ADAPTIVE MODEL REDUCTION TO ACCELERATE OPTIMIZATION PROBLEMS GOVERNED BY PARTIAL DIFFERENTIAL EQUATIONS

A DISSERTATION
SUBMITTED TO THE INSTITUTE FOR
COMPUTATIONAL AND MATHEMATICAL ENGINEERING
AND THE COMMITTEE ON GRADUATE STUDIES
OF STANFORD UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Matthew Joseph Zahr
August 2016

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

(Charbel Farhat) Principal Adviser

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

(Per-Olof Persson)

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

(Michael Saunders)

Approved for the Stanford University Committee on Graduate Studies

Abstract

Optimization problems constrained by Partial Differential Equations (PDEs) are ubiquitous in modern science and engineering. They play a central role in optimal design and control of multiphysics systems, as well as nondestructive evaluation and detection, and inverse problems. Methods to solve these optimization problems rely on potentially many numerical solutions of the underlying equations. For complicated physical interactions taking place on complex domains, these solutions will be computationally expensive—in terms of both time and resources—to obtain, rendering the optimization procedure difficult or intractable.

This dissertation introduces a globally convergent, error-aware trust region algorithm for leveraging inexpensive approximation models to greatly reduce the cost of solving PDE-constrained optimization problems in increasingly complex scenarios. While the trust region theory is general, in that it is agnostic to the particular form of the approximation model, provided it possesses certain properties, this work employs reduced-order models based on the method of snapshots and Proper Orthogonal Decomposition (POD). The trust region algorithm proceeds by progressively refining the fidelity of the reduced-order model while converging to the optimal solution. Thus, the reduced-order model is trained exactly along the optimization trajectory, circumventing the task of training in a potentially high-dimensional parameter space. The proposed method is shown to find the optimal aerodynamic shape of a full aircraft configuration in about half the time required by accepted methods.

The proposed error-aware trust region algorithm is extended to handle the case where uncertainties are present in the governing equations. In such situations, the goal is to find a design or control that is risk-averse with respect to some quantity of interest. The objective function and constraints in these problems usually correspond to integrals of quantities of interest over the stochastic space, which will inevitably require many solutions of the underlying partial differential equation. For this reason, dimension-adaptive sparse grids are combined with reduced-order models to define an inexpensive approximation model, which is wrapped in the error-aware trust region framework to ensure convergence to the optimal risk-averse solution. This framework is demonstrated on a model problem from computational mechanics and shown to be several orders of magnitude faster than existing methods.

Acknowledgments

First and foremost, I would like to thank my advisor, Professor Charbel Farhat, for his advice and guidance during the past five years. From the time you recruited me out of the AHPCRC Summer Institute as an undergraduate through the completion of my PhD and job search, you have been a professional role model and trusted mentor. I am also grateful for the perfect balance between independence and supervision that you have provided as it has given me the opportunity to explore other areas of computational mathematics and form external collaborations. I am very proud to be able to call myself your student. I also want to thank Professor Per-Olof Persson – my co-author, practicum advisor, thesis reader, coding buddy, and guide into the DG world – for serving so many voluntary roles throughout my PhD. I would also like to thank Professor Sanjay Govindjee and Professor Tarek Zohdi – my trusted mentors since my undergraduate days at UC Berkeley – who have provided crucial support and guidance throughout my PhD and job search. I am also very grateful to rest of my thesis committee – Professor Michael Saunders, Professor Walter Murray, and Professor Louis Durlofsky – and my main source of funding – the Department of Energy Computational Science Graduate Fellowship (DOE CSGF).

I have had the pleasure to be in research lab with many talented and entertaining individuals. I am very thankful to my FRG predecessor Kevin Carlberg who provided invaluable guidance and assistance while I was an AHPCRC undergraduate intern and new graduate student, and remains a trusted mentor and close collaborator to this day. I also want to sincerely thank my FRG labmates – Dr Kyle Washabaugh, Dr Alex Main, Todd Chapman, and Raunak Borker – that made Durand 028 not only an intellectually stimulating place, but also an entertaining and fun one; I will always consider you among my closest friends and collaborators. Finally, I would like to acknowledge the constant support and assistance I received from a number of other FRG-ers: Grace Fontanilla, Tatiana Wilson, William Law, and Dr Philip Avery.

Finally, I would like to thank my family and friends for their love, support, and patience, without which, none of this would be possible. To my sweet dove and soon-to-be wife, Theresa Yates. To my dad, Michael J. Zahr: my closest friend, most trusted mentor, and eternal role model. To my mom, Tamara Bradley: the sweetest, most caring and supportive mother imaginable. To my grandpa and grandma, Robert and Marlene Boranian: you are a constant source of love, encouragement, and support in life and the earliest investors in my education. To my sister, Emily Bradley, and stepfather, Robert Bradley: you are, and always will be, a constant source of enjoyment and entertainment in

my life. To my ski and camping buddies, Michael Gardner and Devon Laduzinsky. To my beloved late uncle John "Jack" Zahr: you are a shining example of success in all phases of life – you will forever be missed and remembered. To my second mother Sharee Eisenga; my soon-to-be in-laws, Michael, Christine, and Rebecca Yates; and the entire Zahr, Hoffmann, and Bradley family.

I dedicate this thesis to my future wife, Theresa Yates; parents, Michael Zahr and Tamara Bradley; grandparents, Robert and Marlene Boranian; step-father, Robert Bradley; sister, Emily Bradley; and my late uncle, John "Jack" Zahr

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation

Optimization problems governed by partial differential equations, or PDE-constrained optimization problems, arise in nearly every branch of engineering and science. The most classical PDE-constrained optimization problems arise in the context of design and control of engineering systems. The solutions of these problems promise to deliver engineering systems with superior performance (in some chosen metric) than otherwise possible, and will have the greatest impact in highly complex situations where intuition breaks down and prototyping and experimentation are expensive, difficult, or dangerous. Topological optimization can lead to lightweight, highly optimized designs intended to operate in volatile multiphysics environments [180], which can be realized using 3D printing or additive manufacturing technology [111, 203]. Topology optimization also promises to have widespread impact in medicine, specifically with regard to medical implants, since optimized, patient-specific implants can be realized [213]. Shape optimization has been used to design aircraft and automobiles with superior aerodynamic performance [164, 163, 123, 55] and reduced environmental and noise [50, 73] impact. Shape optimization has also been used in biological applications, e.g., to design the shape of the incoming branch of the aorto-coronaric bypass [160, 171]. Boundary and volumetric control have been used to drive the state of an engineering system toward some desired using source terms, e.g., diverting heat from a microprocessor using a fan.

PDE-constrained optimization problems also arise in the context of material and initial condition inversion. In material inversion problems, an unknown material distribution must be inferred from the response of a system to known inputs. Nondestructive evaluation [39, 72] seeks to determine the material distribution of a solid object *in situ*, i.e., without extracting a sample and performing laboratory tests, from the response measured from structural and acoustic inputs to detect structural defects prior to operation. A similar PDE-constrained optimization problem underlies the emerging technology of full waveform inversion [195] where the material properties of the Earth's crust are sought in order to detect the location and size of oil reservoirs. In initial condition inversion, the

initial state of a system must be inferred from measurements at later times. An important instance of such a PDE-constrained optimization problem occurs in the determination of the source of a contaminant given its current configuration.

The problems considered to this point have been posed in the *ideal* setting of *certain* knowledge of the data defining the coefficients (material properties) and boundary conditions (loads) of the partial differential equation describing the physical system of interest. This is not realistic as all physical systems, particularly those characterized by a high degree of volatility, are plagued with uncertainties. An important consideration in any discipline in science and engineering is the *robustness* of a particular system with respect to these uncertainties. PDE-constrained optimization problems also arise when attempting to quantify the uncertainty in quantities of interest of PDEs as a result of uncertain data or input. For example, in the Bayesian framework, locating the Maximum A Posteriori (MAP) point is a required step in *importance sampling* [134], i.e., where samples are efficiently drawn from the posterior distribution of the uncertain PDE, and amounts to a PDE-constrained optimization problem. Beyond simply using PDE-constrained optimization to facilitate uncertainty quantification, it is important to incorporate uncertainty quantification *into the optimization problem* to obtain designs and controls that are *risk-averse* with respect to the uncertainties. This leads to *PDE-constrained optimization under uncertainty*—optimization problems constrained by stochastic partial differential equations with objective and constraints defined as *risk* or *hazard* measures of its quantities of interests. These hazard measures usually penalize variance from the mean or rare, catastrophic events.

While the potential benefit of widespread adoption of PDE-constrained optimization in engineering and scientific practice are profound, a number of factors prevent this, most notably the large computational cost, in terms of computing time and resources, associated with these problems. Often, particularly in relevant 3D applications, partial differential equations require a massive discretization to accurately resolve the underlying physics and the solution of the resulting (sequence of) nonlinear equations requires significant computing time on a supercomputer. PDE-constrained optimization problems require potentially many queries to the underlying discretized primal and dual PDE to iteratively progress toward the optimal solution. Since even a single PDE solve constitutes a significant investment in computational resources, the *many-query* setting of PDE-constrained optimization exacerbates this problem and, in some cases, can be prohibitively expensive. This situation is further complicated from the presence of uncertainty that exists in every physical setting, particularly those with a high degree of volatility. In reality, the boundary conditions, material properties, and sources terms of a system are not known with certainty and cannot be modeled as such if one wishes to discover solutions that are *robust* with respect to these uncertainties. Depending on the number of stochastic parameters incorporated into the PDE, the quantification of uncertainty in an optimization problem will increase the computational cost by potentially many orders of magnitude. This effectively makes these problems infeasible for all except the smallest academic problems.

1.2 Strategy and Objectives

The primary objective of this thesis is to develop a series of optimization methods to solve large-scale deterministic and stochastic PDE-constrained optimization problems that can largely circumvent the prohibitive cost of repeatedly running computational physics simulations *without compromising the quality of the resulting optimum.* Focus will be placed on methods that apply to complex, nonlinear partial differential equations that *do not possess* significant structure, i.e., linearity, ellipticity, and coercivity, that can be used to develop inexpensive, computable error bounds. The strategy taken to accomplish this objective is modular in the sense that two independent technologies will be developed and later combined and specialized to the context of deterministic and stochastic PDE-constrained optimization. The two foundational technologies that are developed for this purpose are: (1) a globally convergent, generalized trust region method for the management of approximation models in the context of optimization and (2) minimum-residual, projection-based reduced-order and hyperreduced models as low-dimensional approximations of the discretized PDE.

In the context of deterministic PDE-constrained optimization, these technologies, along with the concept of a partially converged PDE solution, will be combined to yield an efficient, globally convergent optimization procedure. In the context of stochastic PDE-constrained optimization, projection-based reduced-order models and dimension-adaptive sparse grids will define an efficient approximation model based on two levels of inexactness. This two-level approximation will be nested in the generalized trust region method to produce an efficient, globally convergent optimization procedure.

1.3 Literature Review

This work builds on several foundational technologies including PDE-constrained optimization, trust region methods, projection-based model reduction, and surrogate methods for PDE-constrained optimization. This section presents a brief literature review of each technology and outlines the contributions of this thesis to the state-of-the-art.

1.3.1 PDE-Constrained Optimization

This work is primarily concerned with the efficient solution of optimization problems governed by partial differential equations, in both the deterministic and stochastic setting. This section provides a brief literature review of deterministic and stochastic PDE-constrained optimization with an extensive mathematical formulation of PDE-constrained optimization problems provided in Chapter 2.

PDE-constrained optimization has been extensively studied in the case where the underlying partial differential equation is deterministic. A thorough review of the topic is provided in the references [78, 96]. The PDE-constrained optimization problem is naturally posed in a continuous setting [100, 135], that is, the PDE itself is a constraint of the optimization problem and the objective function and "side" constraints are defined by integrating the PDE solution over (portions)

of the spatio-temporal domain. The corresponding optimality conditions are a system of partial differential equations that must be discretized to be solvable in a computational setting. A more common and practical approach, particularly in large-scale implementations, defines an optimization problem constrained by the *discretized* PDE, resulting in an optimality system consisting of a system of discrete nonlinear equations [138, 187, 130, 140]. Once the optimization setting has been chosen, the optimization problem can be solved using a full space [147, 69, 91, 116, 2] or reduced space [100, 197, 138, 108, 109, 210] approach. The reduced space approach uses a PDE solver to eliminate the PDE constraint from the optimization problem while the full space approach considers the PDE as a constraint and directly solves the complete optimization problem. If the reduced space method is employed and a gradient-based optimizer is used, the sensitivity [80, 138, 129, 127] or adjoint [100, 197, 187, 130, 123] method are required to compute the required gradients of the optimization functionals. The relative efficiency of these methods depends on the number of optimization variables and constraints: the sensitivity method is more efficient if there are more constraints than variables and the adjoint method is more efficient in the opposite case. These concepts regarding the continuous versus discrete formulation, full space versus reduced space approach, and sensitivity versus adjoint method for computing gradients apply whether the partial differential equation under consideration is static or transient [88, 116, 136, 124, 205, 140, 55, 132, 211, 212]. The case where the PDE is unsteady represents a significant increase in computational expense as there are substantially more optimization variables in the full space approach (one for each spatial degree of freedom at each timestep) or the full transient PDE must be resolved at each optimization iteration in the reduced space approach. This work will solely consider the *discrete* formulation of the PDE-constrained optimization problem, which will be solved using the reduced space approach. Both the sensitivity and adjoint methods will be used to compute gradients of quantities of interest.

In addition to the many considerations involved in the formulation and solution of deterministic PDE-constrained optimization problems, the case where the underlying PDE depends on random data [13, 141, 142, 199, 204, 12] involves an additional component—treatment of the stochastic variables. Stochastic Galerkin [13] and collocation [12] are popular techniques for discretizing the stochastic space associated with the PDE. This work will solely consider the non-intrusive approach of stochastic collocation [24, 23, 22, 178, 188, 107] and the collocation nodes will be defined using sparse grids [184, 144, 66, 145, 156, 18, 157, 67, 29, 146]. While there has been work considering *random optimization variables* [24, 30], this work will consider the optimization variables to be *deterministic* quantities, with the data underlying the PDE (boundary conditions, coefficients) as uncertain. These instances of PDE-constrained optimization problems under uncertainty can be many orders of magnitude more expensive than the deterministic counterpart since the PDE solution must be resolved over the stochastic space, which may be high dimensional. While these problems have been solved in a number of relevant applications [30, 49, 178], they are impractically expensive for many important engineering and science applications.

1.3.2 Trust Region Methods

One of the foundational technologies that this thesis builds upon and extensively utilizes are trust region methods for numerical optimization. Trust region methods are a popular and robust globalization strategy for numerical optimization solvers, that is, a framework for ensuring a local minimum is obtained, regardless of the starting point. While they are not usually considered as *efficient* as line-search methods [71, 143], they are popular due to their robustness and flexibility. Let $F : \mathbb{R}^{N_{\boldsymbol{\mu}}} \to \mathbb{R}$ define a function to be minimized and suppose evaluations of $F(\boldsymbol{\mu})$ and $\nabla F(\boldsymbol{\mu})$ are expensive. Early trust region methods replaced the potentially expensive optimization problem

$$\underset{\boldsymbol{\mu}\in\mathbb{R}^{N_{\boldsymbol{\mu}}}}{\text{minimize}} \quad F(\boldsymbol{\mu})$$

with the inexpensive quadratic program

$$\begin{aligned} \underset{\boldsymbol{\mu}\in\mathbb{R}^{N_{\boldsymbol{\mu}}}}{\text{minimize}} \quad & m_k(\boldsymbol{\mu}) \coloneqq F(\boldsymbol{\mu}_k) + \nabla F(\boldsymbol{\mu}_k)(\boldsymbol{\mu} - \boldsymbol{\mu}_k) + \frac{1}{2}(\boldsymbol{\mu} - \boldsymbol{\mu}_k)^T \boldsymbol{B}_k(\boldsymbol{\mu} - \boldsymbol{\mu}_k) \\ \text{subject to} \quad & \|\boldsymbol{\mu} - \boldsymbol{\mu}_k\| \leq \Delta_k, \end{aligned}$$

where $\boldsymbol{B}_k$ is a symmetric positive-definite approximation of the Hessian $\nabla^2 F(\boldsymbol{\mu}_k)$. The solution of this trust region subproblem provides a candidate for the new trust region center and, depending on how well the reduction actually achieved by accepting the step compares to the reduction predicted by the quadratic model, the step is accepted or rejected and the trust region radius Δ_k is modified accordingly. The quality of the trust region step is assessed by comparing the actual-to-predicted reduction ratio (ρ_k) to unity

$$\rho_k = \frac{F(\boldsymbol{\mu}_k) - F(\hat{\boldsymbol{\mu}}_k)}{m_k(\boldsymbol{\mu}_k) - m_k(\hat{\boldsymbol{\mu}}_k)},$$

where $\hat{\boldsymbol{\mu}}_k$ is the candidate step, defined as the solution of the trust region subproblem. Once the details of the step acceptance and radius modification are complete, it can be shown [48] that the sequence of trust region centers $\{\boldsymbol{\mu}_k\}$ converges to a first-order critical point

$$\lim_{k\to\infty} \|\nabla F(\boldsymbol{\mu}_k)\| = 0.$$

The pivotal work in [159, 133] established convergence under only mild conditions—known as the fraction of Cauchy decrease—on the candidate step produced by the quadratic program. A slew of specialized, efficient solvers have been developed that generate steps guaranteed to satisfy the fraction of Cauchy decrease; see [133, 48] for an extensive overview.

In many applications, it may be expensive or impossible to evaluate the objective function or its gradient to construct the quadratic approximation, e.g., if $F(\boldsymbol{\mu})$ corresponds to the quantity of interest of a partial differential equation or an iterative linear solver [93, 166, 92] is used to compute to compute $F(\boldsymbol{\mu})$ or $\nabla F(\boldsymbol{\mu})$, and a host of work [133, 189, 34, 35, 36, 48, 93, 216, 108, 109] has been done to allow for *inexact gradient evaluations* to be used in the definition of the trust region subproblem

and *inexact objective evaluations* in the computation of ρ_k. Moré [133] introduced an inexact gradient condition that requires the gradient approximation at the trust region center, $\boldsymbol{g}_k$, asymptotically approaches the true gradient, i.e., $\|\nabla F(\boldsymbol{\mu}_k) - \boldsymbol{g}_k\| \to 0$ for any convergent sequence $\{\boldsymbol{\mu}_k\}$. While this provides substantial flexibility over previous work that requires first-order consistency of the approximation and model ($\boldsymbol{g}_k = \nabla F(\boldsymbol{\mu}_k)$), this condition does not suggest an accuracy condition on $\boldsymbol{g}_k$ at a particular iteration. Carter [34, 35, 36] suggested the relative gradient error condition

$$\|\nabla F(\boldsymbol{\mu}_k) - \boldsymbol{g}_k\| \leq \eta \|\boldsymbol{g}_k\| \qquad \eta \in (0,\, 1),$$

which has served as the basis for many trust region model management methods, including the popular Trust Region Proper Orthogonal Decomposition [10] method. The Carter condition is useful because it does not require $\boldsymbol{g}_k$ be recomputed to higher accuracy after a failed step; however, it requires the evaluation of the gradient error (or a tight bound), which may be impractical in many situations. Toint [189] suggested the gradient condition

$$\|\nabla F(\boldsymbol{\mu}_k) - \boldsymbol{g}_k\| \leq \min\{\kappa_1 \Delta_k,\, \kappa_2\} \qquad \kappa_1,\, \kappa_2 > 0$$

that requires increased accuracy as Δ_k decreases, i.e., after failed iterations, but relies on arbitrary constants κ_1, κ_2. Heinkenschloss and Vincent [93] suggested a similar gradient condition in the context of a Sequential Quadratic Programming (SQP) method

$$\|\nabla F(\boldsymbol{\mu}_k) - \boldsymbol{g}_k\| \leq \xi \min\{\|\boldsymbol{g}_k\|\,,\, \Delta_k\} \qquad \xi > 0$$

that requires increased accuracy after failed iterations or near convergence and also depends on an *arbitrary constant*. The arbitrary constants required by the Toint [189] and Heinkenschloss-Vincent [93] bounds are significant as they permit the use of error indicators that can completely circumvent the need to compute or tightly bound the gradient error. Suppose an error indicator $\varphi_k : \mathbb{R}^{N_{\boldsymbol{\mu}}} \to \mathbb{R}$ can be derived such that

$$\|\nabla F(\boldsymbol{\mu}_k) - \boldsymbol{g}_k\| \leq \xi \varphi_k(\boldsymbol{\mu}_k) \qquad \xi > 0,$$

where $\xi > 0$ is an arbitrary constant. Then the Heinkenschloss-Vincent [93] gradient condition will be satisfied if the error indicator satisfies

$$\varphi_k(\boldsymbol{\mu}_k) \leq \kappa \min\{\|\boldsymbol{g}_k\|\,,\, \Delta_k\},$$

where $\kappa > 0$ is *any* user-defined constant. Since the error indicator is solely used to enforce the required gradient condition, the constant $\xi > 0$ is never computed and may depend on quantities that, in general, cannot be computed such as Lipschitz constants or bounds on various quantities. This work employs the Heinkenschloss-Vincent condition due to the required generality in handling complex, nonlinear PDE-constrained optimization problems where tight gradient error bounds are not readily available.

Similar to the inexact gradient condition used in the trust region subproblem, conditions have been developed [34, 216, 109] for using inexact objective function evaluations in the actual-to-predicted reduction ratio, ρ_k. The asymptotic condition in [109] allows for the same flexibility as the Heinkenschloss-Vincent gradient condition and will be used in this work. Kouri [109] replaced the computation of ρ_k with

$$\tilde{\rho}_k = \frac{\psi_k(\boldsymbol{\mu}_k) - \psi_k(\hat{\boldsymbol{\mu}}_k)}{m_k(\boldsymbol{\mu}_k) - m_k(\hat{\boldsymbol{\mu}}_k)},$$

where $\psi_k : \mathbb{R}^{N_\mu} \to \mathbb{R}$ is the inexact objective model that satisfies

$$|F(\boldsymbol{\mu}_k) - F(\hat{\boldsymbol{\mu}}_k) + \psi_k(\hat{\boldsymbol{\mu}}_k) - \psi_k(\boldsymbol{\mu}_k)| \leq \sigma \left[\eta \min\{m_k(\boldsymbol{\mu}_k) - m_k(\hat{\boldsymbol{\mu}}_k),\, r_k\}\right]^{1/\omega} \qquad \sigma > 0$$

and σ is an arbitrary constant, $\{r_k\}_{k=1}^{\infty}$ is a forcing sequence such that $r_k \to 0$, and $\hat{\boldsymbol{\mu}}_k$ is the candidate step at iteration k. This condition permits the use of an error indicator $\theta_k : \mathbb{R}^{N_\mu} \to \mathbb{R}$ such that

$$|F(\boldsymbol{\mu}_k) - F(\boldsymbol{\mu}) + \psi_k(\boldsymbol{\mu}) - \psi_k(\boldsymbol{\mu}_k)| \leq \sigma\theta_k(\boldsymbol{\mu}) \qquad \sigma > 0.$$

The true error can be disregarded and the inexact objective condition enforced solely based on the error indicator

$$\theta_k(\hat{\boldsymbol{\mu}}_k)^\omega \leq \eta \min\{m_k(\boldsymbol{\mu}_k) - m_k(\hat{\boldsymbol{\mu}}_k),\, r_k\},$$

where ω, $\eta \in (0,\, 1)$ are algorithmic constants. When the Heinkenschloss-Vincent gradient condition [93] and Kouri objective condition [109] are combined into a single trust region method, as seen in [109], the entire algorithm proceeds without requiring queries to $F(\boldsymbol{\mu})$ or $\nabla F(\boldsymbol{\mu})$ and guarantees global convergence—this flexibility will be built upon and leveraged in this work as I look to develop methods that address large-scale, expensive problems where inexpensive, tight error bounds are not available.

In many cases, it is possible to obtain an approximation that is superior to the basic quadratic model, which can be used to provide a better approximation model $m_k(\boldsymbol{\mu})$ in the trust region framework. Alexandrov [4, 6] introduced the trust region model management framework that allows for this and proves global convergence, provided the approximation model satisfies first-order consistency at trust region centers

$$m_k(\boldsymbol{\mu}_k) = F(\boldsymbol{\mu}_k) \qquad \nabla m_k(\boldsymbol{\mu}_k) = \nabla F(\boldsymbol{\mu}_k).$$

These requirements can be weakened by introducing the inexact gradient conditions of [189, 35, 93] and inexact objective condition of [34, 109]. This flexibility has been leveraged in a number of contexts, most notably the Trust Region Proper Orthogonal Decomposition method [10, 57, 1, 170, 186] where the approximation model is taken as the projection-based reduced-order model whose reduced basis is computed via Proper Orthogonal Decomposition (POD) and the method of snapshots [183] and the Carter condition [35] is employed. It was also leveraged in [108, 109] in the context of PDE-constrained optimization uncertainty where the model problem employed

dimension-adaptive sparse grids to approximate the integral of the PDE quantity of interest over the stochastic space.

1.3.3 Projection-Based Model Reduction

Another pivotal technology in this work is projection-based model reduction, which will be used to define inexpensive approximation models for the expensive PDE discretization and solver underlying the PDE-constrained optimization problem of interest. The concepts underlying modern reduced-order models have been used in the context of modal decomposition for linear structural dynamics for several decades [65]. In this approach, the dynamics of a particular structure are approximated using its dominant modes, which are computed via an eigenvalue decomposition of the system mass and stiffness matrices. Generalization to the case of a nonlinear structure exist [98], but have not seen the same widespread adoption as the linear case.

Modern approaches to projection-based model reduction include the reduced basis method [121, 122, 17, 173] and methods based on Proper Orthogonal Decomposition (POD) [21, 101] and the method of snapshots [183]. The reduced basis method employs a variational framework and constructs a reduced basis from solutions of the underlying PDE that are greedily sampled in the parameter space at locations where an inexpensive error bound on the reduced-order model is maximized [149, 173]. This is usually embedded in an offline-online framework [17, 149, 173] where all expensive operations related to sampling the PDE and construction of the reduced basis are confined to an offline phase and the inexpensive reduced-order model is repeatedly queried in the online phase. While this method possesses a beautiful mathematical framework, it relies on properties of the underlying PDE such as linearity and ellipticity for the derivation of the error bounds and the efficient offline-online decomposition. POD-based model reduction is a general framework that uses POD to *compress* "snapshots" of the PDE solution at particular time instances and parameter configurations to generate a physics-based basis that will be used to approximate the solution. The governing equations are restricted to a low-dimensional "trial" subspace and projected onto an appropriate "test" subspace. The result is a small nonlinear system of equations—few unknowns from the introduction of the trial subspace and few equations from the projection onto the test subspace. An increasing popular approach in model reduction is to choose the test basis such that the resulting reduced-order model minimizes the residual in some norm [31, 89]. Such reduced-order models have been called "optimal" for a given trial subspace in this particular norm [31]. These "optimal" or minimum-residual reduced-order models have been extensively studied in [115, 28, 31, 89]. Chapter 4 details several properties of minimum-residual reduced-order models, some of which are new, that will be used in Chapters 5–6 in the construction of optimization methods based on reduced-order models. An important contribution of this work is the extension of the concept of minimum-residual reduced-order models from the primal setting to sensitivity and adjoint PDEs. It will be shown that this approach to compute reduced sensitivities and adjoints will circumvent many difficulties that arise in directly considering the sensitivity or adjoint of a minimum-residual reduced-order model. Furthermore, conditions will be provided under which the minimum-residual sensitivity/adjoint

reduced-order models agree with the sensitivity/adjoint of the primal reduced-order model. These contributions are provided in Chapter 4.

Both the reduced basis method and POD-based methods construct the trial basis from solution snapshots. A number of works have considered snapshots based on other types of information, including sensitivities [87, 86, 32, 85, 52, 210, 198], adjoints [57, 74], unconverged solutions [198], residuals at unconverged solutions [198], and Krylov vectors from the linear system solve that arises at each Newton-Raphson iteration [198]. However, it is well-known that the singular value decomposition underlying POD is sensitive to the relative scaling of the columns in the data matrix and this *heterogeneous* collection of snapshots should not be carelessly lumped into a single data matrix for compression. That is, when incorporating fundamentally different types of snapshots (entries have different physical units and likely different scales), care must be taken to ensure the resulting decomposition is useful. The work in [32] weighs sensitivities by increment in the parameter to make their units consistent with primal snapshots. A more general approach taken in [74, 210, 198] is to use POD to compress on each type of snapshot individually then concatenate the resulting basis. In [52] a separate basis was constructed for each sensitivity in the construction of a sensitivity ROM, each of which was computed based on POD of snapshots of the corresponding sensitivity. This work builds on the approach in [74, 210, 198] by using POD to build a basis from homogeneous snapshot types and combining the results into a single basis. I further generalize this method to ensure particular snapshots are preserved in the resulting subspace. This will be pivotal in guaranteeing required accuracy at trust region centers when the model reduction technology is combined with the trust region method of Chapter 3 to produce globally convergent, efficient deterministic and stochastic PDE-constrained optimization solvers in Chapters 5–6.

Finally, partial differential equations that do not possess an affine dependence on their parameters or state vector require an additional level of approximation for online efficiency. This additional approximation, referred to as *hyperreduction*, is required to reduce the complexity of evaluating nonlinear terms that are not amenable to precomputation [17, 175, 115, 41, 31, 59]. An overview of the most popular hyperreduction methods are provided in Section 4.2.3. Nonlinearities that are *polynomial* do not strictly require hyperreduction since they are amenable to precomputation and all dependence on the large dimension of the underlying PDE can be confined to the offline phase [149, 14, 16]. However, the approach quickly scales poorly with the size of the reduced-order model as the highest polynomial degree increases, e.g., they usually scale as $\mathcal{O}(k_{\boldsymbol{u}}^{m+1})$ where $k_{\boldsymbol{u}}$ is the ROM size and m is the polynomial order. Section 4.2.1 provides a detailed formulation of fully discrete PDEs with polynomial nonlinearities in the state and parameter, as well as the details of the precomputation of the monomial terms.

1.3.4 Surrogate Methods for PDE-Constrained Optimization

The methods introduced and developed in this thesis fall into the class of surrogate-based optimization methods, whereby the expensive, high-fidelity model that defines the "true" objective $F(\boldsymbol{\mu})$ and gradient $\nabla F(\boldsymbol{\mu})$ is replaced by an inexpensive approximation. The surrogate models can be based

on response surfaces [63], adaptive spatial discretizations [216], loose tolerances on linear solvers [166, 216], partially converged solutions [62], projection-based reduced-order models [10, 171, 210], and many other approximation models. This section provides a brief overview of methods that use projection-based reduced-order models as a surrogate as they are most relevant to the methods developed in this thesis. For a thorough review of surrogate-based optimization methods, see [63]. The methods reviewed in this section fall into two main categories: (1) those that adhere to a strict offline-online decomposition and (2) those that do not. For chronological accuracy, methods that do not distinguish between an offline and online phase are considered first.

Alexandrov developed the Trust Region Model Management (TRMM) framework that uses a general approximation model that satisfies first-order consistency in the context of unconstrained [4] and nonlinearly constrained optimization [6]. The famous Trust Region Proper Orthogonal Decomposition (TRPOD) method [10, 57] was among the first methods to leverage projection-based reduced-order models in a globally convergent optimization algorithm. This method does not exactly fit into Alexandrov's TRMM framework as TRPOD uses the Carter condition [35] to define the accuracy required of the reduced-order model to ensure convergence. This condition is considerably weaker than fist-order consistency and allows a relatively small reduced-order model to be used. In the TRPOD method, at the control corresponding to the trust region center, the snapshots are collected from the full-order PDE simulation and compressed using POD. The size of the reduced basis is selected to ensure the Carter condition is satisfied, which involves computing the true gradient error at reduced-order models of increasing size. Later work on TRPOD also collected adjoint snapshots and built a separate POD-based ROM for the adjoint PDE [57]. While this leads to gradients that are not consistent with the quantities of interest computed from the primal reduced-order model, it did not hinder convergence in the numerical experiments in [57]. TRPOD was originally developed for unconstrained problems and was later applied to problems with nonlinear constraints [1, 186] following Alexandrov's work [6]. A method similar to TRPOD is called Optimality System POD (OS-POD) [113], which attempts to build a reduced-order model *at the optimal control*. It formulates an optimization problem that consists of the unreduced optimization problem, the reduced optimization problem, and the POD system. The monolithic optimization problem is solved using a simple splitting method that results in a method similar to TRPOD with two main exceptions: a trust region framework is not used to manage the approximation model and OS-POD involves an intermediate step with the true gradient of the objective function. Another surrogate optimization solver similar to TRPOD was developed in [208]. This method used projection-based reduced-order models based on a Krylov-Pade approximation (and therefore specific to linear PDEs) as the approximation model. Two significant contributions of this work are: (1) the flexibility to handle generalized (non-quadratic) trust region constraints and (2) a new method to assess the trust region step without evaluating the HDM. This thesis considers similar generalizations over the standard TRPOD method with the most significant difference being the proposed methods are built on the flexible trust region method of [109]. This flexibility is leveraged to use unconverged solutions as snapshots and in the evaluation of the trust region step. It will also be used later to

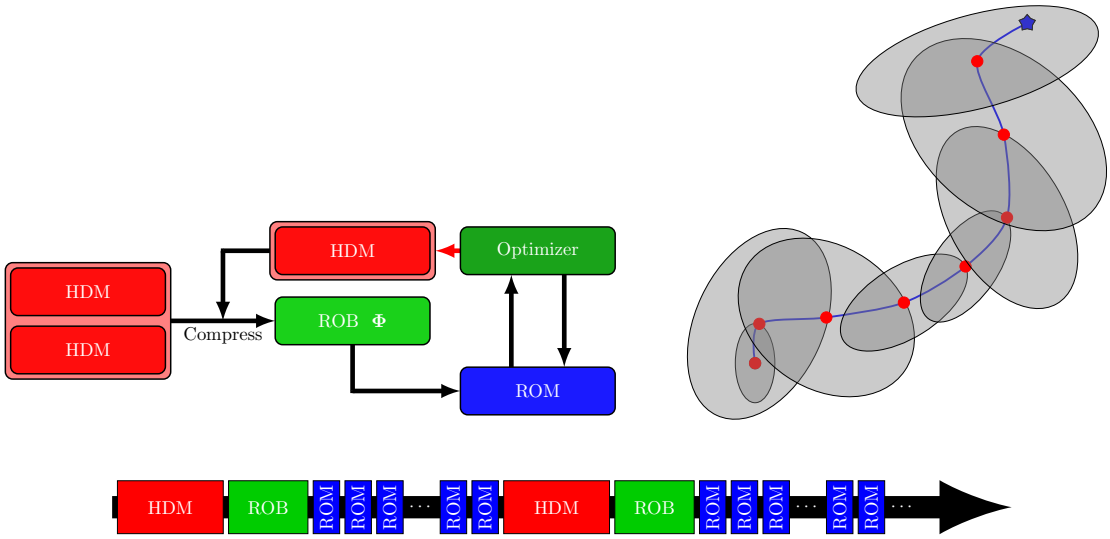


Figure 1.1: The adaptive approach to accelerate PDE-constrained optimization with projection-based reduced-order models. *Top left*: block schematic of the workflow where *few* High-Dimensional Model (HDM) samples are compressed to build the Reduced-Order Basis (ROB) and the resulting Reduced-Order Model (ROM) is used in the optimization procedure, as long as it maintains accuracy. When the accuracy degrades, an additional sample of the HDM is taken at the new point in the parameter space and the ROB is enriched. *Top right*: schematic of parameter space ($\boldsymbol{\mu}$-space) where the black dot and star are the initial guess and solution of the optimization problem, respectively, the red circles indicate HDM samples, the gray regions are the "trust regions" for the ROM constructed at each iteration, the blue line is the trajectory of the ROM optimization procedure, and the blue star is the optimal solution found by the ROM optimization. *Bottom*: schematic of the computational cost where the *expensive* (HDM evaluations and ROB construction) and *inexpensive* components are intermixed throughout the algorithm. These methods are usually equipped with global convergence theory that guarantee convergence to a local optimum of the PDE-constrained optimization problem, as indicated in the top right plot.

build a two-level approximation to accelerate stochastic PDE-constrained optimization. All of the methods based on Alexandrov's TRMM framework, as well as the other variants described here, are categorized as *adaptive* optimization procedures—see Figure 1.1—since the surrogate model is not built *once-and-for-all* in an offline phase and repeatedly queried in the online phase; rather, the surrogates are adaptively built on-the-fly during the optimization procedure.

In contrast to the method that do not distinguish between offline and online cost are the reduced basis methods that do make such a distinction [172, 174, 114, 125, 52]—see Figure 1.2. These methods sample the parameter space in an *offline* phase to collect snapshots, build a reduced basis, and precompute PDE operators contracted with the reduced basis. In the online phase, the reduced-order model is queried many times as the PDE-constrained optimization problem is solved with the ROM in place of the original PDE. Due to the strict offline-online decomposition, global convergence usually cannot be established. However, since these methods usually consider linear, elliptic PDEs and a quadratic objective function (resulting in a convex optimization problem), error bounds between the computed solution and unique optimum can be derived and computed. As the

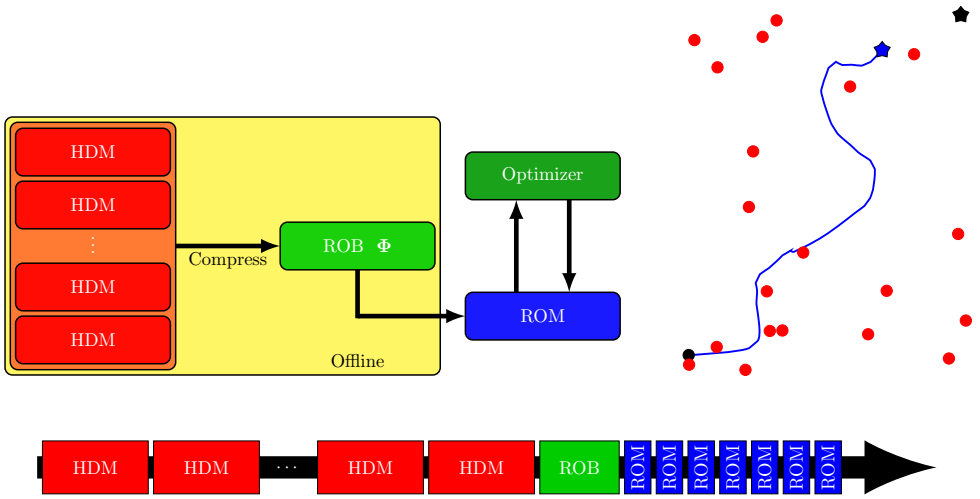


Figure 1.2: The offline-online approach to accelerate PDE-constrained optimization with projection-based reduced-order models. *Top left*: block schematic of the workflow where a number of High-Dimensional Model (HDM) samples are compressed to build the Reduced-Order Basis (ROB) in an *offline* phase; the resulting inexpensive Reduced-Order Model (ROM) is repeatedly queried in the *online* optimization phase. *Top right*: schematic of parameter space ($\boldsymbol{\mu}$-space) where the black dot and star are the initial guess and solution of the optimization problem, respectively, the red circles indicate HDM samples, the blue line is the trajectory of the ROM optimization procedure, and the blue star is the optimal solution found by the ROM optimization. *Bottom*: schematic of the computational cost where there is a clear distinction between the *expensive* components (HDM evaluations and ROB construction) that are done *once-and-for-all* in the offline phase and the inexpensive components (ROM evaluations) that are repeatedly queried in the online phase. In general, these methods are not guaranteed to converge to a local optimum of the PDE-constrained optimization problem, as indicated in the top right plot.

assumptions on these methods are too strong for the applications of interest in this thesis, they will not be considered further.

To this point, only methods developed for deterministic PDE-constrained optimization problems have been considered. In the context of PDE-constrained optimization under uncertainty, Kouri [108, 109] used dimension-adaptive sparse grids to define the quadrature nodes in a stochastic collocation method to define an inexpensive surrogate model (due to a quadrature rule with fewer points than would be required by an isotropic sparse grid or tensor product rule). This approximation model was embedded in the trust region method developed in those papers that allows for inexact gradient and objective evaluations. Chen [44, 42, 43] introduced an additional level of approximation by using reduced-order models in addition to sparse grids. This work focused on simple PDEs and employed an offline-online framework (instead of a globally convergent trust region framework that breaks the offline-online decomposition). The method developed in this work for efficient PDE-constrained optimization under uncertainty is a crossover between these two methods: I develop a two-level approximation based on projection-based model reduction and dimension-adaptive sparse grids and embed the approximation model in a globally convergent trust region framework. This enables the

framework to handle general PDEs, leverage the efficiency benefits of reduced-order models, and ensure global convergence; see Chapter 6 for details.

The methods developed in this thesis most resemble TRPOD in that snapshots of the HDM at the trust region center will define the reduced-order model. A crucial difference that leads to improved efficiency and flexibility is that the proposed methods will be built on a more general and flexible trust region theory that permits the use of inexact gradient and objective evaluations and allows for more general trust region constraints. While the present work mostly focuses on problems with a relatively small parameter space compared to the state space, Appendix C uses concepts from linesearch [71, 143] and subspace [54, 119, 137, 143, 207] methods to remove this restriction. Other research that has considered the more difficult case of a large parameter space employs surrogate models with a variable parametrization [168, 167] in the TRMM framework. Other work that applies reduction to the parameter space include [117, 120]; however, these are not embedded in an adaptation algorithm and cannot establish global convergence.

1.4 Thesis Accomplishments and Outline

The contributions of this thesis are divided into two primary contributions and two auxiliary contributions. The two primary contributions are: (1) the development of an efficient solver for deterministic PDE-constrained optimization problems that leverages projection-based reduced-order models and partially converged PDE solutions and (2) the development of an efficient solver for stochastic PDE-constrained optimization problems that leverages projection-based reduced-order models and anisotropic sparse grids. The primary contributions were built on two independent auxiliary contributions that have applications that extend well beyond the scope of this thesis: (1) the introduction of a globally convergent, generalized trust region method for managing efficient approximation models and (2) the generalization and extension of minimum-residual projection-based reduced-order models [115, 28, 31, 89] to sensitivity and adjoint PDEs.

The proposed multifidelity trust region method extends the trust region method introduced in [109] by allowing a generalized trust region constraint to be used, provided the approximation model and trust region constraint are related by an asymptotic error bound that mirrors the inexact objective condition in [109]. The asymptotic error conditions on the gradient and objective evaluations are identical to those in [109]. It will be shown that the traditional trust region constraint, i.e., the ball in $\mathbb{R}^{N_\mu}$ with center $\boldsymbol{\mu}_k$, trivially satisfies the required asymptotic relationship and therefore the proposed trust region method exactly reduces to the method in [109] under this choice. Global convergence of the proposed generalized trust region method is established and closely follows the convergence theory in [133, 108, 109]. Unlike traditional trust region methods, the non-quadratic trust region constraint eliminates the possibility of using specialized methods to solve the trust region subproblem that automatically satisfy the fraction of Cauchy decrease [48]. As a result, an interior-point method is outlined to solve the trust region subproblem (an optimization problem with a single nonlinear inequality constraint) exactly. While the method is established in the

unconstrained setting, an augmented Lagrangian approach for extending it to nonlinear equality constraints is detailed. This multifidelity trust region method constitutes one of the pillars of this thesis that will be extensively used throughout. The second pillar is the primary PDE approximation technology employed in this work: projection-based model reduction.

While the concept of projection-based model reduction is not new, this work contributes to the understanding of *minimum-residual* reduced-order models and extends it to apply to sensitivity and adjoint PDEs. In particular, the concept of a minimum-residual reduced-order model for the fully discrete sensitivity and adjoint PDE is introduced and important properties are established. In particular, conditions are established that guarantee the reduced sensitivity and adjoint models agree with the sensitivity and adjoint of the primal reduced-order model and exactly reconstruct the high-dimensional model counterpart. These properties are crucial when the reduced-order model is embedded into the trust region framework as they will be used to establish the error conditions required for convergence. These minimum-residual sensitivity and adjoint reduced-order models, and the surrounding theory, represent a significant contribution as it will be shown they are significantly easier to implement in a large code-base and compute than the sensitivity and adjoint of the primal reduced-order model.

These two technologies—the generalized trust region method and minimum-residual projection-based reduced-order models—serve as pillars for the primary contributions of the thesis: efficient optimization methods for deterministic and stochastic PDE-constrained optimization. The proposed method for deterministic PDE-constrained optimization uses projection-based reduced-order models as the approximation model in the generalized trust region method and residual-based error indicators. For additional efficiency, partially converged primal and sensitivity/adjoint solutions are used as snapshots in the construction of the reduced-order models and partially converged primal solutions are used to evaluate the trust region step. The flexibility of the underlying trust region framework is leveraged to ensure the use of partially converged solutions does not hinder convergence. The proposed method for stochastic PDE-constrained optimization employs an additional level of inexactness to efficiently integrate quantities of interest over the stochastic space to form risk measures. This lead to the development of the two-level approximation of risk measures of PDE quantities of interest that uses dimension-adaptive anisotropic sparse grids to perform efficient integration in the stochastic space and model reduction for efficient PDE queries at each collocation node. This approximation is embedded in the multifidelity trust region method and global convergence is established by employing a two-level, dimension-adaptive greedy algorithm to simultaneously construct the sparse grid and reduced-order basis to satisfy required error conditions. The proposed method directly extends the work in [108, 109] that only defines the approximation model using dimension-adaptive sparse grids with PDE queries at collocation nodes performed using the high-dimensional model. It is also similar to [42, 43] that employs the same two-level approximation, but embeds it in an offline-online framework and claims regarding convergence only apply to simple PDEs.

This thesis is organized as follows (Figure 1.3). Chapter 2 provides necessary background on

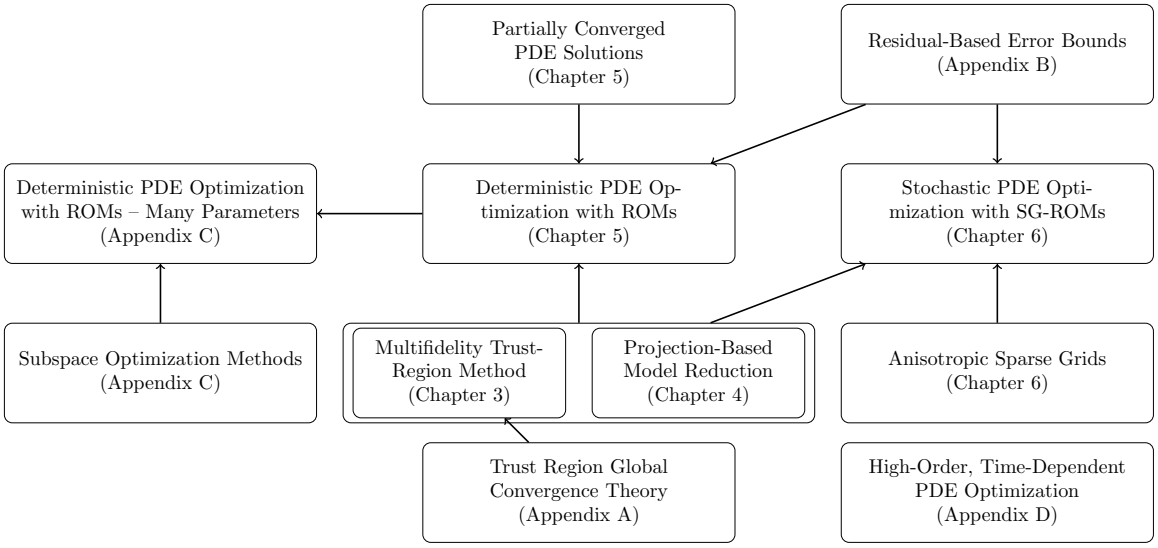


Figure 1.3: Organization of thesis

partial differential equations, their discretization, and PDE-constrained optimization. Chapter 3 discusses some necessary elements of optimization theory and introduces the proposed generalized trust region method for leveraging approximation models in an optimization setting. The convergence proof for the proposed method is provided in Appendix A. This method will serve as a cornerstone for the efficient solvers developed in Chapters 5 and 6 for deterministic and stochastic PDE-constrained optimization. Chapter 4 introduces projection-based model reduction—the approximation models that will eventually define the trust region subproblems—including some novel contributions pertaining to minimum-residual sensitivity and adjoint reduced-order models. Chapter 5 introduces one of the primary contributions of this thesis: the use projection-based reduced-order models in the generalized trust region framework to yield an efficient solver for deterministic PDE-constrained optimization problems. The potential of the method is demonstrated on a number of problems in computational mechanics, including a large-scale industrial examples of aerodynamic shape design of a full aircraft configuration. The second primary contribution of this thesis is presented in Chapter 6: the extension of the method in Chapter 5 to handle stochastic PDE-constrained optimization problems where an additional level of inexactness is introduced into the approximation model through the use of anisotropic sparse grids to efficiently integrate risk measures of PDE quantities of interest over the stochastic space. While the methods introduced in these chapters implicitly assume the number of parameters is small in comparison to the size of the PDE discretization, Appendix C develops a method to generalize these algorithms to the case where the number of parameters and state variables are comparable. Finally, Chapter 7 offers conclusions and ideas for future research and Appendix D introduces an adjoint method for optimization of time-dependent PDEs, possibly with periodicity constraints, discretized with high-order methods.

Chapter 2

PDE-Constrained Optimization

This chapter provides an overview of parametrized partial differential equations and PDE-constrained optimization that will be used extensively in the remainder of this document. The focus is primarily on *static, nonlinear* PDEs with either deterministic or stochastic coefficients and boundary conditions. The various elliptic and hyperbolic PDEs encountered in this document are also introduced, which include: the 1D viscous and inviscid Burgers' equation, linear elasticity, the total Lagrangian form of the finite deformation continuum equations, the compressible Euler equations, and the compressible Navier-Stokes equations. The chapter concludes with relevant concepts pertaining to PDE-constrained optimization including: the continuous and discrete version of the optimization problem, full-space and reduced-space solvers, gradient computations in the reduced-space approach via the sensitivity and adjoint method, and approaches to handle *side constraints*, i.e., optimization constraints *other than the* PDE constraint itself.

2.1 Parametrized Partial Differential Equations

Consider a system of partial differential equations of the form: find U such that

$$\begin{aligned} \frac{\partial U}{\partial t} + \mathcal{G}(U,\, \nabla U) &= g(x,\, t) \qquad && x \in \mathcal{B}, \quad t \in \mathcal{T} \\ \mathcal{H}(U,\, \nabla U) &= h(x,\, t) && x \in \partial\mathcal{B},\ t \in \mathcal{T} \\ U(x,\, t_0) &= U_0(x) && x \in \overline{\mathcal{B}} \end{aligned} \tag{2.1}$$

where $\mathcal{T} = (t_0,\, t_f) \subset \mathbb{R}_+$ is the temporal domain, $\mathcal{B} \subset \mathbb{R}^{n_{sd}}$ is the spatial domain with boundary $\partial\mathcal{B}$, $U(x,\, t)$ is the unknown state vector with n_c components, $\mathcal{G}$ and $\mathcal{H}$ are first-order spatial differential operators, g and h are volumetric and boundary source terms, and $U_0 : \overline{\mathcal{B}} \to \mathbb{R}^{n_{sv}}$ is the initial data. In the most general case, the domain $\overline{\mathcal{B}}$ can be time-dependent, leading to rigid and deforming domain problems. In such settings, an arbitrary Lagrangian-Eulerian description of the PDE can be employed to transform the equations to a fixed domain; see Appendix D for additional details.

From the solution of the partial differential equation (U), relevant Quantities of Interest (QoIs)[1] are defined as space-time integrals of various solution-dependent quantities over the domain. Quantities of interest are essential from a practical perspective as they provide metrics to quantify the performance and behavior of the system under consideration. In this work, QoIs will take the form

$$\mathcal{F}(U) = \int_{\mathcal{T}} \int_{\mathcal{B}} f_{\mathcal{B}}(U)\, dV\, dt + \int_{\mathcal{T}} \int_{\partial\mathcal{B}} f_{\partial\mathcal{B}}(U)\, dA\, dt, \tag{2.2}$$

where $f_{\mathcal{B}}$ and $f_{\partial\mathcal{B}}$ are relevant pointwise quantities. The form in (2.2) is general in that it encompasses integrals over subsets of the spatial and temporal domains, as well as pointwise quantities at fixed spatial locations or times. This results from the lack of regularity imposed on $f_{\mathcal{B}}$ and $f_{\partial\mathcal{B}}$ that allows for the use of indicator or Dirac functions.

In the remainder of this document, the primary interest will be in the behavior of solutions (U) and QoIs ($\mathcal{F}(U)$) of the PDE under perturbations to data of the problem—the domain ($\mathcal{B}$) and boundary ($\partial\mathcal{B}$), source terms (g and h), initial condition (U_0), or coefficients defining the differential operators $\mathcal{G}$ and $\mathcal{H}$. In subsequent sections, these *parameters* of the partial differential equation will be the optimization variables whose values will be sought such that the objective QoI is minimized and other QoI-based constraints are satisfied. Before proceeding to the discussion of PDE-constrained optimization, the various PDEs considered in this document are introduced and details regarding the discretization of parametrized partial differential equations and the corresponding quantities of interest are discussed.

2.1.1 Examples

This section provides specific examples of partial differential equations (2.1) and quantities of interest (2.2) that will be encountered in this thesis. While the examples are mostly from the fields of solid and fluid mechanics, this is not a fundamental restriction in any of the subsequent developments.

Linear Elasticity

Consider a solid body $\mathcal{B} \subset \mathbb{R}^{n_{sd}}$ subject to distributed body forces $b(x, t)$ with boundary $\partial\mathcal{B}$ decomposed into two parts: $\partial\mathcal{B}_u$ and $\partial\mathcal{B}_t$ such that $\partial\mathcal{B} = \overline{\partial\mathcal{B}_u \cup \partial\mathcal{B}_t}$. Displacements are prescribed along $\partial\mathcal{B}_u$ and $\partial\mathcal{B}_t$ is subject to prescribed traction forces. Under the assumption that the resulting deformations are infinitesimal and the pointwise stress and strain are related through a *linear* relationship, the deformation of the body is governed by the following system of partial differential equations

$$\begin{aligned} \rho\ddot{u} &= \nabla \cdot \sigma + b && x \in \mathcal{B},\, t \in \mathcal{T} \\ u &= \bar{u} && x \in \partial\mathcal{B}_u,\, t \in \mathcal{T} \\ \sigma \cdot n &= \bar{t} && x \in \partial\mathcal{B}_t,\, t \in \mathcal{T}, \end{aligned} \tag{2.3}$$

[1] Quantity of Interest (singular) will be abbreviated QoI and Quantities of Interest (plural) will be abbreviated QoIs.

where $u(x,\, t) \in \mathbb{R}^{n_{sd}}$ is the pointwise deformation and state vector of the PDE, $\sigma(x,\, t) \in \mathbb{R}^{n_{sd} \times n_{sd}}$ is the symmetric stress tensor, $\rho(x,\, t) \in \mathbb{R}_+$ is the density of the material that comprises $\mathcal{B}$, $b(x,\, t) \in \mathbb{R}^{n_{sd}}$ is the body force, $\bar{u}(x,\, t) \in \mathbb{R}^{n_{sd}}$ is the prescribed displacement on $\partial\mathcal{B}_u$, $\bar{t}(x,\, t) \in \mathbb{R}^{n_{sd}}$ is the prescribed traction on $\partial\mathcal{B}_t$, and $n(x) \in \mathbb{R}^{n_{sd}}$ is the pointwise outward normal to the boundary. The system of PDEs is closed with the stress-strain relationship (Hooke's law)

$$\sigma = \mathbb{C} : \epsilon, \tag{2.4}$$

where $\mathbb{C} \in \mathbb{R}^{n_{sd} \times n_{sd} \times n_{sd} \times n_{sd}}$ is the elasticity tensor with major and minor symmetry and $\epsilon(x,\, t) \in \mathbb{R}^{n_{sd} \times n_{sd}}$ is the strain tensor. The kinematic constraint relates the deformation to strain

$$\epsilon = \frac{1}{2}\left[\nabla u + \nabla u^T\right]. \tag{2.5}$$

Remark. *The system in (2.3) does not strictly fit into the form in (2.1) due to presence of the second-order temporal derivative, i.e., the inertial term. This can be remedied by introducing the velocity $v = \dot{u}$ and defining the state vector $U = (u,\, v)$. This will not preserve the structure of the governing equations and they are usually treated directly in their second-order form.*

There are number of relevant quantities of interest in linear elasticity including: (1) pointwise displacement magnitude, (2) pointwise stress measures, (3) mass/volume, and (4) global stiffness, to name a few. The volume of the structure and its global stiffness are defined as

$$V = \int_{\mathcal{B}} dV \qquad \text{and} \qquad S(u) = \int_{\mathcal{B}} u_k b_k \, dV + \int_{\partial\mathcal{B}} u_k \bar{t}_k \, dA, \tag{2.6}$$

respectively, where summation from 1 to n_{sd} over repeated indices is implied. The volume is a purely geometric quantity of interest as it does not depend on the solution of the partial differential equation.

Finite Deformation Continuum Mechanics

The system of partial differential equations that governs the physical setup of the previous section in the *general* case, that is, *without* the linearity assumption, is

$$\begin{aligned} \rho\ddot{u} &= \nabla \cdot P + b && X \in \mathcal{B},\, t \in \mathcal{T} \\ u &= \bar{u} && X \in \partial\mathcal{B}_u,\, t \in \mathcal{T} \\ P \cdot N &= \bar{t} && X \in \partial\mathcal{B}_t,\, t \in \mathcal{T} \end{aligned} \tag{2.7}$$

where ρ, u, b, $\bar{u}$, $\bar{t}$ are defined in the previous section on linear elasticity, $P(X,\, t)$ is the first Piola-Kirchhoff stress tensor, and $N(X)$ is the pointwise outward normal to the boundary in the reference configuration ($\mathcal{B}$). The equations are closed with a *general* constitutive relationship

$$P = P(F), \tag{2.8}$$

where $F = I + \dfrac{\partial u}{\partial X}$ is the deformation gradient. The governing equations in (2.7), posed on the reference or *undeformed* configuration ($\mathcal{B}$), are called the total Lagrangian form [19]. The equations can be transformed to the current or physical configuration using the diffeomorphism: $x(X, t) \coloneqq X + u(X, t)$, but the so-called updated Lagrangian form will not be considered in this document. The quantities of interest from the previous section (2.6) will also be used here.

General Conservation Laws

The next sequence of partial differential equations considered take the form viscous or inviscid conservation laws

$$\frac{\partial U}{\partial t} + \nabla \cdot F_I(U) + \nabla \cdot F_V(U, \nabla U) = g(x, t) \qquad x \in \mathcal{B} \tag{2.9}$$

where F_I is the inviscid flux, F_V is the viscous flux, and g is a source term. Hyperbolic systems of partial differential equations of this form describe propagation phenomena such as those in fluid dynamics and electromagnetics.

1D Inviscid Burgers' Equation

The first and simplest conservation law considered is the 1D inviscid Burgers' equation with an inflow boundary condition, which describes shock propagation of a conserved variable, u

$$\begin{aligned} \frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} &= g(x, t) \qquad && x \in (x_l, x_r),\ t \in (t_0, t_f) \\ u(x_l, t) &= h(t) && t \in (t_0, t_f). \end{aligned} \tag{2.10}$$

This constitutes a conservation law of the form (2.9) where the conserved variable, inviscid flux, and viscous flux are

$$U \coloneqq u \qquad F_I(U) \coloneqq \frac{u^2}{2} \qquad F_V(U, \nabla U) \coloneqq 0.$$

Two quantities of interest for Burgers' equation are: the regularized tracking-type functional and the amount of the conserved variable that exits the domain through the outflow boundary

$$T(u) = \frac{1}{2}\int_{t_0}^{t_f}\int_{x_l}^{x_r} \left[(u - \bar{u})^2 + \alpha g^2\right] dx\, dt \qquad \text{and} \qquad R(u) = \int_{t_0}^{t_f} u(x_r, t)\, dt,$$

respectively, where $\bar{u}$ is a target state and $\alpha > 0$ is a prescribed regularization constant.

Compressible Navier-Stokes Equations

The compressible Navier-Stokes equations govern viscous fluid flow in a domain $\mathcal{B}$ and take the form

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x_i}(\rho u_i) = 0, \tag{2.11}$$

$$\frac{\partial}{\partial t}(\rho u_i) + \frac{\partial}{\partial x_i}(\rho u_i u_j + p) = +\frac{\partial \tau_{ij}}{\partial x_j} \quad \text{for } i = 1, 2, 3, \tag{2.12}$$

$$\frac{\partial}{\partial t}(\rho E) + \frac{\partial}{\partial x_i}\left(u_j(\rho E + p)\right) = -\frac{\partial q_j}{\partial x_j} + \frac{\partial}{\partial x_j}(u_j \tau_{ij}), \tag{2.13}$$

where ρ is the fluid density, u_1, u_2, u_3 are the velocity components, and E is the total energy. The viscous stress tensor and heat flux are given by

$$\tau_{ij} = \mu\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \frac{2}{3}\frac{\partial u_k}{\partial x_k}\delta_{ij}\right) \qquad \text{and} \qquad q_j = -\frac{\mu}{\text{Pr}}\frac{\partial}{\partial x_j}\left(E + \frac{p}{\rho} - \frac{1}{2}u_k u_k\right). \tag{2.14}$$

Here, μ is the viscosity coefficient and $\text{Pr} = 0.72$ is the Prandtl number which we assume to be constant. For an ideal gas, the pressure p has the form

$$p = (\gamma - 1)\rho\left(E - \frac{1}{2}u_k u_k\right), \tag{2.15}$$

where γ is the adiabatic gas constant. All walls have no-slip boundary conditions, i.e., $u_i = 0$ for $i = 1, 2, 3$. Equations (2.11)-(2.13) can be written in conservation form as

$$U = \begin{bmatrix} \rho \\ \rho u_1 \\ \rho u_2 \\ \rho u_3 \\ \rho E \end{bmatrix} \qquad F_I(U) = \begin{bmatrix} \rho u_1 & \rho u_2 & \rho u_3 \\ p + \rho u_1^2 & \rho u_1 u_2 & \rho u_1 u_3 \\ \rho u_1 u_2 & p + \rho u_2^2 & \rho u_2 u_3 \\ \rho u_1 u_3 & \rho u_2 u_3 & p + \rho u_3^2 \\ u_1(E + p) & u_2(E + p) & u_3(E + p) \end{bmatrix} \tag{2.16}$$

$$F_V(U, \nabla U) = \begin{bmatrix} 0 & 0 & 0 \\ -\tau_{11} & -\tau_{21} & -\tau_{31} \\ -\tau_{12} & -\tau_{22} & -\tau_{32} \\ -\tau_{13} & -\tau_{23} & -\tau_{33} \\ q_1 - u_i\tau_{i1} & q_2 - u_i\tau_{i2} & q_3 - u_i\tau_{i3} \end{bmatrix}. \tag{2.17}$$

While there are a plethora of quantities of interest in fluid dynamics, the most relevant quantities tend to be time-averaged integrated forces and moments on surfaces, particularly in aerodynamics applications. The time-averaged force in the ith direction on a surface $\partial\mathcal{B}_w$ takes the form

$$F_i = \frac{1}{|\mathcal{T}|}\int_{\mathcal{T}}\int_{\partial\mathcal{B}_w}(p + \rho u_i u_j n_j - \tau_{ji} n_j)\, dA\, dt. \tag{2.18}$$

Compressible Euler Equations

The compressible Euler equations

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x_i}(\rho u_i) = 0, \tag{2.19}$$

$$\frac{\partial}{\partial t}(\rho u_i) + \frac{\partial}{\partial x_i}(\rho u_i u_j + p) = 0 \quad \text{for } i = 1, 2, 3, \tag{2.20}$$

$$\frac{\partial}{\partial t}(\rho E) + \frac{\partial}{\partial x_i}\left(u_j(\rho E + p)\right) = 0 \tag{2.21}$$

model an inviscid fluid. The Navier-Stokes equations in (2.11)-(2.13) reduce to the compressible Euler equations above in the limit of no viscosity, i.e., $\mu \to 0$. The conservation form is identical to the Navier-Stokes case with $F_V(U, \nabla U) \coloneqq 0$ and the time-averaged force on a surface $\partial\mathcal{B}_w$ are defined as

$$F_i = \frac{1}{|\mathcal{T}|}\int_{\mathcal{T}}\int_{\partial\mathcal{B}_w}(p + \rho u_i u_j n_j)\, dA. \tag{2.22}$$

Compressible Navier-Stokes Equations—Isentropic Assumption

In situations where the entropy in the system is constant, i.e., adiabatic and reversible, the Navier-Stokes equations can be simplified to its isentropic form. For a perfect gas, the entropy is defined as

$$s = p/\rho^\gamma = \text{constant}, \tag{2.23}$$

which explicitly relates the pressure and density of the flow, rendering the energy equation redundant and leads to

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x_i}(\rho u_i) = 0, \tag{2.24}$$

$$\frac{\partial}{\partial t}(\rho u_i) + \frac{\partial}{\partial x_i}(\rho u_i u_j + p) = +\frac{\partial \tau_{ij}}{\partial x_j} \quad \text{for } i = 1, 2, 3. \tag{2.25}$$

This effectively reduces the square system of PDEs of size $n_{sd} + 2$ to one of size $n_{sd} + 1$. It can be shown, under suitable assumptions, that the solution of the isentropic approximation of the Navier-Stokes equations converges to the solution of the incompressible Navier-Stokes equations as the Mach number approaches zero [118, 51, 64]. In conservation form, the compressible, isentropic Navier-Stokes equations are

$$U = \begin{bmatrix} \rho \\ \rho u_1 \\ \rho u_2 \\ \rho u_3 \end{bmatrix} \qquad F_I(U) = \begin{bmatrix} \rho u_1 & \rho u_2 & \rho u_3 \\ p + \rho u_1^2 & \rho u_1 u_2 & \rho u_1 u_3 \\ \rho u_1 u_2 & p + \rho u_2^2 & \rho u_2 u_3 \\ \rho u_1 u_3 & \rho u_2 u_3 & p + \rho u_3^2 \end{bmatrix} \tag{2.26}$$

$$F_V(U, \nabla U) = \begin{bmatrix} 0 & 0 & 0 \\ -\tau_{11} & -\tau_{21} & -\tau_{31} \\ -\tau_{12} & -\tau_{22} & -\tau_{32} \\ -\tau_{13} & -\tau_{23} & -\tau_{33} \end{bmatrix}. \tag{2.27}$$

and the time-averaged, integrated forces are computed according to (2.18).

2.1.2 Discretization: Parametrization

The primary interest in this document is not the study of partial differential equations themselves, rather the behavior of PDE solutions (U) and QoIs ($\mathcal{F}(U)$) under perturbations to the PDE itself, e.g., the domain ($\overline{\mathcal{B}}$), source terms (g and h), and coefficients of the differential operators (usually manifest as material properties in physical problems). This will lead naturally to the discussion of optimization in the next section where we seek to find the PDE domain, source term, and coefficients that minimizes some QoI and meets performance constraints on other QoIs.

In general, the quantities defining the PDE lie in infinite-dimensional function spaces, which are not convenient or practical to work with in a computational setting. Furthermore, it is difficult to design practical and relevant perturbation strategies in these spaces that will be useful in engineering and scientific applications. Accordingly, the remainder of this section discusses the finite-dimensional *parametrization* of the partial differential equation in (2.1). This will entail the definition of a vector of $N_{\boldsymbol{\mu}}$ parameters, $\boldsymbol{\mu} \in \mathbb{R}^{N_{\boldsymbol{\mu}}}$, and a precise description of the dependence of the PDE on $\boldsymbol{\mu}$. In general, the parameters can be decomposed as $\boldsymbol{\mu} = (\boldsymbol{\mu}_{\overline{\mathcal{B}}}, \boldsymbol{\mu}_g, \boldsymbol{\mu}_h, \boldsymbol{\mu}_{\mathcal{G}}, \boldsymbol{\mu}_{\mathcal{H}}, \boldsymbol{\mu}_{U_0})$, where

$$\begin{aligned} \overline{\mathcal{B}} &= \overline{\mathcal{B}}(\boldsymbol{\mu}_{\overline{\mathcal{B}}}) & U_0(x) &= U_0(x, \boldsymbol{\mu}_{U_0}) \\ g(x, t) &= g(x, t, \boldsymbol{\mu}_g) & h(x, t) &= h(x, t, \boldsymbol{\mu}_h) \\ \mathcal{G}(U, \nabla U) &= \mathcal{G}(U, \nabla U, \boldsymbol{\mu}_{\mathcal{G}}) & \mathcal{H}(U, \nabla U) &= \mathcal{H}(U, \nabla U, \boldsymbol{\mu}_{\mathcal{H}}). \end{aligned} \tag{2.28}$$

All quantities are assumed to be continuously differentiable with respect to their respective parameters. This level of granularity is not significant for this document, but must be exploited in a computational setting for an efficient implementation. Therefore, only the monolithic vector $\boldsymbol{\mu}$ is considered and the PDE dependence on this parameter takes the form

$$\begin{aligned} \overline{\mathcal{B}} &= \overline{\mathcal{B}}(\boldsymbol{\mu}) & U_0(x) &= U_0(x, \boldsymbol{\mu}) \\ g(x, t) &= g(x, t, \boldsymbol{\mu}) & h(x, t) &= h(x, t, \boldsymbol{\mu}) \\ \mathcal{G}(U, \nabla U) &= \mathcal{G}(U, \nabla U, \boldsymbol{\mu}) & \mathcal{H}(U, \nabla U) &= \mathcal{H}(U, \nabla U, \boldsymbol{\mu}). \end{aligned} \tag{2.29}$$

The various quantities in (2.28)-(2.29) are fundamentally different and specialized techniques have been developed to parametrize each. In the remainder of this section, a few techniques are discussed that are relevant to the shape and topology parametrization of $\overline{\mathcal{B}}$ and parametrization of space-time functions such as the source terms and initial conditions, as they will be most relevant to problems

encountered in subsequent chapters.

Parametrization of spatial functions

The first type of operator that arises in PDE applications, particularly in the context of optimal and distributed control [214, 190], that requires parametrization are spatial functions such as the source terms in (2.1). This section seeks to define a parameter vector $\boldsymbol{\mu}$ such that the set $\{g(x, \boldsymbol{\mu}) \mid \boldsymbol{\mu} \in \mathbb{R}^{N_{\boldsymbol{\mu}}}\}$ includes a relevant set of scalar-valued functions with some level of regularity for a given application. Only scalar-valued functions will be considered in this section as vector-valued function can be parametrized through the parametrization each component—with either the same or different parameters for each.

Local interpolation is a general strategy for parametrizing spatial functions in n_{sd} dimensions. In this setting, the domain $\overline{\mathcal{B}}$ is decomposed into elements of standard shapes—such as simplices or hyper-rectangles—and a set of polynomials of a given degree are introduced over each element. The coefficients of each polynomial in the discretization of $\overline{\mathcal{B}}$ comprise the parameter vector and the parametrized spatial function takes the form

$$g(x, \boldsymbol{\mu}) = \sum_{I=1}^{N_{\boldsymbol{\mu}}} \mu_I N_I(x) \tag{2.30}$$

where N_I are the shape functions and μ_I are the components of $\boldsymbol{\mu}$. While a parametrization of this form can be applied in any number of spatial dimensions and has a built-in refinement mechanism (by subdividing the elements in the discretization and defining polynomials over the new elements), it can lead to large parameter vectors, i.e., $N_{\boldsymbol{\mu}} \gg 1$. An additional benefit of such a parametrization is the use of an *unstructured* discretization of $\overline{\mathcal{B}}$ to refine regions where increased resolution is required, if such information is known.

On the other end of the spectrum lie *global* interpolation methods where the interpolant is defined based on information from the entire domain $\overline{\mathcal{B}}$ or parameter vector $\boldsymbol{\mu}$. Cubic splines in one dimension fall into this category—the parameter vector defines the value of the spline at "knots" and an interpolant is constructed that passes through these values and satisfies boundary conditions. In higher dimensions, radial basis functions [27] serve a similar purpose.

Parmetrization of spatio-temporal functions

In time-dependent applications, spatio-temporal functions are often used to describe source terms, boundary conditions, or even domain deformations. To optimize over these types of terms, they must be parametrized with a finite number of parameters. One option is to consider the domain $\overline{\mathcal{B}} \times \overline{\mathcal{T}}$ as a domain in $n_{sd} + 1$ dimensions and apply the local interpolation method of the previous section. Alternatively, the spatio-temporal function can be defined and parametrized using a separation of variables approach

$$g(x, t, \boldsymbol{\mu}) = g_s(x, \boldsymbol{\mu}) g_t(t, \boldsymbol{\mu}), \tag{2.31}$$

where g_s is a parametrized spatial function in $\mathbb{R}^{n_{sd}}$ and g_t is a parametrized univariate function. Any of the methods discussed in the previous section can be employed to parametrize g_s and g_t. This approach enables certain spatial or temporal requirements—such as periodicity—to be explicitly enforced in the parametrization of $g(x,\, t,\, \boldsymbol{\mu})$ through the selection of g_s and g_t. For example, if $g_t(t)$ is taken as a periodic function of period T, then $g(\,\cdot\,,\, t,\, \cdot\,)$ is guaranteed to be periodic with period T.

Shape parametrization of domain, $\overline{\mathcal{B}}$

Parametrization of the shape (at a fixed topology) of two- and three-dimensional objects with a finite number of intuitive parameters is essential in computer graphics as well as a number of engineering disciplines, usually in the context of design. A plethora of shape parametrization techniques exist [99, 177, 9, 61], each with strengths and weaknesses. These methods can be divided into two distinct classes: (1) those that parametrize $\overline{\mathcal{B}}$ directly and (2) those that parametrize the boundary $\partial\mathcal{B} \coloneqq \overline{\mathcal{B}} \setminus \mathcal{B}$ and extend the deformation to the interior—usually by solving an auxiliary PDE.

Methods that parametrize $\overline{\mathcal{B}}$ directly define an analytical mapping

$$\varphi : \mathbb{R}^{n_{sd}} \times \mathbb{R}^{N_{\boldsymbol{\mu}}} \to \mathbb{R}^{n_{sd}} \tag{2.32}$$

that maps the reference domain $\overline{\mathcal{B}}$ to the new shape $\overline{\mathcal{B}}'$, i.e., $\varphi(\overline{\mathcal{B}},\, \boldsymbol{\mu}) = \overline{\mathcal{B}}'$. These methods are usually easily parallelized as they involve *local* operations, i.e., given $\boldsymbol{\mu}$, any subset $v \subset \overline{\mathcal{B}}$ gets transformed as $\varphi(v,\, \boldsymbol{\mu})$ independent of the action of φ on $\overline{\mathcal{B}} \setminus v$. Smoothness of the new shape, $\overline{\mathcal{B}}'$ is guaranteed from the smoothness of the original shape and mapping. In many cases, a mapping of the form (2.32) can be defined analytically given a geometry of interest and requirements of the parametrization. For example, the camber of the NACA0012 airfoil in Figure 2.1 can be parametrized with three parameters using a Gaussian of the form

$$\varphi(X,\, \boldsymbol{\mu}) = \mu_1 e^{-\mu_2 (X_1 - \mu_3)^2} \tag{2.33}$$

where μ_1, μ_2, μ_3 control the magnitude, sharpness, and center of the camber, respectively; see the shape corresponding to $\mu_1 = 0.2$, $\mu_2 = 2.0$, $\mu_3 = 0.0$ in Figure 2.1. While this approach is trivial to parallelize and can lead to highly intuitive parameters, it can be cumbersome for complex 3D geometries and requires considerable expertise in designing parameters. Another approach for parametrizing $\overline{\mathcal{B}}$ directly that is extremely popular in the computer graphics community is known as Free Form Deformation (FFD) [179]. In this method, a n_{sd}-dimensional lattice of control points define an analytic function on the interior of the lattice, which can be extended to the entire space. In this setting, the displacement of the control nodes of the lattice are the parameters, which induce a deformation on the volume enclosed by the lattice and thus any body embedded in it. Figure 2.2 shows a circle parametrized with FFD based on B-splines, including the undeformed and deformed geometry and FFD lattice. While FFD is more general and flexible than manual parametrization and nearly as parallelizable, it may quickly lead to a large number of parameters, which may lead to

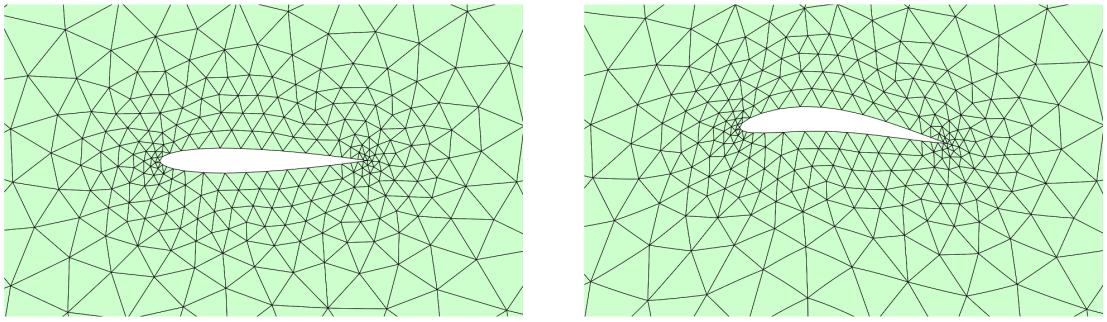


Figure 2.1: *Left*: Undeformed NACA0012 airfoil and surrounding triangular mesh. *Right*: Deformation of $\mathbb{R}^2$ according to mapping φ in (2.33) that deforms the NACA0012 geometry and surrounding mesh.

slower convergence in the context of optimization. When gradient-based optimization techniques are employed, this trade-off is usually worthwhile, particularly in aerodynamic applications. The number of parameters may be reduced by combining the manual parametrization with FFD techniques, that is, introduce a FFD lattice to control the underlying geometry and a manual parametrization that controls the FFD lattice nodes. Figures 2.3 shows the parametrization of a model of a Volkswagen Passat using FFD with two relevant and intuitive shape parameters—the height of the roof and taper of the trunk. Figure 2.4 shows the parametrization of the Common Research Model (CRM) geometry with one intuitive parameter—the dihedral of the wing.

The other class shape parametrization methods defines a parametrization of the boundary $\partial\mathcal{B}$ and propagates the deformation to the interior $\mathcal{B}$, usually via the solution of a partial differential equation such as linear or nonlinear elasticity with prescribed displacement on $\partial\mathcal{B}$ [58, 155]. Analytical methods such as splines ($n_{sd} = 2$) or Non-Uniform Rational B-Splines (NURBS) patches ($n_{sd} = 3$) are commonly used for the surface parametrization. Another popular method uses the *design element concept* where a finite element mesh is defined such that it encloses the geometry of interest, $\partial\mathcal{B}$, and the finite element shape functions define the deformation of the enclosed volume[2]. Figure 2.5 provides an example of a NACA0012 airfoil parametrized with a single cubic design element.

All of the parametrization methods considered in this section are useful in parametrizing the *shape* of an object with a *fixed topology*. Methods for parametrizing the topology of a domain will be discussed in the next section—they are fundamentally different and inevitably lead to a large number of parameters, $N_{\boldsymbol{\mu}} \gg 1$.

Topology parametrization of domain, $\overline{\mathcal{B}}$

Two prevailing methods are available for parametrizing the topology of a domain, $\overline{\mathcal{B}}$: (1) density-based methods [181, 20] and (2) level set methods [192]. Density methods define the topology of the domain using an indicator function

$$\chi : \mathbb{R}^{n_{sd}} \to \{0, 1\}, \tag{2.34}$$

[2]The design element concept can also be used to directly parametrize $\overline{\mathcal{B}}$.

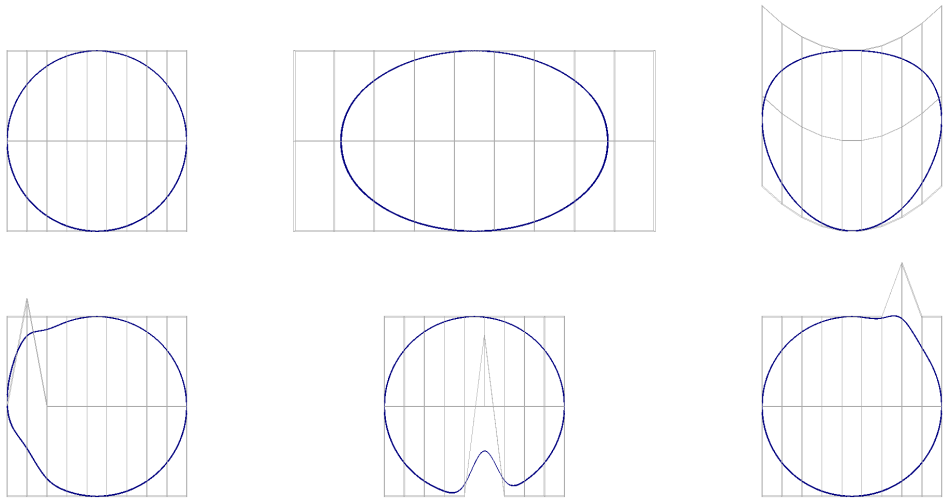


Figure 2.2: *Top left*: Undeformed geometry of a circle (blue) and a FFD lattice (gray). *Top center*: Perturbation of FFD control nodes according to an x-directed elongation mode and resulting shape of the circle. *Top right*: Perturbation of FFD control nodes according to a bending mode and resulting shape of the circle. *Bottom*: Local perturbations to individual FFD control nodes in the y direction and the resulting shape of the circle.

where $\chi(\boldsymbol{x}) = 1$ if $\boldsymbol{x} \in \overline{\mathcal{B}}$ and $\chi(\boldsymbol{x}) = 0$ otherwise. The topology of $\overline{\mathcal{B}}$ is then parametrized by parametrizing the function χ using any of the methods previously discussed. The most common approach to parametrize χ is to partition a subset of $\mathbb{R}^{n_{sd}}$ into $N_{\boldsymbol{\mu}}$ *elements* or patches of finite volume and define χ to be constant within each element k with value $\mu_k \in \{0, 1\}$[3]. Figures 2.6 – 2.9 show the topology of a cantilever, cube, and lacrosse head parametrized with a density-based approach that uses a constant value of χ in each element. This approach has the advantage of a simple implementation, but smooth topologies can only be obtained if an extremely large number of elements are used, i.e., $N_{\boldsymbol{\mu}} \gg 1$.

Conversely, level set methods define the topology implicitly by identifying all surfaces or interfaces as the zero level-set of an implicit function,

$$\phi : \mathbb{R}^{n_{sd}} \to \mathbb{R}, \tag{2.35}$$

where $\overline{\mathcal{B}} = \{\boldsymbol{x} \in \mathbb{R}^{n_{sd}} \mid \phi(\boldsymbol{x}) \leq 0\}$ and $\partial\mathcal{B} = \{\boldsymbol{x} \in \mathbb{R}^{n_{sd}} \mid \phi(\boldsymbol{x}) = 0\}$. The parametrization of the spatial function ϕ using any of the techniques previously discussed leads to the topology parametrization.

2.1.3 Discretization: Governing Equations

With the techniques described in the previous section, the parametrization of the PDE can be encoded in the finite-dimensional vector $\boldsymbol{\mu} \in \mathbb{R}^{N_\mu}$ that contains all types of parameters considered.

[3]It is often necessary to relax the range of μ_k to [0, 1] to obtain a *continuous* optimization problem.

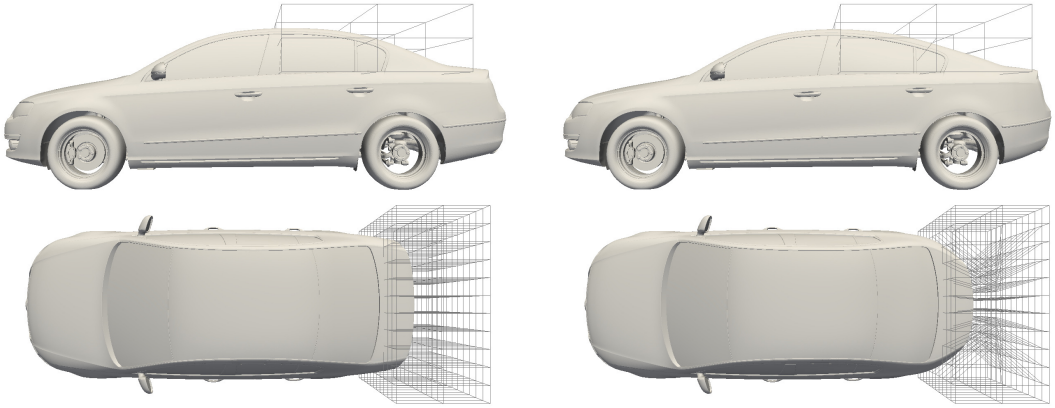


Figure 2.3: Free form deformation lattices and Volkswagen Passat geometry: (*left*) undeformed configuration, (*top right*) deformed configuration with lowered roof, and (*bottom right*) deformed configuration with steeply tapered trunk.

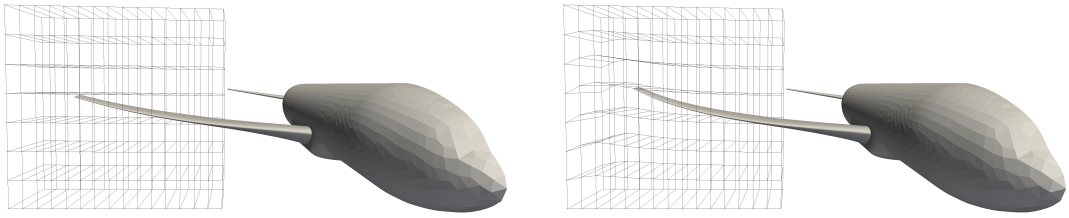
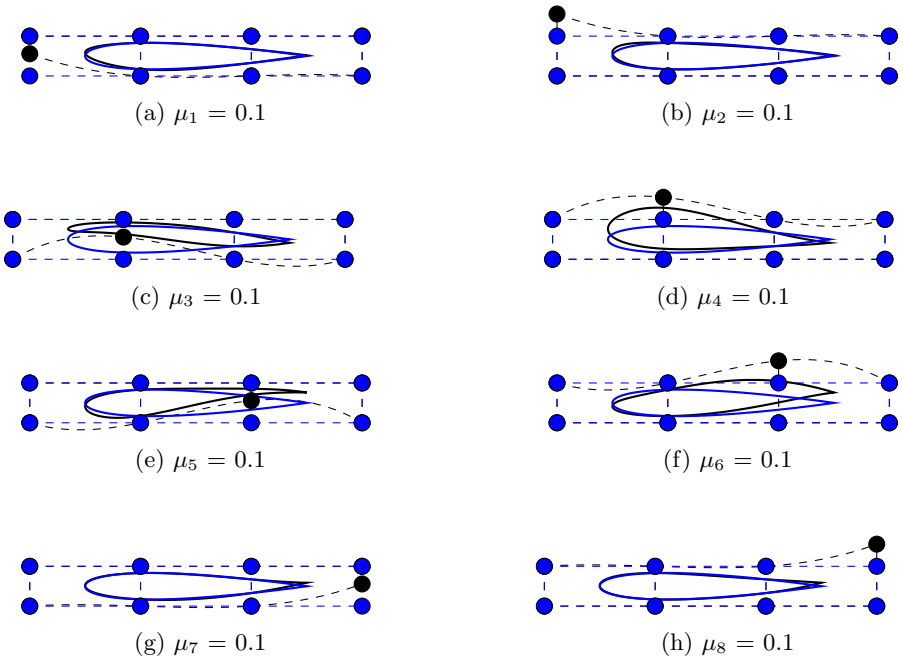


Figure 2.4: Free form deformation lattice and Common Research Model (CRM) geometry: (*left*) undeformed configuration and (*right*) deformed configuration with positive dihedral.

The partial differential equation in (2.1) under the finite-dimensional parametrization takes the form: for any $\boldsymbol{\mu} \in \mathbb{R}^{N_\mu}$, find U such that

$$\begin{aligned} \frac{\partial U}{\partial t} + \mathcal{G}(U, \nabla U, \boldsymbol{\mu}) &= g(x, t, \boldsymbol{\mu}) \qquad && x \in \mathcal{B}(\boldsymbol{\mu}), \quad t \in \mathcal{T} \\ \mathcal{H}(U, \nabla U, \boldsymbol{\mu}) &= h(x, t, \boldsymbol{\mu}) && x \in \partial\mathcal{B}(\boldsymbol{\mu}), \ t \in \mathcal{T} \\ U(x, t_0, \boldsymbol{\mu}) &= U_0(x, \boldsymbol{\mu}) && x \in \overline{\mathcal{B}}(\boldsymbol{\mu}). \end{aligned} \tag{2.36}$$

At this point, the parametrized PDE in (2.36) will be discretized in the usual two-step manner: discretization in space, i.e., semi-discretization, to yield a system of Ordinary Differential Equations (ODEs) and subsequent temporal discretization. A less commonly used alternative is to employ a monolithic space-time discretization. Given the generality of the differential operators in (2.36), it is inappropriate to commit to a single spatial discretization method given the myriad of possibilities including finite differences, finite volumes, finite elements, and discontinuous Galerkin and spectral methods. The most appropriate method depends on a number of factors including the properties of the spatial operators in (2.36), regularity of the solution U, and the complexity of the domain $\overline{\mathcal{B}}$. Finite volume, finite element, and discontinuous Galerkin methods will be used to discretize the



(a) $\mu_1 = 0.1$ (b) $\mu_2 = 0.1$

(c) $\mu_3 = 0.1$ (d) $\mu_4 = 0.1$

(e) $\mu_5 = 0.1$ (f) $\mu_6 = 0.1$

(g) $\mu_7 = 0.1$ (h) $\mu_8 = 0.1$

Figure 2.5: Shape parametrization of a NACA0012 airfoil using a *cubic* design element. Blue nodes and lines designate the undeformed design element and shape and black nodes and lines designate the deformed design element and shape.

various PDEs that arise in this work. At this point, an unspecified spatial discretization is applied to the parametrized PDEs in (2.36) to yield the nonlinear system of ODEs: for any $\boldsymbol{\mu} \in \mathbb{R}^{N_{\boldsymbol{\mu}}}$, find $\boldsymbol{u}$ such that

$$\begin{aligned} \boldsymbol{M}\dot{\boldsymbol{u}} &= \boldsymbol{r}(\boldsymbol{u}, t, \boldsymbol{\mu}) \qquad\qquad t \in \mathcal{T} \\ \boldsymbol{u}(0) &= \boldsymbol{u}_0(\boldsymbol{\mu}) \end{aligned} \tag{2.37}$$

where $\boldsymbol{u}(\,\cdot\,) \in \mathbb{R}^{N_{\boldsymbol{u}}}$ is the semi-discrete state vector, $\boldsymbol{M} \in \mathbb{R}^{N_{\boldsymbol{u}} \times N_{\boldsymbol{u}}}$ is the mass matrix, $\boldsymbol{r} : \mathbb{R}^{N_{\boldsymbol{u}}} \times \mathbb{R}_+ \times \mathbb{R}^{N_{\boldsymbol{\mu}}} \to \mathbb{R}^{N_{\boldsymbol{u}}}$ is the nonlinear function that encodes the spatial discretization of the partial differential equation and boundary conditions in (2.36), and $\boldsymbol{u}_0(\boldsymbol{\mu}) \in \mathbb{R}^{N_{\boldsymbol{u}}}$ is the parameter-dependent initial condition that arises from the spatial discretization of $U_0(x, \boldsymbol{\mu})$. If the partial differential equation in (2.36) is steady or static, i.e., $U_{,t} = 0$, (2.37) becomes

$$\boldsymbol{r}(\boldsymbol{u}, \boldsymbol{\mu}) = 0 \tag{2.38}$$

and the discretization is complete.

Remark. *As written, the mass matrix $\boldsymbol{M}$ is time- and parameter-independent, which will not be the case if the domain is time- or parameter-dependent or, for example, if corotator-based shell elements are used in a finite element discretization of structural problems [19]. In the first case, the mass matrix can be completely fixed by using an arbitrary Lagrangian-Eulerian mapping to a fixed reference domain [211]; see Appendix D for details.*

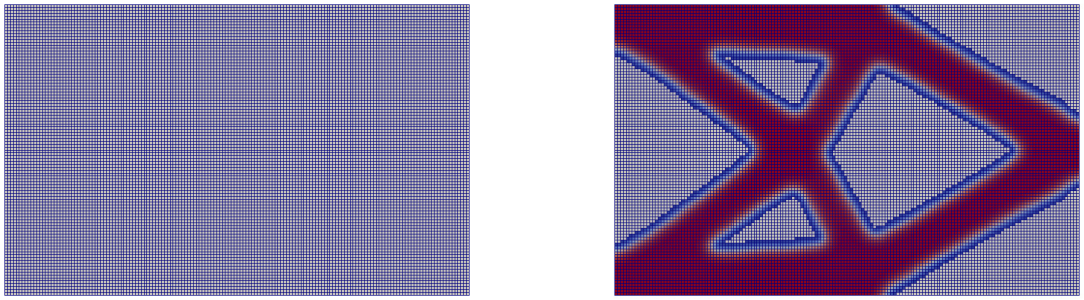


Figure 2.6: *Left*: Quadrilateral mesh of a subset of $\mathbb{R}^2$ corresponding to a rectangle (160×100 elements) whose topology is parametrized by a density-based method. *Right*: An example of an admissible topology of the density-based topological parametrization—an optimized cantilever designed to maximize the global stiffness of the structure under a vertical load at the right end.

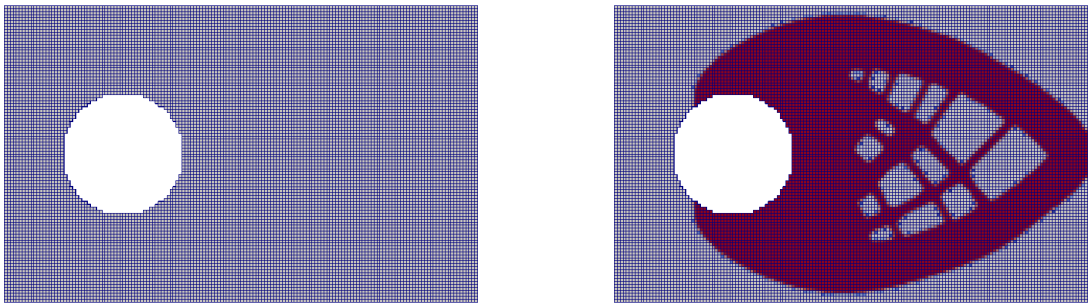


Figure 2.7: *Left*: Quadrilateral mesh of a subset of $\mathbb{R}^2$ corresponding to a rectangle (160×100 elements) with a hole whose topology is parametrized by a density-based method. *Right*: An example of an admissible topology of the density-based topological parametrization—a Michell structure [37, 94] designed to maximize the global stiffness of the structure under a vertical load at the right end.

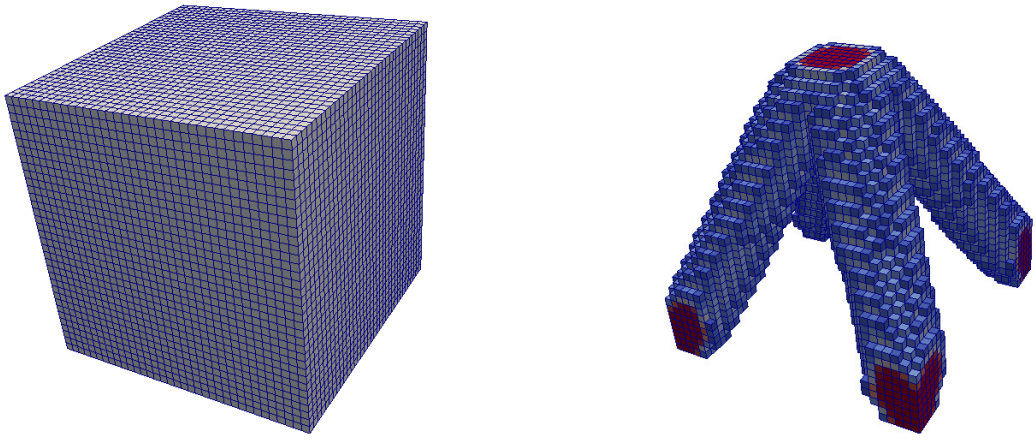


Figure 2.8: *Left*: Hexahedral mesh of a subset of $\mathbb{R}^3$ corresponding to a cube ($35 \times 35 \times 35$ elements) whose topology is parametrized by a density-based method. *Right*: An example of an admissible topology of the density-based topological parametrization—a trestle designed to maximize the global stiffness of the structure under a vertical load.

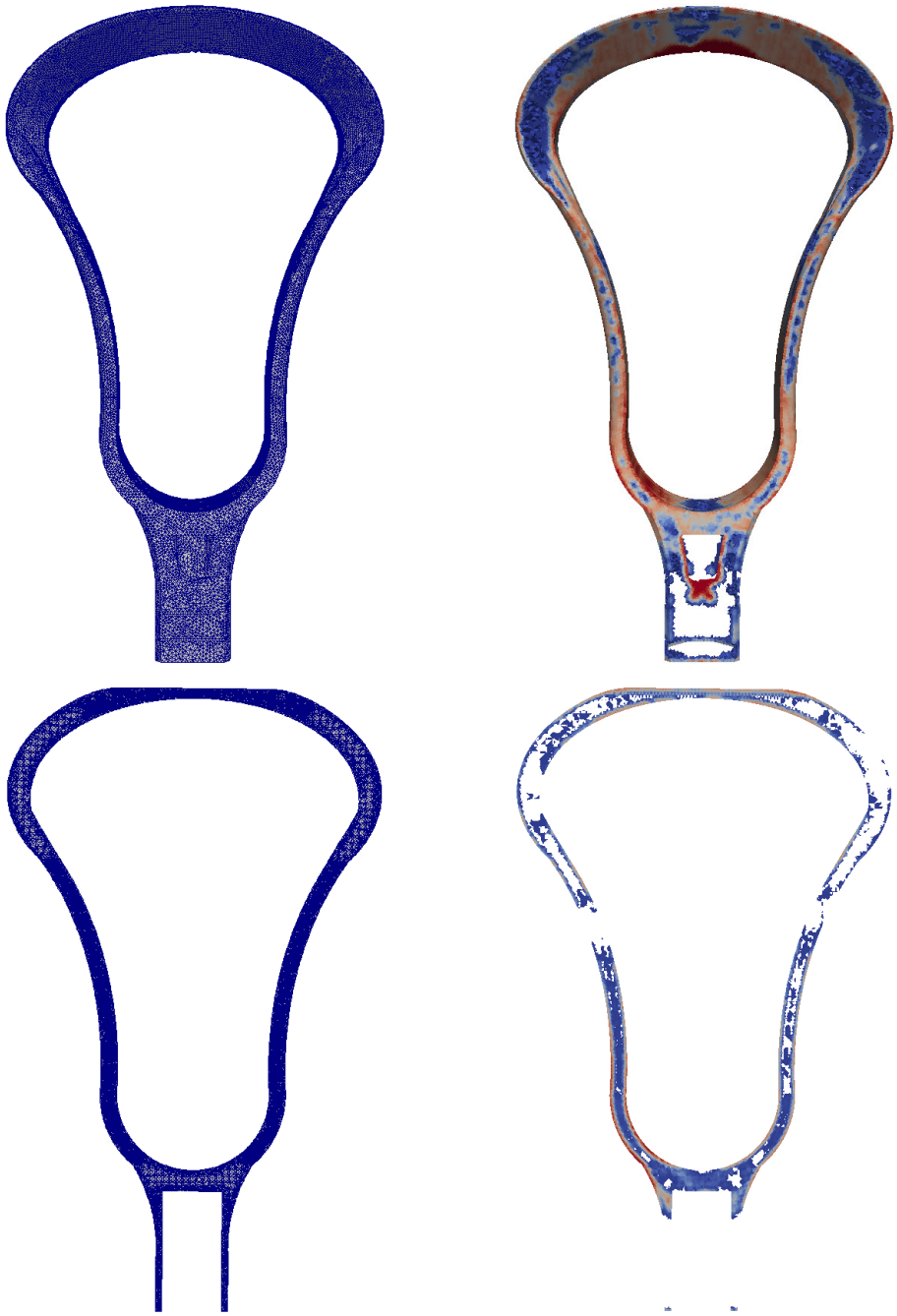


Figure 2.9: *Left*: Tetrahedral mesh of a subset of $\mathbb{R}^3$ corresponding to an unoptimized lacrosse head ($475,666$ elements) whose topology is parametrized by a density-based method. *Right*: An example of an admissible topology of the density-based topological parametrization—an unconverged maximum stiffness topology. The entire object is included in the *top* row and the *bottom* row is a slice to show internal voids in the optimized shape.

