{\mathbf{x}} - \frac{\partial \mathbf{A}}{\partial z_i}(\mathbf{z})\mathbf{x} - \frac{\partial \mathbf{B}}{\partial z_i}(\mathbf{z})\mathbf{u}\right] dt \\
&+ \int_0^{t_f} \lambda_r^T \left[\frac{\partial \mathbf{E}_r}{\partial z_i}(\mathbf{z})\dot{\mathbf{x}}_r - \frac{\partial \mathbf{A}_r}{\partial z_i}(\mathbf{z})\mathbf{x}_r - \frac{\partial \mathbf{B}_r}{\partial z_i}(\mathbf{z})\mathbf{u}\right] dt, \\
&-(\gamma^T + \gamma_r^T \mathbf{\Psi}^T)\frac{\partial \mathbf{x}^0}{\partial z_i},
\end{aligned} \tag{3.65}$$

where the full state $\mathbf{x}$, reduced state $\mathbf{x}_r$, full adjoint variable λ and reduced variable λ_r respectively satisfy the full forward equation (3.52), the reduced forward equation (3.54), the full adjoint equation (3.58)–(3.59), and the reduced adjoint equation (3.61)–(3.62). In addition, the adjoints γ and γ_r satisfy (3.60) and (3.63), respectively.

Note that derivation of the optimization problem for discrete time systems and their corresponding optimality conditions is similar, and hence omitted here.

A question one may immediately ask is what kind of input, $\mathbf{u} = \mathbf{u}(t)$ we should use in the optimization problem. The answer is given by the following proposition.

Proposition 3.1 *Assume the full system* (3.52) *is a single-input multi-output (SIMO) dynamical system. If the output error norm for the impulse response is bounded, i.e.,*

$$\|g(\mathbf{z})\|_O^2 = \int_0^{t_f} \|(h(\tau) - h_r(\tau))\|_2^2\, d\tau \leq \epsilon$$

where $h(t)$ and $h_r(t)$ are the impulse responses of the full and reduced models, respectively, then the error norm for an arbitrary input u with finite norm is also bounded, i.e.,

$$\frac{\|G(\mathbf{z})\|_O^2}{\|u\|_I^2} \leq \epsilon$$

where the norm of the input is given as $\|u\|_I^2 = \int_0^{t_f} \int_0^t u(t-\tau)^2 \, d\tau \, dt$.

Proof: Using the definition of the output norm, convolution integral and Cauchy-Schwarz inequality gives

$$\begin{aligned}
\|G(\mathbf{z})\|_O^2 = \int_0^{t_f} \|y - y_r\|_2^2 \, dt &= \int_0^{t_f} \left\| \int_0^t \left(h(\tau) - h_r(\tau)\right) u(t-\tau) \, d\tau \right\|_2^2 dt \\
&\leq \int_0^{t_f} \int_0^t \|(h(\tau) - h_r(\tau))\|_2^2 \, d\tau \int_0^t u(t-\tau)^2 \, d\tau \, dt \\
&\leq \int_0^{t_f} \|(h(\tau) - h_r(\tau))\|_2^2 \, d\tau \int_0^{t_f} \int_0^t u(t-\tau)^2 \, d\tau \, dt \leq \epsilon \|u\|_I^2.
\end{aligned}$$

□

It should be pointed out that it is sufficient to consider a SIMO system since a multi-input multi-output system is equivalent to a superposition of SIMO systems.

The above proposition is important for the greedy optimization process, that is, we need to only consider the impulse input in the adaptive sampling procedure. Once the error of the reduced model is small for the impulse input, it will be small for an arbitrary input with finite norm. While the result in Proposition 3.1 is general, in practice the frequency content of the input is sometimes known *a priori*. In that case, an appropriate input (i.e. not the impulse) with the given frequency content could be used.

For problems that are linear in state vector $\mathbf{x}$, one can transform the full and the reduced equations from the time domain into the frequency domain. In the frequency domain, the problem can be considered as a steady system, and the forcing input is replaced by the frequency ω. The trade-off here is that the size of the system is double, though the system has some structure, and the frequency ω is an additional parameter which is assumed to be in the range $\omega_{min} \leq \omega \leq \omega_{max}$, where $[\omega_{min}, \omega_{max}]$

is the frequency range of interest. However, conceptually one could solve a greedy optimization problem, by exploiting the special structure of the problem, to select frequencies within this range at which snapshots should be computed.

Chapter 4

Steady Thermal Fin Heat Conduction Models

In this chapter, we consider the thermal fin design problem adopted from [116]. While the thermal fin geometry in [116] is fixed, it is allowed to vary in this thesis. As a result, our problem is described by thirty-four parameters. The detail of the problem in both physical domain and computational domain is first described. The application of the model-constrained greedy-based sampling approach developed in Chapter 3 to obtain reduced models for the thermal fin problem is then performed for different number of parameters. Finally, the thermal fin optimal design problem with 11 parameters using the reduced model is presented and compared to that obtained using the full model.

4.1 Physical Domain Formulation

The two-dimensional thermal fin is shown in Figure 4-1. It consists of the vertical post and four horizontal subfins. The purpose of the thermal fin is, for example, to conduct heat from some machine, which is attached to the root, through the large-surface-area subfins to the surrounding flowing air. The vertical post has eight sub-regions denoted as $\bar{\Omega}_9, \ldots, \bar{\Omega}_{16}$ with corresponding thermal conductivities $\kappa_9, \ldots, \kappa_{16}$ (e.g. these regions could be made of different materials). Each subfin has two different

sub-regions, e.g. $\bar{\Omega}_{2i-1}$ and $\bar{\Omega}_{2i}$ for the i^{th} subfin, where $i = 1, \ldots, 4$, and these sub-regions have different thermal conductivities denoted as $\kappa_1, \ldots, \kappa_8$. In addition, the size of all the sub-regions of the post and the subfins could be varied and they are denoted as $b_1, \ldots, b_{17}$, as shown in Figure 4-1. Another parameter of interest is the Biot number, Bi, which characterizes the convective heat to the air at the fin surfaces. Therefore, we have in total thirty-four parameters, which are represented by the vector of parametric inputs $\mathbf{z}$ as $\mathbf{z} = \{z_1, \ldots, z_{34}\}$, where $z_i = \kappa_i,\ i = 1, \ldots, 16$, $z_{17} = Bi$, and $z_{17+j} = b_j,\ j = 1, \ldots, 17$.

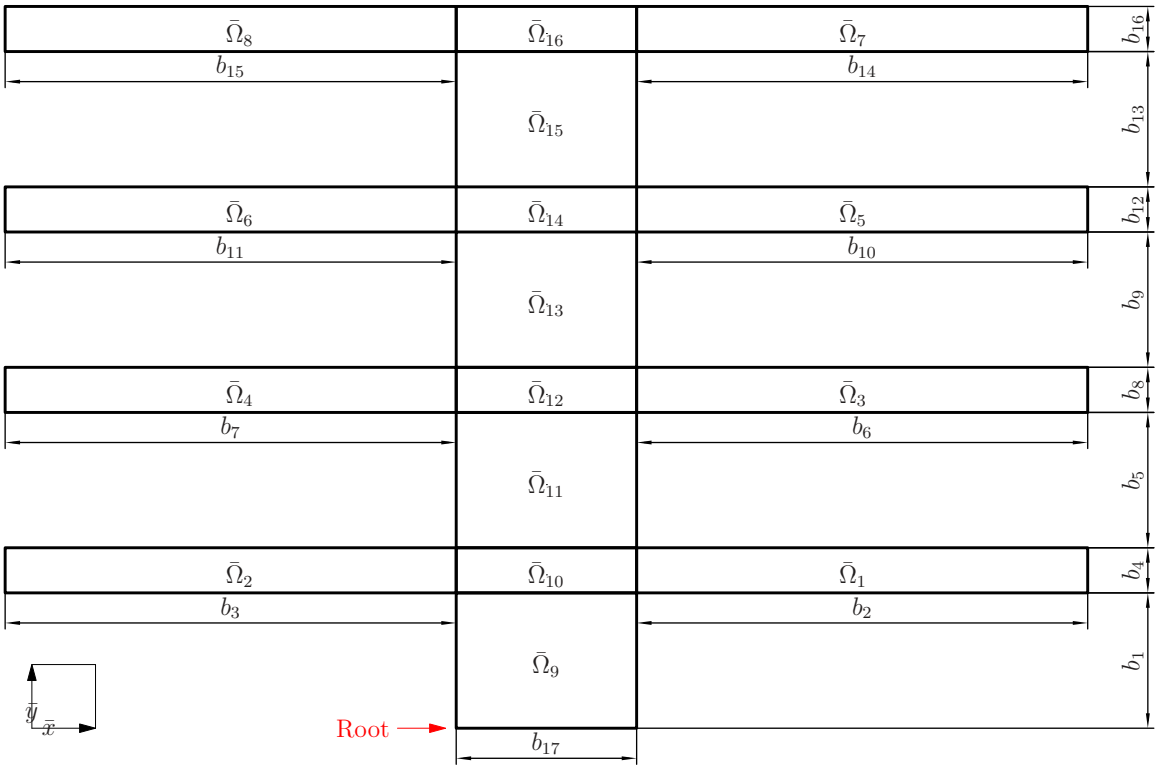


Figure 4-1: The thermal fin geometry in the physical domain.

The steady-state temperature distribution within the fin, $w(\mathbf{z})$, is governed by the following elliptic PDE

$$-\kappa_i \nabla^2 w^i = 0 \text{ in } \bar{\Omega}_i, \quad i = 1, \ldots, 16, \tag{4.1}$$

where w^i denotes the restriction of w to $\bar{\Omega}_i$, $\nabla = \frac{\partial}{\partial \bar{x}}\hat{\mathbf{i}} + \frac{\partial}{\partial \bar{y}}\hat{\mathbf{j}}$, and $\nabla^2 = \frac{\partial^2}{\partial \bar{x}^2} + \frac{\partial^2}{\partial \bar{y}^2}$,

where $\hat{\mathbf{i}}$ and $\hat{\mathbf{j}}$ are the unit vectors pointing along the $\bar{x}-$ and $\bar{y}-$direction. The continuity of temperature and heat flux at the conductivity-discontinuity interfaces $\bar{\Gamma}_{ij}^{\text{int}} = \partial\bar{\Omega}_i \cap \partial\bar{\Omega}_j$ for two adjacent regions $\bar{\Omega}_i$ and $\bar{\Omega}_j$, where $\partial\bar{\Omega}_i$ and $\partial\bar{\Omega}_j$ denote the boundary of $\bar{\Omega}_i$ and $\bar{\Omega}_j$ respectively, are ensured by the following interface condition

$$\left.\begin{array}{rcl} w^i & = & w^j \\ -\kappa_i(\nabla w^i \cdot \hat{\mathbf{n}}^i) & = & -\kappa_j(\nabla w^j \cdot \hat{\mathbf{n}}^j) \end{array}\right\} \text{ on } \bar{\Gamma}_{ij}^{\text{int}}, \tag{4.2}$$

where $\hat{\mathbf{n}}^i$ and $\hat{\mathbf{n}}^j$ denote the outward normal of $\bar{\Omega}_i$ and $\bar{\Omega}_j$ on the interface $\bar{\Gamma}_{ij}^{\text{int}}$, respectively. In order to model the convective heat losses on the external surface of a region $\bar{\Omega}_i$, i.e. $\bar{\Gamma}_i^{\text{ext}} = \partial\bar{\Omega}_i \setminus \bar{\Gamma}_{ij}^{\text{int}}$ and $\bar{\Gamma}_i^{\text{ext}} \neq \emptyset$, we use the following Robin boundary condition

$$-\kappa_i(\nabla w^i \cdot \hat{\mathbf{n}}^i) = \text{Bi } w^i \text{ on } \bar{\Gamma}_i^{\text{ext}} \text{ if } \bar{\Gamma}_i^{\text{ext}} \neq \emptyset,\ i = 1, \ldots, 16. \tag{4.3}$$

Finally, to model the heat source at the root, the Neumann flux boundary condition is imposed as

$$-\kappa_9(\nabla w^9 \cdot \hat{\mathbf{n}}^9) = -1 \text{ on } \bar{\Gamma}^{\text{root}}. \tag{4.4}$$

For this particular problem, the output of interest is the average temperature over the whole thermal fin

$$y = \frac{\sum_{i=1}^{16} \int_{\bar{\Omega}_i} w \; d\bar{\Omega}_i}{\sum_{i=1}^{16} \int_{\bar{\Omega}_i} 1 \; d\bar{\Omega}_i}. \tag{4.5}$$

Following Ref. [116], it can be showed that the temperature distribution solution w belongs to the Hilbert space $H^1(\bar{\Omega})$, where $\bar{\Omega} = \cup_{i=1}^{16} \bar{\Omega}_i$, and satisfies the following weak form

$$a(w, v) = \ell(v),\ \forall v \in H^1(\bar{\Omega}), \tag{4.6}$$

where the bilinear form a is given as

$$a(w, v) = \sum_{i=1}^{16} \kappa_i \int_{\bar{\Omega}_i} \nabla w \cdot \nabla v \; d\bar{\Omega}_i + \text{Bi} \sum_{i=1}^{16} \int_{\bar{\Gamma}_i^{\text{ext}}} wv \; d\bar{\Gamma}_i^{\text{ext}}, \tag{4.7}$$

and the linear form ℓ as

$$\ell(v) = \int_{\bar{\Gamma}^{\text{root}}} v \; d\bar{\Gamma}^{\text{root}} \tag{4.8}$$

4.2 Computational Domain Formulation

As a common practice, it is useful to transform the physical domain to a computational (or reference) domain to carry out the numerical calculation. A clear advantage is that one only needs to mesh the computational domain once, and the computational mesh can be used to solve the problem with any set of thermal conductivities, lengths and Biot number. The computational domain is chosen as in Figure 4-2 in which the dimensions of all computational regions are also presented.

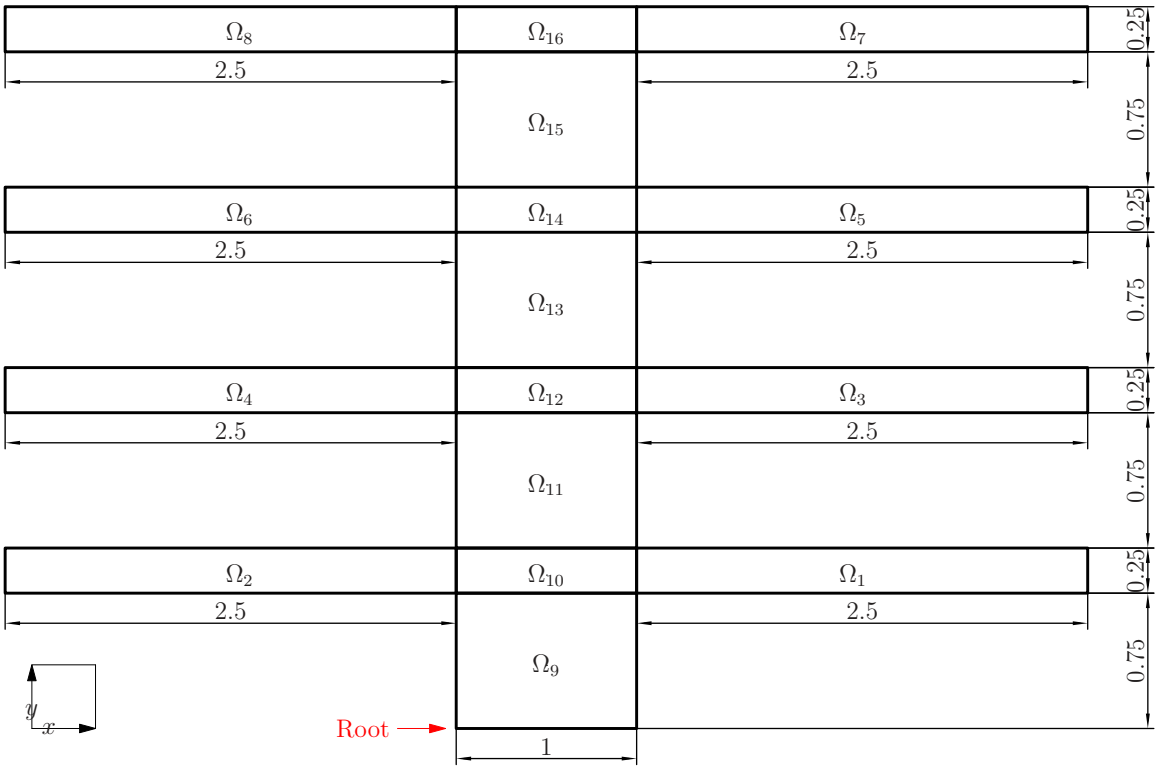


Figure 4-2: The thermal fin geometry in the computational domain.

Using a few simple linear transformation rules from $\bar{x}\bar{y}-$coordinates to $xy-$coordinates,

one can write the weak form (4.6) as

$$a(w,v) = \ell(v),\ \forall v \in H^1(\Omega),\ \Omega = \cup_{i=1}^{16}\Omega_i, \tag{4.9}$$

where the linear form ℓ is now given as

$$\ell(v) = b_{17}\int_{\Gamma^{\text{root}}} v\ d\Gamma^{\text{root}}, \tag{4.10}$$

and the bilinear form a as

$$a(w,v) = \sum_{i=1}^{16}\left\{\Theta_{2i-1}\int_{\Omega_i}\frac{\partial w}{\partial x}\frac{\partial v}{\partial x}\ d\Omega_i + \Theta_{2i}\int_{\Omega_i}\frac{\partial w}{\partial y}\frac{\partial v}{\partial y}\ d\Omega_i\right\} + \sum_{j=1}^{16}\Theta_{j+32}\int_{\Gamma_{b_j}^{\text{ext}}} wv\ d\Gamma_{b_j}^{\text{ext}}, \tag{4.11}$$

where $\Gamma_{b_j}^{\text{ext}}$ denotes the external boundary corresponding to b_j, and Θ_i are given as

$$\begin{aligned}
&\Theta_{4j-3} = 10\tfrac{\kappa_{2j-1}\ b_{4j}}{b_{4j-2}}, && \Theta_{4j-2} = 0.1\tfrac{\kappa_{2j-1}\ b_{4j-2}}{b_{4j}},\\
&\Theta_{4j-1} = 10\tfrac{\kappa_{2j}\ b_{4j}}{b_{4j-1}}, && \Theta_{4j} = 0.1\tfrac{\kappa_{2j}\ b_{4j-1}}{b_{4j}},\\
&\Theta_{13+4j} = \tfrac{\kappa_{7+2j}\ b_{4j-3}}{0.75 b_{17}}, && \Theta_{14+4j} = \tfrac{\kappa_{7+2j}\ b_{17}}{0.25 b_{4j-3}},\\
&\Theta_{15+4j} = 0.25\tfrac{\kappa_{8+2j}\ b_{4j}}{b_{17}}, && \Theta_{29+4j} = \tfrac{\text{Bi}\ b_{4j-3}}{0.75},\\
&\Theta_{30+4j} = \tfrac{\text{Bi}\ b_{4j-2}}{2.5}, && \Theta_{31+4j} = \tfrac{\text{Bi}\ b_{4j-1}}{2.5},\\
&\Theta_{32+4j} = \tfrac{\text{Bi}\ b_{4j}}{0.25}, && j = 1,\ldots,4.
\end{aligned}$$

The output of interest is now given as

$$y = \frac{\sum_{i=1}^{16}\beta_i\int_{\Omega_i} w\ d\Omega_i}{\sum_{i=1}^{16}\beta_i\int_{\Omega_i} 1\ d\Omega_i}, \tag{4.12}$$

where β_i are given by

$$\begin{aligned}
&\beta_{2j-1} = \tfrac{b_{4j-2}\ b_{4j}}{0.625}, && \beta_{2j} = \tfrac{b_{4j-1}\ b_{4j}}{0.625},\\
&\beta_{7+2j} = \tfrac{b_{17}\ b_{4j-3}}{0.75}, && \beta_{8+2j} = \tfrac{b_{17}\ b_{4j}}{0.25},\\
&j = 1,\ldots,4
\end{aligned}$$

Next using the finite element method with triangular linear elements, for example,

the weak form (4.9) can be written in matrix form as

$$\mathbf{A}(\mathbf{z})\mathbf{x} = \mathbf{B}(\mathbf{z}), \quad y = \mathbf{C}(\mathbf{z})\mathbf{x}, \tag{4.13}$$

where the vector $\mathbf{x}$ consists of all nodal temperature values. The matrices $\mathbf{A}(\mathbf{z})$, $\mathbf{B}(\mathbf{z})$, and $\mathbf{C}(\mathbf{z})$ are as follows

$$\mathbf{A}(\mathbf{z}) = \sum_{i=1}^{48} \Theta_i(\mathbf{z})\mathbf{A}_{q_i}, \tag{4.14}$$

$$\mathbf{B}(\mathbf{z}) = z_{34}\mathbf{B}_q, \tag{4.15}$$

$$\mathbf{C}(\mathbf{z}) = \sum_{i=1}^{16} \Lambda_i(\mathbf{z})\mathbf{C}_{q_i}, \tag{4.16}$$

where $\mathbf{A}_{q_i}$, $\mathbf{B}_q$ and $\mathbf{C}_{q_i}$ are the appropriate finite element matrices that do not depend on the parameters $\mathbf{z}$, and Λ_i are given as

$$\Lambda_i(\mathbf{z}) = \frac{\beta_i(\mathbf{z})}{\sum_{j=1}^{16} \beta_j(\mathbf{z}) \int_{\Omega_i} 1 \; d\Omega_i}. \tag{4.17}$$

It should be pointed out that if one chooses the average temperature at the root as the output of interest, the output matrix $\mathbf{C}$ is then exactly $\mathbf{B}_q$ which does not depend on the parameters $\mathbf{z}$.

The matrices in (4.14)–(4.16) admit an affine decomposition in which the affine coefficients, Θ_i and Λ_i, are nonlinear functions of the parameters $\mathbf{z}$. This affine decomposition together with the projection-based model reduction technique enables us to obtain efficient reduced models, i.e. models for which solution cost is independent of the full model size.

The design problem of interest here is to find the best materials and geometry lengths combination to maximize the cooling efficiency. Note that the output is directly related to the cooling efficiency, that is, the smaller the output, the better the cooling efficiency. Therefore the design problem becomes finding the best set of parameters $\mathbf{z}$ to minimize the output y defined in (4.12).