Table 2.1: Butcher Tableau for s-stage diagonally implicit Runge-Kutta scheme

c_1	a_{11}			
c_2	a_{21}	a_{22}		
$\vdots$	$\vdots$	$\vdots$	$\ddots$	
c_s	a_{s1}	a_{s2}	$\cdots$	a_{ss}
	b_1	b_2	$\cdots$	b_s

For time-dependent problems, the system of ODEs is discretized to yield the complete discretization: a sequence of algebraic, nonlinear systems of equations. The two prevailing classes of high-order implicit temporal integration schemes are: (a) Backward Differentiation Formulas (BDF) and (b) Implicit Runge-Kutta (IRK). BDF schemes are multistep schemes that have the general form

$$\boldsymbol{M}\boldsymbol{u}^{(n)} - \sum_{i=0}^{n-1} \alpha_i \boldsymbol{M}\boldsymbol{u}^{(i)} = \kappa \Delta t \boldsymbol{r}(\boldsymbol{u}^{(n)}, t_n, \boldsymbol{\mu}) \tag{2.39}$$

where α_i and κ are constants that define different schemes, such as (1) BDF1 (backward Euler): $\kappa = \alpha_{n-1} = 1$ and $\alpha_0 = \cdots \alpha_{n-2} = 0$, (2) BDF2: $\kappa = 2/3$, $\alpha_{n-1} = 4/3$, $\alpha_{n-2} = -1/3$, $\alpha_0 = \cdots = \alpha_{n-3} = 0$, and (3) BDF3: $\kappa = 6/11$, $\alpha_{n-1} = 18/11$, $\alpha_{n-2} = -9/11$, $\alpha_{n-3} = 2/11$, $\alpha_0 = \cdots = \alpha_{n-4} = 0$. They are popular since high-order accuracy can be achieved at the cost of a single nonlinear solve of size $N_{\boldsymbol{u}}$ at each time step. However, they suffer from initialization issues and are limited to second-order accuracy, if A-stability is required. In contrast, IRK schemes are single-step methods that can be A-stable and arbitrarily high-order, at the cost of solving an *enlarged* nonlinear system of equations of size $s \cdot N_{\boldsymbol{u}}$, for an s-stage scheme, at each time step. For practical problems, this can be prohibitively expensive, in terms of memory and CPU time.

A particular subclass of the IRK schemes, known as Diagonally Implicit Runge-Kutta (DIRK) schemes [3], are capable of achieving high-order accuracy with the desired stability properties, without requiring the solution of an enlarged system of equations. The DIRK schemes are defined by a *lower triangular* Butcher tableau (Table 2.1) and take the following form when applied to (2.37)

$$\begin{aligned} \boldsymbol{u}^{(0)} &= \boldsymbol{u}_0(\boldsymbol{\mu}) \\ \boldsymbol{u}^{(n)} &= \boldsymbol{u}^{(n-1)} + \sum_{i=1}^{s} b_i \boldsymbol{k}_i^{(n)} \\ \boldsymbol{M}\boldsymbol{k}_i^{(n)} &= \Delta t_n \boldsymbol{r}\left(\boldsymbol{u}_i^{(n)}, t_{n-1} + c_i \Delta t_n, \boldsymbol{\mu}\right), \end{aligned} \tag{2.40}$$

for $n = 1, \ldots, N_t$ and $i = 1, \ldots, s$, where N_t are the number of time steps in the temporal discretization and s is the number of stages in the DIRK scheme. The temporal domain $\mathcal{T}$ is discretized into N_t segments with endpoints $\{t_0, t_1, \ldots, t_{N_t}\}$, with the nth segment having length $\Delta t_n = t_n - t_{n-1}$ for $n = 1, \ldots, N_t$. Additionally, in (2.40), $\boldsymbol{u}_i^{(n)}$ is used to denote the approximation of $\boldsymbol{u}^{(n)}$ at the

ith stage of time step n

$$\boldsymbol{u}_i^{(n)} = \boldsymbol{u}_i^{(n)}(\boldsymbol{u}^{(n-1)},\ \boldsymbol{k}_1^{(n)}, \ldots,\ \boldsymbol{k}_s^{(n)}) = \boldsymbol{u}^{(n-1)} + \sum_{j=1}^{i} a_{ij}\boldsymbol{k}_j^{(n)}. \tag{2.41}$$

From (2.40), a complete time step requires the solution of a sequence of s nonlinear systems of equation of size $N_{\boldsymbol{u}}$.

From this exposition on the spatio-temporal discretization of the parametrized partial differential equation in (2.36), finding the solution of the continuous form of the equations—a task that requires searching an infinite-dimensional trial space for the solution U—has been reduced the task of finding the solution of the algebraic nonlinear system of equations in (2.38) or a sequence of such equations. The solution, $\boldsymbol{u}$, of the algebraic equations can be used, along with the shape functions underlying the spatio-temporal discretization, to reconstruct an *approximation* to the solution $U(x,\, t)$.

Remark. *One option to treat second-order temporal problems, such as those in (2.3) and (2.7) is to recast them in first-order form, as discussed in the previous section, and apply a BDF or IRK/DIRK scheme, as developed in this section. However, it is usually better to apply specialized integrators that work directly on the second-order form of the equation, such as the Newmark scheme [139] or generalized α-method [45], as these schemes are constructed with tunable damping to promote stability—a particularly important consideration in these problems.*

At this point, the governing equation and its parameters have been discretized. The final discretization task is to treat the quantity of interest. To ensure the truncation error of the governing equation and quantity of interest exactly match, a *solver-consistent* discretization [211] is employed and detailed in the next section.

2.1.4 Discretization: Quantities of Interest

Quantities of interest are among the most important aspects of a computational physics simulation, particularly in engineering applications. Optimization problems, the main focus of this work, are *completely* driven by quantities of interest as these comprise the objective and constraint functions. Therefore, care must be taken in the discretization of the integrals in (2.2) since this will introduce an *additional* error, i.e., on top of the error in the discretization of the PDE itself. To ensure the quantity of interest discretization does not dominate, thereby lowering the global order of the scheme, it is necessary that its discretization order *matches* that of the the governing equations. Clearly, it is wasteful to discretize this to a higher order than the state equation, using a similar argument.

For these reasons, discretization of (2.2) will be done in a *solver-consistent* manner, i.e., the spatial and temporal discretization used for the governing equation will also be used for the quantities of interest. Define f_h as the approximation of $\displaystyle\int_{\mathcal{B}} f_{\mathcal{B}}(U)\, dV + \int_{\partial\mathcal{B}} f_{\partial\mathcal{B}}(U)\, dA$ using the shape functions underlying the spatial discretization of the governing equations. This ensures the spatial

integration error in the quantity of interest exactly matches that of the governing equations. Next, define

$$\mathcal{F}_h(\boldsymbol{u}, \boldsymbol{\mu}, t) \coloneqq \int_{t_0}^{t} f_h(\boldsymbol{u}, \boldsymbol{\mu}, \tau)\, d\tau, \tag{2.42}$$

where the temporal domain is taken to be $\mathcal{T} = (t_0, t_f)$. Before the temporal discretization of the governing equations can be applied to discretize the integral in (2.42), it must be converted to an ODE. This is accomplished via differentiation of (2.42) with respect to t to yield

$$\dot{\mathcal{F}}_h(\boldsymbol{u}, \boldsymbol{\mu}, t) = f_h(\boldsymbol{u}, \boldsymbol{\mu}, t). \tag{2.43}$$

Augmenting the semi-discrete governing equations with this ODE results in the enlarged system of ODEs

$$\begin{bmatrix} \boldsymbol{M} & \boldsymbol{0} \\ \boldsymbol{0} & 1 \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{u}} \\ \dot{\mathcal{F}}_h \end{bmatrix} = \begin{bmatrix} \boldsymbol{r}(\boldsymbol{u}, \boldsymbol{\mu}, t) \\ f_h(\boldsymbol{u}, \boldsymbol{\mu}, t) \end{bmatrix}. \tag{2.44}$$

At this point, the same temporal discretization used for the governing equations in the previous section can be applied to discretize (2.44). A monolithic discretization of this form ensures the temporal truncation error of the governing equations and quantity of interest will exactly match. The development will proceed assuming a DIRK scheme is used—the same procedure would apply if BDF or another first-order temporal discretization was applied. Application of the DIRK scheme yields the fully discrete governing equations and corresponding solver-consistent discretization of the quantity of interest (2.2)

$$\begin{aligned} \boldsymbol{u}^{(n)} &= \boldsymbol{u}^{(n-1)} + \sum_{i=1}^{s} b_i \boldsymbol{k}_i^{(n)} \\ \mathcal{F}_h^{(n)} &= \mathcal{F}_h^{(n-1)} + \sum_{i=1}^{s} b_i f_h\left(\boldsymbol{u}_i^{(n)}, \boldsymbol{\mu}, t_{n-1} + c_i \Delta t_n\right) \\ \boldsymbol{M}\boldsymbol{k}_i^{(n)} &= \Delta t_n \boldsymbol{r}\left(\boldsymbol{u}_i^{(n)}, \boldsymbol{\mu}, t_{n-1} + c_i \Delta t_n\right). \end{aligned} \tag{2.45}$$

for $n = 1, \ldots, N_t$, $i = 1, \ldots, s$, and $\boldsymbol{u}_i^{(n)}$ is defined in (2.41). Finally, the functional in (2.42) is evaluated at time $t = t_f$ to yield the solver-consistent approximation of $\mathcal{F}_h(\boldsymbol{u}, \boldsymbol{\mu}, t_f)$

$$F(\boldsymbol{u}^{(0)}, \ldots, \boldsymbol{u}^{(N_t)}, \boldsymbol{k}_1^{(1)}, \ldots, \boldsymbol{k}_s^{(N_t)}) \coloneqq \mathcal{F}_h^{(N_t)} \approx \mathcal{F}_h(\boldsymbol{u}, \boldsymbol{\mu}, t_f). \tag{2.46}$$

While the spatially solver-consistent discretization of QoIs is widely used, particularly in the context of finite element methods, temporal discretization is commonly done via low-order quadrature rules, usually the trapezoidal rule [193, 130, 124, 205, 102]. The main advantage of this solver-consistent discretization is the asymptotic discretization order of the governing equation and quantity of interest are guaranteed to exactly match, which ensures there is no wasted error in "over-integrating" one of the terms. The solver-consistent discretization also has the advantage of a natural

and convenient implementation given the spatial and temporal discretization implementation. Finally, this method has the additional convenience of keeping a high-order accurate "current" value of the integral, i.e. at time step n, $\mathcal{F}_h^{(n)} \approx \int_{t_0}^{t_n} f_h(\tau)\, d\tau$ to high-order accuracy. This property does not hold for high-order numerical quadrature since $\int_{t_0}^{t_n} f_h(\tau)\, d\tau$ will involve $\boldsymbol{u}^{(n+j)}$, where $j \geq 1$ depends on the quadrature rule used.

This completes the discussion of parametrized *deterministic* partial differential equations. Before proceeding to the main topic of this work, PDE-constrained optimization, the notion of parametric *stochastic* partial differential equations is introduced and relevant details discussed, such as meaningful risk measures and methods to discretize the stochastic space. This will lead to the discussion of PDE-constrained optimization that will be applicable in both the deterministic and stochastic setting.

2.2 Parametrized Stochastic Partial Differential Equations

This section generalizes the concepts discussed in Section 2.1 to the case where uncertainty is present in the parametrized partial differential equation—for simplicity only static problems are considered. The ultimate goal is to setup the stochastic PDE-constrained optimization problem. The discussion begins with the formulation of parametrized, Stochastic Partial Differential Equations (SPDEs) and introduces the concept of *risk measures* of PDE quantities of interest. These risk measures will comprise the objective and constraint functions in stochastic (risk-averse) optimization problems. The SPDE will be discretized in space using the techniques introduced in Section 2.1 and collocation will be used to discretize the stochastic space. Finally, the deterministic and stochastic PDE-constrained optimization problems will be collectively detailed in Section 2.3.

Let $\mathcal{B}$ be a bounded domain in $\mathbb{R}^{n_{sd}}$ and let $(\Omega, \mathcal{F}, P)$ be a complete probability space. Here Ω is the set of outcomes, $\mathcal{F} \subset 2^{\Omega}$ is the σ-algebra of events, and $P : \mathcal{F} \to [0, 1]$ is a probability measure. Consider the stochastic boundary value problem: find U such that P-almost everywhere in Ω

$$\begin{aligned} \mathcal{G}(U, \nabla U, \boldsymbol{\mu}, \omega) &= g(x, \boldsymbol{\mu}, \omega) \qquad x \in \mathcal{B}(\boldsymbol{\mu}, \omega) \\ \mathcal{H}(U, \nabla U, \boldsymbol{\mu}, \omega) &= h(x, \boldsymbol{\mu}, \omega) \qquad x \in \partial\mathcal{B}(\boldsymbol{\mu}, \omega), \end{aligned} \tag{2.47}$$

where $\mathcal{B} \subset \mathbb{R}^{n_{sd}}$ is the spatial domain with boundary $\partial\mathcal{B}$, U is the unknown solution, $\boldsymbol{\mu} \in \mathbb{R}^{N_{\boldsymbol{\mu}}}$ are deterministic parameters, $\mathcal{G}$ and $\mathcal{H}$ are first-order spatial differential operators, and g and h are volumetric and boundary source terms. For generality, the domain, boundary, source terms, and differential operators are all taken as stochastic. The presence of the parameter vector $\boldsymbol{\mu}$ indicates that the differential operators, source terms, and boundary conditions have already been parametrized using the techniques in Section 2.1.2. Each *realization* of the parametrized, stochastic PDE in (2.47), i.e., for a given $\omega \in \Omega$, constitutes a *deterministic* PDE of the form (2.1), which can be discretized according to the methods outlined in that section. The following finite-noise assumption [13, 12] allows the source of randomness to be approximated using a finite number of

independent random variables.

Assumption 2.1 (Finite-dimensional noise)**.** *The stochastic terms in (2.47) depend on a finite number of real-valued random variables, i.e.,*

$$\begin{aligned} \mathcal{G}(\,\cdot\,,\,\cdot\,,\,\cdot\,,\,\omega) &= \mathcal{G}(\,\cdot\,,\,\cdot\,,\,\cdot\,,\,Y_1(\omega),\,\ldots,\,Y_{N_{\boldsymbol{y}}}(\omega)) \\ g(\,\cdot\,,\,\cdot\,,\,\omega) &= g(\,\cdot\,,\,\cdot\,,,\,Y_1(\omega),\,\ldots,\,Y_{N_{\boldsymbol{y}}}(\omega)) \\ \mathcal{B}(\,\cdot\,,\,\omega) &= \mathcal{B}(\,\cdot\,,\,Y_1(\omega),\,\ldots,\,Y_{N_{\boldsymbol{y}}}(\omega)), \end{aligned} \tag{2.48}$$

where $N_{\boldsymbol{y}} \in \mathbb{N}_+$ and $\{Y_n\}_{n=1}^{N_{\boldsymbol{y}}}$ are real-valued, independent random variables. A similar expansion is assumed to hold for the boundary terms.

Define $\Xi_n \coloneqq Y_n(\Omega)$ as the image of the random variables in Assumption 2.1 and $\boldsymbol{\Xi} = \Xi_1 \otimes \cdots \otimes \Xi_{N_{\boldsymbol{y}}} \subset \mathbb{R}^{N_{\boldsymbol{y}}}$. Let $\rho_n : \Xi_n \to \mathbb{R}_+$ denote the probability density of the random variable Y_n and, due to the independence of $\{Y_n\}_{n=1}^{N_{\boldsymbol{y}}}$, the joint density of the random vector $Y = (Y_1, \ldots, Y_N)$ is $\rho : \boldsymbol{\Xi} \to \mathbb{R}_+$ where $\rho = \rho_1 \otimes \cdots \otimes \rho_{N_{\boldsymbol{y}}}$. The finite noise assumption allows a change of variables that converts the parametrized stochastic partial differential equation in (2.47) to: find $U(\boldsymbol{\mu},\, \boldsymbol{y})$ such that for all $\boldsymbol{y} \in \boldsymbol{\Xi}$

$$\begin{aligned} \mathcal{G}(U,\, \nabla U,\, \boldsymbol{\mu},\, \boldsymbol{y}) &= g(x,\, \boldsymbol{\mu},\, \boldsymbol{y}) \qquad x \in \mathcal{B}(\boldsymbol{\mu},\, \boldsymbol{y}) \\ \mathcal{H}(U,\, \nabla U,\, \boldsymbol{\mu},\, \boldsymbol{y}) &= h(x,\, \boldsymbol{\mu},\, \boldsymbol{y}) \qquad x \in \partial\mathcal{B}(\boldsymbol{\mu},\, \boldsymbol{y}) \end{aligned} \tag{2.49}$$

for $\boldsymbol{\mu} \in \mathbb{R}^{N_{\boldsymbol{\mu}}}$.

2.2.1 Risk Measures of Quantities of Interest

The uncertainty that has been incorporated in the partial differential equation in (2.47) will be propagated to the quantities of interest through the solution $U(\boldsymbol{\mu},\, \boldsymbol{y})$ and possibly the domain $\mathcal{B}(\boldsymbol{\mu},\, \boldsymbol{y})$, boundary $\partial\mathcal{B}(\boldsymbol{\mu},\, \boldsymbol{y})$, and differential operators. To formulate a well-defined and *meaningful* optimization problem, we consider an objective and constraints that consist of *risk measures* of these uncertain quantities of interest. For the remainder of this section, let X be a real-valued random variable, defined as $X(\boldsymbol{y};\, \boldsymbol{\mu}) = \mathcal{F}(U(\boldsymbol{\mu},\, \boldsymbol{y}),\, \boldsymbol{\mu},\, \boldsymbol{y})$ where $\mathcal{F}$ is the quantity of interest in (2.2) without temporal dependence, generalized to the stochastic case, i.e.,

$$\mathcal{F}(U,\, \boldsymbol{\mu},\, \boldsymbol{y}) = \int_{\mathcal{B}(\boldsymbol{\mu},\, \boldsymbol{y})} f_{\mathcal{B}}(U,\, \boldsymbol{\mu},\, \boldsymbol{y})\, dV + \int_{\partial\mathcal{B}(\boldsymbol{\mu},\, \boldsymbol{y})} f_{\partial\mathcal{B}}(U,\, \boldsymbol{\mu},\, \boldsymbol{y})\, dA. \tag{2.50}$$

The dependence of the random variable X on the parameter will be dropped for the remainder of this section as treatment of the stochastic dimension is the focus.

The simplest risk measure is the expected value of the random variable

$$\mathbb{E}[X] = \int_{\boldsymbol{\Xi}} \rho(\boldsymbol{y}) X(\boldsymbol{y})\, d\boldsymbol{y}. \tag{2.51}$$

The mean of the random variable does not necessarily encode a useful measure of risk, but is a

straightforward generalization of the deterministic quantity of interest to the stochastic case. An obvious deficiency in its use as a risk measure is it does not incorporate the spread of random variable about the mean. The mean plus semideviation, defined as

$$\mathcal{R}_\beta[X] = \mathbb{E}[X] + \beta\mathbb{E}[(X - \mathbb{E}[X])_+] \tag{2.52}$$

for $\beta \in \mathbb{R}_+$, where $(x)^+ = \max\{0, x\}$ overcomes this limitation. In an optimization setting, the value of β must be determined to balance minimization of the (expected) quantity of interest with risk aversion, which may be difficult to do in practice. Another relevant risk measure is the β-quantile of the random variable, also called the value-at-risk, defined as the smallest value such that the probability that the random variable lies below said value is at least β, i.e.,

$$\text{VaR}_\beta[X] = \inf\{t \in \mathbb{R} \mid \Pr[X \leq t] \geq \beta\}, \tag{2.53}$$

where

$$\Pr[X \leq t] = \int_{\boldsymbol{y} \in \boldsymbol{\Xi} : X(\boldsymbol{y}) \leq t} \rho(\boldsymbol{y})\, d\boldsymbol{y}. \tag{2.54}$$

The main disadvantage of the value-at-risk is that it fail to emphasize rare and low probability events, which tend to be particularly important in engineering settings since they often correspond to failure. The conditional value-at-risk, defined as

$$\text{CVaR}_\beta[X] = \inf_{t \in \mathbb{R}} F_\beta(t, X), \tag{2.55}$$

where

$$F_\beta(t,\, X) = t + \frac{1}{1-\beta}\mathbb{E}\left[(X - t)_+\right] \tag{2.56}$$

circumvents this limitation. While the conditional value-at-risk is non-smooth (due to the presence of the max operator) and non-trivial to evaluate, it emphasizes rare events for $\beta \gg 0$. In the remainder of this thesis, only the expectation risk measure will be considered for simplicity. All developments will extend to any *smooth* risk measure. For non-smooth risk measures such as the semideviation and conditional value-at-risk, well-defined smoothed approximations can be used [110] in place of the risk measure itself.

2.2.2 Examples

1D Steady, Viscous Burgers' Equation with Uncertain Coefficients

The only stochastic partial differential equation considered in this thesis is the 1D steady, viscous Burgers' equation with uncertain boundary conditions, source term, and viscosity

$$\begin{aligned} -\nu(\boldsymbol{y})\partial_{xx}u(x,\, \boldsymbol{y}) + u(x,\, \boldsymbol{y})\partial_x u(x,\, \boldsymbol{y}) &= g(x,\, \boldsymbol{y}) && x \in (x_l,\, x_r),\ \boldsymbol{y} \in \boldsymbol{\Xi} \\ u(x_l,\, \boldsymbol{y}) = d_0(\boldsymbol{y}), \quad u(x_r,\, \boldsymbol{y}) &= d_1(\boldsymbol{y}) && \boldsymbol{y} \in \boldsymbol{\Xi}. \end{aligned} \tag{2.57}$$

The risk-neutral measure (expectation) of the tracking-type functional

$$T(u) = \mathbb{E}\left[\frac{1}{2}\int_{x_l}^{x_r}\left[(u(x,\,\cdot\,)-\bar{u}(x))^2+\alpha g(x,\,\cdot\,)^2\right]\,dx\right].$$

will be used to define an optimal control problem in Chapter 6.

Static Linear Elasticity with Uncertain Loading

Another stochastic partial differential equation that will be considered in future work is linear elasticity with stochasticity in the load conditions. Consider the same physical setup as the deterministic linear elasticity setup—a solid body $\mathcal{B} \subset \mathbb{R}^{n_{sd}}$ subject to *uncertain* distributed body forces with boundary $\partial\mathcal{B}$ decomposed into two parts: $\partial\mathcal{B}_u$ and $\partial\mathcal{B}_t$ such that $\partial\mathcal{B} = \overline{\partial\mathcal{B}_u \cup \partial\mathcal{B}_t}$. Displacements are prescribed along $\partial\mathcal{B}_u$ and $\partial\mathcal{B}_t$ is subject to *uncertain*, prescribed traction forces. Under the assumption that the resulting deformations are infinitesimal and the pointwise stress and strain are related through a *linear* relationship, the deformation of the body is governed by the following system of partial differential equations

$$\begin{aligned}
\nabla\cdot\sigma(x,\,\boldsymbol{y})+b(x,\,\boldsymbol{y}) &= 0 && x\in\mathcal{B},\quad \boldsymbol{y}\in\boldsymbol{\Xi}\\
u(x,\,\boldsymbol{y}) &= \bar{u}(x) && x\in\partial\mathcal{B}_u\\
\sigma(x,\,\boldsymbol{y})\cdot n &= \bar{t}(x,\,\boldsymbol{y}) && x\in\partial\mathcal{B}_t, \boldsymbol{y}\in\boldsymbol{\Xi},
\end{aligned} \tag{2.58}$$

where u is the pointwise deformation and state vector of the PDE, σ is the stress tensor, b is the uncertain body force, $\bar{u}$ is the prescribed displacement on $\partial\mathcal{B}_u$, $\bar{t}$ is the uncertain, prescribed traction on $\partial\mathcal{B}_t$, and n is the pointwise outward normal to the boundary. The system of PDEs is closed with the stress-strain relationship (Hooke's law)

$$\sigma = \mathbb{C} : \epsilon \tag{2.59}$$

and the kinematic constraint relates deformation to strain as

$$\epsilon = \frac{1}{2}\left[\nabla u + \nabla u^T\right]. \tag{2.60}$$

The quantities of interest considered are the volume of the structure—a deterministic quantity since it is a geometrical quantity and all uncertainty is in the loading—and the expectation of the tracking functional

$$V = \int_{\mathcal{B}} dV \qquad \text{and} \qquad T(u) = \mathbb{E}\left[\frac{1}{2}\int_{\mathcal{B}}(u(x,\,\cdot\,)-\bar{u})_k(u(x,\,\cdot\,)-\bar{u})_k\,dV\right]. \tag{2.61}$$

2.2.3 Finite-Dimensional Approximation

Since analytical techniques cannot, in general, be used to solve parametrized stochastic differential equations in (2.47), discretization techniques are applied to reduce the continuous (differential)

form of the problem to a discrete (algebraic) form. Unlike the continuous formulation, the discrete problem can be solved using computational methods and resources. Two types of discretization must be applied to the SPDE in (2.47) to the yield a (sequence of) algebraic equations that are amenable numerical computation—-spatio-temporal and stochastic discretization.

Each realization of the SPDE in (2.47), i.e., for a given $\boldsymbol{y} \in \boldsymbol{\Xi}$, constitutes a deterministic parametrized partial differential equation and requires spatio-temporal discretization (only spatial discretization for *static* problems), e.g., such as those in Section 2.1.3, to yield a discrete problem. The specific spatial discretization technique is left unspecified since the appropriate choice depends on the properties of the differential operators $\mathcal{G}(\,\cdot\,,\,\cdot\,,\boldsymbol{\mu},\boldsymbol{y})$ and $\mathcal{H}(\,\cdot\,,\,\cdot\,,\boldsymbol{\mu},\boldsymbol{y})$. The semi-discrete form of the SPDE in (2.49) is: find $\boldsymbol{u}$ such that

$$\boldsymbol{r}(\boldsymbol{u},\boldsymbol{\mu},\boldsymbol{y}) = 0 \qquad \forall \boldsymbol{y} \in \boldsymbol{\Xi}. \tag{2.62}$$

A variant of the Implicit Function Theorem (Theorem 2.1) implies the existence of a continuous function $\boldsymbol{u}(\boldsymbol{\mu},\boldsymbol{y})$, defined implicitly as the solution of $\boldsymbol{r}(\,\cdot\,,\boldsymbol{\mu},\boldsymbol{y}) = 0$. The corresponding semi-discrete stochastic quantity of interest and its risk measure take the form

$$f(\boldsymbol{u},\boldsymbol{\mu},\boldsymbol{y}) \qquad \text{and} \qquad \mathcal{R}[f(\boldsymbol{u}(\boldsymbol{\mu},\,\cdot\,),\boldsymbol{\mu},\,\cdot\,)], \tag{2.63}$$

where $\mathcal{R}$ is any risk measure introduced in the previous section.

Despite the spatial discretization, the semi-discrete form of the SPDE in (2.62) can still not be treated computationally as the set $\boldsymbol{\Xi}$ contains infinitely many points. There are a few approaches, including stochastic Galerkin methods and stochastic collocation, to discretize the stochastic dimension and yield a fully discrete form of the SPDE that can be solved in a computational setting. This work uses stochastic collocation whereby the equation is (2.62) is enforced *only on a finite subset of* $\boldsymbol{\Xi}$, that is, (2.62) is replaced with

$$\boldsymbol{r}(\boldsymbol{u},\boldsymbol{\mu},\boldsymbol{y}) = 0 \qquad \forall \boldsymbol{y} \in \boldsymbol{\Xi}_h \tag{2.64}$$

where $\boldsymbol{\Xi}_h \subset \boldsymbol{\Xi}$ and $\text{card}(\boldsymbol{\Xi}_h) < \infty$. The integrals involved in the computation of the risk measure must then be approximated with a quadrature scheme with nodes $\boldsymbol{\Xi}_h$. Section 6.1.3 details an efficient method to construct $\boldsymbol{\Xi}_h$ using anisotropic sparse grids [67].

2.3 PDE-Constrained Optimization

Given the exposition on parametrized partial differential equations in the previous section, attention is turned to the main interest of this document: optimization problems governed by partial differential equations. There are three primary components required to define a PDE-constrained optimization problem:

- the governing partial differential equation and corresponding state vector that define the physical problem of interest,
- an objective function and constraint functions—the *goal* of the optimization problem—these are usually quantities of interest of the partial differential equation that define a performance measure to be optimized and design requirements, and
- optimization parameters—usually a control or design—that are used to meet the performance requirements.

Each of these components were discussed in the previous section, including details pertaining to their formulation and discretization, and concrete examples were provided. The remainder of this section will consider an abstract vector of parameters, $\boldsymbol{\mu} \in \mathbb{R}^{N_\mu}$, i.e., the parameter space has already been discretized and the discrete parameters can control any aspect of the PDE (shape/topology of the domain, boundary conditions, coefficient in differential operators). To encompass the wide array of features in the parametrized partial differential equations discussed previously, an abstract PDE of the form

$$\mathcal{D}(U, \boldsymbol{\mu}) = 0 \tag{2.65}$$

will be considered, where U is the state vector and $\mathcal{D}$ is the differential operator. The abstract quantity of interest will be denoted

$$\mathcal{F}(U, \boldsymbol{\mu}) \tag{2.66}$$

and will be the objective function or *cost functional* in the remainder of this section. This notation will encompass static and time-dependent, deterministic and stochastic PDEs from previous sections. In the static, deterministic case, U is understood to be only a function of space, i.e., $U = U(x)$, and $\mathcal{F}$ is likely an integral over a volume or surface. In the time-dependent, deterministic case, U is a function of space and time, i.e., $U = U(x, t)$, and $\mathcal{F}$ is a space-time integral. In the stochastic counterparts, U also depends on the realization, i.e., $U(\,\cdot\,) = U(\,\cdot\,, \boldsymbol{y})$, and $\mathcal{F}$ requires an integral over the stochastic space to compute the risk measure of the quantity of interest. The discretized PDE and QoI will be denoted

$$\boldsymbol{r}(\boldsymbol{u}, \boldsymbol{\mu}) = 0 \qquad \text{and} \qquad f(\boldsymbol{u}, \boldsymbol{\mu}), \tag{2.67}$$

respectively, where $\boldsymbol{u}$ is the discrete state vector. While this abstract framework would lead to a horribly inefficient implementation, it is useful to consider these various cases at once as the same issues and concepts regarding PDE-constrained optimization arise in all cases. Each case must be considered separately to obtain a formulation that will be efficient from an implementation viewpoint. The remainder of this chapter discusses important details in the formulation of PDE-constrained optimization problems and introduces concepts and notation that will be used throughout the remainder of this thesis.

2.3.1 Continuous vs. Discrete Formulation

The many steps involved in the discretization of partial differential equations provides a large degree of flexibility in the formulation of the PDE-constrained optimization problem. Namely, it can be formulated at the continuous level or at any stage in the discretization process and these will not, in general, be equivalent for finite values of the discretization parameter since the operations of differentiation and discretization do not commute.

The PDE-constrained optimization problem at the continuous level takes the form

$$\begin{aligned} \underset{U,\,\boldsymbol{\mu}}{\text{minimize}} \quad & \mathcal{F}(U,\,\boldsymbol{\mu}) \\ \text{subject to} \quad & \mathcal{D}(U,\,\boldsymbol{\mu}) = 0. \end{aligned} \tag{2.68}$$

The continuous formulation of the optimization problem, also known as the *differentiate-then-discretize* approach [78], proceeds by deriving the optimality conditions of (2.68), which leads to a system of partial differential equations that includes the primal and adjoint PDE and optimality condition. These PDEs are discretized using the methods introduced in the previous section and solved using an iterative method. Since this process is heavily dependent on the specific form of the PDE and QoI under consideration, a specific example is provided next.

Example 1 (Optimal control of Poisson's equation)**.** *Consider the optimal control problem that looks to find a distributed control $z(x)$ such that $u(x)$, the solution of the Poisson equation with homogeneous Dirichlet boundary conditions, matches a given target state $\bar{u}(x)$ with a penalty on the magnitude of the control. This problem is stated precisely as*

$$\begin{aligned} \underset{u(x),\,z(x)}{\textit{minimize}} \quad & \frac{1}{2}\int_\Omega (u(x)-\bar{u}(x))^2\,dx + \frac{\alpha}{2}\int_\Omega z(x)^2\,dx \\ \textit{subject to} \quad & -\Delta u(x) = z(x) \qquad x \in \Omega \\ & u(x) = 0 \qquad\quad x \in \partial\Omega. \end{aligned} \tag{2.69}$$

The Lagrangian of this PDE-constrained optimization problem is

$$\mathcal{L}(u,\,z,\,\lambda) = \frac{1}{2}\int_\Omega (u-\bar{u})^2\,dx + \frac{\alpha}{2}\int_\Omega z^2\,dx - \int_\Omega \lambda\left[-\Delta u - z\right]\,dx - \int_{\partial\Omega} \lambda u\,dx. \tag{2.70}$$

Any solution of (2.69) must render the Lagrangian stationary, i.e.,

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}\epsilon}\,\mathcal{L}(u+\epsilon\delta u,\,z,\,\lambda)|_{\epsilon=0} &= 0 \qquad \forall \delta u \\ \frac{\mathrm{d}}{\mathrm{d}\epsilon}\,\mathcal{L}(u,\,z+\epsilon\delta z,\,\lambda)|_{\epsilon=0} &= 0 \qquad \forall \delta z \\ \frac{\mathrm{d}}{\mathrm{d}\epsilon}\,\mathcal{L}(u,\,z,\,\lambda+\epsilon\delta\lambda)|_{\epsilon=0} &= 0 \qquad \forall \delta \lambda. \end{aligned} \tag{2.71}$$

After direct differentiation of the Lagrangian and subsequent integration-by-parts, the first condition

in (2.71) reduces to

$$\frac{\mathrm{d}}{\mathrm{d}\epsilon}\,\mathcal{L}(u+\epsilon\delta u,\,z,\,\lambda)|_{\epsilon=0} = \int_\Omega \left[-\Delta\lambda - (u-\bar{u})\right]\delta u\,dx - \int_{\partial\Omega}\lambda\left[\delta u + \nabla\delta u\cdot n\right]\,dx = 0 \tag{2.72}$$

Since this relation holds for all δu, it is equivalent to the following partial differential equation

$$\begin{aligned} -\Delta\lambda(x) &= u(x) - \bar{u}(x) && x\in\Omega \\ \lambda(x) &= 0 && x\in\partial\Omega, \end{aligned} \tag{2.73}$$

which is known as the adjoint *PDE. Direct differentiation of the Lagrangian (2.70) reduces the second condition in (2.71) to*

$$\frac{\mathrm{d}}{\mathrm{d}\epsilon}\,\mathcal{L}(u,\,z+\epsilon\delta z,\,\lambda)|_{\epsilon=0} = \int_\Omega(\alpha z+\lambda)\delta z\,dx = 0, \tag{2.74}$$

which is equivalent to the pointwise relationship

$$\lambda(x) = -\alpha z(x). \tag{2.75}$$

This is known as the optimality condition. *Finally, the last condition in (2.71) reduces to*

$$\frac{\mathrm{d}}{\mathrm{d}\epsilon}\,\mathcal{L}(u,\,z,\,\lambda+\epsilon\delta\lambda)|_{\epsilon=0} = \int_\Omega \delta\lambda\left[-\Delta u - z\right]\,dx - \int_{\partial\Omega}\delta\lambda u\,dx = 0 \tag{2.76}$$

and recovers the governing PDE since this holds for all $\delta\lambda$

$$\begin{aligned} -\Delta u(x) &= z(x) && x\in\Omega \\ u(x) &= 0 && x\in\partial\Omega. \end{aligned} \tag{2.77}$$

The adjoint PDE (2.73), optimality condition (2.75), and primal PDE (2.77) comprise the optimality system at the continuous level, known as the Karush-Kuhn-Tucker (KKT) conditions. Thus, the optimal control problem in (2.71) reduces to: find $u(x)$, $z(x)$, and $\lambda(x)$ such that

$$\begin{aligned} -\Delta u(x) &= z(x) && x\in\Omega \\ u(x) &= 0 && x\in\partial\Omega \\ -\Delta\lambda(x) &= u(x) - \bar{u}(x) && x\in\Omega \\ \lambda(x) &= 0 && x\in\partial\Omega \\ z(x) &= -\lambda(x)/\alpha && x\in\Omega. \end{aligned} \tag{2.78}$$

The derivation of the above KKT system at the continuous level is the first step in the differentiate-then-discretize approach to PDE-constrained optimization. The KKT system is solved by discretizing the parameters, quantities of interest, and primal and adjoint PDEs with the methods outlined in

Sections 2.1.2–2.1.4. This leads to a coupled system of equations that are solved to yield an approximation to (2.69). In general, different discretization methods and levels of refinement can be used for the primal and adjoint equations. This is one of the advantages of this approach compared to the discretize-then-differentiate approach discussed next [78].

The discrete formulation, also known as the *discretize-then-differentiate* approach [78], first discretizes the optimization problem in (2.68) to yield

$$\begin{aligned} \underset{\boldsymbol{u},\,\boldsymbol{\mu}}{\text{minimize}} \quad & f(\boldsymbol{u},\,\boldsymbol{\mu}) \\ \text{subject to} \quad & \boldsymbol{r}(\boldsymbol{u},\,\boldsymbol{\mu}) = 0. \end{aligned} \tag{2.79}$$

Subsequently, the optimality conditions of (2.79) are derived by introducing its Lagrangian

$$\mathcal{L}(\boldsymbol{u},\,\boldsymbol{\mu},\,\boldsymbol{\lambda}) = f(\boldsymbol{u},\,\boldsymbol{\mu}) - \boldsymbol{\lambda}^T \boldsymbol{r}(\boldsymbol{u},\,\boldsymbol{\mu}) \tag{2.80}$$

and requiring its stationarity, i.e., $(\boldsymbol{u},\,\boldsymbol{\mu},\,\boldsymbol{\lambda})$ such that

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{u}}(\boldsymbol{u},\,\boldsymbol{\mu},\,\boldsymbol{\lambda}) = 0 \qquad \frac{\partial \mathcal{L}}{\partial \boldsymbol{\mu}}(\boldsymbol{u},\,\boldsymbol{\mu},\,\boldsymbol{\lambda}) = 0 \qquad \frac{\partial \mathcal{L}}{\partial \boldsymbol{\lambda}}(\boldsymbol{u},\,\boldsymbol{\mu},\,\boldsymbol{\lambda}) = 0. \tag{2.81}$$

These are the Karush-Kuhn-Tucker (KKT) conditions [143] and lead to the coupled system of nonlinear algebraic equations

$$\begin{aligned} \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}(\boldsymbol{u},\,\boldsymbol{\mu})^T \boldsymbol{\lambda} &= \frac{\partial f}{\partial \boldsymbol{u}}(\boldsymbol{u},\,\boldsymbol{\mu})^T \\ \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{\mu}}(\boldsymbol{u},\,\boldsymbol{\mu})^T \boldsymbol{\lambda} &= \frac{\partial f}{\partial \boldsymbol{\mu}}(\boldsymbol{u},\,\boldsymbol{\mu})^T \\ \boldsymbol{r}(\boldsymbol{u},\,\boldsymbol{\mu}) &= 0, \end{aligned} \tag{2.82}$$

which are solved simultaneously.

The continuous formulation has a significant disadvantage in that it does not possess *discrete consistency* in the reduced space setting (Section 2.3.2), that is, the computed gradient is not the true gradient of the computed QoI since differentiation and discretization do not commute. This makes it difficult to use blackbox optimizers to solve the optimization problem as convergence may fail or be slowed when supplied with inconsistent gradients. Despite this disadvantage, there a number of advantages to the continuous formulation. Different discretizations can be used for the primal, sensitivity, and adjoint equations (either predefined or adaptively refined grids) depending on required resolution of each. In shape optimization problems, there is no need to account for the grid motion in the sensitivity and adjoint equations since they are posed directly on the new domain. Finally, this approach can naturally be embedded in an optimization framework that leverages and manages inexact gradients since error bounds on the computed sensitivities and adjoints are available [108, 109].

The discrete formulation has a number of advantages as compared to the continuous framework, most notable discrete consistency of computed functionals and gradients. Additionally, the discrete

formulation allows for the use of sophisticated differentiation software, such as automatic [161] and symbolic [126] differentiation, to compute the various quantities that arise in the sensitivity and adjoint equations. The discrete setting also allows for a large degree of flexibility in the quantities of interest and optimization parameters considered, particularly if a well-defined differentiation framework is used in the implementation. The continuous approach requires re-deriving the corresponding adjoint equations for each partial differential equation, boundary condition, and quantity of interest. For these reasons, the discrete formulation will be solely considered throughout the remainder of this thesis.

In certain situations, the continuous and discrete formulations of the optimization problem are equivalent. This equivalence holds if the scheme used to discretize the partial differential equation is *adjoint consistent*—that is, the discrete adjoint equations correspond to a consistent discretization of the continuous adjoint equations [11, 84]. This property is not crucial for this work since the discrete formulation is considered and therefore gradients automatically possess discrete consistency. However, it has been shown that an adjoint consistent discretization of the PDE is necessary for optimal convergence rates in L^2 and in quantities of interest [97, 83, 82].

2.3.2 Full Space vs. Reduced Space Approach

To this point, the PDE-constrained optimization problem has been posed as an optimization problem over the *state* and *parameter*. This is usually called a *full space* or *one-shot* formulation as the solution of the PDE and optimization problem are sought simultaneously. In contrast, the *reduced space* approach explicitly enforces the PDE constraint and considers an optimization problem over the *parameters* only. In the optimization community, this is commonly referred to as *nonlinear elimination* of equality constraints [71, 143].

To consider the reduced space approach to PDE-constrained optimization, the following assumption on existence and uniqueness of solutions of the PDE is crucial.

Assumption 2.2. *For any $\boldsymbol{\mu} \in \mathbb{R}^{N_\mu}$, there exists a unique $\boldsymbol{u}(\boldsymbol{\mu})$ such that $\boldsymbol{r}(\boldsymbol{u}(\boldsymbol{\mu}), \boldsymbol{\mu}) = 0$.*

From this assumption, it is clear that optimization of pure Neumann problems is not possible in the reduced space setting as the solutions of these problems are only unique up to a constant. For such problems, a full space setting is more appropriate. The implicit function theorem guarantees Assumption 2.2 holds if $\boldsymbol{r}$ is sufficiently regular and its Jacobian is invertible; in fact, it guarantees the existence of a smooth function that maps $\boldsymbol{\mu} \in \mathbb{R}^{N_\mu}$ to the corresponding solution of the PDE, $\boldsymbol{u}(\boldsymbol{\mu})$.

Theorem 2.1 (Implicit Function Theorem)**.** *Let A be an open set in $\mathbb{R}^{N_u} \times \mathbb{R}^{N_\mu}$ and suppose $\boldsymbol{r} : A \to \mathbb{R}^{N_u}$ is a C^r function ($r \geq 1$). Consider $\bar{\boldsymbol{u}} \in \mathbb{R}^{N_u}$ and $\bar{\boldsymbol{\mu}} \in \mathbb{R}^{N_\mu}$ such that $\boldsymbol{r}(\bar{\boldsymbol{u}}, \bar{\boldsymbol{\mu}}) = 0$ and $\frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}(\bar{\boldsymbol{u}}, \bar{\boldsymbol{\mu}})$ is invertible. Then, there exists a neighborhood $B \subset \mathbb{R}^{N_\mu}$ of $\bar{\boldsymbol{\mu}}$ and a unique C^r function $\boldsymbol{u} : \mathbb{R}^{N_\mu} \to \mathbb{R}^{N_u}$ such that $\bar{\boldsymbol{u}} = \boldsymbol{u}(\bar{\boldsymbol{\mu}})$ and $\boldsymbol{r}(\boldsymbol{u}(\boldsymbol{\mu}), \boldsymbol{\mu}) = 0$ for all $\boldsymbol{\mu} \in B$.*

This mapping from $\boldsymbol{\mu}$ to $\boldsymbol{u}(\boldsymbol{\mu})$ is used to define a quantity of interest that *only depends on* $\boldsymbol{\mu}$

$$F(\boldsymbol{\mu}) \coloneqq f(\boldsymbol{u}(\boldsymbol{\mu}), \boldsymbol{\mu}). \tag{2.83}$$

Thus, the constrained optimization problem in (2.79) can be reduced to the unconstrained optimization problem

$$\underset{\boldsymbol{\mu}\in\mathbb{R}^{N_{\boldsymbol{\mu}}}}{\text{minimize}} \quad F(\boldsymbol{\mu}) \tag{2.84}$$

since the solution of the PDE is fully accounted for in $\boldsymbol{u}(\boldsymbol{\mu})$. In a gradient-based optimization setting, the reduced space approach requires the computation of *gradients* of quantities of interest that account for the total dependence on $\boldsymbol{\mu}$, that is, the explicit dependence on $\boldsymbol{\mu}$ and the implicit dependence through the solution of the PDE itself. This will be the focus of the next two sections that consider two distinct approaches to obtaining such gradients.

There are a number of advantages of the reduced space approach over full space methods, particularly in the context of large-scale, practical problems. First, the optimization problem is smaller and simpler—it is only posed over the *parameters* since the state variable is taken as an implicit function of these parameters, $\boldsymbol{u}(\boldsymbol{\mu})$, and the nonlinearly constrained optimization problem is reduced to an unconstrained one. The reduced space framework also allows for the use of state-of-the-art PDE solvers and black-box optimizers since it decouples the solution of the PDE and the optimization problem. This is particularly important in the context of computational fluid dynamics where specialized methods exist for solving the steady-state partial differential equation such as pseudo-transient continuation [104, 105]. For these reasons, the remainder of this thesis will focus solely on the reduced space formulation of PDE-constrained optimization. The close this discussion, it is worthwhile to mention some advantages of the full space approach: (1) it does not require Assumption 2.2, thereby enlarging the class of problems to which it can be applied and (2) it is usually more efficient than the reduced space approach since it does not require full resolution of the PDE solution at every iteration.

2.3.3 Sensitivity Method for Computing Gradients

Once one commits to using a reduced space approach, the gradient of $F(\boldsymbol{\mu})$ in (2.84) must be computed, if a gradient-based optimization method is to be employed. This gradient must account for the explicit dependence of f on $\boldsymbol{\mu}$ as well as its implicit dependence through the solution of the PDE. Throughout the remainder of this chapter, $\boldsymbol{u}(\boldsymbol{\mu})$ will be used to denote the function in Theorem 2.1 that maps $\boldsymbol{\mu}$ to the solution of the PDE. Application of the chain rule leads to the expansion

$$\frac{\mathrm{d}F}{\mathrm{d}\boldsymbol{\mu}}(\boldsymbol{\mu}) = \frac{\partial f}{\partial \boldsymbol{\mu}}(\boldsymbol{u}(\boldsymbol{\mu}), \boldsymbol{\mu}) + \frac{\partial f}{\partial \boldsymbol{u}}(\boldsymbol{u}(\boldsymbol{\mu}), \boldsymbol{\mu})\frac{\partial \boldsymbol{u}}{\partial \boldsymbol{\mu}}(\boldsymbol{\mu}). \tag{2.85}$$

Furthermore, since $\boldsymbol{u}(\boldsymbol{\mu})$ is the solution of the PDE *for any* $\boldsymbol{\mu}$, $\boldsymbol{r}(\boldsymbol{u}(\boldsymbol{\mu}), \boldsymbol{\mu}) = 0$ and

$$\frac{\mathrm{d}\boldsymbol{r}}{\mathrm{d}\boldsymbol{\mu}}(\boldsymbol{u}(\boldsymbol{\mu}), \boldsymbol{\mu}) = 0 \quad \Longrightarrow \quad \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{\mu}}(\boldsymbol{u}(\boldsymbol{\mu}), \boldsymbol{\mu}) + \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}(\boldsymbol{u}(\boldsymbol{\mu}), \boldsymbol{\mu})\frac{\partial \boldsymbol{u}}{\partial \boldsymbol{\mu}}(\boldsymbol{\mu}) = 0 \tag{2.86}$$

From the assumptions in Theorem 2.1, the Jacobian matrix $\frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}(\boldsymbol{u}(\boldsymbol{\mu}), \boldsymbol{\mu})$ is invertible, which leads to the following expression for the *sensitivity*

$$\frac{\partial \boldsymbol{u}}{\partial \boldsymbol{\mu}}(\boldsymbol{\mu}) = -\frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}(\boldsymbol{u}(\boldsymbol{\mu}), \boldsymbol{\mu})^{-1} \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{\mu}}(\boldsymbol{u}(\boldsymbol{\mu}), \boldsymbol{\mu}). \tag{2.87}$$

Combining this equation for the sensitivity with the expression for the gradient of F leads to

$$\frac{\mathrm{d}F}{\mathrm{d}\boldsymbol{\mu}}(\boldsymbol{\mu}) = \frac{\partial f}{\partial \boldsymbol{\mu}}(\boldsymbol{u}(\boldsymbol{\mu}), \boldsymbol{\mu}) - \frac{\partial f}{\partial \boldsymbol{u}}(\boldsymbol{u}(\boldsymbol{\mu}), \boldsymbol{\mu}) \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}(\boldsymbol{u}(\boldsymbol{\mu}), \boldsymbol{\mu})^{-1} \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{\mu}}(\boldsymbol{u}(\boldsymbol{\mu}), \boldsymbol{\mu}). \tag{2.88}$$

This method of computing the gradient of F is known as the sensitivity or *direct* method. An important observation is that each column of the sensitivity matrix, $\frac{\partial \boldsymbol{u}}{\partial \boldsymbol{\mu}}$, requires the solution of a linear system of equation with the Jacobian matrix—a total of $N_{\boldsymbol{\mu}}$ linear systems with the same matrix and different right-hand sides. In large-scale applications, particularly for time-dependent problems, this will be a very expensive endeavor—see Appendix D for details regarding the sensitivity method for time-dependent problems. An advantage of the sensitivity method is that once the sensitivity, $\frac{\partial \boldsymbol{u}}{\partial \boldsymbol{\mu}}$ is computed, the gradient of *any* number of functionals can be computed essentially for free. This is useful if the problem has a large number of *side constraints*—see Section 2.3.5.

Before closing this discussion on sensitivity analysis, define the *sensitivity residual* as

$$\boldsymbol{r}^{\partial}(\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{\mu}) \coloneqq \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{\mu}}(\boldsymbol{u}, \boldsymbol{\mu}) + \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}(\boldsymbol{u}, \boldsymbol{\mu})\boldsymbol{v}, \tag{2.89}$$

which is motivated from the sensitivity equations in (2.86). Clearly, we have

$$\boldsymbol{r}^{\partial}\left(\boldsymbol{u}(\boldsymbol{\mu}), \frac{\partial \boldsymbol{u}}{\partial \boldsymbol{\mu}}(\boldsymbol{\mu}), \boldsymbol{\mu}\right) = 0.$$

The sensitivity residual will be used as an *error indicator* for any approximation $\boldsymbol{u}$, $\boldsymbol{w}$ of the true primal solution $\boldsymbol{u}(\boldsymbol{\mu})$ and sensitivity $\frac{\partial \boldsymbol{u}}{\partial \boldsymbol{\mu}}(\boldsymbol{\mu})$, as well as an error bound on the corresponding approximation of $\nabla F(\boldsymbol{\mu})$—see Appendix B. In a similar manner, the gradient computation in (2.88) is generalized to consider non-equilibrium solutions $\boldsymbol{u}$ and sensitivities $\boldsymbol{w}$

$$\boldsymbol{g}^{\partial}(\boldsymbol{u}, \boldsymbol{w}, \boldsymbol{\mu}) = \frac{\partial f}{\partial \boldsymbol{\mu}}(\boldsymbol{u}, \boldsymbol{\mu}) + \frac{\partial f}{\partial \boldsymbol{u}}(\boldsymbol{u}, \boldsymbol{\mu})\boldsymbol{w} \tag{2.90}$$

as this will play a role in the residual-based error bounds on QoIs (Appendix B). The next section introduces a method to compute $\nabla F(\boldsymbol{\mu})$—the adjoint method—that circumvents the large cost of the sensitivity approach when $N_{\boldsymbol{\mu}} \gg 1$.

2.3.4 Adjoint Method for Computing Gradients

The adjoint method is an alternative approach to compute the gradient of F that circumvents the sensitivity computation in (2.87) and only requires a single linear system solve with the transpose

of the Jacobian matrix to compute the entire gradient. In this section, three different derivations of the adjoint method will be provided, leading to various interpretations of the adjoint variable.

The first and simplest derivation of the adjoint method applies a simple algebraic trick to the gradient expression in (2.88) to yield

$$\frac{\mathrm{d}F}{\mathrm{d}\boldsymbol{\mu}} = \frac{\partial f}{\partial \boldsymbol{\mu}} - \frac{\partial f}{\partial \boldsymbol{u}} \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}^{-1} \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{\mu}} = \frac{\partial f}{\partial \boldsymbol{\mu}} - \left(\frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}^{-T} \frac{\partial f}{\partial \boldsymbol{u}}^{T} \right)^{T} \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{\mu}} = \frac{\partial f}{\partial \boldsymbol{\mu}} - \boldsymbol{\lambda}^{T} \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{\mu}} \tag{2.91}$$

where the arguments $\boldsymbol{u}(\boldsymbol{\mu})$ and $\boldsymbol{\mu}$ have been dropped for brevity and $\boldsymbol{\lambda}(\boldsymbol{\mu})$ is defined as the solution of

$$\frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}(\boldsymbol{u}(\boldsymbol{\mu}),\, \boldsymbol{\mu})^{T} \boldsymbol{\lambda}(\boldsymbol{\mu}) = \frac{\partial f}{\partial \boldsymbol{u}}(\boldsymbol{u}(\boldsymbol{\mu}),\, \boldsymbol{\mu})^{T}. \tag{2.92}$$

The linearized equations in (2.92) are known as the *adjoint equations* and $\boldsymbol{\lambda}$ is the adjoint or dual variable. From (2.91) and (2.92) it is clear the gradient of F can be computed from one linear system solve, regardless of $N_{\boldsymbol{\mu}}$.

The second derivation proceeds by introducing $\boldsymbol{\lambda}$ as an *arbitrary* test function, multiplying it by the sensitivity equations in (2.86), and adding the resulting expression to the equation for the gradient of F in (2.85)

$$\frac{\mathrm{d}F}{\mathrm{d}\boldsymbol{\mu}} = \frac{\partial f}{\partial \boldsymbol{\mu}} + \frac{\partial f}{\partial \boldsymbol{u}} \frac{\partial \boldsymbol{u}}{\partial \boldsymbol{\mu}} - \boldsymbol{\lambda}^{T} \left[\frac{\partial \boldsymbol{r}}{\partial \boldsymbol{\mu}} + \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}} \frac{\partial \boldsymbol{u}}{\partial \boldsymbol{\mu}} \right]. \tag{2.93}$$

This equation is valid since the term in brackets on the right side is identically zero from (2.86) and the fact that all terms are evaluated at the primal and sensitivity solutions. Recall the goal is to get an expression for $\dfrac{\mathrm{d}F}{\mathrm{d}\boldsymbol{\mu}}$ that is independent of the sensitivity $\dfrac{\partial \boldsymbol{u}}{\partial \boldsymbol{\mu}}$. To this end, the terms in (2.93) are re-arranged such that the sensitivity is isolated

$$\frac{\mathrm{d}F}{\mathrm{d}\boldsymbol{\mu}} = \frac{\partial f}{\partial \boldsymbol{\mu}} - \boldsymbol{\lambda}^{T} \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{\mu}} + \left[\frac{\partial f}{\partial \boldsymbol{u}} - \boldsymbol{\lambda}^{T} \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}} \right] \frac{\partial \boldsymbol{u}}{\partial \boldsymbol{\mu}}. \tag{2.94}$$

Define $\boldsymbol{\lambda}$, which has remained arbitrary to this point, as the solution of the adjoint equation

$$\frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}(\boldsymbol{u}(\boldsymbol{\mu}),\, \boldsymbol{\mu})^{T} \boldsymbol{\lambda}(\boldsymbol{\mu}) = \frac{\partial f}{\partial \boldsymbol{u}}(\boldsymbol{u}(\boldsymbol{\mu}),\, \boldsymbol{\mu})^{T} \tag{2.95}$$

and the expression in the brackets vanishes, leading to an expression for $\dfrac{\mathrm{d}F}{\mathrm{d}\boldsymbol{\mu}}$ that is independent of the sensitivities

$$\frac{\mathrm{d}F}{\mathrm{d}\boldsymbol{\mu}}(\boldsymbol{\mu}) = \frac{\partial f}{\partial \boldsymbol{\mu}}(\boldsymbol{u}(\boldsymbol{\mu}),\, \boldsymbol{\mu}) - \boldsymbol{\lambda}(\boldsymbol{\mu})^{T} \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{\mu}}(\boldsymbol{u}(\boldsymbol{\mu}),\, \boldsymbol{\mu}) \tag{2.96}$$

and agrees with (2.92).

The final derivation will introduce the adjoint variable as the *Lagrange multipliers* corresponding

to the PDE constraint of the *auxiliary* PDE-constrained optimization problem

$$\begin{aligned} \underset{\boldsymbol{u}}{\text{minimize}} \quad & f(\boldsymbol{u}, \hat{\boldsymbol{\mu}}) \\ \text{subject to} \quad & \boldsymbol{r}(\boldsymbol{u}, \hat{\boldsymbol{\mu}}) = 0, \end{aligned} \tag{2.97}$$

where $\hat{\boldsymbol{\mu}}$ is *fixed*, i.e., not an optimization variable. Assumption 2.2 implies that the optimization problem is *equivalent* to the nonlinear system of equation

$$\boldsymbol{r}(\boldsymbol{u}, \hat{\boldsymbol{\mu}}) = 0.$$

This follows directly from $\hat{\boldsymbol{\mu}}$ being fixed and uniqueness of the solution of $\boldsymbol{r}(\cdot, \boldsymbol{\mu}) = 0$, i.e., the feasible set of the optimization problem in (2.97) is $\{\boldsymbol{u}(\hat{\boldsymbol{\mu}})\}$ and therefore $\boldsymbol{u}(\hat{\boldsymbol{\mu}})$ must be the solution of (2.97), regardless of the objective function. The Lagrangian of the optimization problem in (2.97) is

$$\mathcal{L}(\boldsymbol{u}, \boldsymbol{\lambda}) = f(\boldsymbol{u}, \hat{\boldsymbol{\mu}}) - \boldsymbol{\lambda}^T \boldsymbol{r}(\boldsymbol{u}, \hat{\boldsymbol{\mu}}) \tag{2.98}$$

and the KKT system is

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial \boldsymbol{u}} &= \frac{\partial f}{\partial \boldsymbol{u}}(\boldsymbol{u}, \hat{\boldsymbol{\mu}}) - \boldsymbol{\lambda}^T \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}(\boldsymbol{u}, \hat{\boldsymbol{\mu}}) = 0 \\ \frac{\partial \mathcal{L}}{\partial \boldsymbol{\lambda}} &= -\boldsymbol{r}(\boldsymbol{u}, \hat{\boldsymbol{\mu}}) \qquad\qquad\qquad\quad = 0 \end{aligned} \tag{2.99}$$

The first condition is exactly the adjoint equations in (2.92) and the second condition is the PDE constraint. Substitution into (2.94) yields the familiar expression for $\dfrac{\mathrm{d}F}{\mathrm{d}\boldsymbol{\mu}}$

$$\frac{\mathrm{d}F}{\mathrm{d}\boldsymbol{\mu}}(\boldsymbol{\mu}) = \frac{\partial f}{\partial \boldsymbol{\mu}}(\boldsymbol{u}(\boldsymbol{\mu}), \boldsymbol{\mu}) - \boldsymbol{\lambda}(\boldsymbol{\mu})^T \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{\mu}}(\boldsymbol{u}(\boldsymbol{\mu}), \boldsymbol{\mu}). \tag{2.100}$$

Thus, the adjoint variable has been introduced as an algebraic trick to re-arrange the operations in (2.88), a test function multiplying the sensitivity equations, and the Lagrange multipliers of an auxiliary PDE-constrained optimization problem. Appendix D details the derivation of the adjoint equations—using the test function and Lagrange multiplier approach—for a time-dependent PDE posed on a deforming domain and discretized with high-order spatial and temporal schemes. Similar to the previous section, the adjoint equations in (2.92) are used to motivate the definition of the *adjoint residual*

$$\boldsymbol{r}^{\boldsymbol{\lambda}}(\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{\mu}) \coloneqq \frac{\partial f}{\partial \boldsymbol{u}}(\boldsymbol{u}, \boldsymbol{\mu})^T - \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}(\boldsymbol{u}, \boldsymbol{\mu})^T \boldsymbol{v}, \tag{2.101}$$

which will be used as an error measure when inexact primal and adjoint solution are used to compute $\nabla F(\boldsymbol{\mu})$—see Appendix B. In a similar manner, the gradient computation in (2.91) is generalized to consider non-equilibrium solutions $\boldsymbol{u}$ and adjoints $\boldsymbol{z}$

$$\boldsymbol{g}^{\boldsymbol{\lambda}}(\boldsymbol{u}, \boldsymbol{z}, \boldsymbol{\mu}) = \frac{\partial f}{\partial \boldsymbol{\mu}}(\boldsymbol{u}, \boldsymbol{\mu}) - \boldsymbol{z}^T \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{\mu}}(\boldsymbol{u}, \boldsymbol{\mu}) \tag{2.102}$$

as this will play a role in the residual-based error bounds on QoI gradients (Appendix B).

2.3.5 Optimization Problems with Side Constraints

To this point, *unconstrained* PDE-constrained optimization problems, i.e, optimization problems where the PDE is the only constraint, have been solely considered due to simplicity in the exposition. Nearly all practical problems, particularly in a design setting, will have additional performance constraints, usually referred to as side constraints. In this case, the fully discrete optimization problem in the full space takes the form

$$\begin{aligned} \underset{\boldsymbol{u},\,\boldsymbol{\mu}}{\text{minimize}} \quad & f(\boldsymbol{u},\,\boldsymbol{\mu}) \\ \text{subject to} \quad & \boldsymbol{r}(\boldsymbol{u},\,\boldsymbol{\mu}) = 0 \\ & \boldsymbol{c}(\boldsymbol{u},\,\boldsymbol{\mu}) = 0 \\ & \boldsymbol{d}(\boldsymbol{u},\,\boldsymbol{\mu}) \leq 0, \end{aligned} \tag{2.103}$$

where $\boldsymbol{c}$ and $\boldsymbol{d}$ are equality and inequality side constraints, respectively. In a gradient-based optimization framework, the terms

$$\frac{\partial \boldsymbol{c}}{\partial \boldsymbol{u}}, \frac{\partial \boldsymbol{c}}{\partial \boldsymbol{\mu}}, \frac{\partial \boldsymbol{d}}{\partial \boldsymbol{u}}, \frac{\partial \boldsymbol{d}}{\partial \boldsymbol{\mu}}$$

are required in addition to

$$\frac{\partial f}{\partial \boldsymbol{u}}, \frac{\partial f}{\partial \boldsymbol{\mu}}, \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}, \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{\mu}},$$

the terms required in the case without side constraints. In the reduced space setting, the optimization problem becomes

$$\begin{aligned} \underset{\boldsymbol{\mu}}{\text{minimize}} \quad & F(\boldsymbol{\mu}) \\ \text{subject to} \quad & \boldsymbol{C}(\boldsymbol{\mu}) = 0 \\ & \boldsymbol{D}(\boldsymbol{\mu}) \leq 0, \end{aligned} \tag{2.104}$$

where $F(\boldsymbol{\mu})$ is defined in (2.83) and

$$\boldsymbol{C}(\boldsymbol{\mu}) \coloneqq \boldsymbol{c}(\boldsymbol{u}(\boldsymbol{\mu}),\,\boldsymbol{\mu}) \qquad \text{and} \qquad \boldsymbol{D}(\boldsymbol{\mu}) \coloneqq \boldsymbol{d}(\boldsymbol{u}(\boldsymbol{\mu}),\,\boldsymbol{\mu}). \tag{2.105}$$

This is a nonlinearly constrained optimization problem over $\boldsymbol{\mu}$ and a gradient-based optimization setting will require the Jacobians of the constraints

$$\frac{\mathrm{d}\boldsymbol{C}}{\mathrm{d}\boldsymbol{\mu}}, \frac{\mathrm{d}\boldsymbol{D}}{\mathrm{d}\boldsymbol{\mu}},$$

which can be computed using the sensitivity or adjoint approach discussed previously. In the case where one of the constraints *does not* depend on the state vector $\boldsymbol{u}$, the sensitivity/adjoint method are not needed as the gradient will be equivalent to the partial derivative with respect to $\boldsymbol{\mu}$. If the sensitivity approach is used, the sensitivity $\dfrac{\partial \boldsymbol{u}}{\partial \boldsymbol{\mu}}(\boldsymbol{\mu})$ is computed once-and-for-all and used to

reconstruct the required gradients as

$$\begin{aligned}\frac{\mathrm{d}F}{\mathrm{d}\boldsymbol{\mu}}(\boldsymbol{\mu}) &= \frac{\partial f}{\partial \boldsymbol{\mu}}(\boldsymbol{u}(\boldsymbol{\mu}),\, \boldsymbol{\mu}) + \frac{\partial f}{\partial \boldsymbol{u}}(\boldsymbol{u}(\boldsymbol{\mu}),\, \boldsymbol{\mu})\frac{\partial \boldsymbol{u}}{\partial \boldsymbol{\mu}}(\boldsymbol{\mu}) \\ \frac{\mathrm{d}\boldsymbol{C}}{\mathrm{d}\boldsymbol{\mu}}(\boldsymbol{\mu}) &= \frac{\partial \boldsymbol{c}}{\partial \boldsymbol{\mu}}(\boldsymbol{u}(\boldsymbol{\mu}),\, \boldsymbol{\mu}) + \frac{\partial \boldsymbol{c}}{\partial \boldsymbol{u}}(\boldsymbol{u}(\boldsymbol{\mu}),\, \boldsymbol{\mu})\frac{\partial \boldsymbol{u}}{\partial \boldsymbol{\mu}}(\boldsymbol{\mu}) \\ \frac{\mathrm{d}\boldsymbol{D}}{\mathrm{d}\boldsymbol{\mu}}(\boldsymbol{\mu}) &= \frac{\partial \boldsymbol{d}}{\partial \boldsymbol{\mu}}(\boldsymbol{u}(\boldsymbol{\mu}),\, \boldsymbol{\mu}) + \frac{\partial \boldsymbol{d}}{\partial \boldsymbol{u}}(\boldsymbol{u}(\boldsymbol{\mu}),\, \boldsymbol{\mu})\frac{\partial \boldsymbol{u}}{\partial \boldsymbol{\mu}}(\boldsymbol{\mu})\end{aligned} \tag{2.106}$$

Thus, even though the sensitivity computation requires a linear solve for each entry in $\boldsymbol{\mu}$, it is used to construct the gradient of any number of functionals and is efficient when the number of constraints is large compared to $N_{\boldsymbol{\mu}}$.

Conversely, the adjoint equation is tied to a specific functional and each separate constraint requires the solution of a different adjoint equation

$$\begin{aligned}\frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}(\boldsymbol{u}(\boldsymbol{\mu}),\, \boldsymbol{\mu})^T \boldsymbol{\lambda}_f(\boldsymbol{\mu}) &= \frac{\partial f}{\partial \boldsymbol{u}}(\boldsymbol{u}(\boldsymbol{\mu}),\, \boldsymbol{\mu})^T \\ \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}(\boldsymbol{u}(\boldsymbol{\mu}),\, \boldsymbol{\mu})^T \boldsymbol{\lambda}_{\boldsymbol{c}}(\boldsymbol{\mu}) &= \frac{\partial \boldsymbol{c}}{\partial \boldsymbol{u}}(\boldsymbol{u}(\boldsymbol{\mu}),\, \boldsymbol{\mu})^T \\ \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}(\boldsymbol{u}(\boldsymbol{\mu}),\, \boldsymbol{\mu})^T \boldsymbol{\lambda}_{\boldsymbol{d}}(\boldsymbol{\mu}) &= \frac{\partial \boldsymbol{d}}{\partial \boldsymbol{u}}(\boldsymbol{u}(\boldsymbol{\mu}),\, \boldsymbol{\mu})^T.\end{aligned} \tag{2.107}$$

Once the dual variable for each functional has been computed, the required derivatives are reconstructed as

$$\begin{aligned}\frac{\mathrm{d}F}{\mathrm{d}\boldsymbol{\mu}}(\boldsymbol{\mu}) &= \frac{\partial f}{\partial \boldsymbol{\mu}}(\boldsymbol{u}(\boldsymbol{\mu}),\, \boldsymbol{\mu}) - \boldsymbol{\lambda}_f(\boldsymbol{\mu})^T \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{\mu}}(\boldsymbol{u}(\boldsymbol{\mu}),\, \boldsymbol{\mu}) \\ \frac{\mathrm{d}\boldsymbol{C}}{\mathrm{d}\boldsymbol{\mu}}(\boldsymbol{\mu}) &= \frac{\partial \boldsymbol{c}}{\partial \boldsymbol{\mu}}(\boldsymbol{u}(\boldsymbol{\mu}),\, \boldsymbol{\mu}) - \boldsymbol{\lambda}_{\boldsymbol{c}}(\boldsymbol{\mu})^T \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{\mu}}(\boldsymbol{u}(\boldsymbol{\mu}),\, \boldsymbol{\mu}) \\ \frac{\mathrm{d}\boldsymbol{D}}{\mathrm{d}\boldsymbol{\mu}}(\boldsymbol{\mu}) &= \frac{\partial \boldsymbol{d}}{\partial \boldsymbol{\mu}}(\boldsymbol{u}(\boldsymbol{\mu}),\, \boldsymbol{\mu}) - \boldsymbol{\lambda}_{\boldsymbol{d}}(\boldsymbol{\mu})^T \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{\mu}}(\boldsymbol{u}(\boldsymbol{\mu}),\, \boldsymbol{\mu}).\end{aligned} \tag{2.108}$$

Although it is not common, certain cases arise where it is possible to use nonlinear elimination to remove side constraints, identical to elimination of the PDE constraint in the reduced space approach. In these cases, for each $\boldsymbol{\mu}$, there must exist a $\boldsymbol{u}(\boldsymbol{\mu})$ that satisfies the PDE and side constraint. In general, this mapping will be different from the one defined in Theorem 2.1 and will modify the sensitivity and adjoint equations. Appendix D provides a concrete example of a time-dependent PDE-constrained optimization problem with two side constraints—the first is a lower bound on a QoI (not amenable to elimination) and the second requires time-periodicity of the PDE solution (amenable to elimination). Nonlinear elimination is applied to the periodicity constraint and the adjoint equations are modified accordingly.

Chapter 3

Generalized Multifidelity Trust Region Method

Given the broad discussion on partial differential equations and PDE-constrained optimization in Chapter 2, the scope will be narrowed to consider only the *reduced-space* framework for the remainder of the document. In this setting, nonlinear elimination is used to explicitly enforce the PDE constraint and eliminate the state variables from the optimization problem. This leads to an unconstrained or constrained optimization problem, depending on the presence of side constraints, over only the *parameters*, $\boldsymbol{\mu}$. Each query to the objective or constraint requires a primal PDE solve and each query to the corresponding gradient requires (possibly many) sensitivity or adjoint PDE solves. For PDEs with uncertain coefficients, an ensemble of primal and dual PDE solves are required to evaluate these optimization functionals and gradients (in order to evaluate risk-averse measures of quantities of interest). For large-scale problems that commonly arise in engineering and scientific practice, this will be an expensive endeavor. To mitigate this computational burden, a globally convergent optimization method is developed that enables the use of inexpensive, locally accurate approximation models. This chapter will develop the multifidelity optimization method with an *abstract* approximation model for the sake of generality, i.e., any approximation model that satisfies the assumptions to be laid out. Chapters 5–6 will detail the use of projection-based reduced-order models as the approximation model.

This chapter begins with necessary background regarding unconstrained optimization theory. Subsequently, an error-aware multifidelity trust region method—one of the auxiliary contributions of this thesis—is developed. Finally, the special case of an unconstrained problem, i.e., no side constraints, is generalized to handle nonlinear equality constraints.

3.1 Unconstrained Optimization

Consider the unconstrained optimization of a twice-continuously differentiable function $F : \mathbb{R}^{N_\mu} \to \mathbb{R}$ that is bounded below, i.e., $F \in \{g \in C^2(\mathbb{R}^{N_\mu}) \mid \inf g(\boldsymbol{\mu}) > -\infty\}$, stated as

$$\underset{\boldsymbol{\mu} \in \mathbb{R}^{N_\mu}}{\text{minimize}} \quad F(\boldsymbol{\mu}). \tag{3.1}$$

This is the exact form of the reduced-space PDE-constrained optimization problem in (2.84). In general, it is desirable to find the global minimum of (3.1), i.e., the point $\boldsymbol{\mu}^*$ such that $F(\boldsymbol{\mu}^*) \leq F(\boldsymbol{\mu})$ for all $\boldsymbol{\mu} \in \mathbb{R}^{N_\mu}$; however, it is impossible to construct an efficient and reliable global optimization algorithm for an arbitrary nonlinear function and we settle for *local minima*, as defined in Definition 3.1.

Definition 3.1 (Unconstrained local minima)**.** *A point $\boldsymbol{\mu}^* \in \mathbb{R}^{N_\mu}$ is a* local minima *of F if there is a neighborhood $\mathcal{N}$ of $\boldsymbol{\mu}^*$ such that $F(\boldsymbol{\mu}^*) \leq F(\boldsymbol{\mu})$ for all $\boldsymbol{\mu} \in \mathcal{N}$.*

From Theorem 3.1, if a point $\boldsymbol{\mu}^* \in \mathbb{R}^N$ is a local minima of (3.1), it must be a stationary point (Definition 3.2) of the function F. This is known as a *first-order* condition since it places a requirement on the *gradient* of F.

Theorem 3.1 (First-order unconstrained optimality conditions)**.** *If $\boldsymbol{\mu}^*$ is a local minimizer of $F(\boldsymbol{\mu})$ and F is continuously differentiable in a neighborhood of $\boldsymbol{\mu}^*$, then*

$$\nabla F(\boldsymbol{\mu}^*) = 0. \tag{3.2}$$

Proof. See [143] □

Definition 3.2 (Unconstrained stationary point)**.** *Any point $\boldsymbol{\mu}$ that satisfies $\nabla F(\boldsymbol{\mu}) = 0$ is called a* stationary point.

There are also second-order necessary and sufficient conditions for $\boldsymbol{\mu}^*$ to be a local minima of (3.1) that involve (semi-)positive definiteness of the *Hessian* of F [143]. This work will primarily be concerned with first-order optimality conditions.

3.1.1 Error-Aware Multifidelity Trust Region Method

In this section, we consider optimization problems of the form (3.1) where the evaluation of F and its gradient are expensive and look to develop an optimization algorithm that leverages an *inexpensive* approximation model, $m_k(\boldsymbol{\mu})$, at iteration k. It is assumed that evaluation of $m_k(\boldsymbol{\mu})$ and its gradient are substantially less expensive than the corresponding operation with $F(\boldsymbol{\mu})$. The approximation model $m_k(\boldsymbol{\mu})$ is required to be locally accurate around the kth iterate, $\boldsymbol{\mu}_k$, but may be inaccurate away from this point. An inexpensive optimization procedure involving the approximation model is intended to improve the current iterate and make progress toward the optimal solution. Due to

the inherent locality of the approximation model, it will not suffice to consider the unconstrained optimization problem

$$\underset{\boldsymbol{\mu}\in\mathbb{R}^{N_{\boldsymbol{\mu}}}}{\text{minimize}} \quad m_k(\boldsymbol{\mu}) \tag{3.3}$$

as the new iterate $\boldsymbol{\mu}_{k+1}$ may fail to make progress toward the local minima of (3.1). For this reason, the optimization problem is only posed within a trust region, defined as the sublevel sets of a function $\vartheta_k : \mathbb{R}^{N_{\boldsymbol{\mu}}} \to \mathbb{R}_+$, i.e.,

$$\begin{aligned} \underset{\boldsymbol{\mu}\in\mathbb{R}^{N_{\boldsymbol{\mu}}}}{\text{minimize}} \quad & m_k(\boldsymbol{\mu}) \\ \text{subject to} \quad & \vartheta_k(\boldsymbol{\mu}) \leq \Delta_k. \end{aligned} \tag{3.4}$$

In traditional trust region methods [48], the model is taken as the quadratic approximation of F at $\boldsymbol{\mu}_k$ and the trust region constraint is the Euclidean distance from $\boldsymbol{\mu}_k$

$$\begin{aligned} m_k(\boldsymbol{\mu}) &= F(\boldsymbol{\mu}_k) + \nabla F(\boldsymbol{\mu}_k)(\boldsymbol{\mu} - \boldsymbol{\mu}_k) + \frac{1}{2}(\boldsymbol{\mu} - \boldsymbol{\mu}_k)^T \nabla^2 F(\boldsymbol{\mu}_k)(\boldsymbol{\mu} - \boldsymbol{\mu}_k) \\ \vartheta_k(\boldsymbol{\mu}) &= \|\boldsymbol{\mu} - \boldsymbol{\mu}_k\|_2 , \end{aligned}$$

A plethora of variants have been proposed that leverage inexact gradients and Hessians in the definition of $m_k(\boldsymbol{\mu})$ [189, 35, 48, 108] and non-quadratic model objectives [4, 48, 10, 108]. In this work, the trust region constraint itself is generalized such that error bounds between the objective function and approximation model can be directly leveraged. This will, in a sense, define an *error-aware* trust region.

Before proceeding to the statement of the complete generalized trust region algorithm, an interpretation of an error-aware trust region is provided for a special case. Suppose the scalar-valued constraint function $\vartheta_k : \mathbb{R}^{N_{\boldsymbol{\mu}}} \to \mathbb{R}_+$ is defined as the Euclidean norm of a *linear* vector-valued error indicator $\boldsymbol{\vartheta}_k : \mathbb{R}^{N_{\boldsymbol{\mu}}} \to \mathbb{R}^m$, i.e.,

$$\vartheta_k(\boldsymbol{\mu}) = \|\boldsymbol{\vartheta}_k(\boldsymbol{\mu})\|_2 .$$

Additionally, suppose the approximation model is exact at trust region centers and this is reflected in the vector-valued error indicator ($\boldsymbol{\vartheta}_k(\boldsymbol{\mu}_k) = \mathbf{0}$), i.e.,

$$\boldsymbol{\vartheta}_k(\boldsymbol{\mu}) = \boldsymbol{A}_k(\boldsymbol{\mu} - \boldsymbol{\mu}_k),$$

where $\boldsymbol{A}_k \in \mathbb{R}^{m\times N_{\boldsymbol{\mu}}}$ is a fixed matrix. Then the constraint function can be expanded as

$$\vartheta(\boldsymbol{\mu}) = \|\boldsymbol{\vartheta}_k(\boldsymbol{\mu})\|_2 = \|\boldsymbol{\mu} - \boldsymbol{\mu}_k\|_{\boldsymbol{A}_k^T\boldsymbol{A}_k} , \tag{3.5}$$

which is precisely a traditional trust region constraint in the $\boldsymbol{A}_k^T\boldsymbol{A}_k$-norm. Consider the eigenvalue decomposition of the symmetric positive (semi)-definite matrix $\boldsymbol{A}_k^T\boldsymbol{A}_k$

$$\boldsymbol{A}_k^T\boldsymbol{A}_k = \boldsymbol{Q}_k\boldsymbol{\Lambda}_k\boldsymbol{Q}_k^T, \tag{3.6}$$

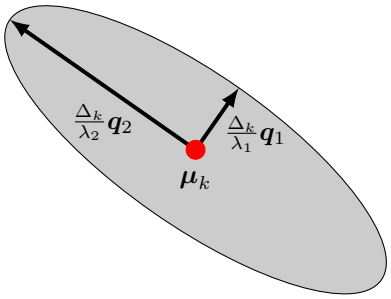


Figure 3.1: Geometry of trust region constraint in special case where $\vartheta_k = \|\boldsymbol{A}_k(\boldsymbol{\mu} - \boldsymbol{\mu}_k)\|_2 = \|\boldsymbol{\mu} - \boldsymbol{\mu}_k\|_{\boldsymbol{A}_k^T\boldsymbol{A}_k}$. The eigenvalue decomposition of $\boldsymbol{A}_k^T\boldsymbol{A}_k$ is $\boldsymbol{A}_k^T\boldsymbol{A}_k = \boldsymbol{Q}_k\boldsymbol{\Lambda}_k\boldsymbol{Q}_k^T$ with eigenvectors $\boldsymbol{q}_i = \boldsymbol{Q}_k\boldsymbol{e}_i$ and eigenvalues $\lambda_i = \boldsymbol{e}_i^T\boldsymbol{\Lambda}_k\boldsymbol{e}_i$.

where $\boldsymbol{Q}_k$ is an orthogonal matrix of eigenvectors of $\boldsymbol{A}_k^T\boldsymbol{A}_k$ and $\boldsymbol{\Lambda}_k$ is the diagonal matrix of non-negative eigenvalues. The trust region constraint $\vartheta_k(\boldsymbol{\mu}) \leq \Delta_k$ with $\vartheta_k(\boldsymbol{\mu})$ defined in (3.5) is an ellipse with principal axis directions $\boldsymbol{q}_i = \boldsymbol{Q}_k\boldsymbol{e}_i$ and lengths $\dfrac{\Delta_k}{\boldsymbol{e}_i^T\boldsymbol{\Lambda}\boldsymbol{e}_i}$ for $i = 1, \ldots, m$, where $\boldsymbol{e}_i \in \mathbb{R}^{N_{\boldsymbol{\mu}}}$ is the ith canonical vector; see Figure 3.1. Thus the ellipse is stretched (compressed) in directions corresponding small (large) eigenvalues. The matrix $\boldsymbol{A}_k^T\boldsymbol{A}_k$ represents the sensitivity of the error indicator with respect to the components of $\boldsymbol{\mu}$, which provides intuition to the ellipse interpretation: directions where the error indicator is highly sensitive to perturbations (large eigenvalues) correspond to small principal axes and vice versa.

For the sake of both generality and efficiency, the proposed generalized trust region method will allow the model gradient to be *inexact* at trust region centers, $\boldsymbol{\mu}_k$. Aside from the standard assumption imposed on the model function, m_k, such as twice-continuous differentiability and uniformly bounded Hessians, the proposed method requires the existence of functions $\vartheta_k : \mathbb{R}^{N_{\boldsymbol{\mu}}} \to \mathbb{R}_+$, $\varphi_k : \mathbb{R}^{N_{\boldsymbol{\mu}}} \to \mathbb{R}_+$ and *arbitrary* constants $\zeta, \xi > 0$ such that

$$\begin{aligned} |F(\boldsymbol{\mu}_k) - F(\boldsymbol{\mu}) + m_k(\boldsymbol{\mu}) - m_k(\boldsymbol{\mu}_k)| &\leq \zeta\vartheta_k(\boldsymbol{\mu}) \qquad \forall \boldsymbol{\mu} \in \mathcal{R}_k \\ \|\nabla F(\boldsymbol{\mu}_k) - \nabla m_k(\boldsymbol{\mu}_k)\| &\leq \xi\varphi_k(\boldsymbol{\mu}_k) \end{aligned} \tag{3.7}$$

where $\mathcal{R}_k := \{\boldsymbol{\mu} \in \mathbb{R}^{N_{\boldsymbol{\mu}}} \mid \vartheta_k(\boldsymbol{\mu}) \leq \Delta_k\}$. The first bound in (3.7) requires the *variation* of m_k from $\boldsymbol{\mu}_k$ to $\boldsymbol{\mu}$ to be related to the variation of F from $\boldsymbol{\mu}_k$ to $\boldsymbol{\mu}$. The does not necessarily place a requirement on the pointwise accuracy of m_k with respect to F, even at the trust region center $\boldsymbol{\mu}_k$. However, a requirement on pointwise accuracy is sufficient to lead to the bound in (3.7) as follows. Suppose there exists a function $\chi_k : \mathbb{R}^{N_{\boldsymbol{\mu}}} \to \mathbb{R}_+$ and arbitrary constant $\kappa > 0$ such that

$$|F(\boldsymbol{\mu}) - m_k(\boldsymbol{\mu})| \leq \kappa\chi_k(\boldsymbol{\mu}). \tag{3.8}$$

A simple application of the triangle inequality lead to the first bound in (3.7) with $\zeta = \kappa$ and $\vartheta_k(\boldsymbol{\mu}) = \chi_k(\boldsymbol{\mu}_k) + \chi_k(\boldsymbol{\mu})$. The second bound in (3.7) is a requirement on the gradient accuracy

at the trust region center. The existence of the *arbitrary constants* implies $\vartheta_k(\boldsymbol{\mu})$ and $\varphi_k(\boldsymbol{\mu})$ are *asymptotic* error bounds, which will provide considerable flexibility in deriving explicit expressions for them, even for *general* PDE-constrained optimization problems, in Chapters 5–6 when specific approximation models are considered.

Each iteration k of the proposed trust region method will rely on four main steps: (1) definition of the trust region model, $m_k(\boldsymbol{\mu})$, and constraint, $\vartheta_k(\boldsymbol{\mu})$, (2) computation of a candidate point, $\hat{\boldsymbol{\mu}}_k$, for the next iterate as the solution of the optimization problem (3.4), (3) computation of the ratio of the actual reduction realized by $\hat{\boldsymbol{\mu}}_k$ to that predicted by the model

$$\rho_k = \frac{F(\boldsymbol{\mu}_k) - F(\hat{\boldsymbol{\mu}}_k)}{m_k(\boldsymbol{\mu}_k) - m_k(\hat{\boldsymbol{\mu}}_k)}, \tag{3.9}$$

and (4) using the value of ρ_k, decide whether to accept or reject the candidate, $\hat{\boldsymbol{\mu}}_k$, and how to modify the trust region radius, Δ_k. Each of these steps will be detailed in the sections to follow. The generalized trust region algorithm that incorporates these steps is summarized in Algorithm 1. A proof of global convergence, that is, convergence to a local minima from any starting point $\boldsymbol{\mu}_0$, is provided in Appendix A. The computation of the actual-to-predicted reduction ratio (ρ_k) is a severe bottleneck of Algorithm 1 since it requires an evaluation of F. Another approximation model will be introduced to enable an approximation of ρ_k to be used in place of the true value without destroying global convergence. Therefore, the modified trust region method, summarized in Algorithm 2, circumvents the primary bottleneck of Algorithm 1.

Step 1: Model and constraint update

The first and most important step in an iteration of the generalized trust region method is the definition of the model function, $m_k(\boldsymbol{\mu})$, and constraint, $\vartheta_k(\boldsymbol{\mu})$. To guarantee global convergence, the model must be equipped with error bounds of the form (3.7) and conditions must be placed on the value of the error indicators, $\vartheta_k(\boldsymbol{\mu})$ and $\varphi_k(\boldsymbol{\mu})$, at trust region centers to control the quality of the approximation. The requirement on $\vartheta_k(\boldsymbol{\mu}_k)$ is simply

$$\vartheta_k(\boldsymbol{\mu}_k) \leq \kappa_\vartheta \Delta_k \tag{3.10}$$

where $0 < \kappa_\vartheta < 1$ is an algorithmic constant, which ensures the feasible set of (3.4) is not empty and the trust region center ($\boldsymbol{\mu}_k$) is in the feasible set. In the special case where $\vartheta_k(\boldsymbol{\mu})$ is a pointwise error indicator of the form $\chi_k(\boldsymbol{\mu}_k) + \chi_k(\boldsymbol{\mu})$, it places a requirement on the accuracy of the model at the trust region center. In the next section and Lemma A.1, this condition will also be used to circumscribe a traditional trust region feasible set (with modified radius) inside the feasible set of (3.4), which enables standard results from trust region theory to be recycled.

The requirement on $\varphi_k(\boldsymbol{\mu}_k)$ is recycled from [93, 108, 109]

$$\varphi_k(\boldsymbol{\mu}_k) \leq \kappa_\varphi \min\{\|\nabla m_k(\boldsymbol{\mu}_k)\|, \Delta_k\} \tag{3.11}$$

Algorithm 1 Error-aware multifidelity trust region method with exact objective evaluations

1: **Initialization**: Given

$$\boldsymbol{\mu}_0,\ \Delta_0,\ 0<\gamma<1,\ \Delta_{\max}>0,\ 0<\eta_1<\eta_2<1,\ 0<\kappa_\vartheta<1,\ 0<\kappa_\varphi$$

2: **Model and constraint update**: Choose a model, $m_k(\boldsymbol{\mu})$, constraint, $\vartheta_k(\boldsymbol{\mu})$, and gradient error bound, $\varphi_k(\boldsymbol{\mu})$, such that

$$\begin{aligned}\|F(\boldsymbol{\mu}_k)-F(\boldsymbol{\mu})+m_k(\boldsymbol{\mu})-m_k(\boldsymbol{\mu}_k)\| &\le \zeta\vartheta_k(\boldsymbol{\mu}) \qquad \boldsymbol{\mu}\in\mathcal{R}_k\\ \|\nabla F(\boldsymbol{\mu}_k)-\nabla m_k(\boldsymbol{\mu}_k)\| &\le \xi\varphi_k(\boldsymbol{\mu}_k)\\ \vartheta_k(\boldsymbol{\mu}_k) &\le \kappa_\vartheta\Delta_k\\ \varphi_k(\boldsymbol{\mu}_k) &\le \kappa_\varphi\min\{\|\nabla m_k(\boldsymbol{\mu}_k)\|,\ \Delta_k\}\end{aligned}$$

where $\zeta,\ \xi>0$ are arbitrary constants and $\mathcal{R}_k=\{\boldsymbol{\mu}\in\mathbb{R}^{N_\mu}\mid \vartheta_k(\boldsymbol{\mu})\le\Delta_k\}$

3: **Step computation**: Approximately solve the trust region subproblem

$$\min_{\boldsymbol{\mu}\in\mathbb{R}^{N_\mu}} m_k(\boldsymbol{\mu}) \qquad \text{subject to} \qquad \vartheta_k(\boldsymbol{\mu})\le\Delta_k$$

for a candidate, $\hat{\boldsymbol{\mu}}_k$, that satisfies $\vartheta_k(\hat{\boldsymbol{\mu}}_k)\le\Delta_k$ and

$$m_k(\boldsymbol{\mu}_k)-m_k(\hat{\boldsymbol{\mu}}_k)\ge\kappa_s\|\nabla m_k(\boldsymbol{\mu}_k)\|\min\left\{(1-\kappa_\vartheta)\kappa_{\nabla\vartheta}^{-1}\Delta_k,\frac{\|\nabla m_k(\boldsymbol{\mu}_k)\|}{\beta_k}\right\}$$

where $\kappa_s\in(0,\,1)$, $\|\nabla\vartheta_k(\boldsymbol{\mu})\|\le\kappa_{\nabla\vartheta}$ for all $\boldsymbol{\mu}\in\mathcal{R}_k$, and $\beta_k\coloneqq 1+\sup_{\boldsymbol{\mu}\in\mathcal{R}_k}\left\|\nabla^2 m_k(\boldsymbol{\mu})\right\|$.

4: **Actual-to-predicted reduction**: Compute actual-to-predicted reduction ratio approximation according to

$$\rho_k=\frac{F(\boldsymbol{\mu}_k)-F(\hat{\boldsymbol{\mu}}_k)}{m_k(\boldsymbol{\mu}_k)-m_k(\hat{\boldsymbol{\mu}}_k)}$$

5: **Step acceptance**:

$$\textbf{if}\quad \rho_k\ge\eta_1 \quad \textbf{then}\quad \boldsymbol{\mu}_{k+1}=\hat{\boldsymbol{\mu}}_k \quad \textbf{else}\quad \boldsymbol{\mu}_{k+1}=\boldsymbol{\mu}_k \quad \textbf{end if}$$

6: **Trust region update**:

$$\begin{array}{llll}\textbf{if}\quad \rho_k\le\eta_1 & \textbf{then}\quad \Delta_{k+1}\in(0,\gamma\vartheta_k(\hat{\boldsymbol{\mu}}_k)] & \textbf{end if}\\ \textbf{if}\quad \rho_k\in(\eta_1,\eta_2) & \textbf{then}\quad \Delta_{k+1}\in[\gamma\vartheta_k(\hat{\boldsymbol{\mu}}_k),\Delta_k] & \textbf{end if}\\ \textbf{if}\quad \rho_k\ge\eta_2 & \textbf{then}\quad \Delta_{k+1}\in[\Delta_k,\Delta_{\max}] & \textbf{end if}\end{array}$$

where $\kappa_\varphi > 0$ is an algorithmic constant. The main purpose of the gradient condition is to ensure sufficient accuracy in the model gradient is obtained near convergence ($\|\nabla m_k(\boldsymbol{\mu}_k)\|$ small) or after failed steps (Δ_k small). When combined with the error bound in (3.7), it also guarantees a local minima of F is approached as $\|\nabla m_k(\boldsymbol{\mu}_k)\| \to 0$. The error bounds and requirements on the error indicators are summarized in (3.12)-(3.15) below

$$|F(\boldsymbol{\mu}_k) - F(\boldsymbol{\mu}) + m_k(\boldsymbol{\mu}) - m_k(\boldsymbol{\mu}_k)| \leq \zeta \vartheta_k(\boldsymbol{\mu}) \qquad \boldsymbol{\mu} \in \mathcal{R}_k \tag{3.12}$$

$$\|\nabla F(\boldsymbol{\mu}_k) - \nabla m_k(\boldsymbol{\mu}_k)\| \leq \xi \varphi_k(\boldsymbol{\mu}_k) \tag{3.13}$$

$$\vartheta_k(\boldsymbol{\mu}_k) \leq \kappa_\vartheta \Delta_k \tag{3.14}$$

$$\varphi_k(\boldsymbol{\mu}_k) \leq \kappa_\varphi \min\{\|\nabla m_k(\boldsymbol{\mu}_k)\|, \Delta_k\}. \tag{3.15}$$

where $\xi, \zeta > 0$ are arbitrary constants.

Traditional trust region methods ($\vartheta_k(\boldsymbol{\mu}) = \|\boldsymbol{\mu} - \boldsymbol{\mu}_k\|$) that allow for inexact objective and gradient evaluations [93, 108, 109] naturally fit requirements (3.12)-(3.15) as follows. Consider an arbitrary model, $m_k(\boldsymbol{\mu})$, and gradient error bound, $\varphi_k(\boldsymbol{\mu}_k)$, such that (3.13) and (3.15) are satisfied. Then, $\vartheta_k(\boldsymbol{\mu}) = \|\boldsymbol{\mu} - \boldsymbol{\mu}_k\|$ automatically satisfies (3.12) and (3.14). Condition (3.14) is trivial to verify since $\vartheta_k(\boldsymbol{\mu}_k) = 0$. Condition (3.12) is verified, following [108], by considering the Taylor expansion of F and m_k about $\boldsymbol{\mu}_k$

$$F(\boldsymbol{\mu}) = F(\boldsymbol{\mu}_k) + \nabla F(\boldsymbol{\mu}_k)(\boldsymbol{\mu} - \boldsymbol{\mu}_k) + \frac{1}{2}(\boldsymbol{\mu} - \boldsymbol{\mu}_k)^T \nabla^2 F(\boldsymbol{y})(\boldsymbol{\mu} - \boldsymbol{\mu}_k)$$

$$m_k(\boldsymbol{\mu}) = m_k(\boldsymbol{\mu}_k) + \nabla m_k(\boldsymbol{\mu}_k)(\boldsymbol{\mu} - \boldsymbol{\mu}_k) + \frac{1}{2}(\boldsymbol{\mu} - \boldsymbol{\mu}_k)^T \nabla^2 m_k(\boldsymbol{z})(\boldsymbol{\mu} - \boldsymbol{\mu}_k)$$

where $\boldsymbol{y}, \boldsymbol{z} \in \mathbb{R}^{N_\mu}$ are arbitrary points that lie on the line between $\boldsymbol{\mu}$ and $\boldsymbol{\mu}_k$. Subtracting these equations, subsequent rearrangement, and application of the triangle inequality leads to

$$|F(\boldsymbol{\mu}_k) - F(\boldsymbol{\mu}) + m_k(\boldsymbol{\mu}) - m_k(\boldsymbol{\mu}_k)| \leq \|\nabla F(\boldsymbol{\mu}_k) - \nabla m_k(\boldsymbol{\mu}_k)\| \|\boldsymbol{\mu} - \boldsymbol{\mu}_k\| + \frac{1}{2} \left\|\nabla^2 F(\boldsymbol{y}) - \nabla m_k(\boldsymbol{z})\right\| \|\boldsymbol{\mu} - \boldsymbol{\mu}_k\|^2.$$

The gradient condition (3.15) and the fact that $\boldsymbol{\mu} \in \mathcal{R}_k = \{\boldsymbol{y} \in \mathbb{R}^N \mid \|\boldsymbol{y} - \boldsymbol{\mu}_k\| \leq \Delta_k\}$ are used to reduce the above inequality to

$$|F(\boldsymbol{\mu}_k) - F(\boldsymbol{\mu}) + m_k(\boldsymbol{\mu}) - m_k(\boldsymbol{\mu}_k)| \leq (\xi \kappa_\varphi + \frac{1}{2}\alpha_k)\Delta_k \|\boldsymbol{\mu} - \boldsymbol{\mu}_k\|$$

where $\alpha_k = \sup_{\boldsymbol{\mu} \in \mathcal{R}_k} \left(\left\|\nabla^2 F(\boldsymbol{\mu})\right\| + \left\|\nabla^2 m_k(\boldsymbol{\mu})\right\|\right)$ is well-defined if the objective and model Hessians are uniformly bounded on $\mathcal{R}_k$. Furthermore, assume these Hessians are uniformly bounded on all of $\mathbb{R}^{N_\mu}$, i.e., there exists $\alpha > 0$ such that $\alpha_k \leq \alpha$. This assumption and the introduction of an algorithmic parameter $\Delta_{\max}$ such that $\Delta_k \leq \Delta_{\max}$ (see discussion of step 4) leads to the desired result

$$|F(\boldsymbol{\mu}_k) - F(\boldsymbol{\mu}) + m_k(\boldsymbol{\mu}) - m_k(\boldsymbol{\mu}_k)| \leq \zeta \vartheta_k(\boldsymbol{\mu})$$

where $\zeta = (\xi \kappa_\varphi + \alpha/2)\Delta_{\max}$ is a constant.

Remark. *A similar generalization of trust region methods was introduced in [208] that used reduced-order models for the approximation model of the* linear *PDE. However, the method in that work requires the pointwise error bound (3.8) on the objective accuracy. The bound on the objective* variation *in (3.7) was shown to be a weaker condition than the pointwise bound since (3.8) is a special case of (3.7). Additionally, the objective variation bound provides considerable flexibility compared to the pointwise bound. For example, the bound in (3.7) encompasses the traditional trust region constraint $\vartheta_k(\boldsymbol{\mu}) = \|\boldsymbol{\mu} - \boldsymbol{\mu}_k\|$ without requiring zeroth-order consistency $m_k(\boldsymbol{\mu}_k) = F(\boldsymbol{\mu}_k)$, whereas this would would be required by the pointwise bound since $\vartheta_k(\boldsymbol{\mu}_k) = 0$. Therefore the bound in (3.7) enables the generalized trust region method to reduce to a traditional trust region method, even when the model objective is* inexact *at trust region centers. This will be exploited in Chapters 5–6.*

Step 2: Step candidate as solution of trust region subproblem

The model and trust region constraint functions are used to form the trust region subproblem in (3.4) whose minimizer is used as the candidate for $\boldsymbol{\mu}_{k+1}$. In traditional trust region methods, it is well-known that the trust region subproblem does not need to be solved exactly. In fact, it may be as difficult to solve the trust region subproblem as the original unconstrained optimization problem (3.1). Define the Cauchy point (Definition 3.3) as the minimizer of the trust region subproblem restricted to the steepest decent direction. It turns out that an essential component in the global convergence theory of trust region methods is the decrease in the model realized by the Cauchy point (Theorem 3.2) [133].