4.3 Numerical Results

In this chapter, the ranges (bound constraints) of the parameters are as follows

$$0.1 \leq \quad \kappa_j \quad \leq 10, \quad j = 1, \ldots, 16, \tag{4.18}$$

$$0.01 \leq \quad \mathrm{Bi} \quad \leq 5, \tag{4.19}$$

$$0.1 \leq \quad b_j \quad , \quad j = 1, \ldots, 17, \tag{4.20}$$

$$b_{4j-3} \leq 5, \quad j = 1, \ldots, 4, \tag{4.21}$$

$$b_{4j-2} \leq 10, \quad j = 1, \ldots, 4, \tag{4.22}$$

$$b_{4j-1} \leq 10, \quad j = 1, \ldots, 4, \tag{4.23}$$

$$b_{4j} \leq 2.5, \quad j = 1, \ldots, 4, \tag{4.24}$$

$$0.1 \leq \quad b_{17} \quad \leq 5. \tag{4.25}$$

It should be pointed out that the parameter range is wide in the sense that the upper bound for a parameter is two orders of magnitude larger than the corresponding lower bound. Thus, this presents a challenge for model reduction methods in creating a reduced model of moderate size that accurately captures the full model behavior over such a wide range of parameter values in multidimensional parametric input space. We also introduce the baseline parameter set as

$$\kappa_i^{\text{baseline}} = 0.4, \quad i = 1, 2, \tag{4.26}$$

$$\kappa_i^{\text{baseline}} = 0.6, \quad i = 3, 4, \tag{4.27}$$

$$\kappa_i^{\text{baseline}} = 0.8, \quad i = 5, 6, \tag{4.28}$$

$$\kappa_i^{\text{baseline}} = 1.2, \quad i = 7, 8, \tag{4.29}$$

$$\kappa_i^{\text{baseline}} = 1.0, \quad i = 9, \ldots, 16, \tag{4.30}$$

$$\mathrm{Bi}^{\text{baseline}} = 0.1, \tag{4.31}$$

$$b_{4i-3}^{\text{baseline}} = 0.75, \quad i = 1, \ldots, 4, \tag{4.32}$$

$$b_{4i-2}^{\text{baseline}} = 2.5, \quad i = 1, \ldots, 4, \tag{4.33}$$

$$b_{4i-1}^{\text{baseline}} = 2.5, \quad i = 1, \ldots, 4, \tag{4.34}$$

$$b_{4i}^{\text{baseline}} = 0.25, \quad i = 1, \ldots, 4, \tag{4.35}$$

$$b_{17}^{\text{baseline}} = 1.0, \tag{4.36}$$

so that if any parameter is not allowed to vary, it will take its corresponding baseline value.

For all problems considered in the following, since it is too computationally expensive to cover multidimensional parameter spaces with full factorial search, we limit ourselves to define the maximum output error to be the maximum error between the full and the reduced model outputs over a random set of 10^5 parameters in the parameter space under consideration. For the model-constrained adaptive sampling method, we choose ϵ in Step 2 of Algorithm 3.1 to be $\epsilon = 1.0e - 14$.

Initial guesses for the model-constrained adaptive sampling method are obtained from logarithmic random sampling. The smallest allowable distance between an initial guess and all existing sample locations is chosen to be $\min_i\{z_{max}^i - z_{min}^i\}$. Unless otherwise specified, the reduced basis is computed by performing Gram-Schmidt orthogonalization on the snapshots. Finally, the computation time is measured on a dual core 64-bit personal computer with 3.2GHz Pentium processor.

4.3.1 Error Indicator Using the Euclidean *2-norm* versus the Hilbert Dual Norm

As discussed in Section 3.1.2, one can use the residual norm as the error indicator in the greedy optimization problem. Intuitively, the adaptive sampling procedure will drive the residual to zero as more greedy cycles are taken. For steady problems that are linear in the state vector, using the residual in the *2-norm*, for example, one can show that

$$\|\mathbf{y} - \mathbf{y_r}\|_2 \leq \|\mathbf{C}\|_2 \|\mathbf{A}^{-1}\|_2 \|\mathbf{R}\|_2, \tag{4.37}$$

and therefore the true output error is also driven down to zero as the residual norm approaches zero. The question is now which norm to use for the residual. The two

obvious choices are the residual in the Euclidean *2-norm*

$$\|\mathbf{R}\|_2^2 = \mathbf{R}^T\mathbf{R}, \tag{4.38}$$

and in the dual norm [61] (discrete $H^{-1}-norm$)

$$\|\mathbf{R}\|_D^2 = \mathbf{R}^T\bar{\mathbf{A}}^{-1}\mathbf{R} \tag{4.39}$$

where $\bar{\mathbf{A}}$ is a SPD matrix that is given by, e.g. using the finite element method,

$$\bar{A}_{ij} = \int_\Omega \nabla\varphi_i \cdot \nabla\varphi_j \; d\Omega + \sum_{k=1}^{16} \int_{\Gamma_{b_j}^{\text{ext}}} \varphi_i\varphi_j \; d\Gamma_{b_j}^{\text{ext}}, \tag{4.40}$$

where φ_i denotes the usual finite element hat functions. The theoretical problem with the Euclidean *2-norm* is that the residual using this norm is a sum of an increasing number of terms as the grid (or mesh) is refined [116], and hence theoretically it will converge to infinity as the grid size approaches zero. The advantage of the Euclidean *2-norm* is that it is simple and may be the only choice for problems that are not from a discretization of some set of PDEs, such as a molecular dynamic simulation problem. The residual in the dual norm, on the other hand, is more expensive to use since it involves the inversion of $\bar{\mathbf{A}}$, but as the grid size approaches zero it will not approach infinity due to the presence of $\bar{\mathbf{A}}^{-1}$ as the stabilization.

In practice, the grid size is finite and we would like to investigate the impact of these norms on the quality of the reduced model using the adaptive sampling approach. In particular, we consider using these norms for the residual in the cost functional in (3.10) to find a reduced basis, and hence a reduced model. We consider a test case of the thermal fin problem with five parameters, namely the thermal conductivities of the four subfins and the Biot number, i.e.

$$\mathbf{z} = \{\kappa_1, \kappa_2, \kappa_3, \kappa_4, \text{Biot}\}. \tag{4.41}$$

Figure 4-3 shows a coarse grid with 1333 nodes, a medium grid with 4760 nodes

and a fine grid with 17899 nodes for the thermal fin. Recall that a finite element model with linear triangular elements is used for the thermal fin problem, and hence the number of nodes is the same as the number of unknowns (because there are no Dirichlet boundary conditions in this problem).

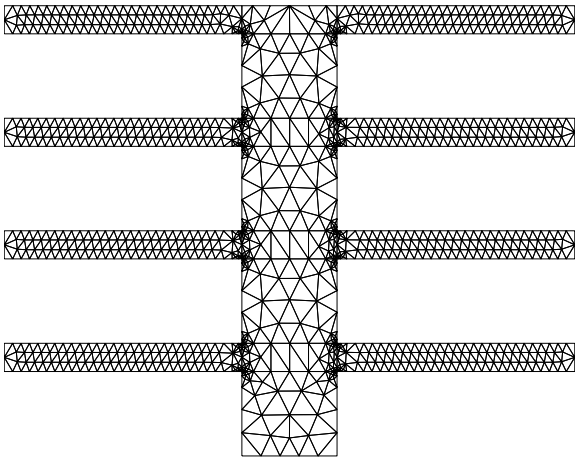
Next, we run the adaptive sampling procedure with 25 greedy cycles using two different greedy optimization problems that have the same form as (3.10)–(3.13) but with the residual in the Euclidean *2-norm* and the Hilbert dual norm as the cost functional. We have used the same sequence of initial guesses with 25 parameter points for the two optimization problems. In Table 4.1 the maximum output error is used to compare the quality of the resulting reduced models when the grid is refined. As can be seen, using the adaptive sampling procedure for the thermal fin problem, the difference between using the residual in the *2-norm* and in the dual norm is small. In fact, for a finite grid size, these norms are equivalent in the sense

$$\alpha_{min}\|\mathbf{R}\|_2^2 \leq \|\mathbf{R}\|_D^2 \leq \alpha_{max}\|\mathbf{R}\|_2^2, \tag{4.42}$$

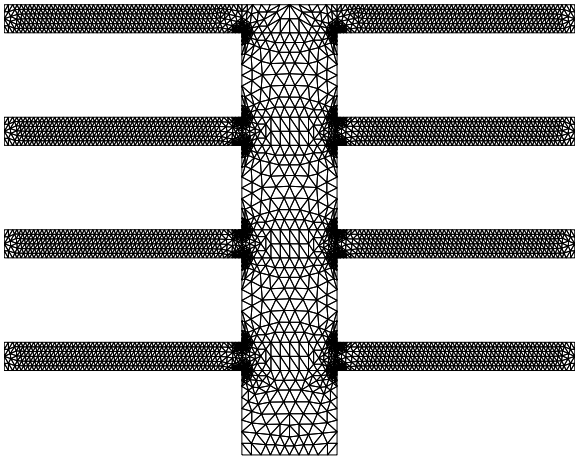
where α_{min} and α_{max} are the maximum and the minimum eigenvalues of the matrix $\bar{\mathbf{A}}^{-1}$. The similarity in performance is further confirmed in Figure 4-4 in which we plot the maximum output error versus the number of reduced basis vectors. Again, the difference is so small that, for a finite grid size, either the residual in the *2-norm* or in the dual norm can be used as the error indicator in the adaptive sampling procedure.

Table 4.1: Adaptive sampling procedure for the thermal fin problem with five parameters using the residual in the *2-norm* and in the dual norm as the cost functional. The maximum output error is shown for all three grids.

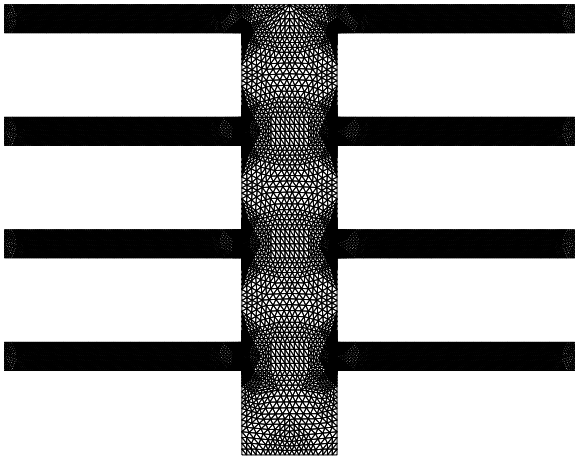
Grids	$\max \|\mathbf{y} - \mathbf{y_r}\|_2^2$ $(\mathcal{G} = \|\mathbf{R}\|_2^2)$	$\max \|\mathbf{y} - \mathbf{y_r}\|_2^2$ $(\mathcal{G} = \|\mathbf{R}\|_D^2)$
Coarse	3.5830e-06	1.5867e-05
Medium	8.9126e-06	1.5575e-05
Fine	7.5282e-06	4.7335e-06



(a) Coarse grid



(b) Medium grid



(c) Fine grid

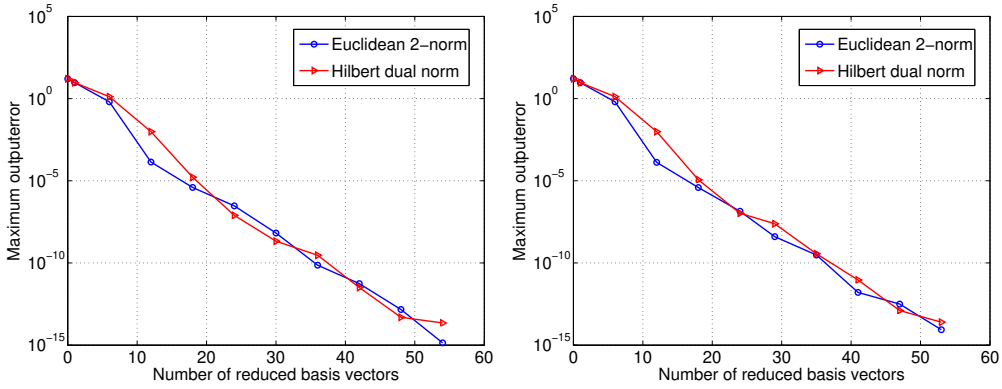
Figure 4-3: The thermal fin with a coarse grid with 1333 nodes, a medium grid with 4760 nodes and a fine grid with 17899 nodes.

From now on to the end of this chapter, the fine grid is used for all problems and only the residual in the *2-norm* is used as the error indicator.

4.3.2 Performance of the Bound-Constrained Optimization Solver

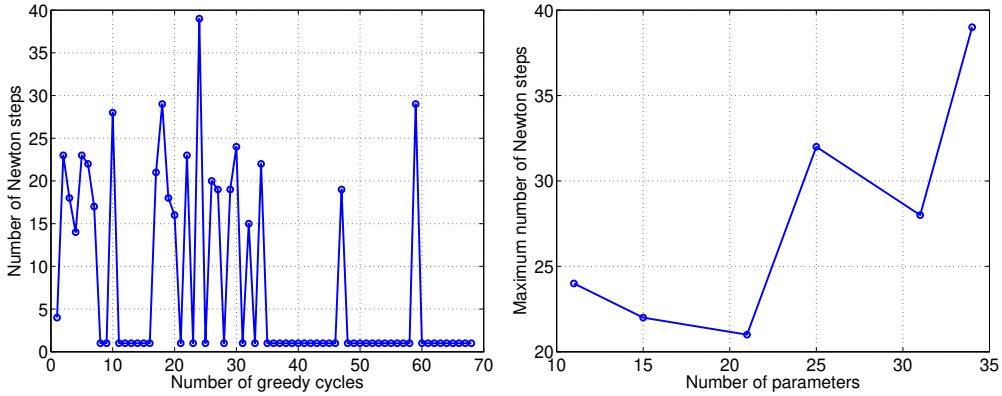
In this section, we investigate the efficiency of the optimization solver for the greedy optimization problem discussed in Section 3.1.3. In Figure 4-5(a), we show the number of Newton steps versus the number of greedy cycles for the case of 34 parameters. As can be seen, the maximum number of Newton steps is 40, which occurs at the 24^{th} greedy cycle. For the other adaptive cycles, the number of Newton steps is less than 30. Figure 4-5(b) shows the maximum number of Newton steps over $2d$ greedy cycles for cases with a varying number of parameters d. It can be seen that for these cases, the number of Newton steps is $\mathcal{O}(d)$.

Recall that, due to the combination of the trust region strategy and the inexact Newton-CG method, quadratic convergence is observed for all greedy cycles that satisfy the quadratic convergence condition (i.e. when the iterate is close enough to a local maximizer). Typical quadratic convergence is shown in Table 4.2 in which the cost functional and the scaled gradient are shown along with the number of Newton



(a) Medium Grid (b) Fine Grid

Figure 4-4: Maximum reduced model output error using the residual in the *2-norm* versus in the dual norm for the thermal problem with five parameters.



(a) Number of Newton steps versus number of greedy cycles for 34-parameter case

(b) Maximum number of Newton steps versus number of parameters

Figure 4-5: Performance of the bound-constrained optimization solver: a) number of Newton steps versus number of greedy cycles for 34-parameter case; b) maximum number of Newton steps versus number of parameters

steps. When the iterate is far way from the local maximizer, the trust region method picks the steepest ascent direction and the step is allowed to be as large as possible. This happens for the first 16 Newton steps. Once the iterate is close to the local maximizer, the Newton direction is picked and quadratic convergence, e.g. clearly for the scaled gradient in Table 4.2, is observed for the last three Newton steps. The fast convergence can also be seen in the cost functional $\mathcal{G}$ which increases by 9 orders of magnitude from the initial guess to the local maximizer within 18 Newton steps.

Table 4.2: Typical quadratic convergence in the scaled gradient $\|\mathbf{D}\nabla\mathcal{G}\|_2$ from the bound constrained optimization solver for the case with 34 parameters. The data are shown for the third greedy cycle.

Number of Newton steps	$\mathcal{G}$	$\|\mathbf{D}\nabla\mathcal{G}\|_2$
1	3.5306e-07	1.1503e-05
⋮	⋮	⋮
13	3.4580e+02	2.5811e-01
14	3.4598e+02	6.0533e-02
15	3.4603e+02	6.2677e-02
16	3.4603e+02	1.5441e-01
17	3.4603e+02	7.5716e-04
18	3.4603e+02	2.0260e-10

4.3.3 Output Error Indicator versus True Output Error

In this section, we will compare the offline cost as well as the quality of the reduced model using the model-constrained adaptive sampling approach in Algorithm 3.1 with an error indicator and the true output error. For the error indicator, since we only compute the full solution snapshots at the optimal parameter points found from the greedy optimization problem, only these snapshots are used to enrich the reduced basis. For the true error, we also take only the snapshots at the optimal parameter points to enrich the reduced basis. For both approaches, the same sequence of initial guesses is used in each greedy optimization cycle, and Gram-Schmidt orthogonalization is used to update the reduced basis.

The examples considered here have 11 and 21 parameters, that is,

$$\mathbf{z} = \{\kappa_1, \ldots, \kappa_5, \mathrm{Bi}, b_1, \ldots, b_5\}, \tag{4.43}$$

or

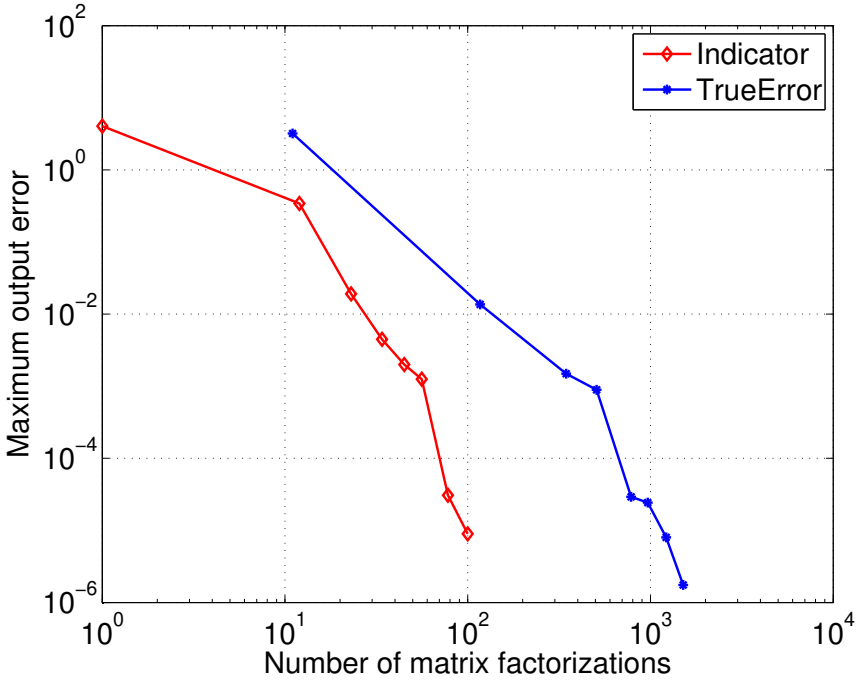
$$\mathbf{z} = \{\kappa_1, \ldots, \kappa_{10}, \mathrm{Bi}, b_1, \ldots, b_{10}\}. \tag{4.44}$$

We use the number of full matrix factorizations as the measure for the offline cost to compare the quality of the reduced models, since this is the dominant cost of the reduction algorithm. For the 11-parameter case, Figure 4-6(a) shows that the required number of matrix factorizations to reach a given error level is approximately an order of magnitude larger for the true-error approach; however, for the same number of basis functions retained in the reduced basis, Figure 4-6(b) shows that using the true error rather than the indicator leads to more efficient (i.e. smaller for a given error level) reduced models. This result might be intuitively expected, since the optimal parameter points based on the true output error should better target reduction of the true output error than those points selected using the error indicator. However, there is no guarantee that this will always be the case, as shown in Figure 4-7 for the case of 21 parameters. Figure 4-7(a) shows that the number of matrix factorizations is again about an order of magnitude larger for the true-error approach. For smaller

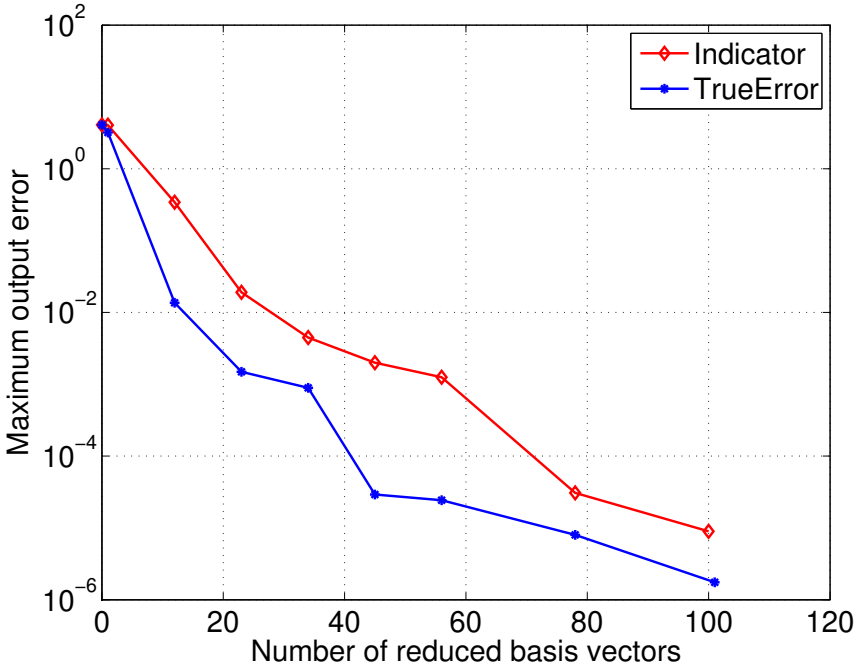
basis sizes, Figure 4-7(b) shows that the output error is again smaller than for models obtained using the error indicator; however, for larger basis sizes, the true-error and error-indicator approaches give equivalently good reduced models.

Figures 4-6 and 4-7 demonstrate a general tradeoff in the model-constrained sampling methodology: if one is willing to invest larger offline cost to compute the reduced model, then using the true error to select the parameter sample points can lead to more efficient models. For some problems, such as real-time applications, minimizing the size of the reduced model may be critical; in that case, one might choose to use the true-error approach. For very large-scale problems, however, the cost of the true-error approach may be prohibitively high; in that case, the error indicator is an attractive option. In many of the numerical experiments performed for the thermal fin problem, and as demonstrated by the results in Figure 4-7, the difference in quality between the true-error and error-indicator sample points tends to be larger in early greedy cycles. Since the error function becomes more multimodal as the number of greedy cycles increases, the chances of sampling at a local maximum are increased, and thus the difference between the error-indicator and true-error approaches may not be as great. One could therefore also conceive of using a combination of the two error metrics, i.e. using the true error for early greedy cycles and the error indicator for later cycles, in an attempt to balance offline cost with the quality of the reduced model.