Definition 3.3 (Cauchy point)**.** *The Cauchy point of the trust region subproblem*

$$\begin{aligned} &\underset{\boldsymbol{\mu}\in\mathbb{R}^{N_{\boldsymbol{\mu}}}}{\text{minimize}} && m_k(\boldsymbol{\mu}) \\ &\text{subject to} && \|\boldsymbol{\mu} - \boldsymbol{\mu}_k\| \leq \Delta_k \end{aligned} \tag{3.16}$$

is $\boldsymbol{\mu}_k^C = \boldsymbol{\mu}_k - s^*\nabla m_k(\boldsymbol{\mu}_k)$, where s^* is the solution of the univariate optimization problem

$$\begin{aligned} &\underset{s\geq 0}{\text{minimize}} && m_k(\boldsymbol{\mu}_k - s\nabla m_k(\boldsymbol{\mu}_k)) \\ &\text{subject to} && s\,\|\nabla m_k(\boldsymbol{\mu}_k)\| \leq \Delta_k. \end{aligned} \tag{3.17}$$

Theorem 3.2 (Cauchy decrease)**.** *The decrease in the model from $\boldsymbol{\mu}_k$ to the Cauchy point $\boldsymbol{\mu}_k^C$ is at least*

$$m_k(\boldsymbol{\mu}_k) - m_k(\boldsymbol{\mu}_k^C) \geq \frac{1}{2}\,\|\nabla m_k(\boldsymbol{\mu}_k)\| \min\left\{\frac{\|m_k(\boldsymbol{\mu}_k)\|}{\beta_k}, \Delta_k\right\}, \tag{3.18}$$

where $\beta_k \coloneqq 1 + \sup_{\boldsymbol{\mu}\in\mathcal{R}_k} \left\|\nabla^2 m_k(\boldsymbol{\mu})\right\|$.

Proof. See Theorem 6.3.1 of [48]. □

Therefore, instead of requiring the candidate for $\boldsymbol{\mu}_{k+1}$ be the exact minimizer of (3.16), it suffices to use any point that achieves a *fraction* of the Cauchy decrease [133, 48]. This not only provides

an opportunity for efficiency, but also a convenient framework for analyzing global convergence properties.

Since these results pertaining to the Cauchy point and its connection to global convergence theory are specific to the quadratic constraint constraint in (3.16), this section aims to generalize these concepts for trust region subproblems of the form (3.4). This can easily be done if the gradient of the constraint is bounded within the trust region, i.e., $\|\nabla\vartheta_k(\boldsymbol{\mu})\| \leq \kappa_{\nabla\vartheta}$ for all $\boldsymbol{\mu} \in \mathcal{R}_k$. In this case, Lemma A.1 guarantees $\{\boldsymbol{\mu} \in \mathbb{R}^{N_\mu} \mid \|\boldsymbol{\mu} - \boldsymbol{\mu}_k\| \leq (1-\kappa_\varphi)\kappa_{\nabla\vartheta}^{-1}\Delta_k\} \subset \{\boldsymbol{\mu} \in \mathbb{R}^{N_\mu} \mid \vartheta_k(\boldsymbol{\mu}) \leq \Delta_k\}$. Thus, from Theorem 3.2, there exists a point in the trust region $\boldsymbol{\mu} \in \mathcal{R}_k$ such that

$$m_k(\boldsymbol{\mu}_k) - m_k(\boldsymbol{\mu}) \geq \kappa_s \|\nabla m_k(\boldsymbol{\mu}_k)\| \min\left\{\frac{\|m_k(\boldsymbol{\mu}_k)\|}{\beta_k}, (1-\kappa_\varphi)\kappa_{\nabla\vartheta}^{-1}\Delta_k\right\} \tag{3.19}$$

where $\kappa_s \in (0, 1)$. Appendix A will show that this condition that resembles the fraction of Cauchy decrease leads to global convergence of the proposed trust region method.

In this work, a substantial cost difference is assumed to separate evaluations of $F(\boldsymbol{\mu})$ and $\nabla F(\boldsymbol{\mu})$ from evaluations $m_k(\boldsymbol{\mu})$ and $\nabla m_k(\boldsymbol{\mu})$ so the trust region subproblem is solved exactly with little penalty. Section 3.1.2 details an interior-point method to solve the trust region subproblem (3.4).

Step 3: Actual-to-predicted decrease ratio

After the candidate $\hat{\boldsymbol{\mu}}_k$ has been computed, the ratio between the reduction in F and m_k that would be realized by taking this step is computed according (3.9). This will be used in the next section to determine if the step should be accepted and to modify the trust region radius. The computation of ρ_k according to (3.9) requires queries to the expensive function $F(\boldsymbol{\mu})$ and therefore constitutes a major bottleneck in the trust region algorithm. Following the work in [109], this bottleneck is mitigated through the introduction of another approximation, $\psi_k(\boldsymbol{\mu})$ that will be used solely in the computation of ρ_k, i.e.,

$$\rho_k = \frac{\psi_k(\boldsymbol{\mu}_k) - \psi_k(\hat{\boldsymbol{\mu}}_k)}{m_k(\boldsymbol{\mu}_k) - m_k(\hat{\boldsymbol{\mu}}_k)}. \tag{3.20}$$

Define $\psi_k : \mathbb{R}^{N_\mu} \to \mathbb{R}$ as an approximation of the $F(\boldsymbol{\mu})$, equipped with a familiar asymptotic error bound: there exists a constant $\sigma > 0$ such that

$$|F(\boldsymbol{\mu}_k) - F(\boldsymbol{\mu}) + \psi_k(\boldsymbol{\mu}) - \psi_k(\boldsymbol{\mu}_k)| \leq \sigma\theta_k(\boldsymbol{\mu}), \tag{3.21}$$

where $\theta_k : \mathbb{R}^{N_\mu} \to \mathbb{R}_+$ is an error indicator. To ensure global convergence (see Appendix A for proof) in the presence of this additional approximation, $\theta_k(\hat{\boldsymbol{\mu}}_k)$ must satisfy

$$\theta_k^\omega(\hat{\boldsymbol{\mu}}_k) \leq \eta \min\{m_k(\boldsymbol{\mu}_k) - m_k(\hat{\boldsymbol{\mu}}_k), r_k\}, \tag{3.22}$$

where $\omega \in (0, 1)$, $\eta < \min\{\eta_1, 1-\eta_2\}$, and $\{r_k\}_{k=1}^\infty$ is a sequence such that $r_k \to 0$. The algorithmic parameters η_1 and η_2 are related to the specifics of the step assessment and radius modification detailed in the next section. The forcing sequence r_k is required to ensure ρ_k in (3.20) approaches

the true ratio between the actual and predicted reduction and is taken in this work as $r_k = 1/(k+1)$.

The flexibility afforded by the use of an approximation model reveals an immediate and obvious improvement to the generalized trust region algorithm. The error bound required between $\psi_k(\boldsymbol{\mu})$ and $\theta_k(\boldsymbol{\mu})$ in (3.21) is identical to the relationship between $m_k(\boldsymbol{\mu})$ and $\vartheta_k(\boldsymbol{\mu})$ in (3.7), which immediately suggests the choice $\psi_k(\boldsymbol{\mu}) = m_k(\boldsymbol{\mu})$ and $\theta_k(\boldsymbol{\mu}) = \vartheta_k(\boldsymbol{\mu})$. From the discussion above, this choice will lead to a globally convergent algorithm provided

$$\vartheta_k(\hat{\boldsymbol{\mu}}_k)^\omega \leq \eta \min\{m_k(\boldsymbol{\mu}_k) - m_k(\hat{\boldsymbol{\mu}}_k),\, r_k\}. \tag{3.23}$$

This condition is inexpensive to check since it only involves queries to the approximation model and error indicator and does not require evaluations of the expensive objective function $F(\boldsymbol{\mu})$. Additionally, the choice $\psi_k(\boldsymbol{\mu}) = m_k(\boldsymbol{\mu})$ guarantees the approximation of the actual-to-predicted reduction ratio is always unity, i.e.,

$$\rho_k = \frac{\psi_k(\boldsymbol{\mu}_k) - \psi_k(\hat{\boldsymbol{\mu}}_k)}{m_k(\boldsymbol{\mu}_k) - m_k(\hat{\boldsymbol{\mu}}_k)} = \frac{m_k(\boldsymbol{\mu}_k) - m_k(\hat{\boldsymbol{\mu}}_k)}{m_k(\boldsymbol{\mu}_k) - m_k(\hat{\boldsymbol{\mu}}_k)} = 1. \tag{3.24}$$

Thus, for a given iteration k, if the approximation model $m_k(\boldsymbol{\mu})$ and error indicator $\vartheta_k(\boldsymbol{\mu})$ chosen in the first step of the generalized trust region algorithm satisfy (3.23), ρ_k can be taken as unity without any additional work. The next section will classify such a step as *very successful* and the step will be accepted $\boldsymbol{\mu}_{k+1} = \hat{\boldsymbol{\mu}}_k$ and the trust region radius increased. In the event that (3.23) is not satisfied, the choice $\psi_k(\boldsymbol{\mu}) = m_k(\boldsymbol{\mu})$ and $\theta_k(\boldsymbol{\mu}) = \vartheta_k(\boldsymbol{\mu})$ is not sufficient to guarantee global convergence and $\psi_k(\boldsymbol{\mu})$ and $\theta_k(\boldsymbol{\mu})$ must be constructed to satisfy (3.21)-(3.22).

Algorithm 2 states the optimized trust region method that incorporates this additional level of approximation. Appendix A details the global convergence proof for this algorithm. Global convergence of Algorithm 1, i.e., without the ψ_k approximation model, follows trivially by taking $\psi_k(\boldsymbol{\mu}) = F(\boldsymbol{\mu})$ and $\theta_k(\boldsymbol{\mu}) = 0$.

Step 4: Step assessment and radius update

Once ρ_k is computed according to either (3.9) or (3.20), the quality of the step is assessed by comparing ρ_k to unity, i.e., the actual-to-predicted ratio if the model was perfect, $m_k(\boldsymbol{\mu}) = F(\boldsymbol{\mu})$. If the ρ_k is close to unity, the step is accepted by setting $\boldsymbol{\mu}_{k+1} = \hat{\boldsymbol{\mu}}_k$ and the trust region radius is increased. In the case where $\rho_k \ll 1$, especially if it is negative (the true objective fails to decrease: $F(\hat{\boldsymbol{\mu}}_k) > F(\boldsymbol{\mu}_k)$), the step is rejected, $\boldsymbol{\mu}_{k+1} = \boldsymbol{\mu}_k$, and trust region radius is decreased.

For a practical algorithm, define algorithmic constants $0 < \eta_1 < \eta_2 < 1$ that will indicate values of ρ_k that govern step acceptance and the radius update. If $\rho_k \leq \eta_1$, the model did not substantially reduce the true objective so the step is rejected and the trust region radius decreased such that $\Delta_{k+1} \leq \gamma \vartheta_k(\hat{\boldsymbol{\mu}}_k)$ where $0 < \gamma < 1$ is a constant. This will be called an *unsuccessful* step. Modification of the radius in this manner ensures that if the optimization problem in (3.4) terminates at a point $\hat{\boldsymbol{\mu}}_k$ strictly *interior* to the feasible set $\mathcal{R}_k = \{\boldsymbol{\mu} \in \mathbb{R}^{N_\mu} \mid \vartheta_k(\boldsymbol{\mu}) \leq \Delta_k\}$, the

Algorithm 2 Error-aware multifidelity trust region method with inexact objective evaluations

1: **Initialization**: Given

$$\boldsymbol{\mu}_0,\ \Delta_0,\ 0<\gamma<1,\ \Delta_{\max}>0,\ 0<\eta_1<\eta_2<1,\ 0<\kappa_\vartheta<1,\ 0<\kappa_\varphi,$$
$$\omega\in(0,\,1),\ \{r_k\}_{k=1}^{\infty}\subset[0,\infty)\text{ such that }r_k\to 0$$

2: **Model and constraint update**: Choose a model, $m_k(\boldsymbol{\mu})$, constraint, $\vartheta_k(\boldsymbol{\mu})$, and gradient error bound, $\varphi_k(\boldsymbol{\mu})$, such that

$$\begin{aligned}|F(\boldsymbol{\mu}_k)-F(\boldsymbol{\mu})+m_k(\boldsymbol{\mu})-m_k(\boldsymbol{\mu}_k)|&\le\zeta\vartheta_k(\boldsymbol{\mu})\qquad\boldsymbol{\mu}\in\mathcal{R}_k\\ \|\nabla F(\boldsymbol{\mu}_k)-\nabla m_k(\boldsymbol{\mu}_k)\|&\le\xi\varphi_k(\boldsymbol{\mu}_k)\\ \vartheta_k(\boldsymbol{\mu}_k)&\le\kappa_\vartheta\Delta_k\\ \varphi_k(\boldsymbol{\mu}_k)&\le\kappa_\varphi\min\{\|\nabla m_k(\boldsymbol{\mu}_k)\|,\,\Delta_k\}\end{aligned}$$

where $\zeta,\,\xi>0$ are arbitrary constants and $\mathcal{R}_k=\{\boldsymbol{\mu}\in\mathbb{R}^{N_{\boldsymbol{\mu}}}\mid\vartheta_k(\boldsymbol{\mu})\le\Delta_k\}$

3: **Step computation**: Approximately solve the trust region subproblem

$$\min_{\boldsymbol{\mu}\in\mathbb{R}^{N_{\boldsymbol{\mu}}}}\ m_k(\boldsymbol{\mu})\qquad\text{subject to}\qquad\vartheta_k(\boldsymbol{\mu})\le\Delta_k$$

for a candidate, $\hat{\boldsymbol{\mu}}_k$, that satisfies $\vartheta_k(\hat{\boldsymbol{\mu}}_k)\le\Delta_k$ and

$$m_k(\boldsymbol{\mu}_k)-m_k(\hat{\boldsymbol{\mu}}_k)\ge\kappa_s\,\|\nabla m_k(\boldsymbol{\mu}_k)\|\min\left\{(1-\kappa_\vartheta)\kappa_{\nabla\vartheta}^{-1}\Delta_k,\,\frac{\|\nabla m_k(\boldsymbol{\mu}_k)\|}{\beta_k}\right\}\tag{3.25}$$

where $\kappa_s\in(0,\,1)$, $\|\nabla\vartheta_k(\boldsymbol{\mu})\|\le\kappa_{\nabla\vartheta}$ for all $\boldsymbol{\mu}\in\mathcal{R}_k$, and $\beta_k\coloneqq 1+\sup_{\boldsymbol{\mu}\in\mathcal{R}_k}\left\|\nabla^2 m_k(\boldsymbol{\mu})\right\|$

4: **Actual-to-predicted reduction**: Compute actual-to-predicted reduction ratio approximation according to

$$\rho_k=\begin{cases}1 & \text{if}\quad\vartheta_k(\hat{\boldsymbol{\mu}}_k)^\omega\le\eta\min\{m_k(\boldsymbol{\mu}_k)-m_k(\hat{\boldsymbol{\mu}}_k),\,r_k\}\\ \dfrac{\psi_k(\boldsymbol{\mu}_k)-\psi_k(\hat{\boldsymbol{\mu}}_k)}{m_k(\boldsymbol{\mu}_k)-m_k(\hat{\boldsymbol{\mu}}_k)} & \text{otherwise}\end{cases}$$

where $\psi_k(\boldsymbol{\mu})$ and $\theta_k(\boldsymbol{\mu})$ satisfy

$$\begin{aligned}\|\psi_k(\boldsymbol{\mu}_k)-\psi_k(\boldsymbol{\mu})+m_k(\boldsymbol{\mu})-m_k(\boldsymbol{\mu}_k)\|&\le\sigma\theta_k(\boldsymbol{\mu})\qquad\boldsymbol{\mu}\in\mathcal{R}_k\\ \theta_k^\omega(\hat{\boldsymbol{\mu}}_k)&\le\eta\min\{m_k(\boldsymbol{\mu}_k)-m_k(\hat{\boldsymbol{\mu}}_k),\ r_k\}\end{aligned}\tag{3.26}$$

where $\eta<\min\{\eta_1,\,1-\eta_2\}$ and $\sigma>0$ is an arbitrary constant

5: **Step acceptance**:

$$\textbf{if}\quad\rho_k\ge\eta_1\quad\textbf{then}\quad\boldsymbol{\mu}_{k+1}=\hat{\boldsymbol{\mu}}_k\quad\textbf{else}\quad\boldsymbol{\mu}_{k+1}=\boldsymbol{\mu}_k\quad\textbf{end if}$$

6: **Trust region update**:

$$\begin{array}{llll}\textbf{if}\quad\rho_k\le\eta_1 & \textbf{then}\quad\Delta_{k+1}\in(0,\gamma\vartheta_k(\hat{\boldsymbol{\mu}}_k)] & \textbf{end if}\\ \textbf{if}\quad\rho_k\in(\eta_1,\eta_2) & \textbf{then}\quad\Delta_{k+1}\in[\gamma\vartheta_k(\hat{\boldsymbol{\mu}}_k),\Delta_k] & \textbf{end if}\\ \textbf{if}\quad\rho_k\ge\eta_2 & \textbf{then}\quad\Delta_{k+1}\in[\Delta_k,\Delta_{\max}] & \textbf{end if}\end{array}$$

feasible set at the next iteration, $\mathcal{R}_{k+1}$, will not include $\hat{\boldsymbol{\mu}}_k$. If $\rho_k \in (\eta_1, \eta_2)$, the step is accepted and the trust region radius is not modified, $\Delta_{k+1} = \Delta_k$. This is called a *successful* step. Finally, if $\rho_k \geq \eta_2$, the step is accepted since the model predicted the decrease in the objective to high accuracy. In this type of *very successful* step, the trust region constraint may be too restrictive so the radius is increased, usually according to $\Delta_{k+1} = \min\{(1/\gamma)\Delta_k, \Delta_{\max}\}$, where $\Delta_{\max}$ is an algorithmic parameter that specifies the maximum trust region radius.

Summary

Two variants of a generalized, multifidelity trust region method were introduced in this section and global convergence was established for both methods in Appendix A. The first version, presented in Algorithm 1, requires the computation of the exact ratio of actual-to-predicted reduction. This method is completely prescribed once the approximation function $m_k(\boldsymbol{\mu})$ and error indicators $\vartheta_k(\boldsymbol{\mu})$ and $\varphi_k(\boldsymbol{\mu})$ that satisfy (3.12)-(3.15) have been defined. Unlike traditional trust region methods, the trust region subproblem of the proposed method is a difficult optimization problem that cannot leverage the many highly efficient trust region solvers—see [48] for a review—and calls for an exact nonlinear optimization solver. Section 3.1.2 presents a simple primal interior point method based on quasi-Newton search directions and a backtracking linesearch to solve the optimization problem in (3.4) exactly. This guarantees the candidate point $\hat{\boldsymbol{\mu}}_k$ will satisfying the fraction of Cauchy decrease condition (Theorem 3.2), an important component of the global convergence theory. The variant of the error-aware multifidelity trust region method presented in Algorithm 2 leverages an approximation of the actual-to-predicted reduction ratio. In additional to $m_k(\boldsymbol{\mu})$ and the error indicators $\vartheta_k(\boldsymbol{\mu})$, $\varphi_k(\boldsymbol{\mu})$, this method requires the construction of an additional approximation model $\psi_k(\boldsymbol{\mu})$ and error indicator $\theta_k(\boldsymbol{\mu})$ that satisfy (3.21)-(3.22). This flexibility was leveraged to define condition (3.23) that ensures the choice $\psi_k(\boldsymbol{\mu}) = m_k(\boldsymbol{\mu})$ and $\theta_k(\boldsymbol{\mu}) = \vartheta_k(\boldsymbol{\mu})$ will preserve global convergence and guarantees step k is *very successful* without requiring queries to the expensive objective $F(\boldsymbol{\mu})$ or construction of a new approximation model.

As written, Algorithms 1 and 2 are skeletons since details pertaining to the construction of the approximation models $m_k(\boldsymbol{\mu})$, $\psi_k(\boldsymbol{\mu})$ and error indicators $\vartheta_k(\boldsymbol{\mu})$, $\varphi_k(\boldsymbol{\mu})$, $\theta_k(\boldsymbol{\mu})$ have been abstracted away. This will serve as the point of departure for Chapters 5–6, which will construct these approximation models and error indicators for the specific class of problems under consideration. In particular, Chapter 5 will use projection-based reduced-order/hyperreduced models and residual-based error bounds to define these trust region functions in the context of deterministic PDE-constrained optimization. Chapter 6 will combine projection-based reduced-order models and sparse grids to define the approximation model to efficiently solve stochastic PDE-constrained optimization problems. The error indicators will use residual-based error bounds to account for pointwise error and dimension-adaptive sparse grids [67] to account for truncation error. The proposed methods will implicitly assume there are relatively few parameters compared to the size of the state vector $N_{\boldsymbol{\mu}} \ll N_{\boldsymbol{u}}$ (since reduction will solely be applied to the state space). Appendix C will discuss the use of linesearch and subspace methods to extend the proposed methods to efficiently handle many

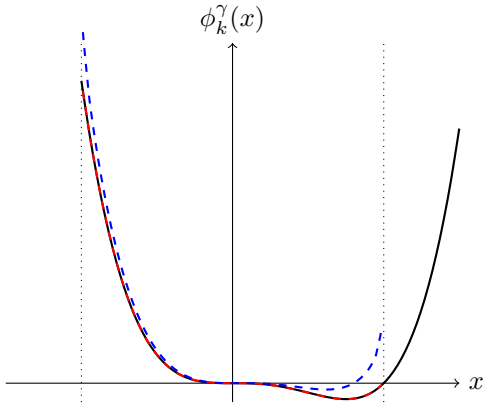


Figure 3.2: Logarithmic barrier function (3.27) corresponding to $m_k(x) = x^4 - x^3$ (——), $\vartheta_k(x) = x^2$, $\Delta_k = 1$ with $\gamma = 0.1$ (- - -) and $\gamma = 0.0001$ (- - -).

optimization variables, i.e., $N_{\boldsymbol{\mu}} = \mathcal{O}(N_{\boldsymbol{u}})$.

3.1.2 Interior-Point Method for Trust Region Subproblem

The trust region subproblem employed in the proposed generalized trust region method is a general nonlinear program and cannot be (approximately) solved with the plethora of highly efficient and specialized trust region subproblem solvers that have been developed [133, 48]. In this work, the trust region subproblem is solved exactly (up to a tolerance on the first-order optimality conditions), which is in opposition to most trust region methods that only seek a point that achieves a fraction of the Cauchy decrease. Due to the assumed substantial cost separation between the evaluation of F and m_k, an exact trust region solver comes at a relatively small penalty in cost. In fact, it may even be substantially more efficient than finding an approximate minimizer if it can result in even one fewer query to F. While any nonlinear optimization solver can be employed to solve the trust region subproblem (3.4), an interior point method [143] is used since the trust region center is strictly interior to the feasible set from condition (3.14).

Consider the logarithmic barrier function associated with the optimization problem (3.4)

$$\phi_k^\gamma(\boldsymbol{\mu}) = m_k(\boldsymbol{\mu}) - \gamma \log\left[\Delta_k - \vartheta_k(\boldsymbol{\mu})\right]. \tag{3.27}$$

This function, shown in Figure 3.2 for a specific choice of m_k and ϑ_k, tends to $+\infty$ as $\boldsymbol{\mu}$ approaches the boundary of the feasible set. This ensures that an *unconstrained* optimization problem of the form

$$\underset{\boldsymbol{\mu}\in\mathbb{R}^{N_{\boldsymbol{\mu}}}}{\text{minimize}} \quad \phi_k^\gamma(\boldsymbol{\mu}) \tag{3.28}$$

will remain interior to the feasible set (provided the initial guess is a feasible point). The unconstrained optimization problem in (3.28) approaches the constrained optimization problem in (3.41) as the barrier parameter goes to zero, i.e., $\gamma \to 0$. Thus, the constrained optimization problem in

(3.4) has been reduced to a sequence of unconstrained optimization problems (3.28) corresponding to a sequence $\gamma_p \to 0$. A more robust variant of the primal interior-point method discussed is a primal-dual approach that avoids the difficulty of solving (3.28) when γ approaches 0; however, only the primal approach will be considered for simplicity. Each unconstrained optimization problem is solved using a quasi-Newton method with Broyden-Fletcher-Goldfarb-Shanno (BFGS) Hessian updates and a backtracking linesearch to satisfy the Armijo sufficient decrease condition. Quasi-Newton methods look to improve an iterate $\boldsymbol{\mu}_k^j$ (the subscript k denotes the trust region, or major, iteration and the superscript j denotes the subproblem, or minor, iteration) by search along a direction, $\boldsymbol{p}_k^j$, defined as the solution of

$$\boldsymbol{B}_k^j \boldsymbol{p}_k^j = -\nabla \phi_k^\gamma(\boldsymbol{\mu}_k^j), \tag{3.29}$$

where $\boldsymbol{B}_k^j$ is a symmetric positive-definite approximation of the Hessian $\nabla^2 \phi_k^\gamma(\boldsymbol{\mu}_k^j)$. The BFGS Hessian update defines $\boldsymbol{B}_k^j$ from $\boldsymbol{B}_k^{j-1}$ according to

$$\boldsymbol{B}_k^{j+1} = \boldsymbol{B}_k^j + \frac{\boldsymbol{y}_k^j \boldsymbol{y}_k^{j\,T}}{\boldsymbol{y}_k^{j\,T} \boldsymbol{s}_k^j} - \frac{\boldsymbol{B}_k^j \boldsymbol{s}_k^j \boldsymbol{s}_k^{j\,T} \boldsymbol{B}_k^j}{\boldsymbol{s}_k^{j\,T} \boldsymbol{B}_k^j \boldsymbol{s}_k^j} \tag{3.30}$$

where

$$\boldsymbol{s}_k^j = \boldsymbol{\mu}_k^{j+1} - \boldsymbol{\mu}_k^j \qquad \boldsymbol{y}_k^j = \nabla \phi_k^\gamma(\boldsymbol{\mu}_k^{j+1}) - \nabla \phi_k^\gamma(\boldsymbol{\mu}_k^j)$$

and the Hessian approximation is initialized as the identity $\boldsymbol{B}_k^0 = \boldsymbol{I}$ (implying the first search direction $\boldsymbol{p}_k^0$ is the steepest descent direction). With the search direction computed according to (3.29), the new subproblem iterate is computed as

$$\boldsymbol{\mu}_k^{j+1} = \boldsymbol{\mu}_k^j + \alpha \boldsymbol{p}_k^j, \tag{3.31}$$

where $\alpha > 0$ is selected such that the Armijo condition

$$\phi_k^\gamma(\boldsymbol{\mu}_k^j + \alpha \boldsymbol{p}_k^j) \leq \phi_k^\gamma(\boldsymbol{\mu}_k^j) + \alpha c \boldsymbol{p}_k^{j\,T} \nabla \phi_k^\gamma(\boldsymbol{\mu}_k^j) \tag{3.32}$$

is satisfied, where $c > 0$ is a constant usually taken as $c = 10^{-4}$. The step length α is determined via a backtracking algorithm, i.e., $\alpha = \tau^n$, where $\tau \in (0, 1)$ is the backtrack factor and $n \geq 0$ is the smallest integer such that (3.32) is satisfied. The complete algorithm is summarized in Algorithm 3.

3.1.3 Numerical Experiment: Contrived

The generality and performance of the multifidelity trust region method proposed in this chapter is demonstrated on the canonical Rosenbrock problem

$$\underset{\boldsymbol{\mu} \in \mathbb{R}^2}{\text{minimize}} \quad F(\boldsymbol{\mu}) \coloneqq 100(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1^2)^2 + (1 - \boldsymbol{\mu}_1)^2. \tag{3.33}$$

Algorithm 3 Interior Point BFGS Method with Backtracking Linesearch

1: **Initialization**: Given

$$\boldsymbol{\mu}_k^0 = \boldsymbol{\mu}_k,\ \gamma,\ \boldsymbol{B}_0 = \boldsymbol{I},\ 0 < c < 1,\ 0 < \tau < 1$$

2: **Search direction computation**: Define step direction, $\boldsymbol{p}_k^j$, as the solution of

$$\boldsymbol{B}_k^j \boldsymbol{p}_k^j = -\nabla \phi_k^\gamma(\boldsymbol{\mu}_k^j)$$

3: **Linesearch**: Define the step length as $\alpha = \tau^n$ where n is the smallest integer such that

$$\phi_k^\gamma(\boldsymbol{\mu}_k^j + \tau^n \boldsymbol{p}_k^j) \leq \phi_k^\gamma(\boldsymbol{\mu}_k^j) + \tau^n c {\boldsymbol{p}_k^j}^T \nabla \phi_k^\gamma(\boldsymbol{\mu}_k^j)$$

4: **Update iterate**: Given search direction $\boldsymbol{p}_k^j$ and step length α, update current iterate

$$\boldsymbol{\mu}_k^{j+1} = \boldsymbol{\mu}_k^j + \alpha \boldsymbol{p}_k^j$$

5: **BFGS update**: Define $\boldsymbol{s}_k^j$ and $\boldsymbol{y}_k^j$ as

$$\boldsymbol{s}_k^j = \boldsymbol{\mu}_k^{j+1} - \boldsymbol{\mu}_k^j \qquad \boldsymbol{y}_k^j = \nabla \phi_k^\gamma(\boldsymbol{\mu}_k^{j+1}) - \nabla \phi_k^\gamma(\boldsymbol{\mu}_k^j)$$

and Hessian approximation update as

$$\boldsymbol{B}_k^{j+1} = \boldsymbol{B}_k^j + \frac{\boldsymbol{y}_k^j {\boldsymbol{y}_k^j}^T}{{\boldsymbol{y}_k^j}^T \boldsymbol{s}_k^j} - \frac{\boldsymbol{B}_k^j \boldsymbol{s}_k^j {\boldsymbol{s}_k^j}^T \boldsymbol{B}_k^j}{{\boldsymbol{s}_k^j}^T \boldsymbol{B}_k^j \boldsymbol{s}_k^j}$$

The approximation model and error indicators that will be used are not less expensive to evaluate than the objective function $F(\boldsymbol{\mu})$, which is an underlying assumption of the proposed methods. The purpose of this section is to study the behavior of the generalized trust region algorithm on a simple problem; Chapters 5 and 6 will consider more interesting applications where the approximation model and error indicators are substantially less expensive to evaluate than $F(\boldsymbol{\mu})$.

The model function will be taken as a quadratic approximation of $F(\boldsymbol{\mu})$ with controllable errors introduced into the value and gradient at the expansion point. For this purpose define

$$G(\boldsymbol{\mu};\, \bar{\boldsymbol{\mu}},\, \epsilon,\, \delta) \coloneqq F(\bar{\boldsymbol{\mu}}) + \epsilon + (\nabla F(\bar{\boldsymbol{\mu}}) + \delta \mathbf{1})^T(\boldsymbol{\mu} - \bar{\boldsymbol{\mu}}) + \frac{1}{2}(\boldsymbol{\mu} - \bar{\boldsymbol{\mu}})^T \nabla^2 F(\bar{\boldsymbol{\mu}})(\boldsymbol{\mu} - \bar{\boldsymbol{\mu}}), \tag{3.34}$$

where the gradient and Hessian of $F(\boldsymbol{\mu})$ are

$$\begin{aligned} \nabla F(\boldsymbol{\mu}) &= \begin{bmatrix} -400\boldsymbol{\mu}_1(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1^2) - 2(1 - \boldsymbol{\mu}_1) \\ 200(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1^2) \end{bmatrix} \\ \nabla^2 F(\boldsymbol{\mu}) &= \begin{bmatrix} 400(3\boldsymbol{\mu}_1^2 - \boldsymbol{\mu}_2) + 2 & -400\boldsymbol{\mu}_1 \\ -400\boldsymbol{\mu}_1 & 200 \end{bmatrix}. \end{aligned} \tag{3.35}$$

The gradient of $G(\boldsymbol{\mu};\, \bar{\boldsymbol{\mu}},\, \epsilon,\, \delta)$ is

$$\nabla G(\boldsymbol{\mu};\, \bar{\boldsymbol{\mu}},\, \epsilon,\, \delta) = \nabla F(\bar{\boldsymbol{\mu}}) + \delta \mathbf{1} + \nabla^2 F(\bar{\boldsymbol{\mu}})(\boldsymbol{\mu} - \bar{\boldsymbol{\mu}}). \tag{3.36}$$

From the definition of G and its gradient, it is clear that ϵ, δ are the errors in the value and gradient, respectively, at the expansion point $\bar{\boldsymbol{\mu}}$ since the evaluation of G and ∇G at $\bar{\boldsymbol{\mu}}$ gives

$$G(\bar{\boldsymbol{\mu}};\, \bar{\boldsymbol{\mu}},\, \epsilon,\, \delta) = F(\bar{\boldsymbol{\mu}}) + \epsilon \qquad \nabla G(\bar{\boldsymbol{\mu}};\, \bar{\boldsymbol{\mu}},\, \epsilon,\, \delta) = \nabla F(\bar{\boldsymbol{\mu}}) + \delta \mathbf{1}. \tag{3.37}$$

The approximation model $m_k(\boldsymbol{\mu})$ is taken as the (inexact) quadratic approximation of $F(\boldsymbol{\mu})$ at the trust region center, the trust region constraint is taken based on the *exact pointwise objective error*, and the gradient error indicator $\varphi_k(\boldsymbol{\mu})$ is taken to be the exact gradient error, i.e.,

$$\begin{aligned} m_k(\boldsymbol{\mu}) &\coloneqq G(\boldsymbol{\mu},\, \boldsymbol{\mu}_k,\, \epsilon_k,\, \delta_k) \\ \vartheta_k(\boldsymbol{\mu}) &\coloneqq |F(\boldsymbol{\mu}) - G(\boldsymbol{\mu};\, \boldsymbol{\mu}_k,\, \epsilon_k,\, \delta_k)| + |F(\boldsymbol{\mu}_k) - G(\boldsymbol{\mu}_k;\, \boldsymbol{\mu}_k,\, \epsilon_k,\, \delta_k)| \\ \varphi_k(\boldsymbol{\mu}) &\coloneqq \|\nabla F(\boldsymbol{\mu}) - \nabla G(\boldsymbol{\mu};\, \boldsymbol{\mu}_k,\, \epsilon_k,\, \delta_k)\| \end{aligned} \tag{3.38}$$

where the error terms ϵ_k, δ_k must be chosen based on the requirements of the global convergence theory. With these choices, the error bounds in (3.12)-(3.13) hold with $\zeta = \xi = 1$. The value of ϵ_k and δ_k will be chosen to ensure the error conditions (3.14) and (3.15) hold. At the trust region centers, the error indicators reduce to the following simple expressions

$$\begin{aligned} \vartheta_k(\boldsymbol{\mu}_k) &= 2\epsilon_k \\ \varphi_k(\boldsymbol{\mu}_k) &= \sqrt{2}\delta_k \end{aligned} \tag{3.39}$$

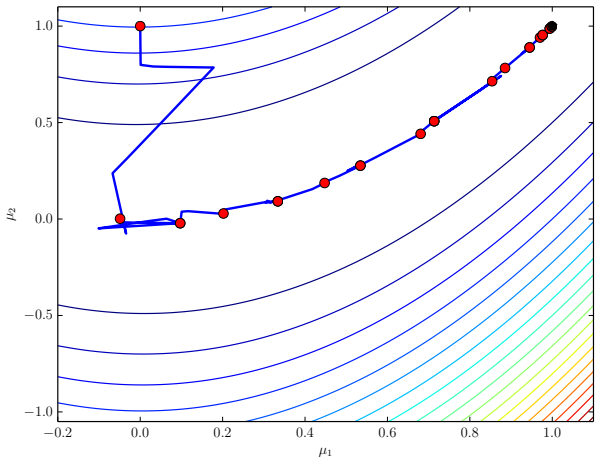


Figure 3.3: Trajectory of Algorithm 1 as applied to the Rosenbrock problem (3.33). The contours represent the true function $F(\boldsymbol{\mu})$, the red dots indicate trust region centers, and the blue line is the trajectory of the trust region subproblem.

and the error conditions in (3.14)-(3.15) become

$$\begin{aligned} \epsilon_k &\leq \frac{\kappa_\vartheta}{2}\Delta_k \\ \delta_k &\leq \frac{\kappa_\varphi}{\sqrt{2}}\min\{\|\nabla m_k(\boldsymbol{\mu}_k)\|\,,\,\Delta_k\} \end{aligned} \tag{3.40}$$

Since the right-hand side of the inequality for ϵ_k is independent of ϵ_k, admissible values are easily determined and $\epsilon_k = \kappa_\vartheta\Delta_k/2$ will be used throughout. The right-hand side of the inequality for δ_k depends on δ_k itself (through $m_k(\boldsymbol{\mu}_k)$) and, in general, an iterative algorithm must be used. A simple backtracking algorithm is used where any initial value of δ_k is chosen and reduced by a predefined factor until (3.40) is satisfied.

With the proposed definitions of $m_k(\boldsymbol{\mu})$, $\vartheta_k(\boldsymbol{\mu})$, and $\varphi_k(\boldsymbol{\mu})$ all ingredients necessary for the complete description of Algorithm 1 have been prescribed. The trust region subproblem (3.4) is solved using the BFGS interior-point method described in Section 3.1.2; the non-quadratic trust region constraint eliminates the possibility of using standard trust region solvers such as Steihaug-Toint CG. The trajectory of the optimization iterations—including the progress of the trust region centers and the trajectory of each trust region subproblem—is shown in Figure 3.3. Figure 3.4 provides additional insight to the Algorithm 1 by showing individual iterations, including the trust region center, candidate step, and feasible region for the trust region subproblem. Notice the substantial difference between the shape of the trust regions in Figure 3.4 and traditional trust regions that are spheres or ellipsoids. These error-aware trust region allows progress to be made toward the optimal solution by searching regions of the parameter space where the model is sufficiently accurate.

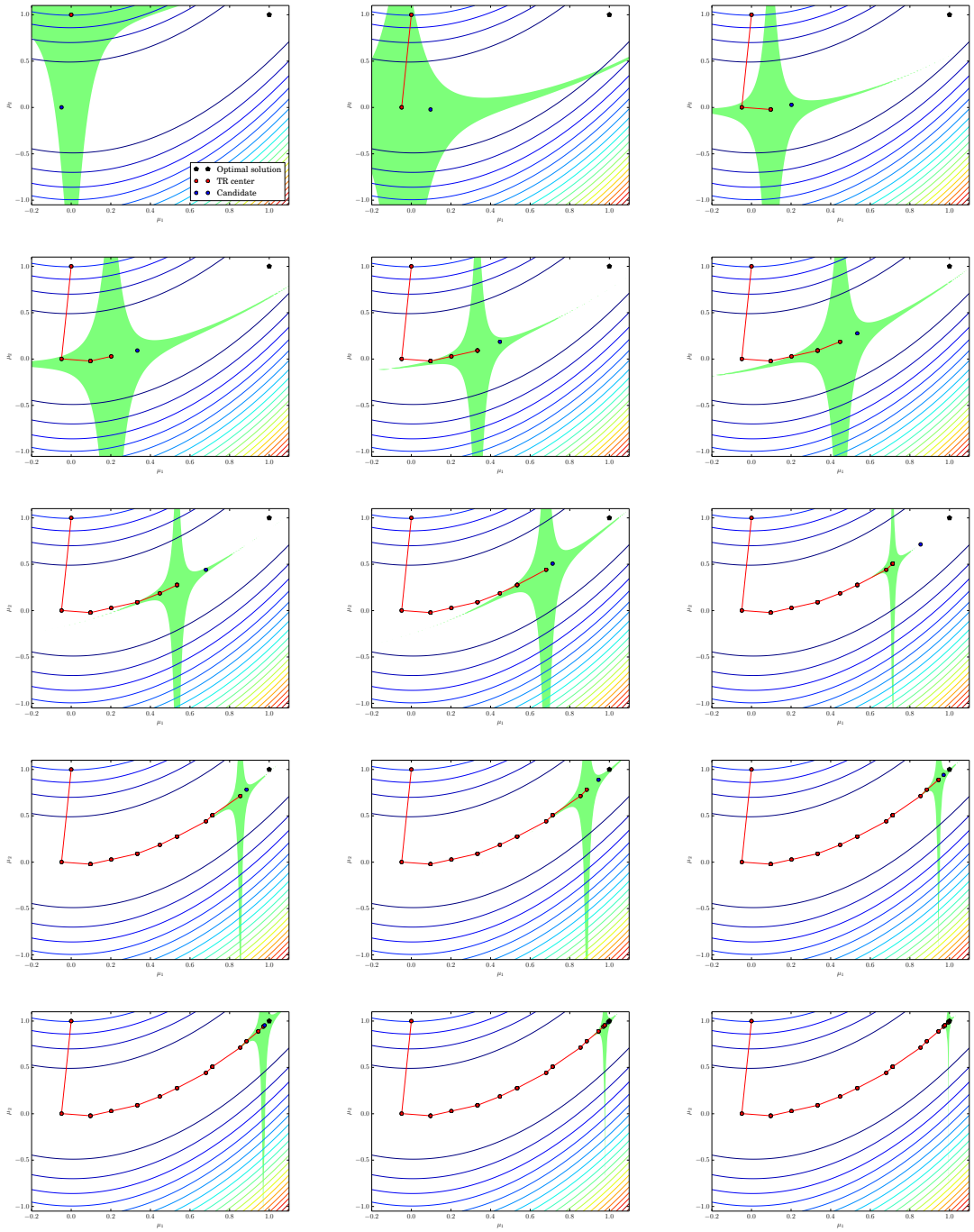


Figure 3.4: Trajectory of Algorithm 1 as applied to the Rosenbrock problem (3.33); iterations proceed from left to right then top to bottom. The contours represent the true function $F(\boldsymbol{\mu})$, the red dots indicate trust region centers $\boldsymbol{\mu}_k$, the blue dots are the candidate for the next trust region center $\hat{\boldsymbol{\mu}}_k$, and the green region indicates the feasible set for the trust region subproblem.

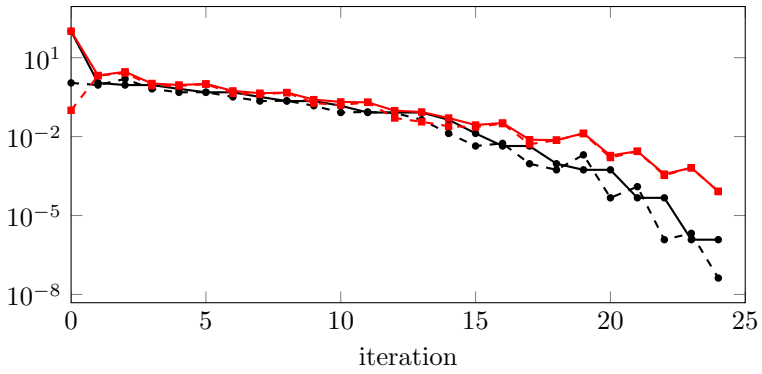


Figure 3.5: Convergence history of the objective quantities using Algorithm 1: $F(\boldsymbol{\mu}_k)$ (—•—), $F(\hat{\boldsymbol{\mu}}_k)$ (- •-), $m_k(\boldsymbol{\mu}_k)$ (—▪—), $m_k(\hat{\boldsymbol{\mu}}_k)$ (- ▪-). Steady progress is made toward the optimal solution, despite the objective and model only agreeing at iteration 0.

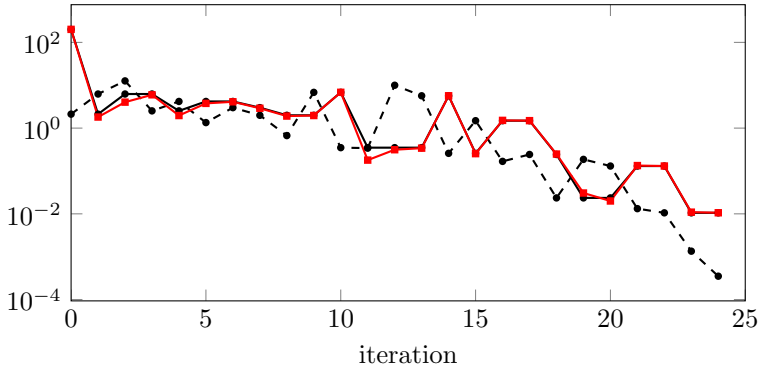


Figure 3.6: Convergence history of gradient quantities using Algorithm 1: $\|\nabla F(\boldsymbol{\mu}_k)\|$ (—•—), $\|\nabla F(\hat{\boldsymbol{\mu}}_k)\|$ (- •-), $\|\nabla m_k(\boldsymbol{\mu}_k)\|$ (—▪—). The gradient of the true objective function decreases 6 orders of magnitude.

Figures 3.5 and 3.6 show the convergence history of the objective and gradient quantities, respectively. From Figure 3.5 it can be seen that the objective function $F(\boldsymbol{\mu})$ continually decreases as the algorithm iterates, despite the fact that the model *values* $m_k(\boldsymbol{\mu})$ do not agree well with $F(\boldsymbol{\mu})$ at either the trust region centers $\boldsymbol{\mu}_k$ or candidates $\hat{\boldsymbol{\mu}}_k$. Figure 3.6 shows that the first-order optimality condition decreases 6 orders of magnitude throughout the iterations. A more detailed report of the convergence history is provided in Table 3.1.

Table 3.1: Convergence history of Algorithm 1 applied to the Rosenbrock problem.

$F(\boldsymbol{\mu}_k)$	$m_k(\boldsymbol{\mu}_k)$	$F(\hat{\boldsymbol{\mu}}_k)$	$m_k(\hat{\boldsymbol{\mu}}_k)$	$\|\nabla F(\boldsymbol{\mu}_k)\|$	ρ_k	Δ_k	Success?
1.0100e+02	1.0150e+02	1.1002e+00	1.0017e-01	2.0001e+02	9.8521e-01	2.0000e+00	True
1.1002e+00	2.1002e+00	9.1177e-01	2.0676e+00	2.1176e+00	5.7778e+00	4.0000e+00	True
9.1177e-01	2.9118e+00	1.5584e+00	2.6152e+00	6.2403e+00	-2.1805e+00	2.6419e-01	False
9.1177e-01	1.0439e+00	6.5172e-01	8.7534e-01	6.2403e+00	1.5431e+00	5.2838e-01	True
6.5172e-01	9.1592e-01	4.8297e-01	8.4624e-01	2.5230e+00	2.4220e+00	1.0568e+00	True
4.8297e-01	1.0114e+00	4.8755e-01	9.8450e-01	4.1851e+00	-1.7086e-01	1.2424e-01	False
4.8297e-01	5.4508e-01	3.2345e-01	4.4402e-01	4.1851e+00	1.5783e+00	2.4847e-01	True
3.2345e-01	4.4769e-01	2.2451e-01	4.0474e-01	2.9992e+00	2.3038e+00	4.9694e-01	True
2.2451e-01	4.7298e-01	2.3107e-01	4.6842e-01	1.9897e+00	-1.4389e+00	5.9337e-02	False
2.2451e-01	2.5418e-01	1.4907e-01	1.8817e-01	1.9897e+00	1.1428e+00	1.1867e-01	True
1.4907e-01	2.0840e-01	8.2419e-02	1.5822e-01	6.8219e+00	1.3281e+00	2.3735e-01	True
8.2419e-02	2.0109e-01	8.6494e-02	2.0100e-01	3.4986e-01	-4.4516e+01	2.8627e-02	False
8.2419e-02	9.6733e-02	7.9495e-02	5.0868e-02	3.4986e-01	6.3763e-02	7.1567e-03	False
8.2419e-02	8.5998e-02	4.3691e-02	3.6534e-02	3.4986e-01	7.8297e-01	1.4313e-02	True
4.3691e-02	5.0847e-02	1.3241e-02	2.4755e-02	5.6732e+00	1.1670e+00	2.8627e-02	True
1.3241e-02	2.7555e-02	4.3950e-03	2.3160e-02	2.5786e-01	2.0128e+00	5.7254e-02	True
4.3950e-03	3.3022e-02	5.5308e-03	3.1229e-02	1.4895e+00	-6.3368e-01	6.4246e-03	False
4.3950e-03	7.6073e-03	9.3057e-04	5.1524e-03	1.4895e+00	1.4112e+00	1.2849e-02	True
9.3057e-04	7.3552e-03	5.4597e-04	7.2723e-03	2.4271e-01	4.6412e+00	2.5699e-02	True
5.4597e-04	1.3395e-02	2.0254e-03	1.2934e-02	2.3541e-02	-3.2065e+00	2.7271e-03	False
5.4597e-04	1.9095e-03	4.7041e-05	1.6024e-03	2.3541e-02	1.6243e+00	5.4543e-03	True
4.7041e-05	2.7742e-03	1.2499e-04	2.7398e-03	1.3032e-01	-2.2700e+00	6.5371e-04	False
4.7041e-05	3.7390e-04	1.2059e-06	3.3688e-04	1.3032e-01	1.2384e+00	1.3074e-03	True
1.2059e-06	6.5492e-04	2.1025e-06	6.5473e-04	1.0560e-02	-4.8064e+00	1.6316e-04	False
1.2059e-06	8.2784e-05	4.2021e-08	8.1934e-05	1.0560e-02	1.3691e+00	3.2631e-04	True

3.2 Nonlinearly Constrained Optimization

This section extends the unconstrained generalized trust region method introduced in Section 3.1.1 to handle nonlinear equality constraints. Consider the nonlinear equality-constrained optimization problem

$$\begin{aligned} \underset{\boldsymbol{\mu}\in\mathbb{R}^{N_{\boldsymbol{\mu}}}}{\text{minimize}} \quad & F(\boldsymbol{\mu}) \\ \text{subject to} \quad & \boldsymbol{c}(\boldsymbol{\mu}) = 0. \end{aligned} \tag{3.41}$$

This is the exact form of the reduced-space PDE-constrained optimization problem in (2.104) without inequality constraints. The feasible set (Definition 3.4) is an important concept in constrained optimization theory as it defines the set of all points that satisfy the constraints of (3.41).

Definition 3.4 (Feasible set)**.** *The set of points that satisfy the constraints of the optimization problem in (3.41)*

$$\Omega \coloneqq \{\boldsymbol{\mu} \in \mathbb{R}^{N_{\boldsymbol{\mu}}} \mid \boldsymbol{c}(\boldsymbol{\mu}) = 0\} \tag{3.42}$$

is called the feasible set*.*

As in the unconstrained case, it is desirable to find the global minimum of (3.41), i.e., the point $\boldsymbol{\mu}^*$ such that $F(\boldsymbol{\mu}^*) \leq F(\boldsymbol{\mu})$ for all $\boldsymbol{\mu} \in \Omega$; however, due to the inherent difficulty of global optimization we settle for *local minima*, as defined in Definition 3.5.

Definition 3.5 (Constrained local minima)**.** *A point $\boldsymbol{\mu}^*$ is a* local minima *of (3.41) if $\boldsymbol{\mu}^* \in \Omega$ and there is a neighborhood $\mathcal{N}$ of $\boldsymbol{\mu}^*$ such that $F(\boldsymbol{\mu}^*) \leq F(\boldsymbol{\mu})$ for all $\boldsymbol{\mu} \in \mathcal{N} \cap \Omega$.*

Before stating the first-order necessary optimality condition, two concepts must be introduced. The first is the concept of constraint qualifications that provide conditions that must be satisfied for the linearized feasible set to resemble the tangent cone [143]. In this work, we solely consider the Linear Independence Constraint Qualifications (Definition 3.6) that requires linear independence of the gradient of the constraints at a particular point.

Definition 3.6 (Linear Independence Constraint Qualification (LICQ))**.** *The LICQ holds at a point $\boldsymbol{\mu} \in \mathbb{R}^{N_{\boldsymbol{\mu}}}$ if the rows of the constraint Jacobian, $\dfrac{\partial \boldsymbol{c}}{\partial \boldsymbol{\mu}}(\boldsymbol{\mu})$ are linearly independent.*

The second concept is the *Lagrangian* (Definition 3.7) that combines the objective and constraints into a single function by introducing auxiliary variables known as *Lagrange multipliers*.

Definition 3.7 (Lagrangian)**.** *The Lagrangian corresponding to the optimization problem in (3.41) is defined as*

$$\mathcal{L}(\boldsymbol{\mu}, \boldsymbol{\tau}) = F(\boldsymbol{\mu}) - \boldsymbol{\tau}^T \boldsymbol{c}(\boldsymbol{\mu}), \tag{3.43}$$

where $\boldsymbol{\tau} \in \mathbb{R}^{N_{\mathbf{c}}}$ is a vector of Lagrange multipliers.

Equipped with these concepts, the first-order necessary optimality conditions for $\boldsymbol{\mu}^*$ to be a local minima (in the sense of Definition 3.5) of (3.41) are stated in Theorem 3.3.

Theorem 3.3 (First-order constrained optimality condition)**.** *Suppose* $\boldsymbol{\mu}^*$ *is a local minima of (3.41), that* F *and* $\boldsymbol{c}_i$ *are continuously differentiable, and LICQ holds at* $\boldsymbol{\mu}*$*. Then there is a Lagrange multiplier vector* $\boldsymbol{\lambda}^*$ *such that*

$$\begin{aligned} \nabla_{\boldsymbol{\mu}} \mathcal{L}(\boldsymbol{\mu}^*, \boldsymbol{\lambda}^*) &= 0 \\ \boldsymbol{c}(\boldsymbol{\mu}^*) &= 0 \\ \boldsymbol{\lambda}_i^* \boldsymbol{c}_i(\boldsymbol{\mu}^*) &= 0 \quad \textit{for} \quad i \in \{1, \dots, N_{\boldsymbol{c}}\}. \end{aligned} \tag{3.44}$$

The first condition requires stationarity of the Lagrangian at a local minima and the second requires the feasibility. The last condition is usually referred to as *complementarity*. There are also second-order necessary and sufficient conditions for $\boldsymbol{\mu}^*$ to be a local minima of (3.41) that involve the Hessian of the Lagrangian [143]. This will not be considered further as this work will primarily be concerned with first-order optimality conditions.

3.2.1 Error-Aware Augmented Lagrangian Multifidelity Trust Region Method

This section extends the multifidelity trust region framework introduced in Section 3.1.1 to handle equality-constrained problems in (3.41). The proposed approach converts the constrained optimization problem in (3.41) to a sequence of unconstrained problems using the the concept of the *augmented Lagrangian*. The unconstrained multifidelity trust region method proposed in Section 3.1.1 is used to solve each unconstrained problem in the sequence for an efficient algorithm that leverages inexpensive approximation models. The augmented Lagrangian corresponding to the optimization problem in (3.41) is

$$\mathcal{L}^\tau(\boldsymbol{\mu}, \boldsymbol{\lambda}) \coloneqq \mathcal{L}(\boldsymbol{\mu}, \boldsymbol{\lambda}) + \tau \boldsymbol{c}(\boldsymbol{\mu})^T \boldsymbol{c}(\boldsymbol{\mu}) = F(\boldsymbol{\mu}) - \boldsymbol{\lambda}^T \boldsymbol{c}(\boldsymbol{\mu}) + \tau \boldsymbol{c}(\boldsymbol{\mu})^T \boldsymbol{c}(\boldsymbol{\mu}), \tag{3.45}$$

where $\tau > 0$ is the *penalty parameter*. A standard result in constrained optimization theory states that, under certain assumptions (Theorem 3.4), there exists a constant $\bar{\tau}$ such that a local minima of (3.41) is a local minima of $\mathcal{L}^\tau(\boldsymbol{\mu}, \boldsymbol{\lambda}^*)$ for $\tau \geq \bar{\tau}$, where $\boldsymbol{\lambda}^*$ are the Lagrange multipliers at the local minima.

Theorem 3.4. *Let* $\boldsymbol{\mu}^*$ *be a local minima of (3.41) with Lagrange multipliers* $\boldsymbol{\lambda}^*$*. If the LICQ holds at* $\boldsymbol{\mu}^*$ *and the second-order sufficient-conditions hold at* $(\boldsymbol{\mu}^*, \boldsymbol{\lambda}^*)$ *(Theorem 12.6 of [143]), there exists* $\bar{\tau}$ *such that* $\boldsymbol{\mu}^*$ *is a local minima of* $\mathcal{L}^\tau(\boldsymbol{\mu}, \boldsymbol{\lambda}^*)$ *for all* $\tau \geq \bar{\tau}$*.*

Proof. See Theorem 17.5 of [143]. □

Therefore, the optimization problem in (3.41) reduces to a sequence of unconstrained optimization problems of the form

$$\underset{\boldsymbol{\mu} \in \mathbb{R}^{N_{\boldsymbol{\mu}}}}{\text{minimize}} \quad \mathcal{L}^{\tau_p}(\boldsymbol{\mu}, \hat{\boldsymbol{\lambda}}_p) \tag{3.46}$$

for a sequence $\tau_p \to \tau > \bar{\tau}$, where $\hat{\boldsymbol{\lambda}}_p$ are Lagrange multiplier estimates, usually taken as

$$\hat{\boldsymbol{\lambda}}_p = \hat{\boldsymbol{\lambda}}_{p-1} - \tau_{p-1}\boldsymbol{c}(\boldsymbol{\mu}^*_{p-1}), \tag{3.47}$$

where $\boldsymbol{\mu}^*_{p-1}$ is the solution of (3.46) at iteration $p-1$. The augmented Lagrangian is employed instead of, e.g., a quadratic penalty function, as equivalence between (3.41) and (3.46) is guaranteed for a *finite* value of the penalty parameter.

The generalized multifidelity trust region method of Section 3.1.1 applies, without modification, to each unconstrained optimization problem in (3.46), i.e., for a fixed τ_p. In this case, the approximation model, $m_k(\boldsymbol{\mu})$, and constraint functions, $\vartheta_k(\boldsymbol{\mu})$ and $\varphi_k(\boldsymbol{\mu})$, must be constructed such that the conditions in (3.12)-(3.15), applied to the augmented Lagrangian in (3.45) hold, that is,

$$\begin{aligned} |\mathcal{L}^{\tau_p}(\boldsymbol{\mu}_k, \hat{\boldsymbol{\lambda}}_p) - \mathcal{L}^{\tau_p}(\boldsymbol{\mu}, \hat{\boldsymbol{\lambda}}_p) + m_k(\boldsymbol{\mu}) - m_k(\boldsymbol{\mu}_k)| &\leq \zeta\vartheta_k(\boldsymbol{\mu}) \qquad \boldsymbol{\mu} \in \mathcal{R}_k \\ \left\|\nabla_{\boldsymbol{\mu}}\mathcal{L}^{\tau_p}(\boldsymbol{\mu}_k, \hat{\boldsymbol{\lambda}}_p) - \nabla m_k(\boldsymbol{\mu}_k)\right\| &\leq \xi\varphi_k(\boldsymbol{\mu}_k) \\ \vartheta_k(\boldsymbol{\mu}_k) &\leq \kappa_\vartheta\Delta_k \\ \varphi_k(\boldsymbol{\mu}_k) &\leq \kappa_\varphi \min\{\|\nabla m_k(\boldsymbol{\mu}_k)\|, \Delta_k\}. \end{aligned} \tag{3.48}$$

Furthermore, the model and constraint functions must satisfy assumptions (AM1)–(AM4). The trust region subproblem in (3.4), based on this model and constraint, is solved using the interior-point method outlined in Section 3.1.2.

Global convergence of this equality-constrained variant of the trust region method of Section 3.1.1 follows trivially from the global convergence of the generalized trust region method (Appendix A) and Theorem 3.4 provided assumptions (AF1)–(AF2) hold for the augmented Lagrangian. Assumption (AF1) holds since the objective F and constraint $\boldsymbol{c}$ are assumed twice-continuously differentiable on $\mathbb{R}^{N_{\boldsymbol{\mu}}}$ and assumption (AF2) holds since (3.45) is bounded below provided F is bounded below.

3.2.2 Numerical Experiment: Contrived

This section closes with the application of Algorithm 1, embedded in the augmented Lagrangian framework of Section 3.2.1, to solve the following optimization problem with a single nonlinear equality constraint

$$\begin{aligned} \underset{\boldsymbol{\mu}\in\mathbb{R}^2}{\text{minimize}} \quad & \boldsymbol{\mu}_1 + \boldsymbol{\mu}_2 \\ \text{subject to} \quad & \boldsymbol{\mu}_1^2 + \boldsymbol{\mu}_2^2 - 2 = 0. \end{aligned} \tag{3.49}$$

The augmented Lagrangian corresponding to this problem, for a fixed penalty τ and Lagrange multiplier estimate λ, is

$$\mathcal{L}^\tau(\boldsymbol{\mu}, \lambda) = \boldsymbol{\mu}_1 + \boldsymbol{\mu}_2 + (\boldsymbol{\mu}_1^2 + \boldsymbol{\mu}_2^2 - 2)\left[\tau(\boldsymbol{\mu}_1^2 + \boldsymbol{\mu}_2^2 - 2) - \lambda\right] \tag{3.50}$$

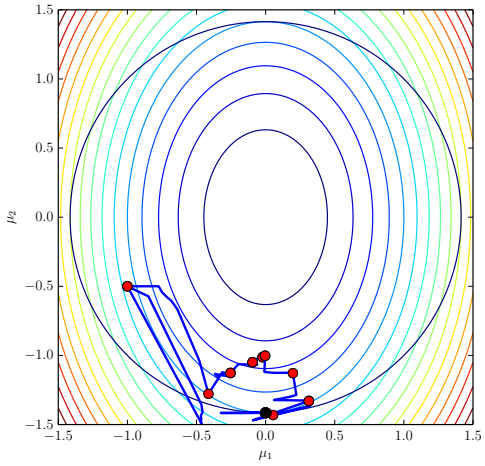


Figure 3.7: Trajectory of Algorithm 1 as applied to the constrained problem (3.49). The contours represent the true function $F(\boldsymbol{\mu})$, the red dots indicate trust region centers, and the blue line is the trajectory of the trust region subproblem.

and the resulting sequence of unconstrained optimization problems are

$$\underset{\boldsymbol{\mu}\in\mathbb{R}^2}{\text{minimize}} \quad \mathcal{L}^{\tau_j}(\boldsymbol{\mu}, \lambda_j), \tag{3.51}$$

which will be solved using Algorithm 1. The model $m_k(\boldsymbol{\mu})$, objective decrease error indicator $\vartheta_k(\boldsymbol{\mu})$, and gradient error indicator $\varphi_k(\boldsymbol{\mu})$ are identical to those in Section 3.1.3 with $F(\boldsymbol{\mu})$ replaced with $\mathcal{L}^{\tau_j}(\boldsymbol{\mu}, \lambda_j)$. The objective and gradient errors ϵ_k and δ_k are similarly chosen using (3.40).

With these definitions of $m_k(\boldsymbol{\mu})$, $\vartheta_k(\boldsymbol{\mu})$, and $\varphi_k(\boldsymbol{\mu})$ all ingredients necessary for the complete description of Algorithm 1 are set. The trust region subproblem, for fixed τ_j and λ_j, is solved using the BFGS interior-point method described in Section 3.1.2. The trajectory of the optimization iterations—including the progress of the trust region centers and the trajectory of each trust region subproblem—are shown in Figure 3.7. These iterations are aggregated over three augmented Lagrangian iterations corresponding to $\tau_0 = 10^{-4}$, $\tau_1 = 10^{-5}$, and $\tau_2 = 10^{-6}$, with λ_j updated according to (3.47) and initialized with $\lambda_0 = 0$. Figure 3.8 provides additional insight to the Algorithm 1 by showing (selected) individual iterations, including the trust region center, candidate step, and feasible region for the trust region subproblem. Again, notice the substantial difference between the shape of the trust regions in Figure 3.8 with traditional trust regions that are spheres or ellipsoids. From both of these figures, it is clear the iterations converge to a feasible point, as expected from the augmented Lagrangian framework.

Figures 3.9 and 3.10 show the convergence history of the objective and gradient quantities, respectively, aggregated over all three augmented Lagrangian iterations. A dashed vertical line separates augmented Lagrangian iterations. From Figure 3.9 it can be seen that the objective function $\mathcal{L}^{\tau_j}(\boldsymbol{\mu}, \lambda_j)$ continually decreases within an augmented Lagrangian iteration, despite the

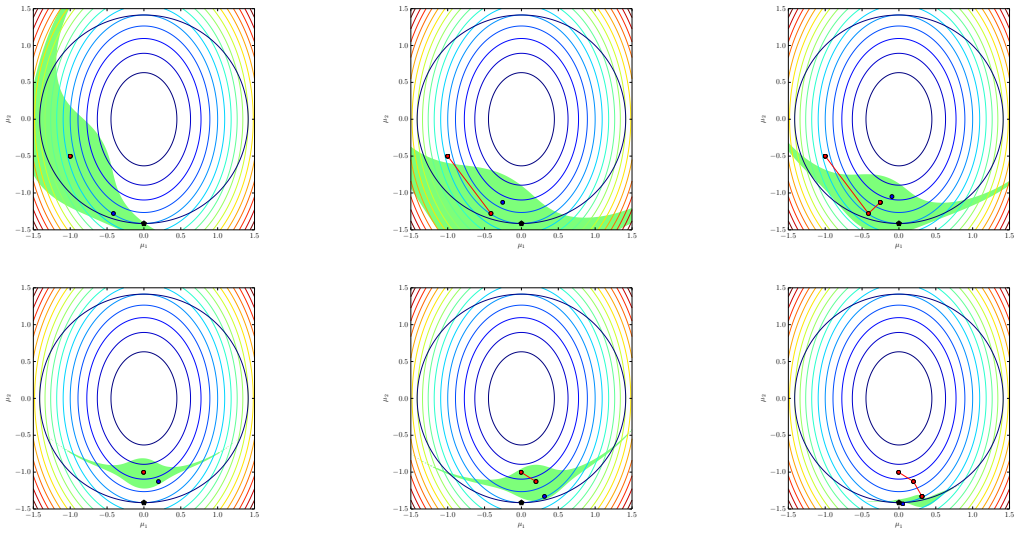


Figure 3.8: Trajectory of Algorithm 1 as applied to the constrained problem (3.49) embedded in the augmented Lagrangian framework. The contours represent the true function $F(\boldsymbol{\mu})$, the red dots indicate trust region centers $\boldsymbol{\mu}_k$, the blue dots are the candidate for the next trust region center $\hat{\boldsymbol{\mu}}_k$, and the green region indicates the feasible set for the trust region subproblem.

fact that the model *values* $m_k(\boldsymbol{\mu})$ do not agree with $\mathcal{L}^{\tau_j}(\boldsymbol{\mu}, \lambda_j)$ at either the trust region centers $\boldsymbol{\mu}_k$ or candidates $\hat{\boldsymbol{\mu}}_k$ when the algorithm is far from convergence. Figure 3.10 shows that the first-order optimality condition decreases $3-4$ orders of magnitude throughout these iterations. A more detailed report of the convergence history is provided in Table 3.2.

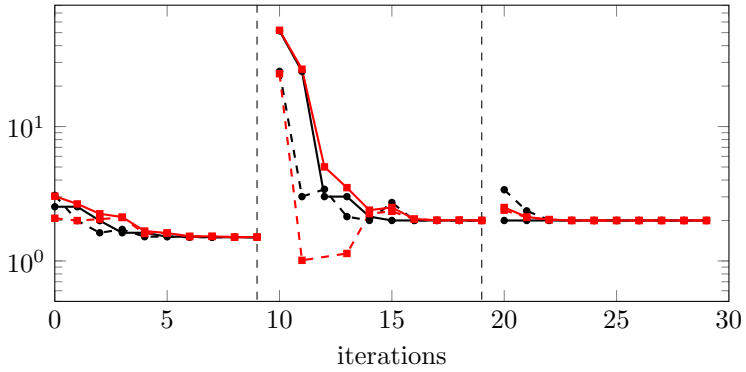


Figure 3.9: Convergence history of the augmented Lagrangian objective quantities using Algorithm 1: $\mathcal{L}^{\tau_j}(\boldsymbol{\mu}_k)$ (—•—), $\mathcal{L}^{\tau_j}(\hat{\boldsymbol{\mu}}_k)$ (-•-), $m_k(\boldsymbol{\mu}_k)$ (—▪—), $m_k(\hat{\boldsymbol{\mu}}_k)$ (-▪-). The three augmented Lagrangian iterations are separated by a vertical dashed line with the following penalty parameters: $\tau_0 = 10^{-4}$ (iterations $0-9$), $\tau_1 = 10^{-5}$ (iterations $10-19$), $\tau_2 = 10^{-6}$ (iterations $20-29$).

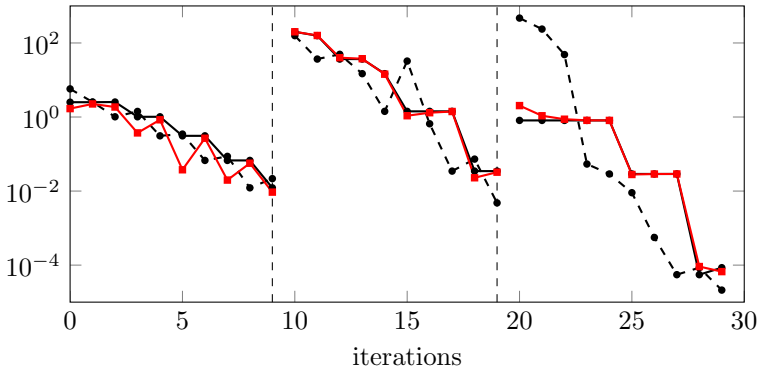


Figure 3.10: Convergence history of the augmented Lagrangian gradient quantities using Algorithm 1: $\|\nabla\mathcal{L}^{\tau_j}(\boldsymbol{\mu}_k)\|$ (—•—), $\|\nabla\mathcal{L}^{\tau_j}(\hat{\boldsymbol{\mu}}_k)\|$ (-•-), $\|\nabla m_k(\boldsymbol{\mu}_k)\|$ (—▪—). The three augmented Lagrangian iterations are separated by a vertical dashed line with the following penalty parameters: $\tau_0 = 10^{-4}$ (iterations $0-9$), $\tau_1 = 10^{-5}$ (iterations $10-19$), $\tau_2 = 10^{-6}$ (iterations $20-29$). For each augmented Lagrangian iteration, the gradient of the true augmented Lagrangian (for fixed τ_j) decreases $3-4$ orders of magnitude.

Table 3.2: Convergence history of Algorithm 1 applied to the constrained problem (3.49). Iterations $0-9$: $\tau_0 = 10^{-4}$, iterations $10-19$: $\tau_1 = 10^{-5}$, iterations $20-29$: $\tau_2 = 10^{-6}$. The norm of the gradient of $\mathcal{L}^{\tau_j}(\boldsymbol{\mu})$, for fixed τ_j, decreases $3-4$ orders of magnitude throughout the iterations despite the values of $\mathcal{L}^{\tau_j}(\boldsymbol{\mu}_k)$ and $m_k(\boldsymbol{\mu}_k)$ or $\mathcal{L}^{\tau_j}(\hat{\boldsymbol{\mu}}_k)$ and $m_k(\hat{\boldsymbol{\mu}}_k)$ not being close until near convergence.

$\mathcal{L}^{\tau_j}(\boldsymbol{\mu}_k)$	$m_k(\boldsymbol{\mu}_k)$	$\mathcal{L}^{\tau_j}(\hat{\boldsymbol{\mu}}_k)$	$m_k(\hat{\boldsymbol{\mu}}_k)$	$\|\nabla\mathcal{L}^{\tau_j}(\boldsymbol{\mu}_k)\|$	ρ_k	Δ_k	Success?
2.5312e+00	3.0312e+00	3.0781e+00	2.0781e+00	2.5125e+00	-5.7368e-01	2.5000e-01	False
2.5312e+00	2.6562e+00	1.9946e+00	1.9971e+00	2.5125e+00	8.1405e-01	5.0000e-01	True
1.9946e+00	2.2446e+00	1.6205e+00	2.0460e+00	2.5410e+00	1.8838e+00	1.0000e+00	True
1.6205e+00	2.1205e+00	1.7166e+00	2.1042e+00	1.0140e+00	-5.9067e+00	9.6906e-02	False
1.6205e+00	1.6689e+00	1.5149e+00	1.5999e+00	1.0140e+00	1.5304e+00	1.9381e-01	True
1.5149e+00	1.6118e+00	1.5188e+00	1.6115e+00	3.1026e-01	-1.4365e+01	2.3193e-02	False
1.5149e+00	1.5265e+00	1.5008e+00	1.5163e+00	3.1026e-01	1.3860e+00	4.6386e-02	True
1.5008e+00	1.5240e+00	1.5013e+00	1.5239e+00	6.7233e-02	-6.9139e+00	5.6381e-03	False
1.5008e+00	1.5036e+00	1.5000e+00	1.5031e+00	6.7233e-02	1.4075e+00	1.1276e-02	True
1.5000e+00	1.5057e+00	1.5001e+00	1.5056e+00	1.2228e-02	-3.5785e+00	1.3905e-03	False
5.1554e+01	5.2054e+01	2.5730e+01	2.4730e+01	1.9955e+02	9.4510e-01	2.0000e+00	True
2.5730e+01	2.6730e+01	3.0107e+00	1.0107e+00	1.5760e+02	8.8336e-01	4.0000e+00	True
3.0107e+00	5.0107e+00	3.4134e+00	-5.8657e-01	3.6765e+01	-7.1943e-02	1.0000e+00	False
3.0107e+00	3.5107e+00	2.1365e+00	1.1365e+00	3.6765e+01	3.6822e-01	5.0000e-01	True
2.1365e+00	2.3865e+00	2.0013e+00	2.2475e+00	1.4813e+01	9.7295e-01	1.0000e+00	True
2.0013e+00	2.5013e+00	2.7229e+00	2.3393e+00	1.4232e+00	-4.4558e+00	9.5895e-02	False
2.0013e+00	2.0492e+00	2.0018e+00	2.0454e+00	1.4232e+00	-1.4312e-01	1.0887e-02	False
2.0013e+00	2.0067e+00	2.0000e+00	2.0054e+00	1.4232e+00	9.5556e-01	2.1775e-02	True
2.0000e+00	2.0109e+00	2.0002e+00	2.0108e+00	3.4607e-02	-1.6948e+00	2.6365e-03	False
2.0000e+00	2.0013e+00	2.0000e+00	2.0013e+00	3.4607e-02	1.6271e+00	5.2730e-03	True
2.0000e+00	2.5000e+00	3.3861e+00	2.3862e+00	8.0931e-01	-1.2175e+01	2.5000e-01	False
2.0000e+00	2.1250e+00	2.3588e+00	2.1088e+00	8.0931e-01	-2.2161e+01	6.2500e-02	False
2.0000e+00	2.0313e+00	2.0165e+00	2.0300e+00	8.0931e-01	-1.3440e+01	3.3820e-03	False
2.0000e+00	2.0017e+00	2.0000e+00	2.0017e+00	8.0931e-01	-3.8261e-02	4.2134e-04	False
2.0000e+00	2.0002e+00	2.0000e+00	2.0002e+00	8.0931e-01	1.0374e+00	8.4269e-04	True
2.0000e+00	2.0004e+00	2.0000e+00	2.0004e+00	2.8971e-02	-8.1375e-01	1.0521e-04	False
2.0000e+00	2.0001e+00	2.0000e+00	2.0001e+00	2.8971e-02	2.2565e-01	1.3148e-05	False
2.0000e+00	2.0000e+00	2.0000e+00	2.0000e+00	2.8971e-02	9.5811e-01	2.6296e-05	True
2.0000e+00	2.0000e+00	2.0000e+00	2.0000e+00	5.5167e-05	3.0124e-01	1.3148e-05	True
2.0000e+00	2.0000e+00	2.0000e+00	2.0000e+00	8.5278e-05	6.9467e+00	2.6296e-05	True

Chapter 4

Projection-Based Model Reduction

The trust region method introduced in the previous chapter is general in that *any* approximation model equipped with error bounds (3.12) and (3.13) can be employed. Projection-based model reduction has been shown to be a promising method to dramatically reduce the cost—in terms of computational time and resources—of PDE simulations, while retaining a high degree of fidelity [31, 198]. In this approach, the solution of the partial differential equation is sought in a *well-chosen, low-dimensional* (possibly affine) *trial* subspace by solving a reduced representation of the governing equations, usually a projection onto a *test* subspace. It has been shown to yield *Reduced-Order Models* (ROMs) that are $\mathcal{O}(10^5)$ smaller (in terms of number of degrees of freedom) and faster to solve than the original discretized PDE [31, 198], which will be called the *High-Dimensional Model* (HDM). This makes reduced-order models a promising candidate for the trust region approximation model from the previous chapter.

This chapter provides background necessary to use projection-based reduced-order models as the approximation model in the error-aware trust region method of Chapter 3. The discussion will include the derivation of the primal, sensitivity, and adjoint reduced-order model and computable error bounds. While some of this discussion is a review, there are novel contributions regarding the formulation of a minimum-residual reduced-order model for sensitivity and adjoint equations that guarantee the reduced quantities optimally approximate the HDM counterparts. When the governing equations are *nonlinear*, a critical bottleneck exists in the evaluation of the ROM that will destroy nearly all resource reduction potential. To eliminate this bottleneck, *hyperreduction* methods [17, 175, 115, 41, 31, 59] have been developed that introduce an additional level of approximation. Another contribution of this chapter is the extension of the minimum-residual formulation of the primal, sensitivity, and adjoint reduced-order model to a specific hyperreduction method known as *collocation*.

4.1 Global Reduced-Order Models

For simplicity, this section will consider a static, deterministic partial differential equation at the *discrete* level

$$\boldsymbol{r}(\boldsymbol{u}, \boldsymbol{\mu}) = 0, \tag{4.1}$$

where $\boldsymbol{u} \in \mathbb{R}^{N_{\boldsymbol{u}}}$ is the state vector, $\boldsymbol{\mu} \in \mathbb{R}^{N_{\boldsymbol{\mu}}}$ is the parameter vector, and $\boldsymbol{r} : \mathbb{R}^{N_{\boldsymbol{u}}} \times \mathbb{R}^{N_{\boldsymbol{\mu}}} \to \mathbb{R}^{N_{\boldsymbol{u}}}$ is the discrete PDE. Most of the developments will extend to the time-dependent case where $\boldsymbol{r}$ is the governing equation and $\boldsymbol{u}$ is the state vector at a *single* time step. The fundamental ansatz of (global) projection-based model reduction is that the state vector $\boldsymbol{u}$ can be well-approximated in a single *low-dimensional* subspace

$$\boldsymbol{u} = \boldsymbol{\Phi}\boldsymbol{u}_r, \tag{4.2}$$

where $\boldsymbol{\Phi} \in \mathbb{R}^{N_{\boldsymbol{u}} \times k_{\boldsymbol{u}}}$ is the reduced-order basis (basis for the trial subspace), $\boldsymbol{u}_r \in \mathbb{R}^{k_{\boldsymbol{u}}}$ are the reduced coordinates of $\boldsymbol{u}$ in the basis $\boldsymbol{\Phi}$, and $k_{\boldsymbol{u}} \ll N_{\boldsymbol{u}}$. It is also common to consider an affine expansion in (4.2); however, this generalization will not significantly contribute to the following developments and is omitted for clarity. Subsequent sections will use this ansatz to arrive at the primal, sensitivity, and adjoint form of the reduced-order model. A central focus will be the concept of *minimum-residual* reduced-order models—defined such that its solution coincides with the first-order optimality condition of residual minimization over the trial subspace in some norm—as they possess desirable properties such as *monotonicity* and *interpolation*.

4.1.1 Primal Formulation

The general form of the projection-based reduced-order model is obtained by substituting the ansatz (4.2) into the governing equation (4.1) and projecting the resulting *overdetermined* nonlinear system of equations onto a test subspace spanned by the columns of the basis $\boldsymbol{\Psi} \in \mathbb{R}^{N_{\boldsymbol{u}} \times k_{\boldsymbol{u}}}$

$$\boldsymbol{r}_r(\boldsymbol{u}_r, \boldsymbol{\mu}) \coloneqq \boldsymbol{\Psi}^T \boldsymbol{r}(\boldsymbol{\Phi}\boldsymbol{u}_r, \boldsymbol{\mu}) = 0, \tag{4.3}$$

where $\boldsymbol{r}_r : \mathbb{R}^{k_{\boldsymbol{u}}} \times \mathbb{R}^{N_{\boldsymbol{\mu}}} \to \mathbb{R}^{k_{\boldsymbol{u}}}$ is a nonlinear system of equations with $k_{\boldsymbol{u}}$ equations and unknowns. Define $\boldsymbol{u}_r(\boldsymbol{\mu}; \boldsymbol{\Phi}, \boldsymbol{\Psi})$ implicitly as the solution of $\boldsymbol{r}_r(\cdot, \boldsymbol{\mu}) = 0$—the Implicit Function Theorem (Theorem 2.1) guarantees the existence of such a function and its smoothness with respect to $\boldsymbol{\mu}$. In the remainder, the notation $\boldsymbol{u}_r(\boldsymbol{\mu}; \boldsymbol{\Phi}, \boldsymbol{\Psi})$ will be simplified to $\boldsymbol{u}_r(\boldsymbol{\mu})$ when there is no risk of confusion regarding the choice of test and trial basis. The reduced coordinates must be reconstructed in the full space according to $\boldsymbol{\Phi}\boldsymbol{u}_r$ prior to the evaluation of a quantity of interest, which leads to the definition of the reduced quantity of interest $f_r : \mathbb{R}^{k_{\boldsymbol{u}}} \times \mathbb{R}^{N_{\boldsymbol{\mu}}} \to \mathbb{R}$ as

$$f_r(\boldsymbol{u}_r, \boldsymbol{\mu}; \boldsymbol{\Phi}) \coloneqq f(\boldsymbol{\Phi}\boldsymbol{u}_r, \boldsymbol{\mu}). \tag{4.4}$$

The reduced quantity of interest becomes purely a function of $\boldsymbol{\mu}$ when the implicit definition $\boldsymbol{u}_r(\boldsymbol{\mu}; \boldsymbol{\Phi}, \boldsymbol{\Psi})$ is used in the above equation, i.e., when the reduced QoI is only evaluated at solutions

of the reduced-order model (4.3),

$$F_r(\boldsymbol{\mu};\, \boldsymbol{\Phi},\, \boldsymbol{\Psi}) \coloneqq f(\boldsymbol{\Phi}\boldsymbol{u}_r(\boldsymbol{\mu};\, \boldsymbol{\Phi},\, \boldsymbol{\Psi}),\, \boldsymbol{\mu}). \tag{4.5}$$

The implicit function $F_r(\,\cdot\,;\, \boldsymbol{\Phi},\, \boldsymbol{\Psi}) : \mathbb{R}^{N_{\boldsymbol{\mu}}} \to \mathbb{R}$ is the reduced-order model approximation of the quantity of interest $F : \mathbb{R}^{N_{\boldsymbol{\mu}}} \to \mathbb{R}$, defined as $F(\boldsymbol{\mu}) \coloneqq f(\boldsymbol{u}(\boldsymbol{\mu}),\, \boldsymbol{\mu})$, where $\boldsymbol{u}(\boldsymbol{\mu})$ is the solution of $\boldsymbol{r}(\,\cdot\,,\, \boldsymbol{\mu}) = 0$. The notation for the reduced quantity of interest in (4.4) and (4.5) will be simplified to $f_r(\boldsymbol{u}_r,\, \boldsymbol{\mu})$ and $F_r(\boldsymbol{\mu})$, respectively, when there is no risk confusion.

In the general case, the test basis may be non-constant, i.e., $\boldsymbol{\Psi} = \boldsymbol{\Psi}(\boldsymbol{u},\, \boldsymbol{\mu})$. Two common choices for the test basis are

$$\boldsymbol{\Psi} = \boldsymbol{\Phi} \qquad \text{and} \qquad \boldsymbol{\Psi} = \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}(\boldsymbol{\Phi}\boldsymbol{u}_r,\, \boldsymbol{\mu})\boldsymbol{\Phi}, \tag{4.6}$$

which correspond to a Galerkin and Least-Squares Petrov-Galerkin [28, 31] projection, respectively. At this point, there have been no restrictions placed on either the test or trial bases—aside from the implicit requirement that they are valid bases, i.e., their columns are linearly independent—nor has any relationship between these bases been specified. Next, the concept of minimum-residual reduced-order models will be introduced that equips the reduced-order models with desirable properties: (1) *monotonicity*—the quality of the solution can only improve (in some well-defined metric) as the trial space is hierarchically refined and (2) *interpolation*—the reduced-order model will recover the HDM solution if it lies in the trial space.

Definition 4.1 (Minimum-Residual Property)**.** *A reduced-order model possesses the minimum-residual property if the solution satisfies the first-order optimality conditions of the following residual minimization problem*

$$\underset{\boldsymbol{u}_r \in \mathbb{R}^{k_{\boldsymbol{u}}}}{\text{minimize}} \;\; \frac{1}{2} \left\| \boldsymbol{r}(\boldsymbol{\Phi}\boldsymbol{u}_r,\, \boldsymbol{\mu}) \right\|_{\boldsymbol{\Theta}}^2 \tag{4.7}$$

for some symmetric positive-definite $\boldsymbol{\Theta} \in \mathbb{R}^{N_{\boldsymbol{u}} \times N_{\boldsymbol{u}}}$.

Proposition 4.1. *Let* $(\boldsymbol{\Phi},\, \boldsymbol{\Psi},\, \boldsymbol{\Theta})$ *define a minimum-residual reduced-order model whose solution coincides with the global minimum of (4.7). Then, the following properties hold for any* $\boldsymbol{\mu} \in \mathbb{R}^{N_{\boldsymbol{\mu}}}$*:*

- *(Optimality) For any* $\boldsymbol{u} \in \mathrm{col}(\boldsymbol{\Phi})$*,*

$$\left\| \boldsymbol{r}(\boldsymbol{\Phi}\boldsymbol{u}_r(\boldsymbol{\mu};\, \boldsymbol{\Phi},\, \boldsymbol{\Psi}),\, \boldsymbol{\mu}) \right\|_{\boldsymbol{\Theta}} \leq \left\| \boldsymbol{r}(\boldsymbol{u},\, \boldsymbol{\mu}) \right\|_{\boldsymbol{\Theta}} \tag{4.8}$$

- *(Monotonicity) Let* $(\boldsymbol{\Phi}',\, \boldsymbol{\Psi}')$ *define a projection-based reduced-order model such that* $\mathrm{col}(\boldsymbol{\Phi}') \subseteq \mathrm{col}(\boldsymbol{\Phi})$*, then*

$$\left\| \boldsymbol{r}(\boldsymbol{\Phi}\boldsymbol{u}_r(\boldsymbol{\mu};\, \boldsymbol{\Phi},\, \boldsymbol{\Psi}),\, \boldsymbol{\mu}) \right\|_{\boldsymbol{\Theta}} \leq \left\| \boldsymbol{r}(\boldsymbol{\Phi}'\boldsymbol{u}_r(\boldsymbol{\mu};\, \boldsymbol{\Phi}',\, \boldsymbol{\Psi}'),\, \boldsymbol{\mu}) \right\|_{\boldsymbol{\Theta}} \tag{4.9}$$

- *(Interpolation) If* $\boldsymbol{u}(\boldsymbol{\mu}) \in \mathrm{col}(\boldsymbol{\Phi})$*, then*

$$\boldsymbol{r}(\boldsymbol{\Phi}\boldsymbol{u}_r(\boldsymbol{\mu};\, \boldsymbol{\Phi},\, \boldsymbol{\Psi}),\, \boldsymbol{\mu}) = 0 \qquad \textit{and} \qquad \boldsymbol{u}(\boldsymbol{\mu}) = \boldsymbol{\Phi}\boldsymbol{u}_r(\boldsymbol{\mu};\, \boldsymbol{\Phi},\, \boldsymbol{\Psi}) \tag{4.10}$$

Proof. Optimality follows trivially from the fact that $\boldsymbol{u}_r(\boldsymbol{\mu};\, \boldsymbol{\Phi},\, \boldsymbol{\Psi})$ is the global minima of the optimization problem in (4.7). A simple application of the optimality property to $\boldsymbol{u} = \boldsymbol{\Phi}'\boldsymbol{u}_r(\boldsymbol{\mu};\, \boldsymbol{\Phi}',\, \boldsymbol{\Psi}') \in \text{col}(\boldsymbol{\Phi}') \subseteq \text{col}(\boldsymbol{\Phi})$ leads to monotonicity. Finally, if the exact solution of $\boldsymbol{r}(\,\cdot\,,\, \boldsymbol{\mu}) = 0$ is contained in the columnspace of $\boldsymbol{\Phi}$, i.e., $\boldsymbol{u}(\boldsymbol{\mu}) \in \text{col}(\boldsymbol{\Phi})$, the optimality property implies

$$\|\boldsymbol{r}(\boldsymbol{\Phi}\boldsymbol{u}_r(\boldsymbol{\mu};\, \boldsymbol{\Phi},\, \boldsymbol{\Psi}),\, \boldsymbol{\mu})\|_{\boldsymbol{\Theta}} \leq \|\boldsymbol{r}(\boldsymbol{u}(\boldsymbol{\mu}),\, \boldsymbol{\mu})\|_{\boldsymbol{\Theta}} = 0. \tag{4.11}$$

This result, along with the assumed uniqueness of solutions of $\boldsymbol{r}(\,\cdot\,,\, \boldsymbol{\mu}) = 0$ (Assumption 2.2), leads to the interpolation property in (4.10). □

Remark. *The results in Proposition 4.1 only hold if the solution of the minimum-residual reduced-order model coincides with the* global *solution of the optimization problem in (4.7). In general this is only guaranteed if the optimization problem is* convex*, which will be the case if the governing equation is* affine *in its first argument, i.e., $\boldsymbol{r}(\boldsymbol{u},\, \boldsymbol{\mu}) = \boldsymbol{A}(\boldsymbol{\mu})\boldsymbol{u} + \boldsymbol{b}(\boldsymbol{\mu})$, where $\boldsymbol{A}(\boldsymbol{\mu}) \in \mathbb{R}^{N_{\boldsymbol{u}} \times N_{\boldsymbol{u}}}$ and $\boldsymbol{b}(\boldsymbol{\mu}) \in \mathbb{R}^{N_{\boldsymbol{u}}}$. When the optimization problem in (4.7) is non-convex, the stationary point that will be found by the minimum-residual reduced-order model will not necessarily be the global minima of (4.7). A heuristic that, in practice, is usually sufficient to lead to the results in Proposition 4.1 (optimality, monotonicity, and interpolation) is to initialize the reduced-order model solver with a quality starting point. Due to the required training phase in model reduction (Section 4.3), a reasonable starting point can typically be obtained via interpolation of the training data [198].*

From Appendix B, the approximation of the reduced quantity of interest is equipped with a residual-based error bound (Lemma B.4) that takes the form

$$|F(\boldsymbol{\mu}) - F_r(\boldsymbol{\mu})| \leq \kappa \, \|\boldsymbol{r}(\boldsymbol{\Phi}\boldsymbol{u}_r(\boldsymbol{\mu}),\, \boldsymbol{\mu})\| \leq \kappa' \, \|\boldsymbol{r}(\boldsymbol{\Phi}\boldsymbol{u}_r(\boldsymbol{\mu}),\, \boldsymbol{\mu})\|_{\boldsymbol{\Theta}} \tag{4.12}$$

for some constants $\kappa,\, \kappa' > 0$, where the second inequality follows from norm equivalence in finite dimensions. The residual-based error bound illuminates one motivation behind the minimum-residual formulation: it minimizes the error bound over the columnspace of $\boldsymbol{\Phi}$.

A general relationship between projection-based reduced-order models and minimum-residual reduced-order models (Definition 4.1) is established by matching terms in the first-order optimality condition of (4.7), i.e.,

$$\left[\frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}(\boldsymbol{\Phi}\boldsymbol{u}_r,\, \boldsymbol{\mu})\boldsymbol{\Phi}\right]^T \boldsymbol{\Theta}\boldsymbol{r}(\boldsymbol{\Phi}\boldsymbol{u}_r,\, \boldsymbol{\mu}) = 0 \tag{4.13}$$

with the form of the projection-based reduced-order model in (4.3). From these two equations, the relationship

$$\boldsymbol{\Psi}(\boldsymbol{u},\, \boldsymbol{\mu}) = \boldsymbol{\Theta}\frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}(\boldsymbol{u},\, \boldsymbol{\mu})\boldsymbol{\Phi} \tag{4.14}$$

is sufficient for a general projection-based reduced-order model in (4.3) to possess the minimum-residual property.

The LSPG reduced-order model in (4.6), i.e., $\boldsymbol{\Psi}(\boldsymbol{u},\, \boldsymbol{\mu}) = \dfrac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}(\boldsymbol{u},\, \boldsymbol{\mu})\boldsymbol{\Phi}$, satisfies the condition in (4.14) with $\boldsymbol{\Theta} = \boldsymbol{I}$, where $\boldsymbol{I}$ is the $N_{\boldsymbol{u}}$ identity matrix, and therefore possesses the minimum-residual

property. For problems with symmetric positive-definite Jacobian matrices

$$\frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}(\boldsymbol{u}, \boldsymbol{\mu}) \succ 0 \qquad \forall \boldsymbol{u} \in \mathbb{R}^{N_{\boldsymbol{u}}}, \boldsymbol{\mu} \in \mathbb{R}^{N_{\boldsymbol{\mu}}}$$

the Galerkin reduced-order model in (4.6), i.e., $\boldsymbol{\Psi} = \boldsymbol{\Phi}$, possesses the minimum-residual property in the metric defined by the Jacobian inverse transpose evaluated at the (reconstructed) solution of the Galerkin reduced-order model ($\boldsymbol{\Phi}\boldsymbol{u}_r(\boldsymbol{\mu}; \boldsymbol{\Phi}, \boldsymbol{\Phi})$), i.e.,

$$\boldsymbol{\Theta} = \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}(\boldsymbol{\Phi}\boldsymbol{u}_r(\boldsymbol{\mu}; \boldsymbol{\Phi}, \boldsymbol{\Phi}), \boldsymbol{\mu})^{-T}. \tag{4.15}$$

This is a valid metric since: (1) the Jacobian is evaluated at a specific state and (2) the Jacobian is symmetric positive-definite at any state (by assumption) and therefore its inverse transpose is symmetric positive-definite. This metric reduces the first-order optimality conditions (4.13) of the residual minimization problem in (4.7) to: find $\boldsymbol{y} \in \mathbb{R}^{k_{\boldsymbol{u}}}$ such that

$$\left[\frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}(\boldsymbol{\Phi}\boldsymbol{y}, \boldsymbol{\mu})\boldsymbol{\Phi}\right]^T \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}(\boldsymbol{\Phi}\boldsymbol{u}_r(\boldsymbol{\mu}; \boldsymbol{\Phi}, \boldsymbol{\Phi}), \boldsymbol{\mu})^{-T} \boldsymbol{r}(\boldsymbol{\Phi}\boldsymbol{y}, \boldsymbol{\mu}) = 0$$

for a fixed $\boldsymbol{\mu} \in \mathbb{R}^{N_{\boldsymbol{\mu}}}$. It is easily verified that the solution of the Galerkin reduced-order model satisfies the above equation, i.e., with $\boldsymbol{y} = \boldsymbol{u}_r(\boldsymbol{\mu}; \boldsymbol{\Phi}, \boldsymbol{\Phi})$. Therefore Galerkin reduced-order models possess the minimum-residual property in the metric in (4.15) for problems with symmetric positive-definite Jacobians.

Remark. *To this point, minimum-residual reduced-order models have been interpreted as a specific projection of the governing equations*

$$\boldsymbol{r}(\boldsymbol{u}, \boldsymbol{\mu}) = 0.$$

Given the existence and uniqueness assumption (Assumption 2.2) regarding solutions of the above equation, it can equivalently be formulated as the solution of the minimum-residual optimization problem

$$\underset{\boldsymbol{u} \in \mathbb{R}^{N_{\boldsymbol{u}}}}{\text{minimize}} \quad \frac{1}{2} \left\| \boldsymbol{r}(\boldsymbol{u}, \boldsymbol{\mu}) \right\|_{\boldsymbol{\Theta}}^2, \tag{4.16}$$

where $\boldsymbol{\Theta}$ is a symmetric, positive-definite matrix, with first-order optimality condition

$$\frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}(\boldsymbol{u}, \boldsymbol{\mu})^T \boldsymbol{\Theta} \boldsymbol{r}(\boldsymbol{u}, \boldsymbol{\mu}) = 0. \tag{4.17}$$

From this equation and (4.13), it is clear that minimum-residual reduced-order models are equivalently derived as a Galerkin *projection, with $\boldsymbol{\Phi}$ as the test and trial basis, of the governing equations in minimum-residual form (4.17).*

4.1.2 Exact and Minimum-Residual Sensitivity Formulation

The intention of this work is to use the reduced-order models of the previous section in a gradient-based, reduced-space PDE-constrained optimization setting, in an attempt to reduce the overall computational cost. This requires a discussion regarding gradients of the reduced quantities of interest. Both the sensitivity and adjoint approaches will be detailed for this purpose. Following the procedure outlined in Section 2.3.3, the total derivative of (4.5) is expanded as

$$\nabla F_r(\boldsymbol{\mu};\, \boldsymbol{\Phi},\, \boldsymbol{\Psi}) = \frac{\partial f}{\partial \boldsymbol{\mu}}(\boldsymbol{\Phi}\boldsymbol{u}_r(\boldsymbol{\mu};\, \boldsymbol{\Phi},\, \boldsymbol{\Psi}),\, \boldsymbol{\mu}) + \frac{\partial f}{\partial \boldsymbol{u}}(\boldsymbol{\Phi}\boldsymbol{u}_r(\boldsymbol{\mu};\, \boldsymbol{\Phi},\, \boldsymbol{\Psi}),\, \boldsymbol{\mu})\boldsymbol{\Phi}\frac{\partial \boldsymbol{u}_r}{\partial \boldsymbol{\mu}}(\boldsymbol{\mu};\, \boldsymbol{\Phi},\, \boldsymbol{\Psi}). \tag{4.18}$$

The reduced sensitivities $\frac{\partial \boldsymbol{u}_r}{\partial \boldsymbol{\mu}}(\boldsymbol{\mu};\, \boldsymbol{\Phi},\, \boldsymbol{\Psi})$[1] are derived by considering the total variation of the reduced-order model in (4.3) with respect to perturbations in $\boldsymbol{\mu}$. In the general case where $\boldsymbol{\Psi}$ is state- and parameter-dependent, the reduced sensitivities are defined as the solution of the linear equations

$$\left[\sum_{j=1}^{N_u} \boldsymbol{r}_j \frac{\partial\left(\boldsymbol{\Psi}^T \boldsymbol{e}_j\right)}{\partial \boldsymbol{u}}\boldsymbol{\Phi} + \boldsymbol{\Psi}^T \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}\boldsymbol{\Phi}\right]\frac{\partial \boldsymbol{u}_r}{\partial \boldsymbol{\mu}} = -\left[\sum_{j=1}^{N_u} \boldsymbol{r}_j \frac{\partial\left(\boldsymbol{\Psi}^T \boldsymbol{e}_j\right)}{\partial \boldsymbol{\mu}} + \boldsymbol{\Psi}^T \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{\mu}}\right] \tag{4.19}$$

where all terms are evaluated at the reconstructed *primal solution*, $\boldsymbol{\Phi}\boldsymbol{u}_r(\boldsymbol{\mu};\, \boldsymbol{\Phi},\, \boldsymbol{\Psi})$ of the reduced-order model (4.3). In the special case where the primal reduced-order model is *exact*, i.e., $\boldsymbol{r}(\boldsymbol{\Phi}\boldsymbol{u}_r(\boldsymbol{\mu};\, \boldsymbol{\Phi},\, \boldsymbol{\Psi}),\, \boldsymbol{\mu}) = 0$, *or* the test basis is *constant*, the expression in (4.19) reduces to

$$\left[\boldsymbol{\Psi}^T \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}\boldsymbol{\Phi}\right]\frac{\partial \boldsymbol{u}_r}{\partial \boldsymbol{\mu}} = -\boldsymbol{\Psi}^T \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{\mu}}. \tag{4.20}$$

A Galerkin projection employs a constant test basis $\boldsymbol{\Psi} = \boldsymbol{\Phi}$ and the equation for the reduced sensitivity in (4.19) or (4.20) reduces to

$$\left[\boldsymbol{\Phi}^T \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}\boldsymbol{\Phi}\right]\frac{\partial \boldsymbol{u}_r}{\partial \boldsymbol{\mu}} = -\boldsymbol{\Phi}^T \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{\mu}}. \tag{4.21}$$

A LSPG projection employs a non-constant test basis $\boldsymbol{\Psi} = \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}\boldsymbol{\Phi}$ and the derivatives of the test basis in (4.19) cannot be ignored in the general case where the primal solution is not exact. In this case, the reduced sensitivities are the solution of the following equation

$$\left[\sum_{j=1}^{N_u} \boldsymbol{r}_j \boldsymbol{\Phi}^T \frac{\partial^2 \boldsymbol{r}_j}{\partial \boldsymbol{u}\partial \boldsymbol{u}}\boldsymbol{\Phi} + \boldsymbol{\Phi}^T \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}^T \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}\boldsymbol{\Phi}\right]\frac{\partial \boldsymbol{u}_r}{\partial \boldsymbol{\mu}} = -\left[\sum_{j=1}^{N_u} \boldsymbol{r}_j \boldsymbol{\Phi}^T \frac{\partial^2 \boldsymbol{r}_j}{\partial \boldsymbol{u}\partial \boldsymbol{\mu}} + \boldsymbol{\Phi}^T \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}^T \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{\mu}}\right]. \tag{4.22}$$

[1] This notation is simplified from $\nabla F_r(\boldsymbol{\mu};\, \boldsymbol{\Phi},\, \boldsymbol{\Psi})$ and $\frac{\partial \boldsymbol{u}_r}{\partial \boldsymbol{\mu}}(\boldsymbol{\mu};\, \boldsymbol{\Phi},\, \boldsymbol{\Psi})$ to $\nabla F_r(\boldsymbol{\mu})$ and $\frac{\partial \boldsymbol{u}_r}{\partial \boldsymbol{\mu}}(\boldsymbol{\mu})$, respectively, when there is no risk of confusion.