As discussed in Section 3.1.6, when using the true output error, one has intermediate full state snapshots and full adjoint snapshots available at all Newton steps, i.e. the available snapshot information includes but is not limited to those solutions at the optimal parameter points. We compare four methods of updating the reduced basis, described in Section 3.1.6 and summarized in Table 4.3. In the first method, we add all the snapshots of the current greedy cycle to the current reduced basis using the Gram-Schmidt procedure (i.e. adding only information that is linearly independent of the current reduced basis vectors). In the second method, we store all the snapshots, which is possible for the thermal fin problem for a small number of greedy cycles, and then perform the POD method on the complete snapshot set (i.e. that comprising

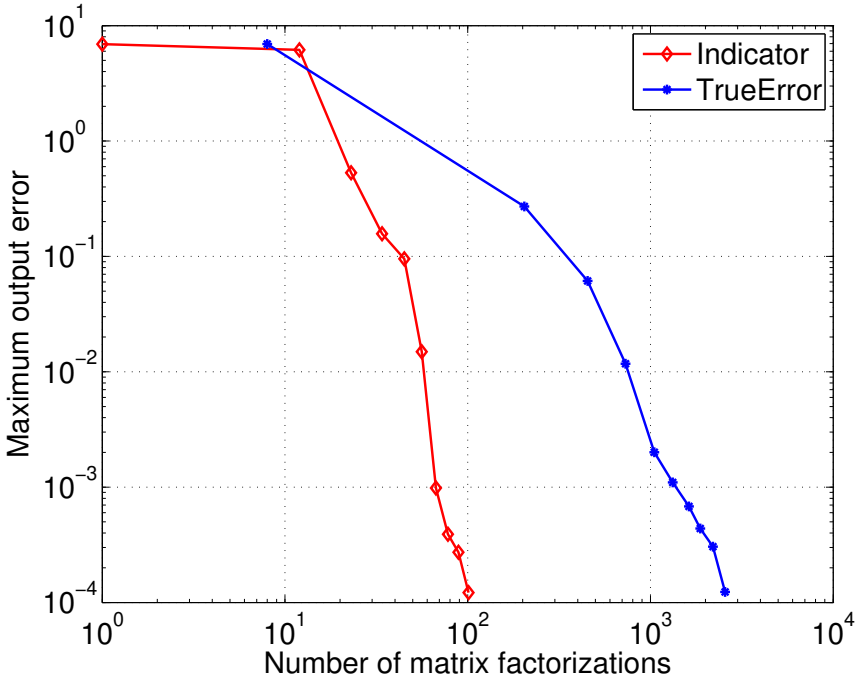


(a) Maximum error versus the number of full matrix factorizations

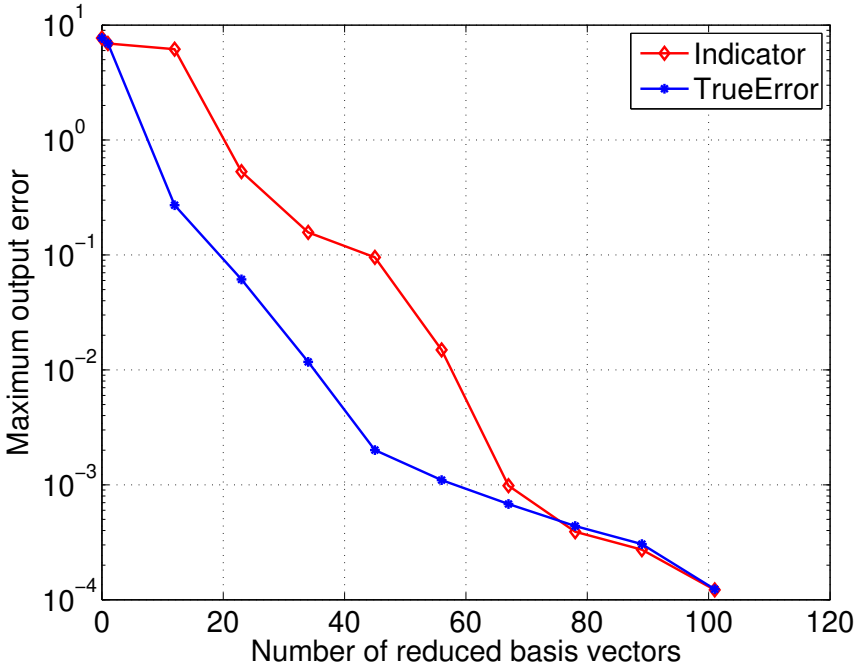


(b) Maximum error versus the number of reduced basis vectors

Figure 4-6: Error indicator versus true output error for the thermal fin with 11 parameters. The same sequence of initial guesses is used for both true-error and error-indicator approaches, and the Gram-Schmidt procedure is used to update the reduced basis.



(a) Maximum error versus the number of full matrix factorizations



(b) Maximum error versus the number of reduced basis vectors

Figure 4-7: Error indicator versus true output error for the thermal fin with 21 parameters. The same sequence of initial guesses are used for both true-error and error-indicator approaches, and the Gram-Schmidt procedure is used to update the reduced basis.

the newly computed snapshots of the current greedy cycle and all the snapshots from the previous greedy cycles). Denote χ_i to be the singular value corresponding to the i^{th} POD basis vector, which is added to the reduced basis if the following condition is satisfied [40, 51]

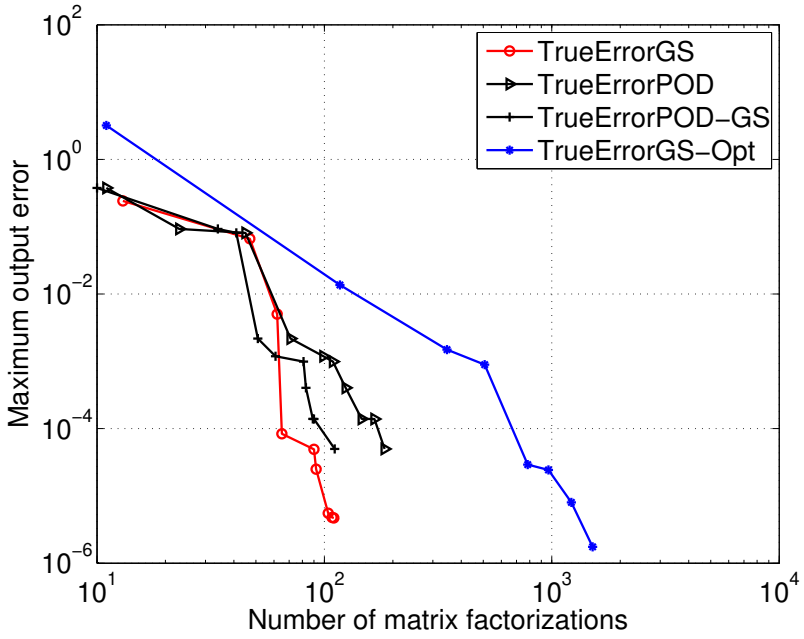
$$\frac{\sum_{j=1}^{i} \chi_j}{\sum_{j=1}^{n_s} \chi_j} \leq \eta, \tag{4.45}$$

where n_s is the number of snapshots, and $\eta \leq 1$. We choose $\eta = 0.99999999$ for the second method. The third method is to perform the POD method on the newly computed snapshots of the current greedy cycle with $\eta = 0.99999999$, and then add only the dominant POD basis vectors to the current reduced basis using the Gram-Schmidt procedure. The fourth method considered is the base case employed in Figures 4-6 and 4-7, i.e. using the Gram-Schmidt procedure only on the snapshots at the optimal parameter points.

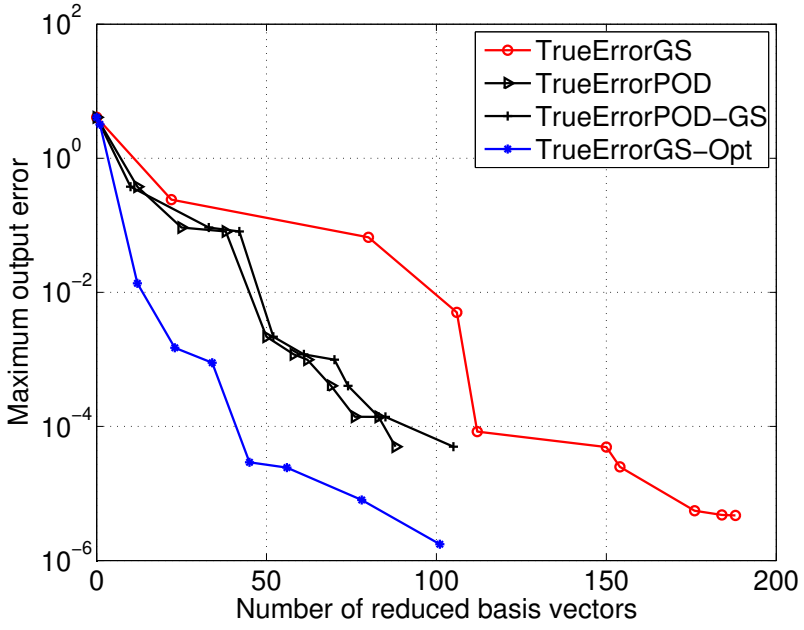
Table 4.3: Four methods of updating the reduced basis when using the true output error.

Method	Intermediate information included	Basis updating
TrueErrorGS	Yes	Gram-Schmidt
TrueErrorPOD	Yes	POD
TrueErrorPOD-GS	Yes	POD and then Gram-Schmidt
TrueErrorGS-Opt	No	Gram-Schmidt

For these four methods, the reduced bases at each greedy cycle are not the same, and hence the snapshots found in each cycle are different. Figures 4-8(a) and 4-9(a) show that, with a same number of matrix factorizations, the first three methods yield reduced models of equivalent accuracy. However, as can be seen in Figures 4-8(b) and 4-9(b), with the same number of reduced basis vectors, the first method leads to reduced models that are less accurate. That is, adding all new snapshots leads to reduced models that are inefficient and unnecessarily large. The comparison with the base approach shows that discarding the intermediate information leads to higher offline cost, but significantly smaller reduced models for a given level of output error.

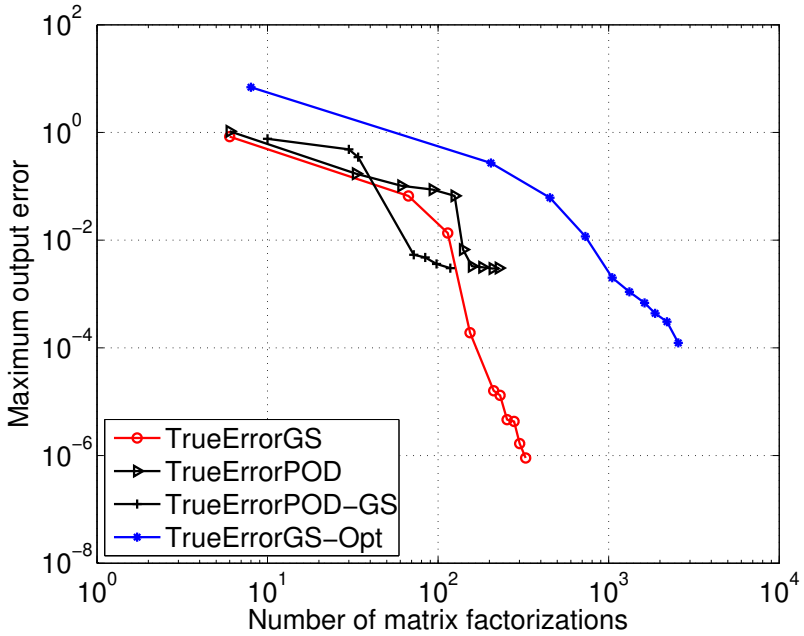


(a) Maximum error versus the number of full matrix factorizations

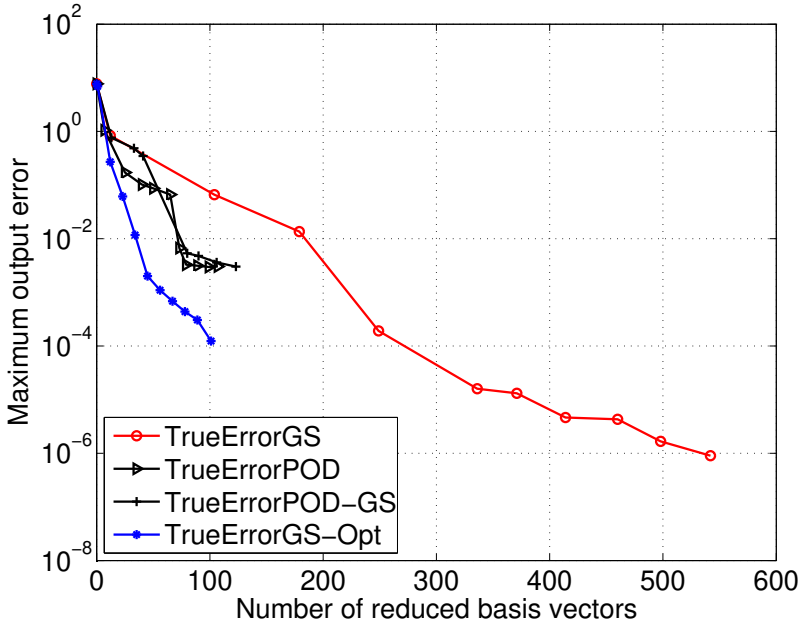


(b) Maximum error versus the number of reduced basis vectors

Figure 4-8: Different methods of updating the reduced basis for the case with 11 parameters. The comparisons are the true output error using the Gram-Schmidt procedure for all snapshots (TrueErrorGS), the true output error with the POD method for all snapshots after each greedy cycle (TrueErrorPOD), the true output error with the Gram-Schmidt procedure on the POD vectors of the snapshots of the current greedy cycle and on the current reduced basis (TrueErrorPOD-GS), and the true output error with the Gram-Schmidt procedure only on the optimal parameter points (TrueErrorGS-Opt).



(a) Maximum error versus the number of full matrix factorizations



(b) Maximum error versus the number of reduced basis vectors

Figure 4-9: Different methods of updating the reduced basis for the case with 21 parameters. The comparisons are the true output error using the Gram-Schmidt procedure for all snapshots (TrueErrorGS), the true output error with the POD method for all snapshots after each greedy cycle (TrueErrorPOD), the true output error with the Gram-Schmidt procedure on the POD vectors of the snapshots of the current greedy cycle and on the current reduced basis (TrueErrorPOD-GS), and the true output error with the Gram-Schmidt procedure only on the optimal parameter points (TrueErrorGS-Opt).

In summary, either the error indicator or the true error can be used as the cost functional in the model-constrained adaptive sampling approach. For the thermal fin problem, the error-indicator approach leads to accurate reduced models with approximately one order of magnitude reduction in offline cost compared to using the true error. Unless it is critical to decrease the size of the reduced model as much as possible, using the error indicator is the recommended approach, especially for very large-scale systems where the additional offline computational cost of using the true error may be prohibitively high. Further, if using the true-error approach, although intermediate state and adjoint information is available and can be included in the basis updating process, the results indicate that doing so compromises the quality of the resulting reduced models.

In the next section, we compare the model-constrained adaptive sampling method with other sampling approaches. For all results that follow, the error-indicator approach is used.

4.3.4 Model-Constrained Adaptive Sampling versus Other Sampling Methods

In this section we compare the model-constrained adaptive sampling approach using the residual error indicator as the objective function with other sampling approaches. We first consider the greedy sampling method, which is the fundamental concept underlying the model-constrained approach. We then compare the model-constrained approach with four statistically-based sampling methods.

Model-Constrained Adaptive Sampling Approach versus the Greedy Sampling Approach

To apply the greedy sampling method in [61, 62, 68], one needs to determine a training parameter set with n_{train} parameters. At each point in this parameter set, the reduced states are computed by solving the reduced model. The error estimator, i.e. here the residual in the *2-norm*, is then computed. The location in the training pa-

rameter set at which the error estimator is maximal is then found, the reduced basis is updated with the snapshot at this sample location, and the updated reduced model is computed. To generate the training set for the greedy sampling approach, we use two different methods, namely logarithmic random sampling and Latin hypercube sampling (LHS).

For the model-constrained adaptive sampling approach, we use logarithmic random sampling to generate initial guesses for the greedy optimization problem. The residual in the *2-norm* is used as the error indicator; hence, only snapshots at the optimal parameter points are computed. A total of 100 greedy cycles are used for both model-constrained sampling and greedy sampling methods; hence, both methods will provide 100 sample points at which the snapshots are computed to form the reduced basis. This comparison set-up gives us the same offline cost corresponding to the full model solve, cost^F, in (3.25) for both methods. The only difference is the offline cost corresponding to the reduced model solve, cost^R. For the model-constrained approach, cost^R for steady problems is approximately given as

$$\text{cost}^R_{\text{model-constrained}} \sim d \sum_{j=1}^{n_G} \frac{2}{3} j^3 + [2(1 + n_{Hv}) + n_\Theta^2] j^2, \tag{4.46}$$

where, following the affine decomposition in (4.14), we have also incorporated the cost of forming the reduced matrix $\mathbf{A_r}$ as $n_\Theta^2 i^2$, where $n_\Theta = 48$ is the number of terms in (4.14). The offline cost corresponding to the reduced model solve for the greedy sampling approach (either with logarithmic random sampling or LHS) is given by

$$\text{cost}^R_{\text{greedy}} \sim n_{\text{train}} \sum_{j=1}^{n_G} \frac{2}{3} j^3 + (n_\Theta^2 + 1) j^2. \tag{4.47}$$

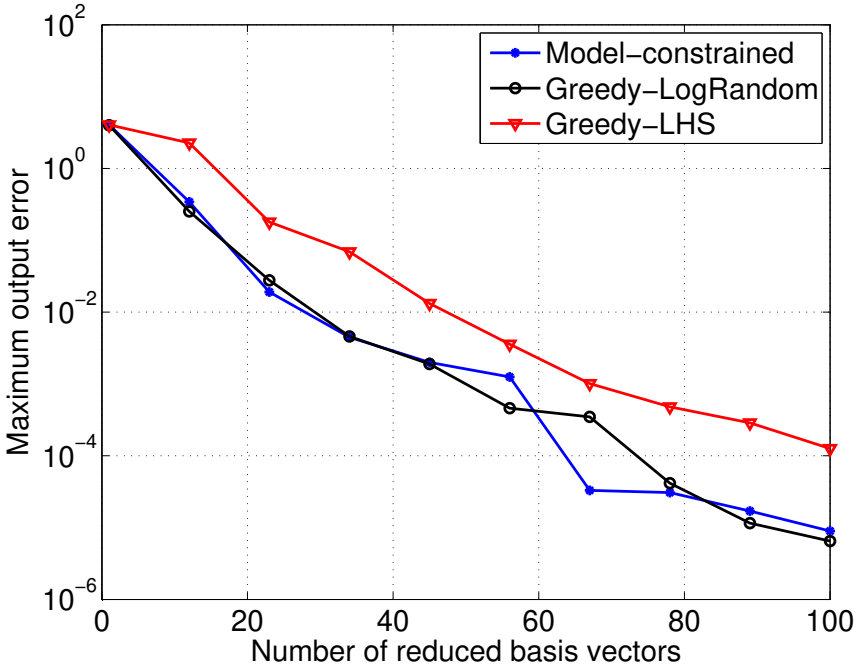
At each greedy cycle, the greedy sampling approach needs to solve n_{train} dense reduced models at n_{train} parameters to find the reduced states in order to compute the error estimator. The model-constrained approach needs to solve the greedy optimization problem in each greedy cycle. The difference in the cost in each greedy cycle is therefore the multiplicative constant d for the model-constrained approach and n_{train}

for greedy sampling methods. For the following comparisons, we choose $n_{\text{train}} = 10^4 \gg d \in \{11, 21\}$.

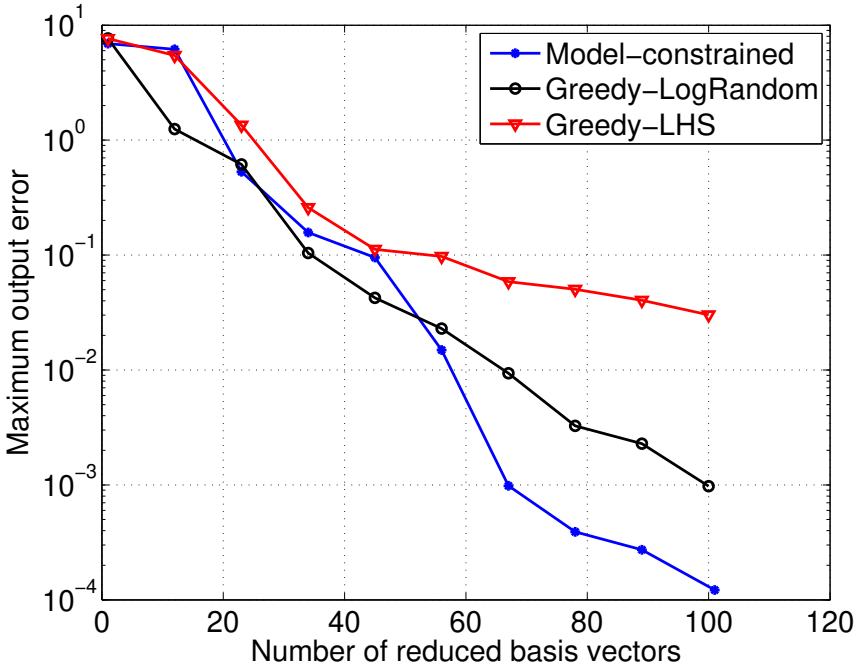
Figures 4-10(a) and 4-10(b) show the comparison for the three methods for the case of 11 and 21 parameters, respectively. It can be seen that the maximum output error obtained using the model-constrained approach is, for the most part, comparable to that of the greedy sampling method with logarithmic random training points. For the case with 21 parameters, the model-constrained method is able to achieve an order of magnitude better error than the greedy approach for larger reduced model sizes. Using the greedy sampling method with LHS training points led to larger errors than those obtained using the other two methods.

A key difference between the methods is that the greedy sampling approach finds globally-optimal parameter points within the discrete training set (via exhaustive search) while the model-constrained approach finds locally-optimal parameter points in the continuous parameter space (by solving an optimization problem). As a result, unless n_{train} is sufficiently large, the training set may not adequately cover important parameter regions, particularly as the dimension of the parametric space becomes large. This difference is highlighted by the results in Figure 4-10(b), where even 10^4 training points were not sufficient for the greedy sampling method to find near-optimal sample points in later greedy iterations.

Although the total number of large-scale matrix factorizations is 100 for both the model-constrained and greedy sampling methods, the actual offline cost differs substantially between the two. Table 4.4 compares the CPU time required to compute the reduced basis for the three approaches for the case of 21 parameters. It can be seen that the model-constrained approach is approximately 16 times faster than the greedy sampling approaches. This difference is due to the need for exhaustive search over 10^4 training points on every greedy iteration in the greedy sampling method, which for this case is the dominant offline cost. This could be a particular concern as the number of parameters, d, and hence the necessary number of training points, n_{train}, increases.



(a) 11-parameter case



(b) 21-parameter case

Figure 4-10: Model-constrained sampling approach versus greedy sampling approaches (Greedy-LogRandom: greedy sampling with logarithmic random training parameter set, and Greedy-LHS: greedy sampling with LHS training parameter set) over 100 greedy cycles. A training set with 10^4 training points is used in the greedy search for the greedy sampling approaches.

Table 4.4: The offline cost in CPU time of the model-constrained sampling approach and the greedy sampling approaches for the case of 21 parameters. 100 greedy cycles are taken for all methods and $n_{\text{train}} = 10^4$ for the greedy sampling approaches.

	Model-constrained sampling	Greedy sampling with logarithmic random	Greedy sampling with LHS
CPU time	0.58 hours	8.33 hours	8.33 hours

Model-Constrained Adaptive Sampling versus Other Sampling Methods

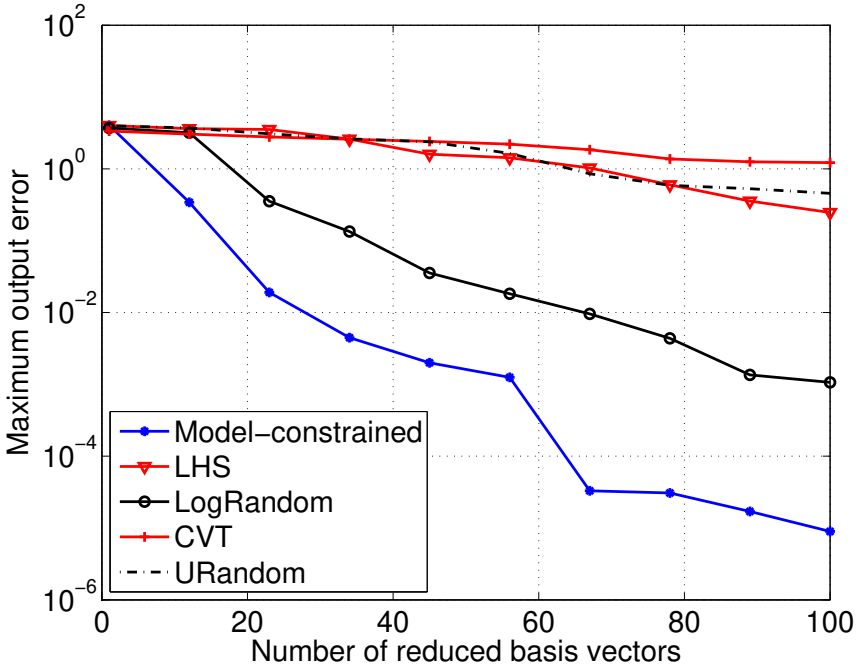
Next, we compare the model-constrained sampling method with statistically-based sampling methods in the context of snapshot generation for model reduction. In particular, we compare our model-constrained sampling with LHS sampling, uniform random sampling, logarithmic random sampling, and CVT sampling. For all methods, we take 100 sample points to generate 100 snapshots, which are then used to form the reduced basis. As can be seen in Figures 4-11(a) and (b) for 11 and 21 parameters, respectively, the model-constrained sampling method outperforms the other methods in the sense that, for a given basis size, the reduced model error is several orders of magnitude smaller than that obtained using the other methods. Furthermore, going from 11 to 21 parameters, the difference in accuracy of the reduced model using model-constrained sample points and those of other methods is larger. This reflects the fact that the model-constrained sampling is a model-based sampling method; that is, the parameter is sampled where the indicator of the error between the full and the reduced models is locally largest. The other methods, on the other hand, use no knowledge of the underlying model when selecting their sample points. As the dimension of the parametric space increases, it becomes more and more difficult to adequately cover the space with a reasonable number of samples using the statistically-based sampling methods.

Generating the sample points using either logarithmic random sampling or uniform random sampling is very rapid. For the other methods, there is additional overhead in LHS (due to stratification and grouping processes), in CVT sampling (due to its iterative nature), and in model-constrained sampling (due to solving the optimization

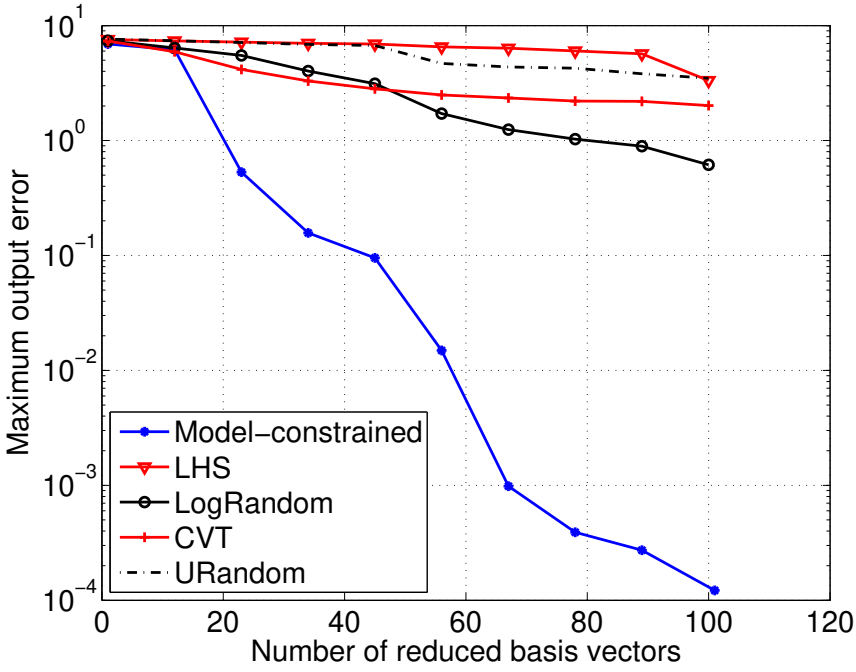
problem). We also note that while logarithmic random sampling is less expensive (in terms of overhead) than the CVT and LHS sampling methods, it leads to more accurate reduced models in the case of the thermal fin problem.

For the statistically-based methods that have some random elements (i.e. LHS, uniform random sampling and logarithmic random sampling), the comparison in Figure 4-11 is limited to one random set of parameters. A different random draw could lead to different results and different reduced model performance. However, first we note that the relative performance difference of the model-constrained sampling approach is significant; thus, variation in performance due to variability in the random draw is not expected to alter the conclusion that the model-constrained method performs better than the other methods. Second, even if one particular random draw could lead to a reduced model that outperforms the model-constrained approach, it is preferable to have a systematic methodology with repeatable performance, rather than relying on a favorable random sample to achieve good results.

Next, we study the sensitivity of the quality of model-constrained sample points using different methods to generate the initial guesses. In particular, we take the points in parameter space corresponding to the logarithmic random, uniform random, LHS, and CVT samplings that were previously used to obtain the results in Figure 4-11 as initial guesses, $\mathbf{z}^0$, for the model-constrained sampling method. For each parameter point, we then solve an optimization problem to determine the corresponding $\mathbf{z}^*$, which becomes the new sample point. Since our algorithm uses the error indicator, we only perform a large-scale solve at the optimal parameter point $\mathbf{z}^*$; hence the cost is not increased (with the exception of the optimization solver overhead). Figures 4-12(a) and (b) show the maximum output error versus the number of reduced basis vectors for both the 11- and 21-parameter cases. It can be seen that the quality of the reduced model resulting from the model-constrained approach is relatively insensitive to the initial guesses, at least for the thermal fin problem considered here. That is, the quality of the sample points using different methods to generate initial guesses is more or less the same. Again, this emphasizes the benefit of a systematic, repeatable point selection criterion.

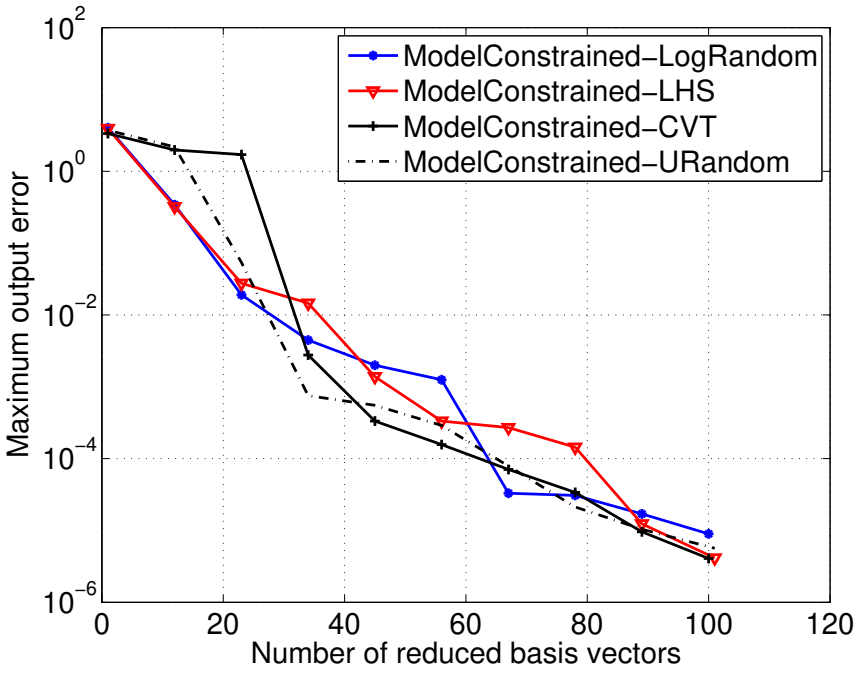


(a) 11-parameter case

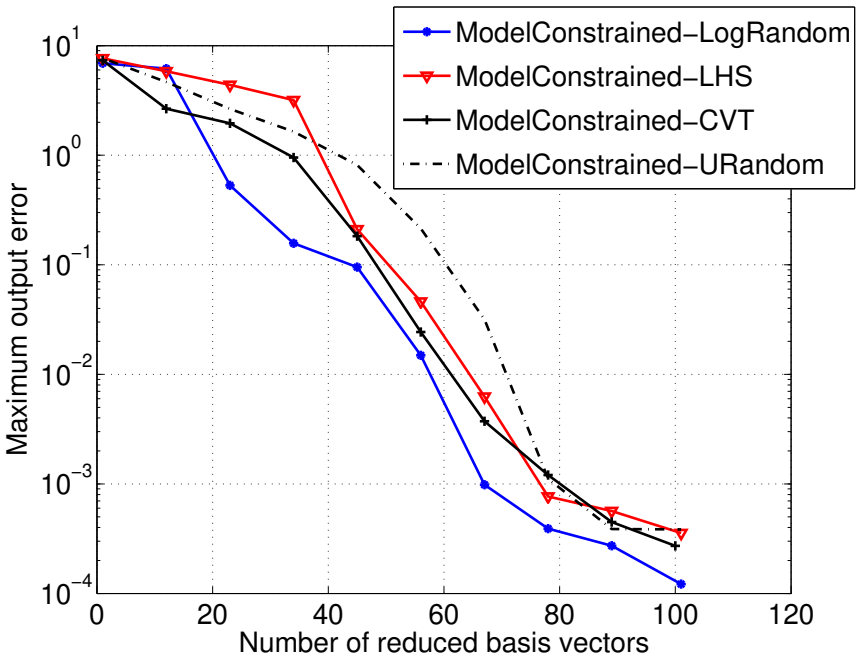


(b) 21-parameter case

Figure 4-11: Model-constrained sampling versus LHS, uniform random sampling (URandom), logarithmic random sampling (LogRandom) and CVT sampling. 100 sample points are used for all methods.



(a) 11-parameter case



(b) 21-parameter case

Figure 4-12: Model-constrained sampling method using logarithmic random sampling (ModelConstrained-LogRandom), LHS (ModelConstrained-LHS), uniform random (ModelConstrained-URandom), and CVT (ModelConstrained-CVT) to generate initial parameter guesses, $\mathbf{z}^0$. 100 sample points are used for all methods of initialization.

4.3.5 Optimal Design Application

In this section, the reduced model is used as a surrogate model in the optimal design context. That is, instead of performing the design task on the full model, the design task is carried out on the reduced model, which is much less expensive. Recall that the design problem is to find a combination of thermal conductivities, Biot number, and lengths to minimize the output average temperature (and hence maximize the cooling efficiency).

Here we consider the thermal fin optimal design problem with 11 parameters, as shown in Table 4.5. The reduced model is generated using the model-constrained approach as in Section 4.3.4 with 100 basis vectors. Using the same initial design, which is shown in the second column in Table 4.5, we obtain different optimal solutions using the full and the reduced model, as shown in the third and fourth columns of Table 4.5, respectively. Although the values of the objective function for these two designs are similar, the optimal solution obtained from the reduced model does not satisfy the optimality conditions of the full problem.

To improve the quality of the reduced model we run 50 additional greedy cycles to obtain 50 more basis vectors. The optimal design for the reduced model with 150 basis vectors is shown in the fifth column of Table 4.5. Again, the optimal solution of the reduced model is different from that of the full one, although the objective function value is the same. In this case, the optimum found by the reduced model is a local minimum of the full model.

The reduction factor in model size is 119 (from the full model with 17899 states to the reduced model with 150 states), but the speed-up factor in solving the optimal design problem is $58.12/2.49 \approx 23$ as shown in Table 4.5. This is because the full system matrix is sparse ($\mathbf{A} \in \mathbb{R}^{17899\times 17899}$ has 120,603 nonzero entries) while the reduced matrix is not ($\mathbf{A_r} \in \mathbb{R}^{150\times 150}$ has 22,500 nonzero entries). For this problem with 17899 states (which is relatively small), the offline cost of determining the reduced model is not offset by computational savings in solving the optimal design problem, unless there is a need to solve the optimization problem in real time (in which case

the reduced model is essential). Chapter 6 presents a large-scale probabilistic analysis application. It will be shown that in that case, the trade-off between the online and offline cost is much more dramatic.

Table 4.5: Optimal design for the thermal fin problem with 11 parameters: full model (17899 states) versus reduced models with 100 and 150 states.

Parameters	Initial design	Optimal solution using the full model	Optimal solution with 100 basis vectors	Optimal solution with 150 basis vectors
κ_1	2.8657	10.0	0.10	10.0
κ_2	2.1151	10.0	10.0	10.0
κ_3	0.1888	10.0	0.1	0.1
κ_4	0.2349	5.25	10.0	10.0
κ_5	0.1947	5.25	10.0	0.1
Bi	2.5531	5.0	5.0	5.0
b_1	0.2529	5.0	5.0	5.0
b_2	0.3324	10.0	10.0	10.0
b_3	8.2753	10.0	10.0	10.0
b_4	0.7236	2.5	2.5	2.5
b_5	1.6142	5.0	5.0	5.0
Optimal output		2.678e-03	2.673e-03	2.678e-03
CPU time		58.12s	6.15s	2.49s

Chapter 5

Computational Fluid Dynamic Models

Recent developments in the field of CFD have led to the use of higher-order finite element discretizations for PDEs. These schemes have advantages over traditional finite-volume methods by introducing higher-order accuracy compactly within grid elements and thus providing a significant decrease in the computational cost to obtain reliably accurate solutions. A Discontinuous Galerkin (DG) formulation is used in this work. The unsteady flow solver described in this thesis is part of a larger effort that includes an adaptive meshing utility, a multigrid solution algorithm, gradient-based optimization capability, and high-order visualization [117].

In this chapter, a steady DG formulation for the Euler equations is first outlined. A linearized unsteady DG formulation is then presented, and validated via numerical comparisons with experimental data. Next, a linearized model for incorporating geometric variability into the unsteady CFD model is described. Finally, the applicability of the linearized geometric variability model is assessed via numerical experiments.

5.1 Steady CFD Model

In this section we briefly review the DG discretization and the solution method for the two-dimensional Euler equations as in Cockburn and Shu [118] and Fidkowski

and Darmofal [119]. The two-dimensional Euler equations are given by:

$$\frac{\partial \mathbf{w}}{\partial t} + \nabla \cdot \mathcal{F}(\mathbf{w}) = 0, \tag{5.1}$$

where $\mathbf{w}$ is the conservative state vector,

$$\mathbf{w} = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho E \end{pmatrix}, \tag{5.2}$$

and $\mathcal{F} = (\mathbf{F}^x, \mathbf{F}^y)$ is the inviscid Euler flux

$$\mathbf{F}^x = \begin{pmatrix} \rho u \\ \rho u^2 + P \\ \rho uv \\ \rho uH \end{pmatrix}, \quad \mathbf{F}^y = \begin{pmatrix} \rho v \\ \rho uv \\ \rho v^2 + P \\ \rho vH \end{pmatrix}. \tag{5.3}$$

In the above equations, ρ is the density, u and v are respectively the $x-$ and $y-$component of velocity, E is the energy, P is the pressure, and $H = E + P/\rho$ is the total enthalpy. The equation of state is

$$P = (\gamma - 1)\left[\rho E - \frac{1}{2}\rho\left(u^2 + v^2\right)\right], \tag{5.4}$$

where γ is the ratio of specific heats.

As in the continuous finite element method, the first step in the DG method is to discretize the domain under consideration, Ω, into elements Ω^e. Next, a space of polynomials of degree at most $\bar{p}$, $\mathcal{U}_h^{\bar{p}}(\Omega^e)$, is defined on each element, where h denotes a representative element size for the discretization (e.g. the size of the smallest element). On each element Ω^e, the approximate solution $\mathbf{w}_h$ can be found by enforcing the

nonlinear conservation law (5.1) locally, for all test functions $\mathbf{v}_h \in \mathcal{U}_h^{\bar{p}}(\Omega^e)$:

$$\int_{\Omega^e} \mathbf{v}_h^T \frac{\partial \mathbf{w}_h}{\partial t} d\Omega^e - \int_{\partial\Omega^e} \nabla \mathbf{v}_h^T \cdot \mathcal{F}(\mathbf{w}_h) d\Omega^e$$

$$+ \int_{\partial\Omega^e \setminus \partial\Omega} \left(\mathbf{v}_h^+\right)^T \mathcal{H}(\mathbf{w}_h^+, \mathbf{w}_h^-, \hat{\mathbf{n}}) ds$$

$$+ \int_{\partial\Omega^e \cap \partial\Omega} \left(\mathbf{v}_h^+\right)^T \mathcal{H}_{\mathrm{bd}}(\mathbf{w}_h^+, \mathbf{w}_h^-, \hat{\mathbf{n}}) ds = 0, \tag{5.5}$$

where $\partial\Omega$ and $\partial\Omega^e$ are the boundaries of the entire domain Ω and the element Ω^e, respectively, and $\hat{\mathbf{n}}$ denotes the outward-pointing normal on the boundaries of the element. The terms $\mathcal{H}(\mathbf{w}_h^+, \mathbf{w}_h^-, \hat{\mathbf{n}})$ and $\mathcal{H}_{\mathrm{bd}}(\mathbf{w}_h^+, \mathbf{w}_h^-, \hat{\mathbf{n}})$ are numerical flux functions for interior and boundary edges, respectively, where $()^+$ and $()^-$ denote values taken from the interior and exterior of the element. The interior flux function is computed using the Roe-averaged flux function [120] and contributes over element boundaries that do not belong to the domain boundary, denoted by $\partial\Omega^e \setminus \partial\Omega$. The fluxes on the common boundaries of $\partial\Omega^e$ and $\partial\Omega$, denoted by $\partial\Omega^e \cap \partial\Omega$, are computed using the inner state and boundary condition data.

The final form of the DG discretization is constructed by selecting a basis for $\mathcal{U}_h^{\bar{p}}(\Omega^e)$. The approximate solution $\mathbf{w}_h$ on each element is assumed to be a linear combination of the basis functions ϕ_j,

$$\mathbf{w}_h(t, x, y) = \sum_{j=1}^{n_b} \hat{\mathbf{w}}_j(t) \phi_j(x, y), \tag{5.6}$$

where $\hat{\mathbf{w}}_j(t)$ gives the modal content of ϕ_j on element Ω^e, and n_b is the number of basis functions required to describe $\mathcal{U}_h^{\bar{p}}(\Omega^e)$ (e.g. $n_b = 1$ for $\bar{p} = 0$ and $n_b = 3$ for $\bar{p} = 1$). The complete set of unknown quantities for the DG formulation comprises the values of $\hat{\mathbf{w}}_j(t)$ for every element in the spatial domain. These quantities will be contained in the vector $\bar{\mathbf{w}} \in \mathbb{R}^n$, where n is the total number of unknowns, which depends both on the number of elements in the discretization and on the polynomial order $\bar{p}$.

For steady-state flows, pseudo time-stepping is used to improve the initial transient

behavior of the solver. A backward Euler discretization in time is used so that the final discrete equations are

$$\mathbf{E}\frac{1}{\Delta t}\left(\bar{\mathbf{w}}^{n+1}-\bar{\mathbf{w}}^{n}\right)+\mathbf{Q}(\bar{\mathbf{w}}^{n+1})=0 \tag{5.7}$$

where Δt is the timestep, $\bar{\mathbf{w}}^n$ is the solution $\bar{\mathbf{w}}$ at a time t^n, $\mathbf{E}$ is the mass matrix, and $\mathbf{Q}$ is the vector representing the final three terms of (5.5). This nonlinear system is solved using a $\bar{p}$-multigrid scheme with a line Jacobi smoother [117, 119].

5.2 Unsteady CFD Model

The unsteady Euler equations using the DG spatial discretization can be written

$$\mathbf{E}\frac{d\bar{\mathbf{w}}}{dt}+\mathbf{Q}(\bar{\mathbf{w}},\mathbf{u})=0 \tag{5.8}$$

where $\mathbf{u}(t)\in\mathbb{R}^p$ is a vector containing p external forcing inputs, such as prescribed motion of the domain boundary or incoming flow disturbances. In addition, we define a set of q output quantities of interest, contained in the vector $\mathbf{y}\in\mathbb{R}^q$ and defined by the nonlinear function $\mathcal{P}$

$$\mathbf{y}=\mathcal{P}(\bar{\mathbf{w}}). \tag{5.9}$$

For unsteady computations, a second-order backward Euler temporal discretization is applied to (5.8). The resulting nonlinear equations are solved using a Newton solver. Grid motion is implemented using a simple Jacobi smoothing formulation. The motion of grid point j is defined by the change in $x-$ and $y-$coordinates, $(\delta x_j, \delta y_j)$, and computed as

$$\delta x_j=\frac{1}{r}\sum_{k\in N_j}\delta x_k,\qquad \delta y_j=\frac{1}{r}\sum_{k\in N_j}\delta y_k \tag{5.10}$$

where r is the number of the neighbors chosen to influence the grid point j and N_j is the set containing the corresponding set of r neighboring points. Larger values of r lead to increased grid motion and smoother grids. The motion of grid points on the domain boundary is prescribed according to the corresponding external input (e.g.

prescribed motion of an airfoil).