Despite the many advantages of minimum-residual reduced-order models (Proposition 4.1), a major disadvantage is that they complicate sensitivity analysis due to the required partial derivatives of the test basis in (4.19). This is particularly true in the case of LSPG where the first-order sensitivities, $\frac{\partial \boldsymbol{u}_r}{\partial \boldsymbol{\mu}}$, require *second-order* information about the partial differential equation—information that is rarely available is large-scale PDE implementations. For this reason, the theory of minimum-residual *primal* reduced-order models is extended to the sensitivity equations in an attempt to avoid terms involving derivatives of the test basis, while generating reduced sensitivities that optimally reconstruct the HDM sensitivities. The main drawback of this approach is the computed sensitivities will not be *consistent* with the reduced-order model to which they correspond since they will not coincide with the solution of (4.19).

Before embarking on the discussion of minimum-residual sensitivity analysis, recall the definition of the *sensitivity* residual

$$\boldsymbol{r}^{\partial}(\boldsymbol{u},\, \boldsymbol{w},\, \boldsymbol{\mu}) = \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{\mu}}(\boldsymbol{u},\, \boldsymbol{\mu}) + \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}(\boldsymbol{u},\, \boldsymbol{\mu})\boldsymbol{w}, \tag{4.23}$$

and the generalization of the gradient of the quantity of interest

$$\boldsymbol{g}^{\partial}(\boldsymbol{u},\, \boldsymbol{w},\, \boldsymbol{\mu}) = \frac{\partial f}{\partial \boldsymbol{\mu}}(\boldsymbol{u},\, \boldsymbol{\mu}) + \frac{\partial f}{\partial \boldsymbol{u}}(\boldsymbol{u},\, \boldsymbol{\mu})\boldsymbol{w} \tag{4.24}$$

introduced in Section 2.3.3. With this notation, the gradient of the reduced quantity of interest takes the form

$$\nabla F_r(\boldsymbol{\mu}) = \boldsymbol{g}^{\partial}\left(\boldsymbol{\Phi}\boldsymbol{u}_r(\boldsymbol{\mu}),\, \boldsymbol{\Phi}\frac{\partial \boldsymbol{u}_r}{\partial \boldsymbol{\mu}}(\boldsymbol{\mu}),\, \boldsymbol{\mu}\right). \tag{4.25}$$

Instead of considering the reconstructed reduced sensitivity, $\boldsymbol{\Phi}\frac{\partial \boldsymbol{u}_r}{\partial \boldsymbol{\mu}}$, as an approximation for the HDM sensitivity $\frac{\partial \boldsymbol{u}}{\partial \boldsymbol{\mu}}$, an approximation of the form

$$\frac{\partial \boldsymbol{u}}{\partial \boldsymbol{\mu}} = \boldsymbol{\Phi}^{\partial}\widehat{\frac{\partial \boldsymbol{u}_r}{\partial \boldsymbol{\mu}}} \tag{4.26}$$

will be considered where $\boldsymbol{\Phi}^{\partial} \in \mathbb{R}^{N_{\boldsymbol{u}} \times k_{\boldsymbol{u}}}$ is a reduced-order basis (linearly independent columns) for the sensitivities and $\widehat{\frac{\partial \boldsymbol{u}_r}{\partial \boldsymbol{\mu}}} \in \mathbb{R}^{k_{\boldsymbol{u}} \times N_{\boldsymbol{\mu}}}$ are the reduced coordinates. The reduced coordinates will be defined as the argument that minimizes the sensitivity residual

$$\widehat{\frac{\partial \boldsymbol{u}_r}{\partial \boldsymbol{\mu}}}(\boldsymbol{\mu};\, \boldsymbol{\Phi}^{\partial},\, \boldsymbol{\Theta}^{\partial},\, \boldsymbol{u}) = \underset{\boldsymbol{w}_r \in \mathbb{R}^{k_{\boldsymbol{u}} \times N_{\boldsymbol{\mu}}}}{\arg\min} \frac{1}{2}\left\|\boldsymbol{r}^{\partial}(\boldsymbol{u},\, \boldsymbol{\Phi}^{\partial}\boldsymbol{w}_r,\, \boldsymbol{\mu})\right\|^2_{\boldsymbol{\Theta}^{\partial}}, \tag{4.27}$$

where $\boldsymbol{u} \in \mathbb{R}^{N_{\boldsymbol{u}}}$ is any linearization point, usually the reconstructed primal solution, i.e., $\boldsymbol{u} = \boldsymbol{\Phi}\boldsymbol{u}_r(\boldsymbol{\mu};\, \boldsymbol{\Phi},\, \boldsymbol{\Psi})$ and $\boldsymbol{\Theta}^{\partial} \succ 0$ is the metric defining the norm. The first-order optimality condition of

the *linear* least-squares problem in (4.27) leads to the normal equations

$$\left(\frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}(\boldsymbol{u},\,\boldsymbol{\mu})\boldsymbol{\Phi}^{\partial}\right)^{T}\boldsymbol{\Theta}^{\partial}\left(\frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}(\boldsymbol{u},\,\boldsymbol{\mu})\boldsymbol{\Phi}^{\partial}\right)\widehat{\frac{\partial \boldsymbol{u}_r}{\partial \boldsymbol{\mu}}} = -\left(\frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}(\boldsymbol{u},\,\boldsymbol{\mu})\boldsymbol{\Phi}^{\partial}\right)^{T}\boldsymbol{\Theta}^{\partial}\frac{\partial \boldsymbol{r}}{\partial \boldsymbol{\mu}}(\boldsymbol{u},\,\boldsymbol{\mu}). \tag{4.28}$$

The following proposition parallels Proposition 4.1 for the minimum-residual sensitivity approximation and provides conditions that result in the reduced sensitivities, $\boldsymbol{\Phi}^{\partial}\widehat{\frac{\partial \boldsymbol{u}_r}{\partial \boldsymbol{\mu}}}$, exactly reconstructing the HDM sensitivities, $\frac{\partial \boldsymbol{u}}{\partial \boldsymbol{\mu}}$.

Proposition 4.2. *Let $(\boldsymbol{\Phi}^{\partial}, \boldsymbol{\Theta}^{\partial})$ define a minimum-residual sensitivity reduced-order model. Then, the following properties hold for any $\boldsymbol{\mu} \in \mathbb{R}^{N_{\boldsymbol{\mu}}}$:*

- *(Optimality) For any $\boldsymbol{u} \in \mathbb{R}^{N_{\boldsymbol{u}}}$ and $\boldsymbol{w} \in col(\boldsymbol{\Phi}^{\partial})$*

$$\left\| \boldsymbol{r}_k^{\partial}\left(\boldsymbol{u},\, \boldsymbol{\Phi}^{\partial}\widehat{\frac{\partial \boldsymbol{u}_r}{\partial \boldsymbol{\mu}}}(\boldsymbol{\mu};\, \boldsymbol{\Phi}^{\partial},\, \boldsymbol{\Theta}^{\partial},\, \boldsymbol{u})\boldsymbol{e}_k,\, \boldsymbol{\mu}\right)\right\|_{\boldsymbol{\Theta}^{\partial}} \leq \left\|\boldsymbol{r}_k^{\partial}\left(\boldsymbol{u},\, \boldsymbol{w},\, \boldsymbol{\mu}\right)\right\|_{\boldsymbol{\Theta}^{\partial}} \tag{4.29}$$

  for $k = 1, \ldots, N_{\boldsymbol{\mu}}$, where $\boldsymbol{r}_k^{\partial}(\boldsymbol{u},\, \boldsymbol{w}\cdot\boldsymbol{e}_k,\, \boldsymbol{\mu}) \coloneqq \boldsymbol{r}^{\partial}(\boldsymbol{u},\, \boldsymbol{w}\boldsymbol{e}_k^{T},\, \boldsymbol{\mu})\boldsymbol{e}_k$ and $\boldsymbol{e}_k$ is the kth canonical unit vector.

- *(Monotonicity) Let $(\boldsymbol{\Phi}^{\partial'}, \boldsymbol{\Theta}^{\partial'})$ define a minimum-residual sensitivity reduced-order model such that $col(\boldsymbol{\Phi}^{\partial'}) \subseteq col(\boldsymbol{\Phi}^{\partial})$, then*

$$\left\|\boldsymbol{r}_k^{\partial}\left(\boldsymbol{u},\, \boldsymbol{w},\, \boldsymbol{\mu}\right)\right\|_{\boldsymbol{\Theta}^{\partial}} \leq \left\|\boldsymbol{r}_k^{\partial}\left(\boldsymbol{u},\, \boldsymbol{w}',\, \boldsymbol{\mu}\right)\right\|_{\boldsymbol{\Theta}^{\partial}}, \tag{4.30}$$

  where $\boldsymbol{w} = \boldsymbol{\Phi}^{\partial}\widehat{\frac{\partial \boldsymbol{u}_r}{\partial \boldsymbol{\mu}}}(\boldsymbol{\mu};\, \boldsymbol{\Phi}^{\partial}, \boldsymbol{\Theta}^{\partial},\, \boldsymbol{u})\boldsymbol{e}_k$ and $\boldsymbol{w}' = \boldsymbol{\Phi}^{\partial'}\widehat{\frac{\partial \boldsymbol{u}_r}{\partial \boldsymbol{\mu}}}(\boldsymbol{\mu};\, \boldsymbol{\Phi}^{\partial'}, \boldsymbol{\Theta}^{\partial'},\, \boldsymbol{u})\boldsymbol{e}_k$, for any $\boldsymbol{u} \in \mathbb{R}^{N_{\boldsymbol{u}}}$.

- *(Interpolation) If $\frac{\partial \boldsymbol{u}}{\partial \boldsymbol{\mu}_k}(\boldsymbol{\mu}) \in col(\boldsymbol{\Phi}^{\partial})$ for $k \in \{1,\, \ldots,\, N_{\boldsymbol{\mu}}\}$, then*

$$\begin{aligned} \boldsymbol{r}_k^{\partial}\left(\boldsymbol{u}(\boldsymbol{\mu}),\, \boldsymbol{\Phi}^{\partial}\widehat{\frac{\partial \boldsymbol{u}_r}{\partial \boldsymbol{\mu}}}(\boldsymbol{\mu};\, \boldsymbol{\Phi}^{\partial}, \boldsymbol{\Theta}^{\partial},\, \boldsymbol{u}(\boldsymbol{\mu}))\boldsymbol{e}_k,\, \boldsymbol{\mu}\right) &= 0 \\ \frac{\partial \boldsymbol{u}}{\partial \boldsymbol{\mu}_k}(\boldsymbol{\mu}) = \boldsymbol{\Phi}^{\partial}\widehat{\frac{\partial \boldsymbol{u}_r}{\partial \boldsymbol{\mu}}}(\boldsymbol{\mu};\, \boldsymbol{\Phi}^{\partial},\, \boldsymbol{\Theta}^{\partial},\, \boldsymbol{u}(\boldsymbol{\mu}))\boldsymbol{e}_k. & \end{aligned} \tag{4.31}$$

Proof. Optimality follows trivially from the fact that $\boldsymbol{\Phi}^{\partial}\widehat{\frac{\partial \boldsymbol{u}_r}{\partial \boldsymbol{\mu}}}(\boldsymbol{\mu};\, \boldsymbol{\Phi}^{\partial},\, \boldsymbol{\Theta}^{\partial},\, \boldsymbol{u})$ is the (unique) minima of the optimization problem in (4.27). Monotonicity follows directly from the optimality property since $\boldsymbol{w}' \in \mathrm{col}(\boldsymbol{\Phi}^{\partial'}) \subseteq \mathrm{col}(\boldsymbol{\Phi}^{\partial})$. Finally, if the exact solution of $\boldsymbol{r}_k^{\partial}(\boldsymbol{u}(\boldsymbol{\mu}), \cdot\,, \boldsymbol{\mu}) = 0$ is contained in the columnspace of $\boldsymbol{\Phi}^{\partial}$, i.e., $\frac{\partial \boldsymbol{u}}{\partial \boldsymbol{\mu}_k}(\boldsymbol{\mu}) \in \mathrm{col}(\boldsymbol{\Phi}^{\partial})$, the optimality property implies

$$\left\|\boldsymbol{r}_k^{\partial}\left(\boldsymbol{u}(\boldsymbol{\mu}),\, \boldsymbol{\Phi}^{\partial}\widehat{\frac{\partial \boldsymbol{u}_r}{\partial \boldsymbol{\mu}}}(\boldsymbol{\mu};\, \boldsymbol{\Phi}^{\partial},\, \boldsymbol{\Theta}^{\partial},\, \boldsymbol{u}(\boldsymbol{\mu}))\boldsymbol{e}_k,\, \boldsymbol{\mu}\right)\right\|_{\boldsymbol{\Theta}^{\partial}} \leq \left\|\boldsymbol{r}_k^{\partial}\left(\boldsymbol{u}(\boldsymbol{\mu}),\, \frac{\partial \boldsymbol{u}}{\partial \boldsymbol{\mu}_k}(\boldsymbol{\mu}),\, \boldsymbol{\mu}\right)\right\|_{\boldsymbol{\Theta}^{\partial}} = 0. \tag{4.32}$$

The interpolation property in (4.31) follows from this and the fact that for a given $\boldsymbol{u} \in \mathbb{R}^{N_{\boldsymbol{u}}}$ and $\boldsymbol{\mu} \in \mathbb{R}^{N_{\boldsymbol{\mu}}}$, the solution of $\boldsymbol{r}_k^{\partial}(\boldsymbol{u}, \cdot, \boldsymbol{\mu}) = 0$ is unique (due to invertibility of the Jacobian and linear independence of the columns of $\boldsymbol{\Phi}^{\partial}$). □

The proposed minimum-residual sensitivity approximation is used to reconstruct an approximation of the gradient of the quantity of interest as

$$\nabla F(\boldsymbol{\mu}) \approx \widehat{\nabla F_r}(\boldsymbol{\mu};\, \boldsymbol{\Phi},\, \boldsymbol{\Psi},\, \boldsymbol{\Phi}^{\partial},\, \boldsymbol{\Theta}^{\partial}) := \boldsymbol{g}^{\partial}\left(\boldsymbol{u},\, \boldsymbol{\Phi}^{\partial}\widehat{\frac{\partial \boldsymbol{u}_r}{\partial \boldsymbol{\mu}}}(\boldsymbol{\mu};\, \boldsymbol{\Phi}^{\partial},\, \boldsymbol{\Theta}^{\partial},\, \boldsymbol{u}),\, \boldsymbol{\mu}\right), \tag{4.33}$$

where $\boldsymbol{u} = \boldsymbol{\Phi}\boldsymbol{u}_r(\boldsymbol{\mu};\, \boldsymbol{\Phi},\, \boldsymbol{\Psi})$ is the reconstructed primal solution[2]. From Appendix B, the approximation of the gradient of the reduced QoI is equipped with a residual-based error bound (Lemma B.4) that takes the form

$$\begin{aligned}\left\|\nabla F(\boldsymbol{\mu}) - \widehat{\nabla F_r}(\boldsymbol{\mu};\, \boldsymbol{\Phi},\, \boldsymbol{\Psi},\, \boldsymbol{\Phi}^{\partial},\, \boldsymbol{\Theta}^{\partial})\right\| &\leq \kappa \left\|\boldsymbol{r}(\boldsymbol{u},\, \boldsymbol{\mu})\right\| + \tau \left\|\boldsymbol{r}^{\partial}\left(\boldsymbol{u},\, \boldsymbol{\Phi}^{\partial}\widehat{\frac{\partial \boldsymbol{u}_r}{\partial \boldsymbol{\mu}}}\left(\boldsymbol{\mu};\, \boldsymbol{\Phi}^{\partial},\, \boldsymbol{\Theta}^{\partial},\, \boldsymbol{u}\right),\, \boldsymbol{\mu}\right)\right\| \\ &\leq \kappa' \left\|\boldsymbol{r}(\boldsymbol{u},\, \boldsymbol{\mu})\right\|_{\boldsymbol{\Theta}} + \tau' \left\|\boldsymbol{r}^{\partial}\left(\boldsymbol{u},\, \boldsymbol{\Phi}^{\partial}\widehat{\frac{\partial \boldsymbol{u}_r}{\partial \boldsymbol{\mu}}}\left(\boldsymbol{\mu};\, \boldsymbol{\Phi}^{\partial},\, \boldsymbol{\Theta}^{\partial},\, \boldsymbol{u}\right),\, \boldsymbol{\mu}\right)\right\|_{\boldsymbol{\Theta}^{\partial}}\end{aligned} \tag{4.34}$$

for some constants $\kappa,\, \kappa',\, \tau,\, \tau' > 0$, where $\boldsymbol{u} = \boldsymbol{\Phi}\boldsymbol{u}_r(\boldsymbol{\mu};\, \boldsymbol{\Phi},\, \boldsymbol{\Psi})$ and the second inequality follows from norm equivalence in finite dimensions. The residual-based error bound illuminates one motivation behind the minimum-residual primal and sensitivity formulations: the minimum-residual primal reduced-order model minimizes the first term in (4.34) over the columnspace of $\boldsymbol{\Phi}$ and the minimum-residual sensitivity reduced-order model minimizes the second term over the columnspace of $\boldsymbol{\Phi}^{\partial}$.

To this point, two different approximations of the high-dimensional model sensitivities $\dfrac{\partial \boldsymbol{u}}{\partial \boldsymbol{\mu}}(\boldsymbol{\mu})$ have been introduced: $\boldsymbol{\Phi}\dfrac{\partial \boldsymbol{u}_r}{\partial \boldsymbol{\mu}}$ and $\boldsymbol{\Phi}^{\partial}\widehat{\dfrac{\partial \boldsymbol{u}_r}{\partial \boldsymbol{\mu}}}$. Each leads to a different approximation of the gradient of the quantity of interest: $\nabla F_r(\boldsymbol{\mu})$ and $\widehat{\nabla F_r}(\boldsymbol{\mu})$. Proposition 4.3 states sufficient conditions under which these two approximations are equal. Specifically, it requires the test basis for the primal ROM ($\boldsymbol{\Psi}$), the sensitivity optimality metric ($\boldsymbol{\Theta}^{\partial}$), and sensitivity basis ($\boldsymbol{\Phi}^{\partial}$) be related according to (4.35). Furthermore, these conditions also imply the reduced coordinates $\dfrac{\partial \boldsymbol{u}_r}{\partial \boldsymbol{\mu}}$ and $\widehat{\dfrac{\partial \boldsymbol{u}_r}{\partial \boldsymbol{\mu}}}$ themselves are equal. Proposition 4.3 is significant since it provides conditions under which the easily computed minimum-residual sensitivities (since they do not require second derivatives of $\boldsymbol{r}$) reduce to the desired reduced-order model sensitivities (since they guarantee consistency of the gradients of reduced quantities of interest).

Proposition 4.3. *Consider a primal reduced-order model defined by trial and test bases $\boldsymbol{\Phi}$ and $\boldsymbol{\Psi}$, respectively, and a minimum-residual sensitivity reduced-order model defined by basis $\boldsymbol{\Phi}^{\partial}$ and metric $\boldsymbol{\Theta}^{\partial}$. Suppose that either: (1) the primal solution of the reduced-order model exactly reconstructs the*

[2]The notation $\widehat{\nabla F_r}(\boldsymbol{\mu})$ will be used in place of $\widehat{\nabla F_r}(\boldsymbol{\mu};\, \boldsymbol{\Phi},\, \boldsymbol{\Psi},\, \boldsymbol{\Phi}^{\partial},\, \boldsymbol{\Theta}^{\partial})$ when there is no risk of confusion.

HDM solution, i.e.,

$$\boldsymbol{u}(\boldsymbol{\mu}) = \boldsymbol{\Phi}\boldsymbol{u}_r(\boldsymbol{\mu};\, \boldsymbol{\Phi},\, \boldsymbol{\Psi})$$

or *(2) the test basis* $\boldsymbol{\Psi}$ *is* constant. *Then, for any* $\boldsymbol{u} \in \mathbb{R}^{N_{\boldsymbol{u}}}$, *the relationships*

$$\begin{aligned} \boldsymbol{\Phi}^{\partial} &= \boldsymbol{\Phi} \\ \boldsymbol{\Psi}(\boldsymbol{u},\, \boldsymbol{\mu}) &= \boldsymbol{\Theta}^{\partial} \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}(\boldsymbol{u},\, \boldsymbol{\mu}) \boldsymbol{\Phi}^{\partial} \end{aligned} \tag{4.35}$$

guarantee the sensitivity of the primal reduced-order model $(\boldsymbol{\Phi},\, \boldsymbol{\Psi})$ *coincides with the solution of the minimum-residual sensitivity reduced-order model* $(\boldsymbol{\Phi}^{\partial},\, \boldsymbol{\Theta}^{\partial})$ *and the corresponding gradient approximations match*

$$\begin{aligned} \frac{\partial \boldsymbol{u}_r}{\partial \boldsymbol{\mu}}(\boldsymbol{\mu};\, \boldsymbol{\Phi},\, \boldsymbol{\Psi}) &= \widehat{\frac{\partial \boldsymbol{u}_r}{\partial \boldsymbol{\mu}}}(\boldsymbol{\mu};\, \boldsymbol{\Phi}^{\partial},\, \boldsymbol{\Theta}^{\partial},\, \boldsymbol{\Phi}\boldsymbol{u}_r(\boldsymbol{\mu};\, \boldsymbol{\Phi},\, \boldsymbol{\Psi})) \\ \nabla F_r(\boldsymbol{\mu};\, \boldsymbol{\Phi},\, \boldsymbol{\Psi}) &= \widehat{\nabla F_r}(\boldsymbol{\mu};\, \boldsymbol{\Phi},\, \boldsymbol{\Psi},\, \boldsymbol{\Phi}^{\partial},\, \boldsymbol{\Theta}^{\partial}). \end{aligned} \tag{4.36}$$

Proof. Let $\boldsymbol{\Phi}\boldsymbol{u}_r = \boldsymbol{\Phi}\boldsymbol{u}_r(\boldsymbol{\mu};\, \boldsymbol{\Phi},\, \boldsymbol{\Psi})$ denote the reconstructed primal solution of the projection-based reduced-order model. If either the primal solution is exact or the test basis is constant, the general form of the reduced-order model sensitivity equations in (4.19) reduce to the equations in (4.20), where all terms are evaluated at the primal solution, i.e.,

$$\left[\boldsymbol{\Psi}(\boldsymbol{\Phi}\boldsymbol{u}_r,\, \boldsymbol{\mu})^T \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}(\boldsymbol{\Phi}\boldsymbol{u}_r,\, \boldsymbol{\mu})\boldsymbol{\Phi}\right] \frac{\partial \boldsymbol{u}_r}{\partial \boldsymbol{\mu}} = -\boldsymbol{\Psi}(\boldsymbol{\Phi}\boldsymbol{u}_r,\, \boldsymbol{\mu})^T \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{\mu}}(\boldsymbol{\Phi}\boldsymbol{u}_r,\, \boldsymbol{\mu}). \tag{4.37}$$

Conversely, the normal form of the minimum-residual sensitivity reduced-order model in (4.28) reduces to

$$\left[\boldsymbol{\Psi}(\boldsymbol{\Phi}\boldsymbol{u}_r,\, \boldsymbol{\mu})^T \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}(\boldsymbol{\Phi}\boldsymbol{u}_r,\, \boldsymbol{\mu})\boldsymbol{\Phi}\right] \widehat{\frac{\partial \boldsymbol{u}_r}{\partial \boldsymbol{\mu}}} = -\boldsymbol{\Psi}(\boldsymbol{\Phi}\boldsymbol{u}_r,\, \boldsymbol{\mu})^T \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{\mu}}(\boldsymbol{\Phi}\boldsymbol{u}_r,\, \boldsymbol{\mu}), \tag{4.38}$$

when the relationships in (4.35) are enforced. Thus, under conditions (4.35), the governing equations for $\dfrac{\partial \boldsymbol{u}_r}{\partial \boldsymbol{\mu}}$ and $\widehat{\dfrac{\partial \boldsymbol{u}_r}{\partial \boldsymbol{\mu}}}$ are identical and the (unique) solutions must be equal, which establishes the first result in (4.36). The second result in (4.36) follows from the simple relation

$$\begin{aligned} \nabla F_r(\boldsymbol{\mu};\, \boldsymbol{\Phi},\, \boldsymbol{\Psi}) &= \boldsymbol{g}^{\partial}\left(\boldsymbol{\Phi}\boldsymbol{u}_r(\boldsymbol{\mu};\, \boldsymbol{\Phi},\, \boldsymbol{\Psi}),\, \boldsymbol{\Phi}\frac{\partial \boldsymbol{u}_r}{\partial \boldsymbol{\mu}}(\boldsymbol{\mu};\, \boldsymbol{\Phi},\, \boldsymbol{\Psi}),\, \boldsymbol{\mu}\right) \\ &= \boldsymbol{g}^{\partial}\left(\boldsymbol{\Phi}\boldsymbol{u}_r(\boldsymbol{\mu};\, \boldsymbol{\Phi},\, \boldsymbol{\Psi}),\, \boldsymbol{\Phi}^{\partial}\widehat{\frac{\partial \boldsymbol{u}_r}{\partial \boldsymbol{\mu}}}(\boldsymbol{\mu};\, \boldsymbol{\Phi}^{\partial},\, \boldsymbol{\Theta}^{\partial},\, \boldsymbol{\Phi}\boldsymbol{u}_r(\boldsymbol{\mu};\, \boldsymbol{\Phi},\, \boldsymbol{\Psi})),\, \boldsymbol{\mu}\right) \\ &= \widehat{\nabla F_r}(\boldsymbol{\mu};\boldsymbol{\Phi},\, \boldsymbol{\Psi},\, \boldsymbol{\Phi}^{\partial},\, \boldsymbol{\Theta}^{\partial}), \end{aligned} \tag{4.39}$$

where the first and last equality use the definition of $\boldsymbol{g}^{\partial}$ and the second equality uses the identity between the true and minimum-residual reduced sensitivities established in the first part. □

To close this section, the specific form of the minimum-residual sensitivity equations in (4.19) are discussed for the special cases of Galerkin and LSPG projections (4.6). For problems with symmetric positive-definite Jacobians, the Galerkin sensitivity equations in (4.21) exactly match the

minimum-residual sensitivity equations with $\boldsymbol{\Psi} = \boldsymbol{\Phi} = \boldsymbol{\Phi}^{\partial}$ and sensitivity metric

$$\boldsymbol{\Theta}^{\partial} = \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}(\boldsymbol{\Phi}\boldsymbol{u}_r(\boldsymbol{\mu};\, \boldsymbol{\Phi},\, \boldsymbol{\Phi}))^{-T}. \tag{4.40}$$

Additionally, these choices satisfy (4.35), which further supports the claim. Thus, for such problems, the true Galerkin sensitivities possess the minimum residual property (and therefore optimality, monotonicity, and interpolation as defined in Proposition 4.2) and are easy to compute since they do not rely on second derivatives of $\boldsymbol{r}$. For the case of a LSPG projection ($\boldsymbol{\Psi}(\boldsymbol{u},\, \boldsymbol{\mu}) = \dfrac{\partial \boldsymbol{u}}{\partial \boldsymbol{\mu}}(\boldsymbol{u},\, \boldsymbol{\mu})\boldsymbol{\Phi}$), the choices $\boldsymbol{\Phi}^{\partial} = \boldsymbol{\Phi}$ and $\boldsymbol{\Theta}^{\partial} = \boldsymbol{I}$ reduce the minimum-residual sensitivity equations to

$$\boldsymbol{\Phi}^T \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}^T \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}} \boldsymbol{\Phi} \widehat{\frac{\partial \boldsymbol{u}_r}{\partial \boldsymbol{\mu}}} = \boldsymbol{\Phi}^T \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}^T \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{\mu}} \tag{4.41}$$

where all nonlinear terms are evaluated at the reconstructed primal solution $\boldsymbol{\Phi}\boldsymbol{u}_r(\boldsymbol{\mu};\, \boldsymbol{\Phi},\, \boldsymbol{\Psi})$. The above equation is identical to the LSPG sensitivity equations when the primal solution is exact and thus the true and minimum-residual sensitivities agree. This is reaffirmed since the choices satisfy (4.35).

4.1.3 Exact and Minimum-Residual Adjoint Formulation

For optimization problems that involve more optimization variables than constraints, it is desirable to employ the adjoint method to compute gradients of quantities of interest. The derivation of the adjoint equations for the reduced-order model can apply any of the three procedures outline in Section 2.3.4 to the governing equation in (4.3), i.e., $\boldsymbol{r}_r(\boldsymbol{u}_r,\, \boldsymbol{\mu}) = 0$, and reduced quantity of interest in (4.4), i.e., $f_r(\boldsymbol{\Phi}\boldsymbol{u}_r,\, \boldsymbol{\mu})$. For brevity, only the optimization approach is detailed. Consider the auxiliary optimization problem

$$\begin{aligned} &\underset{\boldsymbol{u}_r \in \mathbb{R}^{k_{\boldsymbol{u}}}}{\text{minimize}} && f(\boldsymbol{\Phi}\boldsymbol{u}_r,\, \hat{\boldsymbol{\mu}}) \\ &\text{subject to} && \boldsymbol{\Psi}^T \boldsymbol{r}(\boldsymbol{\Phi}\boldsymbol{u}_r,\, \hat{\boldsymbol{\mu}}) = 0 \end{aligned} \tag{4.42}$$

for a fixed $\hat{\boldsymbol{\mu}}$ and the corresponding Lagrangian

$$\mathcal{L}_r(\boldsymbol{u}_r,\, \boldsymbol{\lambda}_r) = f(\boldsymbol{\Phi}\boldsymbol{u}_r,\, \hat{\boldsymbol{\mu}}) - \boldsymbol{\lambda}_r^T \boldsymbol{\Psi}^T \boldsymbol{r}(\boldsymbol{\Phi}\boldsymbol{u}_r,\, \hat{\boldsymbol{\mu}}). \tag{4.43}$$

By comparing this expression for the Lagrangian with that in (2.98), it is clear that the HDM Lagrange multipliers are reconstructed from the reduced Lagrange multipliers as

$$\boldsymbol{\lambda} = \boldsymbol{\Psi}\boldsymbol{\lambda}_r. \tag{4.44}$$

The stationarity of the Lagrangian with respect to $\boldsymbol{u}_r$ leads to the reduced adjoint equations

$$\left[\sum_{j=1}^{N_{\boldsymbol{u}}} \boldsymbol{r}_j \frac{\partial\left(\boldsymbol{\Psi}^T \boldsymbol{e}_j\right)}{\partial \boldsymbol{u}} \boldsymbol{\Phi} + \boldsymbol{\Psi}^T \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}} \boldsymbol{\Phi}\right]^T \boldsymbol{\lambda}_r = \boldsymbol{\Phi}^T \frac{\partial f}{\partial \boldsymbol{u}}^T \tag{4.45}$$

where all terms are evaluated at the reconstructed primal solution, $\boldsymbol{\Phi}\boldsymbol{u}_r(\boldsymbol{\mu};\, \boldsymbol{\Phi},\, \boldsymbol{\Psi})$.

For any $\boldsymbol{\mu} \in \mathbb{R}^{N_{\boldsymbol{\mu}}}$ and bases $\boldsymbol{\Phi}$, $\boldsymbol{\Psi}$, the solution of the above equation is denoted $\boldsymbol{\lambda}_r(\boldsymbol{\mu};\, \boldsymbol{\Phi},\, \boldsymbol{\Psi})$. The gradient of the quantity of interest is then reconstructed as

$$\nabla F_r(\boldsymbol{\mu}) = \frac{\partial f}{\partial \boldsymbol{\mu}}(\boldsymbol{\Phi}\boldsymbol{u}_r(\boldsymbol{\mu};\, \boldsymbol{\Phi},\, \boldsymbol{\Psi}),\, \boldsymbol{\mu}) - \boldsymbol{\lambda}_r(\boldsymbol{\mu};\, \boldsymbol{\Phi},\, \boldsymbol{\Psi})^T \left[\sum_{j=1}^{N_{\boldsymbol{u}}} \boldsymbol{r}_j \frac{\partial\left(\boldsymbol{\Psi}^T \boldsymbol{e}_j\right)}{\partial \boldsymbol{\mu}} + \boldsymbol{\Psi}^T \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{\mu}}\right]_{(\boldsymbol{\Phi}\boldsymbol{u}_r(\boldsymbol{\mu};\, \boldsymbol{\Phi},\, \boldsymbol{\Psi}),\, \boldsymbol{\mu})}. \tag{4.46}$$

In the special case where the primal reduced-order model is exact, i.e., $\boldsymbol{r}(\boldsymbol{\Phi}\boldsymbol{u}_r(\boldsymbol{\mu};\, \boldsymbol{\Phi},\, \boldsymbol{\Psi}),\, \boldsymbol{\mu}) = 0$, or the test basis is *constant*, the expression in (4.45) reduces to

$$\left[\boldsymbol{\Psi}^T \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}} \boldsymbol{\Phi}\right]^T \boldsymbol{\lambda}_r = \boldsymbol{\Phi}^T \frac{\partial f}{\partial \boldsymbol{u}}^T, \tag{4.47}$$

and the gradient of the QoI becomes

$$\nabla F_r(\boldsymbol{\mu}) = \frac{\partial f}{\partial \boldsymbol{\mu}}(\boldsymbol{\Phi}\boldsymbol{u}_r(\boldsymbol{\mu};\, \boldsymbol{\Phi},\, \boldsymbol{\Psi}),\, \boldsymbol{\mu}) - \boldsymbol{\lambda}_r(\boldsymbol{\mu};\, \boldsymbol{\Phi},\, \boldsymbol{\Psi})^T \boldsymbol{\Psi}^T \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{\mu}}(\boldsymbol{\Phi}\boldsymbol{u}_r(\boldsymbol{\mu};\, \boldsymbol{\Phi},\, \boldsymbol{\Psi}),\, \boldsymbol{\mu}). \tag{4.48}$$

In the special case where the primal reduced-order model employs a Galerkin projection ($\boldsymbol{\Psi} = \boldsymbol{\Phi}$), the test basis is state- and parameter-independent and the adjoint equations in (4.45) or (4.47) become

$$\left[\boldsymbol{\Phi}^T \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}} \boldsymbol{\Phi}\right]^T \boldsymbol{\lambda}_r = \boldsymbol{\Phi}^T \frac{\partial f}{\partial \boldsymbol{u}}^T \tag{4.49}$$

and the gradient of the QoI is

$$\nabla F_r(\boldsymbol{\mu}) = \frac{\partial f}{\partial \boldsymbol{\mu}}(\boldsymbol{\Phi}\boldsymbol{u}_r(\boldsymbol{\mu};\, \boldsymbol{\Phi},\, \boldsymbol{\Phi}),\, \boldsymbol{\mu}) - \boldsymbol{\lambda}_r(\boldsymbol{\mu};\, \boldsymbol{\Phi},\, \boldsymbol{\Phi})^T \boldsymbol{\Phi}^T \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{\mu}}(\boldsymbol{\Phi}\boldsymbol{u}_r(\boldsymbol{\mu};\, \boldsymbol{\Phi},\, \boldsymbol{\Phi}),\, \boldsymbol{\mu}). \tag{4.50}$$

In the special case of a LSPG projection, the adjoint equations in (4.45) become

$$\left[\sum_{j=1}^{N_{\boldsymbol{u}}} \boldsymbol{r}_j \boldsymbol{\Phi}^T \frac{\partial^2 \boldsymbol{r}_j}{\partial \boldsymbol{u} \partial \boldsymbol{u}} \boldsymbol{\Phi} + \boldsymbol{\Phi}^T \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}^T \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}} \boldsymbol{\Phi}\right]^T \boldsymbol{\lambda}_r = \boldsymbol{\Phi}^T \frac{\partial f}{\partial \boldsymbol{u}}^T \tag{4.51}$$

and the QoI gradient is

$$\nabla F_r(\boldsymbol{\mu}) = \frac{\partial f}{\partial \boldsymbol{\mu}}(\boldsymbol{\Phi}\boldsymbol{u}_r(\boldsymbol{\mu};\, \boldsymbol{\Phi},\, \boldsymbol{\Psi}),\, \boldsymbol{\mu}) - \boldsymbol{\lambda}_r(\boldsymbol{\mu};\, \boldsymbol{\Phi},\, \boldsymbol{\Psi})^T \left[\sum_{j=1}^{N_{\boldsymbol{u}}} \boldsymbol{r}_j \boldsymbol{\Phi}^T \frac{\partial^2 \boldsymbol{r}_j}{\partial \boldsymbol{u} \partial \boldsymbol{\mu}} + \boldsymbol{\Phi}^T \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}^T \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{\mu}}\right]_{(\boldsymbol{\Phi}\boldsymbol{u}_r(\boldsymbol{\mu};\, \boldsymbol{\Phi},\, \boldsymbol{\Psi}),\, \boldsymbol{\mu})}. \tag{4.52}$$

For cases where the test basis is non-constant, the adjoint equations and QoI gradient are difficult to compute due to the presence of derivatives of the test basis, which usually involves second derivatives of the discrete PDE. These terms are rarely available in large-scale PDE implementations and expensive to compute when available. Furthermore, the adjoint equations for the reduced-order model are not developed such that the HDM adjoint variable will be optimally reconstructed and, therefore, the gradient of the reduced QoI may not be a good approximation of the gradient of the true QoI. For these reasons, *minimum-residual* adjoint equations will be formulated such that the reconstructed reduced adjoint variable minimizes the HDM adjoint residual in some norm. For generality, approximate the HDM adjoint variable in a reduced-order basis $\boldsymbol{\Phi}^{\boldsymbol{\lambda}} \in \mathbb{R}^{N_{\boldsymbol{u}} \times k_{\boldsymbol{u}}}$, i.e.,

$$\boldsymbol{\lambda} = \boldsymbol{\Phi}^{\boldsymbol{\lambda}} \hat{\boldsymbol{\lambda}}_r. \tag{4.53}$$

In general, the adjoint basis may depend on the primal solution and parameter, i.e., $\boldsymbol{\Phi}^{\boldsymbol{\lambda}}(\boldsymbol{u}, \boldsymbol{\mu})$. The reduced coordinates $\hat{\boldsymbol{\lambda}}_r \in \mathbb{R}^{k_{\boldsymbol{u}}}$ are defined as the solution of the *linear* residual minimization problem (linear least-squares)

$$\text{minimize } \frac{1}{2} \left\| \boldsymbol{r}^{\boldsymbol{\lambda}}(\boldsymbol{\Phi} \boldsymbol{u}_r, \boldsymbol{\Phi}^{\boldsymbol{\lambda}} \hat{\boldsymbol{\lambda}}_r, \boldsymbol{\mu}) \right\|_{\boldsymbol{\Theta}^{\boldsymbol{\lambda}}}^2. \tag{4.54}$$

where $\boldsymbol{u} \in \mathbb{R}^{N_{\boldsymbol{u}}}$ is any linearization point, usually the primal ROM solution. Expanding (4.54) with the definition of $\boldsymbol{r}^{\boldsymbol{\lambda}}$ in (2.101) leads to the following definition of $\hat{\boldsymbol{\lambda}}_r$

$$\hat{\boldsymbol{\lambda}}_r(\boldsymbol{\mu}; \boldsymbol{\Phi}^{\boldsymbol{\lambda}}, \boldsymbol{\Theta}^{\boldsymbol{\lambda}}, \boldsymbol{u}) = \underset{\boldsymbol{z}_r \in \mathbb{R}^{k_{\boldsymbol{u}}}}{\arg\min} \frac{1}{2} \left\| -\frac{\partial f}{\partial \boldsymbol{u}}(\boldsymbol{u}, \boldsymbol{\mu})^T + \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}(\boldsymbol{u}, \boldsymbol{\mu})^T \boldsymbol{\Phi}^{\boldsymbol{\lambda}} \boldsymbol{z}_r \right\|_{\boldsymbol{\Theta}^{\boldsymbol{\lambda}}}^2, \tag{4.55}$$

for any $\boldsymbol{u} \in \mathbb{R}^{N_{\boldsymbol{u}}}$. The definition of $\hat{\boldsymbol{\lambda}}_r$ in (4.55) is equivalent to the solution of the normal equations

$$\left(\frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}^T \boldsymbol{\Phi}^{\boldsymbol{\lambda}} \right)^T \boldsymbol{\Theta}^{\boldsymbol{\lambda}} \left(\frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}^T \boldsymbol{\Phi}^{\boldsymbol{\lambda}} \right) \hat{\boldsymbol{\lambda}}_r = \left(\frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}^T \boldsymbol{\Phi}^{\boldsymbol{\lambda}} \right)^T \boldsymbol{\Theta}^{\boldsymbol{\lambda}} \frac{\partial f}{\partial \boldsymbol{u}}^T \tag{4.56}$$

where the dependence on the linearization point ($\boldsymbol{u}$) and parameter ($\boldsymbol{\mu}$) have been dropped.

As with the primal and sensitivity minimum-residual reduced-order models, the minimum-residual adjoint reduced-order models are guaranteed to be monotonic and interpolatory as defined in Proposition 4.4. This result is relevant since it provides conditions under which the minimum-residual adjoint reduced-order model solution monotonically approaches the HDM adjoint solution and the requirement for these solutions to exactly match.

Proposition 4.4. *Let $(\boldsymbol{\Phi}^{\boldsymbol{\lambda}}, \boldsymbol{\Theta}^{\boldsymbol{\lambda}})$ define a minimum-residual adjoint reduced-order model. Then the following properties hold for any $\boldsymbol{\mu} \in \mathbb{R}^{N_{\boldsymbol{\mu}}}$*

- *(Optimality) For any $\boldsymbol{u} \in \mathbb{R}^{N_{\boldsymbol{u}}}$ and $\boldsymbol{z} \in col(\boldsymbol{\Phi}^{\boldsymbol{\lambda}})$*

$$\left\| \boldsymbol{r}^{\boldsymbol{\lambda}} \left(\boldsymbol{u}, \boldsymbol{\Phi}^{\boldsymbol{\lambda}} \hat{\boldsymbol{\lambda}}_r(\boldsymbol{\mu}; \boldsymbol{\Phi}^{\boldsymbol{\lambda}}, \boldsymbol{\Theta}^{\boldsymbol{\lambda}}, \boldsymbol{u}), \boldsymbol{\mu} \right) \right\|_{\boldsymbol{\Theta}^{\boldsymbol{\lambda}}} \leq \left\| \boldsymbol{r}^{\boldsymbol{\lambda}} (\boldsymbol{u}, \boldsymbol{z}, \boldsymbol{\mu}) \right\|_{\boldsymbol{\Theta}^{\boldsymbol{\lambda}}} \tag{4.57}$$

- *(Monotonicity) Let $(\boldsymbol{\Phi}^{\boldsymbol{\lambda}'}, \boldsymbol{\Theta}^{\boldsymbol{\lambda}'})$ define a minimum-residual adjoint reduced-order model such*

that $\mathrm{col}(\boldsymbol{\Phi}^{\boldsymbol{\lambda}'}) \subseteq \mathrm{col}(\boldsymbol{\Phi}^{\boldsymbol{\lambda}})$*, then*

$$\left\|\boldsymbol{r}^{\boldsymbol{\lambda}}\left(\boldsymbol{u}, \boldsymbol{z}, \boldsymbol{\mu}\right)\right\|_{\boldsymbol{\Theta}^{\boldsymbol{\lambda}}} \leq \left\|\boldsymbol{r}^{\boldsymbol{\lambda}}\left(\boldsymbol{u}, \boldsymbol{z}', \boldsymbol{\mu}\right)\right\|_{\boldsymbol{\Theta}^{\boldsymbol{\lambda}}}, \tag{4.58}$$

where $\boldsymbol{z} = \boldsymbol{\Phi}^{\boldsymbol{\lambda}}\hat{\boldsymbol{\lambda}}_r(\boldsymbol{\mu}; \boldsymbol{\Phi}^{\boldsymbol{\lambda}}, \boldsymbol{\Theta}^{\boldsymbol{\lambda}}, \boldsymbol{u})$ *and* $\boldsymbol{z}' = \boldsymbol{\Phi}^{\boldsymbol{\lambda}'}\hat{\boldsymbol{\lambda}}_r(\boldsymbol{\mu}; \boldsymbol{\Phi}^{\boldsymbol{\lambda}'}, \boldsymbol{\Theta}^{\boldsymbol{\lambda}'}, \boldsymbol{u})$*, for any* $\boldsymbol{u} \in \mathbb{R}^{N_{\boldsymbol{\mu}}}$*.*

- *(Interpolatory) If* $\boldsymbol{\lambda}(\boldsymbol{\mu}) \in \mathrm{col}(\boldsymbol{\Phi}^{\boldsymbol{\lambda}})$*, then*

$$\begin{aligned} \boldsymbol{r}^{\boldsymbol{\lambda}}\left(\boldsymbol{u}(\boldsymbol{\mu}), \boldsymbol{\Phi}^{\boldsymbol{\lambda}}\hat{\boldsymbol{\lambda}}_r(\boldsymbol{\mu}; \boldsymbol{\Phi}^{\boldsymbol{\lambda}}, \boldsymbol{\Theta}^{\boldsymbol{\lambda}}, \boldsymbol{u}(\boldsymbol{\mu})), \boldsymbol{\mu}\right) &= 0 \\ \boldsymbol{\lambda}(\boldsymbol{\mu}) = \boldsymbol{\Phi}^{\boldsymbol{\lambda}}\hat{\boldsymbol{\lambda}}_r(\boldsymbol{\mu}; \boldsymbol{\Phi}^{\boldsymbol{\lambda}}, \boldsymbol{\Theta}^{\boldsymbol{\lambda}}, \boldsymbol{u}(\boldsymbol{\mu})). & \end{aligned} \tag{4.59}$$

Proof. Optimality follows trivially from the fact that $\boldsymbol{\Phi}^{\boldsymbol{\lambda}}\hat{\boldsymbol{\lambda}}(\boldsymbol{\mu}; \boldsymbol{\Phi}^{\boldsymbol{\lambda}}, \boldsymbol{\Theta}^{\boldsymbol{\lambda}}, \boldsymbol{u})$ is the (unique) minima of the optimization problem in (4.54). Monotonicity follows directly from the optimality property since $\boldsymbol{z}' \in \mathrm{col}(\boldsymbol{\Phi}^{\boldsymbol{\lambda}'}) \subseteq \mathrm{col}(\boldsymbol{\Phi}^{\boldsymbol{\lambda}})$. Finally, if the exact solution of $\boldsymbol{r}^{\boldsymbol{\lambda}}(\boldsymbol{u}(\boldsymbol{\mu}), \cdot, \boldsymbol{\mu}) = 0$ is contained in the columnspace of $\boldsymbol{\Phi}^{\boldsymbol{\lambda}}$, i.e., $\boldsymbol{\lambda}(\boldsymbol{\mu}) \in \mathrm{col}(\boldsymbol{\Phi}^{\boldsymbol{\lambda}})$, the optimality property implies

$$\left\|\boldsymbol{r}^{\boldsymbol{\lambda}}\left(\boldsymbol{u}(\boldsymbol{\mu}), \boldsymbol{\Phi}^{\boldsymbol{\lambda}}\hat{\boldsymbol{\lambda}}_r(\boldsymbol{\mu}; \boldsymbol{\Phi}^{\boldsymbol{\lambda}}, \boldsymbol{\Theta}^{\boldsymbol{\lambda}}, \boldsymbol{u}(\boldsymbol{\mu})), \boldsymbol{\mu}\right)\right\|_{\boldsymbol{\Theta}^{\boldsymbol{\lambda}}} \leq \left\|\boldsymbol{r}^{\boldsymbol{\lambda}}\left(\boldsymbol{u}(\boldsymbol{\mu}), \boldsymbol{\lambda}(\boldsymbol{\mu}), \boldsymbol{\mu}\right)\right\|_{\boldsymbol{\Theta}^{\boldsymbol{\lambda}}} = 0. \tag{4.60}$$

The interpolation property in (4.59) follows from this and the fact that for a given $\boldsymbol{u} \in \mathbb{R}^{N_{\boldsymbol{u}}}$ and $\boldsymbol{\mu} \in \mathbb{R}^{N_{\boldsymbol{\mu}}}$, the solution of $\boldsymbol{r}^{\boldsymbol{\lambda}}(\boldsymbol{u}, \cdot, \boldsymbol{\mu}) = 0$ is unique (due to invertibility of the Jacobian and linear independence of the columns of $\boldsymbol{\Phi}^{\boldsymbol{\lambda}}$). □

Since $\hat{\boldsymbol{\lambda}}_r$ is chosen to optimally reconstruct the HDM adjoint variable $\boldsymbol{\lambda}$ in the sense of the $\boldsymbol{\Theta}^{\boldsymbol{\lambda}}$-norm of the adjoint residual, the gradient of the QoI will be computed as

$$\begin{aligned} \nabla F(\boldsymbol{\mu}) \approx \widehat{\nabla F_r}(\boldsymbol{\mu}; \boldsymbol{\Phi}, \boldsymbol{\Psi}, \boldsymbol{\Phi}^{\boldsymbol{\lambda}}, \boldsymbol{\Theta}^{\boldsymbol{\lambda}}) &:= \boldsymbol{g}^{\boldsymbol{\lambda}}\left(\boldsymbol{u}, \boldsymbol{\Phi}^{\boldsymbol{\lambda}}\hat{\boldsymbol{\lambda}}_r(\boldsymbol{\mu}; \boldsymbol{\Phi}^{\boldsymbol{\lambda}}, \boldsymbol{\Theta}^{\boldsymbol{\lambda}}, \boldsymbol{u}), \boldsymbol{\mu}\right) \\ &= \frac{\partial f}{\partial \boldsymbol{\mu}}(\boldsymbol{u}, \boldsymbol{\mu}) - \hat{\boldsymbol{\lambda}}_r(\boldsymbol{\mu}; \boldsymbol{\Phi}^{\boldsymbol{\lambda}}, \boldsymbol{\Theta}^{\boldsymbol{\lambda}}, \boldsymbol{u})^T {\boldsymbol{\Phi}^{\boldsymbol{\lambda}}}^T \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{\mu}}(\boldsymbol{u}, \boldsymbol{\mu}). \end{aligned} \tag{4.61}$$

where $\boldsymbol{u} = \boldsymbol{\Phi}\boldsymbol{u}_r(\boldsymbol{\mu}; \boldsymbol{\Phi}, \boldsymbol{\Psi})$. From Appendix B, the approximation of the gradient of the reduced quantity of interest is equipped with a residual-based error bound (Lemma B.4) that takes the form

$$\begin{aligned} \left\|\nabla F(\boldsymbol{\mu}) - \widehat{\nabla F_r}(\boldsymbol{\mu}; \boldsymbol{\Phi}, \boldsymbol{\Psi}, \boldsymbol{\Phi}^{\boldsymbol{\lambda}}, \boldsymbol{\Theta}^{\boldsymbol{\lambda}})\right\| &\leq \kappa \left\|\boldsymbol{r}(\boldsymbol{u}, \boldsymbol{\mu})\right\| + \tau \left\|\boldsymbol{r}^{\boldsymbol{\lambda}}\left(\boldsymbol{u}, \boldsymbol{\Phi}^{\boldsymbol{\lambda}}\hat{\boldsymbol{\lambda}}_r\left(\boldsymbol{\mu}; \boldsymbol{\Phi}^{\boldsymbol{\lambda}}, \boldsymbol{\Theta}^{\boldsymbol{\lambda}}, \boldsymbol{u}\right), \boldsymbol{\mu}\right)\right\| \\ &\leq \kappa' \left\|\boldsymbol{r}(\boldsymbol{u}, \boldsymbol{\mu})\right\|_{\boldsymbol{\Theta}} + \tau' \left\|\boldsymbol{r}^{\boldsymbol{\lambda}}\left(\boldsymbol{u}, \boldsymbol{\Phi}^{\boldsymbol{\lambda}}\hat{\boldsymbol{\lambda}}_r(\boldsymbol{\mu}; \boldsymbol{\Phi}^{\boldsymbol{\lambda}}, \boldsymbol{\Theta}^{\boldsymbol{\lambda}}, \boldsymbol{u}), \boldsymbol{\mu}\right)\right\|_{\boldsymbol{\Theta}^{\boldsymbol{\lambda}}} \end{aligned} \tag{4.62}$$

for some constants $\kappa, \kappa', \tau, \tau' > 0$, where $\boldsymbol{u} = \boldsymbol{\Phi}\boldsymbol{u}_r(\boldsymbol{\mu}; \boldsymbol{\Phi}, \boldsymbol{\Psi})$ and the second inequality follows from norm equivalence in finite dimensions. The residual-based error bound illuminates one motivation behind the minimum-residual primal and adjoint formulations: the minimum-residual primal reduced-order model minimizes the first term in (4.62) over the columnspace of $\boldsymbol{\Phi}$ and the minimum-residual adjoint reduced-order model minimizes the second term over the columnspace of $\boldsymbol{\Phi}^{\boldsymbol{\lambda}}$.

In an exact parallel with the previous section, Proposition 4.5 provides conditions under which the two aforementioned approximations of the HDM gradient $\nabla F(\boldsymbol{\mu})$, i.e., the approximation based on the reduced-order model adjoint $\nabla F_r(\boldsymbol{\mu})$ and that based on the minimum-residual adjoint reduced-order model $\widehat{\nabla F_r}(\boldsymbol{\mu})$, exactly match. The main condition is the requirement (4.63) on the relationship between the trial basis ($\boldsymbol{\Phi}$), adjoint basis ($\boldsymbol{\Phi}^{\boldsymbol{\lambda}}$), and adjoint optimality metric ($\boldsymbol{\Theta}^{\boldsymbol{\lambda}}$). These conditions also ensure the reduced coordinates $\boldsymbol{\lambda}_r$ and $\hat{\boldsymbol{\lambda}}_r$ match. These results are relevant as they provide conditions under which the easily computed minimum-residual adjoint solutions (since the computation does not require second-order derivatives of $\boldsymbol{r}$) reduce to the desired reduced-order model sensitivities (that guarantee consistency of the gradients of the reduced quantities of interest).

Proposition 4.5. *Consider a primal reduced-order model defined by trial and test bases $\boldsymbol{\Phi}$ and $\boldsymbol{\Psi}$, respectively, and a minimum-residual adjoint reduced-order model defined by basis $\boldsymbol{\Phi}^{\boldsymbol{\lambda}}$ and metric $\boldsymbol{\Theta}^{\boldsymbol{\lambda}}$. Suppose that either: (1) the primal solution of the reduced-order model exactly reconstructs the HDM solution, i.e.,*

$$\boldsymbol{u}(\boldsymbol{\mu}) = \boldsymbol{\Phi}\boldsymbol{u}_r(\boldsymbol{\mu};\, \boldsymbol{\Phi},\, \boldsymbol{\Psi}).$$

or *(2) the test basis $\boldsymbol{\Psi}$ is* constant. *Then, for any $\boldsymbol{u} \in \mathbb{R}^{N_{\boldsymbol{u}}}$, the relationships*

$$\boldsymbol{\Phi}^{\boldsymbol{\lambda}}(\boldsymbol{u},\, \boldsymbol{\mu}) = \boldsymbol{\Psi}(\boldsymbol{u},\, \boldsymbol{\mu}) = \left[\boldsymbol{\Theta}^{\boldsymbol{\lambda}} \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}(\boldsymbol{u},\, \boldsymbol{\mu})^T\right]^{-1} \boldsymbol{\Phi} \tag{4.63}$$

guarantee the adjoint solution of the primal reduced-order model ($\boldsymbol{\Phi}$, $\boldsymbol{\Psi}$) matches the minimum-residual adjoint reduced-order model ($\boldsymbol{\Phi}^{\boldsymbol{\lambda}}$, $\boldsymbol{\Theta}^{\boldsymbol{\lambda}}$)

$$\begin{aligned} \boldsymbol{\lambda}_r(\boldsymbol{\mu};\, \boldsymbol{\Phi},\, \boldsymbol{\Psi}) &= \hat{\boldsymbol{\lambda}}_r(\boldsymbol{\mu};\, \boldsymbol{\Phi}^{\boldsymbol{\lambda}},\, \boldsymbol{\Theta}^{\boldsymbol{\lambda}},\, \boldsymbol{\Phi}\boldsymbol{u}_r(\boldsymbol{\mu};\, \boldsymbol{\Phi},\, \boldsymbol{\Psi})) \\ \nabla F_r(\boldsymbol{\mu};\, \boldsymbol{\Phi},\, \boldsymbol{\Psi}) &= \widehat{\nabla F_r}(\boldsymbol{\mu};\, \boldsymbol{\Phi},\, \boldsymbol{\Psi},\, \boldsymbol{\Phi}^{\boldsymbol{\lambda}},\, \boldsymbol{\Theta}^{\boldsymbol{\lambda}}) \end{aligned} \tag{4.64}$$

Proof. Let $\boldsymbol{\Phi}\boldsymbol{u}_r = \boldsymbol{\Phi}\boldsymbol{u}_r(\boldsymbol{\mu};\, \boldsymbol{\Phi},\, \boldsymbol{\Psi})$ denote the reconstructed primal solution of the projection-based reduced-order model. If either the primal solution is exact or the test basis is constant, the general form of the reduced-order model adjoint equations in (4.45) reduce to the equations in (4.47), where all terms are evaluated at the primal solution, i.e.,

$$\left[\boldsymbol{\Psi}(\boldsymbol{\Phi}\boldsymbol{u}_r,\, \boldsymbol{\mu})^T \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}(\boldsymbol{\Phi}\boldsymbol{u}_r,\, \boldsymbol{\mu})\boldsymbol{\Phi}\right]^T \boldsymbol{\lambda}_r = \boldsymbol{\Phi}^T \frac{\partial f}{\partial \boldsymbol{u}}(\boldsymbol{\Phi}\boldsymbol{u}_r,\, \boldsymbol{\mu})^T. \tag{4.65}$$

Conversely, the normal form of the minimum-residual adjoint reduced-order model in (4.56) reduces to

$$\left[\boldsymbol{\Psi}(\boldsymbol{\Phi}\boldsymbol{u}_r,\, \boldsymbol{\mu})^T \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}(\boldsymbol{\Phi}\boldsymbol{u}_r,\, \boldsymbol{\mu})\boldsymbol{\Phi}\right]^T \hat{\boldsymbol{\lambda}}_r = \boldsymbol{\Phi}^T \frac{\partial f}{\partial \boldsymbol{u}}(\boldsymbol{\Phi}\boldsymbol{u}_r,\, \boldsymbol{\mu})^T \tag{4.66}$$

when the relationships in (4.63) are enforced. Thus, under the aforementioned conditions, the governing equations for $\boldsymbol{\lambda}_r$ and $\hat{\boldsymbol{\lambda}}_r$ are identical and the (unique) solutions must be equal, which

establishes the first result in (4.64). The second result in (4.64) follows from the simple relation

$$\begin{aligned}\nabla F_r(\boldsymbol{\mu};\,\boldsymbol{\Phi},\,\boldsymbol{\Psi}) &= \boldsymbol{g}^{\boldsymbol{\lambda}}\left(\boldsymbol{\Phi}\boldsymbol{u}_r(\boldsymbol{\mu};\,\boldsymbol{\Phi},\,\boldsymbol{\Psi}),\,\boldsymbol{\Psi}\boldsymbol{\lambda}_r(\boldsymbol{\mu};\,\boldsymbol{\Phi},\,\boldsymbol{\Psi}),\,\boldsymbol{\mu}\right)\\ &= \boldsymbol{g}^{\boldsymbol{\lambda}}\left(\boldsymbol{\Phi}\boldsymbol{u}_r(\boldsymbol{\mu};\,\boldsymbol{\Phi},\,\boldsymbol{\Psi}),\,\boldsymbol{\Phi}^{\boldsymbol{\lambda}}\hat{\boldsymbol{\lambda}}_r(\boldsymbol{\mu};\,\boldsymbol{\Phi}^{\boldsymbol{\lambda}},\,\boldsymbol{\Theta}^{\boldsymbol{\lambda}},\,\boldsymbol{\Phi}\boldsymbol{u}_r(\boldsymbol{\mu};\,\boldsymbol{\Phi},\,\boldsymbol{\Psi})),\,\boldsymbol{\mu}\right)\\ &= \widehat{\nabla F_r}(\boldsymbol{\mu};\boldsymbol{\Phi},\,\boldsymbol{\Psi},\,\boldsymbol{\Phi}^{\boldsymbol{\lambda}},\,\boldsymbol{\Theta}^{\boldsymbol{\lambda}}),\end{aligned} \tag{4.67}$$

where the first and last equality use the definition of $\boldsymbol{g}^{\boldsymbol{\lambda}}$ and the second equality uses the identity between the true and minimum-residual reduced adjoints established in the first part. □

The section closes with a discussion of the minimum-residual adjoint equations for the special cases of Galerkin and LSPG projections (4.6). For problems with symmetric positive-definite Jacobians, the Galerkin adjoint equations in (4.49) exactly match the minimum-residual adjoint equations in (4.55) with $\boldsymbol{\Psi} = \boldsymbol{\Phi}^{\boldsymbol{\lambda}} = \boldsymbol{\Phi}$ and adjoint metric

$$\boldsymbol{\Theta}^{\boldsymbol{\lambda}} = \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}(\boldsymbol{\Phi}\boldsymbol{u}_r(\boldsymbol{\mu};\,\boldsymbol{\Phi},\,\boldsymbol{\Phi}))^{-T}. \tag{4.68}$$

Additionally, these choices satisfy (4.63), which further supports the claim. Thus, for such problems, the true Galerkin adjoints possess the minimum residual property and are easy to compute since they do not rely on second derivatives of $\boldsymbol{r}$. For the case of a LSPG projection ($\boldsymbol{\Psi}(\boldsymbol{u},\,\boldsymbol{\mu}) = \dfrac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}(\boldsymbol{u},\,\boldsymbol{\mu})\boldsymbol{\Phi}$), the choices $\boldsymbol{\Phi}^{\boldsymbol{\lambda}} = \boldsymbol{\Psi}$ and

$$\boldsymbol{\Theta}^{\boldsymbol{\lambda}} = \left[\frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}^T\frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}\right]^{-1}_{(\boldsymbol{\Phi}\boldsymbol{u}_r(\boldsymbol{\mu};\,\boldsymbol{\Phi},\,\boldsymbol{\Psi}),\,\boldsymbol{\mu})} \tag{4.69}$$

lead to the minimum-residual adjoint equations

$$\boldsymbol{\Phi}^T\frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}^T\frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}\boldsymbol{\Phi}\boldsymbol{u} = \boldsymbol{\Phi}^T\frac{\partial f}{\partial \boldsymbol{u}} \tag{4.70}$$

where all nonlinear terms are evaluated at $\boldsymbol{\Phi}\boldsymbol{u}_r(\boldsymbol{\mu};\,\boldsymbol{\Phi},\,\boldsymbol{\Psi})$. The above equation is identical to the LSPG adjoint equations when the primal solution is exact and thus the true and minimum-residual adjoints agree. This is reaffirmed since the above choices satisfy (4.63).

4.2 Global Hyperreduced Models

Despite the small size of the nonlinear system defining the reduced-order model

$$\boldsymbol{\Psi}^T\boldsymbol{r}(\boldsymbol{\Phi}\boldsymbol{u}_r,\,\boldsymbol{\mu}) = 0,$$

in terms of the number of equations and unknowns ($k_{\boldsymbol{u}}$), it may still be expensive to solve. The major expense emanates from the large-scale operations required to evaluate the reduced residual

and Jacobian (neglecting higher derivatives if $\boldsymbol{\Psi}$ is not constant)

$$\boldsymbol{\Psi}^T \boldsymbol{r}(\boldsymbol{\Phi}\boldsymbol{u}_r, \boldsymbol{\mu}) \qquad \boldsymbol{\Psi}^T \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}(\boldsymbol{\Phi}\boldsymbol{u}_r, \boldsymbol{\mu})\boldsymbol{\Phi}, \tag{4.71}$$

i.e., the reconstruction of the full state from the reduced coordinates $\boldsymbol{u} = \boldsymbol{\Phi}\boldsymbol{u}_r$ and projection of the full residual and Jacobian into the reduced space. Such bottlenecks do not arise in the case where $\boldsymbol{r}$ is *polynomial* in the state and parameter as terms can be precomputed *offline* such that no large-scale operations are required *online*; see Section 4.2.1 for additional details. To overcome such bottlenecks when $\boldsymbol{r}$ is nonlinear in state or parameter, a slew of *hyperreduction*[3] techniques have been proposed that introduce an additional layer of approximation on top of that in (4.3). Among the most popular hyperreduction techniques are (1) polynomialization methods, such as Trajectory Piece-Wise Linear (TPWL) approximation [165], where the governing nonlinear equations are replaced by a weighted sum of the equations linearized about preselected points in parameter space and (2) gappy methods [56, 17, 115, 41, 31, 59] where only a subset of the large-scale equations and degrees of freedom are used in the computation of (4.71)—in the context of PDEs this amounts to only using a subset of the mesh to assemble the reduced residual and Jacobian. This document will only consider gappy methods since they maintain a strong connection to the underlying physics model and enable the reduced residual and Jacobian to be evaluated without incurring operations that scale with the size of the full mesh.

4.2.1 Precomputation for Polynomial Nonlinearities

In the special case where the nonlinearity in the state and parameter is *polynomial*, the contraction of the each monomial term with the reduced basis can be precomputed. As a result, each query to the reduced residual and Jacobian will not involve operations that scale with $N_{\boldsymbol{u}}$. Consider the Taylor expansion of the governing equation of degree m in the state and degree n in the parameter

$$\boldsymbol{r}(\boldsymbol{u}, \boldsymbol{\mu}) = \sum_{j=0}^{m}\sum_{k=0}^{n} \frac{1}{j!k!} \left.\frac{\partial^{j+k}\boldsymbol{r}}{\partial u_{p_1}\cdots\partial u_{p_j}\partial \mu_{q_1}\cdots\partial\mu_{q_k}}\right|_{(\hat{\boldsymbol{u}},\hat{\boldsymbol{\mu}})} (u-\hat{u})_{p_1}\cdots(u-\hat{u})_{p_j}(\mu-\hat{\mu})_{q_1}\cdots(\mu-\hat{\mu})_{q_k} \tag{4.72}$$

where $\hat{\boldsymbol{u}} \in \mathbb{R}^{N_{\boldsymbol{u}}}$ and $\hat{\boldsymbol{\mu}} \in \mathbb{R}^{N_{\boldsymbol{\mu}}}$ are the expansion points. If the governing equation is at most degree m in the state and n in the parameter, this expansion is exact for any expansion points. Otherwise, it is an approximation and its quality will be heavily dependent on $\hat{\boldsymbol{u}}$ and $\hat{\boldsymbol{\mu}}$. The remainder of this section will primarily be concerned with the case where the governing equation is polynomial and the expansion points will be taken as $\hat{\boldsymbol{u}} = \boldsymbol{0}_{N_{\boldsymbol{u}}}$ and $\hat{\boldsymbol{\mu}} = \boldsymbol{0}_{N_{\boldsymbol{\mu}}}$ for simplicity. Define the following $j+k+1$-order tensor for $j = 1,\ldots,m$ and $k = 1,\ldots,n$ as the monomials arising in the above

[3] a term coined in [175]

expansion

$$\boldsymbol{D}^{jk}(\boldsymbol{u},\,\boldsymbol{\mu}) = \left.\frac{\partial^{j+k}\boldsymbol{r}}{\underbrace{\partial\boldsymbol{u}\cdots\partial\boldsymbol{u}}_{j\text{ terms}}\underbrace{\partial\boldsymbol{\mu}\cdots\partial\boldsymbol{\mu}}_{k\text{ terms}}}\right|_{(\boldsymbol{u},\,\boldsymbol{\mu})}. \tag{4.73}$$

When arguments are omitted in the above definition, they are assumed to be zero, i.e., $\boldsymbol{D}^{jk} = \boldsymbol{D}^{jk}(\boldsymbol{0}_{N_{\boldsymbol{u}}}, \boldsymbol{0}_{N_{\boldsymbol{\mu}}})$. With this definition, the expansion of the governing equation becomes

$$\boldsymbol{r}(\boldsymbol{u},\,\boldsymbol{\mu})_i = \sum_{j=0}^{m}\sum_{k=0}^{n}\frac{1}{j!k!}D^{jk}_{ip_1\cdots p_j q_1\cdots q_k}u_{p_1}\cdots u_{p_j}\mu_{q_1}\cdots\mu_{q_k}. \tag{4.74}$$

Introduce the model reduction ansatz for *both the state and parameter*:

$$\boldsymbol{u} = \boldsymbol{\Phi}\boldsymbol{y} \qquad \boldsymbol{\mu} = \boldsymbol{\Upsilon}\boldsymbol{\eta} \tag{4.75}$$

where $\boldsymbol{\Phi} \in \mathbb{R}^{N_{\boldsymbol{u}}\times k_{\boldsymbol{u}}}$ and $\boldsymbol{\Upsilon} \in \mathbb{R}^{N_{\boldsymbol{\mu}}\times k_{\boldsymbol{\mu}}}$ are the reduced bases for the state and parameter spaces, $\boldsymbol{y} \in \mathbb{R}^{k_{\boldsymbol{u}}}$ and $\boldsymbol{\eta} \in \mathbb{R}^{k_{\boldsymbol{\mu}}}$ are the corresponding reduced coordinates, and $k_{\boldsymbol{u}} \ll N_{\boldsymbol{u}}$ and $k_{\boldsymbol{\mu}} \ll N_{\boldsymbol{\mu}}$. Substitution of these ansatz into the polynomial expansion of the governing equations in (4.72) and subsequent projection onto the columnspace of $\boldsymbol{\Phi}$ (a Galerkin projection) leads to

$$[\boldsymbol{r}_r(\boldsymbol{y},\,\boldsymbol{\eta})]_t = \sum_{j=0}^{m}\sum_{k=0}^{n}\frac{1}{j!k!}\left[D_r^{jk}\right]_{tr_1\cdots r_j s_1\cdots s_k}y_{r_1}\cdots y_{r_j}\eta_{s_1}\cdots\eta_{s_k} \tag{4.76}$$

where the reduced monomial terms that have been contracted with the reduced bases are

$$\left[D_r^{jk}\right]_{tr_1\cdots r_j s_1\cdots s_k} = D^{jk}_{ip_1\cdots p_j q_1\cdots q_k}\Phi_{it}\Phi_{p_1 r_1}\cdots\Phi_{p_j r_j}\Upsilon_{q_1 s_1}\cdots\Upsilon_{q_k s_k}. \tag{4.77}$$

From (4.76), the evaluation of the reduced residual $\boldsymbol{r}_r(\boldsymbol{y},\,\boldsymbol{\eta})$ and Jacobian $\dfrac{\partial\boldsymbol{r}_r}{\partial\boldsymbol{y}}(\boldsymbol{y},\,\boldsymbol{\eta})$ are completely independent of the potentially large dimensions $N_{\boldsymbol{u}}$ and $N_{\boldsymbol{\mu}}$ but scale poorly with the reduced dimensions $k_{\boldsymbol{u}}$ and $k_{\boldsymbol{\mu}}$. For example, the evaluation of the reduced residual requires $\mathcal{O}(k_{\boldsymbol{u}}^{m+1}k_{\boldsymbol{\mu}}^{n})$ operations, which makes this feasible for only small polynomial orders m and n and reduced basis sizes $k_{\boldsymbol{u}}$ and $k_{\boldsymbol{\mu}}$. Two common examples of nonlinear partial differential equations that possess polynomial nonlinearities in the state and parameter are: the incompressible Navier-Stokes equations (quadratic in the state) and the geometrically nonlinear structure with a St. Venant-Kirchhoff material law (cubic in the state and linear in the material parameters). While these types of problems arise in a number of important applications, the problems considered in this work will not possess polynomial nonlinearities. Therefore, these methods will not be considered further and attention is turned to the more general *gappy* methods.

4.2.2 Mask and Sample Mesh

The discussion of *gappy* methods in the context of partial differential equations begins with the critical notion of a *mask* and *sample mesh*. Gappy methods are characterized by the distinguishing feature that they only consider a *subset* of the governing equations, that is, only entries $\boldsymbol{r}_i$ are needed, where $i \in \mathcal{M} \subset \{1, \ldots, N_{\boldsymbol{u}}\}$. This subset $\mathcal{M}$ is called the *mask*. Let $\boldsymbol{P} \in \mathbb{R}^{N_{\boldsymbol{u}} \times |\mathcal{M}|}$ be the subset of the columns of the identity matrix that includes $\boldsymbol{e}_i$ only if $i \in \mathcal{M}$. Then, the mask of the governing residual $\boldsymbol{r}$ can be compactly represented as $\boldsymbol{P}^T\boldsymbol{r}$. When $\boldsymbol{r}$ represents a discretized PDE, the evaluation of $\boldsymbol{r}_i$ will require the solution $\boldsymbol{u}_j$ for all $j \in \mathcal{S}_i \subset \{1, \ldots, N_{\boldsymbol{u}}\}$, where the $\mathcal{S}_i$ depends on the discretization scheme and the PDE under consideration. Define the *sample mesh* as the set

$$\mathcal{S} = \bigcup_{i \in \mathcal{M}} \mathcal{S}_i$$

and let $\bar{\boldsymbol{P}} \in \mathbb{R}^{N_{\boldsymbol{u}} \times |\mathcal{S}|}$ be the subset of the columns of the identity matrix that include $\boldsymbol{e}_i$ only if $i \in \mathcal{S}$. Then, the restriction of the state vector $\boldsymbol{u}$ to the sample mesh is compactly written as $\bar{\boldsymbol{P}}^T\boldsymbol{u}$. All of the gappy-based hyperreduction methods considered in this document rely on the computation of the *masked* reduced residual and Jacobian

$$\boldsymbol{P}^T\boldsymbol{r}(\bar{\boldsymbol{P}}\bar{\boldsymbol{P}}^T\boldsymbol{\Phi}\boldsymbol{u}_r, \boldsymbol{\mu}) \qquad \boldsymbol{P}^T\frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}(\bar{\boldsymbol{P}}\bar{\boldsymbol{P}}^T\boldsymbol{\Phi}\boldsymbol{u}_r, \boldsymbol{\mu})\bar{\boldsymbol{P}}\bar{\boldsymbol{P}}^T\boldsymbol{\Phi} \tag{4.78}$$

as they are much less expensive to compute than the terms in (4.71) if $|\mathcal{M}| \ll N_{\boldsymbol{u}}$. While the notation in (4.78) will prove convenient in later sections, it does not necessarily reveal the efficiency of gappy methods. An efficient implementation will compute $\boldsymbol{P}^T\boldsymbol{r}$ and $\boldsymbol{P}^T\frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}\bar{\boldsymbol{P}}$ directly from $\bar{\boldsymbol{P}}^T\boldsymbol{\Phi}\boldsymbol{u}_r$ without reconstructing a $N_{\boldsymbol{u}}$-vector padded with zeros as the notation $\bar{\boldsymbol{P}}\bar{\boldsymbol{P}}^T\boldsymbol{\Phi}\boldsymbol{u}_r$ suggests. Furthermore, the restriction of the reduced-order basis to the mask, $\boldsymbol{P}^T\boldsymbol{\Phi}$, can be precomputed. These implementation details enable the terms in (4.78) to be computed efficiently *online*, i.e., without incurring operations that scale with the full mesh $\mathcal{O}(N_{\boldsymbol{u}})$.

For brevity in the developments to follow, the following notation is introduced for the reconstructed state vector restricted to the sample mesh and reduced residual restricted to the mask

$$\boldsymbol{u}_h = \bar{\boldsymbol{P}}\bar{\boldsymbol{P}}^T\boldsymbol{\Phi}\boldsymbol{u}_r \qquad \boldsymbol{r}_h(\boldsymbol{u}_h, \boldsymbol{\mu}) = \boldsymbol{P}^T\boldsymbol{r}(\boldsymbol{u}_h, \boldsymbol{\mu}). \tag{4.79}$$

Then, the masked Jacobians with respect to the state and parameter are

$$\frac{\partial \boldsymbol{r}_h}{\partial \boldsymbol{u}_h}(\boldsymbol{u}_h, \boldsymbol{\mu}) = \boldsymbol{P}^T\frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}(\boldsymbol{u}_h, \boldsymbol{\mu})\bar{\boldsymbol{P}}\bar{\boldsymbol{P}}^T\boldsymbol{\Phi} \qquad \frac{\partial \boldsymbol{r}_h}{\partial \boldsymbol{\mu}}(\boldsymbol{u}_h, \boldsymbol{\mu}) = \boldsymbol{P}^T\frac{\partial \boldsymbol{r}}{\partial \boldsymbol{\mu}}(\boldsymbol{u}_h, \boldsymbol{\mu}). \tag{4.80}$$

Armed with this notation, the governing equations for gappy-based hyperreduction methods take the general form

$$\boldsymbol{A}(\boldsymbol{u}_h, \boldsymbol{\mu})^T\boldsymbol{r}_h(\boldsymbol{u}_h, \boldsymbol{\mu}) = 0, \tag{4.81}$$

where it is assumed that $\boldsymbol{A}$ can be computed efficiently online or precomputed offline. Regardless

of the hyperreduced approach considered, the quantity of interest is defined according to (4.5), i.e.,

$$F_r(\boldsymbol{\mu}) = f(\boldsymbol{\Phi}\boldsymbol{u}_r(\boldsymbol{\mu}), \boldsymbol{\mu})$$

where $\boldsymbol{u}_r(\boldsymbol{\mu})$ is the solution of the hyperreduced model. In many cases, the entire reconstructed state $\boldsymbol{\Phi}\boldsymbol{u}_r$ is not required to evaluate f, which commonly arises when f corresponds to a surface integral, i.e., only entries corresponding to nodes on the surface are required. In these cases, only a subset of the reconstructed vector $\tilde{\boldsymbol{P}}^T\boldsymbol{\Phi}\boldsymbol{u}_r$ are required and F_r can be computed efficiently. This implementation optimization will not be considered in the remainder as it complicates the exposition. The industry-scale example of the shape optimization of a full aircraft configuration in Section 5.5.4 will leverage this precise optimization since its optimization functionals involve forces integrated along the surface. The next section provides specific examples of gappy-based hyperreduction method that can be written in the general form (4.81).

4.2.3 Examples

The simplest approach to gappy-based hyperreduction is to simply *ignore* information that is not included in the mask. This approach is usually known as *collocation* and the general form of the projection-based reduced-order model is

$$(\boldsymbol{P}^T\boldsymbol{\Psi})^T\boldsymbol{r}_h(\boldsymbol{u}_h, \boldsymbol{\mu}) = 0, \tag{4.82}$$

which fits into the general form in (4.81) with $\boldsymbol{A}(\boldsymbol{u}, \boldsymbol{\mu}) = \boldsymbol{P}^T\boldsymbol{\Psi}(\boldsymbol{u}, \boldsymbol{\mu})$. While this approach is naive in the sense that it makes no attempt to account for missing information, it is simple and robust, provided a sufficiently large sample mesh is used.

In contrast to the naive approach, a number of methods exist that attempt to account for the missing information using ideas setforth in [56], which include (Discrete) Empirical Interpolation Method ((D)EIM) [17, 41] and Gauss-Newton with Approximated Tensors (GNAT) [31]. The (D)EIM approach assumes the residual lies in the low-dimensional subspace defined by the span of a separate basis $\boldsymbol{\Phi}_{\boldsymbol{r}}$[4], i.e.,

$$\boldsymbol{r}(\boldsymbol{\Phi}\boldsymbol{u}_r, \boldsymbol{\mu}) = \boldsymbol{\Phi}_{\boldsymbol{r}}\boldsymbol{h}(\boldsymbol{u}_r, \boldsymbol{\mu}), \tag{4.83}$$

where $\boldsymbol{h} : \mathbb{R}^{k_{\boldsymbol{u}}} \times \mathbb{R}^{N_{\boldsymbol{\mu}}} \to \mathbb{R}^{k_{\boldsymbol{u}}}$ are the reduced coordinates of the residual in the basis $\boldsymbol{\Phi}$. The reduced coordinates are defined such that the representation in (4.83) exactly matches the true residual on the mask

$$\boldsymbol{P}^T\boldsymbol{\Phi}_{\boldsymbol{r}}\boldsymbol{h}(\boldsymbol{\Phi}\boldsymbol{u}_r, \boldsymbol{\mu}) = \boldsymbol{P}^T\boldsymbol{r}(\bar{\boldsymbol{P}}\bar{\boldsymbol{P}}^T\boldsymbol{\Phi}\boldsymbol{u}_r, \boldsymbol{\mu}), \tag{4.84}$$

which leads to the following expression for the residual reduced coordinates $\boldsymbol{h}$

$$\boldsymbol{h}(\boldsymbol{\Phi}\boldsymbol{u}_r, \boldsymbol{\mu}) = (\boldsymbol{P}^T\boldsymbol{\Phi}_{\boldsymbol{r}})^{-1}\boldsymbol{r}_h(\boldsymbol{u}_r, \boldsymbol{\mu}) \tag{4.85}$$

[4]Usually the residual is separated into its linear and nonlinear components and (D)EIM is only applied to the nonlinear portion.

where the definition of $\boldsymbol{r}_h$ in (4.80) was used. Combining (4.83) and (4.85) into the form of the projection-based reduced-order model in (4.3), the (D)EIM governing equations are

$$\left(\boldsymbol{\Phi}_{\boldsymbol{r}}^T\boldsymbol{\Psi}\right)^T\left(\boldsymbol{P}^T\boldsymbol{\Phi}_{\boldsymbol{r}}\right)^{-1}\boldsymbol{r}_h(\boldsymbol{u}_h,\,\boldsymbol{\mu})=0. \tag{4.86}$$

(D)EIM fits into the general form of gappy-based hyperreduced models in (4.81) with $\boldsymbol{A}(\boldsymbol{u},\,\boldsymbol{\mu})=\left(\boldsymbol{\Phi}_{\boldsymbol{r}}^T\boldsymbol{\Psi}\right)^T\left(\boldsymbol{P}^T\boldsymbol{\Phi}_{\boldsymbol{r}}\right)^{-1}$.

The GNAT method is a minor generalization of (D)EIM that also approximates the residual in a separate low-dimensional subspace spanned by the basis $\boldsymbol{\Phi}_{\boldsymbol{r}}\in\mathbb{R}^{N_{\boldsymbol{u}}\times k_{\boldsymbol{r}}}$ $(k_{\boldsymbol{r}}\ll N_{\boldsymbol{u}})$

$$\boldsymbol{r}(\boldsymbol{\Phi}\boldsymbol{u}_r,\,\boldsymbol{\mu})=\boldsymbol{\Phi}_{\boldsymbol{r}}\boldsymbol{h}(\boldsymbol{u}_r,\,\boldsymbol{\mu}), \tag{4.87}$$

and the reduced coordinates $\boldsymbol{h}:\mathbb{R}^{k_{\boldsymbol{u}}}\times\mathbb{R}^{N_{\boldsymbol{\mu}}}\to\mathbb{R}^{k_{\boldsymbol{r}}}$ are defined such that the representation in (4.87) matches, in a least-squares sense, the true residual on the mask

$$\boldsymbol{h}(\boldsymbol{u}_r,\,\boldsymbol{\mu})=\underset{\boldsymbol{z}\in\mathbb{R}^{k_{\boldsymbol{r}}}}{\arg\min}\left\|\boldsymbol{P}^T\boldsymbol{\Phi}_{\boldsymbol{r}}\boldsymbol{z}-\boldsymbol{r}_h(\boldsymbol{u}_h,\,\boldsymbol{\mu})\right\|_2. \tag{4.88}$$

The GNAT governing equations follow from combining (4.87) and (4.88) into the form of the projection-based reduced-order model in (4.3)

$$\left(\boldsymbol{\Phi}_{\boldsymbol{r}}^T\boldsymbol{\Psi}\right)^T\left[\boldsymbol{P}^T\boldsymbol{\Phi}_{\boldsymbol{r}}\right]^{\dagger}\boldsymbol{r}_h(\boldsymbol{u}_h,\,\boldsymbol{\mu})=0. \tag{4.89}$$

The GNAT equations fit into the general form of gappy-based hyperreduction models in (4.81) with $\boldsymbol{A}=\left(\boldsymbol{\Phi}_{\boldsymbol{r}}^T\boldsymbol{\Psi}\right)^T\left[\boldsymbol{P}^T\boldsymbol{\Phi}_{\boldsymbol{r}}\right]^{\dagger}$.

From the above construction, a number of advantages and disadvantages of each approach emerge. As previously mentioned, the (D)EIM and GNAT methods have the desirable property of *attempting* to account for information missing from the mask by approximating the nonlinear residual in a low-dimensional subspace, while collocation simply ignores missing information. However, due to the nonlinearity of the residual $\boldsymbol{r}(\boldsymbol{u},\,\boldsymbol{\mu})$, it cannot be guaranteed (or even expected) to lie in a low-dimensional subspace, even if the state vector $\boldsymbol{u}$ does. In fact, several works [31, 33] have shown that, even to reproduce training data, the residual approximation in (4.87) requires $k_{\boldsymbol{r}}\gg k_{\boldsymbol{u}}$ for the resulting hyperreduced model to be sufficiently accurate. In practice, the (D)EIM and GNAT ansatz in (4.83) and (4.87) cause the corresponding methods to exhibit classical *over-fitting* behavior, i.e., superb accuracy when reproducing training data and low accuracy at predictive points, particularly when applied to real engineering applications [198]. A notable distinction between these methods is that collocation only requires the trial basis ($\boldsymbol{\Phi}$), test basis ($\boldsymbol{\Psi}$), and mask ($\boldsymbol{P}$), while (D)EIM and GNAT require the construction of a separate residual basis ($\boldsymbol{\Phi}_{\boldsymbol{r}}$) that involves substantial additional offline training—usually requiring the collection and compression of residual snapshots [31]. It has been observed [198] that residual snapshots are not necessarily amenable to compression, which

results large number of basis vectors ($k_{\boldsymbol{r}} \gg 1$) and ultimately hurts the performance of the reduced-order model.

Given this discussion, only the collocation hyperreduction method will be considered in the remainder. This is predominantly to avoid the overfitting behavior of the other approaches since the hyperreduced model will be heavily used in *predictive* settings in subsequent chapters. The remainder of this chapter will discuss a formulation of the primal, sensitivity, and adjoint collocation-based hyperreduced model that results in optimal approximations, in the sense of minimizing the residual in some norm on the *mask*.

4.2.4 Minimum-Residual Primal Formulation

The form of the test basis $\boldsymbol{\Psi}$ in the gappy-accelerated projection-based reduced-order models in (4.82), (4.86), (4.89) has remainder arbitrary to this point. Given the desirable properties of minimum-residual reduced-order models detailed in Sections 4.1.1–4.1.3, the test basis is defined in accordance with a parallel concept in the collocation-based hyperreduction setting—the *masked* minimum-residual property. This property (Definition 4.2) requires the solution of the hyperreduced model to minimize the residual *over the mask* in some norm. For the remainder of this document, the solution of the general form of the collocation-based hyperreduced model in (4.81) will be denoted $\boldsymbol{u}_h(\boldsymbol{\mu};\, \boldsymbol{\Phi},\, \boldsymbol{\Psi},\, \boldsymbol{P})$, i.e.,

$$\left(\boldsymbol{P}^T\boldsymbol{\Psi}\right)^T \boldsymbol{r}_h(\boldsymbol{u}_h(\boldsymbol{\mu};\, \boldsymbol{\Phi},\, \boldsymbol{\Psi},\, \boldsymbol{P}),\, \boldsymbol{\mu}) = 0.$$

Furthermore, the mask-reconstructed state vector $\boldsymbol{u}_h$ is identified with its corresponding reduced coordinates $\boldsymbol{u}_r$ from (4.79), i.e., $\boldsymbol{u}_h(\boldsymbol{\mu};\, \boldsymbol{\Phi},\, \boldsymbol{\Psi},\, \boldsymbol{P}) = \bar{\boldsymbol{P}}\bar{\boldsymbol{P}}^T\boldsymbol{\Phi}\boldsymbol{u}_r(\boldsymbol{\mu};\, \boldsymbol{\Phi},\, \boldsymbol{\Psi},\, \boldsymbol{P})$. The sample mesh $\bar{\boldsymbol{P}}$ is not included in the argument list since it is uniquely determined from the mask $\boldsymbol{P}$ and the structure of the governing equation $\boldsymbol{r}$. When there is no risk of confusion regarding the choice of mask, test basis, and trial basis, the arguments will be dropped.

Definition 4.2 (Masked Minimum-Residual Property)**.** *A hyperreduced model of the form (4.81) possesses the masked minimum-residual property if the solution satisfies the first-order optimality conditions of the following* masked *residual minimization problem*

$$\underset{\boldsymbol{u}_r \in \mathbb{R}^{k_{\boldsymbol{u}}}}{\text{minimize}} \quad \frac{1}{2}\left\|\boldsymbol{P}^T\boldsymbol{r}(\bar{\boldsymbol{P}}\bar{\boldsymbol{P}}^T\boldsymbol{\Phi}\boldsymbol{u}_r,\, \boldsymbol{\mu})\right\|_{\boldsymbol{\Theta}}^2 \tag{4.90}$$

for some symmetric positive-definite $\boldsymbol{\Theta} \in \mathbb{R}^{|\mathcal{M}|\times|\mathcal{M}|}$.

Masked minimum-residual hyperreduced models possess a similar monotonicity property as that defined in Proposition 4.1 for minimum-residual projection-based reduced-order models. The interpolation property only holds if the solution of the minimum-residual hyperreduced model is unique (guaranteed if the mask is full, i.e., $\mathcal{M} = \{1, \ldots, N_{\boldsymbol{u}}\}$ by Assumption 2.2). These properties are stated precisely in Proposition 4.6.

Proposition 4.6. *Let* $(\boldsymbol{\Phi},\, \boldsymbol{\Psi},\, \boldsymbol{\Theta},\, \boldsymbol{P})$ *define masked minimum-residual hyperreduced model whose*

solution coincides with the global minimum of (4.90). Then, the following properties hold for any $\boldsymbol{\mu} \in \mathbb{R}^{N_\mu}$*:*

- *(Optimality) For any* $\boldsymbol{u} \in \mathbb{R}^{N_u}$ *such that* $\bar{\boldsymbol{P}}^T\boldsymbol{u} \in col(\bar{\boldsymbol{P}}^T\boldsymbol{\Phi})$*,*

$$\left\|\boldsymbol{P}^T\boldsymbol{r}(\bar{\boldsymbol{P}}\bar{\boldsymbol{P}}^T\boldsymbol{\Phi}\boldsymbol{u}_r(\boldsymbol{\mu};\,\boldsymbol{\Phi},\,\boldsymbol{\Psi},\,\boldsymbol{P}),\,\boldsymbol{\mu})\right\|_{\boldsymbol{\Theta}} \leq \left\|\boldsymbol{P}^T\boldsymbol{r}(\bar{\boldsymbol{P}}\bar{\boldsymbol{P}}^T\boldsymbol{u},\boldsymbol{\mu})\right\|_{\boldsymbol{\Theta}}. \tag{4.91}$$

- *(Monotonicity) Let* $(\boldsymbol{\Phi}',\,\boldsymbol{\Psi}',\,\boldsymbol{P})$ *define a hyperreduced model such that* $col(\bar{\boldsymbol{P}}^T\boldsymbol{\Phi}') \subseteq col(\bar{\boldsymbol{P}}^T\boldsymbol{\Phi})$*, then*

$$\left\|\boldsymbol{P}^T\boldsymbol{r}(\bar{\boldsymbol{P}}\bar{\boldsymbol{P}}^T\boldsymbol{\Phi}\boldsymbol{u}_r(\boldsymbol{\mu};\,\boldsymbol{\Phi},\,\boldsymbol{\Psi},\,\boldsymbol{P}),\,\boldsymbol{\mu})\right\|_{\boldsymbol{\Theta}} \leq \left\|\boldsymbol{P}^T\boldsymbol{r}(\bar{\boldsymbol{P}}\bar{\boldsymbol{P}}^T\boldsymbol{\Phi}'\boldsymbol{u}_r(\boldsymbol{\mu};\,\boldsymbol{\Phi}',\,\boldsymbol{\Psi}',\,\boldsymbol{P}),\,\boldsymbol{\mu})\right\|_{\boldsymbol{\Theta}}. \tag{4.92}$$

- *(Interpolatory) If* $\bar{\boldsymbol{P}}^T\boldsymbol{u}(\boldsymbol{\mu}) \in col(\bar{\boldsymbol{P}}^T\boldsymbol{\Phi})$*, then*

$$\boldsymbol{P}^T\boldsymbol{r}(\bar{\boldsymbol{P}}\bar{\boldsymbol{P}}^T\boldsymbol{\Phi}\boldsymbol{u}_r(\boldsymbol{\mu};\,\boldsymbol{\Phi},\,\boldsymbol{\Psi},\,\boldsymbol{P}),\,\boldsymbol{\mu}) = 0. \tag{4.93}$$

Proof. Optimality follows from the fact that $\boldsymbol{u}_r(\boldsymbol{\mu};\,\boldsymbol{\Phi},\,\boldsymbol{\Psi},\,\boldsymbol{P})$ is the global minima of the optimization problem in (4.90): for $\bar{\boldsymbol{P}}^T\boldsymbol{u} \in \mathrm{col}(\bar{\boldsymbol{P}}^T\boldsymbol{\Phi})$, there exists $\boldsymbol{y} \in \mathbb{R}^{k_u}$ such that $\bar{\boldsymbol{P}}^T\boldsymbol{u} = \bar{\boldsymbol{P}}^T\boldsymbol{\Phi}\boldsymbol{y}$ and since $\boldsymbol{u}_r$ is the global minima of (4.90), we have

$$\left\|\boldsymbol{P}^T\boldsymbol{r}(\bar{\boldsymbol{P}}\bar{\boldsymbol{P}}^T\boldsymbol{\Phi}\boldsymbol{u}_r(\boldsymbol{\mu};\,\boldsymbol{\Phi},\,\boldsymbol{\Psi},\,\boldsymbol{P}),\,\boldsymbol{\mu})\right\|_{\boldsymbol{\Theta}} \leq \left\|\boldsymbol{P}^T\boldsymbol{r}(\bar{\boldsymbol{P}}\bar{\boldsymbol{P}}^T\boldsymbol{\Phi}\boldsymbol{y},\boldsymbol{\mu})\right\|_{\boldsymbol{\Theta}}. \tag{4.94}$$

A simple application of the optimality property to $\bar{\boldsymbol{P}}^T\boldsymbol{\Phi}'\boldsymbol{u}_r(\boldsymbol{\mu};\,\boldsymbol{\Phi}',\,\boldsymbol{\Psi}') \in \mathrm{col}(\bar{\boldsymbol{P}}^T\boldsymbol{\Phi}') \subseteq \mathrm{col}(\bar{\boldsymbol{P}}^T\boldsymbol{\Phi})$ leads to monotonicity. Finally, if the solution of $\boldsymbol{P}^T\boldsymbol{r}(\,\cdot\,,\,\boldsymbol{\mu}) = 0$ is contained in the columnspace of $\bar{\boldsymbol{P}}^T\boldsymbol{\Phi}$, i.e., $\bar{\boldsymbol{P}}^T\boldsymbol{u}(\boldsymbol{\mu}) \in \mathrm{col}(\bar{\boldsymbol{P}}^T\boldsymbol{\Phi})$, the optimality property implies

$$\left\|\boldsymbol{P}^T\boldsymbol{r}(\bar{\boldsymbol{P}}\bar{\boldsymbol{P}}^T\boldsymbol{\Phi}\boldsymbol{u}_r(\boldsymbol{\mu};\,\boldsymbol{\Phi},\,\boldsymbol{\Psi}),\,\boldsymbol{\mu})\right\|_{\boldsymbol{\Theta}} \leq \left\|\boldsymbol{P}^T\boldsymbol{r}(\bar{\boldsymbol{P}}\bar{\boldsymbol{P}}^T\boldsymbol{u}(\boldsymbol{\mu}),\,\boldsymbol{\mu})\right\|_{\boldsymbol{\Theta}} = 0, \tag{4.95}$$

which is precisely the interpolation property in (4.93). □

The first-order optimality condition of (4.90) is

$$\frac{\partial \boldsymbol{r}_h}{\partial \boldsymbol{u}_h}(\boldsymbol{u}_h,\,\boldsymbol{\mu})^T\boldsymbol{\Theta}\boldsymbol{r}_h(\boldsymbol{u}_h,\,\boldsymbol{\mu}) = 0 \tag{4.96}$$

and, therefore, the masked test basis must satisfy

$$\boldsymbol{P}^T\boldsymbol{\Psi}(\boldsymbol{u}_h,\,\boldsymbol{\mu}) = \boldsymbol{\Theta}\frac{\partial \boldsymbol{r}_h}{\partial \boldsymbol{u}_h}(\boldsymbol{u}_h,\,\boldsymbol{\mu}) \tag{4.97}$$

for the collocation-based projection-based hyperreduced model (4.82) to possess the masked minimum-residual property. The special case of a LSPG projection satisfies (4.97) with

$$\boldsymbol{\Theta} = \boldsymbol{I} \qquad \boldsymbol{P}^T\boldsymbol{\Psi}(\boldsymbol{u}_h,\,\boldsymbol{\mu}) = \frac{\partial \boldsymbol{r}_h}{\partial \boldsymbol{u}_h}(\boldsymbol{u}_h,\,\boldsymbol{\mu}). \tag{4.98}$$

where $\boldsymbol{I}$ is the $|\mathcal{M}| \times |\mathcal{M}|$ identity matrix and therefore possess the masked minimum-residual property. The special case of a Galerkin projection ($\boldsymbol{P}^T\boldsymbol{\Psi} = \boldsymbol{P}^T\boldsymbol{\Phi}$) is more cumbersome to interpret as a special case of the optimality conditions in (4.97) in the hyperreduced setting. Unlike the pure projection setting of Section 4.1.1, the reduced Jacobian matrix $\dfrac{\partial \boldsymbol{r}_h}{\partial \boldsymbol{u}_h}$ is not square, in general, and cannot define a valid norm. From (4.97), for a Galerkin projection to possess the *masked* minimum-residual property, $\boldsymbol{\Theta}$ must be selected such that the following constrained linear system of equations (linear in $\boldsymbol{\Theta}$) is satisfied

$$\boldsymbol{\Theta}\frac{\partial \boldsymbol{r}_h}{\partial \boldsymbol{u}_h} = \boldsymbol{P}^T\boldsymbol{\Phi} \qquad \text{subject to} \qquad \boldsymbol{\Theta} \succ 0. \tag{4.99}$$

In general, there is no guarantee this constrained system of equations has a solution. The next section derives the sensitivity equations corresponding to the collocation-based hyperreduced models introduced in this section and develops a minimum-residual variant.

4.2.5 Exact and Minimum-Residual Sensitivity Formulation

The sensitivity analysis for the hyperreduced model parallels the exposition for the reduced-order models in Section 4.1.2. The total derivative of the quantity of interest is expanded as

$$\nabla F_r(\boldsymbol{\mu}) = \frac{\partial f}{\partial \boldsymbol{\mu}}(\boldsymbol{\Phi}\boldsymbol{u}_r(\boldsymbol{\mu};\, \boldsymbol{\Phi},\, \boldsymbol{\Psi},\, \boldsymbol{P}),\, \boldsymbol{\mu}) + \frac{\partial f}{\partial \boldsymbol{u}}(\boldsymbol{\Phi}\boldsymbol{u}_r(\boldsymbol{\mu};\, \boldsymbol{\Phi},\, \boldsymbol{\Psi},\, \boldsymbol{P}),\, \boldsymbol{\mu})\boldsymbol{\Phi}\frac{\partial \boldsymbol{u}_r}{\partial \boldsymbol{\mu}}(\boldsymbol{\mu};\, \boldsymbol{\Phi},\, \boldsymbol{\Psi},\, \boldsymbol{P}) \tag{4.100}$$

where $\boldsymbol{u}_r(\boldsymbol{\mu};\, \boldsymbol{\Phi},\, \boldsymbol{\Psi},\, \boldsymbol{P})$ are the reduced coordinates corresponding to the solution of the hyperreduced model in (4.82) and $\dfrac{\partial \boldsymbol{u}_r}{\partial \boldsymbol{\mu}}(\boldsymbol{\mu};\, \boldsymbol{\Phi},\, \boldsymbol{\Psi},\, \boldsymbol{P})$ is the corresponding sensitivity. The reduced sensitivities are derived by differentiating the governing hyperreduced model in (4.82). In the general case where $\boldsymbol{P}^T\boldsymbol{\Psi}$ is state- and parameter-dependent, the reduced sensitivities are defined as the solution of the linear system of equations

$$\begin{aligned}\left[\sum_{j=1}^{|\mathcal{M}|}(\boldsymbol{P}^T\boldsymbol{r})_j \frac{\partial\left((\boldsymbol{P}^T\boldsymbol{\Psi})^T\boldsymbol{e}_j\right)}{\partial \boldsymbol{u}}\bar{\boldsymbol{P}}\bar{\boldsymbol{P}}^T\boldsymbol{\Phi} + (\boldsymbol{P}^T\boldsymbol{\Psi})^T\frac{\partial \boldsymbol{r}_h}{\partial \boldsymbol{u}_h}\right]\frac{\partial \boldsymbol{u}_r}{\partial \boldsymbol{\mu}} = \\ -\left[\sum_{j=1}^{|\mathcal{M}|}(\boldsymbol{P}^T\boldsymbol{r})_j \frac{\partial\left((\boldsymbol{P}^T\boldsymbol{\Psi})^T\boldsymbol{e}_j\right)}{\partial \boldsymbol{\mu}} + (\boldsymbol{P}^T\boldsymbol{\Psi})^T\frac{\partial \boldsymbol{r}_h}{\partial \boldsymbol{\mu}}\right]\end{aligned} \tag{4.101}$$

where all terms are evaluated at the primal solution $\boldsymbol{u}_h(\boldsymbol{\mu};\, \boldsymbol{\Phi},\, \boldsymbol{\Psi},\, \boldsymbol{P})$. In the special case where the masked primal solution is exact *on the mask*, i.e., $\boldsymbol{r}_h(\boldsymbol{u}_h(\boldsymbol{\mu}),\, \boldsymbol{\mu}) = 0$, or the masked test basis is constant, the expression in (4.101) reduces to

$$\left[(\boldsymbol{P}^T\boldsymbol{\Psi})^T\frac{\partial \boldsymbol{r}_h}{\partial \boldsymbol{u}_h}\right]\frac{\partial \boldsymbol{u}_r}{\partial \boldsymbol{\mu}} = -(\boldsymbol{P}^T\boldsymbol{\Psi})^T\frac{\partial \boldsymbol{r}_h}{\partial \boldsymbol{\mu}}. \tag{4.102}$$

A Galerkin projection uses a constant test basis $\boldsymbol{P}^T\boldsymbol{\Psi} = \boldsymbol{P}^T\boldsymbol{\Phi}$ and the hyperreduced sensitivity equations takes the form

$$\left[(\boldsymbol{P}^T\boldsymbol{\Phi})^T\frac{\partial \boldsymbol{r}_h}{\partial \boldsymbol{u}_h}\right]\frac{\partial \boldsymbol{u}_r}{\partial \boldsymbol{\mu}} = -(\boldsymbol{P}^T\boldsymbol{\Phi})^T\frac{\partial \boldsymbol{r}_h}{\partial \boldsymbol{\mu}}. \tag{4.103}$$

A LSPG projection employs the non-constant test basis $\boldsymbol{P}^T\boldsymbol{\Psi}(\boldsymbol{u},\,\boldsymbol{\mu}) = \dfrac{\partial \boldsymbol{r}_h}{\partial \boldsymbol{u}_h}(\boldsymbol{u},\,\boldsymbol{\mu})$ and derivatives of the test basis cannot be ignored. The resulting hyperreduced sensitivity equations are

$$\begin{aligned}\left[\sum_{j=1}^{|\mathcal{M}|}(\boldsymbol{P}^T\boldsymbol{r})_j(\bar{\boldsymbol{P}}^T\boldsymbol{\Phi})^T\left(\bar{\boldsymbol{P}}^T\frac{\partial^2 \boldsymbol{r}_j}{\partial \boldsymbol{u}\partial \boldsymbol{u}}\bar{\boldsymbol{P}}\right)(\bar{\boldsymbol{P}}^T\boldsymbol{\Phi}) + \frac{\partial \boldsymbol{r}_h}{\partial \boldsymbol{u}_h}^T\frac{\partial \boldsymbol{r}_h}{\partial \boldsymbol{u}_h}\right]\frac{\partial \boldsymbol{u}_r}{\partial \boldsymbol{\mu}} = \\ -\left[\sum_{j=1}^{|\mathcal{M}|}(\boldsymbol{P}^T\boldsymbol{r})_j(\bar{\boldsymbol{P}}^T\boldsymbol{\Phi})^T\left(\bar{\boldsymbol{P}}^T\frac{\partial^2 \boldsymbol{r}_j}{\partial \boldsymbol{u}\partial \boldsymbol{\mu}}\right) + \frac{\partial \boldsymbol{r}_h}{\partial \boldsymbol{u}_h}^T\frac{\partial \boldsymbol{r}_h}{\partial \boldsymbol{\mu}}\right]\end{aligned} \tag{4.104}$$

The difficulty associated with computing derivatives of the test basis, as well as the merits of minimum-residual formulations discussed in Section 4.1, motivate the introduction of a collocation-based equivalent of the minimum-residual sensitivity reduced-order model of Section 4.1.2. For generality, consider the low-dimensional approximation of the high-dimensional model sensitivity

$$\frac{\partial \boldsymbol{u}}{\partial \boldsymbol{\mu}} = \boldsymbol{\Phi}^{\partial}\widehat{\frac{\partial \boldsymbol{u}_r}{\partial \boldsymbol{\mu}}}, \tag{4.105}$$

where $\boldsymbol{\Phi}^{\partial} \in \mathbb{R}^{N_{\boldsymbol{u}}\times k_{\boldsymbol{u}}}$ is the reduced-order basis for the sensitivities and $\widehat{\dfrac{\partial \boldsymbol{u}_r}{\partial \boldsymbol{\mu}}}$ are the corresponding reduced coordinates. The reduced coordinates are defined as the argument that minimizes the sensitivity residual *on the mask*, i.e.,

$$\widehat{\frac{\partial \boldsymbol{u}_r}{\partial \boldsymbol{\mu}}}(\boldsymbol{\mu};\,\boldsymbol{\Phi}^{\partial},\,\boldsymbol{\Theta}^{\partial},\,\boldsymbol{P},\,\boldsymbol{u}) = \underset{\boldsymbol{w}_r\in\mathbb{R}^{k_{\boldsymbol{u}}\times N_{\boldsymbol{\mu}}}}{\arg\min}\ \frac{1}{2}\left\|\boldsymbol{P}^T\boldsymbol{r}^{\partial}(\boldsymbol{u},\,\bar{\boldsymbol{P}}\bar{\boldsymbol{P}}^T\boldsymbol{\Phi}^{\partial}\boldsymbol{w}_r,\,\boldsymbol{\mu})\right\|^2_{\boldsymbol{\Theta}^{\partial}} \tag{4.106}$$

where $\boldsymbol{u} \in \mathbb{R}^{N_{\boldsymbol{u}}}$ is any linearization point, usually the reconstructed primal solution, i.e., $\boldsymbol{u} = \bar{\boldsymbol{P}}\bar{\boldsymbol{P}}^T\boldsymbol{\Phi}\boldsymbol{u}_r(\boldsymbol{\mu};\,\boldsymbol{\Phi},\,\boldsymbol{\Psi},\,\boldsymbol{P})$ and $\boldsymbol{\Theta}^{\partial} \succ 0$ is the metric defining the norm. The first-order optimality condition of the *linear* least-squares problem in (4.106) leads to the normal equations

$$\left(\boldsymbol{P}^T\frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}\bar{\boldsymbol{P}}\bar{\boldsymbol{P}}^T\boldsymbol{\Phi}^{\partial}\right)^T\boldsymbol{\Theta}^{\partial}\left(\boldsymbol{P}^T\frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}\bar{\boldsymbol{P}}\bar{\boldsymbol{P}}^T\boldsymbol{\Phi}^{\partial}\right)\widehat{\frac{\partial \boldsymbol{u}_r}{\partial \boldsymbol{\mu}}} = -\left(\boldsymbol{P}^T\frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}\bar{\boldsymbol{P}}\bar{\boldsymbol{P}}^T\boldsymbol{\Phi}^{\partial}\right)^T\boldsymbol{\Theta}^{\partial}\boldsymbol{P}^T\frac{\partial \boldsymbol{r}}{\partial \boldsymbol{\mu}}. \tag{4.107}$$

where all terms are evaluated at the linearization point. A variant of the monotonicity and interpolation properties of Proposition 4.2 hold for masked minimum-residual sensitivity hyperreduced model. In this case, monotonicity is guaranteed with respect to a fixed metric *and* mask and interpolation requires a sufficiently large mask such that $\boldsymbol{P}^T\boldsymbol{r}(\boldsymbol{u},\,\boldsymbol{\mu}) = 0 \implies \boldsymbol{r}(\boldsymbol{u},\,\boldsymbol{\mu}) = 0$. These results are stated and proved in Proposition 4.7.

Proposition 4.7. *Let $(\boldsymbol{\Phi}^{\partial}, \boldsymbol{\Theta}^{\partial},\ \boldsymbol{P})$ define a masked minimum-residual sensitivity reduced-order*

model. Then the following properties hold for any $\boldsymbol{\mu} \in \mathbb{R}^{N_{\boldsymbol{\mu}}}$

- *(Optimality) For any* $\boldsymbol{u} \in \mathbb{R}^{N_{\boldsymbol{u}}}$, $\boldsymbol{w} \in \mathbb{R}^{N_{\boldsymbol{u}}}$, *and* $\bar{\boldsymbol{P}}^T\boldsymbol{w} \in col(\bar{\boldsymbol{P}}^T\boldsymbol{\Phi}^{\partial})$, *then*

$$\left\| \boldsymbol{P}^T \boldsymbol{r}_k^{\partial} \left(\boldsymbol{u},\, \bar{\boldsymbol{P}}\bar{\boldsymbol{P}}^T\boldsymbol{\Phi}^{\partial} \widehat{\frac{\partial \boldsymbol{u}_r}{\partial \boldsymbol{\mu}}}(\boldsymbol{\mu};\, \boldsymbol{\Phi}^{\partial}, \boldsymbol{\Theta}^{\partial},\, \boldsymbol{P},\, \boldsymbol{u}) \boldsymbol{e}_k,\, \boldsymbol{\mu} \right) \right\|_{\boldsymbol{\Theta}^{\partial}} \leq \left\| \boldsymbol{P}^T \boldsymbol{r}_k^{\partial} \left(\boldsymbol{u},\, \bar{\boldsymbol{P}}\bar{\boldsymbol{P}}^T\boldsymbol{w},\, \boldsymbol{\mu} \right) \right\|_{\boldsymbol{\Theta}^{\partial}} \tag{4.108}$$

  for $k = 1, \ldots, N_{\boldsymbol{\mu}}$, *where* $\boldsymbol{r}_k^{\partial}(\boldsymbol{u},\, \boldsymbol{w} \cdot \boldsymbol{e}_k,\, \boldsymbol{\mu}) \coloneqq \boldsymbol{r}^{\partial}(\boldsymbol{u},\, \boldsymbol{w}\boldsymbol{e}^T,\, \boldsymbol{\mu})\boldsymbol{e}_k$ *and* $\boldsymbol{e}_k$ *is the* k*th canonical unit vector.*

- *(Monotonicity) Let* $(\boldsymbol{\Phi}^{\partial'}, \boldsymbol{\Theta}^{\partial'}, \boldsymbol{P})$ *define a masked minimum-residual sensitivity reduced-order model such that* $col(\boldsymbol{\Phi}^{\partial'}) \subseteq col(\boldsymbol{\Phi}^{\partial})$, *then*

$$\left\| \boldsymbol{P}^T \boldsymbol{r}_k^{\partial} \left(\boldsymbol{u},\, \boldsymbol{w},\, \boldsymbol{\mu} \right) \right\|_{\boldsymbol{\Theta}^{\partial}} \leq \left\| \boldsymbol{P}^T \boldsymbol{r}_k^{\partial} \left(\boldsymbol{u},\, \boldsymbol{w}',\, \boldsymbol{\mu} \right) \right\|_{\boldsymbol{\Theta}^{\partial}}, \tag{4.109}$$

  where $\boldsymbol{w} = \boldsymbol{\Phi}^{\partial} \widehat{\frac{\partial \boldsymbol{u}_r}{\partial \boldsymbol{\mu}}}(\boldsymbol{\mu};\, \boldsymbol{\Phi}^{\partial}, \boldsymbol{\Theta}^{\partial},\, \boldsymbol{P},\, \boldsymbol{u})\boldsymbol{e}_k$ *and* $\boldsymbol{w}' = \boldsymbol{\Phi}^{\partial'} \widehat{\frac{\partial \boldsymbol{u}_r}{\partial \boldsymbol{\mu}}}(\boldsymbol{\mu};\, \boldsymbol{\Phi}^{\partial'}, \boldsymbol{\Theta}^{\partial'},\, \boldsymbol{P},\, \boldsymbol{u})\boldsymbol{e}_k$, *for* $k = 1, \ldots, N_{\boldsymbol{\mu}}$ *and any* $\boldsymbol{u} \in \mathbb{R}^{N_{\boldsymbol{\mu}}}$.

- *(Interpolation) If* $\bar{\boldsymbol{P}}^T \dfrac{\partial \boldsymbol{u}}{\partial \boldsymbol{\mu}_k}(\boldsymbol{\mu}) \in col(\bar{\boldsymbol{P}}^T\boldsymbol{\Phi}^{\partial})$, *then*

$$\boldsymbol{P}^T \boldsymbol{r}_k^{\partial} \left(\boldsymbol{u},\, \bar{\boldsymbol{P}}\bar{\boldsymbol{P}}^T\boldsymbol{\Phi}^{\partial} \widehat{\frac{\partial \boldsymbol{u}_r}{\partial \boldsymbol{\mu}}}(\boldsymbol{\mu};\, \boldsymbol{\Phi}^{\partial}, \boldsymbol{\Theta}^{\partial},\, \boldsymbol{P},\, \boldsymbol{u}) \boldsymbol{e}_k,\, \boldsymbol{\mu} \right) = 0. \tag{4.110}$$

Proof. Optimality follows from the fact that $\widehat{\frac{\partial \boldsymbol{u}_r}{\partial \boldsymbol{\mu}}}(\boldsymbol{\mu};\, \boldsymbol{\Phi}^{\partial},\, \boldsymbol{\Theta}^{\partial},\, \boldsymbol{P},\, \boldsymbol{u})$ is the (unique) minima of the optimization problem in (4.106). A simple application of the optimality property to

$$\bar{\boldsymbol{P}}^T\boldsymbol{\Phi}^{\partial'} \widehat{\frac{\partial \boldsymbol{u}_r}{\partial \boldsymbol{\mu}}}(\boldsymbol{\mu};\, \boldsymbol{\Phi}^{\partial'}, \boldsymbol{\Theta}^{\partial'},\, \boldsymbol{P},\, \boldsymbol{u}) \in \mathrm{col}(\bar{\boldsymbol{P}}^T\boldsymbol{\Phi}^{\partial'}) \subseteq \mathrm{col}(\bar{\boldsymbol{P}}^T\boldsymbol{\Phi}^{\partial})$$

leads to monotonicity. Finally, if a solution of $\boldsymbol{P}^T\boldsymbol{r}^{\partial}(\bar{\boldsymbol{P}}\bar{\boldsymbol{P}}^T\boldsymbol{u}(\boldsymbol{\mu}), \cdot, \boldsymbol{\mu}) = 0$ is contained in the columnspace of $\bar{\boldsymbol{P}}^T\boldsymbol{\Phi}^{\partial}$, i.e., $\bar{\boldsymbol{P}}^T \dfrac{\partial \boldsymbol{u}}{\partial \boldsymbol{\mu}}(\boldsymbol{\mu}) \in \mathrm{col}(\bar{\boldsymbol{P}}^T\boldsymbol{\Phi}^{\partial})$, the optimality property implies

$$\left\| \boldsymbol{P}^T \boldsymbol{r}^{\partial} \left(\hat{\boldsymbol{u}}(\boldsymbol{\mu}),\, \bar{\boldsymbol{P}}\bar{\boldsymbol{P}}^T\boldsymbol{\Phi}^{\partial} \widehat{\frac{\partial \boldsymbol{u}_r}{\partial \boldsymbol{\mu}}}(\boldsymbol{\mu};\, \boldsymbol{\Phi}^{\partial}, \boldsymbol{\Theta}^{\partial},\, \boldsymbol{P},\, \hat{\boldsymbol{u}}(\boldsymbol{\mu})),\, \boldsymbol{\mu} \right) \right\|_{\boldsymbol{\Theta}^{\partial}} \leq \left\| \boldsymbol{P}^T \boldsymbol{r}^{\partial} \left(\hat{\boldsymbol{u}}(\boldsymbol{\mu}),\, \frac{\partial \hat{\boldsymbol{u}}}{\partial \boldsymbol{\mu}}(\boldsymbol{\mu}),\, \boldsymbol{\mu} \right) \right\|_{\boldsymbol{\Theta}^{\partial}} = 0, \tag{4.111}$$

where $\hat{\boldsymbol{u}}(\boldsymbol{\mu}) = \bar{\boldsymbol{P}}\bar{\boldsymbol{P}}^T\boldsymbol{u}(\boldsymbol{\mu})$ and $\dfrac{\partial \hat{\boldsymbol{u}}}{\partial \boldsymbol{\mu}}(\boldsymbol{\mu}) = \bar{\boldsymbol{P}}\bar{\boldsymbol{P}}^T \dfrac{\partial \boldsymbol{u}}{\partial \boldsymbol{\mu}}(\boldsymbol{\mu})$, which is precisely the interpolation property in (4.110). □

In addition to monotonicity and interpolation, conditions exist (Proposition 4.8) that guarantee the two types of sensitivities introduced in this section, i.e., the sensitivities of the hyperreduced model and the masked minimum-residual hyperreduced sensitivities, agree. These conditions also

guarantee that the reconstruction of these sensitivities in the full space yield the same approximation of the high-dimensional model sensitivities. Among these conditions is a required relationship (4.112) between the trial ($\boldsymbol{\Phi}$) and test basis ($\boldsymbol{\Psi}$) for the primal hyperreduced model, the sensitivity metric ($\boldsymbol{\Theta}$), and the sensitivity basis ($\boldsymbol{\Phi}^\partial$). The result of Proposition 4.8 is significant since it provides conditions under which the easily computed masked minimum-residual hyperreduced sensitivities (independent of second derivatives of $\boldsymbol{r}$) match the desired hyperreduction sensitivities (guarantee consistency of gradient computations).

Proposition 4.8. *Consider a primal hyperreduced model defined by trial and test bases $\boldsymbol{\Phi}$ and $\boldsymbol{\Psi}$, respectively, and mask $\boldsymbol{P}$ and a minimum-residual sensitivity hyperreduced model defined by basis $\boldsymbol{\Phi}^\partial$, mask $\boldsymbol{P}$, and metric $\boldsymbol{\Theta}^\partial$. Suppose that either: (1) the primal solution of the hyperreduced model exactly reconstructs the HDM solution on the sample mesh, i.e.,*

$$\bar{\boldsymbol{P}}^T\boldsymbol{u}(\boldsymbol{\mu}) = \bar{\boldsymbol{P}}^T\boldsymbol{\Phi}\boldsymbol{u}_r(\boldsymbol{\mu};\,\boldsymbol{\Phi},\,\boldsymbol{\Psi})$$

or *(2) the masked test basis $\boldsymbol{P}^T\boldsymbol{\Psi}$ is* constant. *Then, for any $\boldsymbol{u} \in \mathbb{R}^{N_{\boldsymbol{u}}}$, the relationships*

$$\begin{aligned} \boldsymbol{P}^T\boldsymbol{\Phi}^\partial &= \boldsymbol{P}^T\boldsymbol{\Phi} \\ \boldsymbol{P}^T\boldsymbol{\Psi}(\boldsymbol{u},\,\boldsymbol{\mu}) &= \boldsymbol{\Theta}^\partial\boldsymbol{P}^T\frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}(\boldsymbol{u},\,\boldsymbol{\mu})\bar{\boldsymbol{P}}\bar{\boldsymbol{P}}^T\boldsymbol{\Phi}^\partial \end{aligned} \tag{4.112}$$

guarantee the sensitivity of the primal hyperreduced model ($\boldsymbol{\Phi}$, $\boldsymbol{\Psi}$, $\boldsymbol{P}$) coincides with the solution of the minimum-residual sensitivity hyperreduced model ($\boldsymbol{\Phi}^\partial$, $\boldsymbol{\Theta}^\partial$, $\boldsymbol{P}$)

$$\frac{\partial \boldsymbol{u}_r}{\partial \boldsymbol{\mu}}(\boldsymbol{\mu};\,\boldsymbol{\Phi},\,\boldsymbol{\Psi},\,\boldsymbol{P}) = \widehat{\frac{\partial \boldsymbol{u}_r}{\partial \boldsymbol{\mu}}}(\boldsymbol{\mu};\,\boldsymbol{\Phi}^\partial,\,\boldsymbol{\Theta}^\partial,\,\boldsymbol{P},\,\boldsymbol{\Phi}\boldsymbol{u}_r(\boldsymbol{\mu};\,\boldsymbol{\Phi},\,\boldsymbol{\Psi},\,\boldsymbol{P})). \tag{4.113}$$

Proof. Let $\bar{\boldsymbol{P}}^T\boldsymbol{\Phi}\boldsymbol{u}_r = \bar{\boldsymbol{P}}^T\boldsymbol{\Phi}\boldsymbol{u}_r(\boldsymbol{\mu};\,\boldsymbol{\Phi},\,\boldsymbol{\Psi},\,\boldsymbol{P})$ denote the reconstructed primal solution of the projection-based reduced-order model, restricted to the primal mesh. If either the primal solution is exact ($\boldsymbol{P}^T\boldsymbol{r}(\bar{\boldsymbol{P}}\bar{\boldsymbol{P}}^T\boldsymbol{\Phi}\boldsymbol{u}_r,\,\boldsymbol{\mu}) = 0$) or the test basis is constant, the general form of the reduced-order model sensitivity equations in (4.101) reduces to the equation in (4.102), where all terms are evaluated at the primal solution, i.e.,

$$\left[\left(\boldsymbol{P}^T\boldsymbol{\Psi}(\bar{\boldsymbol{P}}\bar{\boldsymbol{P}}^T\boldsymbol{\Phi}\boldsymbol{u}_r,\,\boldsymbol{\mu})\right)^T\frac{\partial \boldsymbol{r}_h}{\partial \boldsymbol{u}_h}(\bar{\boldsymbol{P}}\bar{\boldsymbol{P}}^T\boldsymbol{\Phi}\boldsymbol{u}_r,\,\boldsymbol{\mu})\right]\frac{\partial \boldsymbol{u}_r}{\partial \boldsymbol{\mu}} = -\left(\boldsymbol{P}^T\boldsymbol{\Psi}(\bar{\boldsymbol{P}}\bar{\boldsymbol{P}}^T\boldsymbol{\Phi}\boldsymbol{u}_r,\,\boldsymbol{\mu})\right)^T\frac{\partial \boldsymbol{r}_h}{\partial \boldsymbol{\mu}}(\bar{\boldsymbol{P}}\bar{\boldsymbol{P}}^T\boldsymbol{\Phi}\boldsymbol{u}_r,\,\boldsymbol{\mu}). \tag{4.114}$$

Conversely, the normal form of the minimum-residual sensitivity reduced-order model in (4.107) reduces to

$$\left[\left(\boldsymbol{P}^T\boldsymbol{\Psi}(\bar{\boldsymbol{P}}\bar{\boldsymbol{P}}^T\boldsymbol{\Phi}\boldsymbol{u}_r,\,\boldsymbol{\mu})\right)^T\frac{\partial \boldsymbol{r}_h}{\partial \boldsymbol{u}_h}(\bar{\boldsymbol{P}}\bar{\boldsymbol{P}}^T\boldsymbol{\Phi}\boldsymbol{u}_r,\,\boldsymbol{\mu})\right]\widehat{\frac{\partial \boldsymbol{u}_r}{\partial \boldsymbol{\mu}}} = -\left(\boldsymbol{P}^T\boldsymbol{\Psi}(\bar{\boldsymbol{P}}\bar{\boldsymbol{P}}^T\boldsymbol{\Phi}\boldsymbol{u}_r,\,\boldsymbol{\mu})\right)^T\frac{\partial \boldsymbol{r}_h}{\partial \boldsymbol{\mu}}(\bar{\boldsymbol{P}}\bar{\boldsymbol{P}}^T\boldsymbol{\Phi}\boldsymbol{u}_r,\,\boldsymbol{\mu}). \tag{4.115}$$

when the relationships in (4.112) are enforced. Thus, under the aforementioned conditions, the

governing equations for $\dfrac{\partial \boldsymbol{u}_r}{\partial \boldsymbol{\mu}}$ and $\widehat{\dfrac{\partial \boldsymbol{u}_r}{\partial \boldsymbol{\mu}}}$ are identical and the (unique) solutions must be equal, which establishes (4.113). □

4.2.6 Adjoint Formulation

The adjoint equations for the collocation-based hyperreduced model are derived using the optimization procedure, outlined in Section 2.3.4, applied to the governing equation $(\boldsymbol{P}^T\boldsymbol{\Psi})^T\boldsymbol{r}_h(\bar{\boldsymbol{P}}\bar{\boldsymbol{P}}^T\boldsymbol{\Phi}\boldsymbol{u}_r,\, \boldsymbol{\mu}) = 0$ and reduced quantity of interest $f(\boldsymbol{\Phi}\boldsymbol{u}_r,\, \boldsymbol{\mu})$. Consider the auxiliary optimization problem

$$\begin{aligned} &\underset{\boldsymbol{u}_r \in \mathbb{R}^{k_{\boldsymbol{u}}}}{\text{minimize}} && f(\boldsymbol{\Phi}\boldsymbol{u}_r,\, \hat{\boldsymbol{\mu}}) \\ &\text{subject to} && (\boldsymbol{P}^T\boldsymbol{\Psi})^T\boldsymbol{r}_h(\boldsymbol{u}_h,\, \hat{\boldsymbol{\mu}}) = 0 \end{aligned} \tag{4.116}$$

for a fixed $\hat{\boldsymbol{\mu}}$ and the corresponding Lagrangian

$$\mathcal{L}_r(\boldsymbol{u}_r,\, \boldsymbol{\lambda}_r) = f(\boldsymbol{\Phi}\boldsymbol{u}_r,\, \hat{\boldsymbol{\mu}}) - \boldsymbol{\lambda}_r^T(\boldsymbol{P}^T\boldsymbol{\Psi})^T\boldsymbol{r}_h(\bar{\boldsymbol{P}}\bar{\boldsymbol{P}}^T\boldsymbol{\Phi}\boldsymbol{u}_r,\, \hat{\boldsymbol{\mu}}). \tag{4.117}$$

by comparing this expression for the Lagrangian with that in (2.98), it is clear that the masked HDM Lagrange multipliers are reconstructed from the reduced Lagrange multipliers as

$$\boldsymbol{\lambda} = \boldsymbol{\Psi}\boldsymbol{\lambda}_r. \tag{4.118}$$

The stationarity of the Lagrangian with respect to $\boldsymbol{u}_r$ leads to the reduced adjoint equations

$$\left[\sum_{j=1}^{|\mathcal{M}|}(\boldsymbol{P}^T\boldsymbol{r})_j \frac{\partial\left((\boldsymbol{P}^T\boldsymbol{\Psi})^T\boldsymbol{e}_j\right)}{\partial \boldsymbol{u}}\bar{\boldsymbol{P}}\bar{\boldsymbol{P}}^T\boldsymbol{\Phi} + (\boldsymbol{P}^T\boldsymbol{\Psi})^T\frac{\partial \boldsymbol{r}_h}{\partial \boldsymbol{u}_h}\right]\boldsymbol{\lambda}_r = \boldsymbol{\Phi}^T\frac{\partial f}{\partial \boldsymbol{u}}^T \tag{4.119}$$

where all terms are evaluated at the reconstructed primal solution, $\bar{\boldsymbol{P}}\bar{\boldsymbol{P}}^T\boldsymbol{\Phi}\boldsymbol{u}_r(\boldsymbol{\mu};\, \boldsymbol{\Phi},\, \boldsymbol{\Psi},\, \boldsymbol{P})$. For any $\boldsymbol{\mu} \in \mathbb{R}^{N_{\boldsymbol{\mu}}}$, bases $\boldsymbol{\Phi}$, $\boldsymbol{\Psi}$, and mask $\boldsymbol{P}$, the solution of the above equation is denoted $\boldsymbol{\lambda}_r(\boldsymbol{\mu};\, \boldsymbol{\Phi},\, \boldsymbol{\Psi},\, \boldsymbol{P})$. The gradient of the quantity of interest is reconstructed as

$$\begin{aligned} \nabla F_r(\boldsymbol{\mu}) =& \frac{\partial f}{\partial \boldsymbol{\mu}}(\boldsymbol{\Phi}\boldsymbol{u}_r(\boldsymbol{\mu};\, \boldsymbol{\Phi},\, \boldsymbol{\Psi},\, \boldsymbol{P}),\, \boldsymbol{\mu}) - \\ & \boldsymbol{\lambda}_r(\boldsymbol{\mu};\, \boldsymbol{\Phi},\, \boldsymbol{\Psi},\, \boldsymbol{P})^T\left[\sum_{j=1}^{|\mathcal{M}|}(\boldsymbol{P}^T\boldsymbol{r})_j\frac{\partial\left(\boldsymbol{\Psi}^T\boldsymbol{e}_j\right)}{\partial \boldsymbol{\mu}} + (\boldsymbol{P}^T\boldsymbol{\Psi})^T\frac{\partial \boldsymbol{r}_h}{\partial \boldsymbol{\mu}}\right]_{(\boldsymbol{u}_h,\, \boldsymbol{\mu})} \end{aligned} \tag{4.120}$$

where $\boldsymbol{u}_h = \bar{\boldsymbol{P}}\bar{\boldsymbol{P}}^T\boldsymbol{\Phi}\boldsymbol{u}_r(\boldsymbol{\mu};\, \boldsymbol{\Phi},\, \boldsymbol{\Psi},\, \boldsymbol{P}),\, \boldsymbol{\mu})$. In the special case where the primal solution is exact on the mask, in the sense that $\boldsymbol{r}_h(\boldsymbol{u}_h,\, \boldsymbol{\mu}) = 0$, or the masked test basis is constant, the adjoint equations in (4.119) reduce to

$$\left[(\boldsymbol{P}^T\boldsymbol{\Psi})^T\frac{\partial \boldsymbol{r}_h}{\partial \boldsymbol{u}_h}\right]^T\boldsymbol{\lambda}_r = \boldsymbol{\Phi}^T\frac{\partial f}{\partial \boldsymbol{u}}^T \tag{4.121}$$

and the gradient of the QoI becomes

$$\nabla F_r(\boldsymbol{\mu}) = \frac{\partial f}{\partial \boldsymbol{\mu}}(\boldsymbol{\Phi}\boldsymbol{u}_r(\boldsymbol{\mu};\, \boldsymbol{\Phi},\, \boldsymbol{\Psi},\, \boldsymbol{P}),\, \boldsymbol{\mu}) - \boldsymbol{\lambda}_r(\boldsymbol{\mu};\, \boldsymbol{\Phi},\, \boldsymbol{\Psi},\, \boldsymbol{P})^T \left[(\boldsymbol{P}^T\boldsymbol{\Psi})^T \frac{\partial \boldsymbol{r}_h}{\partial \boldsymbol{\mu}}\right]_{(\boldsymbol{u}_h,\, \boldsymbol{\mu})} \tag{4.122}$$

In general, the term $\boldsymbol{\Phi}^T \frac{\partial f}{\partial \boldsymbol{u}}^T$ requires $\mathcal{O}(N_{\boldsymbol{u}})$ works and memory to evaluate. However, in many applications f corresponds to an integral over a surface or small portion of the domain and therefore $\partial f/\partial \boldsymbol{u}$ is sparse. Then this terms is exactly equal to $\boldsymbol{\Phi}^T \frac{\partial f}{\partial \boldsymbol{u}}^T = (\hat{\boldsymbol{P}}^T\boldsymbol{\Phi})^T \hat{\boldsymbol{P}}^T \frac{\partial f}{\partial \boldsymbol{u}}^T$, where $\hat{\boldsymbol{P}}$ is the subset of columns of the identity matrix that exactly restricts $\partial f/\partial \boldsymbol{u}$ to its nonzero entries and can be computed without requiring large-scale operations. This is an important implementation detail; however, for simplicity, the additional notation will not be continued.

In the special case where the primal hyperreduced model employs a Galerkin projection ($\boldsymbol{P}^T\boldsymbol{\Psi} = \boldsymbol{P}^T\boldsymbol{\Phi}$), the test basis is state- and parameter-independent and the adjoint equations in (4.119) or (4.121) become

$$\left[(\boldsymbol{P}^T\boldsymbol{\Phi})^T \frac{\partial \boldsymbol{r}_h}{\partial \boldsymbol{u}_h}\right]^T \boldsymbol{\lambda}_r = \boldsymbol{\Phi}^T \frac{\partial f}{\partial \boldsymbol{u}}^T \tag{4.123}$$

and the gradient of the QoI is

$$\nabla F_r(\boldsymbol{\mu}) = \frac{\partial f}{\partial \boldsymbol{\mu}}(\boldsymbol{\Phi}\boldsymbol{u}_r(\boldsymbol{\mu};\, \boldsymbol{\Phi},\, \boldsymbol{\Phi},\, \boldsymbol{P}),\, \boldsymbol{\mu}) - \boldsymbol{\lambda}_r(\boldsymbol{\mu};\, \boldsymbol{\Phi},\, \boldsymbol{\Phi},\, \boldsymbol{P})^T \left[(\boldsymbol{P}^T\boldsymbol{\Phi})^T \frac{\partial \boldsymbol{r}_h}{\partial \boldsymbol{\mu}}\right]_{(\boldsymbol{u}_h,\, \boldsymbol{\mu})} \tag{4.124}$$

In the special case of a LSPG projection, the adjoint equations in (4.119) become

$$\left[\sum_{j=1}^{|\mathcal{M}|} (\boldsymbol{P}^T\boldsymbol{r})_j (\bar{\boldsymbol{P}}^T\boldsymbol{\Phi})^T \left(\bar{\boldsymbol{P}}^T \frac{\partial^2 \boldsymbol{r}_j}{\partial \boldsymbol{u}\partial \boldsymbol{u}} \bar{\boldsymbol{P}}\right) (\bar{\boldsymbol{P}}^T\boldsymbol{\Phi}) + \frac{\partial \boldsymbol{r}_h}{\partial \boldsymbol{u}_h}^T \frac{\partial \boldsymbol{r}_h}{\partial \boldsymbol{u}_h}\right] \boldsymbol{\lambda}_r = \boldsymbol{\Phi}^T \frac{\partial f}{\partial \boldsymbol{u}}^T \tag{4.125}$$

and the gradient of the QoI is

$$\begin{aligned}\nabla F_r(\boldsymbol{\mu}) =& \frac{\partial f}{\partial \boldsymbol{\mu}}(\boldsymbol{\Phi}\boldsymbol{u}_r(\boldsymbol{\mu};\, \boldsymbol{\Phi},\, \boldsymbol{\Psi},\, \boldsymbol{P}),\, \boldsymbol{\mu}) - \\ & \boldsymbol{\lambda}_r(\boldsymbol{\mu};\, \boldsymbol{\Phi},\, \boldsymbol{\Psi},\, \boldsymbol{P})^T \left[\sum_{j=1}^{|\mathcal{M}|} (\boldsymbol{P}^T\boldsymbol{r})_j (\bar{\boldsymbol{P}}^T\boldsymbol{\Phi})^T \left(\bar{\boldsymbol{P}}^T \frac{\partial^2 \boldsymbol{r}_j}{\partial \boldsymbol{u}\partial \boldsymbol{\mu}}\right) + (\boldsymbol{P}^T\boldsymbol{\Psi})^T \frac{\partial \boldsymbol{r}_h}{\partial \boldsymbol{\mu}}\right]_{(\boldsymbol{u}_h,\, \boldsymbol{\mu})}.\end{aligned} \tag{4.126}$$

The introduction and derivation of the masked minimum-residual hyperreduced adjoint model will be deferred to future work. There are special considerations that rise when considering the minimization problem

$$\underset{\boldsymbol{z}_r \in \mathbb{R}^{k_{\boldsymbol{u}}}}{\text{minimize}} \quad \left\| \boldsymbol{P}^T \boldsymbol{r}^{\boldsymbol{\lambda}}(\boldsymbol{u},\, \boldsymbol{\Phi}^{\boldsymbol{\lambda}}\boldsymbol{z}_r,\, \boldsymbol{\mu}) \right\|_{\boldsymbol{\Theta}^{\boldsymbol{\lambda}}} = \underset{\boldsymbol{z}_r \in \mathbb{R}^{k_{\boldsymbol{u}}}}{\text{minimize}} \quad \left\| -\boldsymbol{P}^T \frac{\partial f}{\partial \boldsymbol{u}}^T + \boldsymbol{P}^T \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}(\boldsymbol{u},\, \boldsymbol{\mu})^T \boldsymbol{\Phi}^{\boldsymbol{\lambda}} \boldsymbol{z}_r \right\|_{\boldsymbol{\Theta}^{\boldsymbol{\lambda}}} \tag{4.127}$$

since the term $\boldsymbol{P}^T \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}(\boldsymbol{u},\, \boldsymbol{\mu})^T \boldsymbol{\Phi}^{\boldsymbol{\lambda}} \boldsymbol{z}_r$ requires a separate restriction matrix $\hat{\boldsymbol{P}}$ (subset of the columns

of the identity matrix) for the following identity to hold

$$\boldsymbol{P}^T \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}(\boldsymbol{u}, \boldsymbol{\mu})^T \boldsymbol{\Phi}^{\boldsymbol{\lambda}} = \boldsymbol{P}^T \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}(\boldsymbol{u}, \boldsymbol{\mu})^T \hat{\boldsymbol{P}} \hat{\boldsymbol{P}}^T \boldsymbol{\Phi}^{\boldsymbol{\lambda}} \tag{4.128}$$

and lead to operations independent of the large dimension $N_{\boldsymbol{u}}$. This is the exact type of implementation optimization discussed in Section 4.2.2 that lead to the efficient computations with the masked reduced Jacobian (no transpose)

$$\boldsymbol{P}^T \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}(\boldsymbol{u}, \boldsymbol{\mu}) \boldsymbol{\Phi} = \boldsymbol{P}^T \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}(\boldsymbol{u}, \boldsymbol{\mu}) \bar{\boldsymbol{P}} \bar{\boldsymbol{P}}^T \boldsymbol{\Phi}. \tag{4.129}$$

4.3 Construction of Reduced-Order Basis and Residual Mask

The present exposition on projection-based reduced-order models has focused on the formulation of the governing equations that guarantee desirable properties *for a fixed* trial basis $\boldsymbol{\Phi}$ and mask $\boldsymbol{P}$; however, there has been no mention of the *origin* of these quantities, a process usually called *training*. The specific training strategy will vary for the various applications encountered in Chapters 5–6 and an in-depth discussion will be deferred to the appropriate chapter. This section details commonalities between the training methods employed in those chapters to facilitate the discussion. Additionally, a general discussion is provided on training concepts used to enforce conditions required for Propositions 4.1, 4.2, 4.4 to hold.

A ubiquitous theme in all reduced-order model training algorithms considered in this document is the *method of snapshots* [183]. This is the idea of building the reduced-order basis $\boldsymbol{\Phi}$ from solutions, or snapshots, of the high-dimensional model. In addition to building the basis from fully converged solutions (individual time steps for unsteady problems [183] or steady states for steady problems), unconverged nonlinear iterations [198], unconverged linear system iterates [198], sensitivities [87, 86, 32, 85, 52, 210, 198], and adjoint solutions [57, 74] have also been used. These various snapshots are combined into the columns of a snapshot matrix with N_s columns. This approach ensures the reduced-order basis includes relevant, information-rich basis vectors that incorporate physics from the underlying PDE and, in many cases, even a small reduced-order basis can result in an accurate reduced-order model.

When the number of snapshots becomes large, it is desirable to apply a compression method to retain most of the original information contained in the snapshots. Among the most popular methods is the Proper Orthogonal Decomposition (POD), Algorithm 4, also known as the truncated Singular Value Decomposition (SVD), Karhunen-Loève (KL) decomposition, and Principal Component Analysis (PCA). POD possess the desirable property of ordering the potential basis vectors according to energy, or importance with regard to reconstructing the snapshot matrix. Therefore the reduced-order basis is taken as the first $k_{\boldsymbol{u}}$ vectors, where $k_{\boldsymbol{u}}$ is chosen based on the singular value decay or naively according to a desired basis size. The latter approach is described in Algorithm 4 and the operation of applying POD to build a reduced-order basis $\boldsymbol{\Phi}$ from the snapshot matrix $\boldsymbol{X}$

will be denoted $\boldsymbol{\Phi} = \texttt{POD}(\boldsymbol{X})$.

Algorithm 4 Proper Orthogonal Decomposition

$$\boldsymbol{\Phi} = \texttt{POD}(\boldsymbol{X})$$

Input: Snapshot matrix $\boldsymbol{X} \in \mathbb{R}^{N_{\boldsymbol{u}} \times k_s}$ and reduced-order basis size $k_{\boldsymbol{u}}$
Output: Reduced-order basis $\boldsymbol{\Phi}$
1: Compute the thin SVD of $\boldsymbol{X}$: $\boldsymbol{X} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$, where $\boldsymbol{U} = \begin{bmatrix}\boldsymbol{u}_1 & \boldsymbol{u}_2 & \cdots & \boldsymbol{u}_{k_s}\end{bmatrix}$
2: $\boldsymbol{\Phi} = \begin{bmatrix}\boldsymbol{u}_1 & \boldsymbol{u}_2 & \cdots & \boldsymbol{u}_{k_{\boldsymbol{u}}}\end{bmatrix}$

POD is well-known to be susceptible to bias when there is substantial variation in the scale of the columns of the snapshot matrix. This will occur, for example, when a heterogeneous collection of snapshots are used, i.e., states and sensitivities, since the *units* of the columns will not be consistent. The result is sub-optimal compression that favors snapshots with the largest size. Following the work in [210], this is remedied by partitioning the heterogeneous snapshot matrix $\boldsymbol{X}$ into homogeneous snapshot matrices $\boldsymbol{Y}$ and $\boldsymbol{Z}$ according to $\boldsymbol{X} = [\boldsymbol{Y}, \boldsymbol{Z}]$. Each homogeneous snapshot matrix is optimally compressed using POD and the results are combined via concatenation to yield the reduced-order basis, i.e., $\boldsymbol{\Phi} = [\texttt{POD}(\boldsymbol{Y}), \texttt{POD}(\boldsymbol{Z})]$. This algorithm, denoted $\boldsymbol{\Phi} = \texttt{PODH}(\boldsymbol{Y}, \boldsymbol{Z})$, is summarized in Algorithm 5 and includes a final step that employs a QR factorization to orthogonalize the basis. An alternate approach, known as *Compact Proper Orthogonal Decomposition* [32], to remove the potential bias of POD, specifically when states and sensitivity snapshot are used, weights the sensitivity snapshots according to the magnitude of parameter perturbations. The former approach based on compression of homogeneous submatrices is preferred in this work due to its generality in handling any types of snapshots, flexibility in handling more than two types of snapshots, and optimality in compressing individual snapshot types (since the compression is POD-based).

Algorithm 5 Proper Orthogonal Decomposition for Heterogeneous Data

$$\boldsymbol{\Phi} = \texttt{PODH}(\boldsymbol{Y}, \boldsymbol{Z})$$

Input: Heterogeneous snapshot matrix, $\boldsymbol{X} = \begin{bmatrix}\boldsymbol{Y} & \boldsymbol{Z}\end{bmatrix}$ and truncation sizes k_y and k_z
Output: Reduced-order basis $\boldsymbol{\Phi}$
1: Compute the thin SVD of $\boldsymbol{Y}$: $\boldsymbol{Y} = \boldsymbol{U}_{\boldsymbol{Y}}\boldsymbol{\Sigma}_{\boldsymbol{Y}}\boldsymbol{V}_{\boldsymbol{Y}}^T$
2: Compute the thin SVD of $\boldsymbol{Z}$: $\boldsymbol{Z} = \boldsymbol{U}_{\boldsymbol{Z}}\boldsymbol{\Sigma}_{\boldsymbol{Z}}\boldsymbol{V}_{\boldsymbol{Z}}^T$
3: Form matrix of dominant singular vectors
4: $\boldsymbol{W} = \begin{bmatrix}(\boldsymbol{u}_{\boldsymbol{Y}})_1 & \cdots & (\boldsymbol{u}_{\boldsymbol{Y}})_{k_y} & (\boldsymbol{u}_{\boldsymbol{Z}})_1 & \cdots & (\boldsymbol{u}_{\boldsymbol{Z}})_{k_z}\end{bmatrix}$
5: Orthogonalize columns of $\boldsymbol{W}$ via QR factorization, $\boldsymbol{\Phi}\boldsymbol{R} = \boldsymbol{W}$

Another desirable property that POD does not possess is the exact preservation of a particular subset snapshots in the span of the reduced basis. The interpolation property of minimum-residual reduced-order models motivates such a property. For some $\boldsymbol{\mu} \in \mathbb{R}^{N_{\boldsymbol{\mu}}}$, suppose $\boldsymbol{X} = [\boldsymbol{u}(\boldsymbol{\mu}), \boldsymbol{X}_2]$ where $\boldsymbol{u}(\boldsymbol{\mu})$ is the exact solution of the high-dimensional model and $\boldsymbol{X}_2$ contains other snapshots. If $\boldsymbol{u}(\boldsymbol{\mu}) \in \text{span}(\boldsymbol{\Phi})$, the resulting (minimum-residual) reduced-order model will exactly recover this solution (Proposition 4.1). This property will prove particularly important in Chapters 5–6, where a certain level of accuracy is required *at trust region centers*. However, if POD is applied to $\boldsymbol{X}$,

$\boldsymbol{u}(\boldsymbol{\mu}) \notin \text{span}(\boldsymbol{\Phi})$, in general, even if $\boldsymbol{u}(\boldsymbol{\mu}) \in \text{span}(\boldsymbol{X})$. To enhance POD to exactly preserve a subsets of the snapshots in the reduced subspace, consider the decomposition of the snapshots as $\boldsymbol{X} = [\boldsymbol{X}_1, \boldsymbol{X}_2]$, where $\boldsymbol{X}_1$ contains the snapshots to be preserved. POD compression is applied only to the snapshot matrix $\boldsymbol{X}_2$ and the reduced-order basis is defined as $\boldsymbol{\Phi} = [\boldsymbol{X}_1, \texttt{POD}(\boldsymbol{X}_2)]$. This algorithm, denoted $\boldsymbol{\Phi} = \texttt{PODSP}(\boldsymbol{X}_1, \boldsymbol{X}_2)$, is summarized in Algorithm 6 and includes a final step that employs a QR factorization to orthogonalize the basis.

Algorithm 6 Proper Orthogonal Decomposition with Span Preservation

$$\boldsymbol{\Phi} = \texttt{PODSP}(\boldsymbol{X}_1, \boldsymbol{X}_2)$$

Input: Snapshot matrix $\boldsymbol{X} \in \mathbb{R}^{N_{\boldsymbol{u}} \times k_s}$ where $\boldsymbol{X} = \begin{bmatrix} \boldsymbol{X}_1 & \boldsymbol{X}_2 \end{bmatrix}$ and truncation size k_x
Output: Reduced-order basis $\boldsymbol{\Phi}$ such that span $\boldsymbol{X}_1 \subset$ span $\boldsymbol{\Phi}$
1: Compute the thin SVD of $\boldsymbol{X}_2$: $\boldsymbol{X}_2 = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$, where $\boldsymbol{U} = \begin{bmatrix} \boldsymbol{u}_1 & \boldsymbol{u}_2 & \cdots & \boldsymbol{u}_{k_s} \end{bmatrix}$
2: $\boldsymbol{W} = \begin{bmatrix} \boldsymbol{X}_1 & \boldsymbol{u}_1 & \cdots & \boldsymbol{u}_{k_x} \end{bmatrix}$
3: Orthogonalize columns of $\boldsymbol{W}$ via QR factorization, $\boldsymbol{\Phi}\boldsymbol{R} = \boldsymbol{W}$

In many cases, heterogeneous snapshots are encountered and certain subsets of each homogeneous snapshot collection must be preserved in the span of the basis. For example, the minimum-residual sensitivity reduced-order models of Section 4.1.2 exactly recover the exact sensitivities at a parameter configuration $\boldsymbol{\mu} \in \mathbb{R}^{N_{\boldsymbol{\mu}}}$ if $\boldsymbol{u}(\boldsymbol{\mu}) \in \text{span}(\boldsymbol{\Phi})$ and $\dfrac{\partial \boldsymbol{u}}{\partial \boldsymbol{\mu}}(\boldsymbol{\mu}) \in \text{span}(\boldsymbol{\Phi})$. In this situation, it is desirable to utilize both state and sensitivity snapshots and preserve the exact state and sensitivity corresponding to parameter configuration $\boldsymbol{\mu}$. This will have important implications in the context of the trust region method introduced in Chapter 5 that requires a certain level of accuracy, in both the objective and gradient, at trust region centers. In such situations, it is desirable to combine the basic enhancements to POD introduced in Algorithms 5 and 6. For this purpose, decompose the heterogeneous snapshot matrix $\boldsymbol{X}$ into homogeneous snapshot matrices $\boldsymbol{Y}$ and $\boldsymbol{Z}$. Further decompose these snapshot matrices according to the subset that must be preserved in the reduced subspace, i.e., $\boldsymbol{Y} = [\boldsymbol{Y}_1, \boldsymbol{Y}_2]$ and $\boldsymbol{Z} = [\boldsymbol{Z}_1, \boldsymbol{Z}_2]$ where the columns of $\boldsymbol{Y}_1$ and $\boldsymbol{Z}_1$ must be contained in the span of $\boldsymbol{\Phi}$. This yields the decomposition of the original snapshot matrix as $\boldsymbol{X} = [\boldsymbol{Y}_1, \boldsymbol{Y}_2, \boldsymbol{Z}_1, \boldsymbol{Z}_2]$ and the basis is defined via POD-based compression to $\boldsymbol{Y}_2$ and $\boldsymbol{Z}_2$ only, i.e., $\boldsymbol{\Phi} = [\boldsymbol{Y}_1, \boldsymbol{Z}_1, \texttt{POD}(\boldsymbol{Y}_2), \texttt{POD}(\boldsymbol{Z}_2)] = [\boldsymbol{Y}_1, \boldsymbol{Z}_1, \texttt{PODH}(\boldsymbol{Y}_2, \boldsymbol{Z}_2)]$. This algorithm, denoted $\boldsymbol{\Phi} = \texttt{PODHSP}(\boldsymbol{Y}_1, \boldsymbol{Y}_2, \boldsymbol{Z}_1, \boldsymbol{Z}_2)$, is summarized in Algorithm 7 and includes a final step that employs a QR factorization to orthogonalize the basis.

At the core of POD, and all the variants introduced in this section, lies a singular value decomposition, which remains among the most expensive matrix factorizations. In many large-scale PDE applications, particularly time-dependent applications, the snapshot matrix that is passed to POD for compression may have $\mathcal{O}(10^8)$ rows and $\mathcal{O}(10^3)$ columns, which requires a substantial amount of computational resources and will be extremely time- and memory-intensive. To reduce the burden of the large-scale SVD computation, a low-rank approximation of the singular value decomposition [81] will be employed for the large-scale CFD problems encountered in Section 5.5.4. The randomized, low-rank SVD, summarized in Algorithm 8, computes the standard SVD of original matrix

Algorithm 7 Proper Orthogonal Decomposition for Heterogeneous Data with Span Preservation

$$\boldsymbol{\Phi} = \texttt{PODHSP}(\boldsymbol{Y}_1, \boldsymbol{Y}_2, \boldsymbol{Z}_1, \boldsymbol{Z}_2)$$

Input: Heterogeneous snapshot matrix $\boldsymbol{X} = \begin{bmatrix}\boldsymbol{Y} & \boldsymbol{Z}\end{bmatrix}$, where $\boldsymbol{Y} = \begin{bmatrix}\boldsymbol{Y}_1 & \boldsymbol{Y}_2\end{bmatrix}$ and $\boldsymbol{Z} = \begin{bmatrix}\boldsymbol{Z}_1 & \boldsymbol{Z}_2\end{bmatrix}$, and truncation sizes, k_y and k_z

Output: Reduced-order basis $\boldsymbol{\Phi}$ such that span $\boldsymbol{Y}_1 \subset$ span $\boldsymbol{\Phi}$ and span $\boldsymbol{Z}_1 \subset$ span $\boldsymbol{\Phi}$

1: Compute the thin SVD of $\boldsymbol{Y}_2$: $\boldsymbol{Y}_2 = \boldsymbol{U}_{\boldsymbol{Y}}\boldsymbol{\Sigma}_{\boldsymbol{Y}}\boldsymbol{V}_{\boldsymbol{Y}}^T$

2: Compute the thin SVD of $\boldsymbol{Z}_2$: $\boldsymbol{Z}_2 = \boldsymbol{U}_{\boldsymbol{Z}}\boldsymbol{\Sigma}_{\boldsymbol{Z}}\boldsymbol{V}_{\boldsymbol{Z}}^T$

3: Form matrix of dominant singular vectors

$$\boldsymbol{W} = \begin{bmatrix}\boldsymbol{Y}_1 & \boldsymbol{Z}_1 & (\boldsymbol{u}_{\boldsymbol{Y}})_1 & \cdots & (\boldsymbol{u}_{\boldsymbol{Y}})_{k_y} & (\boldsymbol{u}_{\boldsymbol{Z}})_1 & \cdots & (\boldsymbol{u}_{\boldsymbol{Z}})_{k_z}\end{bmatrix}$$

4: Orthogonalize columns of $\boldsymbol{W}$ via QR factorization, $\boldsymbol{\Phi}\boldsymbol{R} = \boldsymbol{W}$

projected into a low-dimensional subspace that is constructed through random linear combinations of the matrix. Since a SVD computation scales linearly with the number of rows and quadratically in the number of columns, a substantial performance improvement comes with performing the SVD in the reduced space.

Another bottleneck encountered with all variants of POD is that even low-rank modifications to the underlying snapshots, in general, requires re-computing the SVD from scratch. This is significant since appending new snapshots to the snapshot matrix or re-centering the snapshot matrix cannot necessarily re-use the previous singular factors. A series of papers by Brand [25, 26] changed this landscape as they introduced a series of algorithms for low-rank updates to the SVD. This algorithm, summarized in Algorithm 9 for the case of appending new snapshots to the snapshot matrix and Algorithm 10 for the case for re-centering the snapshot matrix, enables the singular factors of the original SVD to be re-used to compute the SVD of the low-rank update to the snapshot matrix. The cost is mostly independent of operations that scale with the size of the original snapshot matrix.

Algorithm 8 Low-Rank Probabilistic SVD Approximation

$$\boldsymbol{U}, \boldsymbol{\Sigma}, \boldsymbol{V} = \texttt{ProbSVD}(\boldsymbol{X}, k, q)$$

Input: $\mathbf{A} \in \mathbb{R}^{m\times n}$ (usually $n \ll m$), approximation rank k, and number of power iterations q

Output: Approximate SVD of $\mathbf{A} \approx \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T$

1: Generate $n \times 2k$ Gaussian test matrix $\boldsymbol{\Omega}$

2: Form $\mathbf{Y} = (\mathbf{A}\mathbf{A}^T)^q\mathbf{A}\boldsymbol{\Omega}$

3: Compute QR factorization of $\mathbf{Y}$: $\mathbf{Y} = \mathbf{Q}\mathbf{R}$

4: Form $\mathbf{B} = \mathbf{Q}^T\mathbf{A}$

5: Compute SVD of $\mathbf{B} = \tilde{\mathbf{U}}\boldsymbol{\Sigma}\mathbf{V}^T$

6: Set $\mathbf{U} = \mathbf{Q}\tilde{\mathbf{U}}$

This completes the discussion of the algorithms that will prove useful in defining the trial basis $\boldsymbol{\Phi}$ from snapshot data. Chapters 5–6 will provide specific training methods that collect snapshots according to the requirements of the trust region-based optimization algorithm. This section closes with a brief note on the construction of the *mask* $\boldsymbol{P}$ when collocation-based hyperreduction is employed. The sample mesh $\bar{\boldsymbol{P}}$ will not be discussed since it is determined uniquely from the mask,

Algorithm 9 Brand's Algorithm for low-rank SVD updates: appending vector

$$\bar{\boldsymbol{U}}, \bar{\boldsymbol{\Sigma}}, \bar{\boldsymbol{V}} = \texttt{BrandAppendSVD}(\boldsymbol{U}, \boldsymbol{\Sigma}, \boldsymbol{V}, \boldsymbol{Y})$$

Input: Data matrix $\boldsymbol{X} \in \mathbb{R}^{m \times n}$ of rank r, thin SVD of data matrix $\boldsymbol{X} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$, and full-rank matrix of vectors $\boldsymbol{Y} \in \mathbb{R}^{m \times k}$.

Output: SVD of updated data matrix: $\begin{bmatrix}\boldsymbol{X} & \boldsymbol{Y}\end{bmatrix} = \bar{\boldsymbol{U}}\bar{\boldsymbol{\Sigma}}\bar{\boldsymbol{V}}^T$

1: Compute $\boldsymbol{M} = \boldsymbol{U}^T\boldsymbol{Y} \in \mathbb{R}^{r \times k}$

2: Compute $\bar{\boldsymbol{P}} = \boldsymbol{Y} - \boldsymbol{U}\boldsymbol{M} \in \mathbb{R}^{m \times k}$

3: Compute QR decomposition of $\bar{\boldsymbol{P}} = \boldsymbol{P}\boldsymbol{R}_A$, where $\boldsymbol{P} \in \mathbb{R}^{m \times k}$, $\boldsymbol{R}_A \in \mathbb{R}^{k \times k}$

4: Form $\boldsymbol{K} = \begin{bmatrix}\boldsymbol{\Sigma} & \boldsymbol{M} \\ \boldsymbol{0} & \boldsymbol{R}_A\end{bmatrix} \in \mathbb{R}^{(r+k)\times(r+k)}$

5: Compute SVD of $\boldsymbol{K} = \boldsymbol{C}\boldsymbol{S}\boldsymbol{D}^T$, where $\boldsymbol{C}, \boldsymbol{S}, \boldsymbol{D} \in \mathbb{R}^{(r+k)\times(r+k)}$

6:

$$\bar{\boldsymbol{U}} = \begin{bmatrix}\boldsymbol{U} & \boldsymbol{P}\end{bmatrix}\boldsymbol{C} \qquad \bar{\boldsymbol{\Sigma}} = \boldsymbol{S} \qquad \bar{\boldsymbol{V}} = \begin{bmatrix}\boldsymbol{V} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I}\end{bmatrix}\boldsymbol{D}$$

Algorithm 10 Brand's Algorithm for low-rank SVD updates: translating columns

$$\bar{\boldsymbol{U}}, \bar{\boldsymbol{\Sigma}}, \bar{\boldsymbol{V}} = \texttt{BrandTranslateSVD}(\boldsymbol{U}, \boldsymbol{\Sigma}, \boldsymbol{V}, \boldsymbol{a})$$

Input: Data matrix $\boldsymbol{X} \in \mathbb{R}^{m \times n}$ of rank r, thin SVD of data matrix $\boldsymbol{X} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$, and desired translation vector, $\boldsymbol{a} \in \mathbb{R}^m$

Output: SVD of updated data matrix: $\boldsymbol{X} + \boldsymbol{a}\boldsymbol{1}^T = \bar{\boldsymbol{U}}\bar{\boldsymbol{\Sigma}}\bar{\boldsymbol{V}}^T$

1: Compute $\boldsymbol{n} = \boldsymbol{V}^T\boldsymbol{1}$, $\boldsymbol{q} = \boldsymbol{1} - \boldsymbol{V}\boldsymbol{n}$, $q = \|\boldsymbol{q}\|_2$, and $\boldsymbol{Q} = \frac{1}{q}\boldsymbol{q}$

2: Compute $\boldsymbol{m} = \boldsymbol{U}^T\boldsymbol{a} \in \mathbb{R}^r$

3: Compute $\boldsymbol{p} = \boldsymbol{a} - \boldsymbol{U}\boldsymbol{m} \in \mathbb{R}^m$

4: Define $\hat{r} \in \mathbb{R}$ and $\boldsymbol{v} \in \mathbb{R}^N$ such that: $\boldsymbol{p} = \hat{r}\boldsymbol{v}$ where $\|\boldsymbol{v}\|_2 = 1$

5: Form $\boldsymbol{K} = \begin{bmatrix}\boldsymbol{\Sigma} + \boldsymbol{m}\boldsymbol{n}^T & q\boldsymbol{m} \\ \hat{r}\boldsymbol{n} & \hat{r}q\end{bmatrix} \in \mathbb{R}^{(r+1)\times(r+1)}$

6: Compute SVD of $\boldsymbol{K} = \boldsymbol{C}\boldsymbol{S}\boldsymbol{D}^T$, where $\boldsymbol{C}, \boldsymbol{S}, \boldsymbol{D} \in \mathbb{R}^{(r+1)\times(r+1)}$

7:

$$\bar{\boldsymbol{U}} = \begin{bmatrix}\boldsymbol{U} & \boldsymbol{v}\end{bmatrix}\boldsymbol{C} \qquad \bar{\boldsymbol{\Sigma}} = \boldsymbol{S} \qquad \bar{\boldsymbol{V}} = \begin{bmatrix}\boldsymbol{V} & \boldsymbol{Q}\end{bmatrix}\boldsymbol{D}$$

as discussed in Section 4.2.2. The mask is constructed according to the DEIM algorithm introduced in [41] and generalized in [31, 198]. The variant introduced in [198] will be employed in this work due to its proven robustness in handling vector-valued PDE solutions where the variables in each component have different scales and the flexibility afforded by injecting expert knowledge, which proved crucial in large-scale CFD applications [198].

4.4 Summary

With all of the ingredients for efficient and optimal projection-based model reduction introduced in Sections 4.1–4.3, this section provides an overview of the overall framework and makes important connections between the various components that will be leveraged in Chapters 5–6. The general form of projection-based reduced-order models was introduced in (4.3)

$$\text{find } \boldsymbol{u}_r \in \mathbb{R}^{k_{\boldsymbol{u}}} \text{ such that } \boldsymbol{\Psi}^T \boldsymbol{r}(\boldsymbol{\Phi}\boldsymbol{u}_r, \boldsymbol{\mu}) = 0. \tag{4.130}$$

It is uniquely defined by a *trial basis* $\boldsymbol{\Phi}$ and test basis $\boldsymbol{\Psi}$, which may be chosen arbitrarily and independently, in general. In order to ensure the reduced-order model possesses the minimum-residual property, the test and trial basis must be related to one another and the optimality metric $\boldsymbol{\Theta}$ according to (4.14), i.e.,

$$\boldsymbol{\Psi}(\boldsymbol{u}, \boldsymbol{\mu}) = \boldsymbol{\Theta}\frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}(\boldsymbol{u}, \boldsymbol{\mu})\boldsymbol{\Phi}. \tag{4.131}$$

The minimum-residual property is a desirable since it guarantees the approximation generated by the reduced-order model monotonically improves (in terms of the residual norm in some metric) as the trial basis is expanded and exactly reconstructs training data. These properties are known as *monotonicity* and *interpolation* (Proposition 4.1). The interpolation property requires $\boldsymbol{u}(\boldsymbol{\mu}) \in \text{col}(\boldsymbol{\Phi})$, where $\boldsymbol{u}(\boldsymbol{\mu})$ is the solution of $\boldsymbol{r}(\,\cdot\,, \boldsymbol{\mu}) = 0$, which will be guaranteed using the span-preserving variant of POD (`PODSP` in Algorithm 6). However, it will not be efficient or even practical to require the reduced-order model be interpolatory at *every* $\boldsymbol{\mu} \in \mathbb{R}^{N_{\boldsymbol{\mu}}}$. Instead, n interpolation points $\{\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_n\}$ are selected and a snapshot matrix is constructed as

$$\boldsymbol{X} = \begin{bmatrix} \boldsymbol{u}(\boldsymbol{\mu}_1) & \cdots & \boldsymbol{u}(\boldsymbol{\mu}_n) \end{bmatrix}. \tag{4.132}$$

Additionally, let $\boldsymbol{X}'$ be any collection of primal snapshot to be used to construct the trial basis whose columns will not necessarily be preserved in the span of the trial space. The trial basis is then constructed as

$$\boldsymbol{\Phi} = \texttt{PODSP}(\boldsymbol{X}, \boldsymbol{X}'). \tag{4.133}$$

In Chapter 5, $\boldsymbol{X}$ will consist of the high-dimensional snapshot *at the trust region* center, i.e., $\boldsymbol{u}(\boldsymbol{\mu}_k)$, since conditions (3.14) and (3.15) require a prescribed level of accuracy at the center.

With the primal reduced-order model constructed, the sensitivity or adjoint methods introduced

in Sections 4.1.2–4.1.3 can be used to derive the gradients of any reduced quantities of interest. However, it was shown that for general minimum-residual reduced-order models, these computations will require second derivatives of the governing residual $\boldsymbol{r}$, which are expensive to compute and rarely available in large-scale PDE implementations. An alternative that will break discrete consistency, i.e., the computed gradient of the reduced quantity of interest will not match the true gradient of the quantity, is to employ minimum-residual reduced-order models directly for the high-dimensional sensitivity and adjoint equations. Despite breaking discrete consistency, these quantities are computable since they do not require second derivatives of $\boldsymbol{r}$ and also possess the minimum-residual properties of monotonicity and interpolation.

The minimum-residual sensitivities, defined in (4.28), are uniquely defined through the specification of a sensitivity basis $\boldsymbol{\Phi}^\partial$ and optimality metric $\boldsymbol{\Theta}^\partial$, i.e.,

$$\text{find } \widehat{\frac{\partial \boldsymbol{u}_r}{\partial \boldsymbol{\mu}}} \in \mathbb{R}^{k_{\boldsymbol{u}} \times N_{\boldsymbol{\mu}}} \text{ such that } \left(\frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}\boldsymbol{\Phi}^\partial\right)^T \boldsymbol{\Theta}^\partial \left(\frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}\boldsymbol{\Phi}^\partial\right) \widehat{\frac{\partial \boldsymbol{u}_r}{\partial \boldsymbol{\mu}}} = -\left(\frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}\boldsymbol{\Phi}^\partial\right)^T \boldsymbol{\Theta}^\partial \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{\mu}} \tag{4.134}$$

where all terms are evaluated at the (reconstructed) solution of the primal reduced-order model $\boldsymbol{\Phi}\boldsymbol{u}_r(\boldsymbol{\mu}; \boldsymbol{\Phi}, \boldsymbol{\Psi})$. Proposition 4.3 guarantees these two choices for the reduced sensitivities match when the primal solution is exact or the test basis is constant, provided the relationships in (4.35) hold, i.e.,

$$\begin{aligned} \boldsymbol{\Phi}^\partial &= \boldsymbol{\Phi} \\ \boldsymbol{\Psi}(\boldsymbol{u}, \boldsymbol{\mu}) &= \boldsymbol{\Theta}^\partial \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}(\boldsymbol{u}, \boldsymbol{\mu})\boldsymbol{\Phi}^\partial, \end{aligned} \tag{4.135}$$

which will be enforced in the remainder. As with the primal ROM, the minimum-residual property is desirable since it ensures the reduced sensitivity model is monotonic and interpolatory. Interpolation requires $\frac{\partial \boldsymbol{u}}{\partial \boldsymbol{\mu}}(\boldsymbol{\mu}) \in \text{col}(\boldsymbol{\Phi})$, where $\frac{\partial \boldsymbol{u}}{\partial \boldsymbol{\mu}}(\boldsymbol{\mu})$ is the solution of $\boldsymbol{r}^\partial(\boldsymbol{u}(\boldsymbol{\mu}), \cdot, \boldsymbol{\mu}) = 0$ and the requirement $\boldsymbol{\Phi}^\partial = \boldsymbol{\Phi}$ has been imposed. This condition, along with the requirement for interpolation of the primal solution ($\boldsymbol{u}(\boldsymbol{\mu}) \in \text{col}(\boldsymbol{\Phi})$), will be enforced using the heterogeneous span-preserving variant of POD. Define the sensitivity snapshot matrix

$$\boldsymbol{Y} = \begin{bmatrix} \frac{\partial \boldsymbol{u}}{\partial \boldsymbol{\mu}}(\boldsymbol{\mu}_1) & \cdots & \frac{\partial \boldsymbol{u}}{\partial \boldsymbol{\mu}}(\boldsymbol{\mu}_n), \end{bmatrix} \tag{4.136}$$

where $\{\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_n\}$ are the interpolation points previously defined, and let $\boldsymbol{Y}'$ be any other collection of sensitivity snapshots. Then, the trial basis is defined according to

$$\boldsymbol{\Phi} = \texttt{PODHSP}(\boldsymbol{X}, \boldsymbol{X}', \boldsymbol{Y}, \boldsymbol{Y}'). \tag{4.137}$$

This guarantees the primal and sensitivity reduced-order models will be interpolatory if they both possess the minimum-residual property since $\boldsymbol{u}(\boldsymbol{\mu}_i) \in \text{col}(\boldsymbol{\Phi})$ and $\frac{\partial \boldsymbol{u}}{\partial \boldsymbol{\mu}}(\boldsymbol{\mu}_i) \in \text{col}(\boldsymbol{\Phi})$. In Chapter 5, $\boldsymbol{X}$ and $\boldsymbol{Y}$ will consist of the high-dimensional primal and sensitivity snapshots *at the trust region* center, i.e., $\boldsymbol{u}(\boldsymbol{\mu}_k)$ and $\frac{\partial \boldsymbol{u}}{\partial \boldsymbol{\mu}}(\boldsymbol{\mu}_k)$, since conditions (3.14) and (3.15) require a prescribed level of accuracy at

the center.

The minimum-residual adjoint reduced-order model, defined in (4.56), is uniquely defined through the specification of an adjoint basis $\boldsymbol{\Phi}^{\boldsymbol{\lambda}}$ and optimality metric $\boldsymbol{\Theta}^{\boldsymbol{\lambda}}$, i.e.,

$$\text{find } \hat{\boldsymbol{\lambda}}_r \in \mathbb{R}^{k_{\boldsymbol{u}}} \text{ such that } \left(\frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}^T \boldsymbol{\Phi}^{\boldsymbol{\lambda}}\right)^T \boldsymbol{\Theta}^{\boldsymbol{\lambda}} \left(\frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}^T \boldsymbol{\Phi}^{\boldsymbol{\lambda}}\right) \hat{\boldsymbol{\lambda}}_r = \left(\frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}^T \boldsymbol{\Phi}^{\boldsymbol{\lambda}}\right)^T \boldsymbol{\Theta}^{\boldsymbol{\lambda}} \frac{\partial f}{\partial \boldsymbol{u}}^T \tag{4.138}$$

where all terms are evaluated at the (reconstructed) solution of the primal reduced-order model $\boldsymbol{\Phi}\boldsymbol{u}_r(\boldsymbol{\mu}; \boldsymbol{\Phi}, \boldsymbol{\Psi})$. Proposition 4.5 guarantees these two choices for the reduced adjoints match when the primal solution is exact or the test basis is constant, provided the relationships in (4.63) hold, i.e.,

$$\boldsymbol{\Phi}^{\boldsymbol{\lambda}} = \boldsymbol{\Psi} = \left[\boldsymbol{\Theta}^{\boldsymbol{\lambda}} \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}(\boldsymbol{u}, \boldsymbol{\mu})^T\right]^{-1} \boldsymbol{\Phi} \tag{4.139}$$

which will be enforced in the remainder. As with the primal ROM, the minimum-residual property is desirable since it ensures the reduced adjoint model is monotonic and interpolatory. Interpolation requires $\boldsymbol{\lambda}(\boldsymbol{\mu}) \in \text{col}(\boldsymbol{\Psi})$ where $\boldsymbol{\lambda}(\boldsymbol{\mu})$ is the solution of $\boldsymbol{r}^{\boldsymbol{\lambda}}(\boldsymbol{u}(\boldsymbol{\mu}), \cdot, \boldsymbol{\mu}) = 0$ and the requirement $\boldsymbol{\Phi}^{\boldsymbol{\lambda}} = \boldsymbol{\Psi}$ has been imposed. Due to the relationship between $\boldsymbol{\Psi}$, $\boldsymbol{\Phi}^{\boldsymbol{\lambda}}$, $\boldsymbol{\Phi}$, and $\boldsymbol{\Theta}^{\boldsymbol{\lambda}}$ imposed in (4.63) from Proposition 4.5, the following equivalence holds

$$\boldsymbol{\lambda}(\boldsymbol{\mu}) \in \text{col}(\boldsymbol{\Psi}) \iff \boldsymbol{\Theta}^{\boldsymbol{\lambda}} \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}(\boldsymbol{u}(\boldsymbol{\mu}), \boldsymbol{\mu})^T \boldsymbol{\lambda}(\boldsymbol{\mu}) \in \text{col}(\boldsymbol{\Phi}) \tag{4.140}$$

This condition, along with the requirement for interpolation of the primal solution ($\boldsymbol{u}(\boldsymbol{\mu}) \in \text{col}(\boldsymbol{\Phi})$), will be enforced using the heterogeneous span-preserving variant of POD to construct the trial basis $\boldsymbol{\Phi}$. Define the (modified) adjoint snapshot matrix

$$\boldsymbol{Z} = \left[\boldsymbol{\Theta}^{\boldsymbol{\lambda}} \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}(\boldsymbol{u}(\boldsymbol{\mu}_1), \boldsymbol{\mu}_1)^T \boldsymbol{\lambda}(\boldsymbol{\mu}_1) \quad \cdots \quad \boldsymbol{\Theta}^{\boldsymbol{\lambda}} \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}(\boldsymbol{u}(\boldsymbol{\mu}_n), \boldsymbol{\mu}_n)^T \boldsymbol{\lambda}(\boldsymbol{\mu}_n)\right] \tag{4.141}$$

where $\{\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_n\}$ are the interpolation points previously defined, and let $\boldsymbol{Z}'$ be any other collection of (modified) adjoint snapshots. Then, the trial basis is defined according to

$$\boldsymbol{\Phi} = \texttt{PODHSP}(\boldsymbol{X}, \boldsymbol{X}', \boldsymbol{Z}, \boldsymbol{Z}'). \tag{4.142}$$

This guarantees the primal and adjoint reduced-order models will be interpolatory if they both possess the minimum-residual property since $\boldsymbol{u}(\boldsymbol{\mu}_i) \in \text{col}(\boldsymbol{\Phi})$ and $\boldsymbol{\lambda}(\boldsymbol{\mu}_i) \in \text{col}(\boldsymbol{\Psi})$. In Chapter 5, $\boldsymbol{X}$ and $\boldsymbol{Y}$ will consist of the high-dimensional primal and adjoint snapshots *at the trust region* center, i.e., $\boldsymbol{u}(\boldsymbol{\mu}_k)$ and $\boldsymbol{\lambda}(\boldsymbol{\mu}_k)$, since conditions (3.14) and (3.15) require a prescribed level of accuracy at the center.

There may be cases where it is desirable for the reduced-order model to be monotonic and interpolatory in the primal, sensitivity, and adjoint states. The logical extension of the previous development employs a minimum-residual reduced-order model for the primal, sensitivity, and adjoint

and defines the trial basis according to

$$\boldsymbol{\Phi} = \texttt{PODHSP}(\boldsymbol{X}, \boldsymbol{X}', \boldsymbol{Y}, \boldsymbol{Y}', \boldsymbol{Z}, \boldsymbol{Z}') \tag{4.143}$$

where the above is the obvious extension in Algorithm 7 to three types of snapshots.

Before closing this section, the abstract discussion regarding minimum-residual reduced-order models is made concrete by considering the special case of a Galerkin and LSPG projection. Reduced-order models based on a Galerkin projection take the test basis to be the same as the trial basis

$$\boldsymbol{\Psi}(\boldsymbol{u}, \boldsymbol{\mu}) = \boldsymbol{\Phi}$$

for any $\boldsymbol{u} \in \mathbb{R}^{N_{\boldsymbol{u}}}$ and $\boldsymbol{\mu} \in \mathbb{R}^{N_{\boldsymbol{\mu}}}$, rendering the test basis *constant* and immediately qualifying such reduced-order models for the results of Propositions 4.3 and 4.5. Galerkin reduced-order models possess the minimum-residual property in the metric

$$\boldsymbol{\Theta} = \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}(\boldsymbol{\Phi}\boldsymbol{u}_r(\boldsymbol{\mu}; \boldsymbol{\Phi}, \boldsymbol{\Phi}), \boldsymbol{\mu})^{-T}, \tag{4.144}$$

provided the PDE Jacobian is symmetric, positive definite. Given this relation between test and trial basis and requirement that the PDE Jacobian is SPD, the minimum-residual sensitivity and adjoint reduced-order models follow from the choices

$$\boldsymbol{\Phi}^{\partial} = \boldsymbol{\Phi}^{\boldsymbol{\lambda}} = \boldsymbol{\Phi} \qquad \boldsymbol{\Theta}^{\partial} = \boldsymbol{\Theta}^{\boldsymbol{\lambda}} = \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}(\boldsymbol{\Phi}\boldsymbol{u}_r(\boldsymbol{\mu}; \boldsymbol{\Phi}, \boldsymbol{\Phi}), \boldsymbol{\mu})^{-T} \tag{4.145}$$

This relations also ensure (4.35) and (4.63) of Propositions 4.3 and 4.5 are satisfied, which implies the *true* Galerkin sensitivities and adjoint match the minimum-residual counterparts. In contrast, reduced-order models based on the Least-Squares Petrov-Galerkin projection take the test basis according to

$$\boldsymbol{\Psi}(\boldsymbol{u}, \boldsymbol{\mu}) = \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}(\boldsymbol{u}, \boldsymbol{\mu})\boldsymbol{\Phi} \tag{4.146}$$

for any $\boldsymbol{u} \in \mathbb{R}^{N_{\boldsymbol{u}}}$ and $\boldsymbol{\mu} \in \mathbb{R}^{N_{\boldsymbol{\mu}}}$, resulting in a non-constant test basis and the results of Propositions 4.3 and 4.5 will only hold when the primal solution of the reduced-order model is exact, i.e., $\boldsymbol{u}(\boldsymbol{\mu}) = \boldsymbol{\Phi}\boldsymbol{u}_r(\boldsymbol{\mu}; \boldsymbol{\Phi}, \boldsymbol{\Psi})$. The LSPG reduced-order model possesses the minimum-residual property *by construction* in the metric $\boldsymbol{\Theta} = \boldsymbol{I}$ and therefore applies in the most general case, i.e., without requiring SPD Jacobians. Given this relationship between the test and trial basis and the following requirements from Propositions 4.3 and 4.5

$$\boldsymbol{\Phi}^{\partial} = \boldsymbol{\Phi} \qquad \boldsymbol{\Phi}^{\boldsymbol{\lambda}} = \boldsymbol{\Psi}, \tag{4.147}$$

the minimum-residual sensitivity and adjoint reduced-order models for the LSPG projection follow

from the choices

$$\boldsymbol{\Theta}^{\partial} = \boldsymbol{I} \qquad \boldsymbol{\Theta}^{\boldsymbol{\lambda}} = \left[\frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}^T \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}} \right]^{-1}_{(\boldsymbol{\Phi}\boldsymbol{u}_r(\boldsymbol{\mu};\boldsymbol{\Phi},\boldsymbol{\Psi}),\boldsymbol{\mu})} . \tag{4.148}$$

The above relationship satisfy all conditions in Propositions 4.3 and 4.5, thereby ensuring the true and minimum-residual sensitivities and adjoints agree when the primal solution is exact.

Most of the developments detailed in this section extend to the case where collocation-based hyperreduced models are used in place of the pure projection-based reduced-order models. In particular, the relationship between the various bases and optimality metrics, when imposed only on the hyperreduction *mask*, lead to a weaker form of the minimum-residual property, i.e., the masked minimum-residual property (Definition 4.2). This ultimately leads to a weaker form of monotonicity and interpolation that only holds under stricter assumptions on solutions of the discrete PDE. In this work, the mask $\boldsymbol{P}$ is constructed solely from the primal reduced-order basis $\boldsymbol{\Phi}$ and problem-specific information following the approach in [198].

Chapter 5

Optimization via Model Reduction and Residual-Based Trust Regions

With the globally convergent, multifidelity trust region method introduced in Chapter 3 and projection-based reduced-order models introduced in Chapter 4, these technologies are combined to yield an efficient algorithm for deterministic PDE-constrained optimization. The approximation model will be taken as the quantity of interest evaluated at the reconstructed reduced-order model solution and residual-based error bounds (Appendix B) will define the objective and gradient error bounds that are required for global convergence of the multifidelity trust region method. In addition to exploiting inexpensive reduced-order (hyperreduced) models in the trust region subproblem, the flexible multifidelity trust region framework of Section 3.1.1 allows for several other opportunities for efficiency. First, the objective accuracy condition (3.14), restated here for convenience,

$$\vartheta_k(\boldsymbol{\mu}_k) \leq \kappa_\vartheta \Delta_k \qquad \kappa_\vartheta \in (0,\, 1),$$

implies the reduced-order model does not need to be exact at the trust region centers. This is exploited by using *partially converged* solutions to build the reduced-order basis. Similarly, the gradient accuracy condition (3.15)

$$\varphi_k(\boldsymbol{\mu}_k) \leq \kappa_\varphi \min\{\|\nabla m_k(\boldsymbol{\mu}_k)\|\,,\, \Delta_k\} \qquad \kappa_\varphi > 0$$

allows for the use of partially converged sensitivity or adjoint snapshots. Partially converged primal and dual solutions can substantially reduce the burden of collecting snapshots, particularly in large-scale applications encountered in computational fluid dynamics that require slowly converging nonlinear solvers such as pseudo-transient continuation [104, 105] for robust convergence behavior. Partially converged primal solutions are also used to efficiently evaluate the performance of a trust region subproblem using the concepts outlined in Section 3.1.1. Sections 5.2 and 5.3 detail the use

of partially converged solutions for these purposes. While most of this chapter focuses on approximation models based on projection-based reduced-order models without hyperreduction, Section 5.4 discusses the extension to collocation-based hyperreduced models. Finally, Section 5.5 provides several numerical examples from various computational mechanics disciplines, including the large-scale industrial demonstration of shape optimization of a full aircraft configuration, to study the proposed approach.

5.1 Residual-Based Trust Region Method

Consider the fully discrete partial differential equation $\boldsymbol{r}(\boldsymbol{u}, \boldsymbol{\mu}) = 0$, where $\boldsymbol{u} \in \mathbb{R}^{N_{\boldsymbol{u}}}$ is the state vector and $\boldsymbol{\mu} \in \mathbb{R}^{N_{\boldsymbol{\mu}}}$ are the design or control parameters and $f : \mathbb{R}^{N_{\boldsymbol{u}}} \times \mathbb{R}^{N_{\boldsymbol{\mu}}} \to \mathbb{R}$ is a quantity of interest to be optimized. The reduced-space approach to PDE-constrained optimization (Section 2.3) considers the optimization problem

$$\underset{\boldsymbol{\mu} \in \mathbb{R}^{N_{\boldsymbol{\mu}}}}{\text{minimize}} \quad F(\boldsymbol{\mu}), \tag{5.1}$$

where $F(\boldsymbol{\mu}) = f(\boldsymbol{u}(\boldsymbol{\mu}), \boldsymbol{\mu})$ and $\boldsymbol{u}(\boldsymbol{\mu})$ is the solution of $\boldsymbol{r}(\,\cdot\,, \boldsymbol{\mu}) = 0$. Due to the large expense associated with the evaluation of $F(\boldsymbol{\mu})$ and $\nabla F(\boldsymbol{\mu})$, the multifidelity trust region method and projection-based model reduction techniques are combined to efficiently solve (5.1). The multifidelity trust region method of Chapter 3 was completely specified in terms of the approximation model $m_k(\boldsymbol{\mu})$, the objective error indicator $\vartheta_k(\boldsymbol{\mu})$ that satisfies (3.12), the gradient error indicator $\varphi_k(\boldsymbol{\mu})$ that satisfies (3.13), and the inexact objective model $\psi_k(\boldsymbol{\mu})$ and error indicator $\theta_k(\boldsymbol{\mu})$ that satisfy (3.21). Therefore the focus of this section is the specification of these functions using projection-based reduced-order (hyperreduced) models and error indicators from Chapter 4 and Appendix B, respectively. From the overview of the multifidelity trust region method provided in Section 3.1.1, there are two other critical pieces required to fully prescribe the method such that global convergence is guaranteed: a trust region subproblem solver that ensures the fraction of Cauchy decrease (A.9) is obtained and a refinement mechanism for $m_k(\boldsymbol{\mu})$, $\psi_k(\boldsymbol{\mu})$ and the associated error indicators to ensure the error conditions (3.14), (3.15), (3.22) are met. Due to the significant cost separation between reduced-order (hyperreduced) models and the high-dimensional model, the trust region subproblem is solved exactly to guarantee the FCD is satisfied. The error conditions will be met through construction of the reduced-order basis, which will be detailed in Section 5.1.2.

5.1.1 Multifidelity Trust Region Ingredients

At the kth iteration, the approximation model based on projection-based reduced-order models takes the form

$$m_k(\boldsymbol{\mu}) = f(\boldsymbol{\Phi}_k \boldsymbol{u}_r(\boldsymbol{\mu}; \boldsymbol{\Phi}_k, \boldsymbol{\Psi}_k), \boldsymbol{\mu}), \tag{5.2}$$

where $\boldsymbol{\Phi}_k$ is the reduced-order basis used at the kth iteration of the trust region method—details pertaining to the construction of $\boldsymbol{\Phi}_k$ will be deferred to Section 5.1.2 as they will be intimately linked to the error conditions in (3.14), (3.15) and therefore the global convergence theory—and

$\boldsymbol{u}_r(\boldsymbol{\mu};\, \boldsymbol{\Phi}_k,\, \boldsymbol{\Psi}_k)$ is the solution of the reduced-order model

$$\boldsymbol{\Psi}_k^T \boldsymbol{r}(\boldsymbol{\Phi}_k \boldsymbol{u}_r,\, \boldsymbol{\mu}) = 0$$

with test basis $\boldsymbol{\Psi}_k$. The test basis is chosen according to (4.14) to ensure the reduced-order model possesses the minimum-residual property, which in turn ensures it is monotonic and interpolatory. The gradient of the model in (5.2) will be computed using the *exact* reduced sensitivity or adjoint method *if the test basis is constant*. This ensures the gradients can be computed without requiring second derivatives of $\boldsymbol{r}$ and will be consistent with the corresponding function. The minimum-residual variants will be used if the test basis is not constant; however, the requirements between the sensitivity/adjoint basis and optimality metric in (4.35) and (4.63) will be enforced to ensure the minimum-residual gradients match the true gradients at any point where the primal reduced-order model solution is exact. In Section 5.1.2, the reduced-order basis $\boldsymbol{\Phi}_k$ will be constructed such that the primal and sensitivity/adjoint reduced-order model is exact at the trust region center $\boldsymbol{\mu}_k$. Following this discussion, the construction of the trial basis $\boldsymbol{\Phi}_k$ and selection of minimum-residual optimality metrics is sufficient to completely define the remaining ingredients of the reduced-order model, i.e., the test basis $\boldsymbol{\Psi}_k$, sensitivity basis $\boldsymbol{\Phi}_k^{\partial} = \boldsymbol{\Phi}_k$, and adjoint basis $\boldsymbol{\Phi}_k^{\boldsymbol{\lambda}} = \boldsymbol{\Psi}_k$.

There are two natural choices for the trust region constraint function. The first is the standard Euclidean distance

$$\vartheta_k(\boldsymbol{\mu}) \coloneqq \|\boldsymbol{\mu} - \boldsymbol{\mu}_k\|$$

which leads to a traditional trust region algorithm and recovers a method similar to the original Trust Region Proper Orthogonal Decomposition (TRPOD) method [10]. As discussed in Section 3.1.1, this choice automatically satisfies requirements in (3.12) and (3.14), provided a gradient error indicator $\varphi_k(\boldsymbol{\mu})$ is chosen that satisfies (3.13) and (3.15). Another choice for the trust region constraint that was proposed in the author's previous research [210], and earlier in [208] in the context of linear PDEs, is the norm of the residual evaluated at the reconstructed ROM solution

$$\vartheta_k(\boldsymbol{\mu}) \coloneqq \|\boldsymbol{r}(\boldsymbol{\Phi}_k \boldsymbol{u}_r(\boldsymbol{\mu}_k;\, \boldsymbol{\Phi}_k,\, \boldsymbol{\Psi}_k),\, \boldsymbol{\mu}_k)\|_{\boldsymbol{\Theta}_k} + \|\boldsymbol{r}(\boldsymbol{\Phi}_k \boldsymbol{u}_r(\boldsymbol{\mu};\, \boldsymbol{\Phi}_k,\, \boldsymbol{\Psi}_k),\, \boldsymbol{\mu})\|_{\boldsymbol{\Theta}_k}\,, \tag{5.3}$$

which leads to an *error-aware* trust region. With this choice of $\vartheta_k(\boldsymbol{\mu})$, the bound in (3.12) is verified as follows

$$\begin{aligned}
|F(\boldsymbol{\mu}_k) - F(\boldsymbol{\mu}) + m_k(\boldsymbol{\mu}) - m_k(\boldsymbol{\mu}_k)| &\leq |F(\boldsymbol{\mu}_k) - m_k(\boldsymbol{\mu}_k)| + |F(\boldsymbol{\mu}) - m_k(\boldsymbol{\mu})| \\
&\leq \zeta \left(\|\boldsymbol{r}(\boldsymbol{u},\, \boldsymbol{\mu}_k)\| + \|\boldsymbol{r}(\boldsymbol{u},\, \boldsymbol{\mu})\|\right) \\
&\leq \hat{\zeta} \left(\|\boldsymbol{r}(\boldsymbol{u},\, \boldsymbol{\mu}_k)\|_{\boldsymbol{\Theta}_k} + \|\boldsymbol{r}(\boldsymbol{u},\, \boldsymbol{\mu})\|_{\boldsymbol{\Theta}_k}\right) \\
&= \hat{\zeta} \vartheta_k(\boldsymbol{\mu}),
\end{aligned}$$

where $\boldsymbol{u} = \boldsymbol{\Phi}_k \boldsymbol{u}_r(\boldsymbol{\mu}_k;\, \boldsymbol{\Phi}_k,\, \boldsymbol{\Psi}_k)$, $\zeta > 0$ is an arbitrary constant, and $\hat{\zeta} = \left\|\boldsymbol{\Theta}_k^{-1/2}\right\| \zeta$ is a related constant. The second inequality uses Lemma B.4 that bounds errors in quantities of interest by

the primal residual norm and the third inequality invokes the identity $\|\boldsymbol{x}\| = \left\|\boldsymbol{\Theta}_k^{-1/2}\boldsymbol{x}\right\|_{\boldsymbol{\Theta}_k}$ and the triangle inequality (or simply norm equivalence). The function used for the gradient error indicator is also a residual-based quantity, but the specific form depends on whether the sensitivity or adjoint method is employed in the gradient computation.

Remark. *Some of the optimality metrics— $\boldsymbol{\Theta}$, $\boldsymbol{\Theta}^\partial$, $\boldsymbol{\Theta}^{\boldsymbol{\lambda}}$ — introduced in this document are parameter-dependent. This is not an issue in Chapter 4 since the residual minimization problem was only posed over the* state space *for a fixed parameter. In the context of PDE-constrained optimization, the metric must be valid over the entire state and parameter space. Therefore, all parameter-dependent metrics are fixed at the trust region center $\boldsymbol{\mu}_k$. For a Galerkin projection, the optimality metrics become*

$$\begin{aligned}
\boldsymbol{\Theta}_k &= \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}(\boldsymbol{\Phi}_k \boldsymbol{u}_r(\boldsymbol{\mu}_k;\, \boldsymbol{\Phi}_k,\, \boldsymbol{\Phi}_k),\, \boldsymbol{\mu}_k)^{-T} \\
\boldsymbol{\Theta}_k^\partial &= \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}(\boldsymbol{\Phi}_k \boldsymbol{u}_r(\boldsymbol{\mu}_k;\, \boldsymbol{\Phi}_k,\, \boldsymbol{\Phi}_k),\, \boldsymbol{\mu}_k)^{-T} \\
\boldsymbol{\Theta}_k^{\boldsymbol{\lambda}} &= \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}(\boldsymbol{\Phi}_k \boldsymbol{u}_r(\boldsymbol{\mu}_k;\, \boldsymbol{\Phi}_k,\, \boldsymbol{\Phi}_k),\, \boldsymbol{\mu}_k)^{-T}.
\end{aligned}$$

For a LSPG projection, the optimality metrics become

$$\begin{aligned}
\boldsymbol{\Theta}_k &= \boldsymbol{I} \\
\boldsymbol{\Theta}_k^\partial &= \boldsymbol{I} \\
\boldsymbol{\Theta}_k^{\boldsymbol{\lambda}} &= \left[\frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}^T \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}\right]^{-1}_{(\boldsymbol{\Phi}_k \boldsymbol{u}_r(\boldsymbol{\mu}_k;\, \boldsymbol{\Phi}_k,\, \boldsymbol{\Psi}_k),\, \boldsymbol{\mu}_k)}.
\end{aligned}$$

In this work, the minimum-residual sensitivity and adjoint models are used to compute (approximate) gradients of the projection-based reduced-order model that comprises the approximation model. For the sensitivity method, the approximate gradient, denoted $\widehat{\nabla m_k}(\boldsymbol{\mu})$, is computed according to

$$\widehat{\nabla m_k}(\boldsymbol{\mu}) = \boldsymbol{g}^\partial\left(\boldsymbol{u},\, \boldsymbol{\Phi}_k^\partial \widehat{\frac{\partial \boldsymbol{u}_r}{\partial \boldsymbol{\mu}}}(\boldsymbol{\mu};\, \boldsymbol{\Phi}_k^\partial,\, \boldsymbol{\Theta}_k^\partial,\, \boldsymbol{u}),\, \boldsymbol{\mu}\right), \tag{5.4}$$

where $\widehat{\dfrac{\partial \boldsymbol{u}_r}{\partial \boldsymbol{\mu}}}$ is the solution of the minimum-residual sensitivity reduced-order model in (4.28). For the adjoint method, the approximate gradient is computed according to

$$\widehat{\nabla m_k}(\boldsymbol{\mu}) = \boldsymbol{g}^{\boldsymbol{\lambda}}(\boldsymbol{u},\, \boldsymbol{\Phi}_k^{\boldsymbol{\lambda}} \hat{\boldsymbol{\lambda}}(\boldsymbol{\mu};\, \boldsymbol{\Phi}_k^{\boldsymbol{\lambda}},\, \boldsymbol{\Theta}_k^{\boldsymbol{\lambda}},\, \boldsymbol{u}),\, \boldsymbol{\mu}). \tag{5.5}$$

where $\hat{\boldsymbol{\lambda}}_r$ is the solution of the minimum-residual adjoint reduced-order model in (4.56). In both of the above expressions, $\boldsymbol{u} = \boldsymbol{\Phi}_k \boldsymbol{u}_r(\boldsymbol{\mu};\, \boldsymbol{\Phi}_k,\, \boldsymbol{\Psi}_k)$ is the solution of the primal reduced-order model. The relationships in (4.35) and (4.63) between $\boldsymbol{\Phi}$, $\boldsymbol{\Psi}$, $\boldsymbol{\Phi}^\partial$, $\boldsymbol{\Phi}^{\boldsymbol{\lambda}}$, $\boldsymbol{\Theta}_k^\partial$, and $\boldsymbol{\Theta}_k^{\boldsymbol{\lambda}}$ are employed to guarantee, by Propositions 4.3 and 4.5, that $\widehat{\nabla m_k}(\boldsymbol{\mu}) = \nabla m_k(\boldsymbol{\mu})$ whenever $\boldsymbol{\Psi}_k$ is constant, i.e., a Galerkin projection, or the primal solution is exact. Unfortunately, unless one of these criteria is satisfied $\widehat{\nabla m_k}(\boldsymbol{\mu}) \neq \nabla m_k(\boldsymbol{\mu})$, which may cause convergence issues for the trust region subproblem. To ensure

these subproblems terminate, a maximum number of iterations is imposed on the solver, which may slightly slow convergence of the overall trust region method. Once the optimization algorithm reaches the vicinity of a local minima (of $F(\boldsymbol{\mu})$), the reduced-order model is sufficiently accurate at (and near) the new trust region center and the primal reduced-order model solution will be sufficiently accurate that the two gradients closely match.