In many cases of interest, the unsteady flow solution can be assumed to be a small perturbation from steady-state conditions. This allows the unsteady governing equations to be linearized, which reduces the computational cost of solution considerably. The linearized version of equations (5.8, 5.9) can be written in standard state-space form, which is a special case of the general model (2.4) that is obtained from the DG CFD linearization,

$$\mathbf{E}\frac{d\mathbf{x}}{dt} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}, \qquad \mathbf{y} = \mathbf{C}\mathbf{x}, \tag{5.11}$$

where $\mathbf{x} \in \mathbb{R}^n$ is the state vector containing the n perturbations in flow unknowns from the steady-state solution $\bar{\mathbf{w}}_{ss}$, that is $\bar{\mathbf{w}}(t) = \bar{\mathbf{w}}_{ss} + \mathbf{x}(t)$. The matrices $\mathbf{A} \in \mathbb{R}^{n\times n}$, $\mathbf{B} \in \mathbb{R}^{n\times p}$, and $\mathbf{C} \in \mathbb{R}^{q\times n}$ in (5.11) have constant coefficients evaluated at steady-state conditions and arise from the linearization of (5.8) and (5.9) as follows.

$$\mathbf{A} = \frac{\partial \mathbf{R}}{\partial \bar{\mathbf{w}}}, \quad \mathbf{B} = \frac{\partial \mathbf{R}}{\partial \mathbf{u}}, \quad \mathbf{C} = \frac{\partial \mathcal{P}}{\partial \bar{\mathbf{w}}}. \tag{5.12}$$

By considering harmonic inputs at a frequency ω, $\mathbf{u} = \bar{\mathbf{u}}e^{j\omega t}$, the linearized equations (5.11) can also be written in the frequency domain as

$$[j\omega\mathbf{E} - \mathbf{A}]\,\bar{\mathbf{x}} = \mathbf{B}\bar{\mathbf{u}}, \qquad \bar{\mathbf{y}} = \mathbf{C}\bar{\mathbf{x}}, \tag{5.13}$$

where $\mathbf{x} = \bar{\mathbf{x}}e^{i\omega t}$ and $\mathbf{y} = \bar{\mathbf{y}}e^{i\omega t}$.

5.3 CFD Model Validation

Results are presented for two unsteady examples: a NACA 0012 airfoil and the first standard cascade configuration. Results are shown to validate the unsteady CFD models by comparison with airfoil and cascade experimental data.

5.3.1 NACA 0012 Airfoil Example

Figure 5-1 shows nonlinear and linearized CFD results compared with experimental data [1] for a NACA 0012 airfoil in rigid pitching motion. The steady-state flow has a Mach number of 0.6 and angle of attack of 2.89°. The unsteady simulation is then carried out with the pitching input $\alpha(t) = 2.41\sin(0.8874t)$, where $\alpha(t)$ is the perturbation angle about the steady state angle of attack. Figure 5-1 shows the pressure coefficient distribution at one particular instant in time; it can be seen that the nonlinear CFD predictions match well with the experimental data. There is a region with a weak shock on the upper surface that the linearized code cannot resolve. Elsewhere the agreement between the experimental data and linearized model is acceptable.

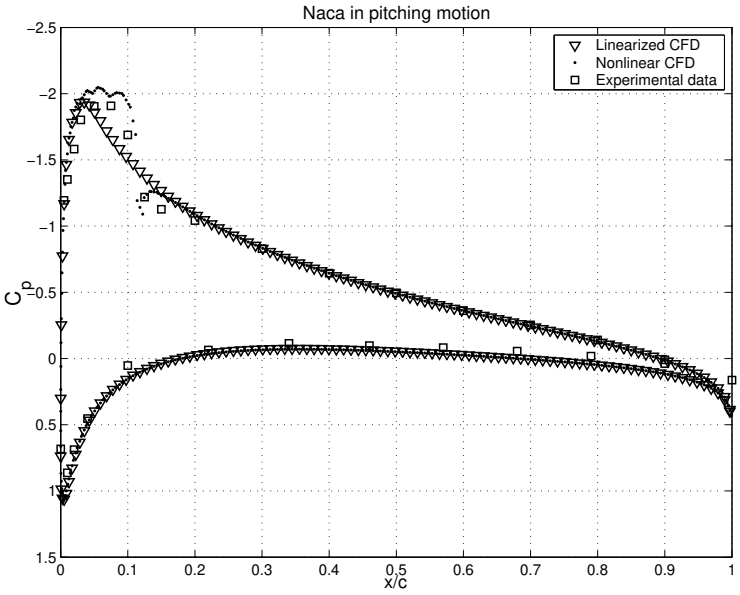


Figure 5-1: Comparison of CFD predictions and experimental data [1] for NACA 0012 airfoil in unsteady pitching motion. The pressure coefficient distribution on the airfoil surface is shown for $t = 0.00311$.

5.3.2 The First Standard Cascade Configuration Example

Unsteady computations for cascade flows can be carried out efficiently by exploiting spatial periodicity and linearity. By working with the frequency domain equations

(5.13), complex periodicity conditions can be used to represent the effects of neighboring blade passages for each interblade phase (IBP) angle [121]. All cascade linearized CFD computations are therefore performed in the frequency domain on a single blade passage. Similarly, this periodicity can be exploited to provide efficient implementations for creating the reduced-order model.

For cascade flows, experimental data for a number of standard configurations are available [122]. The first standard configuration with a steady-state inflow Mach number of 0.18 and flow angle β of 62° is considered. The cascade operates in unsteady rigid pitching motion. Both experimental data and other CFD data [123, 124] are taken from Bolcs and Fransson [122] and Fransson and Verdon [125]. Figures 5-2 and 5-3 show the magnitude and the phase of the unsteady pressure coefficients on the first blade as a function of pitching frequency at an IBP of $\sigma = 45°$. Figures 5-4 and 5-5 show the pressure coefficients at $\sigma = -45°$. It can be seen that the DG linearized results are comparable to other CFD results and are acceptably close to the experimental data. It should be noted that the DG method is very sensitive to the geometry representation of the blade. A very coarse cascade geometry is available from Bolcs and Fransson [122]. This geometry is then splined to obtain smoother blade surfaces. It is expected that a more accurate result could be obtained with the DG method if more accurate geometry data were available.

5.4 Geometric Variability Model

Mistuning, or blade-to-blade variation, is an important consideration for aeroelastic analysis of bladed disks, since even small variations among blades can have a large impact on the forced response and consequently the high-cycle fatigue properties of the engine. The effects of blade structural mistuning (variations in mass and stiffness properties) have been extensively studied, see for example Refs. 14–19; however, due to the prohibitively high computational cost of performing probabilistic analysis with a CFD model, the aerodynamic effects due variations in geometry are less understood. Lim *et al.* [126] have incorporated geometric mistuning effects into structural

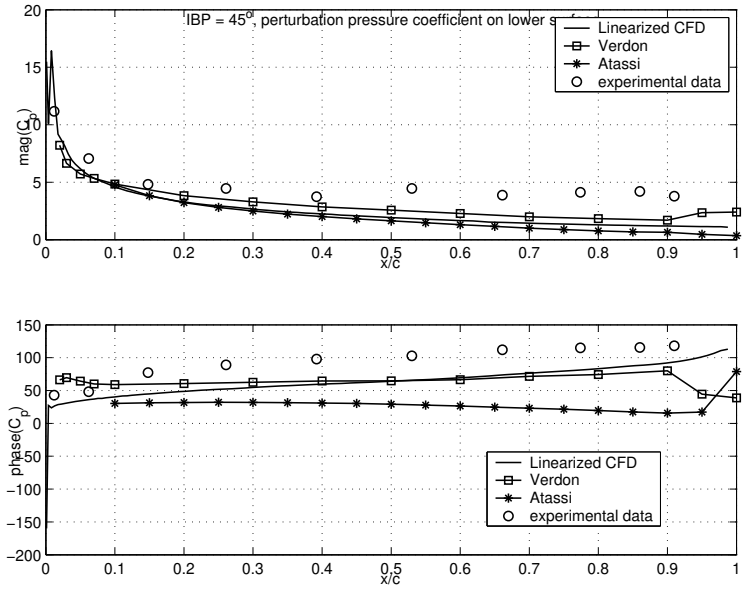


Figure 5-2: First standard configuration in unsteady pitching motion with $M = 0.18$, $\beta = 62°$. Magnitude and phase of the unsteady pressure coefficient distribution on the lower surface with $\sigma = 45°$.

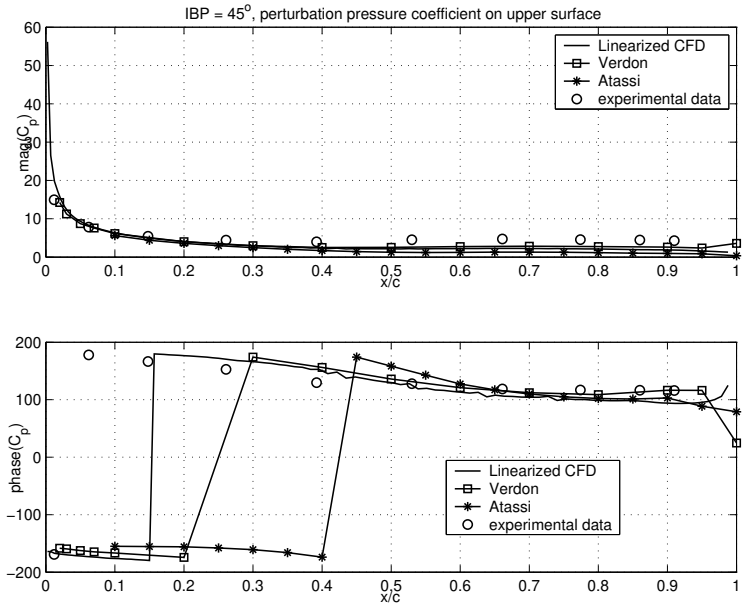


Figure 5-3: First standard configuration in unsteady pitching motion with $M = 0.18$, $\beta = 62°$. Magnitude and phase of the unsteady pressure coefficient distribution on the upper surface with $\sigma = 45°$.

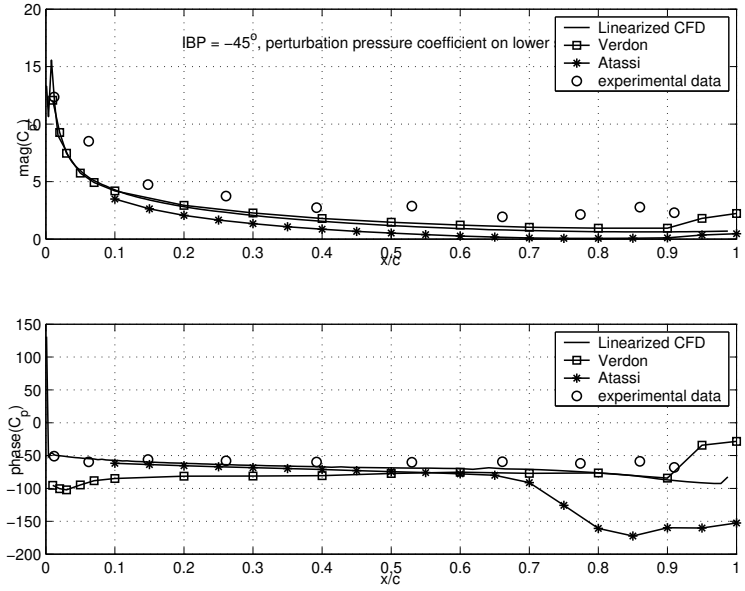


Figure 5-4: First standard configuration in unsteady pitching motion with $M = 0.18$, $\beta = 62°$. Magnitude and phase of the unsteady pressure coefficient distribution on the lower surface with $\sigma = -45°$.

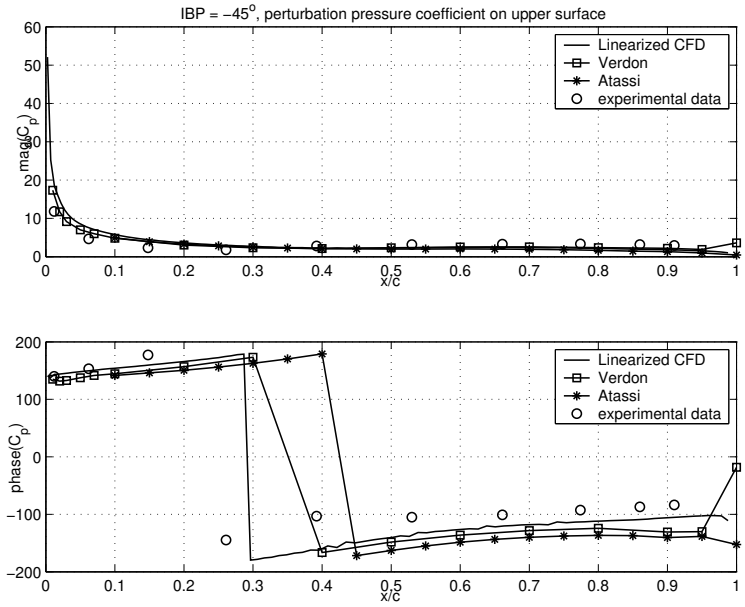


Figure 5-5: First standard configuration in unsteady pitching motion with $M = 0.18$, $\beta = 62°$. Magnitude and phase of the unsteady pressure coefficient distribution on the upper surface with $\sigma = -45°$.

responses of bladed disks. In their work, the mode-acceleration method was used to convert the effect of geometric mistuning to that of external forces of the tuned disks. Truncated sets of tuned system modes compensated by static modes—generated by external forces that were constructed from mistuning—were then used to obtain efficient and accurate structural reduced models.

Since the manufactured geometric mistuning space is large, Garzon and Darmofal [127–129], Brown *et al.* [130], Ghiocel [131], and Sinha *et al.* [132] have used the principle component analysis (PCA) [133] to construct a reduced geometric variability model. It was found that a handful of PCA geometric variability modes can capture the manufactured variability well. In particular, Garzon and Darmofal [127–129] have used the reduced geometric variability model to investigate the impact of geometric variability on axial compressor steady aerodynamic performance using Monte Carlo simulation based on a large-scale nonlinear CFD model. They found that the mean loss under the presence of geometric mistuning was approximately 20% larger than the nominal loss. Since each large-scale nonlinear CFD simulation is very expensive, parallel computers were used in order to perform the probabilistic analysis using Monte Carlo simulation. Here we consider incorporating the effects of geometry variability into the linearized unsteady CFD model.

Following Ref. 127, a general geometry, g, can be expressed as

$$g = g_n + \bar{g} + \sum_{i=1}^{n_s} \sigma_i z_i v_i, \tag{5.14}$$

where g_n is the nominal geometry, $\bar{g}$ is the average geometric variation, v_i are geometric mode shapes, and n_s is the number of mode shapes used to represent the variation in geometry. The geometric mode shapes could be computed, for example, by performing the PCA on a manufacturing sample of geometries. In that case, the parameters z_i in (5.14) are random numbers normally distributed with zero mean and unity variance, $z_i \in N(0,1)$, and σ_i is the standard deviation of the geometric data attributable to the i^{th} mode; thus the product $\sigma_i z_i$ is the amount by which the mode v_i contributes to the geometry g. A detailed description of the methodology

underlying this geometric model can be found in Ref. 127.

The key assumption in (5.14) is that any manufacturing geometry is a linear combination of geometries in the manufacturing geometry sample. In other words, the manufacturing geometry sample is assumed to be large enough to span the entire mistuning space.

5.5 Linearized Unsteady CFD Model with Geometric Variability

Using the model (5.14), a general geometry $g(\mathbf{z})$ is specified by the parameter vector $\mathbf{z} = [z_1, z_2, \ldots, z_{n_s}]^T$, which describes the geometry variability in terms of the geometry modes. The linearized CFD system corresponding to geometry $g(\mathbf{z})$ is given by

$$\mathbf{E}(\bar{\mathbf{w}}_{ss}(g(\mathbf{z})), g(\mathbf{z}))\dot{\mathbf{x}} = \mathbf{A}(\bar{\mathbf{w}}_{ss}(g(\mathbf{z})), g(\mathbf{z}))\mathbf{x} + \mathbf{B}(\bar{\mathbf{w}}_{ss}(g(\mathbf{z})), g(\mathbf{z}))\mathbf{u}, \quad (5.15)$$

$$\mathbf{y} = \mathbf{C}(\bar{\mathbf{w}}_{ss}(g(\mathbf{z})), g(\mathbf{z}))\mathbf{x}, \quad (5.16)$$

where the CFD system matrices $\mathbf{E}, \mathbf{A}, \mathbf{B}$ and $\mathbf{C}$ are in general both a function of the geometry, $g(\mathbf{z})$, and of the steady-state solution, $\bar{\mathbf{w}}_{ss}(g(\mathbf{z}))$, which is itself also a function of the geometry. To solve the CFD system (5.15), (5.16), for each geometry g we must firstly compute the steady-state solution, $\bar{\mathbf{w}}_{ss}(g(\mathbf{z}))$, secondly evaluate the linearized matrices $\mathbf{E}, \mathbf{A}, \mathbf{B}$ and $\mathbf{C}$, and thirdly solve the resulting large-scale linear system. This is a computationally prohibitive proposition for applications such as probabilistic analysis, where thousands of geometry perturbations may be analyzed over many random samples $\mathbf{z}$. For example, if one such analysis takes three minutes to perform, then 50,000 analyses would take more than three months of CPU time!

For convenience of notation, we write the dependence of the CFD matrices on the parameter $\mathbf{z}$ as $\mathbf{E}(\bar{\mathbf{w}}_{ss}(g(\mathbf{z})), g(\mathbf{z})) = \mathbf{E}(\mathbf{z})$, $\mathbf{A}(\bar{\mathbf{w}}_{ss}(g(\mathbf{z})), g(\mathbf{z})) = \mathbf{A}(\mathbf{z})$, $\mathbf{B}(\bar{\mathbf{w}}_{ss}(g(\mathbf{z})), g(\mathbf{z})) = \mathbf{B}(\mathbf{z})$, and $\mathbf{C}(\bar{\mathbf{w}}_{ss}(g(\mathbf{z})), g(\mathbf{z})) = \mathbf{C}(\mathbf{z})$. We use the expansion given by equation (5.14), which represents a general geometry as a perturbation

about the average geometry $g_0 = g_n + \bar{g}$, to derive an approximate model for representing the effects of geometry variations. Instead of computing the linearized CFD matrices exactly for any random variability $\mathbf{z}$, we choose to linearize the relationships $\mathbf{E}(\mathbf{z})$, $\mathbf{A}(\mathbf{z})$, $\mathbf{B}(\mathbf{z})$, and $\mathbf{C}(\mathbf{z})$ [65,66]. A more general approach for a general nonlinear function can be found in Barrault *et al.* [61,62,68]. We define the linearized unsteady CFD model for the average geometry $g_0 = g_n + \bar{g}$ by the matrices $\mathbf{E_0}$, $\mathbf{A_0}$, $\mathbf{B_0}$, and $\mathbf{C_0}$, with corresponding solution $\mathbf{x}_0$. That is, for $\mathbf{z} = 0$ we have

$$\mathbf{E}_0\dot{\mathbf{x}}_0 = \mathbf{A}_0\mathbf{x}_0 + \mathbf{B}_0\mathbf{u}, \tag{5.17}$$

$$\mathbf{y}_0 = \mathbf{C}_0\mathbf{x}_0. \tag{5.18}$$

Using a Taylor series expansion about $\mathbf{z} = 0$ for the matrix $\mathbf{A}(\mathbf{z})$ gives

$$\mathbf{A}(\mathbf{z}) = \mathbf{A}_0 + \left.\frac{\partial \mathbf{A}}{\partial z_1}\right|_{\mathbf{z}=0} z_1 + \ldots + \left.\frac{\partial \mathbf{A}}{\partial z_{n_s}}\right|_{\mathbf{z}=0} z_{n_s} + \ldots, \tag{5.19}$$

where the matrix partial derivatives denote componentwise derivatives, which can be evaluated through application of the chain rule. These derivatives are evaluated at average conditions, $\mathbf{z} = 0$. The matrices $\mathbf{E}(\mathbf{z}), \mathbf{B}(\mathbf{z})$ and $\mathbf{C}(\mathbf{z})$ can be expanded using formulae analogous to (5.19).

If the geometric variability (given by the product $\sigma_i z_i$) is sufficiently small, the constant and linear terms in the Taylor expansion (5.19) are sufficient to approximate the linearized matrices $\mathbf{A}(\mathbf{z})$ accurately, that is,

$$\mathbf{A}(\mathbf{z}) \approx \mathbf{A}_0 + \left.\frac{\partial \mathbf{A}}{\partial z_1}\right|_{\mathbf{z}=0} z_1 + \ldots + \left.\frac{\partial \mathbf{A}}{\partial z_{n_s}}\right|_{\mathbf{z}=0} z_{n_s}. \tag{5.20}$$

For $i = 1, 2, \ldots, n_s$, we define

$$\bar{\mathbf{E}}_i = \left.\frac{\partial \mathbf{E}}{\partial z_i}\right|_{\mathbf{z}=0}, \ \bar{\mathbf{A}}_i = \left.\frac{\partial \mathbf{A}}{\partial z_i}\right|_{\mathbf{z}=0}, \ \bar{\mathbf{B}}_i = \left.\frac{\partial \mathbf{B}}{\partial z_i}\right|_{\mathbf{z}=0}, \ \bar{\mathbf{C}}_i = \left.\frac{\partial \mathbf{C}}{\partial z_i}\right|_{\mathbf{z}=0}, \tag{5.21}$$

where the matrices $\bar{\mathbf{E}}_i$, $\bar{\mathbf{A}}_i$, $\bar{\mathbf{B}}_i$, and $\bar{\mathbf{C}}_i$ can be computed, for example, using a finite difference approximation of the respective derivatives. The approximate linearized

CFD model for any geometric variability $\mathbf{z}$ is then given by

$$\underbrace{\left(\mathbf{E}_0 + \sum_{i=1}^{n_s} \bar{\mathbf{E}}_i z_i\right)}_{\mathbf{E}(\mathbf{z})} \dot{\mathbf{x}} = \underbrace{\left(\mathbf{A}_0 + \sum_{i=1}^{n_s} \bar{\mathbf{A}}_i z_i\right)}_{\mathbf{A}(\mathbf{z})} \mathbf{x} + \underbrace{\left(\mathbf{B}_0 + \sum_{i=1}^{n_s} \bar{\mathbf{B}}_i z_i\right)}_{\mathbf{B}(\mathbf{z})} \mathbf{u}, \quad (5.22)$$

$$\mathbf{y} = \underbrace{\left(\mathbf{C}_0 + \sum_{i=1}^{n_s} \bar{\mathbf{C}}_i z_i\right)}_{\mathbf{C}(\mathbf{z})} \mathbf{x}. \quad (5.23)$$

It should be noted here that a number of large-scale steady state CFD solves are required in order to determine the matrices $\mathbf{A}_0$, $\mathbf{B}_0$, $\mathbf{C}_0$, $\mathbf{E}_0$, $\bar{\mathbf{A}}_i$, $\bar{\mathbf{B}}_i$, $\bar{\mathbf{C}}_i$ and $\bar{\mathbf{E}}_i$. For example, if central difference approximations to the matrix derivatives are used, a total of $2\sum_{i=1}^{n_s}+1$ large-scale steady state CFD solves is required. This is a one-time offline cost; once the matrices are computed, the approximate linearized system (5.22), (5.23) can be readily evaluated for an arbitrary geometry $g(\mathbf{z})$ without running the CFD steady solver. It is important to know that the size and components of $\mathbf{A}(\mathbf{z})$ are a function of the CFD grid. The CFD grid is in turn a function of the geometry, which is in turn a function of geometric variability $\mathbf{z}$. This implies that, for the finite difference to be accurate, the grid generation must satisfy the requirements that the size of the linearized matrices be the same for any geometric variability and that the components of the linearized matrices be a smooth function of the geometric variability $\mathbf{z}$. In order to satisfy these requirements, we first generate a CFD grid for the average geometry $g_0 = g_n + \bar{g}$. The grid for any new geometry corresponding to a nonzero geometric variability $\mathbf{z}$ is then generated by adding the perturbation $\sum_{i=1}^{n_s} \sigma_i z_i v_i$ to the boundary grid points and computing the new grid points using the Jacobi smoothing in (5.10).