For sensitivity-based gradient computations, the gradient error indicator is

$$\varphi_k(\boldsymbol{\mu}) \coloneqq \alpha_1 \left\|\boldsymbol{r}(\boldsymbol{u}, \boldsymbol{\mu})\right\|_{\boldsymbol{\Theta}_k} + \alpha_2 \left\| \boldsymbol{r}^{\partial}\left(\boldsymbol{u}, \boldsymbol{\Phi}_k^{\partial}\widehat{\frac{\partial \boldsymbol{u}_r}{\partial \boldsymbol{\mu}}}(\boldsymbol{\mu}; \boldsymbol{\Phi}_k^{\partial}, \boldsymbol{\Theta}_k^{\partial}, \boldsymbol{u}), \boldsymbol{\mu}\right)\right\|_{\boldsymbol{\Theta}_k^{\partial}} \tag{5.6}$$

where $\boldsymbol{u} = \boldsymbol{\Phi}_k \boldsymbol{u}_r(\boldsymbol{\mu}; \boldsymbol{\Phi}_k, \boldsymbol{\Psi}_k)$ is the reconstructed primal solution and $\widehat{\frac{\partial \boldsymbol{u}_r}{\partial \boldsymbol{\mu}}}$ is the solution of the minimum-residual reduced sensitivity equations (4.28). The primal trial basis $\boldsymbol{\Phi}_k$ is also used as the sensitivity basis $\boldsymbol{\Phi}_k^{\partial}$ to ensure the minimum-residual sensitivities agree with the true reduced-order model sensitivities when the primal solution is exact or $\boldsymbol{\Psi}_k$ is constant (Proposition 4.3). For adjoint-based gradient computations, the gradient error indicator is

$$\varphi_k(\boldsymbol{\mu}) \coloneqq \alpha_1 \left\|\boldsymbol{r}(\boldsymbol{u}, \boldsymbol{\mu})\right\|_{\boldsymbol{\Theta}_k} + \alpha_2 \left\| \boldsymbol{r}^{\boldsymbol{\lambda}}\left(\boldsymbol{u}, \boldsymbol{\Phi}_k^{\boldsymbol{\lambda}}\hat{\boldsymbol{\lambda}}_r(\boldsymbol{\mu}; \boldsymbol{\Phi}_k^{\boldsymbol{\lambda}}, \boldsymbol{\Theta}_k^{\boldsymbol{\lambda}}, \boldsymbol{u}), \boldsymbol{\mu}\right)\right\|_{\boldsymbol{\Theta}_k^{\boldsymbol{\lambda}}} \tag{5.7}$$

where $\boldsymbol{u} = \boldsymbol{\Phi}_k \boldsymbol{u}_r(\boldsymbol{\mu}; \boldsymbol{\Phi}_k, \boldsymbol{\Psi}_k)$ is the reconstructed primal solution and $\hat{\boldsymbol{\lambda}}_r$ is the solution of the minimum-residual reduced adjoint equations (4.56). The primal test basis $\boldsymbol{\Psi}_k$ is used as the adjoint basis $\boldsymbol{\Phi}_k^{\boldsymbol{\lambda}}$ to ensure the minimum-residual adjoints agree with the true reduced-order model adjoints when the primal solution is exact or $\boldsymbol{\Psi}_k$ is constant (Proposition 4.5). In (5.6) and (5.7), α_1, $\alpha_2 > 0$ are user-defined constants intended to balance the contribution of the primal and dual residuals. From Lemma B.7 and B.8, there exists a constant $\xi > 0$ such that

$$\left\|\nabla F(\boldsymbol{\mu}_k) - \widehat{\nabla m_k}(\boldsymbol{\mu}_k)\right\| \leq \xi \varphi_k(\boldsymbol{\mu}_k), \tag{5.8}$$

holds regardless of the values of α_1 and α_2 (provided they are positive) for both the sensitivity and adjoint form of the gradient error indicator. An error bound of this form is a critical ingredient in the global convergence theory of the proposed trust region method, as well as in related methods [93, 108].

Remark. *The objective decrease condition (3.14) introduced in the proposed generalized trust region method is considerably weaker than the conditions required for previous methods. The work by Alexander introduced a trust region framework to manage the use of general approximation models to solve constrained and unconstrained optimization problems with expensive optimization functionals [4, 6, 5]. The trust region model management framework required the approximation model possess* first-order consistency *at trust region centers*

$$m_k(\boldsymbol{\mu}_k) = F(\boldsymbol{\mu}_k) \qquad \nabla m_k(\boldsymbol{\mu}_k) = \nabla F(\boldsymbol{\mu}_k). \tag{5.9}$$

The Trust Region Proper Orthogonal Decomposition (TRPOD) method introduced in [10] and studied extensively thereafter [57, 170, 186], removed the zeroth-order condition entirely and weakened the first-order condition by replacing it with the Carter condition (see discussion to follow). Unlike the present work, TRPOD strictly employed a tradition trust region constraint of the form $\|\boldsymbol{\mu} - \boldsymbol{\mu}_k\| \leq \Delta_k$. *The work in [208] generalized the TRPOD method to an error-aware trust region that required a pointwise error bound on the objective at the trust region center (3.8), which is a considerably stronger requirement than the objective decrease condition (3.14), as discussed in Chapter 3.*

Remark. *The gradient condition (3.15) leveraged in the proposed generalized trust region method was originally proposed in [93] and extensively used in [108, 109, 92, 166]. It is substantially more flexible than the Carter condition [35]*

$$\|\nabla F(\boldsymbol{\mu}_k) - \nabla m_k(\boldsymbol{\mu}_k)\| \leq \eta \|\nabla m_k(\boldsymbol{\mu}_k)\| \qquad \eta \in (0,\, 1) \tag{5.10}$$

that was used in the original TRPOD method [10] and a related method proposed that uses generalized trust regions [208]. Global convergence is predicated on construction of a model that satisfies this bound with any value of η that satisfies $0 < \eta < 1$. *Since global convergence relies critically on value of* η *being in this range, it does not permit the use of error indicators since they are only bounds when multiplied by an* arbitrary constant. *Therefore,* $\nabla F(\boldsymbol{\mu}_k)$ *must be computed along with* $\nabla m_k(\boldsymbol{\mu}_k)$ *corresponding to an increasingly refined basis until (5.10) is met.*

An opportunity for efficiency afforded by the flexible trust region framework introduced in Section 3.1.1 is the use of an approximation model to compute the ratio of actual-to-predicted ratio, ρ_k. The true expression for ρ_k can be replaced with

$$\rho_k = \frac{\psi(\boldsymbol{\mu}_k) - \psi(\hat{\boldsymbol{\mu}}_k)}{m_k(\boldsymbol{\mu}_k) - m_k(\hat{\boldsymbol{\mu}}_k)} \tag{5.11}$$

where $\psi_k : \mathbb{R}^{N_\mu} \to \mathbb{R}$ is an approximation model that satisfies

$$\begin{aligned} |F(\boldsymbol{\mu}_k) - F(\hat{\boldsymbol{\mu}}_k) + \psi_k(\hat{\boldsymbol{\mu}}_k) - \psi_k(\boldsymbol{\mu}_k)| &\leq \sigma\theta_k(\hat{\boldsymbol{\mu}}_k) \\ \theta_k^\omega(\hat{\boldsymbol{\mu}}_k) &\leq \eta \min\{m_k(\boldsymbol{\mu}_k) - m_k(\hat{\boldsymbol{\mu}}_k),\, r_k\}, \end{aligned} \tag{5.12}$$

without destroying global convergence of the overall algorithm. In (5.12), $\sigma > 0$ is an arbitrary constant, $r_k \to 0$ is a forcing sequence, $\eta < \min\{\eta_1,\, 1 - \eta_2\}$, and $0 < \eta_1 < \eta_2 < 1$ and $\omega \in (0,\, 1)$ are algorithmic constant. The error bound in (5.12) is identical to the required relationship between $m_k(\boldsymbol{\mu})$ and $\vartheta_k(\boldsymbol{\mu})$ in (3.12). Thus, a natural and efficient choice is

$$\psi_k(\boldsymbol{\mu}) = m_k(\boldsymbol{\mu}) \qquad \text{and} \qquad \theta_k(\boldsymbol{\mu}) = \vartheta_k(\boldsymbol{\mu}), \tag{5.13}$$

which implies

$$\rho_k = \frac{\psi(\boldsymbol{\mu}_k) - \psi(\hat{\boldsymbol{\mu}}_k)}{m_k(\boldsymbol{\mu}_k) - m_k(\hat{\boldsymbol{\mu}}_k)} = \frac{m_k(\boldsymbol{\mu}_k) - m_k(\hat{\boldsymbol{\mu}}_k)}{m_k(\boldsymbol{\mu}_k) - m_k(\hat{\boldsymbol{\mu}}_k)} = 1. \tag{5.14}$$

Therefore, if the error condition

$$\vartheta_k^\omega(\hat{\boldsymbol{\mu}}_k) = \theta_k^\omega(\hat{\boldsymbol{\mu}}_k) \leq \eta \min\{m_k(\boldsymbol{\mu}_k) - m_k(\hat{\boldsymbol{\mu}}_k),\, r_k\} \tag{5.15}$$

is satisfied for the reduced-order basis used during the kth iteration, the step can automatically be accepted and trust region radius increased *without referring to the high-dimensional model*. If this condition is not satisfied, the exact expression for ratio of actual-to-predicted reduction (3.9) is used, i.e., $\psi_k(\boldsymbol{\mu}) = F(\boldsymbol{\mu})$ and $\theta_k(\boldsymbol{\mu}) = 0$. Section 5.3 introduces another choice for ψ_k and θ_k that leverages *partially converged solutions* for enhanced efficiency.

The choice of objective and gradient error indicators in (5.3) and (5.6), (5.7) provides a strong connection to the minimum-residual theory of Chapter 4 since the norms are taken to exactly coincide with the optimality metrics defining the minimum-residual reduced-order model. As a result, the optimality property and monotonicity hold (Propositions 4.1, 4.2, 4.4). Optimality implies that

$$\vartheta_k(\boldsymbol{\mu}_k) \leq 2 \left\| \boldsymbol{r}(\boldsymbol{\Phi}_k \boldsymbol{x},\, \boldsymbol{\mu}_k) \right\|_{\boldsymbol{\Theta}_k} \tag{5.16}$$

for any $\boldsymbol{x} \in \mathbb{R}^{k_{\boldsymbol{u}}}$. A similar statement holds for $\varphi_k(\boldsymbol{\mu})$

$$\begin{aligned}
\varphi_k(\boldsymbol{\mu}_k) &\leq \alpha_1 \left\| \boldsymbol{r}(\boldsymbol{\Phi}_k \boldsymbol{y},\, \boldsymbol{\mu}_k) \right\|_{\boldsymbol{\Theta}_k} + \alpha_2 \left\| \boldsymbol{r}^\partial \left(\boldsymbol{\Phi}_k \boldsymbol{u}_r(\boldsymbol{\mu}_k;\, \boldsymbol{\Phi}_k,\, \boldsymbol{\Psi}_k),\, \boldsymbol{\Phi}_k^\partial \boldsymbol{w},\, \boldsymbol{\mu}_k \right) \right\|_{\boldsymbol{\Theta}_k^\partial} \\
\varphi_k(\boldsymbol{\mu}_k) &\leq \alpha_1 \left\| \boldsymbol{r}(\boldsymbol{\Phi}_k \boldsymbol{y},\, \boldsymbol{\mu}_k) \right\|_{\boldsymbol{\Theta}_k} + \alpha_2 \left\| \boldsymbol{r}^{\boldsymbol{\lambda}} \left(\boldsymbol{\Phi}_k \boldsymbol{u}_r(\boldsymbol{\mu}_k;\, \boldsymbol{\Phi}_k,\, \boldsymbol{\Psi}_k),\, \boldsymbol{\Phi}_k^{\boldsymbol{\lambda}} \boldsymbol{z},\, \boldsymbol{\mu}_k \right) \right\|_{\boldsymbol{\Theta}_k^{\boldsymbol{\lambda}}}
\end{aligned} \tag{5.17}$$

for any $\boldsymbol{y},\, \boldsymbol{z} \in \mathbb{R}^{k_{\boldsymbol{u}}}$ and $\boldsymbol{w} \in \mathbb{R}^{k_{\boldsymbol{u}} \times N_{\boldsymbol{\mu}}}$. Notice that this bound requires the sensitivity and adjoint residual to be defined (linearized) about the primal reduced-order model solution, i.e., $\boldsymbol{\Phi}_k \boldsymbol{u}_r^k(\boldsymbol{\mu})$. Monotonicity means that hierarchically refining $\boldsymbol{\Phi}_k$ can only reduce $\vartheta(\boldsymbol{\mu})$, provided $\boldsymbol{\Theta}$ is independent of $\boldsymbol{\Phi}_k$. The same statement does not hold for $\varphi_k(\boldsymbol{\mu})$ since monotonicity, as defined in Proposition 4.2, 4.4 requires linearization about a *fixed* primal solution and hierarchically refining either $\boldsymbol{\Phi}_k^\partial$ or $\boldsymbol{\Phi}_k^{\boldsymbol{\lambda}}$. Given the relation between $\boldsymbol{\Phi}_k$, $\boldsymbol{\Phi}_k^\partial$, and $\boldsymbol{\Phi}_k^{\boldsymbol{\lambda}}$ in (4.35) and (4.63), it is impossible to refine $\boldsymbol{\Phi}_k^\partial$ or $\boldsymbol{\Phi}_k^{\boldsymbol{\lambda}}$ without also modifying $\boldsymbol{\Phi}_k$ and therefore changing the primal reduced-order model solution (the linearization point). For these reasons, the choice of $\vartheta_k(\boldsymbol{\mu})$ and $\varphi_k(\boldsymbol{\mu})$ in (5.3) and (5.6), (5.7) are highly desirable. On the other hand, these norms may be difficult to compute if the metric requires computation of the Jacobian of $\boldsymbol{r}$ or its inverse, which will be the case for Galerkin reduced-order models. In such cases, it is desirable to simply use the $\boldsymbol{I}$-norm to define all terms in $\vartheta_k(\boldsymbol{\mu})$ and $\varphi_k(\boldsymbol{\mu})$, i.e.,

$$\begin{aligned}
\vartheta_k(\boldsymbol{\mu}) &\coloneqq \left\| \boldsymbol{r}(\boldsymbol{\Phi}_k \boldsymbol{u}_r(\boldsymbol{\mu}_k;\, \boldsymbol{\Phi}_k,\, \boldsymbol{\Psi}_k),\, \boldsymbol{\mu}_k) \right\| + \left\| \boldsymbol{r}(\boldsymbol{\Phi}_k \boldsymbol{u}_r(\boldsymbol{\mu};\, \boldsymbol{\Phi}_k,\, \boldsymbol{\Psi}_k),\, \boldsymbol{\mu}) \right\| \\
\varphi_k(\boldsymbol{\mu}) &\coloneqq \alpha_1 \left\| \boldsymbol{r}(\boldsymbol{\Phi}_k \boldsymbol{u}_r(\boldsymbol{\mu};\, \boldsymbol{\Phi}_k,\, \boldsymbol{\Psi}_k),\, \boldsymbol{\mu}) \right\| + \alpha_2 \left\| \boldsymbol{r}^\partial \left(\boldsymbol{\Phi}_k \boldsymbol{u}_r(\boldsymbol{\mu};\, \boldsymbol{\Phi}_k,\, \boldsymbol{\Psi}),\, \boldsymbol{\Phi}_k \frac{\partial \boldsymbol{u}_r}{\partial \boldsymbol{\mu}}(\boldsymbol{\mu};\, \boldsymbol{\Phi}_k,\, \boldsymbol{\Psi}_k),\, \boldsymbol{\mu} \right) \right\| \\
\varphi_k(\boldsymbol{\mu}) &\coloneqq \alpha_1 \left\| \boldsymbol{r}(\boldsymbol{\Phi}_k \boldsymbol{u}_r(\boldsymbol{\mu};\, \boldsymbol{\Phi}_k,\, \boldsymbol{\Psi}_k),\, \boldsymbol{\mu}) \right\| + \alpha_2 \left\| \boldsymbol{r}^{\boldsymbol{\lambda}} \left(\boldsymbol{\Phi}_k \boldsymbol{u}_r(\boldsymbol{\mu};\, \boldsymbol{\Phi}_k,\, \boldsymbol{\Psi}_k),\, \boldsymbol{\Psi}_k \boldsymbol{\lambda}_r(\boldsymbol{\mu};\, \boldsymbol{\Phi}_k,\, \boldsymbol{\Psi}_k),\, \boldsymbol{\mu} \right) \right\|
\end{aligned} \tag{5.18}$$

since the norms are trivial to evaluate given the corresponding residual. Fortunately for the case

of LSPG-based reduced-order models, the primal and sensitivity optimality metrics are taken as identity (Sections 4.1.1 and 4.1.2) and the definitions of $\vartheta_k(\boldsymbol{\mu})$ and $\varphi_k(\boldsymbol{\mu})$ in (5.3)-(5.7) and (5.18) agree. However, even when LSPG is used, the adjoint optimality metric is not the identity matrix (Section 4.1.3). Furthermore, note that the interpolation property of minimum-residual reduced-order models does not depend on the metric used in the residual norm, due to the equivalence of norms in finite dimensions, i.e., for the definition of $\vartheta_k(\boldsymbol{\mu})$ in (5.3) or (5.18), $\vartheta_k(\boldsymbol{\mu}) = 0$ if $\boldsymbol{u}(\boldsymbol{\mu}) \in \text{col}(\boldsymbol{\Phi}_k)$, and similarly for $\varphi_k(\boldsymbol{\mu})$. This implies that the choice of $\vartheta_k(\boldsymbol{\mu})$ and $\varphi_k(\boldsymbol{\mu})$ still possesses this critical property that will be used in the next section. Finally, due to the equivalence of norms in finite dimensions, the bounds in (3.12) and (3.13) will still hold (with different constants) and therefore the choice of $\vartheta_k(\boldsymbol{\mu})$ and $\varphi_k(\boldsymbol{\mu})$ in (5.18) will not destroy global convergence.

5.1.2 Basis Construction via Proper Orthogonal Decomposition and the Method of Snapshots

The use of reduced-order models in the context of optimization has predominantly employed an offline-online procedure [17, 149, 173] where expensive operations involving the HDM are performed in the offline phase to build the reduced-order basis $\boldsymbol{\Phi}$, i.e., train the reduced-order model, and the inexpensive reduced-order model is employed in the online optimization phase. A number of drawbacks to this approach exist, the most critical ones being that global convergence can only be established for relatively simple partial differential equations and it is difficult to train a robust ROM in a high-dimensional parameter space. TRPOD [10] was among the first methods to break the offline-online barrier and guarantee global convergence in a general setting. In TRPOD and the many variants to follow [57, 1, 186], the reduced-order basis is constructed during the optimization procedure such that conditions on the objective and gradient accuracy at trust region centers are met, thereby avoiding the issue of sampling in possibly high-dimensional parameter spaces. This is also the approach taken here.

For the remainder of this chapter, only the residual-based constraint function is considered. From the previous section, the choice of the reduced-order model approximation $m_k(\boldsymbol{\mu})$ and residual-based error indicators $\vartheta_k(\boldsymbol{\mu})$ and $\varphi_k(\boldsymbol{\mu})$ satisfy the error bounds in (3.12), (3.13), regardless of the choice of reduced-order basis. However, the accuracy criterion in (3.14) and (3.15)

$$\begin{aligned} \vartheta_k(\boldsymbol{\mu}_k) &\leq \kappa_\vartheta \Delta_k \\ \varphi_k(\boldsymbol{\mu}_k) &\leq \kappa_\varphi \min\{\|\nabla m_k(\boldsymbol{\mu}_k)\|, \Delta_k\} \end{aligned} \tag{5.19}$$

depend critically on the choice of reduced-order basis.

At each iteration k, the reduced-order bases $\boldsymbol{\Phi}_k$, $\boldsymbol{\Phi}_k^\partial$, $\boldsymbol{\Phi}_k^{\boldsymbol{\lambda}}$ are constructed to ensure

$$\boldsymbol{u}(\boldsymbol{\mu}_k) \in \text{col}(\boldsymbol{\Phi}_k) \qquad \frac{\partial \boldsymbol{u}}{\partial \boldsymbol{\mu}}(\boldsymbol{\mu}_k) \in \text{col}(\boldsymbol{\Phi}_k^\partial) \qquad \boldsymbol{\lambda}(\boldsymbol{\mu}_k) \in \text{col}(\boldsymbol{\Phi}_k^{\boldsymbol{\lambda}}). \tag{5.20}$$

The interpolation property of minimum-residual reduced-order models ensures the reconstructed solutions exactly recover the high-dimensional counterparts. In turn, this ensures $\vartheta_k(\boldsymbol{\mu}_k) = \varphi_k(\boldsymbol{\mu}_k) = 0$ and therefore the error conditions (3.14), (3.15) are trivially satisfied and global convergence is guaranteed. As discussed in the previous section, the sensitivity and adjoint bases are chosen in accordance with Proposition 4.3 and 4.5, i.e., $\boldsymbol{\Phi}_k^{\partial} = \boldsymbol{\Phi}_k$ and $\boldsymbol{\Phi}_k^{\boldsymbol{\lambda}} = \boldsymbol{\Psi}_k$, and the requirements in (5.20) reduce to

$$\boldsymbol{u}(\boldsymbol{\mu}_k) \in \operatorname{col}(\boldsymbol{\Phi}_k) \qquad \frac{\partial \boldsymbol{u}}{\partial \boldsymbol{\mu}}(\boldsymbol{\mu}_k) \in \operatorname{col}(\boldsymbol{\Phi}_k) \qquad \boldsymbol{\lambda}(\boldsymbol{\mu}_k) \in \operatorname{col}(\boldsymbol{\Psi}_k). \tag{5.21}$$

The condition between the test basis $\boldsymbol{\Psi}_k$ and adjoint optimality metric $\boldsymbol{\Theta}^{\boldsymbol{\lambda}}$ required in Proposition 4.5 reduces (5.21) to

$$\boldsymbol{u}(\boldsymbol{\mu}_k) \in \operatorname{col}(\boldsymbol{\Phi}_k) \qquad \frac{\partial \boldsymbol{u}}{\partial \boldsymbol{\mu}}(\boldsymbol{\mu}_k) \in \operatorname{col}(\boldsymbol{\Phi}_k) \qquad \boldsymbol{\Theta}^{\boldsymbol{\lambda}}(\boldsymbol{u}(\boldsymbol{\mu}_k),\, \boldsymbol{\mu}_k) \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}(\boldsymbol{u}(\boldsymbol{\mu}_k),\, \boldsymbol{\mu}_k)^T \boldsymbol{\lambda}(\boldsymbol{\mu}_k) \in \operatorname{col}(\boldsymbol{\Phi}_k). \tag{5.22}$$

Remark. *In the case of a Galerkin projection (for problems with SPD Jacobians) with adjoint optimality metric* $\boldsymbol{\Theta}^{\boldsymbol{\lambda}} = \dfrac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}(\boldsymbol{\Phi}\boldsymbol{u}_r(\boldsymbol{\mu};\, \boldsymbol{\Phi},\, \boldsymbol{\Phi}))^{-T}$*, the adjoint snapshots reduce to*

$$\boldsymbol{\Theta}^{\boldsymbol{\lambda}}(\boldsymbol{u}(\boldsymbol{\mu}),\, \boldsymbol{\mu}) \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}(\boldsymbol{u}(\boldsymbol{\mu}),\, \boldsymbol{\mu})^T \boldsymbol{\lambda}(\boldsymbol{\mu}) = \boldsymbol{\lambda}(\boldsymbol{\mu}).$$

In the case of a LSPG projection with adjoint optimality metric $\boldsymbol{\Theta}^{\boldsymbol{\lambda}} = \left[\dfrac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}^T \dfrac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}} \right]^{-1}_{(\boldsymbol{\Phi}\boldsymbol{u}_r(\boldsymbol{\mu};\, \boldsymbol{\Phi},\, \boldsymbol{\Psi}),\, \boldsymbol{\mu})}$*, the adjoint snapshots reduce to*

$$\boldsymbol{\Theta}^{\boldsymbol{\lambda}}(\boldsymbol{u}(\boldsymbol{\mu}),\, \boldsymbol{\mu}) \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}(\boldsymbol{u}(\boldsymbol{\mu}),\, \boldsymbol{\mu})^T \boldsymbol{\lambda}(\boldsymbol{\mu}) = \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}(\boldsymbol{u}(\boldsymbol{\mu}),\, \boldsymbol{\mu})^{-1} \boldsymbol{\lambda}(\boldsymbol{\mu}).$$

The above requirements reveal the nature of the snapshots that should be used in the construction of the trial basis $\boldsymbol{\Phi}_k$. In practice, sensitivities and adjoints are rarely required simultaneously. Usually the sensitivity method is employed when the number of constraints is larger than the number of optimization variables and vice versa for the adjoint method. To generalize the notation such that the sensitivity method and adjoint method can be considered simultaneously, define $\boldsymbol{v}(\boldsymbol{\mu})$ as the sensitivity or adjoint state, depending on which method is used to compute gradients of quantities of interest, i.e.,

$$\boldsymbol{v}(\boldsymbol{\mu}) = \begin{cases} \dfrac{\partial \boldsymbol{u}}{\partial \boldsymbol{\mu}}(\boldsymbol{\mu}) & \text{sensitivity method} \\ \boldsymbol{\lambda}(\boldsymbol{\mu}) & \text{adjoint method} \end{cases} \tag{5.23}$$

and let $\hat{\boldsymbol{v}}(\boldsymbol{\mu})$ denote the corresponding snapshot, i.e.,

$$\hat{\boldsymbol{v}}(\boldsymbol{\mu}) = \begin{cases} \dfrac{\partial \boldsymbol{u}}{\partial \boldsymbol{\mu}}(\boldsymbol{\mu}) & \text{sensitivity method} \\ \boldsymbol{\Theta}^{\boldsymbol{\lambda}}(\boldsymbol{u}(\boldsymbol{\mu}),\, \boldsymbol{\mu}) \dfrac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}(\boldsymbol{u}(\boldsymbol{\mu}),\, \boldsymbol{\mu})^T \boldsymbol{\lambda}(\boldsymbol{\mu}) & \text{adjoint method.} \end{cases} \tag{5.24}$$

With this notation, the requirements in (5.22) are weakened to

$$\boldsymbol{u}(\boldsymbol{\mu}_k) \in \operatorname{col}(\boldsymbol{\Phi}_k) \qquad \hat{\boldsymbol{v}}(\boldsymbol{\mu}_k) \in \operatorname{col}(\boldsymbol{\Phi}_k) \tag{5.25}$$

while still guaranteeing $\vartheta_k(\boldsymbol{\mu}_k) = \varphi_k(\boldsymbol{\mu}_k) = 0$ where it is understood that $\varphi_k(\boldsymbol{\mu})$ corresponds to (5.6) if the sensitivity method is employed and (5.7) for the adjoint method. The conditions in (3.14) and (3.15) will be guaranteed using the heterogeneous span-preserving variant of POD (Section 4.3). Define snapshot matrices at iteration k consisting of $\boldsymbol{u}(\boldsymbol{\mu})$ and $\hat{\boldsymbol{v}}(\boldsymbol{\mu})$ at the trust region centers of all *previous iterations*, i.e.,

$$\begin{aligned} \mathbf{U}_k &= \begin{bmatrix} \boldsymbol{u}(\boldsymbol{\mu}_0) & \cdots & \boldsymbol{u}(\boldsymbol{\mu}_{k-1}) \end{bmatrix} \\ \hat{\mathbf{V}}_k &= \begin{bmatrix} \hat{\boldsymbol{v}}(\boldsymbol{\mu}_0) & \cdots & \hat{\boldsymbol{v}}(\boldsymbol{\mu}_{k-1}) \end{bmatrix}. \end{aligned} \tag{5.26}$$

and define the reduced-order basis as

$$\boldsymbol{\Phi}_k = \texttt{PODHSP}(\boldsymbol{u}(\boldsymbol{\mu}_k),\, \mathbf{U}_k,\, \hat{\boldsymbol{v}}(\boldsymbol{\mu}_k),\, \hat{\mathbf{V}}_k). \tag{5.27}$$

where `PODHSP` is defined in Algorithm 7. By construction, the conditions in (3.14) and (3.15) are satisfied since $\boldsymbol{u}(\boldsymbol{\mu}_k)$ and $\hat{\boldsymbol{v}}(\boldsymbol{\mu}_k)$ are preserved in the columnspace of $\boldsymbol{\Phi}_k$, which implies $\vartheta_k(\boldsymbol{\mu}_k) = \varphi_k(\boldsymbol{\mu}_k) = 0$ and global convergence is guaranteed. Even though the in the information in $\mathbf{U}_k$ and $\mathbf{V}_k$ is not necessarily useful in satisfying the trust region error conditions, i.e., the information in $\boldsymbol{u}(\boldsymbol{\mu}_k)$ and $\boldsymbol{v}(\boldsymbol{\mu}_k)$ is sufficient to do so, it provides the reduced-order model with additional fidelity, which is useful in improving its robustness away from $\boldsymbol{\mu}_k$.

Remark. *There may be instances where the reduced-order basis defined at iteration $k-1$ is sufficient to satisfy the error conditions (3.14), (3.15) at iteration k, i.e.,*

$$\begin{aligned} &\vartheta_{k-1}(\boldsymbol{\mu}_k) \leq \kappa_\vartheta \Delta_k \\ &\varphi_{k-1}(\boldsymbol{\mu}_k) \leq \kappa_\varphi \min\{\|\nabla m_{k-1}(\boldsymbol{\mu}_k)\|,\, \Delta_k\}. \end{aligned} \tag{5.28}$$

This is likely to occur when the initial trust region radius Δ_0 is chosen too small. In this situation, there is no need to update the reduced-order basis and the same model and error indicators are used, i.e.,

$$m_k(\boldsymbol{\mu}) \coloneqq m_{k-1}(\boldsymbol{\mu}) \qquad \vartheta_k(\boldsymbol{\mu}) \coloneqq \vartheta_{k-1}(\boldsymbol{\mu}) \qquad \varphi_k(\boldsymbol{\mu}) \coloneqq \varphi_{k-1}(\boldsymbol{\mu}). \tag{5.29}$$

This choice saves queries to the expensive high-dimensional model and still guarantees global convergence when (5.28) is satisfied.

As written, the above approach to compute $\boldsymbol{\Phi}_k$ requires the singular value decomposition of the snapshot matrices $\mathbf{U}_k$ and $\mathbf{V}_k$ that have an increasing number of columns. This quickly becomes prohibitively expensive since the cost of the SVD scales quadratically in the number of columns [75]. However, the snapshot matrices satisfy the simple relation

$$\begin{aligned} \mathbf{U}_k &= \begin{bmatrix} \mathbf{U}_{k-1} & \boldsymbol{u}(\boldsymbol{\mu}_{k-1}) \end{bmatrix} \\ \hat{\mathbf{V}}_k &= \begin{bmatrix} \hat{\mathbf{V}}_{k-1} & \hat{\boldsymbol{v}}(\boldsymbol{\mu}_{k-1}) \end{bmatrix}, \end{aligned} \tag{5.30}$$

and therefore the thin SVD updates in Algorithm 9 can be used to compute the SVD of $\mathbf{U}_k$ from the SVD of $\mathbf{U}_{k-1}$. Only the QR decomposition of the compressed snapshot matrices must be recomputed at each iteration.

Remark. *For time-dependent problems, exact preservation of $\boldsymbol{u}(\boldsymbol{\mu}_k)$ in the column space of $\boldsymbol{\Phi}_k$ may be unrealistic since $\boldsymbol{u}(\boldsymbol{\mu}_k)$ corresponds to an entire time history. In this case, POD (Algorithm 4) can be applied to $\boldsymbol{u}(\boldsymbol{\mu}_k)$ and $\hat{\boldsymbol{v}}(\boldsymbol{\mu}_k)$ with the level of compression set such that (3.14) and (3.15) are satisfied. Then the basis can be defined as*

$$\boldsymbol{\Phi}_k = \texttt{PODHSP}(\texttt{POD}(\boldsymbol{u}(\boldsymbol{\mu}_k)), \mathbf{U}_k, \texttt{POD}(\hat{\boldsymbol{v}}(\boldsymbol{\mu}_k)), \hat{\mathbf{V}}_k).$$

This is similar to the original TRPOD method [10] that constructs the reduced basis according to $\boldsymbol{\Phi}_k = \texttt{POD}(\boldsymbol{u}(\boldsymbol{\mu}_k))$ or the extension presented in [57] that also constructs a reduced-order model for the adjoint that constructs the basis according to $\boldsymbol{\Phi}_k^{\boldsymbol{v}} = \texttt{POD}(\boldsymbol{v}(\boldsymbol{\mu}_k))$.

To close this section, global convergence of Algorithm 11 is established based on Theorem A.1. Suppose Assumptions (AF1)–(AF2) and (AM1)–(AM4) (Appendix A) hold and let $\{\boldsymbol{\mu}_k\}$ denote the sequence of iterations produced by Algorithm 11. To apply Theorem A.1 and conclude that this algorithm is globally convergent, the choice of $m_k(\boldsymbol{\mu})$, $\vartheta_k(\boldsymbol{\mu})$, $\varphi_k(\boldsymbol{\mu})$ in (5.2), (5.3), (5.6)-(5.7) must satisfy the error bounds in (3.12), (3.13) and the conditions in (3.14), (3.15). The objective and gradient error bounds are established based on the residual-based error bounds detailed in Appendix B. The construction of the reduced-order basis $\boldsymbol{\Phi}_k$ in (5.27) combined with the fact that $\boldsymbol{\Psi}_k$ is defined according to (4.14) to ensure the reduced-order model possesses the minimum-residual property guarantees the objective (3.14) and gradient (3.15) error conditions. Therefore, by Theorem A.1, the sequences of iterates produced by Algorithm 11 satisfies

$$\lim_{k\to\infty} \inf \|\nabla F(\boldsymbol{\mu}_k)\| = 0. \tag{5.31}$$

Algorithm 11 Residual-based trust region method based on reduced-order models

1: **Initialization**: Given

$$\boldsymbol{\mu}_0,\ \mathbf{U}_{-1}=\emptyset,\ \hat{\mathbf{V}}_{-1}=\emptyset,\ \Delta_0,\ 0<\gamma<1,\ \Delta_{\max}>0,\ 0<\eta_1<\eta_2<1,$$
$$0<\kappa_\vartheta<1,\ 0<\kappa_\varphi,\ 0<\omega<1,\ \{r_k\}_{k=0}^{\infty}\text{ such that } r_k\to 0$$

2: **Model and constraint update**: If previous model and constraint are sufficient for convergence

$$\vartheta_{k-1}(\boldsymbol{\mu}_k)\le\kappa_\vartheta\Delta_k \qquad \varphi_{k-1}(\boldsymbol{\mu}_k)\le\kappa_\varphi\min\{\|\nabla m_{k-1}(\boldsymbol{\mu}_k)\|,\Delta_k\},$$

re-use for the current iteration: $m_k(\boldsymbol{\mu})\coloneqq m_{k-1}(\boldsymbol{\mu})$ and $\vartheta_k(\boldsymbol{\mu})\coloneqq\vartheta_{k-1}(\boldsymbol{\mu})$. Otherwise, evaluate primal and sensitivity or adjoint solution of high-dimensional model

$$\boldsymbol{u}_k\coloneqq\boldsymbol{u}(\boldsymbol{\mu}_k)\qquad \hat{\boldsymbol{v}}_k\coloneqq\frac{\partial\boldsymbol{u}}{\partial\boldsymbol{\mu}}(\boldsymbol{\mu}_k)\ \text{ or }\ \boldsymbol{\Theta}_k^{\boldsymbol{\lambda}}(\boldsymbol{u}(\boldsymbol{\mu}_k),\boldsymbol{\mu}_k)\frac{\partial\boldsymbol{r}}{\partial\boldsymbol{u}}(\boldsymbol{u}(\boldsymbol{\mu}_k),\boldsymbol{\mu}_k)^T\boldsymbol{\lambda}(\boldsymbol{\mu}_k)$$

and compute reduced-order basis via span-preserving variant of POD (Algorithm 7)

$$\boldsymbol{\Phi}_k=\texttt{PODHSP}(\boldsymbol{u}_k,\mathbf{U}_k,\hat{\boldsymbol{v}}_k,\hat{\mathbf{V}}_k),$$

define model and constraint as

$$m_k(\boldsymbol{\mu})=f(\boldsymbol{\Phi}_k\boldsymbol{u}_r(\boldsymbol{\mu};\boldsymbol{\Phi}_k,\boldsymbol{\Psi}_k),\boldsymbol{\mu})$$
$$\vartheta_k(\boldsymbol{\mu})=\|\boldsymbol{r}(\boldsymbol{\Phi}_k\boldsymbol{u}_r(\boldsymbol{\mu}_k;\boldsymbol{\Phi}_k,\boldsymbol{\Psi}_k),\boldsymbol{\mu}_k)\|_{\boldsymbol{\Theta}_k}+\|\boldsymbol{r}(\boldsymbol{\Phi}_k\boldsymbol{u}_r(\boldsymbol{\mu};\boldsymbol{\Phi}_k,\boldsymbol{\Psi}_k),\boldsymbol{\mu})\|_{\boldsymbol{\Theta}_k},$$

and update snapshot matrices

$$\mathbf{U}_{k+1}\leftarrow\begin{bmatrix}\mathbf{U}_{k-1} & \boldsymbol{u}_k\end{bmatrix}\qquad \hat{\mathbf{V}}_{k+1}\leftarrow\begin{bmatrix}\hat{\mathbf{V}}_{k-1} & \hat{\boldsymbol{v}}_k\end{bmatrix}.$$

3: **Step computation**: Solve (exactly) the trust region subproblem

$$\min_{\boldsymbol{\mu}\in\mathbb{R}^{N_{\boldsymbol{\mu}}}} m_k(\boldsymbol{\mu})\qquad\text{subject to}\qquad\vartheta_k(\boldsymbol{\mu})\le\Delta_k$$

for a candidate, $\hat{\boldsymbol{\mu}}_k$, using interior-point method of Section 3.1.2.

4: **Actual-to-predicted reduction**: Compute actual-to-predicted reduction ratio

$$\rho_k=\begin{cases}1 & \text{if}\quad \vartheta_k(\hat{\boldsymbol{\mu}}_k)^\omega\le\eta\min\{m_k(\boldsymbol{\mu}_k)-m_k(\hat{\boldsymbol{\mu}}_k),r_k\}\\ \dfrac{F(\boldsymbol{\mu}_k)-F(\hat{\boldsymbol{\mu}}_k)}{m_k(\boldsymbol{\mu}_k)-m_k(\hat{\boldsymbol{\mu}}_k)} & \text{otherwise}\end{cases}$$

where $\eta<\min\{\eta_1,1-\eta_2\}$

5: **Step acceptance**:

if $\rho_k\ge\eta_1$ **then** $\boldsymbol{\mu}_{k+1}=\hat{\boldsymbol{\mu}}_k$ **else** $\boldsymbol{\mu}_{k+1}=\boldsymbol{\mu}_k$ **end if**

6: **Trust region update**:

if $\rho_k\le\eta_1$ **then** $\Delta_{k+1}\in(0,\gamma\vartheta_k(\hat{\boldsymbol{\mu}}_k)]$ **end if**
if $\rho_k\in(\eta_1,\eta_2)$ **then** $\Delta_{k+1}\in[\gamma\vartheta_k(\hat{\boldsymbol{\mu}}_k),\Delta_k]$ **end if**
if $\rho_k\ge\eta_2$ **then** $\Delta_{k+1}\in[\Delta_k,\Delta_{\max}]$ **end if**

5.2 Snapshots from Partially Converged Solutions

In many large-scale applications, particularly those arising in turbulent computational fluid dynamics, it is difficult and expensive to compute a steady-state solution, and the corresponding sensitivity and adjoint solutions, to tight tolerances. In these cases, the generation of snapshots in Line 2 of Algorithm 11 will dominate the cost of the trust region method. To speed up this step and leverage the flexibility afforded by the trust region method of Section 3.1.1, *partially converged* solutions are used as snapshots.

Let $\boldsymbol{u}(\boldsymbol{\mu};\, \tau_{\boldsymbol{u}})$ denote a partially converged primal solution of tolerance $\tau_{\boldsymbol{u}}$, defined as any point that satisfies

$$\|\boldsymbol{r}(\,\cdot\,,\, \boldsymbol{\mu})\|_{\boldsymbol{\Theta}} \leq \tau_{\boldsymbol{u}}. \tag{5.32}$$

While the fully converged solution $\boldsymbol{u}(\boldsymbol{\mu})$ is assumed to be unique (Assumption 2.2), there are many points satisfying (5.32) for a given $\tau_{\boldsymbol{u}} > 0$. A simple method to find a point that satisfies (5.32) is to use the chosen nonlinear solver (Newton-Raphson, Gauss-Newton, pseudo-transient continuation) with (5.32) used as the convergence criteria[1]. Similarly, let $\boldsymbol{v}(\boldsymbol{\mu};\, \tau_{\boldsymbol{u}},\, \tau_{\boldsymbol{v}})$ be a partially converged sensitivity or adjoint solution of tolerance $\tau_{\boldsymbol{v}}$ about a partially converged primal solution of tolerance $\tau_{\boldsymbol{u}}$, defined as any point satisfying

$$\|\boldsymbol{r}^{\boldsymbol{v}}(\boldsymbol{u}(\boldsymbol{\mu};\, \tau_{\boldsymbol{u}}),\, \cdot\,,\, \boldsymbol{\mu})\|_{\boldsymbol{\Theta}^{\boldsymbol{v}}} \leq \tau_{\boldsymbol{v}} \tag{5.33}$$

where $\boldsymbol{r}^{\boldsymbol{v}}$ is the sensitivity or adjoint residual, depending on which method is used to compute reduced-space gradients[2]. Furthermore, these definition are extended to define the partially converged snapshot $\hat{\boldsymbol{v}}(\boldsymbol{\mu};\, \tau_{\boldsymbol{u}},\, \tau_{\boldsymbol{v}})$ as

$$\hat{\boldsymbol{v}}(\boldsymbol{\mu};\, \tau_{\boldsymbol{u}},\, \tau_{\boldsymbol{v}}) = \begin{cases} \boldsymbol{v}(\boldsymbol{\mu};\, \tau_{\boldsymbol{u}},\, \tau_{\boldsymbol{v}}) & \text{sensitivity method} \\ \boldsymbol{\Theta}^{\boldsymbol{\lambda}}(\boldsymbol{u}(\boldsymbol{\mu};\, \tau_{\boldsymbol{u}}),\, \boldsymbol{\mu}) \dfrac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}(\boldsymbol{u}(\boldsymbol{\mu};\, \tau_{\boldsymbol{u}}),\, \boldsymbol{\mu})^T \boldsymbol{v}(\boldsymbol{\mu};\, \tau_{\boldsymbol{u}},\, \tau_{\boldsymbol{v}}) & \text{adjoint method.} \end{cases} \tag{5.34}$$

With these definitions, the snapshot matrices of partially converged solutions are defined as

$$\begin{aligned} \mathbf{U}_k &= \begin{bmatrix} \boldsymbol{u}(\boldsymbol{\mu}_0;\, \tau_{\boldsymbol{u}}^0) & \cdots & \boldsymbol{u}(\boldsymbol{\mu}_{k-1},\, \tau_{\boldsymbol{u}}^{k-1}) \end{bmatrix} \\ \hat{\mathbf{V}}_k &= \begin{bmatrix} \hat{\boldsymbol{v}}(\boldsymbol{\mu}_0;\, \tau_{\boldsymbol{u}}^0,\, \tau_{\boldsymbol{v}}^0) & \cdots & \hat{\boldsymbol{v}}(\boldsymbol{\mu}_{k-1},\, \tau_{\boldsymbol{u}}^{k-1},\, \tau_{\boldsymbol{v}}^{k-1}) \end{bmatrix} \end{aligned} \tag{5.35}$$

and the reduced-order basis is constructed from these snapshots using the heterogeneous span-preserving variant of POD (Algorithm 7)

$$\boldsymbol{\Phi}_k = \texttt{PODHSP}(\boldsymbol{u}(\boldsymbol{\mu}_k;\, \tau_{\boldsymbol{u}}^k),\, \mathbf{U}_k,\, \hat{\boldsymbol{v}}(\boldsymbol{\mu}_k;\, \tau_{\boldsymbol{u}}^k,\, \tau_{\boldsymbol{v}}^k),\, \hat{\mathbf{V}}_k), \tag{5.36}$$

[1] This is not a restrictive requirement since residual-based convergence criteria are usually, if not always, used for nonlinear solvers. However, using a norm other than the 2-norm is non-standard.

[2] This assumes an iterative solvers is used to solve the linear sensitivity or adjoint system since a direct solver will always return the solution to machine precision.

where $\tau_{\boldsymbol{u}}^k$ and $\tau_{\boldsymbol{v}}^k$ are iteration-dependent tolerances. This definition of $\boldsymbol{\Phi}_k$ guarantees that the partially converged primal and sensitivity/adjoint solutions are contained in the reduced subspace to the exact accuracy at which they were computed, i.e.,

$$\boldsymbol{u}(\boldsymbol{\mu}_k;\, \tau_{\boldsymbol{u}}^k) \in \text{col}(\boldsymbol{\Phi}_k) \qquad \hat{\boldsymbol{v}}(\boldsymbol{\mu}_k;\, \tau_{\boldsymbol{u}}^k,\, \tau_{\boldsymbol{v}}^k) \in \text{col}(\boldsymbol{\Phi}_k), \tag{5.37}$$

which in turn implies

$$\frac{\partial \boldsymbol{u}}{\partial \boldsymbol{\mu}}(\boldsymbol{\mu}_k;\, \tau_{\boldsymbol{u}}^k,\, \tau_{\boldsymbol{v}}^k) \in \text{col}(\boldsymbol{\Phi}_k) \qquad \text{or} \qquad \boldsymbol{\lambda}(\boldsymbol{\mu}_k;\, \tau_{\boldsymbol{u}}^k,\, \tau_{\boldsymbol{v}}^k) \in \text{col}(\boldsymbol{\Psi}_k), \tag{5.38}$$

assuming the conditions in Propositions 4.3 or 4.5 are satisfied. If a minimum-residual primal reduced-order model with optimality metric $\boldsymbol{\Theta}$ is used, the optimality property gives

$$\|\boldsymbol{r}(\boldsymbol{\Phi}_k \boldsymbol{u}_r(\boldsymbol{\mu}_k;\, \boldsymbol{\Phi}_k,\, \boldsymbol{\Psi}_k)\|_{\boldsymbol{\Theta}} \leq \left\|\boldsymbol{r}(\boldsymbol{u}(\boldsymbol{\mu}_k;\, \tau_{\boldsymbol{u}}^k),\, \boldsymbol{\mu}_k)\right\|_{\boldsymbol{\Theta}} \leq \tau_{\boldsymbol{u}}^k \tag{5.39}$$

The first inequality holds from the optimality property (Proposition 4.1) since $\boldsymbol{u}(\boldsymbol{\mu}_k;\, \tau_{\boldsymbol{u}}^k) \in \text{col}(\boldsymbol{\Phi}_k)$ and the second inequality holds from the definition of the partially converged solution in (5.32). For the residual-based error indicators (5.3), (5.6), and (5.7), this implies

$$\begin{aligned} \vartheta_k(\boldsymbol{\mu}_k) &\leq 2\tau_{\boldsymbol{u}}^k \\ \varphi_k(\boldsymbol{\mu}_k) &\leq \alpha_1 \tau_{\boldsymbol{u}}^k + \alpha_2 \tau_{\boldsymbol{v}}^k, \end{aligned} \tag{5.40}$$

provided minimum-residual reduced-order models are used. Therefore the objective error condition (3.14) is satisfied if

$$\tau_{\boldsymbol{u}}^k \leq (1/2)\kappa_\vartheta \Delta_k \tag{5.41}$$

holds and the gradient error condition (3.15) is satisfied if

$$\begin{aligned} \alpha_1 \tau_{\boldsymbol{u}}^k &\leq \kappa_\varphi \min\{\|\nabla m_k(\boldsymbol{\mu}_k)\|\,,\, \Delta_k\} \\ \alpha_2 \tau_{\boldsymbol{v}}^k &\leq \kappa_\varphi \min\{\|\nabla m_k(\boldsymbol{\mu}_k)\|\,,\, \Delta_k\} \end{aligned} \tag{5.42}$$

holds. These bounds are combined to yield the following requirement on $\tau_{\boldsymbol{u}}^k$ and $\tau_{\boldsymbol{v}}^k$

$$\begin{aligned} \tau_{\boldsymbol{u}}^k &\leq (1/\alpha_1)\kappa_\varphi \min\{\|\nabla m_k(\boldsymbol{\mu}_k)\|\,,\, \kappa\Delta_k\} \\ \tau_{\boldsymbol{v}}^k &\leq (1/\alpha_2)\kappa_\varphi \min\{\|\nabla m_k(\boldsymbol{\mu}_k)\|\,,\, \Delta_k\}, \end{aligned} \tag{5.43}$$

where $\kappa = \min\{1,\, \alpha_1\kappa_\vartheta/(2\kappa_\varphi)\}$. Since this condition ensures (3.12)-(3.15), global convergence of the resulting trust region method is guaranteed. The relationship in (5.43), and the results that follow, only hold in the $\boldsymbol{\Theta}$-norm (Proposition 4.1) so the $\boldsymbol{I}$-norm form of the residual-based error indicators in (5.18) cannot be used, unless $\boldsymbol{\Theta} = \boldsymbol{I}$ (LSPG).

Remark. *The value of $\tau_{\boldsymbol{u}}^k$ and $\tau_{\boldsymbol{v}}^k$ depend on $\|\nabla m_k(\boldsymbol{\mu}_k)\|$, which in turn depends on the values of*

$\tau_{\boldsymbol{u}}^k$ and $\tau_{\boldsymbol{v}}^k$ used to define snapshots for $\boldsymbol{\Phi}_k$. Therefore, the values of $\tau_{\boldsymbol{u}}^k$ and $\tau_{\boldsymbol{v}}^k$ cannot be simply determined from (5.43). Instead, an iterative method is employed that begins initially selects large values $\tau_{\boldsymbol{u}}^k$ and $\tau_{\boldsymbol{v}}^k$ and systematically reduces them, i.e., via backtracking, until the conditions in (3.14), (3.15) are satisfied. This will lead to an efficient algorithm since the partially converged primal and dual solutions for a given $\tau_{\boldsymbol{u}}^k$ and $\tau_{\boldsymbol{v}}^k$ can be used to warm-start the nonlinear solvers for any smaller values for these tolerances. An alternate approach replaces the gradient condition in (3.15) with

$$\varphi_k(\boldsymbol{\mu}_k) \leq \kappa_\varphi \min\{\left\|\nabla m_{k-1}(\boldsymbol{\mu}_{k-1})\right\|, \Delta_k\}. \tag{5.44}$$

It can be verified that this will preserve the convergence result in Theorem A.1 of Appendix A. This replaces the the gradient condition in (5.43) with

$$\begin{aligned} \tau_{\boldsymbol{u}}^k &\leq \frac{1}{2\alpha_1}\kappa_\varphi \min\{\left\|\nabla m_{k-1}(\boldsymbol{\mu}_{k-1})\right\|, \kappa\Delta_k\} \\ \tau_{\boldsymbol{v}}^k &\leq \frac{1}{2\alpha_2}\kappa_\varphi \min\{\left\|\nabla m_{k-1}(\boldsymbol{\mu}_{k-1})\right\|, \Delta_k\}. \end{aligned} \tag{5.45}$$

This alternate gradient condition preserves global convergence and allows for the direct computation of $\tau_{\boldsymbol{u}}^k$ and $\tau_{\boldsymbol{v}}^k$ since all terms on the right-hand side of the inequality are independent of $\tau_{\boldsymbol{u}}^k$ and $\tau_{\boldsymbol{v}}^k$.

Remark. *The case with the traditional trust region constraint, $\vartheta_k(\boldsymbol{\mu}) = \|\boldsymbol{\mu} - \boldsymbol{\mu}_k\|$, satisfies $\vartheta_k(\boldsymbol{\mu}_k) = 0$ trivially and therefore the lighter restrictions on $\tau_{\boldsymbol{u}}^k$ and $\tau_{\boldsymbol{v}}^k$ in (5.42) can be used in place of those in (5.43).*

5.3 Efficient Trust Region Assessment with Partially Converged Solutions

Another opportunity for efficiency afforded by the flexible trust region framework introduced in Section 3.1.1, that has not been fully leveraged in the residual-based reduced-order model trust region method of this chapter, is the use of an approximation model to compute the ratio of actual-to-predicted ratio, ρ_k. Section 3.1.1 outlined the use of this flexibility to *effectively skip* the computation of the actual-to-predicted reduction ratio by taking

$$\psi_k(\boldsymbol{\mu}) = m_k(\boldsymbol{\mu}) \qquad \text{and} \qquad \theta_k(\boldsymbol{\mu}) = \vartheta_k(\boldsymbol{\mu})$$

whenever $\vartheta_k(\hat{\boldsymbol{\mu}}_k)^\omega \leq \eta \min\{m_k(\boldsymbol{\mu}_k) - m_k(\hat{\boldsymbol{\mu}}_k), r_k\}$. In this situation, the approximation to the actual-to-predicted reduction ratio is always unity and the step is accepted and the radius increased. This is implies the trust region assessment step is effectively free since it does not require a query to $F(\boldsymbol{\mu})$ and is guaranteed to preserve global convergence since it conforms to (3.21), (3.22). In Section 5.1, the true value of ρ_k is computed (3.9) when $\vartheta_k(\hat{\boldsymbol{\mu}}_k)$ fails to satisfy the above bound. This section seeks to improve on this using the approximate form of ρ_k in (3.20) where $\psi_k(\boldsymbol{\mu})$ leverages *partially converged* solutions.

In the event that the error condition in (3.22) is not satisfied, the choice $\psi_k(\boldsymbol{\mu}) = m_k(\boldsymbol{\mu})$ is not sufficient to ensure convergence. Instead, partially converged solutions are used as they can substantially less expensive to compute than fully converged ones and can be tailored to exactly meet the error criteria in (3.22). Consider the objective model $\psi_k(\boldsymbol{\mu})$ defined by the quantity of interest evaluated at a partially converged steady state and the corresponding residual-based error indicator

$$\begin{aligned} \psi_k(\boldsymbol{\mu}) &= f(\boldsymbol{u}(\boldsymbol{u};\, \hat{\tau}_{\boldsymbol{u}}^k),\, \boldsymbol{\mu}) \\ \theta_k(\boldsymbol{\mu}) &= \left\|\boldsymbol{r}(\boldsymbol{u}(\boldsymbol{\mu}_k;\, \hat{\tau}_{\boldsymbol{u}}^k),\, \boldsymbol{\mu}_k)\right\|_{\boldsymbol{\Theta}} + \left\|\boldsymbol{r}(\boldsymbol{u}(\boldsymbol{\mu};\, \hat{\tau}_{\boldsymbol{u}}^k),\, \boldsymbol{\mu})\right\|_{\boldsymbol{\Theta}}. \end{aligned} \tag{5.46}$$

Unlike in the previous section where partially converged solutions are used as snapshots, the $\boldsymbol{\Theta}$-norm above can be freely replaced with the $\boldsymbol{I}$-norm for simplicity (and computational efficiency), provided partially converged solutions are defined with respect to the $\boldsymbol{I}$-norm. Either norm can be used in this case since the optimality property of minimum-residual ROMs (Proposition 4.1) is not required as it was in the previous section. However, it is desirable to use the same norm in both cases since the computation of $\boldsymbol{u}(\hat{\boldsymbol{\mu}}_k;\, \hat{\tau}_{\boldsymbol{u}}^k)$, required to compute $\psi_k(\hat{\boldsymbol{\mu}}_k)$, will provide a better warm-start for the snapshot computation $\boldsymbol{u}(\boldsymbol{\mu}_k;\, \tau_{\boldsymbol{u}}^{k+1})$ at iteration $k+1$.

With these choices, the bound in (3.21) holds from an identical argument to that in (5.43). From the definition of $\boldsymbol{u}(\boldsymbol{\mu};\, \hat{\tau}_{\boldsymbol{u}}^k)$ in Section 5.2, the following relation holds

$$\theta_k(\hat{\boldsymbol{\mu}}_k) \leq 2\hat{\tau}_{\boldsymbol{u}}^k. \tag{5.47}$$

Therefore, the accuracy condition in (3.22) holds provided

$$\hat{\tau}_{\boldsymbol{u}}^k \leq \frac{1}{2}\left[\eta \min\{m_k(\boldsymbol{\mu}_k) - m_k(\hat{\boldsymbol{\mu}}_k),\, r_k\}\right]^{1/\omega} \tag{5.48}$$

and global convergence is ensured. In addition to being a less expensive option than fully converged evaluations of $F(\boldsymbol{\mu})$, this method fits seamlessly with the use of partially converged solutions in the snapshot computations of the previous section. When an iteration is accepted, i.e., $\boldsymbol{\mu}_{k+1} = \hat{\boldsymbol{\mu}}_k$, the partially converged solution $\boldsymbol{u}(\boldsymbol{\mu}_{k+1},\, \hat{\tau}_{\boldsymbol{u}}^k) = \boldsymbol{u}(\hat{\boldsymbol{\mu}}_k,\, \hat{\tau}_{\boldsymbol{u}}^k)$ can be used to warm-start the computation of $\boldsymbol{u}(\boldsymbol{\mu}_{k+1},\, \tau_{\boldsymbol{u}}^{k+1})$ that is required to compute snapshots for iteration $k+1$. In fact, if $\hat{\tau}_{\boldsymbol{u}}^k \leq \tau_{\boldsymbol{u}}^{k+1}$ the computation can be skipped entirely since $\boldsymbol{u}(\hat{\boldsymbol{\mu}}_k,\, \hat{\tau}_{\boldsymbol{u}}^k)$ already satisfies

$$\boldsymbol{r}(\boldsymbol{u}(\hat{\boldsymbol{\mu}}_k,\, \hat{\tau}_{\boldsymbol{u}}^k),\, \boldsymbol{\mu}_{k+1}) \leq \tau_{\boldsymbol{u}}^{k+1}. \tag{5.49}$$

The complete algorithm that uses partially converged solutions as snapshots in the model update and in the computation of the actual-to-predicted reduction ratio is provided in Algorithm 12. To establish global convergence of this algorithm based on Theorem A.1, suppose Assumptions (AF1)–(AF2) and (AM1)–(AM4) (Appendix A) hold and let $\{\boldsymbol{\mu}_k\}$ denote the sequence of iterates produced by Algorithm 12. Section 5.1.1 already established that the choice of $m_k(\boldsymbol{\mu})$, $\vartheta_k(\boldsymbol{\mu})$, $\varphi_k(\boldsymbol{\mu})$ satisfy the error bounds in (3.14), (3.15). The construction of the reduced-order basis $\boldsymbol{\Phi}_k$ in (5.27) and the requirements placed on the partially converged solutions in (5.43), combined with the fact that $\boldsymbol{\Psi}_k$

is defined according to (4.14) to ensure the reduced-order model possesses the minimum-residual property guarantees the objective (3.14) and gradient (3.15) error conditions hold. Finally, $\psi_k(\boldsymbol{\mu})$ and $\theta_k(\boldsymbol{\mu})$ in (5.46) must satisfy the error bound (3.21) and condition (3.22) to preserve global convergence when the approximate actual-to-predicted ratio is used to assess the trust region step. The residual-based error bounds established in Lemma B.4, B.7, B.8 ensures the error bound holds. The requirements on the partially converged solution in (5.47)-(5.48) ensure the error condition (3.22) holds. Therefore, by Theorem A.1, the sequences of iterates produced by Algorithm 12 satisfies

$$\lim_{k\to\infty} \inf \|\nabla F(\boldsymbol{\mu}_k)\| = 0. \tag{5.50}$$

Algorithm 12 Residual-based trust region method based on reduced-order models and partially converged solutions

1: **Initialization**: Given

$$\boldsymbol{\mu}_0,\ \mathbf{U}_{-1}=\emptyset,\ \hat{\mathbf{V}}_{-1}=\emptyset,\ \Delta_0,\ 0<\gamma<1,\ \Delta_{\max}>0,\ 0<\eta_1<\eta_2<1,$$
$$0<\kappa_\vartheta<1,\ 0<\kappa_\varphi,\ 0<\omega<1,\ \{r_k\}_{k=0}^{\infty}\text{ such that }r_k\to 0$$

2: **Model and constraint update**: If previous model and constraint are sufficient for convergence

$$\vartheta_{k-1}(\boldsymbol{\mu}_k)\le\kappa_\vartheta\Delta_k\qquad\qquad\varphi_{k-1}(\boldsymbol{\mu}_k)\le\kappa_\varphi\min\{\|\nabla m_{k-1}(\boldsymbol{\mu}_k)\|,\Delta_k\},$$

re-use for the current iteration: $m_k(\boldsymbol{\mu})\coloneqq m_{k-1}(\boldsymbol{\mu})$ and $\vartheta_k(\boldsymbol{\mu})\coloneqq\vartheta_{k-1}(\boldsymbol{\mu})$. Otherwise, evaluate primal and sensitivity or adjoint solution of high-dimensional model to tolerances $\tau_{\boldsymbol{u}}^k$ and $\tau_{\boldsymbol{v}}^k$, respectively,

$$\begin{aligned}\boldsymbol{u}_k&\coloneqq\boldsymbol{u}(\boldsymbol{\mu}_k;\tau_{\boldsymbol{u}}^k)\\ \hat{\boldsymbol{v}}_k&\coloneqq\frac{\partial\boldsymbol{u}}{\partial\boldsymbol{\mu}}(\boldsymbol{\mu}_k;\tau_{\boldsymbol{u}}^k,\tau_{\boldsymbol{v}}^k)\ \text{ or }\ \boldsymbol{\Theta}^{\boldsymbol{\lambda}}(\boldsymbol{u}(\boldsymbol{\mu}_k;\tau_{\boldsymbol{u}}^k),\boldsymbol{\mu}_k)\frac{\partial\boldsymbol{r}}{\partial\boldsymbol{u}}(\boldsymbol{u}(\boldsymbol{\mu}_k;\tau_{\boldsymbol{u}}^k),\boldsymbol{\mu}_k)^T\boldsymbol{\lambda}(\boldsymbol{\mu}_k;\tau_{\boldsymbol{u}}^k,\tau_{\boldsymbol{v}}^k)\end{aligned}$$

with tolerances given by

$$\begin{aligned}\tau_{\boldsymbol{u}}^k&\le\frac{1}{2\alpha_1}\kappa_\varphi\min\{\|\nabla m_k(\boldsymbol{\mu}_k)\|,\kappa\Delta_k\}\qquad\kappa=\min\{1,\alpha_1\kappa_\vartheta/(2\kappa_\varphi)\}\\ \tau_{\boldsymbol{v}}^k&\le\frac{1}{2\alpha_2}\kappa_\varphi\min\{\|\nabla m_k(\boldsymbol{\mu}_k)\|,\Delta_k\},\end{aligned}$$

and compute reduced-order basis via span-preserving variant of POD (Algorithm 7)

$$\boldsymbol{\Phi}_k=\texttt{PODHSP}(\boldsymbol{u}_k,\mathbf{U}_k,\hat{\boldsymbol{v}}_k,\hat{\mathbf{V}}_k),$$

define model and constraint as

$$\begin{aligned}m_k(\boldsymbol{\mu})&=f(\boldsymbol{\Phi}_k\boldsymbol{u}_r(\boldsymbol{\mu};\boldsymbol{\Phi}_k,\boldsymbol{\Psi}_k),\boldsymbol{\mu})\\ \vartheta_k(\boldsymbol{\mu})&=\|\boldsymbol{r}(\boldsymbol{\Phi}_k\boldsymbol{u}_r(\boldsymbol{\mu}_k;\boldsymbol{\Phi}_k,\boldsymbol{\Psi}_k),\boldsymbol{\mu}_k)\|_{\boldsymbol{\Theta}}+\|\boldsymbol{r}(\boldsymbol{\Phi}_k\boldsymbol{u}_r(\boldsymbol{\mu};\boldsymbol{\Phi}_k,\boldsymbol{\Psi}_k),\boldsymbol{\mu})\|_{\boldsymbol{\Theta}},\end{aligned}$$

and update snapshot matrices

$$\mathbf{U}_{k+1}\leftarrow\begin{bmatrix}\mathbf{U}_{k-1}&\boldsymbol{u}_k\end{bmatrix}\qquad\hat{\mathbf{V}}_{k+1}\leftarrow\begin{bmatrix}\hat{\mathbf{V}}_{k-1}&\hat{\boldsymbol{v}}_k\end{bmatrix}.$$

3: **Step computation**: identical to Line 3 in Algorithm 11

4: **Computed-to-predicted reduction**: Compute computed-to-predicted reduction ratio

$$\rho_k=\begin{cases}1&\text{if}\quad\vartheta_k(\hat{\boldsymbol{\mu}}_k)^\omega\le\hat{\tau}_{\boldsymbol{u}}^k\\ \dfrac{\psi_k(\boldsymbol{\mu}_k)-\psi_k(\hat{\boldsymbol{\mu}}_k)}{m_k(\boldsymbol{\mu}_k)-m_k(\hat{\boldsymbol{\mu}}_k)}&\text{otherwise}\end{cases}$$

$$\psi_k(\boldsymbol{\mu})\coloneqq f(\boldsymbol{u}(\boldsymbol{\mu};\hat{\tau}_{\boldsymbol{u}}^k),\boldsymbol{\mu})\qquad\hat{\tau}_{\boldsymbol{u}}^k=\frac{1}{2}\left[\eta\min\{m_k(\boldsymbol{\mu}_k)-m_k(\hat{\boldsymbol{\mu}}_k),r_k\}\right]^{1/\omega}\qquad\eta<\min\{\eta_1,1-\eta_2\}$$

5: **Step acceptance**: identical to Line 5 in Algorithm 11

6: **Trust region update**: identical to Line 6 in Algorithm 11

5.4 Extension to Hyperreduced Models

To this point, projection-based reduced-order models have solely been considered as the trust region approximation model. However, as discussed in Chapter 4, this will not be sufficient to realize non-trivial speedups for nonlinear problems. For such problems, the computational complexity associated with the evaluation of the reduced residual and Jacobian scales with the size of the original HDM since they require reconstruction of the full state vector from the reduced coordinates, assembly over the entire mesh, and subsequent projection onto the columnspace of the test basis. For this reason, the developments in this chapter are extended to use collocation-based hyperreduced models as the approximation model.

The approximation model takes the same form as in the previous sections, i.e.,

$$m_k(\boldsymbol{\mu}) = f(\boldsymbol{\Phi}_k \boldsymbol{u}_r(\boldsymbol{\mu};\, \boldsymbol{\Phi}_k,\, \boldsymbol{\Psi}_k,\, \boldsymbol{P}_k),\, \boldsymbol{\mu}), \tag{5.51}$$

with the exception that the reduced coordinates $\boldsymbol{u}_r^k(\boldsymbol{\mu})$ are defined as the solution of the collocation-based hyperreduced model

$$(\boldsymbol{P}_k^T \boldsymbol{\Psi}_k)^T \boldsymbol{P}_k^T \boldsymbol{r}(\bar{\boldsymbol{P}}_k \bar{\boldsymbol{P}}_k^T \boldsymbol{\Phi}_k \boldsymbol{u}_r(\boldsymbol{\mu};\, \boldsymbol{\Phi}_k,\, \boldsymbol{\Psi}_k,\, \boldsymbol{P}_k),\, \boldsymbol{\mu}) = 0. \tag{5.52}$$

The gradient $\nabla m_k(\boldsymbol{\mu})$ is computed according to the adjoint or sensitivity method presented in Sections 4.2.5–4.2.6 or approximated using the minimum-residual variants. For the sake of efficiency, the residual-based trust region constraint $\vartheta_k(\boldsymbol{\mu})$ in (5.3) is replaced with the *masked* residual

$$\vartheta_k(\boldsymbol{\mu}) = \left\| \boldsymbol{P}_k^T \boldsymbol{r}(\bar{\boldsymbol{P}}_k \bar{\boldsymbol{P}}_k^T \boldsymbol{\Phi}_k \boldsymbol{u}_r^k(\boldsymbol{\mu}),\, \boldsymbol{\mu}) \right\| \tag{5.53}$$

and the gradient error indicator $\varphi_k(\boldsymbol{\mu})$ is similarly replaced with its *masked* counterpart, i.e.,

$$\begin{aligned} \varphi_k(\boldsymbol{\mu}) = \;& \alpha_1 \left\| \boldsymbol{P}_k^T \boldsymbol{r}(\bar{\boldsymbol{P}}_k \bar{\boldsymbol{P}}_k^T \boldsymbol{\Phi}_k \boldsymbol{u}_r(\boldsymbol{\mu};\, \boldsymbol{\Phi}_k,\, \boldsymbol{\Psi}_k,\, \boldsymbol{P}_k),\, \boldsymbol{\mu}) \right\| + \\ & \alpha_2 \left\| \boldsymbol{P}_k^T \boldsymbol{r}^{\partial} \left(\bar{\boldsymbol{P}}_k \bar{\boldsymbol{P}}_k^T \boldsymbol{\Phi}_k \boldsymbol{u}_r(\boldsymbol{\mu};\, \boldsymbol{\Phi}_k,\, \boldsymbol{\Psi}_k,\, \boldsymbol{P}_k),\, \bar{\boldsymbol{P}}_k \bar{\boldsymbol{P}}_k^T \boldsymbol{\Phi}_k \frac{\partial \boldsymbol{u}_r}{\partial \boldsymbol{\mu}}(\boldsymbol{\mu};\, \boldsymbol{\Phi}_k,\, \boldsymbol{\Psi}_k,\, \boldsymbol{P}_k),\, \boldsymbol{\mu} \right) \right\|. \end{aligned} \tag{5.54}$$

Only the sensitivity method is considered since details pertaining to the hyperreduced minimum-residual adjoint method is deferred to future work.

For general nonlinear systems of equations $\boldsymbol{r}(\boldsymbol{u},\, \boldsymbol{\mu}) = 0$, these choices of error indicators $\vartheta_k(\boldsymbol{\mu})$ and $\varphi_k(\boldsymbol{\mu})$ do not lead to the required bounds in (3.12) and (3.13) and global convergence cannot be rigorously established. However, due to the concept of a *stencil* in the discretization of partial differential equations, i.e., the fact that the ith entry of $\boldsymbol{r}$ depends on the jth entry of $\boldsymbol{u}$ for all $j \in \mathcal{S}_i$ (defined in Section 4.2.2), it is reasonable to expect such bounds to hold (with larger constants ζ and ξ), provided the mask is sufficiently large.

The details pertaining to the construction of $\boldsymbol{\Phi}_k$ from fully (Section 5.1.2) or partially (Sections 5.2) converged solutions carries over to the case of collocation-based hyperreduced models;

however, the bounds on the error indicators can no longer be guaranteed due to the introduction of the mask in the reduced-order model. Once the trial basis is constructed, the mask $\boldsymbol{P}_k$ is constructed according to the algorithm detailed in [198] that relies solely on $\boldsymbol{\Phi}_k$ and possibly problem-specific information. The approximation of the ratio of actual-to-predicted reduction that uses $\psi_k(\boldsymbol{\mu})$ and $\theta_k(\boldsymbol{\mu})$ based on partially converged solutions is not specific to the case of projection-based reduced-order models and therefore trivially carries over to the hyperreduced case. The complete trust region algorithm based on collocation-based hyperreduced models is identical to Algorithms 11 and 12, once the step that constructs the mask $\boldsymbol{P}_k$ from $\boldsymbol{\Phi}_k$ is added, with the above definitions of $m_k(\boldsymbol{\mu})$, $\vartheta_k(\boldsymbol{\mu})$, and $\varphi_k(\boldsymbol{\mu})$.

5.5 Numerical Experiments

In this section, the error-aware trust region method using projection-based reduced-order models as the approximation model is applied to solve a number of problems in computational fluid dynamics, ranging from optimal control of the 1D inviscid Burgers' equation to shape optimization of a full aircraft configuration.

5.5.1 Optimal Control of 1D Inviscid Burgers' Equation

This section presents a thorough investigation of the trust region methods proposed in this chapter based on the various projection-based reduced-order models of Chapter 4. The model PDE-constrained optimization problem considered is optimal control of the steady, inviscid, one-dimensional Burgers' equation in only a few control parameters. The optimization problem takes the form

$$\underset{\boldsymbol{\mu}\in\mathbb{R}^{n_{\boldsymbol{\mu}}}}{\text{minimize}} \quad \int_0^1 \frac{1}{2}(u(\boldsymbol{\mu},\, x) - \bar{u}(x))^2\, dx \tag{5.55}$$

where $u(\boldsymbol{\mu},\, x)$ is the solution of the inviscid Burgers' equation under a specific parametrization of the inflow boundary condition and control

$$\begin{aligned} u(\boldsymbol{\mu},\, x)\partial_x u(\boldsymbol{\mu},\, x) &= \mu_2\, e^{\mu_3 x} \quad x \in (0,\, 100) \\ u(\boldsymbol{\mu},\, 0) &= \mu_1 \end{aligned} \tag{5.56}$$

and $\bar{u}(x)$ is the target state. The PDE is discretized with a first-order, vertex-centered finite volume method with 1000 vertices for a state space of dimension $N_{\boldsymbol{u}} = 999$ after application of the inflow boundary condition. The functional form of the control in (5.56) was made to minimize the number of optimization parameters to allow the sensitivity-based approach to be included in the study. The target state corresponds to the solution of the (5.56) at the target parameter configuration $\bar{\boldsymbol{\mu}} = (2.5,\, 0.02,\, 0.0425)$. Therefore, the target state is realizable since it lies within the parametrization of the optimization problem and the optimal value of the objective function is 0. All methods considered will start from an initial guess of $\boldsymbol{\mu}_0 = (1.0,\, 1.0,\, 0.0)$. Figure 5.1 shows the control

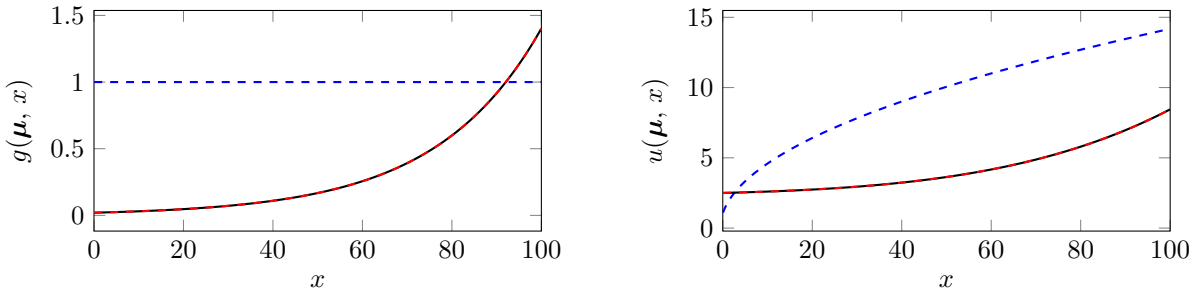


Figure 5.1: Control (left) and corresponding solution (right) of the inviscid Burgers' equation in (5.56) at: the initial condition $\boldsymbol{\mu} = (1.0, 1.0, 0.0)$ (- - -), the target solution $\boldsymbol{\mu} = (2.5, 0.02, 0.0425)$ (——), and solution of the baseline optimization method (- - -).

$g(\boldsymbol{\mu}, x)$ and state vector $u(\boldsymbol{\mu}, x)$ at the initial guess and optimal value of $\boldsymbol{\mu}$. Figure 5.2 shows the contours of the objective function (after discretization) in the $\mu_1 - \mu_2$ plane at a slice of the parameter space at $\mu_3 = 0$, with the initial condition $\boldsymbol{\mu}_0$ and optimal solution $\boldsymbol{\mu}^*$ indicated.

Trust region geometry and impact of snapshots

Before studying the entire performance of the proposed optimization solvers on the optimal control problem in (5.55), the geometry of the various trust region constraints presented in this document are considered: the traditional trust region constraint

$$\|\boldsymbol{\mu} - \boldsymbol{\mu}_0\| \leq \Delta$$

and the residual-based trust region constraint

$$\|\boldsymbol{r}(\boldsymbol{\Phi}\boldsymbol{u}_r(\boldsymbol{\mu}_0; \boldsymbol{\Phi}, \boldsymbol{\Psi}), \boldsymbol{\mu}_0)\| + \|\boldsymbol{r}(\boldsymbol{\Phi}\boldsymbol{u}_r(\boldsymbol{\mu}; \boldsymbol{\Phi}, \boldsymbol{\Psi}), \boldsymbol{\mu})\| \leq \Delta.$$

A trust region constraint based on the true error in the quantity of interest

$$|f(\boldsymbol{u}(\boldsymbol{\mu}), \boldsymbol{\mu}) - f(\boldsymbol{\Phi}\boldsymbol{u}_r(\boldsymbol{\mu}; \boldsymbol{\Phi}, \boldsymbol{\Psi}))| \leq \Delta$$

is included in this study for illustration purposes only as it is far too expensive to use in practice. The traditional trust region is purely geometric and therefore does not depend on the reduced-order model, while the residual- and error-based trust regions are heavily dependent on the type of reduced-order model employed and the trial subspace chosen. This section considers reduced-order models based on a Galerkin and LSPG projection. Since the Jacobians of the discrete inviscid Burgers' equation are not symmetric positive-definite, reduced-order models based on a Galerkin projection do not necessarily possess the minimum-residual property. However, LSPG-based ROMs do possess the minimum-residual property, by definition. The trial basis will be constructed in three different ways following the developments in Section 5.1.2: from snapshots of the primal solution only, from

snapshots of the primal and sensitivity solutions, and from snapshots of the primal and adjoint solutions. All snapshots will be computed at the control corresponding to the initial condition of the optimization problem, $\boldsymbol{\mu}_0$. That is,

$$\text{col}(\boldsymbol{\Phi}) = \text{span}\{\boldsymbol{u}(\boldsymbol{\mu}_0)\} \qquad \text{col}(\boldsymbol{\Phi}) = \text{span}\{\boldsymbol{u}(\boldsymbol{\mu}_0),\, \boldsymbol{\lambda}(\boldsymbol{\mu}_0)\} \qquad \text{col}(\boldsymbol{\Phi}) = \text{span}\left\{\boldsymbol{u}(\boldsymbol{\mu}_0),\, \frac{\partial \boldsymbol{u}}{\partial \boldsymbol{\mu}}(\boldsymbol{\mu}_0)\right\}$$

depending on which trial subspace is being considered.

The trust regions for the reduced-order models based on a Galerkin projection are provided in Figure 5.3 and those based on a LSPG projection are in Figure 5.4. These figures show the contours of the reduced objective function $f(\boldsymbol{\Phi}\boldsymbol{u}_r(\boldsymbol{\Phi}, \boldsymbol{\Psi}), \boldsymbol{\mu})$, which can be compared to the contours of the true objective function $f(\boldsymbol{u}(\boldsymbol{\mu}), \boldsymbol{\mu})$ in Figure 5.2. It can be seen that the residual-based trust regions do not match the trust regions based on the true error. Even though the use of residuals as a surrogate for the true error partially motivated the introduction of the *error-aware* trust region theory in Chapter 3, it is not a requirement since the asymptotic bound (3.12) holds due to the derivation in Appendix B. From Figures 5.3 and 5.4, a few more observations are made that agree with the minimum-residual reduced-order model theory in Chapter 4. First, the residual-based trust regions corresponding to a Galerkin ROM (Figure 5.3) are a subset of those corresponding to the LSPG ROM (Figure 5.4). This is expected since LSPG *minimizes* the residual in the $\boldsymbol{I}$-norm, which is exactly the quantity defining the trust region. Despite the larger residual-based trust regions of LSPG ROMs, the Galerkin ROMs have larger trust regions based on the true error. While non-intuitive, this does not contradict the theory outlined in Chapter 4 since LSPG is only guaranteed to minimize the residual over the trial subspace, not the error in a quantity of interest. Finally, for both the Galerkin and LSPG ROMs, the trial subspaces built from primal states and sensitivities produce larger residual- and error-based trust regions than only those that only use primal snapshots. There is disagreement between the two types of reduced-order models when it comes to the use of adjoint snapshots. For the Galerkin ROMs, the incorporation of adjoint snapshots improve the prediction capability of the reduced-order model with respect to the quantity of interest, but have little influence on the extent of the residual-based trust region. In contrast, the incorporation of adjoint snapshots increases the extent of the residual-based trust region—as expected from the monotonicity property of minimum-residual reduced-order models— however, they actually *reduce* the extent of the error-based trust region. This provides some evidence that the incorporation of non-physical snapshots can cause the residual minimization to produce worse solutions with respect to prediction of the quantity of interest [198].

Performance of proposed optimization solvers

This section provides a thorough comparison of the variants of the multifidelity trust region method based on reduced-order models and partially converged solutions in Algorithms 11 and 12. The following aspects of the algorithms will be considered in this study:

- the *type* of reduced-order model underlying the approximation model: Galerkin and LSPG

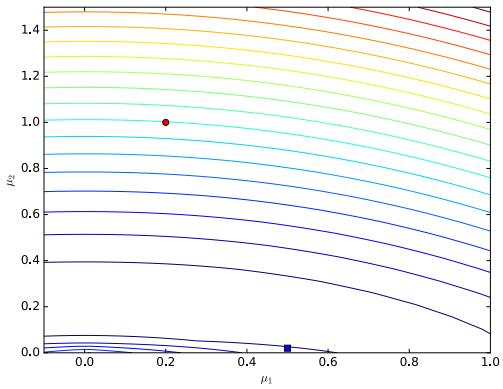


Figure 5.2: Contours of the objective function $f(\boldsymbol{u}(\boldsymbol{\mu}), \boldsymbol{\mu})$ in (5.55) in the $\mu_1 - \mu_2$ plane corresponding to a slice at $\mu_3 = 0.0$. The initial condition for the optimization problem and target solution are shown with a red circle and blue square, respectively.

projections will be considered,

- the *type of snapshots* used to define the trial subspace: primal snapshot alone, primal and sensitivity snapshots, and primal and adjoint snapshots will be considered,
- the trust region *constraint* used to define the trust region subproblem: the traditional ball constraint and residual-based constraint (5.3) will be considered, and
- the optimization *solver* used for the trust region subproblem: the interior point method of Section 3.1.2 based on a Newton-CG solver[3] will be used to exactly solve the subproblem and Steihaug-Toint CG will be used to approximately solve the subproblem (when the traditional trust region constraint is used).

Table 5.1 summarizes the variants of Algorithms 11 and 12 considered in this section and provides appropriate names for convenient reference. For the Galerkin reduced-order models, the true sensitivities and adjoints will be computed according to (4.20), (4.47) since this is amenable to implementation due to the constant test basis and will lead to consistency of the reduced functionals and their gradients. For the LSPG reduced-order models, the minimum-residual sensitivity and adjoint approximations in (4.28) and (4.56) will be employed (only guarantees consistency of functionals and gradients at trust region centers). Finally, all numerical experiments use the following trust region parameters:

$$\begin{aligned} \kappa_\vartheta = 0.5 \quad \kappa_\varphi = 2.0 \quad \gamma = 0.5 \quad \eta_1 = 0.25 \quad \eta_2 = 0.75 \\ r_k = 1/(k+1) \quad \Delta_0 = 10^{-1} \quad \Delta_{\max} = 10^5. \end{aligned} \tag{5.57}$$

[3]The interior point method based on a BFGS unconstrained solver of Algorithm 3 is replaced with a Newton-CG unconstrained solver for fair comparison to the second-order Steihaug-Toint CG method.

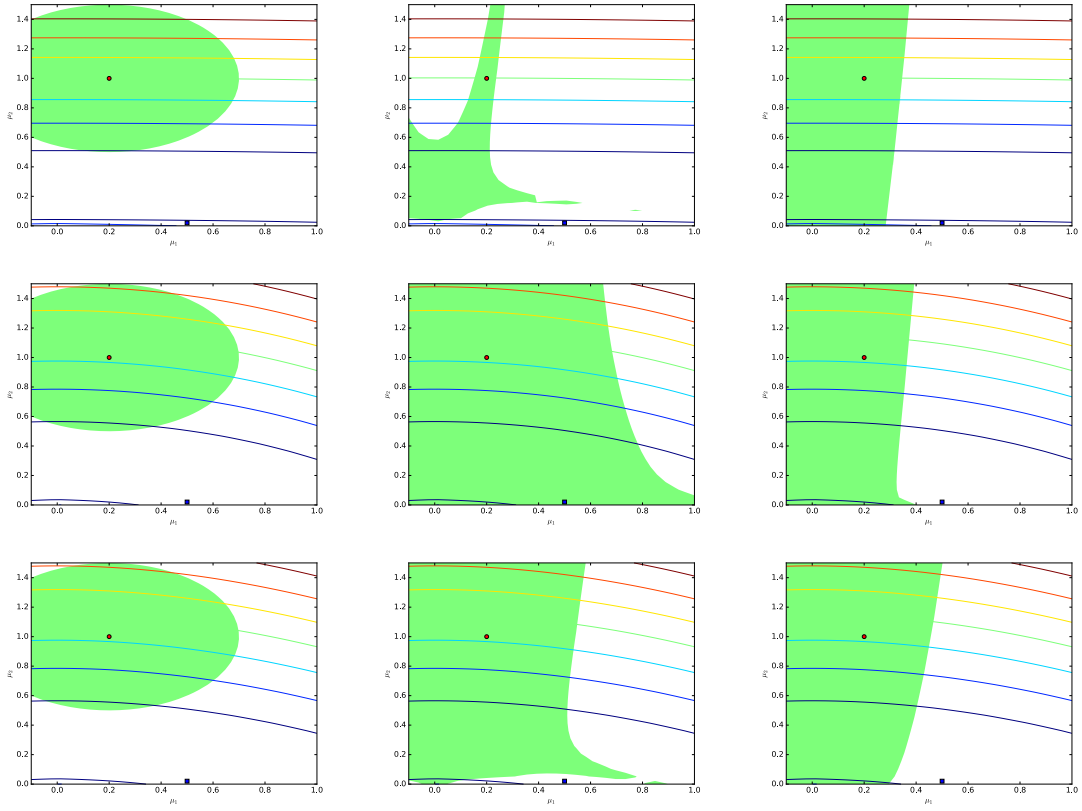


Figure 5.3: Contour of the *reduced* objective function $f(\boldsymbol{\Phi}\boldsymbol{u}_r(\boldsymbol{\mu};\,\boldsymbol{\Phi},\,\boldsymbol{\Psi}),\,\boldsymbol{\mu})$ in (5.55) in the $\mu_1 - \mu_2$ plane corresponding to a slice at $\mu_3 = 0.0$. The reduced-order model employs a *Galerkin* projection and the trial basis is constructed from: (top) the primal solution at $\boldsymbol{\mu}_0$, i.e., $\text{col}(\boldsymbol{\Phi}) = \text{span}\{\boldsymbol{u}(\boldsymbol{\mu}_0)\}$; (middle) the primal and adjoint solution at $\boldsymbol{\mu}_0$, i.e., $\text{col}(\boldsymbol{\Phi}) = \text{span}\{\boldsymbol{u}(\boldsymbol{\mu}_0),\,\boldsymbol{\lambda}(\boldsymbol{\mu}_0)\}$; (bottom) the primal and sensitivity solution at $\boldsymbol{\mu}_0$, i.e., $\text{col}(\boldsymbol{\Phi}) = \text{span}\left\{\boldsymbol{u}(\boldsymbol{\mu}_0),\,\dfrac{\partial \boldsymbol{u}}{\partial \boldsymbol{\mu}}(\boldsymbol{\mu}_0)\right\}$. The green shaded region indicates the areas where: (left) the Euclidean ball is bounded by 0.5, i.e., $\|\boldsymbol{\mu} - \boldsymbol{\mu}_0\| \leq 0.5$, (center) the error between the true and reduced objective function is bounded by 100, i.e., $|f(\boldsymbol{u}(\boldsymbol{\mu}),\,\boldsymbol{\mu}) - f(\boldsymbol{\Phi}\boldsymbol{u}_r(\boldsymbol{\mu};\,\boldsymbol{\Phi},\,\boldsymbol{\Psi}),\,\boldsymbol{\mu})| \leq 100$, and (right) the residual norm of the reconstructed ROM solution is bounded by 10, i.e., $\|\boldsymbol{r}(\boldsymbol{\Phi}\boldsymbol{u}_r(\boldsymbol{\mu};\,\boldsymbol{\Phi},\,\boldsymbol{\Psi}),\,\boldsymbol{\mu})\| \leq 10$. The initial condition for the optimization problem and target solution are shown with a red circle and blue square, respectively.

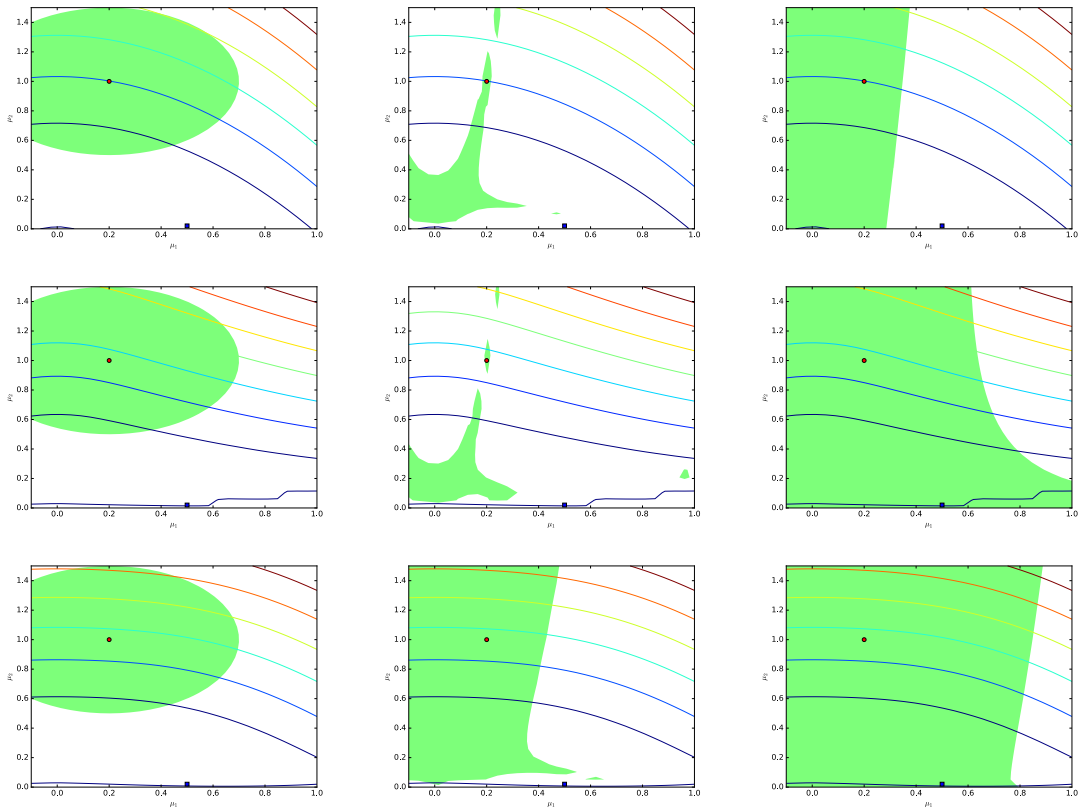


Figure 5.4: Contour of the *reduced* objective function $f(\boldsymbol{\Phi}\boldsymbol{u}_r(\boldsymbol{\mu};\,\boldsymbol{\Phi},\,\boldsymbol{\Psi}),\,\boldsymbol{\mu})$ in (5.55) in the $\mu_1-\mu_2$ plane corresponding to a slice at $\mu_3 = 0.0$. The reduced-order model employs a *LSPG* projection and the trial basis is constructed from: (top) the primal solution at $\boldsymbol{\mu}_0$, i.e., $\text{col}(\boldsymbol{\Phi}) = \text{span}\{\boldsymbol{u}(\boldsymbol{\mu}_0)\}$; (middle) the primal and adjoint solution at $\boldsymbol{\mu}_0$, i.e., $\text{col}(\boldsymbol{\Phi}) = \text{span}\{\boldsymbol{u}(\boldsymbol{\mu}_0),\,\boldsymbol{\lambda}(\boldsymbol{\mu}_0)\}$; (bottom) the primal and sensitivity solution at $\boldsymbol{\mu}_0$, i.e., $\text{col}(\boldsymbol{\Phi}) = \text{span}\left\{\boldsymbol{u}(\boldsymbol{\mu}_0),\,\dfrac{\partial \boldsymbol{u}}{\partial \boldsymbol{\mu}}(\boldsymbol{\mu}_0)\right\}$. The green shaded region indicates the areas where: (left) the Euclidean ball is bounded by 0.5, i.e., $\|\boldsymbol{\mu}-\boldsymbol{\mu}_0\| \leq 0.5$, (center) the error between the true and reduced objective function is bounded by 100, i.e., $|f(\boldsymbol{u}(\boldsymbol{\mu}),\,\boldsymbol{\mu}) - f(\boldsymbol{\Phi}\boldsymbol{u}_r(\boldsymbol{\mu};\,\boldsymbol{\Phi},\,\boldsymbol{\Psi}),\,\boldsymbol{\mu})| \leq 100$, and (right) the residual norm of the reconstructed ROM solution is bounded by 10, i.e., $\|\boldsymbol{r}(\boldsymbol{\Phi}\boldsymbol{u}_r(\boldsymbol{\mu};\,\boldsymbol{\Phi},\,\boldsymbol{\Psi}),\,\boldsymbol{\mu})\| \leq 10$. The initial condition for the optimization problem and target solution are shown with a red circle and blue square, respectively.

Table 5.1: Variants of the multifidelity trust region method based on projection-based reduced-order models introduced in Algorithms 11 and 12. The first three methods are not guaranteed to be globally convergent since they do not necessarily satisfy the gradient condition (3.15). The methods that employ the traditional trust region employ two trust region subproblem solvers: an exact solver based on the interior point method in Algorithm 3 and the inexact Steihaug-Toint CG solver. The methods that employ the residual-based trust region rely on the exact interior point solver in Algorithm 3. The interior point solver considered in this section uses Newton-CG to solve the unconstrained subproblem (instead of BFGS) for fair comparison with the second-order Steihaug-Toint CG. The snapshot matrices $\mathbf{U}_k$, $\mathbf{W}_k$, $\mathbf{Z}_k$ consist of state, sensitivity, and adjoint snapshots, respectively, of the high-dimensional model at all *previous* trust region centers, i.e., $\boldsymbol{\mu}_0, \ldots, \boldsymbol{\mu}_{k-1}$.

Name	Reduced basis ($\boldsymbol{\Phi}_k$)	TR constraint ($\vartheta_k(\boldsymbol{\mu})$)	TR solver
prim-etr-intpt	PODSP($\boldsymbol{u}(\boldsymbol{\mu}_k)$, $\mathbf{U}_k$)	residual-based (5.3)	Intpt Newton-CG
prim-ctr-intpt	PODSP($\boldsymbol{u}(\boldsymbol{\mu}_k)$, $\mathbf{U}_k$)	$\|\boldsymbol{\mu} - \boldsymbol{\mu}_k\|$	Intpt Newton-CG
prim-ctr-stcg	PODSP($\boldsymbol{u}(\boldsymbol{\mu}_k)$, $\mathbf{U}_k$)	$\|\boldsymbol{\mu} - \boldsymbol{\mu}_k\|$	Steihaug-Toint CG
sens-etr-intpt	PODHSP($\boldsymbol{u}(\boldsymbol{\mu}_k)$, $\mathbf{U}_k$, $\frac{\partial \boldsymbol{u}}{\partial \boldsymbol{\mu}}(\boldsymbol{\mu}_k)$, $\mathbf{W}_k$)	residual-based (5.3)	Intpt Newton-CG
sens-ctr-intpt	PODHSP($\boldsymbol{u}(\boldsymbol{\mu}_k)$, $\mathbf{U}_k$, $\frac{\partial \boldsymbol{u}}{\partial \boldsymbol{\mu}}(\boldsymbol{\mu}_k)$, $\mathbf{W}_k$)	$\|\boldsymbol{\mu} - \boldsymbol{\mu}_k\|$	Intpt Newton-CG
sens-ctr-stcg	PODHSP($\boldsymbol{u}(\boldsymbol{\mu}_k)$, $\mathbf{U}_k$, $\frac{\partial \boldsymbol{u}}{\partial \boldsymbol{\mu}}(\boldsymbol{\mu}_k)$, $\mathbf{W}_k$)	$\|\boldsymbol{\mu} - \boldsymbol{\mu}_k\|$	Steihaug-Toint CG
adj-etr-intpt	PODHSP($\boldsymbol{u}(\boldsymbol{\mu}_k)$, $\mathbf{U}_k$, $\boldsymbol{\lambda}_k(\boldsymbol{\mu}_k)$, $\mathbf{Z}_k$)	residual-based (5.3)	Intpt Newton-CG
adj-ctr-intpt	PODHSP($\boldsymbol{u}(\boldsymbol{\mu}_k)$, $\mathbf{U}_k$, $\boldsymbol{\lambda}_k(\boldsymbol{\mu}_k)$, $\mathbf{Z}_k$)	$\|\boldsymbol{\mu} - \boldsymbol{\mu}_k\|$	Intpt Newton-CG
adj-ctr-stcg	PODHSP($\boldsymbol{u}(\boldsymbol{\mu}_k)$, $\mathbf{U}_k$, $\boldsymbol{\lambda}_k(\boldsymbol{\mu}_k)$, $\mathbf{Z}_k$)	$\|\boldsymbol{\mu} - \boldsymbol{\mu}_k\|$	Steihaug-Toint CG

The convergence history of the methods in Table 5.1, in terms of the objective function and gradient decrease, is provided in Figures 5.5 for reduced-order models that employ a Galerkin projection and 5.6 for reduced-order models that employ an LSPG projection. The convergence of the baseline solver, an L-BFGS linesearch method (without model reduction), is also included in the figures for comparison. All of methods in Table 5.1 based on Galerkin ROMs converge to a first-order critical point of tolerance at least 10^{-4} (9 orders of magnitude reduction from the initial control), even though global convergence cannot be rigorously established for the methods that build the reduced basis from only primal snapshots ('prim-etr-intpt', 'prim-ctr-intpt', 'prim-ctr-stcg'). In contrast, the methods based on LSPG ROMs that build the reduced basis from primal and adjoint snapshots ('adj-etr-intpt', 'adj-ctr-intpt', 'adj-ctr-stcg') do not converge. These methods are supposed to be globally convergent since the inclusion of adjoint snapshots ensures the error conditions (3.14) and (3.15) holds. The failure of these methods is attributed to *failed* trust region subproblem solves that results from using inconsistent gradients away from trust region centers. Figures 5.5 and 5.6 lead to two more observations. First, all methods converge faster, in terms of major iterations, when exact trust region subproblem solvers are used. Later in this section, the convergence rate will be assessed in terms of a cost metric that accounts for the cost of each major iteration in the respective methods. Second, the methods that include more information exhibit faster convergence. For example, the methods that incorporate sensitivity information converge faster than those that incorporate adjoint information which converge faster than those that consider solely primal snapshots. In addition to the sensitivities providing more information than adjoints (there are 3 sensitivities and 1 adjoint for this problem), the information is also *richer* since they equip the basis with first order information

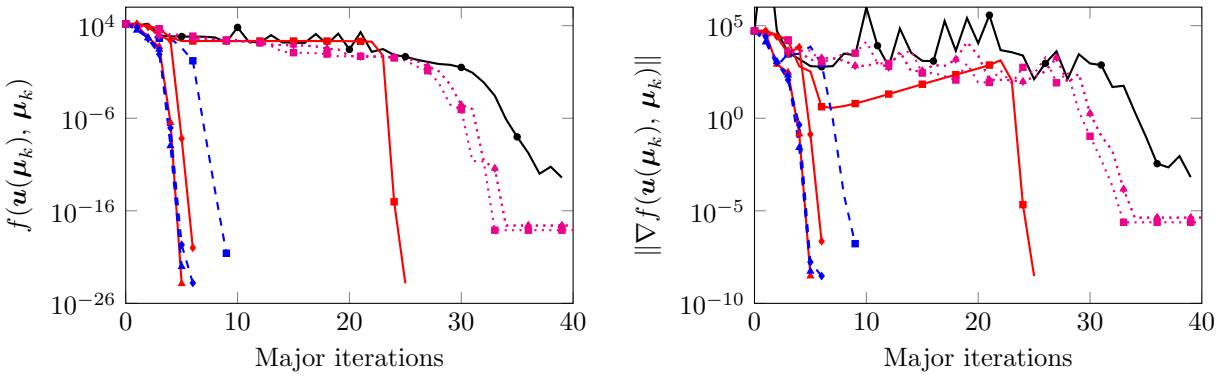


Figure 5.5: Convergence history of various optimization solvers for optimal control of the inviscid Burgers' equation when *Galerkin* reduced-order model defines the approximation model. Optimization solvers considered: L-BFGS solver with only HDM evaluations (—●—), prim-etr-intpt (—■—), prim-ctr-intpt (- -■- -), prim-ctr-stcg (···■···), sens-etr-intpt (—▲—), sens-ctr-intpt (- -▲- -), sens-ctr-stcg (···▲···), adj-etr-intpt (—♦—), adj-ctr-intpt (- -♦- -), adj-ctr-stcg (···♦···).