It should also be noted that the model (5.22), (5.23) is valid only for small variations from the average geometry. Larger variations will incur larger errors, due to the neglect of the higher-order terms in the Taylor series expansion. In the next section, we present some analyses to quantify these errors. Even with this restriction, the model is useful for many applications where small geometric variations are of inter-

est; however, the approximate linearized model is still of high dimension, and thus is computationally too expensive for applications such as probabilistic analysis in which one needs to determine the unsteady aerodynamic response for many random geometries. In Chapter 6 we further reduce the cost of solving the approximate linearized system by using the model reduction method developed in Chapter 3 to create a reduced-order model that is accurate over both time and the geometric parameter space, described here by the vector $\mathbf{z}$.

5.6 Linearized Geometric Variability CFD Model Validation

Results are presented for forced response of a subsonic rotor blade that moves in unsteady rigid motion. The flow is modeled using the two-dimensional Euler equations written at the blade mid-section. The average geometry of the blade is shown in Figure 5-6 along with the unstructured grid for a single blade passage, which contains 4292 triangular elements. The Euler equations are discretized in space with the discontinuous Galerkin (DG) method described in Section 5.1. For the case considered here, the incoming steady-state flow has a Mach number of $M = 0.113$ and a flow angle of $\beta = 59°$. Flow tangency boundary conditions are applied on the blade surfaces. Since the rotor is cyclically symmetric, the steady flow in each blade passage is the same and the steady-state solution can be computed on a computational domain that describes just a single blade passage. Periodic boundary conditions are applied on the upper and lower boundaries of the grid to represent the effects of neighboring blade passages.

A linearized model is derived for unsteady flow computations by assuming that the unsteady flow is a small deviation from steady state as described in Section 5.2. An affine dependence of the linearized system matrices on the blade geometries is derived using the method described in Section 5.5. This leads to a system of the form (5.22), (5.23), where the state vector, $\mathbf{x}(t)$, contains the unknown perturbation

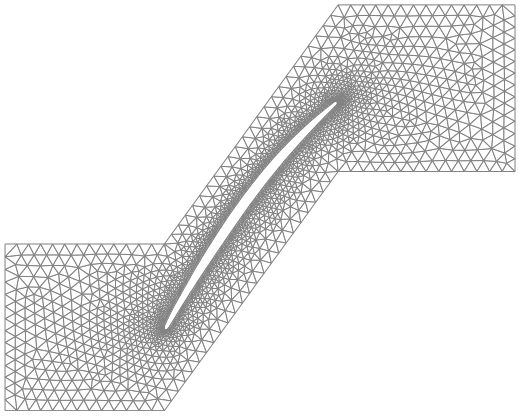


Figure 5-6: Geometry and CFD mesh for a single blade passage.

flow quantities (density, Cartesian momentum components and energy). For the DG formulation, the states are the coefficients corresponding to each nodal finite element shape function. Using linear elements, there are 12 degrees of freedom per element, giving a total state-space size of $n = 51,504$ states per blade passage. For the problem considered here, the forcing input, $\mathbf{u}(t)$, describes the unsteady motion of each blade, which in this case is assumed to be rigid plunging motion (vertical motion with no rotation). The outputs of interest, $\mathbf{y}(t)$, are the unsteady lift forces and pitching moments generated on each blade. The initial perturbation flow is given by $\mathbf{x}^0 = 0$.

Geometric modes were computed using a PCA model of data from 145 actual blades, measured at thirteen sections along the radial direction. The mid-section geometries were then extracted. Thus the parameter vector $\mathbf{z}$ contains the normally distributed random variables that describe perturbations in the geometry of each blade according to the model (5.14). Since the approximate linearized CFD model is only valid for small variations from the average geometry, the standard deviation of the actual manufacturing data was reduced by a factor of 6. As the results below show, this ensures that the geometric model remains in its region of applicability; however, it also highlights a limitation in the geometric model used here. By including ideas from [61, 62, 68] to handle a general nonlinear term in an efficient way together with the framework proposed here, a more general geometric model could be derived that

is applicable for larger geometric deviations.

In Figure 5-7, we consider a geometric model that uses the two dominant variability modes, $n_s = 2$. The figure shows the lift coefficient, C_L, and moment coefficient, C_M, of a blade in response to a pulse input in plunge for a particular geometry that corresponds to $z_1 = 1.59$, $z_2 = 1.59$. The response is computed using the exact linearized CFD model, i.e. the system (5.15), (5.16) and the approximate linearized model (5.22), (5.23) with $n_s = 2$ geometry modes. For reference, the response of the nominal blade is also shown in the figure. It can be seen that despite the small perturbation in geometry, the change in lift and moment coefficient responses is significant. The approximate linearized geometric model captures the unsteady response accurately.

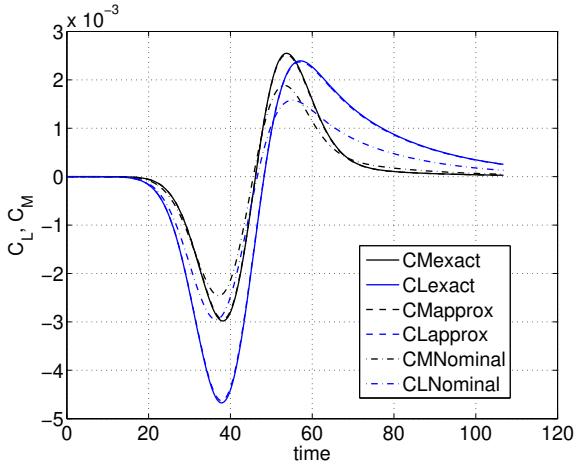


Figure 5-7: Lift coefficient, C_L, and moment coefficient, C_M, in response to a pulse input in blade plunge displacement for the nominal geometry and a perturbed geometry described by two geometric PCA modes with $z_1 = 1.59$, $z_2 = 1.59$. Perturbed geometry results are computed with both the exact and approximate linearized CFD model.

Table 5.1 shows the error in lift and moment outputs due to the linearized geometry approximation for several different blade geometries with a pulse input in plunge. The error e is defined as the 2-norm of the difference between the approximate and the exact linearized output as a percentage of the change between the exact and the

nominal output,

$$e = \frac{\sqrt{\int_0^{t_f} \|\mathbf{y}_e - \mathbf{y}_a\|_2^2 dt}}{\sqrt{\int_0^{t_f} \|\mathbf{y}_e - \mathbf{y}_o\|_2^2 dt}} \times 100\%, \tag{5.24}$$

where $\mathbf{y}_e$, $\mathbf{y}_a$, and $\mathbf{y}_o$ are respectively the exact, approximate, and nominal outputs. In the table, e_{C_M} denotes the error in moment coefficient response, while e_{C_L} denotes the error in lift coefficient response. In general, we expect the quality of the approximate model to be compromised as the size of the geometric perturbation increases. The errors shown in Table 5.1 for blade geometries in the tails of the distribution, i.e. those with large geometry variation, are deemed to be acceptable for the probabilistic application of interest here (although again we note that the actual variations observed in manufacturing data are larger). In fact, as will be shown in probabilistic analysis in Chapter 6, the error in aggregate quantities will be shown to be less. For applications where greater accuracy for large geometry variations is important (for example, determining the probability of failure would require the tail of the distribution to be resolved accurately), the results suggest that the approximate linearized CFD system is not appropriate. In such cases, one might consider including high order terms in (5.20), the Taylor series expansion of the CFD matrices.

Table 5.1: Error in approximate linearized model predictions for a pulse input in blade displacement for several different geometries.

Variability amplitudes	$e_{C_M}(\%)$	$e_{C_L}(\%)$
$z_1 = 1.59, z_2 = 1.59$	5.04	2.6
$z_1 = 1.59, z_2 = -1.59$	0.3	0.1
$z_1 = -1.59, z_2 = -1.59$	2.0	0.8
$z_1 = 3.0, z_2 = 3.0$	16.6	9.2
$z_1 = 3.0, z_2 = -3.0$	4.1	2.3
$z_1 = -3.0, z_2 = -3.0$	12.4	4.7

Chapter 6

CFD Probabilistic Analysis Application

The chapter begins with a numerical result for the Petrov-Galerkin projection in Section 2.3. The Petrov-Galerkin projection is then applied to obtain reduced models for all examples that follow. The model-constrained greedy-based adaptive sampling approach developed in Chapter 3 is applied to probabilistic analysis problems that depend on a large number of parameters. In particular, probabilistic analysis problems with four and ten geometric variability parameters will be considered. Finally, we compare the model-constrained sampling method with statistically-based sampling methods in the context of snapshot generation for model reduction for the probabilistic analysis problems with four and ten geometric variability parameters.

6.1 Galerkin Projection versus Petrov-Galerkin Projection

First, we present results to emphasize the importance of using an appropriate projection basis to perform the reduction. In particular, we verify the stability and the convergence of the projection method discussed in Section 2.3, in which the residuals are minimized sequentially. We consider a subsonic rotor blade that moves in

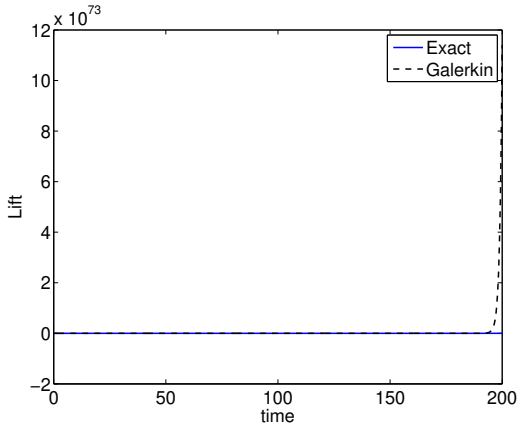
unsteady rigid motion, as described in Section 5.6. In this example, the nominal geometry with zero interblade phase angle, i.e. IBP = 0°, is used. The linearized CFD system (5.11) therefore does not depend on the geometric variability vector $\mathbf{z}$, and the unsteady flow can be computed on a computational domain which describes just a single blade passage. The forcing input, $\mathbf{u}(t)$ is assumed to be rigid plunging motion (vertical motion with no rotation). The output of interest, $\mathbf{y}$, is the unsteady lift force generated on the blade. The initial perturbation flow is given by $\mathbf{x}^0 = 0$.

Snapshots are taken by computing the response of the blade to a pulse input in plunging motion. For this input, the blade vertical position as a function of time is given by

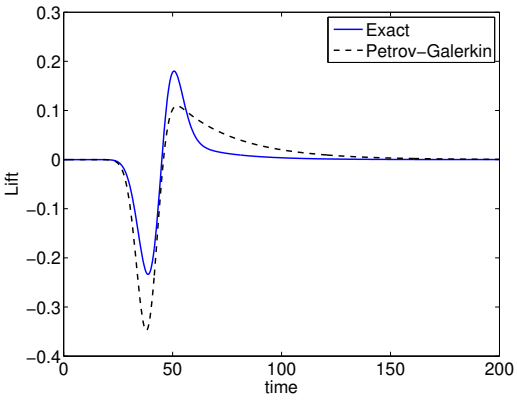
$$h(t) = \bar{h}e^{-g(t-t_0)^2}, \tag{6.1}$$

where the parameters $\bar{h} = 0.1$, $g = 0.02$, and $t_0 = 40$ are chosen based on the range of motions that are expected in practice, and all quantities are non-dimensionalized with the blade chord as a reference length and the inlet speed of sound as a reference velocity. The unsteady simulation is performed with a timestep of $\Delta t = 0.1$ from $t = 0$ to $t_f = 200$. A set of POD basis vectors is computed from this collection of 2000 snapshots and is used as the trial reduced basis $\mathbf{\Phi}$.

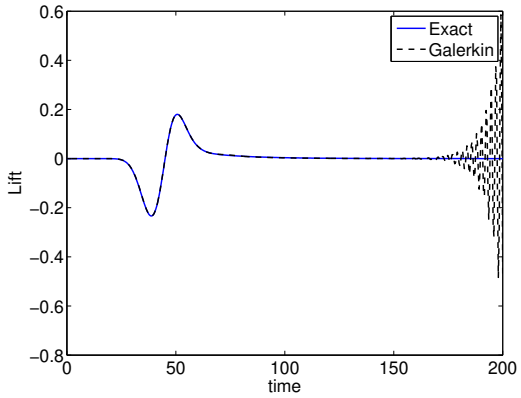
This example is of interest since the reduced model is unstable, depending on the number of reduced basis vectors, if the usual Galerkin projection is used. Here we guarantee the stability of the reduced model by minimizing the residual at each time step sequentially, and hence by the Petrov-Galerkin projection (2.35) in Chapter 2. The comparison between the Galerkin-projection-based and Petrov-Galerkin-projection-based reduced models can be seen in Figure 6-1. Note that we have used the same POD basis vectors for both methods. As expected, the Petrov-Galerkin approach is stable and improves the reduced model as more basis vectors are used while the Galerkin approach yields unstable models for any basis size less than $m = 10$.



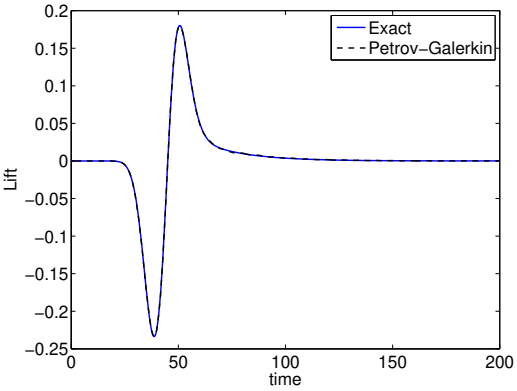
(a) Galerkin POD with 2 basis vectors



(b) Petrov-Galerkin POD with 2 basis vectors



(c) Galerkin POD with 8 basis vectors



(d) Petrov-Galerkin POD with 8 basis vectors

Figure 6-1: Comparison between the conventional Galerkin projection and the Petrov-Galerkin projection approaches. Solid lines are the full (exact) solution and dashed lines are reduced model results.

6.2 Probabilistic Analysis Application

The model-constrained adaptive model reduction method is applied to probabilistic analysis of a subsonic rotor blade that moves in unsteady rigid motion. Using Monte Carlo simulation (MCS) of a CFD model to quantify the impact of geometric variability on *unsteady* performance is a computationally prohibitive proposition. For example, if the unsteady analysis for one geometry takes three minutes to compute (a conservative estimate), the $\mathcal{O}(50,000)$ such analyses that would be required for a MCS would take more than three months of CPU time. Therefore, we desire to obtain a reduced-order model that captures both unsteady response and variation over blade geometries. Our method combines the reduced geometric variability model and the model-constrained adaptive sampling procedure of Algorithm 3.1 to obtain a reduced-order model that is valid over a range of forcing frequencies, aerodynamic damping, and small perturbations in blade geometries, and thus enables fast and accurate probabilistic analysis.

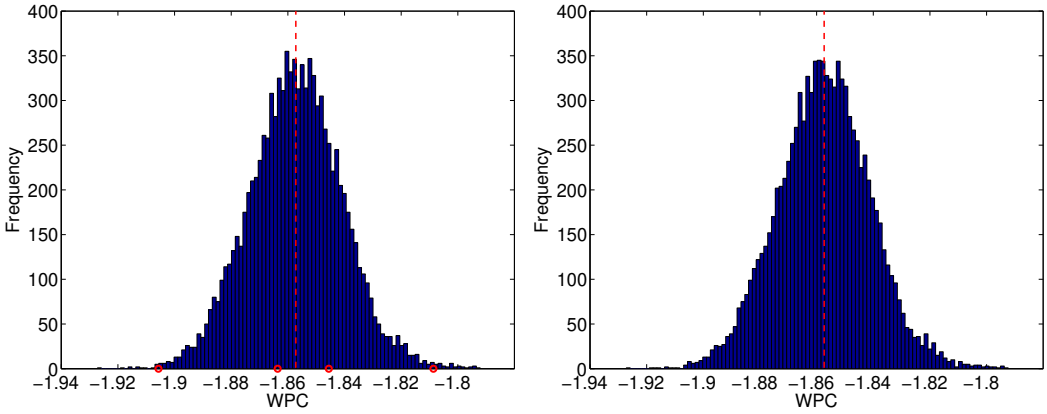
Results are shown here for the case of two blades moving with an interblade phase angle of 180°. First, each blade geometry is represented by two variability modes, giving $d = n_s = 4$ geometric parameters in this example. Applying the adaptive model reduction methodology with $\varepsilon = 10^{-4}$ and with the lift coefficients of the blades as the outputs of interest yields a reduced-order model of size $n_r = 201$ (for two blades). Again, in each greedy cycle, the number of Newton steps is observed to scale as $\mathcal{O}(d)$. Algorithm 3.1 requires 21 greedy cycles, over which a total of 21 optimal parameter points are found. Recall that when using the error indicator, the snapshots are only computed at the optimal parameters. Therefore, the full model is solved 21 times, and this computational cost dominates the other calculations. At each optimal parameter point, unlike the steady cases in which only one snapshot is found, for unsteady cases, 401 snapshots are taken uniformly over the time horizon $t_f = 200$. These snapshots are then used to update the reduced basis using a strategy similar to the third method in Table 4.3. That is, the newly computed snapshots are pre-processed by the POD method (where $\eta = 0.99999999$), and the dominant POD

basis vectors are then added to the current reduced basis using the Gram-Schmidt procedure. The geometric variability parameters, $\mathbf{z}$, are random numbers normally distributed with zero mean and unity variance, 99.7% of which are distributed in the interval $[-3.0, 3.0]$. The bound constraints for all parameters, in the greedy optimization problem, are therefore chosen to be

$$-3.0 \leq z_i \leq 3.0, \quad i = 1, \ldots, d. \tag{6.2}$$

Finally, the training input $\mathbf{u}(t)$ is chosen to be the pulse function in (6.1).

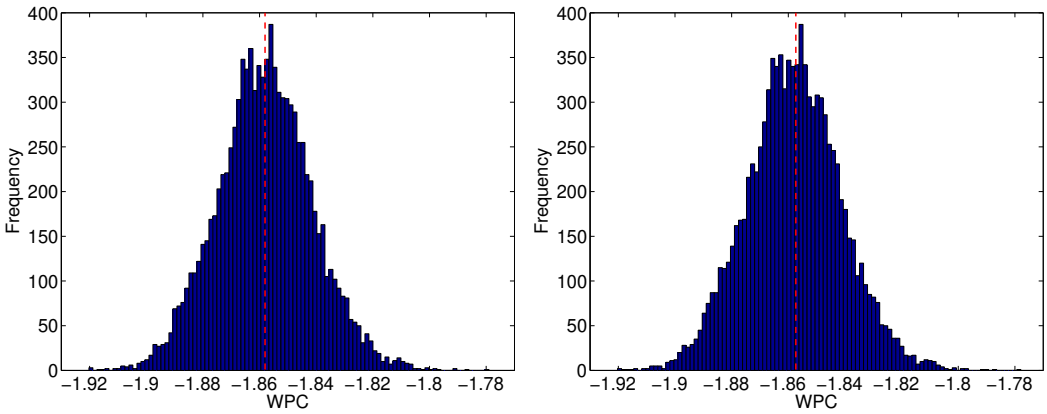
We now have a reduced model of size $n_r = 201$ that accurately captures the unsteady response of the original two-blade system with $n = 103,008$ states over the range of geometries described by the four geometric parameters. As an example of an application for which this reduced model is useful, we consider probabilistic analysis of the system. Specifically, we consider the impact of blade geometry variabilities on the work per cycle, which is defined as the integral of the blade motion times the lift force over one unsteady cycle. A MCS is performed in which 10,000 blade geometries are selected randomly from the given distributions for each blade. The same 10,000 geometries are analyzed using the approximate linearized CFD model and the reduced model. Figure 6-2 shows the resulting probability density functions (PDFs) of work per cycle for the first blade, computed using the approximate linearized CFD model and the reduced-order model. Figure 6-3 shows the PDFs of work per cycle for the second blade. Table 6.1 shows that the online CPU time required to compute the reduced model MCS is a factor of 2414 times smaller than that required for the CFD MCS. As can be also seen, the savings in the online cost are substantial, and more than justify the offline cost required to compute the reduced model. In practice, many more than 10,000 blade geometries are required to obtain a converged MCS; in this case, the computational cost of using the CFD model becomes prohibitive. These computational results were obtained on a dual core 64-bit personal computer with 3.2GHz Pentium processor. Tables 6.1 also shows the number of nonzeros in the full system matrix (sparse), $\mathbf{A}$, and the reduced system matrix (dense), $\mathbf{A}_r$.



(a) Full WPC for Blade 1

(b) Reduced WPC for Blade 1

Figure 6-2: Comparison of linearized CFD and reduced-order model predictions of work per cycle (WPC) for Blade 1. MCS results are shown for 10,000 blade geometries with four parameters. The same geometries are analyzed in each case. Dashed line denotes the mean.



(a) Full WPC for Blade 2

(b) Reduced WPC for Blade 2

Figure 6-3: Comparison of linearized CFD and reduced-order model predictions of work per cycle for Blade 2. MCS results are shown for 10,000 blade geometries for the case of four parameters. The same geometries were analyzed in each case. Dashed line denotes the mean.