[52, 210].

The increased convergence rate, in terms of major iterations (and therefore HDM evaluations), of Algorithms 11 and 12 comes at the price of a large number of ROM evaluations. Figure 5.7 shows the cumulative number of primal ROM queries as a function of major iteration and a histogram of the number of primal ROM evaluations at a given reduced basis size ($k_{\boldsymbol{u}}$). The methods that use the *residual-based* trust region constraint constitute more difficult trust region subproblems and require more ROM evaluations than those that use a traditional trust region. The benefit of using the residual-based trust region is fewer major iterations, and thus HDM queries (Figures 5.5 and 5.6). Another observation is that, as expected, the inexact trust region solver (Steihaug-Toint CG) requires far fewer ROM queries than the exact solver (interior point Newton-CG), at the cost of additional major iterations (HDM evaluations).

To assess the speedups that can be realized by the variants of the proposed ROM-based trust region methods in Table 5.1, the following simplified cost model is introduced

$$C = n_{hp} + n_{hs} + \tau^{-1}(n_{rp} + n_{rs}) \tag{5.58}$$

where C is the total cost associated with a particular method in the units of *equivalent number of primal HDM queries*, n_{hp} is the number of primal HDM queries, n_{hs} is the number of sensitivity HDM queries, n_{rp} is the number of primal ROM queries, n_{rs} is the number of sensitivity ROM queries, and τ is the ratio of the cost of a primal HDM query to a primal ROM query. This cost model assume the cost of computing the primal HDM (ROM) solution is the same as computing all three sensitivities. Under this cost model, Figure 5.8 contains the convergence rates of the various algorithms as a function of cost for three values of τ: two moderate values for the expected speedup of the reduced-order model ($\tau = 20, 50$) and the asymptotic case of a *free* reduced-order model ($\tau = \infty$). All variants of the trust region method outperform the baseline L-BFGS method, with

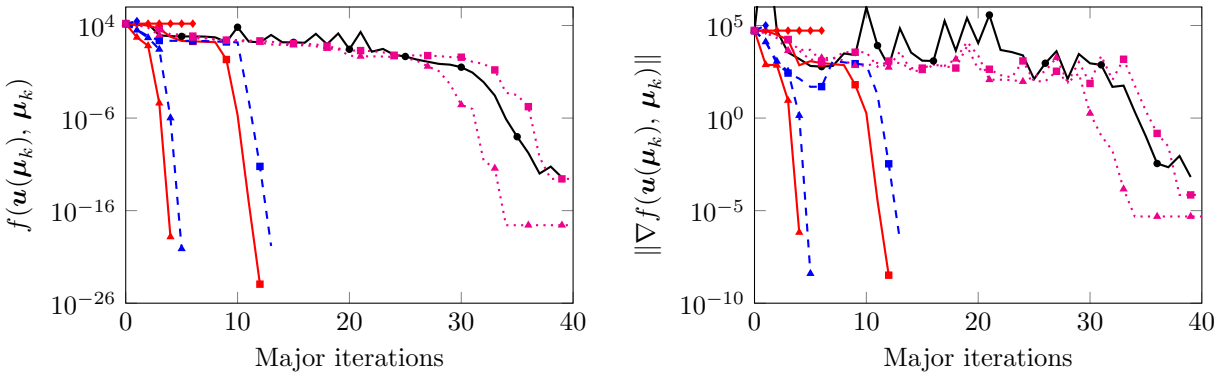


Figure 5.6: Convergence history of various optimization solvers for optimal control of the inviscid Burgers' equation when *LSPG* reduced-order model defines the approximation model. Optimization solvers considered: L-BFGS solver with only HDM evaluations (—●—), prim-etr-intpt (—■—), prim-ctr-intpt (- -■- -), prim-ctr-stcg (···■···), sens-etr-intpt (—▲—), sens-ctr-intpt (- -▲- -), sens-ctr-stcg (···▲···), adj-etr-intpt (—◆—), adj-ctr-intpt (- -◆- -), adj-ctr-stcg (···◆···).

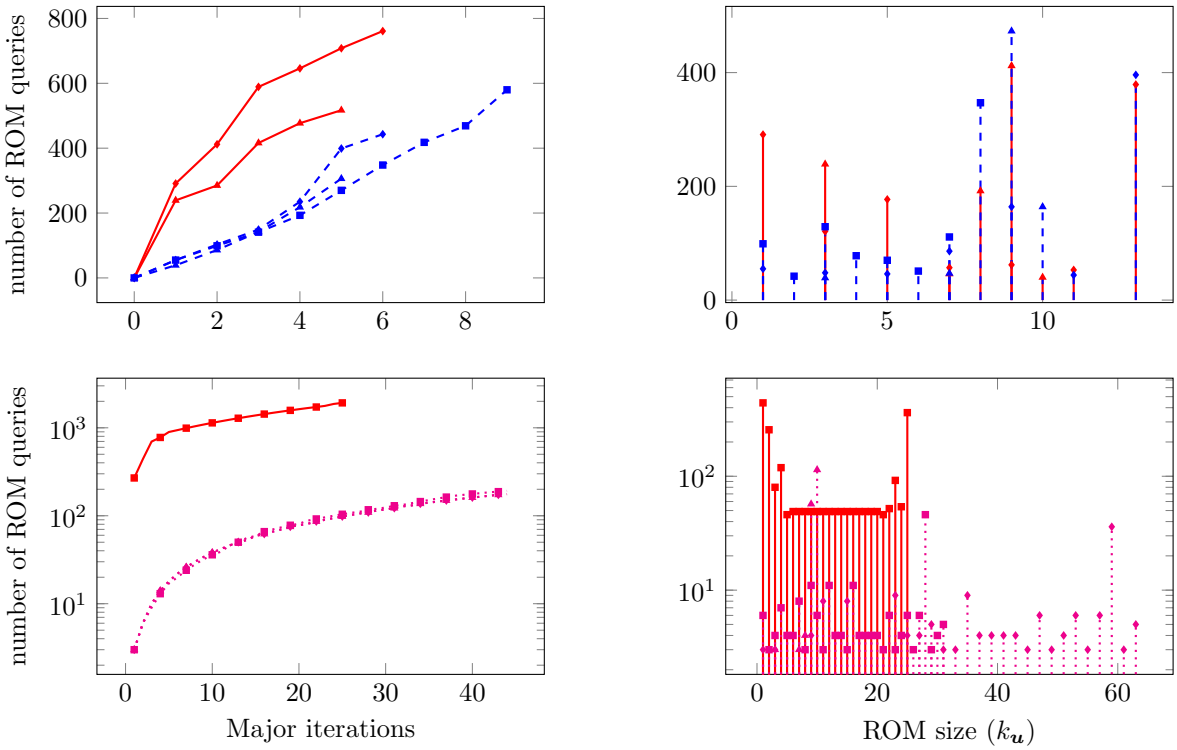


Figure 5.7: *Left*: Cumulative number of primal ROM queries as a function of major iteration in the trust region algorithm based on reduced-order models (Algorithm 11) as applied to optimal control of the inviscid Burgers' equation. *Right*: Histogram of the number of primal ROM queries at a given basis size. Data separated into the top and bottom rows to deal with the disparate x-scales. All reduced-order models use a Galerkin projection. Optimization solvers considered: prim-etr-intpt (—■—), prim-ctr-intpt (- -■- -), prim-ctr-stcg (···■···), sens-etr-intpt (—▲—), sens-ctr-intpt (- -▲- -), sens-ctr-stcg (···▲···), adj-etr-intpt (—◆—), adj-ctr-intpt (- -◆- -), adj-ctr-stcg (···◆···).

the variants based on exact trust region solvers ('sens-etr-intpt' and 'sens-ctr-intpt') outperforming the inexact solver ('sens-ctr-intpt'), even if ROM queries are only 20× faster than HDM queries. Depending on the speedup of the ROM, a given value of the objective function or gradient can be achieved by methods 'sens-etr-intpt' or 'sens-ctr-intpt' at roughly $10-50\%$ the cost required by the baseline method.

The section closes with a study of the *convergence behavior* of the trust region method that uses a residual-based trust region constraint when Algorithms 11 (fully converged solutions as snapshots and for trust region assessment) and 12 (partially converged solutions as snapshots and for trust region assessment) are used. Figure 5.9 contains the convergence history of the objective function and approximation model at trust region centers and candidate steps. Figures 5.10 and 5.11 contain the same information for the gradient and trust region constraint, respectively. From these figures, the model is first-order consistent at trust region centers for Algorithm 11 (left plots) since the basis is constructed with the span-preserving variant of POD (Algorithm 7) and uses fully converged snapshots. This is not the case for Algorithm 12 (right plots) that uses partially converged snapshots. Despite relatively poor agreement of the model and objective (and the corresponding gradients) at trust region centers and candidate steps, rapid progress is made toward the optimal solution. From Figure 5.11, the trust region constraints are *active* at early iterations of the trust region algorithm and inactive later. This suggests that, as the optimal solution is approached, the reduced-order model is only queried in regions of the parameter space where it is very accurate, i.e., near training points. Finally, Algorithm 12 requires one additional iteration than Algorithm 11 to converge to a similar tolerance. This is expected since Algorithm 12 utilizes partially converged solutions in the construction of the reduced basis, an additional level of inexactness. These observations are verified in Tables 5.2–5.5 that contains the convergence history of the relevant trust region quantities for the variants 'sens-etr-intpt' and 'sens-ctr-stcg' of Algorithms 11 and 12.

5.5.2 Optimal Control of 1D Viscous Burgers' Equation

The investigation into the methods introduced in this chapter continues in this section with an emphasis on problems where the number of parameters is sufficiently large that gradients must be computed with the adjoint method. The model PDE-constrained optimization problem considered is optimal control of the steady, viscous, one-dimensional Burgers' equation. The optimization problem takes the form

$$\underset{\boldsymbol{\mu}\in\mathbb{R}^{n_{\boldsymbol{\mu}}}}{\text{minimize}} \quad \left[\int_0^1 \frac{1}{2}(u(\boldsymbol{\mu},\, x) - \bar{u}(x))^2\, dx + \frac{\alpha}{2}\int_0^1 z(\boldsymbol{\mu},\, x)^2\, dx\right] \tag{5.59}$$

where $u(\boldsymbol{\mu},\, x)$ is the solution of the viscous Burgers' equation with a general parametrization of the control $z(\boldsymbol{\mu},\, x)$

$$\begin{aligned} -\nu\partial_{xx}u(\boldsymbol{\mu},\, x) + u(\boldsymbol{\mu},\, x)\partial_x u(\boldsymbol{\mu},\, x) = z(\boldsymbol{\mu},\, x) \quad x\in(0,\, 1) \\ u(\boldsymbol{\mu},\, 0) = 1 \qquad u(\boldsymbol{\mu},\, 1) = 0 \end{aligned} \tag{5.60}$$

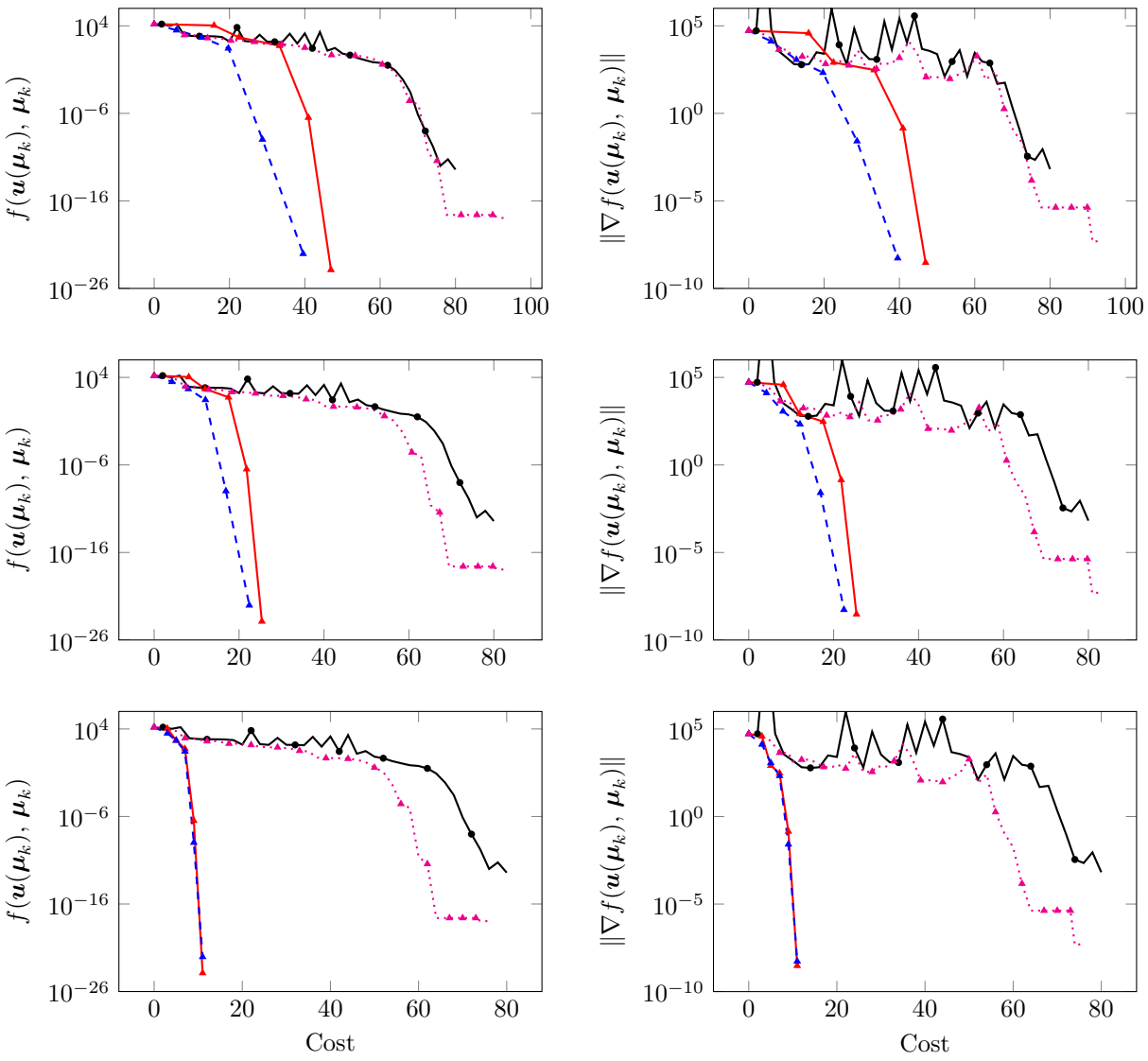


Figure 5.8: Convergence of the objective function (left) and gradient (right) as a function of the cost metric in (5.58) for several values of the speedup factor of the reduced-order model: $\tau = 20$ (top row), $\tau = 50$ (middle row), $\tau = \infty$ (bottom row) for optimal control of the inviscid Burgers' equation. All reduced-order models use a Galerkin projection. Optimization solvers considered: L-BFGS solver with only HDM evaluations (—●—), sens-etr-intpt (—▲—), sens-ctr-intpt (- -▲- -), sens-ctr-stcg (···▲···).

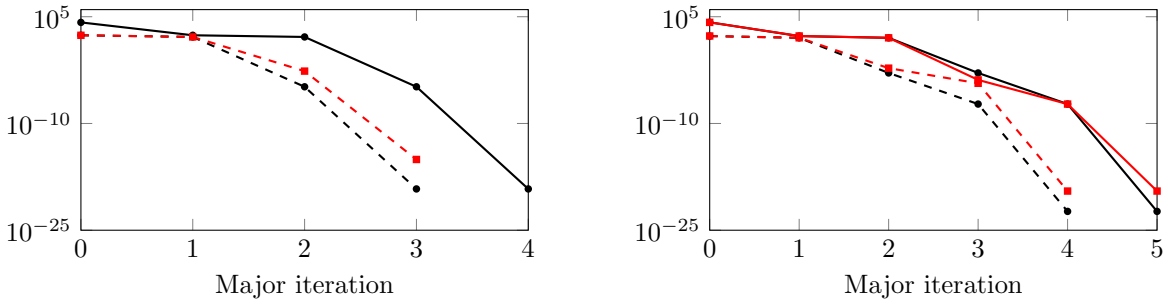


Figure 5.9: Convergence history of the objective quantities for optimal control of the inviscid Burgers' equation using Algorithm 11 (left – fully converged solutions as snapshots and in the evaluation of trust region steps) and Algorithm 12 (right – partially converged solutions as snapshots and in the evaluation of trust region steps): $F(\boldsymbol{\mu}_k)$ (—•—), $F(\hat{\boldsymbol{\mu}}_k)$ (-•-), $m_k(\boldsymbol{\mu}_k)$ (—■—), $m_k(\hat{\boldsymbol{\mu}}_k)$ (-■-). The variant 'sens-etr-intpt' (Table 5.1) of the multifidelity trust region algorithm with Galerkin-based reduced-order models is used. Since the approximation model in the left plot is first-order consistent at trust region centers, $m_k(\boldsymbol{\mu}_k)$ is omitted.

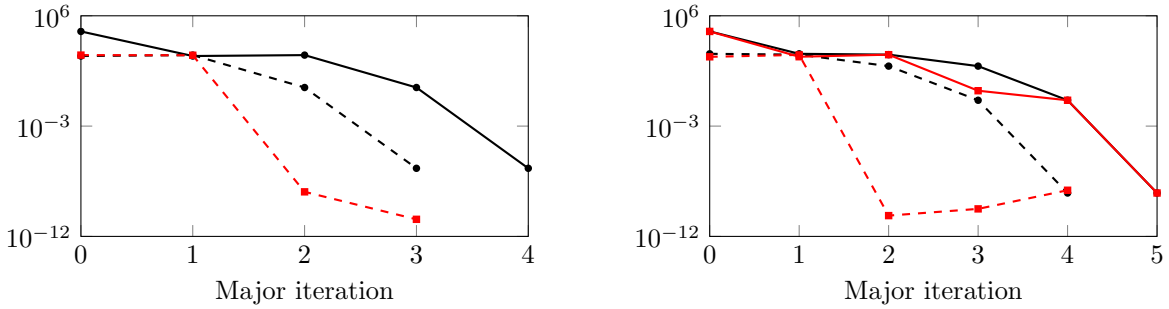


Figure 5.10: Convergence history of the gradient quantities for optimal control of the inviscid Burgers' equation using Algorithm 11 (left – fully converged solutions as snapshots and in the evaluation of trust region steps) and Algorithm 12 (right – partially converged solutions as snapshots and in the evaluation of trust region steps): $\|\nabla F(\boldsymbol{\mu}_k)\|$ (—•—), $\|\nabla F(\hat{\boldsymbol{\mu}}_k)\|$ (-•-), $\|\nabla m_k(\boldsymbol{\mu}_k)\|$ (—■—), $\|\nabla m_k(\hat{\boldsymbol{\mu}}_k)\|$ (-■-). The variant 'sens-etr-intpt' (Table 5.1) of the multifidelity trust region algorithm with Galerkin-based reduced-order models is used. Since the approximation model in the left plot is first-order consistent at trust region centers, $\|\nabla m_k(\boldsymbol{\mu}_k)\|$ is omitted.

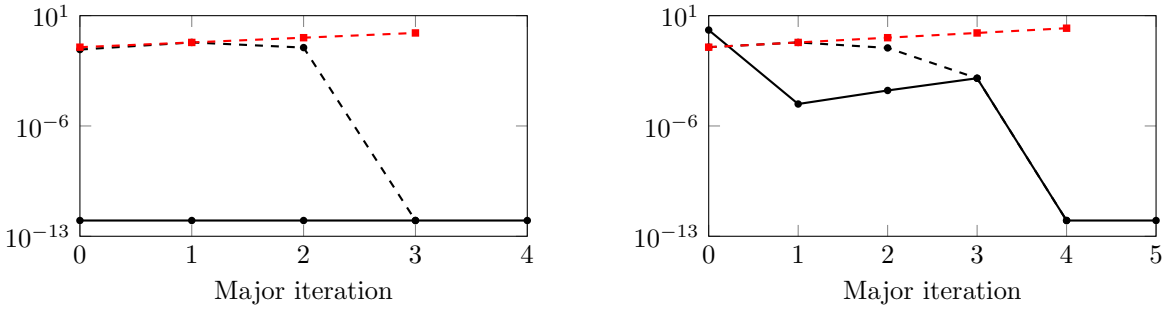


Figure 5.11: Convergence history of the constraint quantities for optimal control of the inviscid Burgers' equation using Algorithm 11 (left – fully converged solutions as snapshots and in the evaluation of trust region steps) and Algorithm 12 (right – partially converged solutions as snapshots and in the evaluation of trust region steps): $\vartheta_k(\boldsymbol{\mu}_k)$ (—•—), $\vartheta_k(\hat{\boldsymbol{\mu}}_k)$ (- •-), Δ_k (- ▪-). The variant 'sens-etr-intpt' (Table 5.1) of the multifidelity trust region algorithm with Galerkin-based reduced-order models is used.

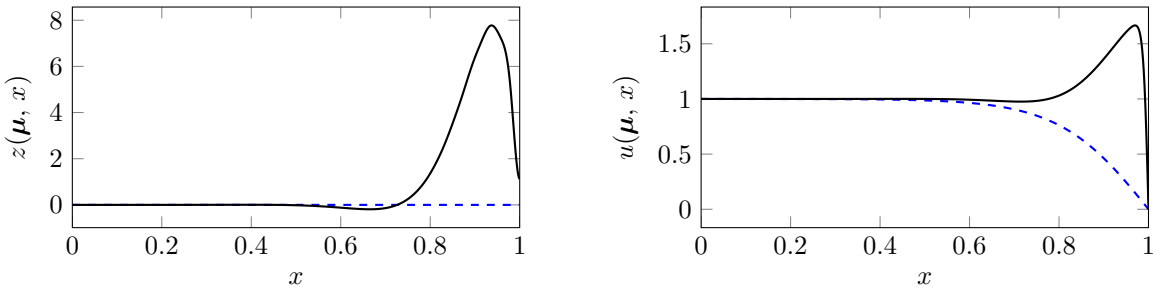


Figure 5.12: Control (left) and corresponding solution (right) of the viscous Burgers' equation in (5.60) at: the initial guess for the optimization problem (- - -) and the optimal solution of (5.59) (——).

and $\bar{u}(x)$ is the target state. The viscosity is fixed at $\nu = 10^{-2}$ and the PDE is discretized with 1000 linear finite elements for a state space of dimension $N_{\boldsymbol{u}} = 999$, after application of the essential boundary conditions. The target state is chosen as the constant solution $\bar{u}(x) \equiv 1$, which is not reachable due to the boundary conditions on the PDE. The control is parametrized with 50 cubic splines with *clamped* boundary conditions for a total of 53 optimization variables[4], i.e., $N_{\boldsymbol{\mu}} = 53$. The control is parametrized in this way, instead of the standard approach [78, 96, 108, 109] of interpolating the control using the underlying finite element shape functions to avoid a parameter space whose dimension is comparable to that of the state space, i.e., $N_{\boldsymbol{\mu}} = \mathcal{O}(N_{\boldsymbol{u}})$, as this case requires special consideration (Appendix C).

Even though the number of parameters does not scale with the dimension of the state space, the large number of parameter ($N_{\boldsymbol{\mu}} = 53$) calls for the adjoint approach to compute gradients of

[4]The optimization variables are the value of each spline knot (the location of each knot is fixed) and the slope of the curve its boundaries.

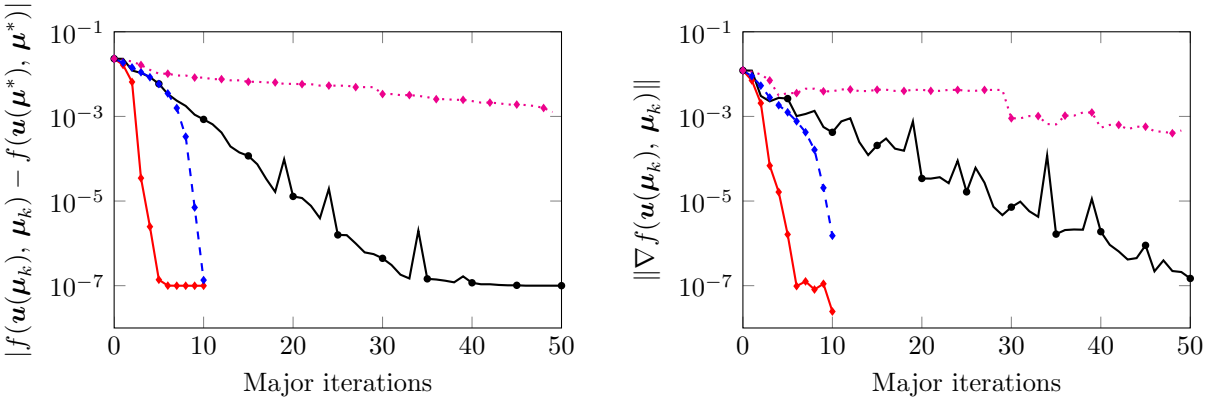


Figure 5.13: Convergence history of various optimization solvers for optimal control of the viscous Burgers' equation when *Galerkin* reduced-order model defines the approximation model. Optimization solvers considered: L-BFGS solver with only HDM evaluations (—●—), adj-etr-intpt (—♦—), adj-ctr-intpt (- -♦- -), adj-ctr-stcg (···♦···).

quantities of interest; therefore, this section only studies 'adj-etr-intpt', 'adj-ctr-intpt', and 'adj-ctr-stcg' from Table 5.1. Furthermore, this section only considers reduced-order models based on a *Galerkin* projection to ensure *consistent* gradients, which is a particularly important consideration when the number of parameters is large. The convergence of these methods, as a function of major iteration, is provided in Figure 5.13, along with the convergence of the baseline method that uses an L-BFGS method (without model reduction). The trust region method with a residual-based constraint converges most rapidly and, similar to the previous section, the methods that employ *exact* trust region solvers outperform the inexact Steihaug-Toint CG solver. In fact, the method based on the Steihaug-Toint CG solver is converging; however, after the maximum number of iterations (50) the iterates are not close enough to the solution for quadratic convergence to be realized and does not converge to the same tolerance as the other methods.

The increased convergence rate, in terms of major iterations (and therefore HDM evaluations), of Algorithm 11 comes at the price of a large number of ROM evaluations. Figure 5.14 shows the cumulative number of primal ROM queries as a function of major iteration and a histogram of the number of primal ROM evaluations at a given reduced basis size ($k_{\boldsymbol{u}}$). Similar to the previous section, the inexact solver requires far fewer ROM queries than the exact solvers. However, unlike the previous section, the number of ROM queries required by the residual-based trust region and traditional trust region are not significantly different.

To assess the speedups that can be realized by the variants of the proposed ROM-based trust region methods in Table 5.1, the following simplified cost model is introduced

$$C = n_{hp} + n_{ha}/2 + \tau^{-1}(n_{rp} + n_{ra}/2) \tag{5.61}$$

where C is the total cost associated with a particular method in the units of *equivalent number of primal HDM queries*, n_{hp} is the number of primal HDM queries, n_{ha} is the number of adjoint HDM

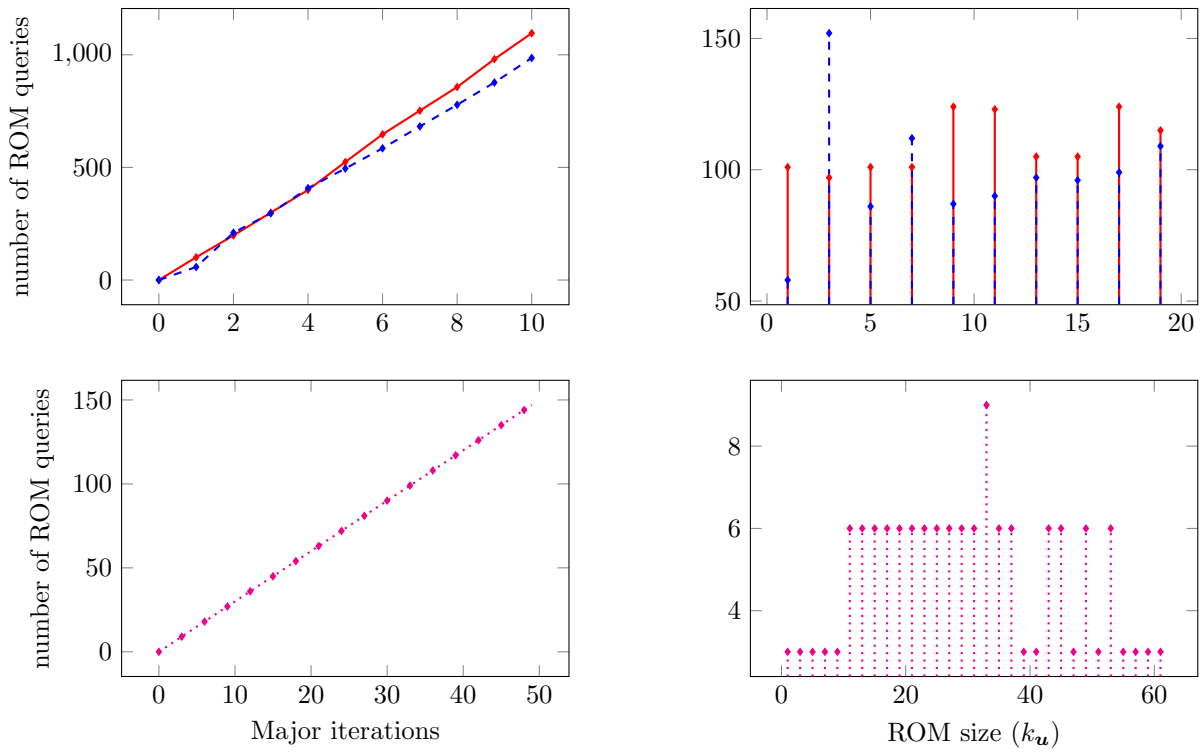


Figure 5.14: *Left*: Cumulative number of primal ROM queries as a function of major iteration in the trust region algorithm based on reduced-order models (Algorithm 11) as applied to optimal control of the viscous Burgers' equation. *Right*: Histogram of the number of primal ROM queries at a given basis size. Data separated into the top and bottom rows to deal with the disparate x-scales. All reduced-order models use a Galerkin projection. Optimization solvers considered: adj-etr-intpt (—♦—), adj-ctr-intpt (- -♦- -), adj-ctr-stcg (···♦···).

queries, n_{rp} is the number of primal ROM queries, n_{ra} is the number of adjoint ROM queries, and τ is the ratio of the cost of a primal HDM query to a primal ROM query. This cost model assume the cost of computing the primal HDM (ROM) solution is twice that of computing an adjoint solution. Under this cost model, Figure 5.15 contains the convergence rates of the various algorithms as a function of cost for three values of τ: two moderate values for the expected speedup of the reduced-order model ($\tau = 50, 100$) and the asymptotic case of a *free* reduced-order model ($\tau = \infty$). The variants of the trust region method based on the exact trust region solver ('adj-etr-intpt' and 'adj-ctr-intpt') outperform the baseline L-BFGS method, even if ROM queries are only $50\times$ faster than HDM queries. Depending on the speedup of the ROM, a given value of the objective function or gradient can be achieved by methods 'adj-etr-intpt' at less than 50% the cost required by the baseline method.

This section closes with a study of the *convergence behavior* of the trust region method that uses a residual-based trust region constraint. Figure 5.16 contains the convergence history of the objective function and approximation model (left) and their gradients (right) at trust region centers and candidate steps. The approximation model is first-order consistent at trust region centers since the basis is constructed with the span-preserving variant of POD (Algorithm 7) and uses fully converged snapshots. Despite relatively poor agreement of the model and objective (and the corresponding gradients) at the candidate steps, rapid progress is made toward the optimal solution. These observations are verified in Tables 5.6–5.7 that contains the convergence history of the relevant trust region quantities for methods 'adj-etr-intpt' and 'adj-ctr-intpt'.

5.5.3 Shape Optimization of Airfoil in Inviscid, Subsonic Flow

In this section, we consider the *inverse* shape design of an airfoil in inviscid, subsonic flow: given only the pressure distribution of a target shape—the RAE2822 airfoil, in this case—the goal is to use shape optimization to recover the underlying shape. The initial guess for the optimization problem is the symmetric NACA0012 airfoil.

Shape parametrization and problem setup

A plethora of shape parametrization techniques exist [177, 9], each with strengths and weaknesses. They typically trade-off between efficiency and flexibility. A subset of these techniques have been studied in the context of model order reduction [174]. In this work, the SDESIGN software [129, 127, 128], based on the design element approach [99, 61], is used for shape parametrization (Section 2.1.2). Here, a single "cubic" design element is used to parametrize the deformation of the NACA0012 airfoil. Such a design element has 8 control nodes. They are used to define cubic Lagrangian polynomials to describe the displacement field along the horizontal edges of the element, and linear functions to define the displacement field along its vertical edges. For this application, the set of admissible shapes is further restricted by constraining the control nodes to move in the *vertical* direction only. This results in a parametrization with 8 variables where each of them represents the *displacement* of a control node in the vertical direction. The case where all parameters are equal, $\boldsymbol{\mu} = c\mathbf{1}$ for

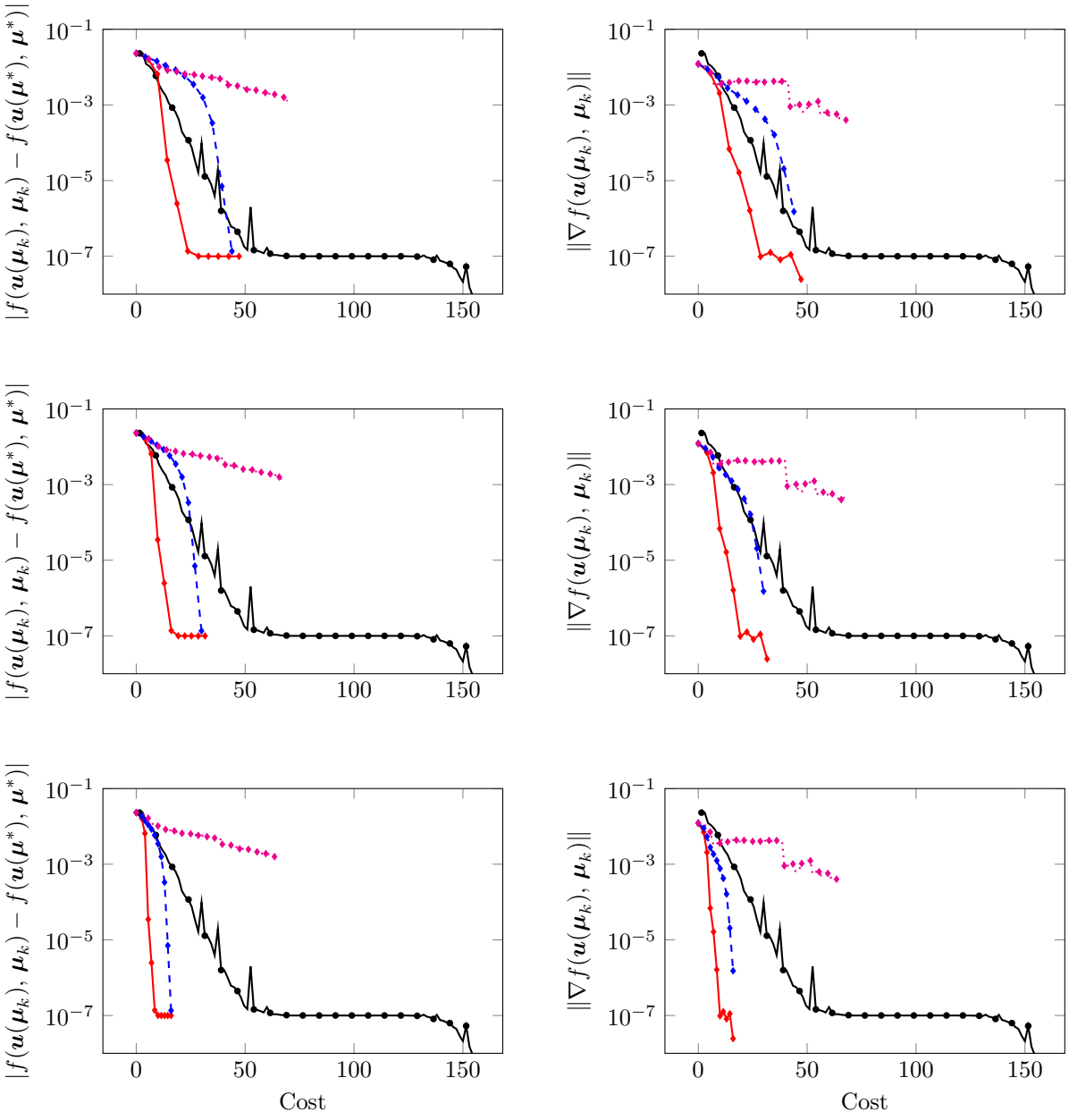


Figure 5.15: Convergence of the objective function (left) and gradient (right) as a function of the cost metric in (5.61) for several values of the speedup factor of the reduced-order model: $\tau = 50$ (top row), $\tau = 100$ (middle row), $\tau = \infty$ (bottom row) for optimal control of the viscous Burgers' equation. All reduced-order models use a Galerkin projection. Optimization solvers considered: L-BFGS solver with only HDM evaluations (—●—), adj-etr-intpt (—♦—), adj-ctr-intpt (- -♦- -), adj-ctr-stcg (···♦···).

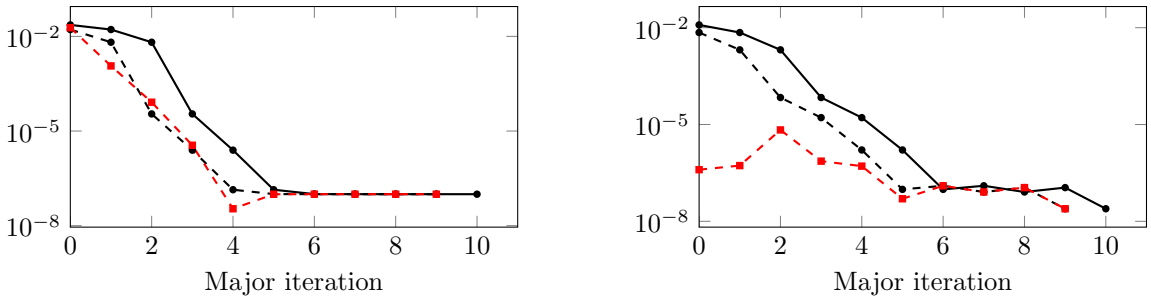


Figure 5.16: Convergence history of the objective (left) and gradient (right) quantities for optimal control of the viscous Burgers' equation using Algorithm 11 (fully converged solutions as snapshots and in the evaluation of trust region steps). *Left*: $|F(\boldsymbol{\mu}_k) - F(\boldsymbol{\mu}^*)|$ (—•—), $|F(\hat{\boldsymbol{\mu}}_k) - F(\boldsymbol{\mu}^*)|$ (- •-), $|m_k(\hat{\boldsymbol{\mu}}_k) - F(\boldsymbol{\mu}^*)|$ (- ▪-). *Right*: $\|\nabla F(\boldsymbol{\mu}_k)\|$ (—•—), $\|\nabla F(\hat{\boldsymbol{\mu}}_k)\|$ (- •-), $\|\nabla m_k(\hat{\boldsymbol{\mu}}_k)\|$ (- ▪-). The variant 'adj-etr-intpt' (Table 5.1) of the multifidelity trust region algorithm with Galerkin-based reduced-order models is used. Since the approximation model is first-order consistent at trust region centers $m_k(\boldsymbol{\mu}_k)$ and $\|\nabla m_k(\boldsymbol{\mu}_k)\|$ are omitted.

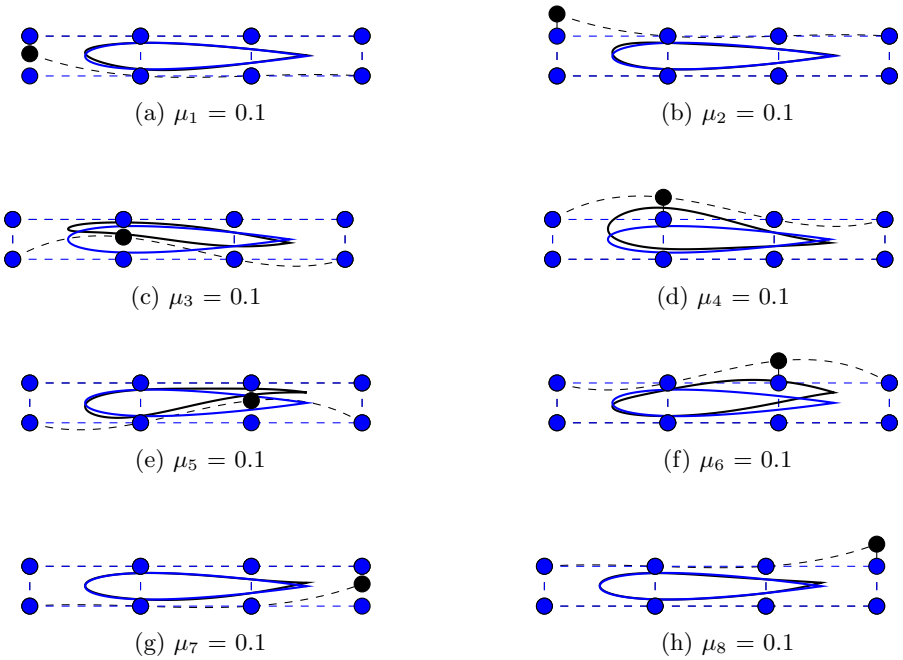
$c \in \mathbb{R}$, corresponds to a rigid translation in the vertical direction. Because such a translation does not affect the definition of a shape, it is eliminated by constraining one of the displacement variables to zero. Furthermore, because the control nodes are allowed to move only in the vertical direction, rigid rotations are automatically eliminated. A visualization of the vertices of the design element and the deformation induced by perturbing each design variable is given in Figure 5.17. While SDESIGN is used to deform the surface nodes of the airfoil, a robust mesh motion algorithm based on a structural analogy is used to deform the surrounding body-fitted CFD mesh accordingly.

The flow over the airfoil is modeled using the compressible Euler equations, and these are solved numerically using AERO-F [68]. Because this flow solver is three-dimensional, the two-dimensional fluid domain around the airfoil is represented as a slice of a three-dimensional domain. This slice is discretized using a body-fitted CFD mesh with 54 816 tetrahedra and 19 296 nodes (Figure 5.18a). Specifically, the flow equations are semi-discretized by AERO-F on this CFD mesh using a second-order finite volume method based on Roe's flux [169].

For each airfoil configuration generated during the iterative optimization procedure, the steady state solution of the flow problem is computed iteratively using pseudo-transient continuation. For this purpose, each sought-after steady state solution is initialized using the *best* previously computed steady state solution available in the database[5]. The best steady state solution is defined here as that steady state solution available in the database which, for the given airfoil configuration, minimizes the residual of the discretized steady state Euler equations. Because the database of steady state flow solutions is initially empty, the iterative computation of the steady state flow over the initial shape—in this case, that of the NACA0012 airfoil—is initialized with the uniform flow solution.

The trust region method described in this chapter that employs ROMs as the approximation model is used to solve the aerodynamic shape optimization problem. At each HDM sample, the

[5]In this context, the database refers to the flow solutions computed for all shapes previously visited by the optimization trajectory.



(a) $\mu_1 = 0.1$ (b) $\mu_2 = 0.1$

(c) $\mu_3 = 0.1$ (d) $\mu_4 = 0.1$

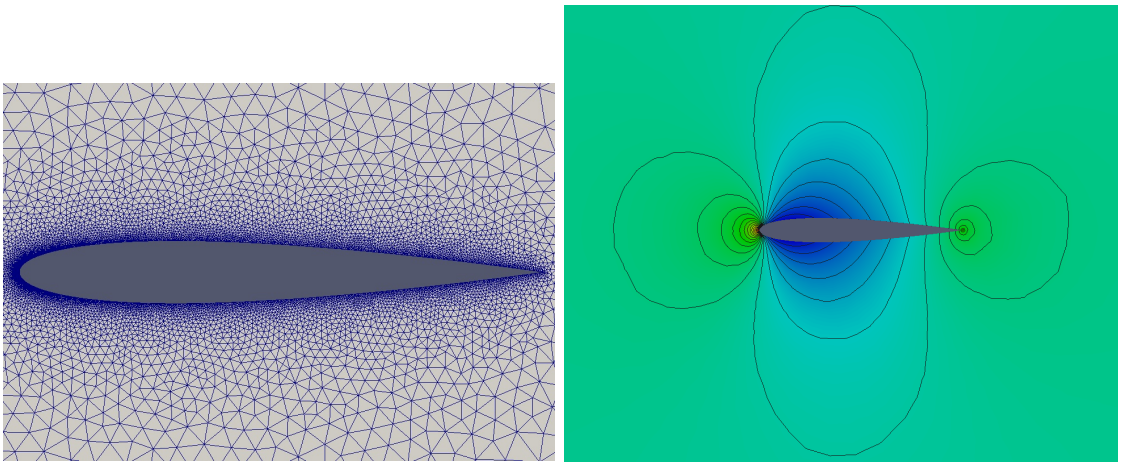
(e) $\mu_5 = 0.1$ (f) $\mu_6 = 0.1$

(g) $\mu_7 = 0.1$ (h) $\mu_8 = 0.1$

Figure 5.17: Shape parametrization of a NACA0012 airfoil using a *cubic* design element (the notation μ_i designates the i-th component of the vector $\boldsymbol{\mu}$ which refers to the i-th displacement degree of freedom of the shape parametrization)

steady state solution and sensitivities with respect to shape parameters are computed and used as snapshots. As the chosen shape parametrization has 8 parameters, 9 snapshots are generated per HDM sample: one snapshot corresponding to the steady state solution and 8 solution sensitivities. A ROB is extracted from these snapshots using the heterogeneous span-preserving variant of the POD method in Algorithm 7. Because very few snapshots are generated for this problem, the truncation step in the POD algorithm is skipped. Consequently, the size of the constructed ROB is $k_{\boldsymbol{u}} = 9s$, where s is the number of sampled HDMs. The nonlinear least-squares problem describing the ROM is solved using the Gauss-Newton method equipped with a backtracking linesearch algorithm. The python interface to the SNOPT [70] software, pyOpt [151], is used to solve the optimization problem itself.

At this point, it is noted that since the exact profile of the RAE2822 airfoil does not lie in the space of admissible airfoil profiles defined by the cubic design element parametrization, it is approximated by the closest admissible profile. This approximation is referred to in the remainder of this section as the Cub-RAE2822 airfoil. It is graphically depicted in Figure 5.19 which also shows the pressure isolines computed for this airfoil at the free-stream Mach number $M_\infty = 0.5$ and angle of attack $\alpha = 0.0°$.



(a) CFD mesh for the NACA0012 airfoil (undeformed, 19 296 nodes)

(b) Pressure field ($M_\infty = 0.5$, $\alpha = 0.0°$)

Figure 5.18: NACA0012 mesh and pressure distribution at Mach 0.5 and zero angle of attack.

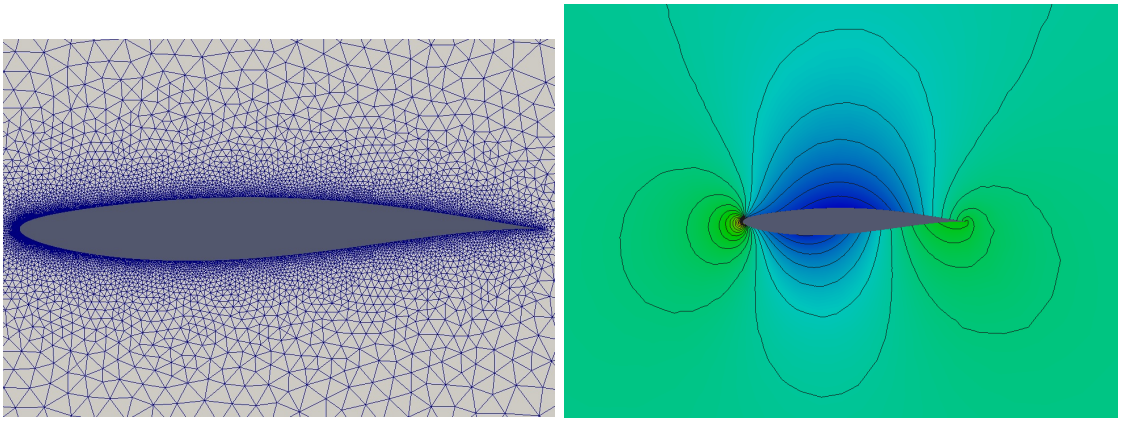
Subsonic inverse design

The free-stream conditions of interest are set to the subsonic Mach number $M_\infty = 0.5$ and zero angle of attack ($\alpha = 0°$), and the following optimization problem is considered

$$\begin{aligned} \underset{\boldsymbol{\mu}\in\mathbb{R}^{N_{\boldsymbol{\mu}}}}{\text{minimize}} \quad & \frac{1}{2}\left\|\boldsymbol{p}(\boldsymbol{u}(\boldsymbol{\mu})) - \boldsymbol{p}(\boldsymbol{u}(\boldsymbol{\mu}^{\text{RAE2822}}))\right\|_2^2 \\ \text{subject to} \quad & \mu_3 = 0 \\ & \boldsymbol{\mu}_{\boldsymbol{l}} \leq \boldsymbol{\mu} \leq \boldsymbol{\mu}_{\boldsymbol{u}} \end{aligned} \tag{5.62}$$

where $\boldsymbol{p}(\boldsymbol{u})$ is the vector of nodal pressures, and $\boldsymbol{\mu}^{\text{RAE2822}}$ designates the parameter solution vector morphing the NACA0012 airfoil into the Cub-RAE2822 airfoil. The first constraint is introduced to eliminate the rigid body translation in the vertical direction as discussed in the previous section. The box constraints prohibit the optimization trajectory from going through highly distorted shapes that would cause the flow solver to fail.

To obtain a reference solution that can be used for assessing the performance of the proposed ROM-based optimization method, problem (5.62) is first solved using the HDM as the constraining PDE. In this case, the optimizer is found to reduce the initial value of the objective function by 9 orders of magnitude, before numerical difficulties cause it to terminate (Figure 5.20). Relevant statistics associated with this HDM-based reference solution of the optimization problem are gathered in Table 5.8. Essentially, 24 optimization iterations are required to obtain a solution with a relative error well below 0.1%. These iterations incur a total of 29 HDM queries (including those associated with the linesearch iterations). Figure 5.21 shows the pressure distribution associated with this reference solution matches the target pressure distribution very well.



(a) CFD mesh for the Cub-RAE2822 airfoil (b) Pressure field ($M_\infty = 0.5$, $\alpha = 0.0^\circ$)

Figure 5.19: Cub-RAE2822 mesh and pressure isolines computed at Mach 0.5 and zero angle of attack.

Next, the ROM-based trust region method developed in this chapter and summarized in Algorithm 11 is applied to solve problem (5.62), which solves a sequence of trust region subproblems of the form

$$\begin{aligned} \underset{\boldsymbol{\mu}\in\mathbb{R}^{N_{\boldsymbol{\mu}}}}{\text{minimize}} \quad & \frac{1}{2}\left\|\boldsymbol{p}(\boldsymbol{\Phi}_k\boldsymbol{u}_r(\boldsymbol{\mu})) - \boldsymbol{p}(\boldsymbol{u}(\boldsymbol{\mu}^{\text{RAE2822}}))\right\|_2^2 \\ \text{subject to} \quad & \mu_3 = 0 \\ & \boldsymbol{\mu_l} \leq \boldsymbol{\mu} \leq \boldsymbol{\mu_u} \\ & \frac{1}{2}\left\|\boldsymbol{r}(\boldsymbol{\Phi}_k\boldsymbol{u}_r(\boldsymbol{\mu}), \boldsymbol{\mu})\right\|_2^2 \leq \Delta_k. \end{aligned} \tag{5.63}$$

The HDM is sampled at the initial configuration and the resulting 9 snapshots are used to build a ROB using Algorithm 7, without truncation. The resulting ROB is used to construct a reduced-order model based on a LSPG projection and the corresponding minimum-residual sensitivity model to solve (5.63). Indeed, as the minimum-residual sensitivity computation described in Section 4.1.2 is not consistent with the true reduced sensitivities for large residuals, convergence of the optimization problem is not guaranteed. To address this issue, an upper bound is set on the number of optimization iterations (25 in this case) and the goal of the reduced optimization problem is set to finding an improvement to the current solution before updating the ROB. The HDM is sampled at the termination point of each reduced optimization problem yielding 9 additional snapshots which are appended to the ROB using Algorithm 9. Linear independence of the basis is maintained by truncating vectors corresponding to singular values below some tolerance. For the present application, such truncation was not necessary as the snapshots added to the ROB at a given iteration were not contained in the span of the snapshots from previous iterations.

Using only 7 HDM samples, the progressive ROM optimization framework reduces the initial pressure discrepancy by 18 orders of magnitude, to essentially machine zero. Interestingly, this is 4 times fewer HDM queries than required by the HDM-based optimization. Figure 5.21 shows that

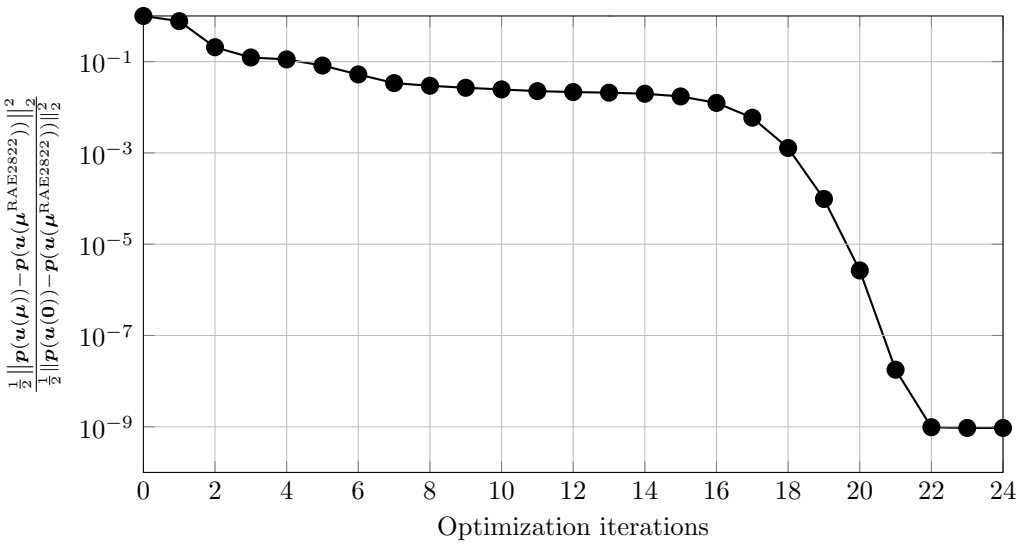


Figure 5.20: Progression of the objective function during the HDM-based optimization. The initial guess is defined as the 0th optimization iteration.

both the shape of the airfoil and the associated pressure distribution discovered by the ROM-based optimization method match the target shape and pressure distribution very well.

Surprisingly, the ROM-based optimization process achieves a *lower* value of the objective function than the HDM-based one. This can be traced to convergence tolerance on the HDM sensitivity analysis. The HDM-based sensitivities are obtained by solving the multiple right-hand side linear system of equations in (2.87) using GMRES. The convergence tolerance is $\|\mathbf{A}\mathbf{x}-\mathbf{b}\|_2 \leq \gamma\|\mathbf{b}\|_2$ for solving the linear system of equations $\mathbf{A}\mathbf{x}=\mathbf{b}$, with $\gamma = 10^{-10}$ in this case. If $\|\mathbf{b}\|$ is large ($\mathbf{b} = \partial\boldsymbol{r}/\partial\boldsymbol{\mu}$ in this case), the convergence requirement may be rather flexible. Conversely, the minimum-residual ROM sensitivities in (4.28) are solved to machine precision using a direct QR factorization.

Recall from Chapter 4 that the minimum-residual reduced sensitivities approach the true sensitivities for LSPG projection as the HDM residual approaches zero. Figure 5.24 verifies that the HDM residual is small after 6 HDM samples are taken, which implies the minimum-residual ROM sensitivities are (nearly) consistent with the true ROM sensitivities. This consistency will guarantee convergence of the reduced optimization problem when using a globally convergent optimization solver. Additionally, the small HDM residual implies that the ROM is highly accurate in this region, making it likely that the reduced optimization problem will converge to a point close to the true optimum.

Figure 5.22 reports on the evolution of the objective function with the number of optimization iterations, and marks each new HDM query along the optimization trajectory. The reader can observe that the proposed ROM-based optimization method performs a total of 160 trust region subproblem iterations (Figure 5.23) requiring 346 ROM evaluations (see Table 5.8) and 7 HDM

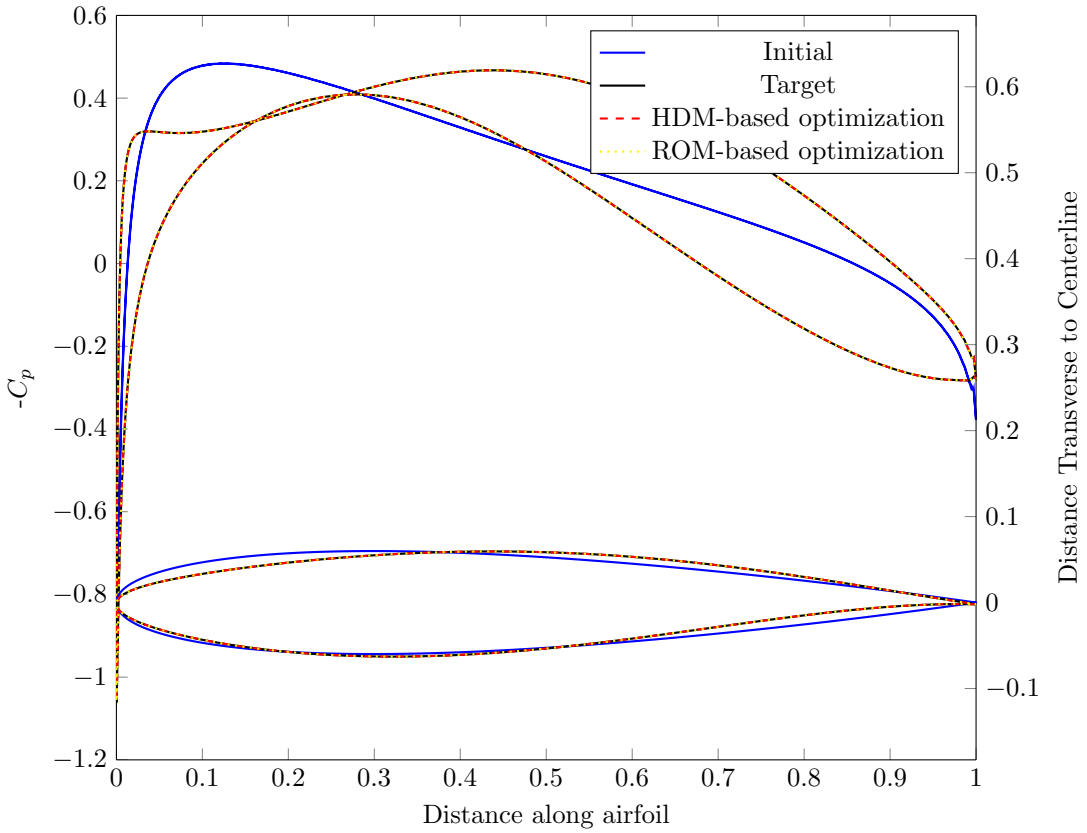


Figure 5.21: Subsonic inverse design of the airfoil Cub-RAE2822: initial shape (NACA0012) and associated C_p function, and final shape (Cub-RAE2822) and associated C_p functions delivered by the HDM- and ROM-based optimizations, respectively.

queries. From a computational complexity viewpoint, this compares favorably with the 24 HDM-based optimization iterations requiring 29 HDM queries (see Table 5.8). Figure 5.23 graphically depicts the progression of the *reduced* objective function across all reduced optimization problems using a dashed line to indicate a new HDM sample and a subsequent update of the ROB. For each optimization problem, it also reports the size of the ROM.

Finally, Figure 5.24 shows the evolution of the HDM residual evaluated at the solution of the ROM—which is an indicator of the ROM error—across all reduced optimization problems, along with the trust region radius Δ_k. It is common practice in nonlinear programming software to allow violation of nonlinear constraints during an optimization procedure, which explains the residual bound violation seen in this figure. Figure 5.24 also shows that the ROM solution coincides with the HDM solution at the initial condition of each optimization problem, as expected from the interpolation property of minimum-residual reduced-order models. In the first few major iterations that are far the optimal solution, the residual grows rapidly as the iterates move into areas of the parameter space away from HDM samples. However, near the optimal solution, the residual remains small as the optimization iterates remain in a small neighborhood of the most recent HDM sample.

Remark. *There are two mechanisms that prevent the reduced optimization problem from venturing into regions of the parameter space where it lacks accuracy: (1) the objective function and (2) the nonlinear trust region. In the present inverse design example, the objective function is mostly sufficient to keep the ROM in regions of accuracy, as can be seen from Figure 5.24 where the trust region bound is only reached once and the upper bound always increases. For other objective functions such as drag, the nonlinear trust region will be necessary as it is likely that an inaccurate ROM can predict a lower value in such objective functions than is actually present. In practice, inaccurate ROMs have been observed to predict the nonphysical situation of negative drag (i.e. thrust), which motivates the need for the residual-based trust region.*

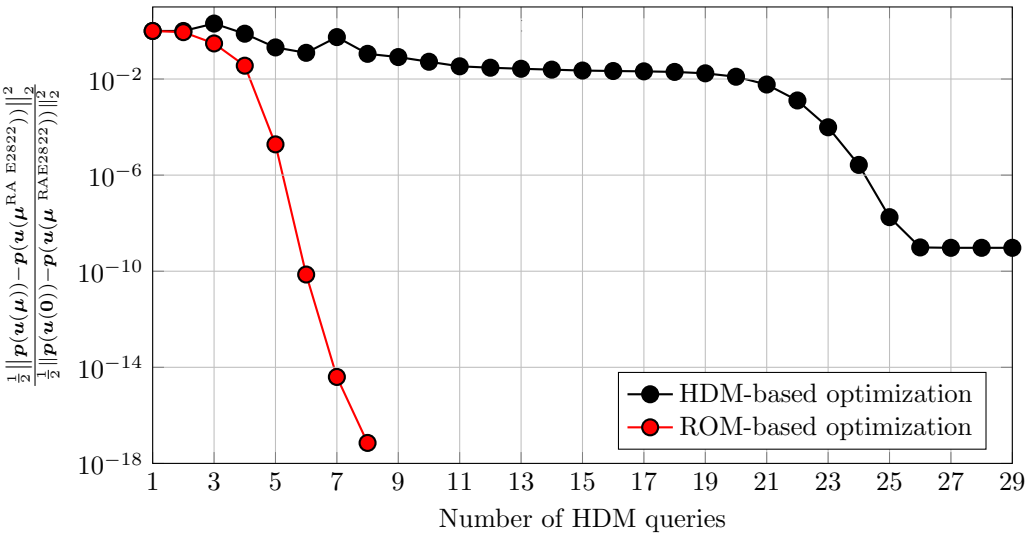


Figure 5.22: Objective function versus number of queries to the HDM: ROM-based optimization (red) and HDM-based optimization (black).

5.5.4 Shape Optimization of the Common Research Model in Viscous, Turbulent Flow

This section applies the proposed trust region method based on reduced-order approximation models to shape design of a full aircraft configuration—the Common Research Model (CRM)—in viscous, turbulent flow. The goal of the optimization problem is to maximize the lift-to-drag ratio of the aircraft while maintaining a constant lift. The flow is modeled using the Reynolds' Averaged Navier-Stokes (RANS) equations with a Spalart-Allmaras turbulence model. The freestream Mach number and angle of attack are taken as $M = 0.85$ and $\alpha = 2.32°$, which are standard operating conditions for a commercial aircraft of this size. The Reynolds' number is $Re = 5 \times 10^6$, which is based on wind tunnel model conditions and the reference chord length in the undeformed configuration. The chosen

[6]The last HDM sample in Figure 5.22 was not included in this count as the residual-based error indicator is small at this configuration (Figure 5.24). A similar argument could also be made for the 7th HDM sample.

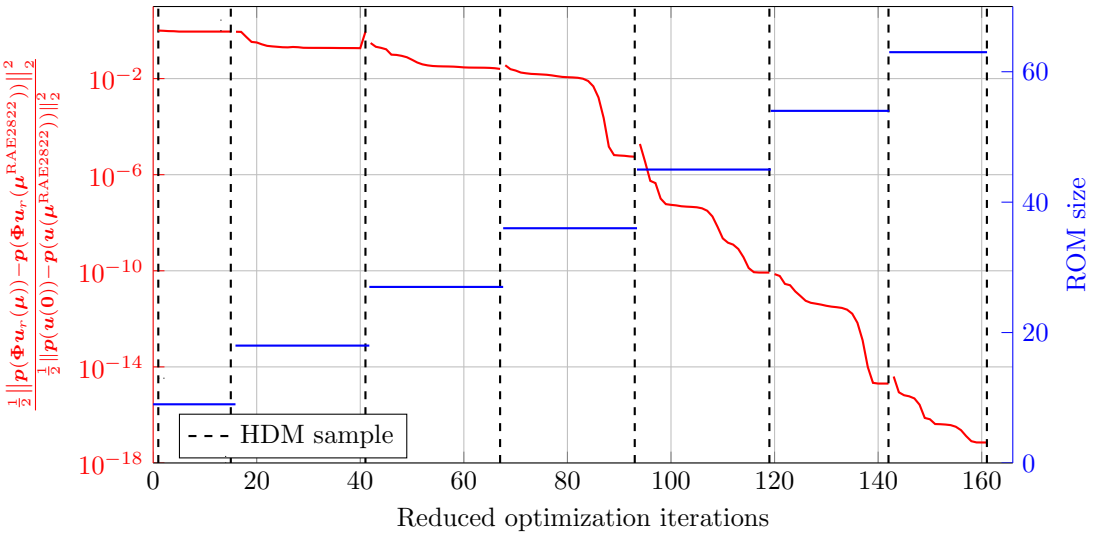


Figure 5.23: Progression of *reduced* objective function: dashed line indicates an HDM sample and a subsequent update of the ROB.

freestream Mach number place the flow in the transonic regime, which implies shocks will develop on the wing of the aircraft. The governing equations are discretized with a second-order, vertex-centered finite volume scheme using the AERO-F software [68] and the resulting system of nonlinear equations are solved using pseudo-transient continuation. The mesh employed was validated for these freestream conditions [198] and consists of 11 454 702 nodes for a total of 68 728 212 degrees of freedom.

The design problem (5.64) looks to maximize the lift-to-drag ratio at a constant lift subject to box constraints over a four-dimensional shape design space

$$\begin{aligned} \underset{\boldsymbol{\mu}\in\mathbb{R}^4}{\text{maximize}} \quad & L_z(\boldsymbol{\mu})/L_x(\boldsymbol{\mu}) \\ \text{subject to} \quad & L_z(\boldsymbol{\mu}) = L_z(\mathbf{0}) \\ & \boldsymbol{\mu}_{\boldsymbol{l}} \leq \boldsymbol{\mu} \leq \boldsymbol{\mu}_{\boldsymbol{u}}. \end{aligned} \tag{5.64}$$

The four shape parameters considered in this problem are: wingspan (μ_1), localized sweep (μ_2), twist (μ_3), and localized dihedral (μ_4); see Figure 5.25 for an illustration of each parameter. The lift constraint is included to ensure the optimized aircraft can carry the same payload as the original aircraft. The box constraints are included to ensure the shape changes are reasonable and the computational mesh does not tangle. The optimization problem in (5.64) is initialized from a perturbed CRM configuration that shortens the wing and adds negative twist. The optimized configuration achieves a drag count reduction of 2.2 by lengthening the wing and adding positive sweep, dihedral, and twist. This solution was obtained using by embedding a L-BFGS-B [215] bound-constrained optimization solver in an augmented Lagrangian framework to handle the nonlinear equality constraint and solve (5.64) directly. This method, which solely relies on HDM solves for objective and gradient

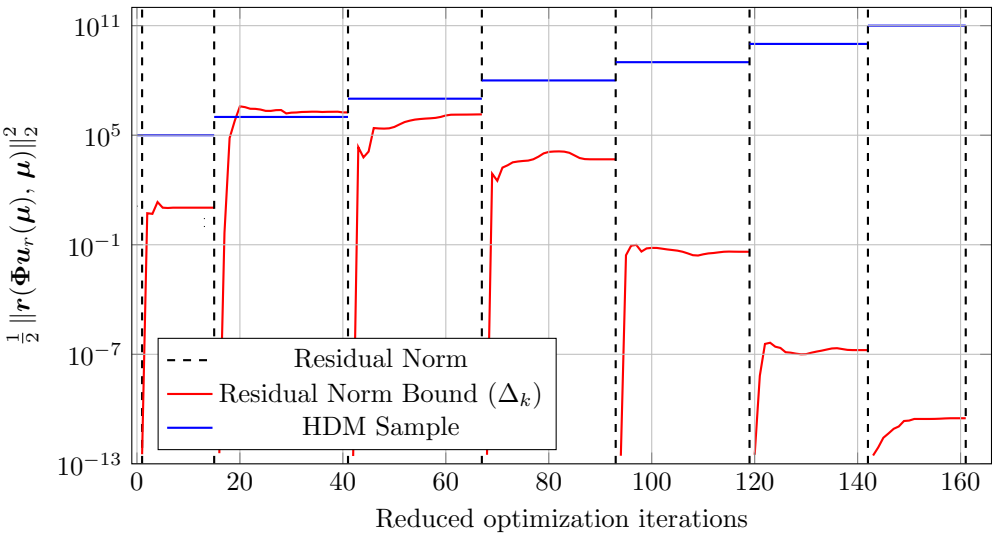


Figure 5.24: Progression of HDM residual: dashed line indicates an HDM sample and a subsequent update of the ROB.

queries, will serve as a baseline for comparison with the proposed hyperreduced trust region method in the remainder. Figure 5.26 provides two different views of the initial and optimized shapes to illustrate the changes that occur during the maximization process. Figure 5.27 shows the initial and optimized shapes colored by the pressure coefficient distribution on the surface. The optimized shape weakens the shock near the wing tip, which explains the 2.2 drag count reduction.

The proposed trust region method based on masked minimum-residual hyperreduced approximation models (with an underlying LSPG projection) is applied to solve the optimization problem in (5.64). Gradients are computed according to the masked minimum-residual sensitivity method and primal/sensitivity snapshots are used in the heterogeneous, span-preserving variant of POD *without truncation*. Therefore, the size of the reduced-order model increases by 5 at each iteration since a single primal snapshot and four sensitivity snapshots are added to the reduced-order basis. Due to the presence of the nonlinear equality constraint, the unconstrained trust region method is wrapped in the augmented Lagrangian framework described in Section 3.2.1. Figure 5.28 shows the convergence history of the drag count reduction as a function of the number of HDM queries for the baseline method and the hyperreduced trust region method. The hyperreduced trust region method requires half as many queries to the HDM to converge to a prescribed tolerance. However, this does not account for all sources of cost in the hyperreduced trust region method since there is cost associated with solving the trust region subproblem (hyperreduced model queries) and construction of the hyperreduced model at each iteration. Figure 5.29 includes these additional sources of cost by showing the convergence of the drag count reduction as a function of *wall time* normalized by the wall time of a single HDM solve. When properly accounting for all sources of cost, the speedup of the hyperreduced trust region method decreases marginally from 2× to 1.6× (80% efficiency). Finally, Figure 5.30 shows the *sample mesh* used at intermediate of the hyperreduced trust region

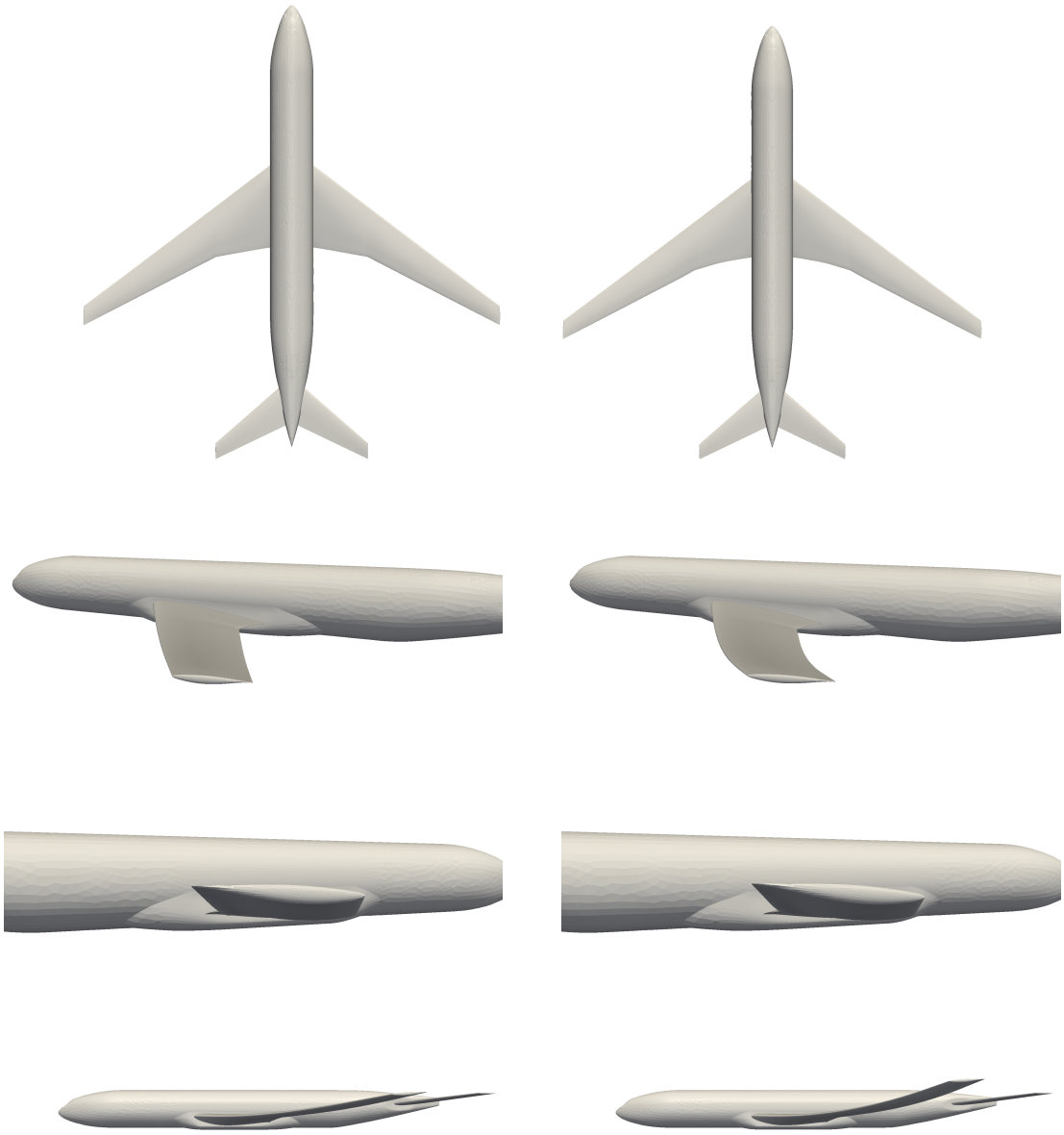


Figure 5.25: Parametrization of CRM. *Left*: Undeformed CRM configuration. *Right*: Deformed CRM configuration with positive perturbation to the wingspan μ_1 (top row), localized sweep μ_2 (second row), twist μ_3 (third row), and localized dihedral μ_4 (bottom row).

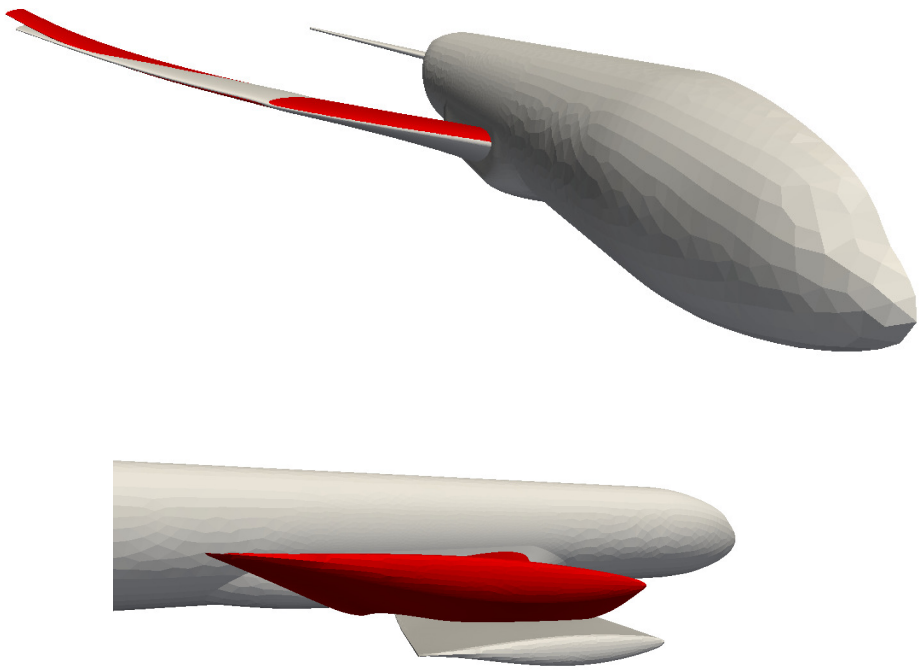


Figure 5.26: Two different views of the initial guess (gray) and solution (red) of the optimization problem in (5.64). The displacement from the undeformed configuration to the optimal solution (red) is magnified by 2×. There is a 2.2 drag count reduction from the initial to optimized shape.

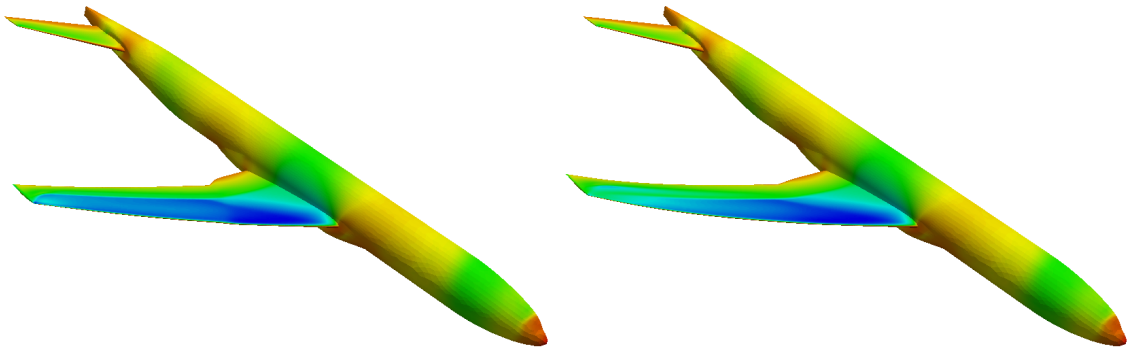


Figure 5.27: *Left*: Initial guess for optimization problem in (5.64). *Right*: Solution of optimization problem in (5.64). Both plots are colored by the coefficient of pressure C_p. There is a 2.2 drag count reduction from the initial to optimized shape.

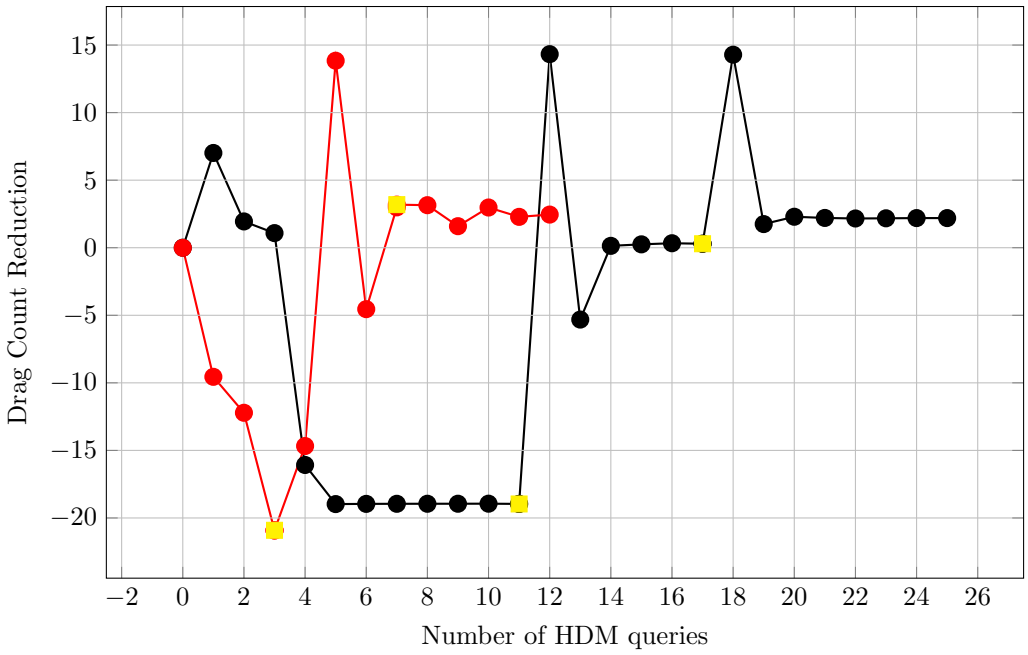


Figure 5.28: Convergence history of the baseline PDE-constrained optimization solver without model reduction (—●—) and proposed trust region method based on hyperreduced approximation models (—●—). A yellow square (■) indicates an augmented Lagrangian update. The reduction in drag count is taken as the performance metric and the number of primal HDM queries is the cost model. With respect to this cost metric, the ROM-based optimization solver converges 2× faster than the HDM-based solver.

method, which contains only 72 110 nodes—0.6% of the original mesh.

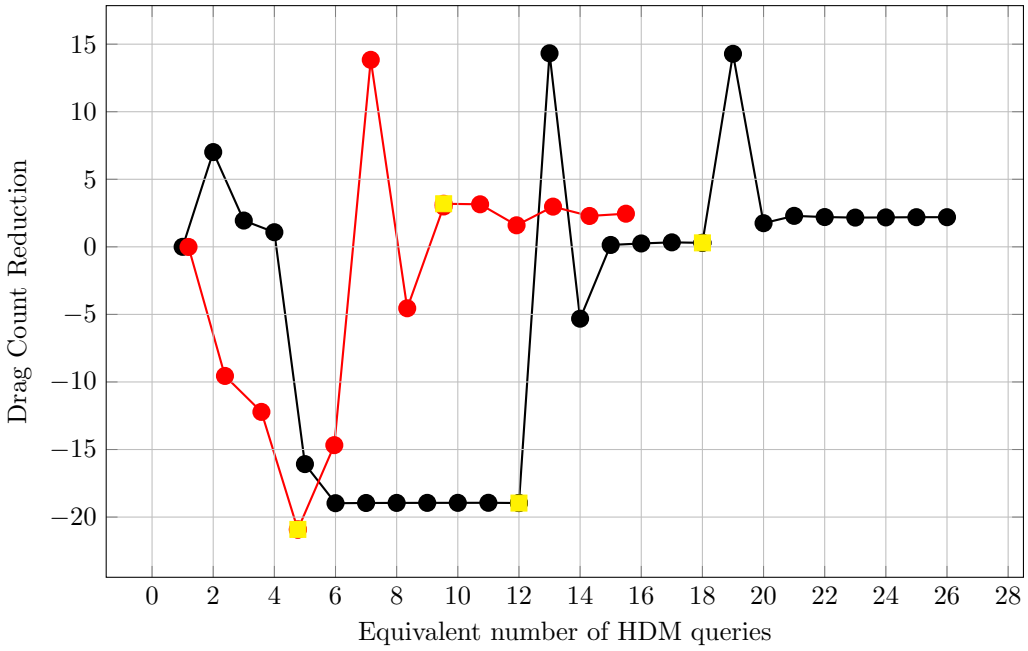


Figure 5.29: Convergence history of the baseline PDE-constrained optimization solver without model reduction (—●—) and proposed trust region method based on hyperreduced approximation models (—●—). A yellow square (■) indicates an augmented Lagrangian update. The reduction in drag count is taken as the performance metric and the total wall time of the optimization procedure (normalized by the wall time of a single primal HDM solve) is the cost model. With respect to this cost metric, the ROM-based optimization solver converges 1.6× faster than the HDM-based solver.

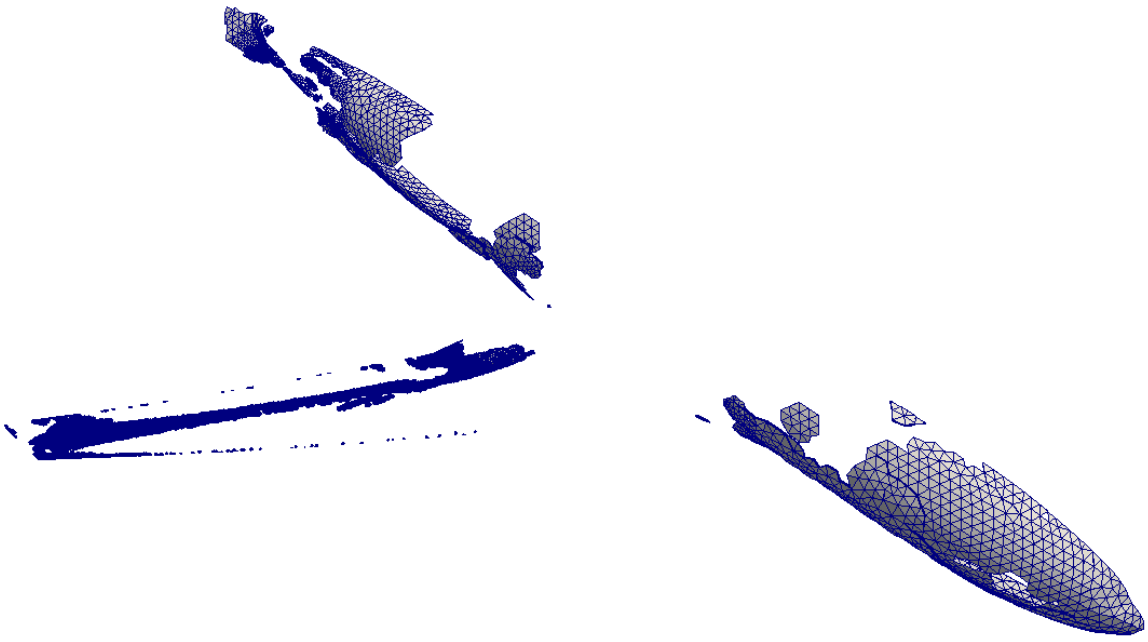


Figure 5.30: The sample mesh (72×10^3 nodes) used at an intermediate iteration of the trust region method based on hyperreduced (collocation) approximation models.

Table 5.2: Convergence history of Algorithm 11 applied to optimal control of the inviscid Burgers' equation using method 'sens-etr-intpt' using reduced-order models based on a Galerkin projection.

$F(\boldsymbol{\mu}_k)$	$m_k(\boldsymbol{\mu}_k)$	$F(\hat{\boldsymbol{\mu}}_k)$	$m_k(\hat{\boldsymbol{\mu}}_k)$	$\|\nabla F(\boldsymbol{\mu}_k)\|$	ρ_k	Δ_k	Success?
1.5922e+04	1.5922e+04	2.4783e+02	2.4528e+02	5.3000e+04	9.9984e-01	1.0000e-01	1.0000e+00
2.4783e+02	2.4783e+02	1.4391e+02	1.4451e+02	5.2213e+02	1.0057e+00	2.0000e-01	1.0000e+00
1.4391e+02	1.4391e+02	1.4317e-05	2.2758e-03	6.1988e+02	1.0000e+00	4.0000e-01	1.0000e+00
1.4317e-05	1.4317e-05	6.3338e-20	8.7340e-16	1.4083e+00	1.0000e+00	8.0000e-01	1.0000e+00
6.3338e-20	8.7340e-16	-	-	3.6015e-07	-	-	-

Table 5.3: Convergence history of Algorithm 12 applied to optimal control of the inviscid Burgers' equation using method 'sens-etr-intpt' using reduced-order models based on a Galerkin projection.

$F(\boldsymbol{\mu}_k)$	$m_k(\boldsymbol{\mu}_k)$	$F(\hat{\boldsymbol{\mu}}_k)$	$m_k(\hat{\boldsymbol{\mu}}_k)$	$\|\nabla F(\boldsymbol{\mu}_k)\|$	ρ_k	Δ_k	Success?
1.6431e+04	1.6033e+04	1.8873e+02	1.9717e+02	5.3540e+04	1.0257e+00	1.0000e-01	1.0000e+00
1.8873e+02	1.9784e+02	1.0874e+02	1.0973e+02	7.9026e+02	9.0777e-01	2.0000e-01	1.0000e+00
1.0874e+02	1.0884e+02	1.2551e-03	5.9436e-03	6.7856e+02	9.9911e-01	4.0000e-01	1.0000e+00
1.2551e-03	1.3252e-04	5.3325e-08	4.4413e-05	7.8316e+01	1.4245e+01	8.0000e-01	1.0000e+00
5.3325e-08	5.3325e-08	4.4166e-23	3.1841e-20	1.2847e-01	1.0000e+00	1.6000e+00	1.0000e+00
4.4166e-23	3.1841e-20	-	-	3.4743e-09	-	-	-

Table 5.4: Convergence history of Algorithm 11 applied to optimal control of the inviscid Burgers' equation using method 'sens-ctr-stcg' using reduced-order models based on a Galerkin projection.

$F(\boldsymbol{\mu}_k)$	$m_k(\boldsymbol{\mu}_k)$	$F(\hat{\boldsymbol{\mu}}_k)$	$m_k(\hat{\boldsymbol{\mu}}_k)$	$\|\nabla F(\boldsymbol{\mu}_k)\|$	ρ_k	Δ_k	Success?
1.5922e+04	1.5922e+04	1.1372e+04	1.1346e+04	5.3000e+04	9.9439e-01	1.0000e-01	1.0000e+00
1.1372e+04	1.1372e+04	5.5895e+03	5.5895e+03	3.8805e+04	1.0000e+00	2.0000e-01	1.0000e+00
5.5895e+03	5.5895e+03	8.9676e+02	8.9676e+02	2.0968e+04	1.0000e+00	4.0000e-01	1.0000e+00
8.9676e+02	8.9676e+02	8.0315e+05	3.4795e+04	4.4001e+03	2.3666e+01	8.0000e-01	0.0000e+00
8.9676e+02	8.9676e+02	6.0483e+02	6.0483e+02	4.4001e+03	1.0000e+00	1.6000e-01	1.0000e+00
6.0483e+02	6.0483e+02	4.0276e+02	4.0276e+02	1.3683e+03	1.0000e+00	3.2000e-01	1.0000e+00
4.0276e+02	4.0276e+02	5.1981e+05	6.9325e+03	1.7695e+03	7.9545e+01	6.4000e-01	0.0000e+00
4.0276e+02	4.0276e+02	3.3109e+02	3.3109e+02	1.7695e+03	1.0000e+00	1.0240e-01	1.0000e+00
3.3109e+02	3.3109e+02	2.1174e+02	2.1174e+02	1.2858e+03	1.0000e+00	2.0480e-01	1.0000e+00
2.1174e+02	2.1174e+02	4.9377e+02	4.9378e+02	6.7866e+02	9.9998e-01	4.0960e-01	0.0000e+00
2.1174e+02	2.1174e+02	1.9116e+02	1.9116e+02	6.7866e+02	1.0000e+00	4.1943e-02	1.0000e+00
1.9116e+02	1.9116e+02	1.4938e+02	1.4938e+02	9.5647e+02	1.0000e+00	8.3886e-02	1.0000e+00
1.4938e+02	1.4938e+02	9.0046e+01	9.0046e+01	5.5266e+02	1.0000e+00	1.6777e-01	1.0000e+00
9.0046e+01	9.0046e+01	7.9533e+01	7.9533e+01	3.8955e+03	1.0000e+00	3.3554e-01	1.0000e+00
7.9533e+01	7.9533e+01	1.8002e+02	1.8002e+02	3.5646e+02	1.0000e+00	6.7109e-01	0.0000e+00
7.9533e+01	7.9533e+01	7.1672e+01	7.1672e+01	3.5646e+02	1.0000e+00	2.3488e-02	1.0000e+00
7.1672e+01	7.1672e+01	5.6588e+01	5.6588e+01	7.3693e+02	1.0000e+00	4.6976e-02	1.0000e+00
5.6588e+01	5.6588e+01	3.2573e+01	3.2573e+01	6.6885e+02	1.0000e+00	9.3951e-02	1.0000e+00
3.2573e+01	3.2573e+01	2.6527e+01	2.6527e+01	1.5016e+03	1.0000e+00	1.8790e-01	1.0000e+00
2.6527e+01	2.6527e+01	5.4958e+00	5.4958e+00	1.3171e+04	1.0000e+00	3.7581e-01	1.0000e+00
5.4958e+00	5.4958e+00	4.6810e+00	4.6810e+00	2.3990e+03	1.0000e+00	7.5161e-01	1.0000e+00
4.6810e+00	4.6810e+00	6.7425e+00	6.7425e+00	1.1796e+02	1.0000e+00	1.5032e+00	0.0000e+00
4.6810e+00	4.6810e+00	4.4251e+00	4.4251e+00	1.1796e+02	1.0000e+00	3.1837e-03	1.0000e+00
4.4251e+00	4.4251e+00	3.9356e+00	3.9356e+00	1.1226e+02	1.0000e+00	6.3673e-03	1.0000e+00
3.9356e+00	3.9356e+00	3.0492e+00	3.0492e+00	9.4010e+01	1.0000e+00	1.2735e-02	1.0000e+00
3.0492e+00	3.0492e+00	1.6303e+00	1.6303e+00	1.7259e+02	1.0000e+00	2.5469e-02	1.0000e+00
1.6303e+00	1.6303e+00	3.8792e-01	3.8792e-01	3.3882e+02	1.0000e+00	5.0939e-02	1.0000e+00
3.8792e-01	3.8792e-01	1.1734e-01	1.1734e-01	1.8628e+03	1.0000e+00	1.0188e-01	1.0000e+00
1.1734e-01	1.1734e-01	4.6898e-03	4.6898e-03	8.2406e+01	1.0000e+00	2.0375e-01	1.0000e+00
4.6898e-03	4.6898e-03	2.7171e-05	2.7171e-05	2.5736e+02	1.0000e+00	4.0751e-01	1.0000e+00
2.7171e-05	2.7171e-05	9.5916e-06	9.5916e-06	1.8011e+00	1.0000e+00	8.1502e-01	1.0000e+00
9.5916e-06	9.5916e-06	2.5087e-11	2.5087e-11	1.1313e-01	1.0000e+00	1.6300e+00	1.0000e+00
2.5087e-11	2.5087e-11	3.7967e-12	3.7967e-12	1.7368e-02	1.0000e+00	3.2601e+00	1.0000e+00
3.7967e-12	3.7967e-12	2.5080e-18	2.5047e-18	1.4714e-04	1.0000e+00	6.5202e+00	1.0000e+00
2.5080e-18	2.5080e-18	-	-	4.2318e-06	-	-	-

Table 5.5: Convergence history of Algorithm 12 applied to optimal control of the inviscid Burgers' equation using method 'sens-ctr-stcg' using reduced-order models based on a Galerkin projection.

$F(\boldsymbol{\mu}_k)$	$m_k(\boldsymbol{\mu}_k)$	$F(\hat{\boldsymbol{\mu}}_k)$	$m_k(\hat{\boldsymbol{\mu}}_k)$	$\|\nabla F(\boldsymbol{\mu}_k)\|$	ρ_k	Δ_k	Success?
1.6431e+04	1.6033e+04	1.1573e+04	1.1464e+04	5.3540e+04	1.0634e+00	1.0000e-01	1.0000e+00
1.1573e+04	1.1372e+04	5.6146e+03	5.5885e+03	3.9065e+04	1.0303e+00	2.0000e-01	1.0000e+00
5.6146e+03	5.5885e+03	9.0886e+02	8.9613e+02	2.1019e+04	1.0028e+00	4.0000e-01	1.0000e+00
9.0886e+02	8.9613e+02	8.8161e+07	7.2819e+04	4.6131e+03	1.2258e+03	8.0000e-01	0.0000e+00
9.0886e+02	8.9613e+02	5.9847e+02	6.0460e+02	4.6131e+03	1.0647e+00	1.6000e-01	1.0000e+00
5.9847e+02	6.0459e+02	3.9555e+02	4.0246e+02	1.2978e+03	1.0039e+00	3.2000e-01	1.0000e+00
3.9555e+02	4.0246e+02	2.4110e+05	2.6063e+04	2.0151e+03	9.3805e+00	6.4000e-01	0.0000e+00
3.9555e+02	4.0246e+02	3.3074e+02	3.3080e+02	2.0151e+03	9.0433e-01	1.0240e-01	1.0000e+00
3.3074e+02	3.3080e+02	2.1140e+02	2.1150e+02	1.2856e+03	1.0004e+00	2.0480e-01	1.0000e+00
2.1140e+02	2.1150e+02	4.9353e+02	4.9353e+02	6.7904e+02	1.0003e+00	4.0960e-01	0.0000e+00
2.1140e+02	2.1150e+02	1.9084e+02	1.9093e+02	6.7904e+02	1.0000e+00	4.1943e-02	1.0000e+00
1.9084e+02	1.9093e+02	1.4906e+02	1.4917e+02	9.5568e+02	1.0003e+00	8.3886e-02	1.0000e+00
1.4906e+02	1.4917e+02	8.9792e+01	8.9903e+01	5.5440e+02	1.0001e+00	1.6777e-01	1.0000e+00
8.9792e+01	8.9903e+01	7.9260e+01	7.9375e+01	3.8951e+03	1.0003e+00	3.3554e-01	1.0000e+00
7.9260e+01	7.9375e+01	1.7709e+02	1.7709e+02	3.5553e+02	1.0012e+00	6.7109e-01	0.0000e+00
7.9260e+01	7.9375e+01	7.1456e+01	7.1569e+01	3.5553e+02	9.9979e-01	2.3349e-02	1.0000e+00
7.1456e+01	7.1569e+01	5.6462e+01	5.6577e+01	7.4645e+02	1.0002e+00	4.6697e-02	1.0000e+00
5.6462e+01	5.6577e+01	3.2554e+01	3.2667e+01	6.5370e+02	9.9991e-01	9.3395e-02	1.0000e+00
3.2554e+01	3.2667e+01	2.7533e+01	2.7533e+01	1.4597e+03	9.7789e-01	1.8679e-01	1.0000e+00
2.7533e+01	2.7533e+01	5.5226e+00	5.5226e+00	1.3537e+04	1.0000e+00	3.7358e-01	1.0000e+00
5.5226e+00	5.5226e+00	4.6138e+00	4.6138e+00	2.5382e+03	1.0000e+00	7.4716e-01	1.0000e+00
4.6138e+00	4.6138e+00	6.6933e+00	6.6933e+00	1.2584e+02	1.0000e+00	1.4943e+00	0.0000e+00
4.6138e+00	4.6138e+00	4.3612e+00	4.3612e+00	1.2584e+02	1.0000e+00	3.1598e-03	1.0000e+00
4.3612e+00	4.3612e+00	3.8793e+00	3.8793e+00	1.0487e+02	1.0000e+00	6.3196e-03	1.0000e+00
3.8793e+00	3.8793e+00	3.0053e+00	3.0053e+00	1.0029e+02	9.9998e-01	1.2639e-02	1.0000e+00
3.0053e+00	3.0053e+00	1.6074e+00	1.6075e+00	1.5696e+02	1.0000e+00	2.5279e-02	1.0000e+00
1.6074e+00	1.6075e+00	3.7861e-01	3.7862e-01	3.2953e+02	1.0000e+00	5.0557e-02	1.0000e+00
3.7861e-01	3.7862e-01	1.1784e-01	1.1784e-01	1.8295e+03	9.9998e-01	1.0111e-01	1.0000e+00
1.1784e-01	1.1784e-01	4.5753e-03	4.5761e-03	7.9865e+01	9.9998e-01	2.0223e-01	1.0000e+00
4.5753e-03	4.5761e-03	3.4610e-05	3.4584e-05	2.5399e+02	9.9982e-01	4.0446e-01	1.0000e+00
3.4610e-05	3.4584e-05	1.3090e-05	1.3090e-05	1.8009e+00	1.0012e+00	8.0891e-01	1.0000e+00
1.3090e-05	1.3090e-05	4.7238e-11	4.7238e-11	1.3249e-01	1.0000e+00	1.6178e+00	1.0000e+00
4.7238e-11	4.7238e-11	6.3287e-12	6.3287e-12	2.4075e-02	1.0000e+00	3.2357e+00	1.0000e+00
6.3287e-12	6.3287e-12	5.0572e-21	5.0530e-21	1.9695e-04	1.0000e+00	6.4713e+00	1.0000e+00
5.0572e-21	5.0572e-21	4.4579e-24	5.0572e-21	2.6757e-07	inf	1.2943e+01	1.0000e+00
4.4579e-24	5.0572e-21	-	-	8.1378e-11	-	-	-

Table 5.6: Convergence history of Algorithm 11 applied to optimal control of the viscous Burgers' equation using method 'dual-etr-intpt' using reduced-order models based on a Galerkin projection.