Table 6.1 also compares the statistics of the two distributions. It can be seen from Figure 6-2, Figure 6-3 and Table 6.1 that the reduced-order model predicts the mean, variance and shape of the distribution of work per cycle accurately. To further verify the quality of the reduced model, we apply the Kolmogorov-Smirnov method [134], to test whether the reduced work per cycle results and the full work per cycle results are drawn from a same distribution. The results show that we cannot reject the hypothesis that the distribution is the same at a 5% significance level. It should be pointed out the Kolmogorov-Smirnov method is based on the maximum difference between the empirical cumulative distribution functions of work per cycle of the full and the reduced models. As shown in Figure 6-4 for Blade 1, the empirical cumulative distribution functions of work per cycle of the full and the reduced models are close to one another.

To further compare the reduced-order and CFD results, we pick four particular geometries corresponding to the left tail, right tail, mid-left and mid-right locations on the PDF of the first blade as indicated by the circles in Figure 6-2(a). In Table 6.2, the work per cycle is given for these four blade geometries as computed by the exact CFD model, the approximate linearized CFD model, and the reduced-order model. The table shows that again the approximate linearized CFD is in good agreement with the exact CFD, especially for the mid-left and mid-right cases, which have

Table 6.1: Linearized CFD model and reduced-order model MCS results for the case of four parameters. Work per cycle (WPC) is predicted for blade plunging motion at an interblade phase angle of 180° for 10,000 randomly selected blade geometries.

	CFD	Reduced
Model size	103,008	201
Number of nonzeros	2,846,056	40,401
Offline cost	—	2.8 hours
Online cost	501.1 hours	0.21 hours
Blade 1 WPC mean	-1.8572	-1.8573
Blade 1 WPC variance	2.687e-4	2.6819e-4
Blade 2 WPC mean	-1.8581	-1.8580
Blade 2 WPC variance	2.797e-4	2.799e-4

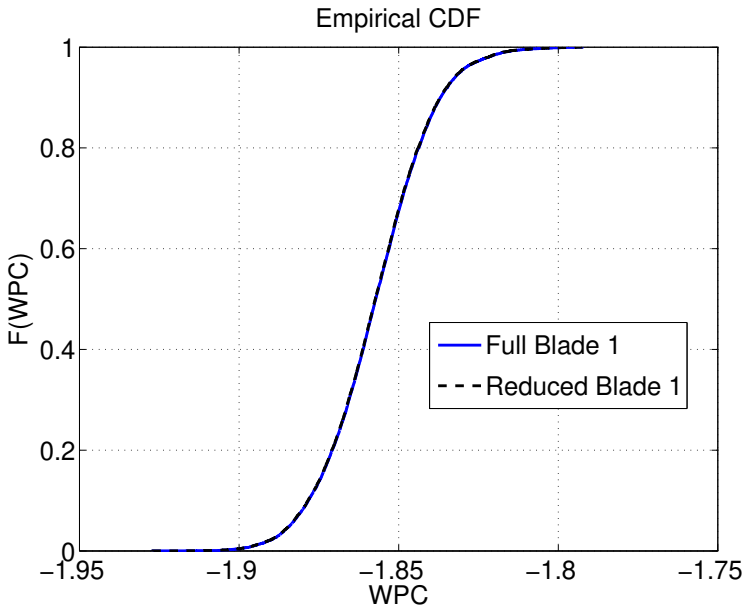


Figure 6-4: Comparison of the approximate linearized CFD and the reduced-order model empirical cumulative distribution functions (CDF) of work per cycle (WPC). The results are shown for 10,000 blade geometries for the case of four parameters. The same geometries were analyzed for both the full CFD and its reduced model.

smaller variability. In addition, the effectiveness of the adaptive model reduction methodology of Algorithm 3.1 can be seen from the good agreement between the approximate linearized CFD and the reduced results.

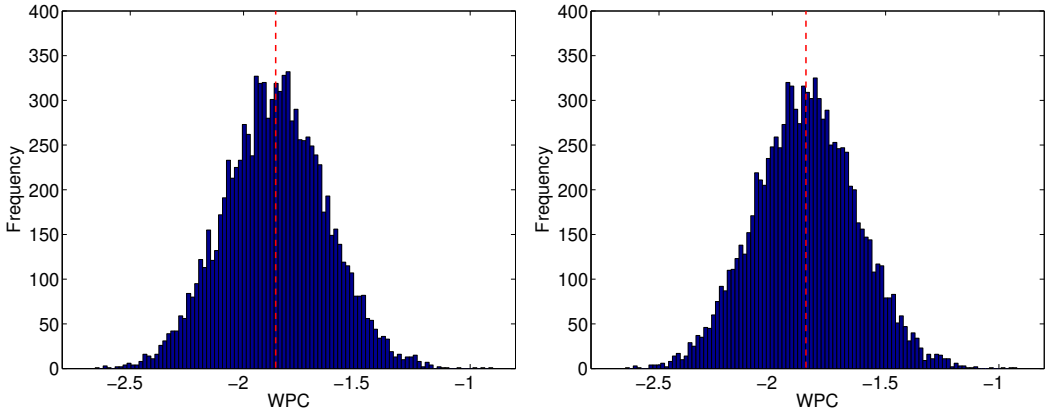
Table 6.2: Exact CFD, approximate CFD, and reduced-order model work per cycle prediction for the four geometries indicated in Figure 6-2(a).

	Exact	Approximate	Reduced
Left tail	-1.8973	-1.9056	-1.9060
Mid-left	-1.8637	-1.8636	-1.8638
Mid-right	-1.8459	-1.8455	-1.8458
Right tail	-1.8014	-1.8086	-1.8088

Next, we consider the case in which each blade geometry is represented by five variability modes, giving $d = n_s = 10$ geometric parameters in this example. Applying the adaptive model reduction methodology with $\varepsilon = 10^{-4}$ and with the lift coefficients of the blades as the outputs of interest yields a reduced-order model of size $n_r = 290$

(for two blades). Algorithm 3.1 requires 29 greedy cycles, over which a total of 29 optimal parameter points are found.

We now have a reduced model of size $n_r = 290$ that accurately captures the unsteady response of the original two-blade system with $n = 103,008$ states over the range of geometries described by the ten geometric parameters. A MCS is then performed in which 10,000 blade geometries are selected randomly from the given distributions for each blade. The same 10,000 geometries are analyzed using the approximate linearized CFD model and the reduced model. Figure 6-5 shows the PDFs of work per cycle for the first blade. Figure 6-6 shows the PDFs of work per cycle for the second blade. Table 6.3 shows that the online CPU time required to compute the reduced model MCS is a factor of 468 times smaller than that required for the CFD MCS. Note that the observed speed-up factor in this case is smaller than that observed in the case of four parameters. This is due to the fact that not only we have more parameters but also the reduced model size is now larger. Nonetheless, the savings in the online cost are substantial and offset the offline cost.

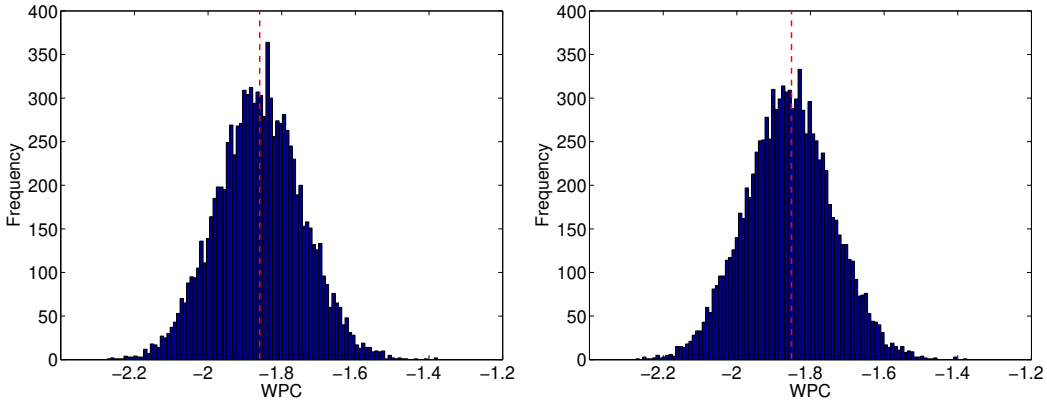


(a) Full WPC for Blade 1

(b) Reduced WPC for Blade 1

Figure 6-5: Comparison of linearized CFD and reduced-order model predictions of work per cycle for Blade 1. MCS results are shown for 10,000 blade geometries with ten parameters. The same geometries are analyzed in each case. Dashed line denotes the mean.

Table 6.3 also compares the statistics of the two distributions. It can be seen from Figure 6-5, Figure 6-6 and Table 6.3 that the reduced-order model predicts the



(a) Full WPC for Blade 2

(b) Reduced WPC for Blade 2

Figure 6-6: Comparison of linearized CFD and reduced-order model predictions of work per cycle for Blade 2. MCS results are shown for 10,000 blade geometries for the case of ten parameters. The same geometries are analyzed in each case. Dashed line denotes the mean.

Table 6.3: Linearized CFD model and reduced-order model MCS results for the case of ten parameters. Work per cycle (WPC) is predicted for blade plunging motion at an interblade phase angle of 180° for 10,000 randomly selected blade geometries.

	CFD	Reduced
Model size	103,008	290
Number of nonzeros	2,846,056	84,100
Offline cost	—	10.92 hours
Online cost	515.61 hours	1.10 hours
Blade 1 WPC mean	-1.8583	-1.8515
Blade 1 WPC variance	0.0503	0.0506
Blade 2 WPC mean	-1.8599	-1.8583
Blade 2 WPC variance	0.0136	0.0138

mean, variance and shape of the distribution of work per cycle accurately. Again, to further verify the quality of the reduced model, we apply the Kolmogorov-Smirnov method, to test whether the reduced work per cycle results and the full work per cycle results are drawn from a same distribution. The results show that we cannot reject the hypothesis that the distribution is the same at a 5% significance level.

It should be also pointed out that compared to the case with four parameters, the variances of work per cycle of the 10-parameter case are larger; however the means are almost the same. This is because as more geometric variability parameters are added, and hence more POD modes (which are variations about the average geometry) are added in (5.14), more variability is added to the blade geometries. It is therefore expected that the variances of work per cycle increase as more geometric variability parameters are included, while the means stay the same.

Next, we compare the model-constrained sampling method with statistically-based sampling methods in the context of snapshot generation for model reduction of the linearized CFD model. In particular, we compare our model-constrained sampling with LHS sampling, logarithmic random sampling, and CVT sampling. Tables 6.4 and 6.5 show the percentage errors in predicting the means and variances of work per cycle of the resulting reduced models for the case with four and ten geometric variability parameters. The percentage error in predicting the quantity h (h can be the mean or the variance of work per cycle) is defined as

$$\text{Percentage error} = \frac{|h_{\text{reduced}} - h_{\text{full}}|}{|h_{\text{full}}|} \times 100\%, \tag{6.3}$$

where h_{full} and h_{reduced} are the corresponding values obtained from the full and the reduced models, respectively. As can be seen, the quality of the resulting reduced models is similar. While the system matrices of the steady thermal fin problem in (4.14)–(4.16) are nonlinear functions of the parameters, the system matrices of the linearized CFD model in (5.22)–(5.23) are linear functions of the geometric variability parameters. As a result, the parameter-to-output relation for the linearized CFD model is expected to be weakly nonlinear, and thus statistically-based sampling

methods may be expected to perform adequately. In particular, for problems that are linear in the state vector, if the parameter-to-output relation is also linear, with one sample point, all methods yield a same reduced model. In addition, as mentioned in Section 6.2, the bound constraints of the parameters in the linearized CFD model are relatively small as compared to those in the steady thermal fin problem. For a small parameter range, unless the problem under consideration is very nonlinear with respect to the parameters in that small parameter range, it is expected that the sampling methods will perform adequately.

Table 6.4: Model-constrained (MC) adaptive sampling method versus logarithmic random (LogRandom) sampling, LHS and CVT for the case with four geometric variability parameters. For all methods, 21 sample points are generated, and the percentage errors in predictions of means and variances of work per cycle (WPC) of the resulting reduced models are shown for 10,000 randomly selected blade geometries.

	MC (%)	LogRandom (%)	LHS (%)	CVT (%)
Blade 1 WPC mean	0.00789	0.01298	0.0231	0.0176
Blade 1 WPC variance	0.2	1.44	0.657	1.544
Blade 2 WPC mean	0.003	0.01299	0.02113	0.0132
Blade 2 WPC variance	0.083	0.16	0.647	0.315

Table 6.5: Model-constrained (MC) adaptive sampling method versus logarithmic random (LogRandom) sampling, LHS, and CVT for the case with ten geometric variability parameters. For all methods, 29 sample points are generated, and the percentage errors in predictions of means and variances of work per cycle (WPC) of the resulting reduced models are shown for 10,000 randomly selected blade geometries.

	MC (%)	LogRandom (%)	LHS (%)	CVT (%)
Blade 1 WPC mean	0.3682	0.0428	0.025	0.029
Blade 1 WPC variance	0.6082	3.4950	0.3211	0.3759
Blade 2 WPC mean	0.0869	0.091	0.0054	0.0209
Blade 2 WPC variance	1.879	4.819	2.435	0.055

Chapter 7

Conclusions and Recommendations

7.1 Thesis Summary and Contributions

In this thesis, we have proposed a model-constrained greedy-based adaptive sampling approach to address the challenge of sampling high-dimensional parameter spaces for model reduction of large-scale systems. The method provides a systematic procedure for sampling the parametric input space, and the forcing input space. In particular, we formulate the sampling problem as an optimization problem that targets an error estimation (which could be an output error indicator, an output error bound, or the true output error) of reduced model output prediction. The optimization problem is defined by introducing as constraints the systems of equations representing the reduced model (and possibly the full model if the true output error is targeted). The optimization formulation treats the parameter space as continuous; that is, we do not require *a priori* selection of a discrete training parameter set. Further, since any error estimation can be used as our selection criteria, our approach is applicable in cases for which output error bounds are unavailable. Finally, we use a state-of-the-art optimization technique, namely the subspace trust-region interior reflective inexact Newton-CG method, to solve the resulting greedy PDE-constrained optimization problem.

In principle, one can treat the forcing input as another parameter, albeit a special parameter, and the adaptive sampling method can be applied for both parametric

input space and unsteady forcing space.

We have also provided an analysis of the model-constrained adaptive sampling approaches with two key results as follows. First, the greedy optimization problem is well-posed in the sense that there exists at least one solution. Second, if the full model is linear in the state vector, then the adaptive sampling approach will never sample at the previous sampled points in the parameter space.

The model-constrained adaptive sampling approach has been applied to a steady thermal fin optimal design problem and to probabilistic analysis of geometric variability in a turbomachinery application. While these two examples are both linear in the state vector, our sampling approach could also be applied to general nonlinear problems. In the nonlinear case, efficiency of the reduced model could be addressed, for example, using the interpolation method of [61,62,68]. In the following, we summarize the results and the conclusions from numerical studies of these two examples.

The model-constrained adaptive reduction method has been able to create reduced models that accurately represent the full models over a wide range of parameter values in high-dimensional parametric spaces. Since the thermal fin problem is quite small, the offline cost of generating the reduced model is not offset by the savings in online cost even though using the reduced model in optimal design is an order of magnitude faster than using the full model. The trade-off between the online and offline cost has been seen to be much more dramatic for probabilistic analysis problems in which the savings in the online cost are substantial and offset the offline cost. In particular, for probabilistic analysis of a subsonic blade row with ten geometric variability parameters, the reduced model provided a factor of 468 speed-up for a Monte Carlo simulation of 10,000 samples.

Either the true error and the error indicator can be used as sampling selection in the model-constrained adaptive sampling. Which method to use is a matter of trading offline cost with reduced model accuracy. For the thermal fin problem, using the error-indicator approach leads to less expensive offline cost, but the resulting reduced model can be less accurate than that obtained using the true-error approach with the same basis size. However, both steady and unsteady results have shown

that, using the error indicator, in particular the residual, one can obtain reduced models that are accurate over a wide range of parameter values in multi-dimensional parametric input spaces.

Both steady and unsteady results numerically show that the subspace trust-region interior reflective inexact Newton CG solver is efficient for the greedy optimization problem in the sense that the number of Newton steps scales as $\mathcal{O}(d)$, where d is the number of parameters. In addition, one can also theoretically show that if a good preconditioner for the CG solver is available so that the number of Hessian-vector products is independent of the number of the parameters, the offline cost for each greedy cycle scales linearly with the dimension of the parameter space, d. In addition, if the number of greedy cycles is fixed, i.e. the size of the reduced model is given, then one can prove that the total offline cost of constructing the reduced basis scales linearly with the number of parameters. As a result, the model-constrained adaptive sampling approach can be used for problems that depend on a large number of parameters.

In this thesis, various sampling methods are compared in the context of snapshot generation for model reduction. The numerical results for the thermal fin show that with the same number of sampling points, the resulting reduced model from logarithmic random sampling approach is more accurate than those obtained from LHS and CVT sampling methods. Compared to these methods, the model-constrained sampling yields a reduced model with error of several orders of magnitude smaller than those obtained using logarithmic random sampling, uniform random sampling, LHS and CVT sampling methods. As the size of the parametric input space increases, the difference in accuracy of the reduced model using model-constrained sampling method and those of other methods is larger. This reflects the fact that the model-constrained sampling is a model-based sampling method; that is, the parameter is sampled where the indicator of the error between the full and the reduced models is locally largest, whereas the other statistically-based sampling methods use no knowledge of the underlying model when selecting their sample points. As the dimension of the parametric space increases, it becomes more and more difficult to adequately cover

the space with a reasonable number of samples using the statistically-based sampling methods. However, if the problems under consideration are weakly nonlinear in the parameters or the parameter ranges are small, the results of the linearized CFD model show that the model-constrained sampling approach and the other statistically-based sampling methods yield sample points of similar quality in the context of snapshot generation for model reduction.

7.2 Extensions and Future Work

There are a number of extensions that could be applied to the model-constrained greedy-based adaptive sampling approach in this research. These extensions include improvements to the adaptive sampling method, possible treatments for its limitations, and possibilities for additional model applications. In the following, we present some extensions and future work.

We have used a "black-box" approach (also called the reduced-space approach) to solve the greedy optimization problem. That is, the states are eliminated by solving the PDE constraints exactly (up to machine zero), and hence the PDE constraints are not visible to the optimizer. As a result, feasibility is always guaranteed and the black-box approach only moves towards optimality. The all-at-once approach (sometimes called the full-space approach) is more efficient in the sense that it moves towards optimality and feasibility at the same time; hence it does not require the possibly very expensive solution of the PDE constraints, especially for problems that are nonlinear in the state vector.

It should be emphasized again that while we have mainly presented both theoretical and numerical results for problems that are linear in the state vector, the model-constrained sampling approach is applicable for problems that are nonlinear in the state vector as well. Therefore, if a numerical solver is available for the nonlinear problem under consideration—in particular, given a parameter set as the input, the output of the solver is the state solution corresponding to that input set—it is straightforward to apply the adaptive sampling approach using the framework proposed in

this thesis. However, the main assumption in Corollary 3.2 is that the problem under consideration is linear in the state vector so that the *a priori* result of the form stated in Theorem 2.2 holds. Note that if *a priori* result of this form is also available for a problem that is nonlinear in the state vector, Corollary 3.2 still holds. This is, however, not a severe limitation if, for example, one uses the empirical interpolation method [61, 62, 68] and its recent extensions [69, 114, 115] to pre-process the nonlinear terms by a linear combination of empirical interpolation basis functions. In that case, the results in Corollary 3.2 still apply for the pre-processed model. It should be pointed out that while the model-constrained sampling approach addresses the question on how to construct a reduced basis that is valid over wide ranges of parameter values in multi-dimensional parametric input space, the empirical interpolation approach addresses the question on how to construct an efficient reduced model given a reduced basis. Therefore, the combination of the model-constrained approach and the empirical interpolation method is a promising model reduction method that yields accurate and efficient reduced models for a general nonlinear problem that depends on a large number of parameters. An example in which this combination could be used is the probabilistic analysis. For probabilistic analysis applications in this research, we have limited ourselves to a linearized model in both states and parameters so that efficient reduced models are available for the online stage. Therefore the model is only valid for a small perturbation in states and parameters about the linearization point. An improved model is therefore to combine the adaptive sampling method and the empirical interpolation method so that the reduced model is applicable for wider ranges of geometric variations.

Bibliography

[1] R.H. Landon. Compendium of unsteady aerodynamic measurements. Technical report, AGARD Report No. 702, August 1982.

[2] K.J. Bathe. *Finite Element Procedures*. Prentice-Hall, New Jersey, 1996.

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

[4] L. Sirovich. Turbulence and the dynamics of coherent structures. Part 1: Coherent structures. *Quarterly of Applied Mathematics*, 45(3):561–571, October 1987.

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

[6] B.C. Moore. Principal component analysis in linear systems: controllability, observability, and model reduction. *IEEE Transactions on Automatic Control*, AC-26(1):17–31, August 1981.

[7] 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.

[8] 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*, December 1994.

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

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

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

[12] K. Ito and S. S. Ravindran. Reduced basis method for optimal control of unsteady viscous flows. *International J. of Computational Fluid Dynamics*, 15:97–113, 2001.

[13] K.T. Tang, W.R. Graham, and J. Peraire. Active flow control using a reduced order model and optimum control. *Fluid Dynamics Conference, 27th, New Orleans, LA, June 17-20, AIAA Paper 1996-1946*, 1996.

[14] M. P. Castanier, G. Ottarsson, and C. Pierre. A Reduced Order Modeling Technique for Mistuned Bladed Disks. *Journal of Vibration and Acoustics*, 119(3):439–447, 1997.

[15] M.-T. Yang and J. H. Griffin. A Reduced Order Model of Mistuning Using a Subset of Nominal Modes. *Journal of Engineering for Gas Turbines and Power*, 123(4):893–900, 2001.

[16] E. Petrov and D. Ewins. Analysis of the Worst Mistuning Patterns in Bladed Disc Assemblies. In *Proceedings of ASME Turbo Expo 2001 , New Orleans, Lousiana*, June 2001.

[17] D.M. Feiner and J. H. Griffin. A Fundamental Model of Mistuning for a Single Family of Modes. *Journal of Turbomachinery*, 124(4):597–605, 2002.

[18] D.M. Feiner and J. H. Griffin. Mistuning Identification of Bladed Disks Using a Fundamental Mistuning Model – Part I: Theory. *ASME Journal of Turbomachinery*, 126(1):150–158, 2003.

[19] D.M. Feiner and J. H. Griffin. Mistuning Identification of Bladed Disks Using a Fundamental Mistuning Model – Part II: Application. *ASME Journal of Turbomachinery*, 126(1):159–165, 2003.

[20] L. Daniel and O.C. Siong and L.S. Chay and K.H. Lee and J. White. Geometrically parameterized interconnect performance models for interconnect synthesis. In *Proceedings of the 2002 International Symposium on Physical Design*, pages 202–207, 2002.

[21] L. Daniel, O.C. Siong, L.S. Chay, K.H. Lee, and J. White. Multiparameter Moment Matching Model Reduction Approach for Generating Geometrically Parameterized Interconnect Performance Models. *Transactions on Computer Aided Design of Integrated Circuits*, 23(5):678–693, May 2004.

[22] B. Bond and L. Daniel. Parameterized Model Order Reduction of Nonlinear Dynamical Systems. In *Proceedings of the IEEE Conference on Computer-Aided Design, San Jose*, November 2005.

[23] J. Elliot and J. Peraire. Practical 3D aerodynamic design and optimization using unstructured meshes. *AIAA J.*, 35(9), 1997.

[24] S. S. Collis, K. Ghayour, M. Heinkenschloss, M. Ulbrich, and S. Ulbrich. Optimal control of unsteady compressible viscous flows. *Int. J. Numer. Meth. Fluids*, 40(11):1401–1429, 2002.

[25] V.E. Garzon and D.L. Darmofal. Impact of geometric variability on axial compressor performance. *ASME Journal of Turbomachinery*, 125(4):692–703, 2003.

[26] V. Akcelik, J. Bielak, G. Biros, I. Epanomeritakis, A. Fernandez, O. Ghattas, E.J. Kim, J. Lopez, D. O'Hallaron, T. Tu, and J. Urbanic. High-resolution

forward and inverse earthquake modeling on terascale computers. In *Proceedings of ACM/IEEE SC2003, Phoenix, AZ*, 2003.

[27] M. Hinze and S. Volkwein. Proper orthogonal decomposition surrogate models for nonlinear dynamical systems: Error estimates and suboptimal control. In P. Benner, V. Mehrmann, and D. Sorensen, editors, *Dimension Reduction of Large-Scale Systems*, Lecture Notes in Computational and Applied Mathematics, pages 261–306, 2005.

[28] K. Kunisch and S. Volkwein. Control of Burgers' equation by reduced order approach using proper orthogonal decomposition. *Journal of Optimization Theory and Applications*, 102:345–371, 1999.

[29] V.M. Adamjan, D.Z. Arov, and M.G. Krein. Analytic Properties of Schmidt Pairs for a Hankel Operator and the Generalized Schur-Takagi Problem. *Math. USSR Sbornik*, 15:31–73, 1971.

[30] M. Bettayeb, L.M. Silverman, and M.G. Safonov. Optimal Approximation of Continuous-Time Systems. *Proceedings of the 19th IEEE Conference on Decision and Control*, 1, December 1980.

[31] S-Y. Kung and D.W. Lin. Optimal Hankel-Norm Model Reductions: Multivariable Systems. *IEEE Transactions on Automatic Control*, AC-26(1):832–52, August 1981.

[32] D.C. Sorensen and A.C. Antoulas. The Sylvester equation and approximate balanced reduction. *Linear Algebra and its Applications*, 351–352:671–700, 2002.

[33] J. Li and J. White. Low rank solution of Lyapunov equations. *SIAM Journal on Matrix Analysis and Applications*, 24(1):260–280, 2002.

[34] S. Gugercin and A. Antoulas. A survey of model reduction by balanced truncation and some new results. *International Journal of Control*, 77:748–766, 2004.

[35] T. Penzl. Algorithms for model reduction of large dynamical systems. *Linear Algebra and its Applications*, 415(2–3):322–343, 2006.

[36] G.D. Wood, P.J. Goddard, and K. Glover. Approximation of linear parameter-varying systems. In *Proceedings of the 35th IEEE Conference on Decision and Control*, volume 1, pages 406–411, 1996.

[37] P.J. Goddard. *Performance-Preserving Controller Approximation*. PhD thesis, University of Cambridge, UK, 1995.

[38] S. Shokoohi and L. Silverman and P. Van Dooren. Linear time-variable systems: balancing and model reduction. *IEEE Transactions on Automatic Control*, AC-28:810–822, August 1983.

[39] A.E. Deane, I.G. Kevrekidis, G.E. Karniadakis, and S.A. Orszag. Low-dimensional models for complex geometry flows: Application to grooved channels and circular cylinders. *Phys. Fluids*, 3(10):2337–2354, 1991.

[40] A. J. Newman. *Modeling and Reduction with Application to Semiconductor Processing*. PhD thesis, University of Maryland, 1999.

[41] M.C. Romanowski. Reduced Order Unsteady Aerodynamic and Aeroelastic Models using Karhunen-Loéve Eigenmodes. *AIAA Paper 96-194, AIAA, NASA, and ISSMO, Symposium on Multidisciplinary Analysis and Optimization, 6th, Bellevue, WA, Technical Papers. Pt. 1; UNITED STATES; 4-6 Sept*, 1996.

[42] E.H. Dowell and K.C. Hall. Modeling of fluid-structure interaction. *Annual Review of Fluid Mechanics*, 33:445–90, 2001.

[43] 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, 1979.

[44] Q. Du, V. Faber, and M. Gunzburger. Centroidal voronoi tessellations: applications and algorithms. *SIAM Review*, 41:637–676, 1999.

[45] V.J. Romero, J.S. Burkardt, M.D. Gunzburger, J.S. Peterson, and T. Krishnamurthy. Initial Application and Evaluation of a Promising New Sampling Method for Response Surface Generation: Centroidal Voronoi Tessellation. *AIAA Paper 2003-2008, 44th Structures, Structural Dynamics, and Materials Conference (5th AIAA Non-Deterministic Approaches Forum) Norfolk, VA*, 2003.

[46] V.J. Romero, J.S. Burkardt, M.D. Gunzburger, and J.S. Peterson. Initial Evaluation of Centroidal Voronoi Tessellation Method for Statistical Sampling and Function Integration. Technical report, Sandia National Laboratories library archive SAND2003-3672C, an extended version (with comparison of exceedence probabilities) of the paper in the proceedings of the 4th International Symposium on Uncertainty Modeling and Analysis (ISUMA 03), Sept. 21-24, 2003, University of Maryland, College Park, MD, 2003.

[47] V.J. Romero, J.S. Burkardt, M.D. Gunzburger, and J.S. Peterson. Initial Evaluation of Pure and Latinized Centroidal Voronoi Tessellation for Non-Uniform Statistical Sampling. In Kenneth M. Hanson and Francois M. Hemez, editors, *Sensitivity Analysis of Model Output*, Los Alamos National Laboratory, 2005; http://library.lanl.gov/, pages 380–401, 2005.

[48] K. Afanasiev and M. Hinze. Adaptive control of a wake flow using proper orthogonal decomposition. In *Lecture Notes in Pure and Applied Mathematics*, volume 216, pages 317–332. Marcel Dekker, 2001.

[49] E. Arian, M. Fahl, and E.W. Sachs. Trust-region proper orthogonal decomposition for optimal flow control. Technical Report ICASE 2000-25, Institute for Computer Applications in Science and Engineering, May 2000.

[50] B.I. Epureanu, E.H. Dowell, and K.C Hall. A parametric analysis of reduced order models of potential flows in turbomachinery using proper orthogonal de-

composition. In *Proceedings of ASME TURBO EXPO 2001, New Orleans, Louisiana, 2001-GT-0434*, June 2001.

[51] T. Bui-Thanh. Proper Orthogonal Decomposition Extensions and their Applications in Steady Aerodynamics. Master's thesis, Singapore -MIT Alliance, June 2003.

[52] H.V. Ly and H.T. Tran. Modeling and Control of Physical Processes using Proper Orthogonal Decomposition. *Journal of Mathematical and Computer Modeling*, February 1999.

[53] M.D. Gunzburger, J.S. Peterson, and J.N. Shadid. Reduced-order modeling of time-dependent PDEs with multiple parameters in the boundary data. *Comput. Methods Appl. Mech. Engrg.*, 196:1030–1047, 2007.

[54] S. Gugercin, A. Antoulas, and C. Beattie. A rational Krylov iteration for optimal H2 model reduction. In *Proceedings of MTNS 2006, Japan*, 2006.

[55] C. Prud'homme, D.V. Rovas, K. Veroy, L. Machiels, Y. Maday, A.T. Patera, and G. Turinici. Reliable real-time solution of parametrized partial differential equations: Reduced-basis output bound methods. *J Fluids Engineering*, 124:70–84, 2002.

[56] 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. In *Proceedings of the 16th AIAA Computational Fluid Dynamics Conference, AIAA Paper 2003-3847*, 2003.

[57] L.M. Silveira and J.R. Phillips. Resampling plans for sample point selection in multipoint model order reduction. *IEEE transactions on computer-adied design of integrated circuits and systems*, 25(12):2775–2783, December 2006.

[58] M.A. Grepl and A.T. Patera. A posteriori error bounds for reduced-basis approximations of parametrized parabolic partial differential equations. *ESAIM-Mathematical Modelling and Numerical Analysis (M2AN)*, 39(1):157–181, 2005.

[59] 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.

[60] K. Veroy. *Reduced-Basis Methods Applied to Problems in Elasticity: Analysis and Applications*. PhD thesis, Massachusetts Institute of Technology, June 2003.

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

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

[63] K. Veroy and A. 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, 2005.

[64] C. W. Rowley, T. Colonius, and R. M. Murray. Model reduction for compressible flows using POD and Galerkin projection. *Physica D. Nonlinear Phenomena*, 189(1-2):115–129, 2004.

[65] Y. Chen. Model Order Reduction for Nonlinear Systems. Master's thesis, MIT, Cambridge, MA, 1999.

[66] M. Rewienski and J. White. A Trajectory Piecewise-Linear Approach to Model Order Reduction and Fast Simulation of Nonlinear Circuits and Micromachined Devices. *ICCAD, 2001 International Conference on Computer-Aided Design (ICCAD '01)*, pages 252–257, 2001.

[67] D. Gratton and K. Willcox. Reduced-order, trajectory piecewise-linear models for nonlinear computational fluid dynamics. *34th AIAA Fluid Dynamics Conference and Exhibit, Portland, Oregon, June 28-1, AIAA Paper 2004-2329*, 2004.

[68] 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. *CR Acad Sci Paris Series.*, I:339–667, 2004.

[69] N. C. Nguyen and J. Peraire. An efficient reduced-order approach for nonaffine and nonlinear partial differential equations. Technical report, Computational Engineering, MIT, 2007.

[70] I.B. Oliveira and A.T. Patera. Reduced-basis techniques for rapid reliable optimization of systems described by affinely parametrized coercive elliptic partial differential equations. *Optimization and Engineering*, 2007. To appear.

[71] M. Fahl. *Trust-region methods for flow control based on Reduced Order Modeling*. PhD thesis, Trier University, 2000.

[72] E. Arian, M. Fahl, and Sachs E. W. Trust-Region Proper Orthogonal Decomposition for Flow Control. Technical report, ICASE report 2000-25.

[73] K. Kunisch and S. Volkwein. Proper Orthogonal Decomposition for Optimality Systems (OS-POD). *Adaptive Model Reduction Methods for PDE Constrained Optimization, Rice University 17-19, 2006.*

[74] D. Vasilyev, M. Rewienski, and J. White. A TBR-based trajectory piecewise-linear algorithm for generating accurate low-order models for nonlinear analog circuits and MEMS. *Design Automation Conference, 2003. Proceedings , June 2-6*, pages 490 –495, 2003.

[75] K. Willcox. *Reduced-Order Aerodynamic Models for Aeroelastic Control of Turbomachines*. PhD thesis, MIT, 2000.

[76] J.P. Thomas, E.H. Dowell, and K.C. Hall. Three-Dimensional Transoic Aeroelasticity Using Proper Orthogonal Decomposition Based Reduced Order Models. *AIAA Paper 2001-1526, 42nd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials (SDM) Conference, Seattle, WA, April 16–19*, 2001.

[77] E.H. Dowell, J.P. Thomas, and K.C. Hall. Transonic Limit Cycle Oscillation Analysis Using Reduced Order Modal Aerodynamics Models. *Journal of Fluids and Structures*, 19(1):17–27, January 2004.

[78] T. Lieu, C. Farhat, and M. Lesoinne. POD-based Aeroelastic Analysis of a Complete F-16 Configuration: ROM Adaptation and Demonstration. *AIAA Paper 2005-2295 46th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference , Austin, Texas, Apr. 18-21*, 2005.

[79] L.N. Trefethen and D. Bau III. *Numerical Linear Algebra*. SIAM, Philadelphia, 1997.

[80] P. Bochev and M. Gunzburger. Finite element methods of least squares type. *SIAM Review*, 40:789–837, 1998.

[81] B.N. Jiang. *The Least-Squares Finite Element Method: Theory and Applications in Computational Fluid Dynamics and Electromagnetics*. Springer, New York, 1998.

[82] L.Q. Tang and T.T.H. Tsang. A least-squares finite element method for time-dependent incompressible flows with thermal convection. *International Journal for Numerical Methods in Fluids*, 17(4):271–289, 1993.

[83] A. Barrett and G. Reddien. On the reduced basis method. *Z. Angew. Math. Mech.*, 75(7):543–549, 1995.

[84] D. Rovas. *Reduced-basis output bound methods for parametrized partial differential equations*. PhD thesis, Massachusetts Institute of Technology, 2003.

[85] G. Rozza and K. Veroy. On the stability of the reduced basis method for stokes equations in parametrized domains. *submitted to Elsevier Science*, 2006.

[86] A.C. Antoulas. *Approximation of large-scale dynamical systems.* SIAM, Philadelphia, 2005.

[87] G.H. Golub and C.F. Van Loan. *Matrix Computations.* The Johns Hopkins University Press, third edition, 1996.

[88] A.T. Patera Y. Maday and D.V. Rovas. A blackbox reduced-basis output bound method for noncoercive linear problems. In D. Cioranescu and J.L. Lions, editors, *Studies in Mathematics and its applications*, number 31. 2002.

[89] S.C. Brenner and L.R. Scott. *The Mathematical Theory of Finite Elements Methods.* Springer, New York, second edition, 2002.

[90] P.J. Psarrakos and M.J. Tsatsomeros. Numerical range: (in) a matrix nutshell. http://www.math.wsu.edu/faculty/tsat/files/short.ps.

[91] M.D. Gunzburger. *Perspectives in Flow Control and Optimization.* SIAM, Philadelphia, 2002.

[92] C.J. Lin and J.J. More. Newton's method for large bound-constrained optimization problems. *SIAM J. Optim.*, 9(4):1100–1127, 1999.

[93] F. Facchinei, S. Lucidi, and L. Palagi. A truncated Newton algorithm for large scale box constrained optimization. *SIAM J. Optim.*, 12(4):1100–1125, 2002.

[94] C. Zhu, R. Byrd, P. Lu, and J. Nocedal. L-BFGS-B: Fortran subroutines for large-scale bound constrained optimization, 1994, citeseer.ist.psu.edu/zhu94lbfgsb.html.

[95] M. Heinkenschloss, M. Ulbrich, and S. Ulbrich. Superlinear and quadratic convergence of affine-scaling interior-point Newton methods for problems with simple bounds without strict complementarity assumption. *Math Programming, Ser. A*, 86:615–635, 1999.

[96] J.E. Dennis, M. Heinkenschloss, and L.N. Vicente. Trust-region interior-point SQP algorithm for a class of nonlinear programming problems. *SIAM J. Control Optim.*, 36(5):1750–1794, 1998.

[97] J.E. Dennis and L.N. Vicente. Trust-region interior-point algorithms for minimization problems with simple bounds. Technical Report TR94-42, Rice University, 1995.

[98] T.F. Coleman and Y. Li. On the convergence of reflective newton methods for large-scale nonlinear minimization subject to bounds. *Math Programming,*, 67:189–224, 1994.

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

[100] C. Kanzow and A. Klug. On affine-scaling interior-point Newton methods for nonlinear minimization with bound constraints. *Computationtal optimization and applications*, 35(2):177–297, 2006.

[101] M.A Branch, T.F. Coleman, and Y. Li. A subspace, interior, and conjugate gradient method for large-scale bound-constrained minimization problems. *SIAM J. Sci. Comput.*, 21(1):1–23, 1999.

[102] S. Eisenstat and H. Walker. Choosing the forcing terms in an inexact Newton method. *SIAM Journal on Scientific Computing*, 17:16–32, 1996.

[103] T. Steihaug. The conjugate gradient method and trust regions in large scale optimization. *SIAM J. Numer. Anal.*, 20:626–637, 1983.

[104] J. Nocedal and S. Wright. *Numerical Optimization*. Springer, New York, 1999.

[105] T. A. Davis. A column pre-ordering strategy for the unsymmetric-pattern multifrontal method. *ACM Transactions on Mathematical Software*, 30(2):165–195, June 2004.

[106] T. A. Davis. Algorithm 832: UMFPACK, an unsymmetric-pattern multifrontal method. *ACM Transactions on Mathematical Software*, 30(2):196–199, June 2004.

[107] T. A. Davis and I. S. Duff. A combined unifrontal/multifrontal method for unsymmetric sparse matrices. *ACM Transactions on Mathematical Software*, 25(1):1–19, March 1999.

[108] T. A. Davis and I. S. Duff. A combined unifrontal/multifrontal method for unsymmetric sparse matrices. *SIAM Journal on Matrix Analysis and Applications*, 18(1):140–158, Jan 1997.

[109] J.R. Gilbert. Computer science 290H, Fall 2004: Sparse matrix algorithms, 2004. http://www.cs.ucsb.edu/ gilbert/cs290hFall2004/.

[110] P.O. Persson. MIT 18.335: Introduction to numerical methods (Fall 2006), 2006. http://www-math.mit.edu/ persson/18.335/.

[111] Y. Saad. *Iterative Methods for Sparse Linear Systems*. SIAM, Philadelphia, second edition, 2003.

[112] T. Bui-thanh, K. Willcox, O. Ghattas, and B. van Bloemen Waanders. Goal-oriented, model-constrained optimization for reduction of large-scale systems. *Journal of Computational Physcis*, 2007. In press.

[113] W. Rudin. *Principles of Mathematical Analysis*. McGraw-Hill, New York, third edition, 1976.

[114] N. C. Nguyen M. A. Grepl, Y. Maday and A. T. Patera. Efficient reduced-basis treatment of nonaffine and nonlinear partial differential equations. *M2AN (Math. Model. Numer. Anal.)*, 2007. To appear.

[115] N. C. Nguyen Y. Maday A. T. Patera E. Cancs, C. LeBris and G. S. H. Pau. Feasibility and competitiveness of a reduced basis approach for rapid electronic structure calculations in quantum chemistry. In *Proceedings of the Workshop*

for High-dimensional Partial Differential Equations in Science and Engineering (Montreal), 2005. Submitted.

[116] MIT OpenCourseWare. Numerical methods for partial differential equations, 2003. http://ocw.mit.edu/OcwWeb/Aeronautics-and-Astronautics/16-920JNumerical-Methods-for-Partial-Differential-EquationsSpring2003/CourseHome/.

[117] D.L. Darmofal and R. Haimes. Towards the next generation in CFD. *AIAA Paper 2005-0087, presented at the 43rd AIAA Aerospace Sciences Meeting and Exhibit, Reno, NV January 10–13*, 2005.

[118] B. Cockburn and C.-W. Shu. Runge-kutta discontinuous galerkin methods for convection-dominated problems. *Journal of Scientific Computing*, 16(3):173–261, 2001.

[119] K.J. Fidkowski and D.L. Darmofal. Development of a higher-order solver for aerodynamic applications. *AIAA Paper 2004-0436, presented at the 42nd AIAA Aerospace Sciences Meeting and Exhibit, Reno, NV January 5–8*, 2004.

[120] P.L. Roe. Approximate Riemann solvers, parametric vectors, and difference schemes. *Journal of Computational Physics*, 43:357–372, 1981.

[121] K.C. Hall. *A Linearized Euler Analysis of Unsteady Flows in Turbomachinery.* PhD thesis, Dept. of Aeronautics and Astronautics, MIT, May 1987.

[122] A. Bolcs and T.H. Fransson. Aeroelasticity in Turbomachines; Comparison of Theoretical and Experimental Cascade Results. Technical Report AFOSR-TR-870605, Ecole Polytechnique Federale de Lausanne, 1986.

[123] H. Atassi and T.J. Akai. Aerodynamic and aeroelastic characteristics of oscillating loaded cascades at low Mach number. *Trans. ASME, J.Eng. Power*, 102:344–351, 1980.

[124] J.M. Verdon and W.J. Usab. Application of a linearized unsteady aerodynamic analysis to standard cascade configurations. Technical report, NASA contractor report, August 1985.

[125] T.H. Fransson and J.M. Verdon. Standard Configurations for Unsteady Flow Through Vibrating Axial-Flow Turbomachine-Cascades. Technical Report, Updated version, http://www.energy.kth.se/proj/projects/Markus%20Joecker/STCF/default.htm, July 1991.

[126] S.H. Lim, M.P. Castanier, and C. Pierre. Vibration Modeling of Bladed Disks Subject to Geometric Mistuning and Design Changes. *AIAA Paper 2004-1686, 45th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference*, 2004.

[127] V.E. Garzon and D.L. Darmofal. Impact of geometric variability on axial compressor performance. *Journal of Turbomachinery*, 125(4):692–703, October 2003.

[128] V.E. Garzon and D.L. Darmofal. On the aerodynamic design of compressor airfoils for robustness under geometric uncertainty. In *Proceedings of 2004 ASME Turbo Expo, Vienna Austria, 2004-53581*, 2004.

[129] V.E. Garzon. *Probabilistic Aerothermal Design of Compressor Airfoils*. PhD thesis, MIT, Cambridge, MA, February 2003.

[130] J.M. Brown, J. Slater, and R.V. Grandhi. Probabilistic Analysis of Geometric Uncertainty Effects of Blade Modal Response. *ASME GT2003-38557, Proceedings of ASME Turbo Expo, Atlanta, GA*, June 2003.

[131] D.M. Ghiocel. Stochastic Field Models for Aircraft Jet Engine Applications. *ASCE Journal of Aerospace Engineering*, 14(4), 2001.

[132] A. Sinha, B. Hall, B. Cassenti, and G. Hilbert. Vibratory Parameters of Blades from Coordinate Measurement Machine (CMM) Data. In *Proceedings of 9th*

National Turbine Engine High Cycle Fatigue (HCF) Conference, Pinehurst, North Carolina, 16-19 March 2004.

[133] R.G. Ghanem and P.D. Spanos. *Stochastic Finite Elements: A Spectral Approach*. Springer-Verlag, 1991.

[134] M.H. DeGroot. *Probability and Statistics*. Addison-Wesley, MA, 1991.