$F(\boldsymbol{\mu}_k)$	$m_k(\boldsymbol{\mu}_k)$	$F(\hat{\boldsymbol{\mu}}_k)$	$m_k(\hat{\boldsymbol{\mu}}_k)$	$\|\nabla F(\boldsymbol{\mu}_k)\|$	ρ_k	Δ_k	Success?
3.8620e-02	3.8620e-02	3.2002e-02	3.4377e-02	1.2191e-02	1.5596e+00	1.0000e-01	1.0000e+00
3.2002e-02	3.2002e-02	2.2060e-02	1.6674e-02	7.0854e-03	6.4861e-01	2.0000e-01	1.0000e+00
2.2060e-02	2.2060e-02	1.5559e-02	1.5606e-02	2.0611e-03	1.0072e+00	1.8000e-01	1.0000e+00
1.5559e-02	1.5559e-02	1.5527e-02	1.5521e-02	6.9906e-05	8.4404e-01	3.6000e-01	1.0000e+00
1.5527e-02	1.5527e-02	1.5524e-02	1.5524e-02	1.6373e-05	9.3144e-01	7.2000e-01	1.0000e+00
1.5524e-02	1.5524e-02	1.5524e-02	1.5524e-02	1.5633e-06	9.7164e-01	1.4400e+00	1.0000e+00
1.5524e-02	1.5524e-02	1.5524e-02	1.5524e-02	9.1808e-08	9.9643e-01	2.8800e+00	1.0000e+00
1.5524e-02	1.5524e-02	1.5524e-02	1.5524e-02	5.5276e-07	9.9994e-01	5.7600e+00	1.0000e+00
1.5524e-02	1.5524e-02	1.5524e-02	1.5524e-02	3.2305e-08	1.0020e+00	1.1520e+01	1.0000e+00
1.5524e-02	1.5524e-02	1.5524e-02	1.5524e-02	6.9441e-08	1.0042e+00	2.3040e+01	1.0000e+00
1.5524e-02	1.5524e-02	-	-	6.1859e-08	-	-	-

Table 5.7: Convergence history of Algorithm 11 applied to optimal control of the viscous Burgers' equation using method 'dual-ctr-intpt' using reduced-order models based on a Galerkin projection.

$F(\boldsymbol{\mu}_k)$	$m_k(\boldsymbol{\mu}_k)$	$F(\hat{\boldsymbol{\mu}}_k)$	$m_k(\hat{\boldsymbol{\mu}}_k)$	$\|\nabla F(\boldsymbol{\mu}_k)\|$	ρ_k	Δ_k	Success?
3.8620e-02	3.8620e-02	3.3959e-02	3.4919e-02	1.2191e-02	1.2596e+00	1.0000e-01	1.0000e+00
3.3959e-02	3.3959e-02	2.9644e-02	2.9645e-02	8.7651e-03	1.0003e+00	2.0000e-01	1.0000e+00
2.9644e-02	2.9644e-02	2.6406e-02	2.6406e-02	5.1460e-03	1.0000e+00	4.0000e-01	1.0000e+00
2.6406e-02	2.6406e-02	2.3727e-02	2.3727e-02	2.6859e-03	1.0000e+00	8.0000e-01	1.0000e+00
2.3727e-02	2.3727e-02	2.1113e-02	2.1113e-02	1.7850e-03	1.0000e+00	1.6000e+00	1.0000e+00
2.1113e-02	2.1113e-02	1.8771e-02	1.8771e-02	1.1968e-03	1.0000e+00	3.2000e+00	1.0000e+00
1.8771e-02	1.8771e-02	1.6877e-02	1.6877e-02	7.2259e-04	1.0000e+00	6.4000e+00	1.0000e+00
1.6877e-02	1.6877e-02	1.5700e-02	1.5700e-02	3.8119e-04	1.0000e+00	1.2800e+01	1.0000e+00
1.5700e-02	1.5700e-02	1.5524e-02	1.5524e-02	1.1431e-04	1.0000e+00	2.5600e+01	1.0000e+00
1.5524e-02	1.5524e-02	1.5524e-02	1.5524e-02	3.7482e-08	1.0042e+00	5.1200e+01	1.0000e+00
1.5524e-02	1.5524e-02	-	-	6.3774e-08	-	-	-

Table 5.8: Performance of the HDM- and ROM-based optimization methods.

	HDM-based optimization	ROM-based optimization
# of HDM Evaluations	29	7[6]
# of ROM Evaluations	-	346
$\frac{\|\boldsymbol{\mu}^* - \boldsymbol{\mu}^{RAE2822}\|}{\|\boldsymbol{\mu}^{RAE2822}\|}$	$2.28 \times 10^{-3}\%$	$4.17 \times 10^{-6}\%$

Chapter 6

Model Reduction and Sparse Grids for Efficient Stochastic Optimization

To this point, all partial differential equations, and the corresponding optimization problems, have been posed in a deterministic setting, that is, the PDE itself and all its data are assumed *known*. This is not a realistic assumption since all PDEs are merely mathematical models of physical phenomena and even if the PDE is an accurate approximation of reality, its data—coefficients, boundary conditions, source terms, etc—will rarely be known with certainty. This is particularly true in physical systems characterized by a high degree of volatility or those where physical measurements are difficult to take. In such settings, the uncertainty must be incorporated into the optimization problem if a robust, risk-averse design or control is to be attained. In this work, *parametrized* uncertainties are considered and risk-averse measures (Section 2.2.1) of quantities of interest will be used as the objective and constraint functions for the stochastic PDE-constrained optimization problem. The mathematical construction and discretized of parametrized stochastic partial differential equations is provided in Section 2.2, including the introduction of a complete probability space, the finite noise assumption, spatio-temporal discretization of a realization of the stochastic partial differential equation, and collocation-based discretization of the stochastic space. Since risk-averse measures usually require the computation of an integral over the stochastic space, a single query to an optimization function requires the evaluation of an integral whose integrand depends on the solution of a realization of the stochastic partial differential equation. In general, this requires a (possibly large) ensemble of deterministic PDE solves and makes stochastic PDE-constrained optimization problems potentially many orders of magnitude more expensive than the deterministic counterparts. In fact, if there are large number of stochastic parameters, it is difficult to evaluate an integral over a high-dimensional space *even if the integrand is inexpensive to evaluate* due to the curse of dimensionality. A straightforward or brute force approach to solve such optimization problems is

infeasible for all but the simplest problems.

To address the large cost of PDE-constrained optimization under uncertainty, a multifidelity trust region method based on the theory introduced in Chapter 3 is developed. The approximation model incorporates *two levels* of inexactness: dimension-adaptive sparse grids for efficient collocation-based integration in moderate-to-high dimensional spaces and reduced-order models to reduce the cost of queries to a realization of the stochastic partial differential equation. Both levels of the approximation will be incorporated into the required error indicators. A two-level greedy method is proposed to construct the sparse grid and reduced-order basis such that, at a given iteration of the trust region method, the required error conditions are satisfied, thus ensuring global convergence. The proposed method is demonstrated on a one-dimensional optimal flow control problem. For simplicity, the remainder of this document will consider only the risk-neutral measure, or expectation, of a quantity of interest. Extension to other risk-averse measures will be deferred to later work.

6.1 Background

This chapter begins with an overview of ingredients that will be necessary to develop the proposed trust region method based on the two-level approximation of risk measures of quantities of interest of stochastic PDEs: stochastic reduced-order models and anisotropic sparse grids.

6.1.1 Stochastic High-Dimensional Model

Consider the discrete collocation-based stochastic PDE introduced in Section 2.2

$$\boldsymbol{r}(\boldsymbol{u},\, \boldsymbol{\mu},\, \boldsymbol{y}) = 0 \qquad \forall \boldsymbol{y} \in \boldsymbol{\Xi} \tag{6.1}$$

where $\boldsymbol{u} \in \mathbb{R}^{N_{\boldsymbol{u}}}$ is the state vector, $\boldsymbol{\mu} \in \mathbb{R}^{N_{\boldsymbol{\mu}}}$ is the parameter vector, $\boldsymbol{y} \in \boldsymbol{\Xi}$ are the stochastic variables, and $\boldsymbol{\Xi} \subset \mathbb{R}^{N_{\boldsymbol{y}}}$ is the stochastic space. The existence of a continuously differentiable function $\boldsymbol{u}(\boldsymbol{\mu};\, \boldsymbol{y})$, defined as the solution of $\boldsymbol{r}(\,\cdot\,,\, \boldsymbol{\mu},\, \boldsymbol{y}) = 0$, is guaranteed by Theorem 2.1, under suitable assumptions. Depending on the nature of the stochastic variables, the quantity of interest may be stochastic as well and a realization will take the form

$$f(\boldsymbol{u},\, \boldsymbol{\mu},\, \boldsymbol{y}) \tag{6.2}$$

for $\boldsymbol{y} \in \boldsymbol{\Xi}$, which can be considered only a function of $\boldsymbol{\mu}$ and $\boldsymbol{y}$ using the implicit definition $\boldsymbol{u}(\boldsymbol{\mu};\, \boldsymbol{y})$

$$F(\boldsymbol{\mu};\, \boldsymbol{y}) = f(\boldsymbol{u}(\boldsymbol{\mu};\, \boldsymbol{y}),\, \boldsymbol{\mu},\, \boldsymbol{y}). \tag{6.3}$$

The risk-neutral measure of the QoI, which will be used as the objective for the stochastic optimization problem in this work, is

$$F(\boldsymbol{\mu}) = \mathbb{E}[f(\boldsymbol{u}(\boldsymbol{\mu};\, \cdot\,),\, \boldsymbol{\mu},\, \cdot\,)] = \mathbb{E}[F(\boldsymbol{\mu},\, \cdot\,)]. \tag{6.4}$$

Generalization to other risk-averse measures proceeds by replacing $\mathbb{E}[\cdot]$ with $\mathcal{R}[\cdot]$ defined in (2.51)-(2.55); however, special care will be required in the construction of quadrature rules for non-smooth risk-averse measures.

The gradient of the risk-neutral measure of the QoI, $\nabla F(\boldsymbol{\mu})$, is computed via the sensitivity or adjoint method, depending on the number of QoIs versus the number of parameters ($N_{\boldsymbol{\mu}}$). Since the differentiation operation (with respect to $\boldsymbol{\mu}$) can be pulled inside the expectation operation (integration with respect to $\boldsymbol{y}$), the gradient takes the form

$$\nabla F(\boldsymbol{\mu}) = \mathbb{E}\left[\nabla f(\boldsymbol{u}(\boldsymbol{\mu};\,\cdot\,),\,\boldsymbol{\mu},\,\cdot\,)\right] = \mathbb{E}\left[\nabla F(\boldsymbol{\mu},\,\cdot\,)\right]. \tag{6.5}$$

This illustrates that the computation of the gradient of the risk-neutral measure of the QoI reduces to an integral over realizations of the QoI gradients, i.e., for a fixed $\boldsymbol{y} \in \boldsymbol{\Xi}$ and $\boldsymbol{\mu} \in \mathbb{R}^{N_{\boldsymbol{\mu}}}$. The gradient of a particular realization proceeds exactly according to the adjoint or sensitivity method outlined in Sections 2.3.3 and 2.3.4. Following the procedures outlined in that section, the stochastic variant of the sensitivity and adjoint residuals are

$$\begin{aligned} \boldsymbol{r}^{\partial}(\boldsymbol{u},\,\boldsymbol{w},\,\boldsymbol{\mu},\,\boldsymbol{y}) &\coloneqq \frac{\partial f}{\partial \boldsymbol{\mu}}(\boldsymbol{u},\,\boldsymbol{\mu},\,\boldsymbol{y}) + \frac{\partial f}{\partial \boldsymbol{u}}(\boldsymbol{u},\,\boldsymbol{\mu},\,\boldsymbol{y})\boldsymbol{w} \\ \boldsymbol{r}^{\boldsymbol{\lambda}}(\boldsymbol{u},\,\boldsymbol{z},\,\boldsymbol{\mu},\,\boldsymbol{y}) &\coloneqq \frac{\partial f}{\partial \boldsymbol{\mu}}(\boldsymbol{u},\,\boldsymbol{\mu},\,\boldsymbol{y}) - \boldsymbol{z}^T \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{\mu}}(\boldsymbol{u},\,\boldsymbol{\mu},\,\boldsymbol{y}). \end{aligned} \tag{6.6}$$

Then, a realization of the sensitivity problem for a fixed $\boldsymbol{y} \in \boldsymbol{\Xi}$ and $\boldsymbol{\mu} \in \mathbb{R}^{N_{\boldsymbol{\mu}}}$ is: given the primal solution $\boldsymbol{u}(\boldsymbol{\mu};\,\boldsymbol{y})$ that satisfies $\boldsymbol{r}(\,\cdot\,,\,\boldsymbol{\mu},\,\boldsymbol{y}) = 0$, find $\dfrac{\partial \boldsymbol{u}}{\partial \boldsymbol{\mu}}$ such that $\boldsymbol{r}^{\partial}\left(\boldsymbol{u}(\boldsymbol{\mu};\,\boldsymbol{y}),\,\dfrac{\partial \boldsymbol{u}}{\partial \boldsymbol{\mu}},\,\boldsymbol{\mu},\,\boldsymbol{y}\right) = 0$. Similarly, a realization of the adjoint problem for a fixed $\boldsymbol{y} \in \boldsymbol{\Xi}$ and $\boldsymbol{\mu} \in \mathbb{R}^{N_{\boldsymbol{\mu}}}$ is: given the primal solution $\boldsymbol{u}(\boldsymbol{\mu};\,\boldsymbol{y})$, find $\boldsymbol{\lambda}$ such that $\boldsymbol{r}^{\boldsymbol{\lambda}}(\boldsymbol{u}(\boldsymbol{\mu};\,\boldsymbol{y}),\,\boldsymbol{\lambda},\,\boldsymbol{\mu},\,\boldsymbol{y}) = 0$. The sensitivity and adjoint solution, for a particular $\boldsymbol{y} \in \boldsymbol{\Xi}$ and $\boldsymbol{\mu} \in \mathbb{R}^{N_{\boldsymbol{\mu}}}$, will be denoted $\dfrac{\partial \boldsymbol{u}}{\partial \boldsymbol{\mu}}(\boldsymbol{\mu};\,\boldsymbol{y})$ and $\boldsymbol{\lambda}(\boldsymbol{\mu};\,\boldsymbol{y})$, respectively. The reconstruction of the gradient of a QoI from a sensitivity or adjoint solution are generalized from the deterministic case in (2.90), (2.102) to the stochastic case as

$$\begin{aligned} \boldsymbol{g}^{\partial}(\boldsymbol{u},\,\boldsymbol{w},\,\boldsymbol{\mu},\,\boldsymbol{y}) &\coloneqq \frac{\partial f}{\partial \boldsymbol{\mu}}(\boldsymbol{u},\,\boldsymbol{\mu},\,\boldsymbol{y}) + \frac{\partial f}{\partial \boldsymbol{u}}(\boldsymbol{u},\,\boldsymbol{\mu},\,\boldsymbol{y})\boldsymbol{w} \\ \boldsymbol{g}^{\boldsymbol{\lambda}}(\boldsymbol{u},\,\boldsymbol{z},\,\boldsymbol{\mu},\,\boldsymbol{y}) &\coloneqq \frac{\partial f}{\partial \boldsymbol{\mu}}(\boldsymbol{u},\,\boldsymbol{\mu},\,\boldsymbol{y}) + \boldsymbol{z}^T \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{\mu}}(\boldsymbol{u},\,\boldsymbol{\mu},\,\boldsymbol{y}). \end{aligned} \tag{6.7}$$

With these definitions, a realization of the gradient of a QoI corresponding to $\boldsymbol{y} \in \boldsymbol{\Xi}$ and $\boldsymbol{\mu} \in \mathbb{R}^{N_{\boldsymbol{\mu}}}$ takes the form

$$\nabla F(\boldsymbol{\mu},\,\boldsymbol{y}) = \boldsymbol{g}^{\partial}\left(\boldsymbol{u}(\boldsymbol{\mu};\,\boldsymbol{y}),\,\frac{\partial \boldsymbol{u}}{\partial \boldsymbol{\mu}}(\boldsymbol{\mu};\,\boldsymbol{y}),\,\boldsymbol{\mu},\,\boldsymbol{y}\right) = \boldsymbol{g}^{\boldsymbol{\lambda}}\left(\boldsymbol{u}(\boldsymbol{\mu};\,\boldsymbol{y}),\,\boldsymbol{\lambda}(\boldsymbol{\mu};\,\boldsymbol{y}),\,\boldsymbol{\mu},\,\boldsymbol{y}\right) \tag{6.8}$$

and the gradient of the risk-neutral measure in (6.4) is

$$\nabla F(\boldsymbol{\mu}) = \mathbb{E}\left[\boldsymbol{g}^{\partial}\left(\boldsymbol{u}(\boldsymbol{\mu};\,\cdot\,),\,\frac{\partial \boldsymbol{u}}{\partial \boldsymbol{\mu}}(\boldsymbol{\mu};\,\cdot\,),\,\boldsymbol{\mu},\,\cdot\,\right)\right] = \mathbb{E}\left[\boldsymbol{g}^{\boldsymbol{\lambda}}\left(\boldsymbol{u}(\boldsymbol{\mu};\,\cdot\,),\,\boldsymbol{\lambda}(\boldsymbol{\mu};\,\cdot\,),\,\boldsymbol{\mu},\,\cdot\,\right)\right]. \tag{6.9}$$

6.1.2 Stochastic Reduced-Order Model

The dimension of the discretized stochastic PDE in (6.1) is reduced through the introduction of the model reduction ansatz $\boldsymbol{u} = \boldsymbol{\Phi}\boldsymbol{u}_r$ from (4.2) into (6.1), where $\boldsymbol{\Phi} \in \mathbb{R}^{N_{\boldsymbol{u}} \times k_{\boldsymbol{u}}}$ is the trial basis that defines a subspace that (approximately) contains the solution of any realization of the stochastic PDE, i.e., $\boldsymbol{u}(\boldsymbol{\mu}, \boldsymbol{y})$ for $\boldsymbol{\mu} \in \mathbb{R}^{N_{\boldsymbol{\mu}}}$ and $\boldsymbol{y} \in \boldsymbol{\Xi}$. The result is an overdetermined nonlinear system of equations $\boldsymbol{r}(\boldsymbol{\Phi}\boldsymbol{u}_r, \boldsymbol{\mu}, \boldsymbol{y}) = 0$ for any realization $\boldsymbol{y} \in \boldsymbol{\Xi}$. Projection of these equations onto the columnspace of the test basis $\boldsymbol{\Psi} \in \mathbb{R}^{N_{\boldsymbol{u}} \times k_{\boldsymbol{u}}}$ leads to the projection-based reduced-order model with $k_{\boldsymbol{u}}$ equations and unknowns

$$\boldsymbol{r}_r(\boldsymbol{u}_r, \boldsymbol{\mu}, \boldsymbol{y}) \coloneqq \boldsymbol{\Psi}^T \boldsymbol{r}(\boldsymbol{\Phi}\boldsymbol{u}_r, \boldsymbol{\mu}, \boldsymbol{y}) = 0. \tag{6.10}$$

This work will primarily be consider minimum-residual reduced-order models (Definition 4.1), which completely prescribes the test basis $\boldsymbol{\Psi}$ based on the trial basis $\boldsymbol{\Phi}$ and optimality metric $\boldsymbol{\Theta}$. For a given $\boldsymbol{\mu} \in \mathbb{R}^{N_{\boldsymbol{\mu}}}$ and realization $\boldsymbol{y} \in \boldsymbol{\Xi}$, the solution of (6.10) will be denoted $\boldsymbol{u}_r(\boldsymbol{\mu}; \boldsymbol{y}, \boldsymbol{\Phi}, \boldsymbol{\Psi})$, which will be shortened to $\boldsymbol{u}_r(\boldsymbol{\mu}; \boldsymbol{y})$ when there is no risk of confusion regarding the choice of test and trial basis. From Theorem 2.1, $\boldsymbol{u}_r(\boldsymbol{\mu}; \boldsymbol{y}, \boldsymbol{\Phi}, \boldsymbol{\Psi})$ is a continuously differentiable function of $\boldsymbol{\mu}$. A realization of the reduced quantity of interest takes the form $f(\boldsymbol{\Phi}\boldsymbol{u}_r, \boldsymbol{\mu}, \boldsymbol{y})$, which can be considered solely a function of $\boldsymbol{\mu}$ and $\boldsymbol{y}$ through the implicit solution of (6.10)

$$F_r(\boldsymbol{\mu}; \boldsymbol{y}, \boldsymbol{\Phi}, \boldsymbol{\Psi}) = f(\boldsymbol{\Phi}\boldsymbol{u}_r(\boldsymbol{\mu}; \boldsymbol{y}, \boldsymbol{\Phi}, \boldsymbol{\Psi}), \boldsymbol{\mu}, \boldsymbol{y}). \tag{6.11}$$

Finally, the risk-neutral measure of the reduced quantity of interest, which serves as an approximation for the risk-neutral measure of the true quantity of interest in (6.4), is

$$F_r(\boldsymbol{\mu}; \boldsymbol{\Phi}, \boldsymbol{\Psi}) = \mathbb{E}[f(\boldsymbol{\Phi}\boldsymbol{u}_r(\boldsymbol{\mu}; \cdot, \boldsymbol{\Phi}, \boldsymbol{\Psi}), \boldsymbol{\mu}, \cdot)] = \mathbb{E}[F_r(\boldsymbol{\mu}; \cdot, \boldsymbol{\Phi}, \boldsymbol{\Psi})]. \tag{6.12}$$

Following the exposition in Sections 4.1.2 and 4.1.3, the gradient of the risk-neutral measure is computed according to the sensitivity or adjoint method as

$$\begin{aligned} \nabla F_r(\boldsymbol{\mu}; \boldsymbol{\Phi}, \boldsymbol{\Psi}) &= \mathbb{E}\left[\boldsymbol{g}^{\partial}\left(\boldsymbol{\Phi}\boldsymbol{u}_r(\boldsymbol{\mu}; \cdot, \boldsymbol{\Phi}, \boldsymbol{\Psi}), \boldsymbol{\Phi}\frac{\partial \boldsymbol{u}_r}{\partial \boldsymbol{\mu}}(\boldsymbol{\mu}; \cdot, \boldsymbol{\Phi}, \boldsymbol{\Psi}), \boldsymbol{\mu}, \cdot\right)\right] \\ &= \mathbb{E}\left[\boldsymbol{g}^{\boldsymbol{\lambda}}\left(\boldsymbol{\Phi}\boldsymbol{u}_r(\boldsymbol{\mu}; \cdot, \boldsymbol{\Phi}, \boldsymbol{\Psi}), \boldsymbol{\Psi}\boldsymbol{\lambda}_r(\boldsymbol{\mu}; \cdot, \boldsymbol{\Phi}, \boldsymbol{\Psi}), \boldsymbol{\mu}, \cdot\right)\right]. \end{aligned} \tag{6.13}$$

where $\boldsymbol{\Phi}\boldsymbol{u}_r(\boldsymbol{\mu}; \boldsymbol{y}, \boldsymbol{\Phi}, \boldsymbol{\Psi})$ is the reconstructed primal solution for realization $\boldsymbol{y} \in \boldsymbol{\Xi}$, $\dfrac{\partial \boldsymbol{u}_r}{\partial \boldsymbol{\mu}}(\boldsymbol{\mu}; \boldsymbol{y}, \boldsymbol{\Phi}, \boldsymbol{\Psi})$ is the reduced sensitivity, and $\boldsymbol{\lambda}_r(\boldsymbol{\mu}; \boldsymbol{y}, \boldsymbol{\Phi}, \boldsymbol{\Psi})$ is the reduced adjoint. The minimum-residual variants of the reduced gradient computation in (6.13) can be used in place of $\nabla F_r(\boldsymbol{\mu})$

$$\begin{aligned} \widehat{\nabla F_r}(\boldsymbol{\mu}; \boldsymbol{\Phi}, \boldsymbol{\Psi}, \boldsymbol{\Phi}^{\partial}, \boldsymbol{\Theta}^{\partial}) &= \mathbb{E}\left[\boldsymbol{g}^{\partial}\left(\boldsymbol{u}(\cdot), \boldsymbol{\Phi}^{\partial}\widehat{\frac{\partial \boldsymbol{u}_r}{\partial \boldsymbol{\mu}}}(\boldsymbol{\mu}; \cdot, \boldsymbol{\Phi}^{\partial}, \boldsymbol{\Theta}^{\partial}, \boldsymbol{u}(\cdot)), \boldsymbol{\mu}, \cdot\right)\right] \\ \widehat{\nabla F_r}(\boldsymbol{\mu}; \boldsymbol{\Phi}, \boldsymbol{\Psi}, \boldsymbol{\Phi}^{\boldsymbol{\lambda}}, \boldsymbol{\Theta}^{\boldsymbol{\lambda}}) &= \mathbb{E}\left[\boldsymbol{g}^{\boldsymbol{\lambda}}\left(\boldsymbol{u}(\cdot), \boldsymbol{\Phi}^{\boldsymbol{\lambda}}\hat{\boldsymbol{\lambda}}_r(\boldsymbol{\mu}; \cdot, \boldsymbol{\Phi}^{\boldsymbol{\lambda}}, \boldsymbol{\Theta}^{\boldsymbol{\lambda}}, \boldsymbol{u}(\cdot)), \boldsymbol{\mu}, \cdot\right)\right] \end{aligned} \tag{6.14}$$

where $\boldsymbol{u}(\boldsymbol{y}) = \boldsymbol{\Phi}\boldsymbol{u}_r(\boldsymbol{\mu};\, \boldsymbol{y},\, \boldsymbol{\Phi},\, \boldsymbol{\Psi})$ is the reconstructed primal solution for realization $\boldsymbol{y} \in \boldsymbol{\Xi}$, $\widehat{\frac{\partial \boldsymbol{u}_r}{\partial \boldsymbol{\mu}}}(\boldsymbol{\mu};\, \boldsymbol{y},\, \boldsymbol{\Phi}^{\partial},\, \boldsymbol{\Theta}^{\partial},\, \boldsymbol{u}(\boldsymbol{y}))$ is the solution of the minimum-residual sensitivity equations in (4.28) and $\hat{\boldsymbol{\lambda}}_r(\boldsymbol{\mu};\, \boldsymbol{y},\, \boldsymbol{\Phi}^{\boldsymbol{\lambda}},\, \boldsymbol{\Theta}^{\boldsymbol{\lambda}},\, \boldsymbol{u}(\boldsymbol{y}))$ is the solution of the minimum-residual adjoint equations in (4.56) for the realization corresponding to $\boldsymbol{y} \in \boldsymbol{\Xi}$. Using Propositions 4.3 and 4.5 as motivation, the sensitivity/adjoint bases and optimality metrics are chosen according to (4.35) and (4.63). This implies that the selection of $\boldsymbol{\Phi}$ and $\boldsymbol{\Psi}$ completely specify $\boldsymbol{\Phi}^{\partial}$ and $\boldsymbol{\Phi}^{\boldsymbol{\lambda}}$. From these propositions, the exact and minimum-residual sensitivities will only agree if the test basis $\boldsymbol{\Psi}$ is constant (such as a Galerkin projection) or the primal reduced-order model solution is exact for each realization $\boldsymbol{y} \in \boldsymbol{\Xi}$. Since training a reduced-order model to be exact for all $\boldsymbol{y} \in \boldsymbol{\Xi}$ is impractical, the relation $\nabla F_r(\boldsymbol{\mu}) = \widehat{\nabla F_r}(\boldsymbol{\mu})$ will only hold if the test basis is constant.

The residual-based error bounds derived in Appendix B hold for a particular realization $\boldsymbol{y} \in \boldsymbol{\Xi}$ of the stochastic PDE, provided Assumptions (AR1)–(AR8), (AQ1)–(AQ4) hold for this realization. The primal, sensitivity, and adjoint residual error bounds for a realization $\boldsymbol{y} \in \boldsymbol{\Xi}$ are

$$\begin{aligned}
|f(\boldsymbol{u}(\boldsymbol{\mu},\, \boldsymbol{y}),\, \boldsymbol{\mu},\, \boldsymbol{y}) - f(\boldsymbol{u},\, \boldsymbol{\mu},\, \boldsymbol{y})| &\leq \zeta \left\|\boldsymbol{r}(\boldsymbol{u},\, \boldsymbol{\mu},\, \boldsymbol{y})\right\| \\
\left\|\boldsymbol{g}^{\partial}\left(\boldsymbol{u}(\boldsymbol{\mu},\, \boldsymbol{y}),\, \frac{\partial \boldsymbol{u}}{\partial \boldsymbol{\mu}}(\boldsymbol{\mu},\, \boldsymbol{y}),\, \boldsymbol{\mu},\, \boldsymbol{y}\right) - \boldsymbol{g}^{\partial}(\boldsymbol{u},\, \boldsymbol{w},\, \boldsymbol{\mu},\, \boldsymbol{y})\right\| &\leq \kappa \left\|\boldsymbol{r}(\boldsymbol{u},\, \boldsymbol{\mu},\, \boldsymbol{y})\right\| + \tau \left\|\boldsymbol{r}^{\partial}(\boldsymbol{u},\, \boldsymbol{w},\, \boldsymbol{\mu},\, \boldsymbol{y})\right\| \\
\left\|\boldsymbol{g}^{\boldsymbol{\lambda}}(\boldsymbol{u}(\boldsymbol{\mu},\, \boldsymbol{y}),\, \boldsymbol{\lambda}(\boldsymbol{\mu},\, \boldsymbol{y}),\, \boldsymbol{\mu},\, \boldsymbol{y}) - \boldsymbol{g}^{\boldsymbol{\lambda}}(\boldsymbol{u},\, \boldsymbol{z},\, \boldsymbol{\mu},\, \boldsymbol{y})\right\| &\leq \kappa \left\|\boldsymbol{r}(\boldsymbol{u},\, \boldsymbol{\mu},\, \boldsymbol{y})\right\| + \tau \left\|\boldsymbol{r}^{\boldsymbol{\lambda}}(\boldsymbol{u},\, \boldsymbol{z},\, \boldsymbol{\mu},\, \boldsymbol{y})\right\|
\end{aligned} \tag{6.15}$$

where $\boldsymbol{u} = \boldsymbol{\Phi}\boldsymbol{u}_r(\boldsymbol{\mu};\, \boldsymbol{y},\, \boldsymbol{\Phi},\, \boldsymbol{\Psi})$ is the reconstructed primal solution, $\boldsymbol{w}$ is the reconstructed reduced sensitivity (exact or minimum-residual), i.e., $\boldsymbol{w} = \boldsymbol{\Phi}\frac{\partial \boldsymbol{u}_r}{\partial \boldsymbol{\mu}}(\boldsymbol{\mu};\, \boldsymbol{y},\, \boldsymbol{\Phi},\, \boldsymbol{\Psi})$ or $\boldsymbol{w} = \boldsymbol{\Phi}^{\partial}\widehat{\frac{\partial \boldsymbol{u}_r}{\partial \boldsymbol{\mu}}}(\boldsymbol{\mu};\, \boldsymbol{y},\, \boldsymbol{\Phi}^{\partial},\, \boldsymbol{\Theta}^{\partial},\, \boldsymbol{u})$, and $\boldsymbol{z}$ is the reconstructed reduced adjoint (exact or minimum-residual), i.e., $\boldsymbol{z} = \boldsymbol{\Psi}\boldsymbol{\lambda}_r(\boldsymbol{\mu};\, \boldsymbol{y},\, \boldsymbol{\Phi},\, \boldsymbol{\Psi})$ or $\boldsymbol{z} = \boldsymbol{\Phi}^{\boldsymbol{\lambda}}\boldsymbol{\lambda}_r(\boldsymbol{\mu};\, \boldsymbol{y},\, \boldsymbol{\Phi}^{\boldsymbol{\lambda}},\, \boldsymbol{\Theta}^{\boldsymbol{\lambda}},\, \boldsymbol{u})$.

Finally, the stochastic generalization of the collocation-based hyperreduced models of Section 4.2 follows immediately from the construction in that section and takes the form

$$(\boldsymbol{P}^T\boldsymbol{\Psi})^T\boldsymbol{P}^T\boldsymbol{r}(\boldsymbol{\Phi}\boldsymbol{u}_r,\, \boldsymbol{\mu},\, \boldsymbol{y}) = 0 \qquad \forall \boldsymbol{y} \in \boldsymbol{\Xi} \tag{6.16}$$

The case of stochastic hyperreduction will not be considered further as only problems amenable to precomputations (polynomial nonlinearities) will be considered in the numerical experiments (Section 6.4).

While the introduction of the stochastic reduced-order and hyperreduced models in this section reduces the cost of evaluating risk-averse measures of PDE quantities of interest, e.g., for stochastic optimization, they may still be prohibitively expensive due to the curse of dimensionality. The next section introduces anisotropic sparse grids to mitigate or delay the impact of the curse of dimensionality when evaluating risk-averse measures in moderate-to-large dimensional stochastic spaces.

6.1.3 Anisotropic Sparse Grids

Consider the difficult problem of evaluating the expectation of a smooth function $g : \mathbb{R}^{N_{\boldsymbol{y}}} \to \mathbb{R}$

$$\mathbb{E}[g] = \int_{\boldsymbol{\Xi}} \rho(\boldsymbol{y}) g(\boldsymbol{y})\, d\boldsymbol{y} \tag{6.17}$$

where $\boldsymbol{\Xi} \subset [-1, 1]^{N_{\boldsymbol{y}}}$ is the stochastic space and $\rho : \boldsymbol{\Xi} \to \mathbb{R}_+$ is the joint probability density function with marginal probability density functions $\rho_k : \Xi_k \to \mathbb{R}_+$ for $k = 1, \ldots, N_{\boldsymbol{y}}$ such that $\rho = \rho_1 \otimes \cdots \otimes \rho_{N_y}$). When $N_{\boldsymbol{y}}$ is moderate-to-large, the evaluation of the integral in (6.17) is difficult since multidimensional quadrature rules derived from optimal 1D quadrature rules suffer from the *curse of dimensionality*. Isotropic sparse grids, originally introduced in [184] and extensively studied since [144, 66, 145, 156, 18, 157], generate efficient quadrature rules that *delay* the influence of the curse of dimensionality and allows for larger stochastic spaces to be considered. Anisotropic sparse grids [67] further optimize the quadrature rules by leveraging anisotropy of the integrand.

The sparse grid construction begins with the definition of a one-dimensional quadrature rule of level i that will be used in the kth dimension, $\mathbb{E}_k^i$. The level is an integer used to indicate refinement of the one-dimensional quadrature rule such that

$$\mathbb{E}_k^i[h] \to \mathbb{E}_k[h] = \int_{\Xi_k} \rho_k(y) h(y)\, dy \quad \text{as} \quad i \to \infty. \tag{6.18}$$

for $h : \Xi_k \to \mathbb{R}$. Let $\Xi_k^i \subset [-1, 1]$ be the quadrature nodes associated with the quadrature rule $\mathbb{E}_k^i$. While the sparse grid construction to follow holds for any valid and refinable quadrature rule that satisfies (6.18), only *nested* quadrature rules will be considered. That is, the nodes at level i are a subset of the nodes at level $i + 1$, $\Xi_k^i \subset \Xi_k^{i+1}$. The nested property will not be used in the sparse grid construction, but leads to an efficient implementation since, at level $i + 1$, only $h(y)$ must be evaluated for $y \in \Xi_k^{i+1} \setminus \Xi_k^i$.

From the one-dimensional quadrature rules, the corresponding difference operators are defined as

$$\Delta_k^1 \coloneqq \mathbb{E}_k^1 \qquad \text{and} \qquad \Delta_k^i \coloneqq \mathbb{E}_k^i - \mathbb{E}_k^{i-1} \quad \text{for} \quad i \geq 2. \tag{6.19}$$

The requirement in (6.18) on the quadrature rules implies $\Delta_k^i[g] \to 0$ as $i \to 0$. The one-dimensional quadrature rule $\mathbb{E}_k^i$ is recovered by summing over all difference operators in dimension k up through level i

$$\mathbb{E}_k^i = \sum_{j=1}^{i} \Delta_k^j \tag{6.20}$$

since the sum telescopes due to the definition of Δ_k^i in (6.19). A multi-dimensional difference operator is constructed from a tensor product of one-dimensional difference operators, each possibly at a different level of refinement

$$\Delta^{\boldsymbol{i}} \coloneqq \Delta_1^{i_1} \otimes \cdots \otimes \Delta_{N_{\boldsymbol{y}}}^{i_{N_{\boldsymbol{y}}}}. \tag{6.21}$$

A multi-index $\boldsymbol{i} \in \mathbb{N}_+^{N_{\boldsymbol{y}}}$ with components $\boldsymbol{i} = (i_1, \ldots, i_{N_{\boldsymbol{y}}})$ is used to track the refinement level of

each one-dimensional difference operator, i.e., i_k is the refinement level of the difference operator in dimension k. From the multi-dimensional difference operator, a quadrature rule $\mathbb{E}_{\mathcal{I}}$ is defined by summing over all multi-indices in a multi-index set $\mathcal{I} \subset \mathbb{N}_+^{N_{\boldsymbol{y}}}$

$$\mathbb{E}_{\mathcal{I}} = \sum_{\boldsymbol{i} \in \mathcal{I}} \Delta^{\boldsymbol{i}} := \sum_{\boldsymbol{i} \in \mathcal{I}} \Delta_1^{i_1} \otimes \cdots \otimes \Delta_{N_{\boldsymbol{y}}}^{i_{N_{\boldsymbol{y}}}}. \tag{6.22}$$

Let $\boldsymbol{\Xi}_{\mathcal{I}} \subset [-1, 1]^{N_{\boldsymbol{y}}}$ denote the quadrature nodes associated with the multi-dimensional quadrature rule $\mathbb{E}_{\mathcal{I}}$. The use of nested, one-dimensional quadrature rules implies $\boldsymbol{\Xi}_{\mathcal{I}} \subset \boldsymbol{\Xi}_{\mathcal{J}}$ for $\mathcal{I}$, $\mathcal{J}$ multi-index sets such that $\mathcal{I} \subset \mathcal{J}$. In the multi-dimensional case, this leads to substantial savings as evaluations of g can be recycled as the sparse grid is refined.

For $\mathbb{E}_{\mathcal{I}}$ to be a convergent quadrature rule, i.e., $\mathbb{E}_{\mathcal{I}} \to \mathbb{E}$ as $\mathcal{I} \to \mathbb{N}_+^{N_{\boldsymbol{y}}}$, a telescoping property similar to that in (6.20) must hold. This requirement is satisfied if the multi-index $\mathcal{I}$ is admissible in the sense of Definition 6.1.

Definition 6.1. *An index set* $\mathcal{I} \subset \mathbb{N}_+^{N_{\boldsymbol{y}}}$ *is* admissible *if for all* $\boldsymbol{k} \in \mathcal{I}$,

$$\boldsymbol{k} - \boldsymbol{e}_j \in \mathcal{I} \quad \textit{for} \quad 1 \leq j \leq N_{\boldsymbol{y}},\ k_j > 1 \tag{6.23}$$

This completes the construction of general, anisotropic sparse grids. From this general construction, some well-known special cases can be recovered. The tensor product quadrature rule of level i, $\mathbb{E}_1^i \otimes \cdots \otimes \mathbb{E}_{N_{\boldsymbol{y}}}^i$, can written as $\mathbb{E}_{\mathcal{I}_\infty^i}$ with $\mathcal{I}_\infty^i = \{\boldsymbol{i} \in \mathbb{N}_+^{N_{\boldsymbol{y}}} \mid |\boldsymbol{i}|_\infty \leq i\}$, i.e.,

$$\mathbb{E}_1^i \otimes \cdots \otimes \mathbb{E}_{N_{\boldsymbol{y}}}^i[g] = \sum_{|\boldsymbol{i}|_\infty \leq i} (\Delta_1^i \otimes \cdots \otimes \Delta_{N_{\boldsymbol{y}}}^i)[g] = \mathbb{E}_{\mathcal{I}_\infty^i}. \tag{6.24}$$

Figure 6.1 provides an example of a tensor product quadrature rule, and the corresponding index set, based on Clenshaw-Curtis quadrature rules; the index set is dense and leads to a quadrature rule with the maximum number of nodes. The isotropic Smolyak sparse grid of level i is

$$\sum_{|\boldsymbol{i}|_1 \leq i + N_{\boldsymbol{y}} + 1} (\Delta_1^i \otimes \cdots \otimes \Delta_{N_{\boldsymbol{y}}}^i)[g] = \mathbb{E}_{\mathcal{I}_{\text{iso}}^i}, \tag{6.25}$$

where $\mathcal{I}_{\text{iso}}^i = \{\boldsymbol{i} \in \mathbb{N}_+^{N_{\boldsymbol{y}}} \mid |\boldsymbol{i}|_1 \leq i + N_{\boldsymbol{y}} - 1\}$. See Figure 6.2 for an illustration of the quadrature nodes and index set; the index set is only refined along the diagonal, which leads to much sparser quadrature rules than direct tensor products. Finally, Figure 6.3 illustrates an anisotropic sparse grid, including quadrature nodes and index set, which further reduces the number of quadrature compared to the other options and (potentially) takes advantage of anisotropy in the integrand $g(\boldsymbol{y})$ and probability density function $\rho(\boldsymbol{y})$.

The *neighbor* of a sparse grid is the final concept introduced in this section and will be used extensively in assessing the truncation error that arises from approximating $\mathbb{E}[g]$ by $\mathbb{E}_{\mathcal{I}}[g]$. The set

of *neighbors* corresponding to a sparse grid $\mathcal{I} \subset \mathbb{N}_+^{N_{\boldsymbol{y}}}$, denoted $\mathcal{N}(\mathcal{I}) \subset \mathbb{N}_+^{N_{\boldsymbol{y}}}$ is defined as

$$\mathcal{N}(\mathcal{I}) \coloneqq \{\boldsymbol{i} \in \mathcal{I}^c \mid \mathcal{I} \cup \{\boldsymbol{i}\} \text{ is admissible}\} \tag{6.26}$$

where $\mathcal{I}^c$ is the complement of the multi-index set $\mathcal{I}$ in $\mathbb{N}_+^{N_{\boldsymbol{y}}}$, i.e., $\mathcal{I}^c = \{\boldsymbol{i} \in \mathbb{N}_+^{N_{\boldsymbol{y}}} \mid \boldsymbol{i} \notin \mathcal{I}\}$. Figure 6.4 shows the quadrature nodes and index set corresponding to the anisotropic sparse grid, including neighbors, in Figure 6.3. Following the work in [67, 108, 109], the truncation error, which can be written as the infinite sum

$$\mathbb{E}[g] - \mathbb{E}_{\mathcal{I}}[g] = \sum_{\boldsymbol{i} \in \mathcal{I}^c} (\Delta_1^{i_1} \otimes \cdots \otimes \Delta_{N_{\boldsymbol{y}}}^{i_{N_{\boldsymbol{y}}}})[g] \tag{6.27}$$

can be approximated as

$$\mathbb{E}[g] - \mathbb{E}_{\mathcal{I}}[g] \approx \sum_{\boldsymbol{i} \in \mathcal{N}(\mathcal{I})} (\Delta_1^{i_1} \otimes \cdots \otimes \Delta_{N_{\boldsymbol{y}}}^{i_{N_{\boldsymbol{y}}}})[g]. \tag{6.28}$$

The concept and notation used to represent neighbors of a sparse grid is easily extended to handle j layers of neighbors, that is, $\mathcal{N}(\mathcal{I})$ is the 1st layer of neighbors, $\mathcal{N}(\mathcal{N}(\mathcal{I}))$ is the 2nd layer, and $\mathcal{N}^j(\mathcal{I}) \coloneqq \underbrace{\mathcal{N} \circ \cdots \circ \mathcal{N}}_{j \text{ terms}} \circ \mathcal{I}$ is the jth layer. A more accurate approximation of the truncation error is attainable by including more distant neighbors, but expense of the corresponding computation rapidly increases. For this reason, usually only the first layer of neighbors is used to approximate the truncation error [67, 108, 109], which is the approach taken in this work.

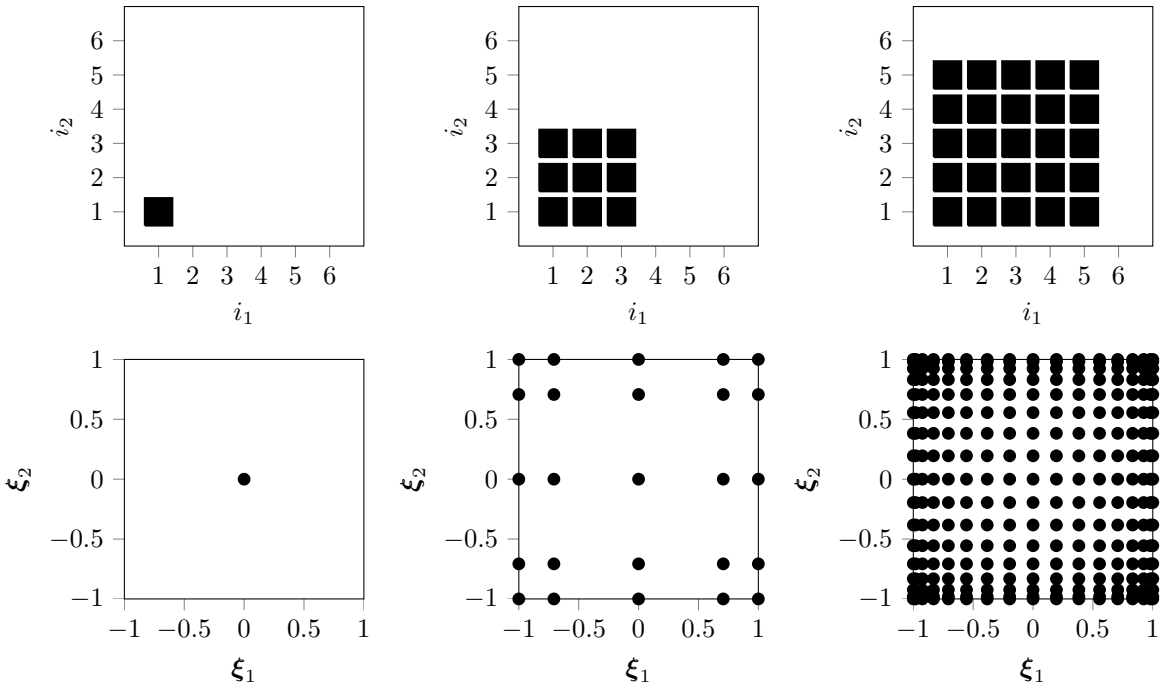


Figure 6.1: Full tensor product based on Clenshaw-Curtis (levels 1, 3, 5)

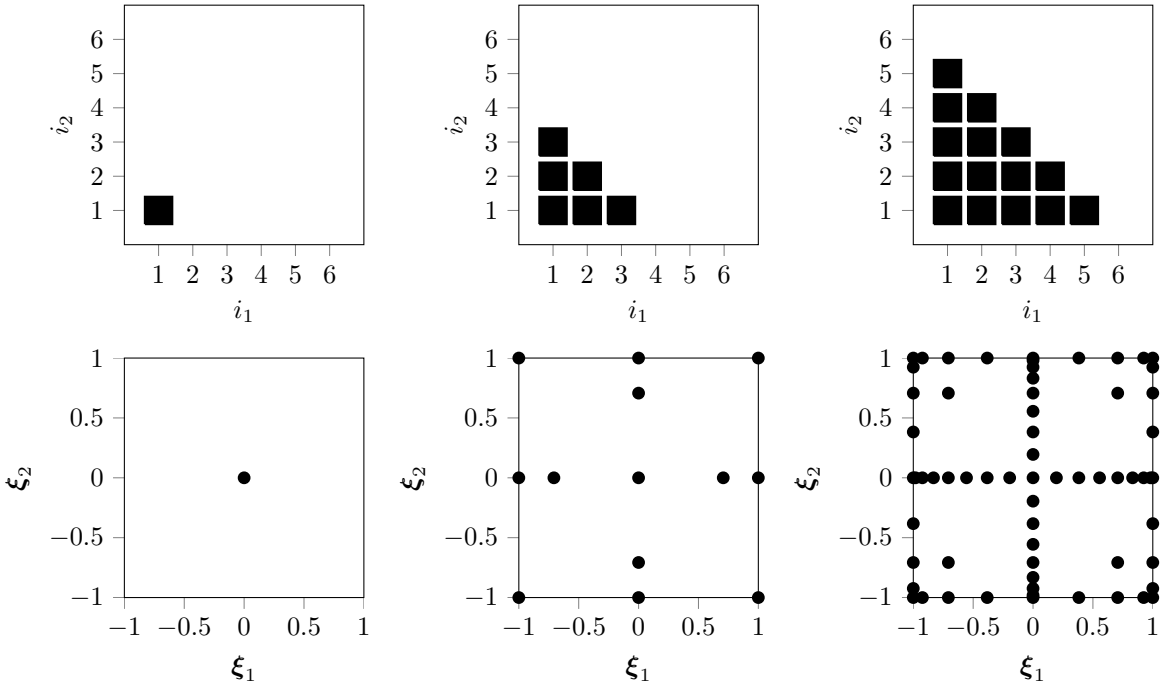


Figure 6.2: Isotropic sparse grid based on Clenshaw-Curtis (levels 1, 3, 5)

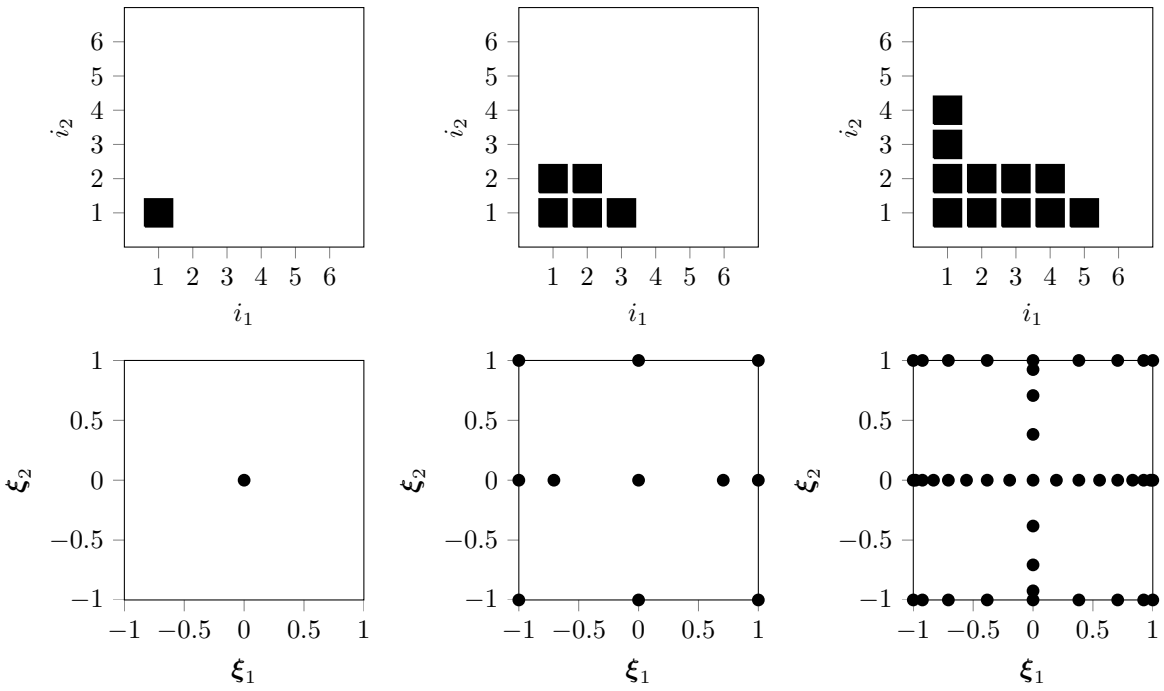


Figure 6.3: Anisotropic sparse grid based on Clenshaw-Curtis (levels 1, 3, 5)

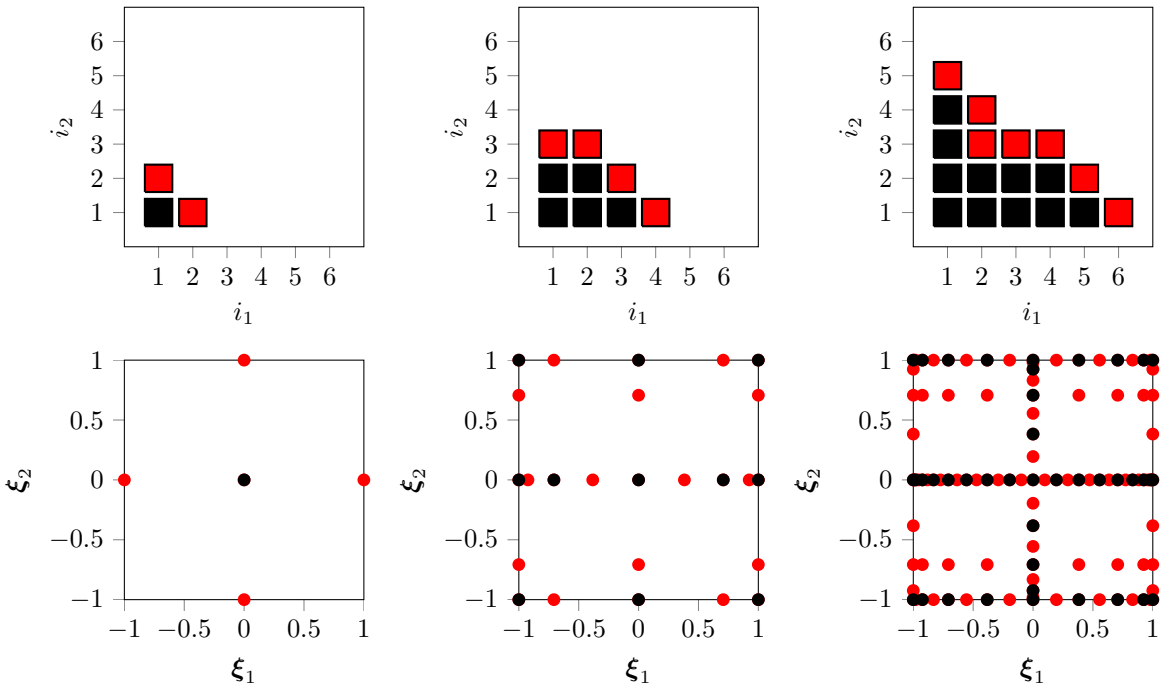


Figure 6.4: Anisotropic sparse grid based on Clenshaw-Curtis with all (including non-admissible) forward neighbors (levels 1, 3, 5)

6.2 Two Levels of Approximation of Risk-Averse Measures

The two approximation technologies introduced—anisotropic sparse grids for the efficient approximation of high-dimensional integrals and stochastic reduced-order models to reduce the cost associated with solving a realization of the SPDE—are combined to yield an inexpensive approximation of risk-averse measures of quantities of interest of high-fidelity partial differential equations. For the remainder of this section, suppose a sparse grid $\mathcal{I}$ and reduced-order model $(\boldsymbol{\Phi}, \boldsymbol{\Psi})$ are given – the construction of each will be considered in detail in Section 6.3.2. The two-level approximation of the risk-averse measure of the quantity of interest in (6.4) is

$$F_r(\boldsymbol{\mu}; \boldsymbol{\Phi}, \boldsymbol{\Psi}, \mathcal{I}) \coloneqq \mathbb{E}_{\mathcal{I}}[f(\boldsymbol{\Phi}\boldsymbol{u}_r(\boldsymbol{\mu}; \cdot, \boldsymbol{\Phi}, \boldsymbol{\Psi}), \boldsymbol{\mu}, \cdot)]. \tag{6.29}$$

The introduction of the sparse grid introduces a truncation error into the evaluation of the integral and the reduced-order model introduces an error in the evaluation of the quantity of interest at each collocation node. The benefit of such an approximation is that the many high-dimensional model solutions required to evaluate $F(\boldsymbol{\mu})$ are replace by few reduced-order model solutions to evaluate $F_r(\boldsymbol{\mu})$. The introduction of the sparse grid further benefits the reduced-order model since it only needs to be trained *on the collocation nodes* instead of everywhere in $\boldsymbol{\Xi}$. The gradient of the approximation in (6.29) is computed according to the sensitivity or adjoint method as

$$\begin{aligned}\nabla F_r(\boldsymbol{\mu}; \boldsymbol{\Phi}, \boldsymbol{\Psi}, \mathcal{I}) &= \mathbb{E}_{\mathcal{I}}\left[\boldsymbol{g}^{\partial}\left(\boldsymbol{\Phi}\boldsymbol{u}_r(\boldsymbol{\mu}; \cdot, \boldsymbol{\Phi}, \boldsymbol{\Psi}), \boldsymbol{\Phi}\frac{\partial \boldsymbol{u}_r}{\partial \boldsymbol{\mu}}(\boldsymbol{\mu}; \cdot, \boldsymbol{\Phi}, \boldsymbol{\Psi}), \boldsymbol{\mu}, \cdot\right)\right] \\ &= \mathbb{E}_{\mathcal{I}}\left[\boldsymbol{g}^{\boldsymbol{\lambda}}\left(\boldsymbol{\Phi}\boldsymbol{u}_r(\boldsymbol{\mu}; \cdot, \boldsymbol{\Phi}, \boldsymbol{\Psi}), \boldsymbol{\Psi}\boldsymbol{\lambda}_r(\boldsymbol{\mu}; \cdot, \boldsymbol{\Phi}, \boldsymbol{\Psi}), \boldsymbol{\mu}, \cdot\right)\right].\end{aligned} \tag{6.30}$$

If the true sensitivity and adjoint of the stochastic reduced-order model are too cumbersome to compute, i.e., if second derivatives of $\boldsymbol{r}$ are required (see Chapter 4), the minimum-residual variants can be used to compute an approximation to $\nabla F_r(\boldsymbol{\mu})$ as

$$\begin{aligned}\widehat{\nabla F_r}(\boldsymbol{\mu}; \boldsymbol{\Phi}, \boldsymbol{\Psi}, \boldsymbol{\Phi}^{\partial}, \boldsymbol{\Theta}^{\partial}, \mathcal{I}) &= \mathbb{E}_{\mathcal{I}}\left[\boldsymbol{g}^{\partial}\left(\boldsymbol{u}(\cdot), \boldsymbol{\Phi}^{\partial}\widehat{\frac{\partial \boldsymbol{u}_r}{\partial \boldsymbol{\mu}}}(\boldsymbol{\mu}; \cdot, \boldsymbol{\Phi}^{\partial}, \boldsymbol{\Theta}^{\partial}, \boldsymbol{u}(\cdot)), \boldsymbol{\mu}, \cdot\right)\right] \\ \widehat{\nabla F_r}(\boldsymbol{\mu}; \boldsymbol{\Phi}, \boldsymbol{\Psi}, \boldsymbol{\Phi}^{\boldsymbol{\lambda}}, \boldsymbol{\Theta}^{\boldsymbol{\lambda}}, \mathcal{I}) &= \mathbb{E}_{\mathcal{I}}\left[\boldsymbol{g}^{\boldsymbol{\lambda}}\left(\boldsymbol{u}(\cdot), \boldsymbol{\Phi}^{\boldsymbol{\lambda}}\hat{\boldsymbol{\lambda}}_r(\boldsymbol{\mu}; \cdot, \boldsymbol{\Phi}^{\boldsymbol{\lambda}}, \boldsymbol{\Theta}^{\boldsymbol{\lambda}}, \boldsymbol{u}(\cdot)), \boldsymbol{\mu}, \cdot\right)\right]\end{aligned} \tag{6.31}$$

where $\boldsymbol{u}(\boldsymbol{y}) = \boldsymbol{\Phi}\boldsymbol{u}_r(\boldsymbol{\mu}; \boldsymbol{y}, \boldsymbol{\Phi}, \boldsymbol{\Psi})$ is the reconstructed primal reduced-order model solution of the realization corresponding to $\boldsymbol{y} \in \boldsymbol{\Xi}$.

The error incurred by approximating $F(\boldsymbol{\mu})$ with $F_r(\boldsymbol{\mu}; \boldsymbol{\Phi}, \boldsymbol{\Psi}, \mathcal{I})$ must account for both the truncation error introduced by the sparse grid and the pointwise error in the reduced-order model. These terms arise naturally from a simple application of the triangle inequality to the error

$$\begin{aligned}|F(\boldsymbol{\mu}) - F_r(\boldsymbol{\mu}; \boldsymbol{\Phi}, \boldsymbol{\Psi}, \mathcal{I})| &= |\mathbb{E}[f(\boldsymbol{u}(\boldsymbol{\mu}; \cdot), \boldsymbol{\mu}, \cdot) - \mathbb{E}_{\mathcal{I}}[f(\boldsymbol{\Phi}\boldsymbol{u}_r(\boldsymbol{\mu}; \cdot, \boldsymbol{\Phi}, \boldsymbol{\Psi}), \boldsymbol{\mu}, \cdot)]| \\ &\leq \mathbb{E}[|f(\boldsymbol{u}(\boldsymbol{\mu}; \cdot), \boldsymbol{\mu}, \cdot) - f(\boldsymbol{\Phi}\boldsymbol{u}_r(\boldsymbol{\mu}; \cdot, \boldsymbol{\Phi}, \boldsymbol{\Psi}), \boldsymbol{\mu}, \cdot)|] \\ &\quad + \mathbb{E}_{\mathcal{I}^c}[|f(\boldsymbol{\Phi}\boldsymbol{u}_r(\boldsymbol{\mu}; \cdot, \boldsymbol{\Phi}, \boldsymbol{\Psi}), \boldsymbol{\mu}, \cdot)|]\end{aligned} \tag{6.32}$$

where $\mathbb{E}_{\mathcal{I}^c} \coloneqq \mathbb{E} - \mathbb{E}_{\mathcal{I}}$ was used (Section 6.1.3). The first term in the error bound is the integrated reduced-order model error and the second is the truncation error that results from using the sparse grid $\mathcal{I}$ to integrate the reduced quantity of interest. While the error bound is instructive in understanding the sources of error, it can not be efficiently computed due to the presence of the true error in the first integrand and the infinite sum required to compute the expectations in both terms (quadrature over an infinite set of collocation nodes to compute the integral exactly). The residual-based error bounds from Appendix B are used to circumvent the first issue by bounding the true error by an arbitrary constant ($\zeta > 0$) times the residual norm

$$\begin{aligned} |F(\boldsymbol{\mu}) - F_r(\boldsymbol{\mu};\, \boldsymbol{\Phi},\, \boldsymbol{\Psi},\, \mathcal{I})| &\leq \zeta \mathbb{E}[\|\boldsymbol{r}(\boldsymbol{\Phi}\boldsymbol{u}_r(\boldsymbol{\mu};\, \cdot\,,\, \boldsymbol{\Phi},\, \boldsymbol{\Psi}),\, \boldsymbol{\mu},\, \cdot\,)\|] + \mathbb{E}_{\mathcal{I}^c}[|f(\boldsymbol{\Phi}\boldsymbol{u}_r(\boldsymbol{\mu};\, \cdot\,,\, \boldsymbol{\Phi},\, \boldsymbol{\Psi}),\, \boldsymbol{\mu},\, \cdot\,)|] \\ &= \zeta \mathbb{E}[\|\boldsymbol{r}(\boldsymbol{\Phi}\boldsymbol{u}_r(\boldsymbol{\mu};\, \cdot\,,\, \boldsymbol{\Phi},\, \boldsymbol{\Psi}),\, \boldsymbol{\mu},\, \cdot\,)\|] + \mathbb{E}_{\mathcal{I}^c}[|F_r(\boldsymbol{\mu};\, \cdot\,,\, \boldsymbol{\Phi},\, \boldsymbol{\Psi})|], \end{aligned} \tag{6.33}$$

where the definition of F_r, introduced in Section 6.1.2, was used in the second line. The infinite sums required to compute both expectations are reduced to finite sums by approximating the complement of the sparse grid $\mathcal{I}^c$ (infinite set of collocation points) with the forward neighbors of the sparse grid $\mathcal{N}(\mathcal{I})$ (finite set of collocation points). With this approximation, the expectation operator $\mathbb{E}$ and truncation operator $\mathbb{E}_{\mathcal{I}^c}$ become

$$\mathbb{E} \coloneqq \mathbb{E}_{\mathcal{I}\cup\mathcal{I}^c} \approx \mathbb{E}_{\mathcal{I}\cup\mathcal{N}(\mathcal{I})} \qquad \text{and} \qquad \mathbb{E}_{\mathcal{I}^c} \approx \mathbb{E}_{\mathcal{N}(\mathcal{I})}. \tag{6.34}$$

The introduction of this approximation into the error bound in (6.33) reduces the uncomputable right-hand side (due to the infinite sums required for evaluation of the expectation and truncation operators) to

$$|F(\boldsymbol{\mu}) - F_r(\boldsymbol{\mu};\, \boldsymbol{\Phi},\, \boldsymbol{\Psi},\, \mathcal{I})| \lesssim \zeta \mathbb{E}_{\mathcal{I}\cup\mathcal{N}(\mathcal{I})}[\|\boldsymbol{r}(\boldsymbol{\Phi}\boldsymbol{u}_r(\boldsymbol{\mu};\, \cdot\,,\, \boldsymbol{\Phi},\, \boldsymbol{\Psi}),\, \boldsymbol{\mu},\, \cdot\,)\|] + \mathbb{E}_{\mathcal{N}(\mathcal{I})}[|F_r(\boldsymbol{\mu};\, \cdot\,,\, \boldsymbol{\Phi},\, \boldsymbol{\Psi})|], \tag{6.35}$$

which is amenable to computation as only finite summations are required. In general, the right-hand side of (6.35) does not *bound* the left-hand side due to the introduction of the *approximation* $\mathcal{I}^c \approx \mathcal{N}(\mathcal{I})$. While this approximation does not necessarily preserve the error bound in (6.33), it leads to an inexpensive error indicators: the right-hand side of (6.35) only requires reduced-order models solves and residual evaluations on the sparse grid $\mathcal{I}$ and its neighbors $\mathcal{N}(\mathcal{I})$.

An identical procedure is carried out to convert the pointwise error bounds in (6.13) for a given realization of the stochastic PDE to an inexpensive error indicator. The error indicator for gradients computed via the sensitivity method takes the form

$$\begin{aligned} |\nabla F(\boldsymbol{\mu}) - \nabla F_r(\boldsymbol{\mu};\, \boldsymbol{\Phi},\, \boldsymbol{\Psi})| \lesssim\ & \kappa \mathbb{E}_{\mathcal{I}\cup\mathcal{N}(\mathcal{I})}\left[\|\boldsymbol{r}(\boldsymbol{\Phi}\boldsymbol{u}_r(\boldsymbol{\mu};\, \cdot\,,\, \boldsymbol{\Phi},\, \boldsymbol{\Psi}),\, \boldsymbol{\mu},\, \cdot\,)\|\right] + \\ & \tau \mathbb{E}_{\mathcal{I}\cup\mathcal{N}(\mathcal{I})}\left[\left\|\boldsymbol{r}^{\partial}(\boldsymbol{\Phi}\boldsymbol{u}_r(\boldsymbol{\mu};\, \cdot\,,\, \boldsymbol{\Phi},\, \boldsymbol{\Psi}),\, \boldsymbol{\Phi}\frac{\partial \boldsymbol{u}_r}{\partial \boldsymbol{\mu}}(\boldsymbol{\mu};\, \cdot\,,\, \boldsymbol{\Phi},\, \boldsymbol{\Psi}),\, \boldsymbol{\mu},\, \cdot\,)\right\|\right] + \\ & \mathbb{E}_{\mathcal{N}(\mathcal{I})}\left[\|\nabla F_r(\boldsymbol{\mu};\, \boldsymbol{\Phi},\, \boldsymbol{\Psi})\|\right] \end{aligned} \tag{6.36}$$

and for gradients computed via the adjoint method, it takes the form

$$\begin{aligned}|\nabla F(\boldsymbol{\mu}) - \nabla F_r(\boldsymbol{\mu};\, \boldsymbol{\Phi},\, \boldsymbol{\Psi})| \lesssim \kappa \mathbb{E}_{\mathcal{I}\cup\mathcal{N}(\mathcal{I})}\left[\|\boldsymbol{r}(\boldsymbol{\Phi}\boldsymbol{u}_r(\boldsymbol{\mu};\,\cdot\,,\,\boldsymbol{\Phi},\,\boldsymbol{\Psi}),\,\boldsymbol{\mu},\,\cdot\,)\|\right] + \\ \tau \mathbb{E}_{\mathcal{I}\cup\mathcal{N}(\mathcal{I})}\left[\left\|\boldsymbol{r}^{\boldsymbol{\lambda}}(\boldsymbol{\Phi}\boldsymbol{u}_r(\boldsymbol{\mu};\,\cdot\,,\,\boldsymbol{\Phi},\,\boldsymbol{\Psi}),\,\boldsymbol{\Psi}\boldsymbol{\lambda}_r(\boldsymbol{\mu};\,\cdot\,,\,\boldsymbol{\Phi},\,\boldsymbol{\Psi}),\,\boldsymbol{\mu},\,\cdot\,)\right\|\right] + \\ \mathbb{E}_{\mathcal{N}(\mathcal{I})}\left[\|\nabla F_r(\boldsymbol{\mu};\,\boldsymbol{\Phi},\,\boldsymbol{\Psi})\|\right].\end{aligned} \tag{6.37}$$

Similar to the error indicator in (6.35) for the value of quantity of interest, the gradient error indicators in (6.36) and (6.37) have terms that separately account for the reduced-order model error and integral truncation error. The three terms in these error indicators account for the error in the primal reduced-order model solution, the error in the reduced-order model sensitivity/adjoint, and truncation error from approximating the expectation operators with the sparse grid $\mathcal{I}$, respectively. The gradient error indicators must include the terms that accounts for the error in the primal solution since, in general, the sensitivity/adjoint equations are defined about an approximate linearization point. The gradient error indicators for the minimum-residual sensitivity and adjoint reduced-order model follow in a similar manner

$$\begin{aligned}|\nabla F(\boldsymbol{\mu}) - \widehat{\nabla F_r}(\boldsymbol{\mu};\,\boldsymbol{\Phi},\,\boldsymbol{\Psi},\,\boldsymbol{\Phi}^{\partial},\,\boldsymbol{\Theta}^{\partial})| \lesssim \kappa \mathbb{E}_{\mathcal{I}\cup\mathcal{N}(\mathcal{I})}\left[\|\boldsymbol{r}(\boldsymbol{u}(\,\cdot\,),\,\boldsymbol{\mu},\,\cdot\,)\|\right] + \\ \tau \mathbb{E}_{\mathcal{I}\cup\mathcal{N}(\mathcal{I})}\left[\left\|\boldsymbol{r}^{\partial}(\boldsymbol{u}(\,\cdot\,),\,\boldsymbol{\Phi}^{\partial}\widehat{\frac{\partial \boldsymbol{u}_r}{\partial \boldsymbol{\mu}}}(\boldsymbol{\mu};\,\cdot\,,\,\boldsymbol{\Phi}^{\partial},\,\boldsymbol{\Theta}^{\partial};\,\boldsymbol{u}(\,\cdot\,)),\,\boldsymbol{\mu},\,\cdot\,)\right\|\right] + \\ \mathbb{E}_{\mathcal{N}(\mathcal{I})}\left[\left\|\widehat{\nabla F_r}(\boldsymbol{\mu};\,\boldsymbol{\Phi},\,\boldsymbol{\Psi},\,\boldsymbol{\Phi}^{\partial},\,\boldsymbol{\Theta}^{\partial})\right\|\right] \\ |\nabla F(\boldsymbol{\mu}) - \widehat{\nabla F_r}(\boldsymbol{\mu};\,\boldsymbol{\Phi},\,\boldsymbol{\Psi},\,\boldsymbol{\Phi}^{\boldsymbol{\lambda}},\,\boldsymbol{\Theta}^{\boldsymbol{\lambda}})| \lesssim \kappa \mathbb{E}_{\mathcal{I}\cup\mathcal{N}(\mathcal{I})}\left[\|\boldsymbol{r}(\boldsymbol{\Phi}\boldsymbol{u}_r(\boldsymbol{\mu};\,\cdot\,,\,\boldsymbol{\Phi},\,\boldsymbol{\Psi}),\,\boldsymbol{\mu},\,\cdot\,)\|\right] + \\ \tau \mathbb{E}_{\mathcal{I}\cup\mathcal{N}(\mathcal{I})}\left[\left\|\boldsymbol{r}^{\boldsymbol{\lambda}}(\boldsymbol{u}(\,\cdot\,),\,\boldsymbol{\Phi}^{\boldsymbol{\lambda}}\hat{\boldsymbol{\lambda}}_r(\boldsymbol{\mu};\,\cdot\,,\,\boldsymbol{\Phi}^{\boldsymbol{\lambda}},\,\boldsymbol{\Theta}^{\boldsymbol{\lambda}},\,\boldsymbol{u}(\,\cdot\,)),\,\boldsymbol{\mu},\,\cdot\,)\right\|\right] + \\ \mathbb{E}_{\mathcal{N}(\mathcal{I})}\left[\left\|\widehat{\nabla F_r}(\boldsymbol{\mu};\,\boldsymbol{\Phi},\,\boldsymbol{\Psi},\,\boldsymbol{\Phi}^{\boldsymbol{\lambda}},\,\boldsymbol{\Theta}^{\boldsymbol{\lambda}})\right\|\right].\end{aligned} \tag{6.38}$$

where $\boldsymbol{u}(\boldsymbol{y}) = \boldsymbol{\Phi}\boldsymbol{u}_r(\boldsymbol{\mu};\,\boldsymbol{y},\,\boldsymbol{\Phi},\,\boldsymbol{\Psi})$ is the reconstructed primal solution for realization $\boldsymbol{y} \in \boldsymbol{\Xi}$.

At this point, the proposed two-level approximation of risk-averse measures of quantities of interest based on anisotropic sparse grids and model reduction has been introduced and relevant details pertaining to gradients and computable error indicators have been discussed. The next section uses this technology as the approximation model in the multifidelity trust region method of Chapter 3 to yield an efficient algorithm to solve stochastic PDE-constrained optimization problems. To simplify the exposition in the next section, details pertaining to the use of the minimum-residual sensitivity/adjoint reduced-order models to approximate $\nabla F_r(\boldsymbol{\mu})$ with $\widehat{\nabla F_r}(\boldsymbol{\mu})$ will be dropped. These details follow in a straightforward manner from those corresponding to the exact sensitivity/adjoint method to compute $\nabla F_r(\boldsymbol{\mu})$. Furthermore, the numerical experiments in Section 6.4 will solely consider a reduced-order models based on a Galerkin projection $\boldsymbol{\Psi} = \boldsymbol{\Phi}$, which implies the test basis is constant and $\widehat{\nabla F_r}(\boldsymbol{\mu}) = \nabla F_r(\boldsymbol{\mu})$, provided the sensitivity and adjoint bases are chosen according to (4.35), (4.63). Therefore, the distinction between the exact and minimum-residual sensitivity/adjoint methods is irrelevant since they are identical in this case.

6.3 Multifidelity Trust Region Method Based on Two-Level Approximation

This section presents the primary contribution of this chapter: the use of sparse grids and model reduction in the multifidelity trust region framework of Chapter 3 to yield an efficient algorithm for stochastic PDE-constrained optimization. The approximation model, $m_k(\boldsymbol{\mu})$, that is central to the trust region theory will be taken as the two-level approximation of risk-averse measures of quantities of interest introduced in the previous section. The error indicators required for the trust region theory are inspired from the error indicators in (6.35), (6.36)-(6.37). A two-level greedy algorithm will be introduced in Section 6.3.2 that combines dimension-adaptive sparse grid construction [67] with a classical reduced basis greedy method [149, 173]. The purpose of the greedy algorithm is to simultaneously construct a sparse grid $\mathcal{I}_k$ and reduced-order model $\boldsymbol{\Phi}_k$, $\boldsymbol{\Psi}_k$ such that the two-level approximation is sufficiently accurate to guarantee convergence based on the requirements (3.14), (3.15), (3.22) detailed in Chapter 3.

Before proceeding to the exposition of the multifidelity trust region method, additional notation will be introduced for convenience. In particular, each component of the error indicators in (6.35) and (6.36)-(6.37) are separated into individual terms. Define the following primal error terms from (6.35)

$$\begin{aligned}\mathcal{E}_1(\boldsymbol{\Phi},\,\boldsymbol{\Psi},\,\mathcal{I},\,\boldsymbol{\mu}) &\coloneqq \mathbb{E}_{\mathcal{I}\cup\mathcal{N}(\mathcal{I})}\left[\|\boldsymbol{r}(\boldsymbol{\Phi}\boldsymbol{u}_r(\boldsymbol{\mu};\,\cdot\,,\,\boldsymbol{\Phi},\,\boldsymbol{\Psi}),\,\boldsymbol{\mu},\,\cdot)\|\right]\\ \mathcal{E}_2(\boldsymbol{\Phi},\,\boldsymbol{\Psi},\,\mathcal{I},\,\boldsymbol{\mu}) &\coloneqq \mathbb{E}_{\mathcal{N}(\mathcal{I})}\left[|f(\boldsymbol{\Phi}\boldsymbol{u}_r(\boldsymbol{\mu};\,\cdot\,,\,\boldsymbol{\Phi},\,\boldsymbol{\Psi}),\,\boldsymbol{\mu},\,\cdot)|\right],\end{aligned} \tag{6.39}$$

where $\mathcal{E}_1$ is the (integrated) reduced-order model error indicator and $\mathcal{E}_2$ is the truncation error indicator associated with using the sparse grid $\mathcal{I}$ in place of the true expectation. The gradient error terms depend on whether the sensitivity or adjoint method are used in the gradient computation. If the sensitivity method is employed, define the error terms

$$\begin{aligned}\mathcal{E}_3(\boldsymbol{\Phi},\,\boldsymbol{\Psi},\,\mathcal{I},\,\boldsymbol{\mu}) &\coloneqq \mathbb{E}_{\mathcal{I}\cup\mathcal{N}(\mathcal{I})}\left[\left\|\boldsymbol{r}^{\partial}(\boldsymbol{\Phi}\boldsymbol{u}_r(\boldsymbol{\mu};\,\cdot\,,\,\boldsymbol{\Phi},\,\boldsymbol{\Psi}),\,\boldsymbol{\Phi}\frac{\partial \boldsymbol{u}_r}{\partial \boldsymbol{\mu}}(\boldsymbol{\mu};\,\cdot\,,\,\boldsymbol{\Phi},\,\boldsymbol{\Psi}),\,\boldsymbol{\mu},\,\cdot)\right\|\right]\\ \mathcal{E}_4(\boldsymbol{\Phi},\,\boldsymbol{\Psi},\,\mathcal{I},\,\boldsymbol{\mu}) &\coloneqq \mathbb{E}_{\mathcal{N}(\mathcal{I})}\left[\left\|\boldsymbol{g}^{\partial}(\boldsymbol{\Phi}\boldsymbol{u}_r(\boldsymbol{\mu};\,\cdot\,,\,\boldsymbol{\Phi},\,\boldsymbol{\Psi}),\,\boldsymbol{\Phi}\frac{\partial \boldsymbol{u}_r}{\partial \boldsymbol{\mu}}(\boldsymbol{\mu};\,\cdot\,,\,\boldsymbol{\Phi},\,\boldsymbol{\Psi}),\,\boldsymbol{\mu},\,\cdot)\right\|\right].\end{aligned} \tag{6.40}$$

Otherwise, the adjoint method is used and the error terms are defined as

$$\begin{aligned}\mathcal{E}_3(\boldsymbol{\Phi},\,\boldsymbol{\Psi},\,\mathcal{I},\,\boldsymbol{\mu}) &\coloneqq \mathbb{E}_{\mathcal{I}\cup\mathcal{N}(\mathcal{I})}\left[\left\|\boldsymbol{r}^{\boldsymbol{\lambda}}(\boldsymbol{\Phi}\boldsymbol{u}_r(\boldsymbol{\mu};\,\cdot\,,\,\boldsymbol{\Phi},\,\boldsymbol{\Psi}),\,\boldsymbol{\Psi}\boldsymbol{\lambda}_r(\boldsymbol{\mu};\,\cdot\,,\,\boldsymbol{\Phi},\,\boldsymbol{\Psi}),\,\boldsymbol{\mu},\,\cdot)\right\|\right]\\ \mathcal{E}_4(\boldsymbol{\Phi},\,\boldsymbol{\Psi},\,\mathcal{I},\,\boldsymbol{\mu}) &\coloneqq \mathbb{E}_{\mathcal{N}(\mathcal{I})}\left[\left\|\boldsymbol{g}^{\boldsymbol{\lambda}}(\boldsymbol{\Phi}\boldsymbol{u}_r(\boldsymbol{\mu};\,\cdot\,,\,\boldsymbol{\Phi},\,\boldsymbol{\Psi}),\,\boldsymbol{\Psi}\boldsymbol{\lambda}_r(\boldsymbol{\mu};\,\cdot\,,\,\boldsymbol{\Phi},\,\boldsymbol{\Psi}),\,\boldsymbol{\mu},\,\cdot)\right\|\right].\end{aligned} \tag{6.41}$$

Regardless of whether the sensitivity or adjoint method is used, $\mathcal{E}_3$ is the (integrated) sensitivity/adjoint reduced-order model error indicator and $\mathcal{E}_4$ is the truncation error indicator associated with using the sparse grid $\mathcal{I}$ in place of the true expectation in the gradient computation. With the

above definitions of the individual error terms, the error indicators in (6.35), (6.36)-(6.37) become

$$\begin{aligned} |F(\boldsymbol{\mu}) - F_r(\boldsymbol{\mu};\, \boldsymbol{\Phi},\, \boldsymbol{\Psi})| &\leq \zeta \mathcal{E}_1(\boldsymbol{\Phi},\, \boldsymbol{\Psi},\, \mathcal{I},\, \boldsymbol{\mu}) + \mathcal{E}_2(\boldsymbol{\Phi},\, \boldsymbol{\Psi},\, \mathcal{I},\, \boldsymbol{\mu}) \\ \|\nabla F(\boldsymbol{\mu}) - \nabla F_r(\boldsymbol{\mu};\, \boldsymbol{\Phi},\, \boldsymbol{\Psi})\| &\leq \kappa \mathcal{E}_1(\boldsymbol{\Phi},\, \boldsymbol{\Psi},\, \mathcal{I},\, \boldsymbol{\mu}) + \tau \mathcal{E}_3(\boldsymbol{\Phi},\, \boldsymbol{\Psi},\, \mathcal{I},\, \boldsymbol{\mu}) + \mathcal{E}_4(\boldsymbol{\Phi},\, \boldsymbol{\Psi},\, \mathcal{I},\, \boldsymbol{\mu}), \end{aligned} \tag{6.42}$$

where ζ, κ, $\tau > 0$ are arbitrary constants.

6.3.1 Trust Region Ingredients

This section will detail the various ingredients required to leverage the two-level approximation of risk-averse measures of quantities of interest in the multifidelity trust region framework of Chapter 3. In particular, the approximation model $m_k(\boldsymbol{\mu})$, objective decrease error indicator $\vartheta_k(\boldsymbol{\mu})$, gradient error indicator $\vartheta_k(\boldsymbol{\mu})$, and inexact objective model $\psi_k(\boldsymbol{\mu})$ and associated error indicator $\theta_k(\boldsymbol{\mu})$ will be specified using the developments of Section 6.2. For the remainder of this section, it is assumed that, at iteration k, the sparse grid $\mathcal{I}_k$ and reduced-order model $\boldsymbol{\Phi}_k$, $\boldsymbol{\Psi}_k$ have been constructed. Details pertaining to their construction will be provided in the next section.

At the kth iteration, the approximation model is taken as the two-level approximation of the risk-averse measure of the PDE quantity of interest, i.e.,

$$m_k(\boldsymbol{\mu}) \coloneqq F_r(\boldsymbol{\mu};\, \boldsymbol{\Phi}_k,\, \boldsymbol{\Psi}_k,\, \mathcal{I}_k) = \mathbb{E}_{\mathcal{I}_k}\left[f(\boldsymbol{\Phi}_k \boldsymbol{u}_r(\boldsymbol{\mu},\, \cdot\,,\, \boldsymbol{\Phi}_k,\, \boldsymbol{\Psi}_k),\, \boldsymbol{\mu},\, \cdot\,)\right]. \tag{6.43}$$

Similar to the trust region method detailed in Chapter 5, there are two options for the objective decrease error indicators: (1) the two-level residual-based indicator introduced in Section 6.2 and (2) the classical trust region constraint. The residual-based error indicator requires the pointwise form of the objective condition (3.14) to leverage the error terms $\mathcal{E}_1$ and $\mathcal{E}_2$

$$\begin{aligned} |F(\boldsymbol{\mu}_k) - F(\boldsymbol{\mu}) + m_k(\boldsymbol{\mu}) - m_k(\boldsymbol{\mu}_k)| &\leq |F(\boldsymbol{\mu}_k) - m_k(\boldsymbol{\mu}_k)| + |F(\boldsymbol{\mu}) - m_k(\boldsymbol{\mu})| \\ &\lesssim \zeta(\mathcal{E}_1(\boldsymbol{\Phi}_k,\, \boldsymbol{\Psi}_k,\, \mathcal{I}_k,\, \boldsymbol{\mu}_k) + \mathcal{E}_1(\boldsymbol{\Phi}_k,\, \boldsymbol{\Psi}_k,\, \mathcal{I}_k,\, \boldsymbol{\mu})) + \\ &\qquad \mathcal{E}_2(\boldsymbol{\Phi}_k,\, \boldsymbol{\Psi}_k,\, \mathcal{I}_k,\, \boldsymbol{\mu}_k) + \mathcal{E}_2(\boldsymbol{\Phi}_k,\, \boldsymbol{\Psi}_k,\, \mathcal{I}_k,\, \boldsymbol{\mu}). \end{aligned} \tag{6.44}$$

for an arbitrary constant $\zeta > 0$. Inspired from the above error indicator, the residual-based trust region constraint is defined as

$$\begin{aligned} \vartheta_k(\boldsymbol{\mu}) = &\alpha_1 \left(\mathcal{E}_1(\boldsymbol{\Phi}_k,\, \boldsymbol{\Psi}_k,\, \mathcal{I}_k,\, \boldsymbol{\mu}_k) + \mathcal{E}_1(\boldsymbol{\Phi}_k,\, \boldsymbol{\Psi}_k,\, \mathcal{I}_k,\, \boldsymbol{\mu})\right) + \\ &\alpha_2 \left(\mathcal{E}_2(\boldsymbol{\Phi}_k,\, \boldsymbol{\Psi}_k,\, \mathcal{I}_k,\, \boldsymbol{\mu}_k) + \mathcal{E}_2(\boldsymbol{\Phi}_k,\, \boldsymbol{\Psi}_k,\, \mathcal{I}_k,\, \boldsymbol{\mu})\right) \end{aligned} \tag{6.45}$$

for user-defined parameters α_1, $\alpha_2 > 0$ that balance the contribution of the reduced-order model error and truncation error. However, unlike the approach taken in Chapter 5, the classical trust region is primarily used in this section

$$\vartheta_k(\boldsymbol{\mu}) = \|\boldsymbol{\mu} - \boldsymbol{\mu}_k\|. \tag{6.46}$$

This choice is primarily due to the fact that the objective error bound required for global convergence of the trust region method (3.14)

$$|F(\boldsymbol{\mu}_k) - F(\boldsymbol{\mu}) + m_k(\boldsymbol{\mu}) - m_k(\boldsymbol{\mu}_k)| \leq \zeta \vartheta_k(\boldsymbol{\mu})$$

for some constant $\zeta > 0$, cannot be guaranteed with the residual-based choice of $\vartheta_k(\boldsymbol{\mu})$ in (6.35) due to *approximate* bound that results from approximating the truncation error using only the collocation nodes corresponding to the neighbors of the sparse grid. From the discussion in Chapter 3, the classical choice of $\vartheta_k(\boldsymbol{\mu})$ in (6.46) guarantees the above error bound if the gradient conditions (3.13),(3.15) are satisfied. Another reason for the choice of the classical trust region is that (6.45) can significantly increase the cost of an iteration of the trust region subproblem since $\vartheta_k(\boldsymbol{\mu})$ requires an expectation computation over $\mathcal{I} \cup \mathcal{N}(\mathcal{I})$ at two points $\boldsymbol{\mu}_k$ and $\boldsymbol{\mu}$, which may have substantially more nodes that $\mathcal{I}$ alone (used for the evaluation of $m_k(\boldsymbol{\mu})$). Finally, the definition of $\vartheta_k(\boldsymbol{\mu})$ is not differentiable for all $\boldsymbol{\mu} \in \mathbb{R}^{N_\mu}$ due to the presence of the norm in $\mathcal{E}_1$ and absolute value in $\mathcal{E}_2$, which may cause convergence issues in the interior-point trust region subproblem solver discussed in Section 3.1.2.

From the choice of $m_k(\boldsymbol{\mu})$, the gradient is $\nabla m_k(\boldsymbol{\mu}) = \nabla F_r(\boldsymbol{\mu};\, \boldsymbol{\Phi}_k,\, \boldsymbol{\Psi}_k,\, \mathcal{I}_k)$, which suggests the following gradient error bound based on the approximate bound in (6.42)

$$\varphi_k(\boldsymbol{\mu}) = \beta_1 \mathcal{E}_1(\boldsymbol{\Phi}_k,\, \boldsymbol{\Psi}_k,\, \mathcal{I}_k,\, \boldsymbol{\mu}) + \beta_2 \mathcal{E}_3(\boldsymbol{\Phi}_k,\, \boldsymbol{\Psi}_k,\, \mathcal{I}_k,\, \boldsymbol{\mu}) + \beta_3 \mathcal{E}_4(\boldsymbol{\Phi}_k,\, \boldsymbol{\Psi}_k,\, \mathcal{I}_k,\, \boldsymbol{\mu}) \tag{6.47}$$

This choice of $\varphi_k(\boldsymbol{\mu})$ does not guarantee the bound required by the global convergence theory in Chapter 3, i.e.,

$$\|\nabla F(\boldsymbol{\mu}_k) - \nabla m_k(\boldsymbol{\mu}_k)\| \leq \xi \varphi_k(\boldsymbol{\mu}_k), \tag{6.48}$$

for a constant $\xi > 0$, due to the approximation of the truncation error on the neighbors of the sparse grid. Therefore global convergence is not strictly guaranteed; however, the numerical results in Sections 6.4 suggest this choice does lead to global convergence for these problems.

With these choices of $\vartheta_k(\boldsymbol{\mu})$ and $\varphi_k(\boldsymbol{\mu})$, the sparse grid $\mathcal{I}_k$ and reduced-order model $\boldsymbol{\Phi}_k$, $\boldsymbol{\Psi}_k$ must be constructed to satisfy the error conditions in (3.14), (3.15), i.e.,

$$\begin{aligned} \vartheta_k(\boldsymbol{\mu}_k) &\leq \kappa_\vartheta \Delta_k \\ \varphi_k(\boldsymbol{\mu}_k) &\leq \kappa_\varphi \min\{\nabla m_k(\boldsymbol{\mu}_k),\, \Delta_k\}. \end{aligned}$$

The construction of these quantities such that the above error bounds are satisfied is somewhat delicate since the error terms $\mathcal{E}_1$ and $\mathcal{E}_3$ behave differently than $\mathcal{E}_2$ and $\mathcal{E}_4$ when $\boldsymbol{\Phi}_k$, $\boldsymbol{\Psi}_k$ and $\mathcal{I}_k$ are refined. For a fixed basis $\boldsymbol{\Phi}_k$, refinement of the sparse grid $\mathcal{I}_k$ decreases the truncation error terms $\mathcal{E}_2$ and $\mathcal{E}_4$. However, refinement of $\mathcal{I}_k$ may cause the model reduction error terms $\mathcal{E}_1$ and $\mathcal{E}_3$ to increase since the pointwise error is integrated over an expanded set of collocation nodes. A dimension-adaptive greedy algorithm that accounts for this interplay between the various error terms in response to refinement of the sparse grid and reduced-order basis will be introduced in the

next section.

At this point, the basic version of the multifidelity trust region method in Algorithm 1 is fully defined using the proposed two-level approximation and corresponding error indicators. An immediate issue with using these definitions in Algorithm 1 of Chapter 3 pertains to the evaluation of $F(\boldsymbol{\mu})$ in the computation of ρ_k in (3.9). From its definition, evaluation of $F(\boldsymbol{\mu})$ requires an infinite sum to evaluate the true expectation. The true expectation can be approximated on a "fine" quadrature rule (possibly based on a refined sparse grid) to evaluate $F(\boldsymbol{\mu})$ to high precision. While this option is simple and effective, it requires a large number of collocation nodes and the computation will constitute a bottleneck in the trust region algorithm since it must be performed at each major iteration. Instead, we opt to use the flexibility afforded by the trust region method in Chapter 3 for inexact objective evaluations in the computation of the actual-to-predicted ratio. This follows the work in [109] that uses dimension-adaptive sparse grids (without reduced-order models) for inexact objective evaluations. For this purpose, a separate sparse grid $\mathcal{I}'_k$ and reduced-order model $\boldsymbol{\Phi}'_k$, $\boldsymbol{\Psi}'_k$ are introduced and, following the notation in Chapter 3, the inexact objective function, $\psi_k(\boldsymbol{\mu})$, is employed with corresponding error indicator $\theta_k(\boldsymbol{\mu})$ defined as

$$\begin{aligned}\psi_k(\boldsymbol{\mu}) &= F_r(\boldsymbol{\mu};\, \boldsymbol{\Phi}'_k,\, \boldsymbol{\Psi}'_k,\, \mathcal{I}'_k)\\ \theta_k(\boldsymbol{\mu}) &= \alpha_1(\mathcal{E}_1(\boldsymbol{\Phi}'_k,\, \boldsymbol{\Psi}'_k,\, \mathcal{I}'_k,\, \boldsymbol{\mu}_k) + \mathcal{E}_1(\boldsymbol{\Phi}'_k,\, \boldsymbol{\Psi}'_k,\, \mathcal{I}'_k,\, \boldsymbol{\mu}))+\\ &\quad\ \alpha_2(\mathcal{E}_2(\boldsymbol{\Phi}'_k,\, \boldsymbol{\Psi}'_k,\, \mathcal{I}'_k,\, \boldsymbol{\mu}_k) + \mathcal{E}_2(\boldsymbol{\Phi}'_k,\, \boldsymbol{\Psi}'_k,\, \mathcal{I}'_k,\, \boldsymbol{\mu})).\end{aligned} \tag{6.49}$$

These choices are identical to $m_k(\boldsymbol{\mu})$ and the residual-based definition of $\vartheta_k(\boldsymbol{\mu})$, based on a (possibly refined) sparse grid $\mathcal{I}'_k$ and reduced-order model $\boldsymbol{\Phi}'_k$, $\boldsymbol{\Psi}'_k$. They do not necessarily guarantee the bound required for global convergence (3.21), again due to the approximation of the truncation errors on the neighbors of the sparse grid in (6.35). With these definitions, the actual-to-predicted ratio is computed as

$$\rho_k = \frac{\psi_k(\boldsymbol{\mu}_k) - \psi_k(\hat{\boldsymbol{\mu}}_k)}{m_k(\boldsymbol{\mu}_k) - m_k(\hat{\boldsymbol{\mu}}_k)} \tag{6.50}$$

where $\hat{\boldsymbol{\mu}}_k$ is the solution of the trust region subproblem, i.e.,

$$\hat{\boldsymbol{\mu}}_k \;=\; \underset{\boldsymbol{\mu}\in\mathbb{R}^{N_{\boldsymbol{\mu}}}}{\arg\min}\;\; m_k(\boldsymbol{\mu}) \quad \text{subject to} \quad \vartheta_k(\boldsymbol{\mu}) \leq \Delta_k. \tag{6.51}$$

The sparse grid $\mathcal{I}'_k$ and reduced-order model $\boldsymbol{\Phi}'_k$, $\boldsymbol{\Psi}'_k$ are constructed to guarantee

$$\theta_k^{\omega} \leq \eta \min\{m_k(\boldsymbol{\mu}_k) - m_k(\hat{\boldsymbol{\mu}}_k),\, r_k\}, \tag{6.52}$$

where $\omega \in (0,\, 1)$, $\eta < \min\{\eta_1,\, 1-\eta_2\}$, and $\{r_k\}_{k=1}^{\infty}$ is a sequence such that $r_k \to 0$, using the two-level dimension-adaptive greedy algorithm to be introduced in the next section. Once this training algorithm is completely specified, the complete trust region algorithm will be fully prescribed and is summarized in Section 6.3.3 and Algorithm 15.

6.3.2 Greedy Construction of Sparse Grid and Reduced Basis

The quality of the two-level approximation of risk-averse measures of PDE quantities of interest introduced in the previous section depends critically on the sparse grid $\mathcal{I}$ and reduced-order basis $\boldsymbol{\Phi}$ used in (6.43). This section develops dimension-adaptive greedy methods for the simultaneous construction of the sparse grid $\mathcal{I}_k$ ($\mathcal{I}'_k$) and reduced-order basis $\boldsymbol{\Phi}_k$ ($\boldsymbol{\Phi}'_k$) that targets each term in the error indicators $\varphi_k(\boldsymbol{\mu})$ and $\theta_k(\boldsymbol{\mu})$ such that the error conditions (required for global convergence) in (3.14), (3.15), (3.22) are satisfied. Since the classical trust region constraint is used to define $\vartheta_k(\boldsymbol{\mu})$, the objective decrease condition (3.14) will be automatically satisfied if the gradient bound (3.13) and condition (3.15) are satisfied (Chapter 3). Thus, the task reduces to construction of $\mathcal{I}_k$, $\boldsymbol{\Phi}_k$ such that the gradient condition (3.15) is satisfied and $\mathcal{I}'_k$, $\boldsymbol{\Phi}'_k$ such that the inexact objective condition (3.22) is satisfied. We begin with the gradient condition.

Recall from (6.47), the gradient error indicator is a weighted sum of three terms: the primal error $\mathcal{E}_1$, the sensitivity/adjoint error $\mathcal{E}_3$, and the gradient truncation error $\mathcal{E}_4$

$$\varphi_k(\boldsymbol{\mu}) = \beta_1 \mathcal{E}_1(\boldsymbol{\Phi}_k, \boldsymbol{\Psi}_k, \mathcal{I}_k, \boldsymbol{\mu}) + \beta_2 \mathcal{E}_3(\boldsymbol{\Phi}_k, \boldsymbol{\Psi}_k, \mathcal{I}_k, \boldsymbol{\mu}) + \beta_3 \mathcal{E}_4(\boldsymbol{\Phi}_k, \boldsymbol{\Psi}_k, \mathcal{I}_k, \boldsymbol{\mu}).$$

From Chapter 3, global convergence of the multifidelity trust region method is predicated on the satisfaction of the gradient condition: $\varphi_k(\boldsymbol{\mu}_k) \leq \kappa_\varphi \min\{\|\nabla m_k(\boldsymbol{\mu}_k)\|, \Delta_k\}$. A sufficient condition for this gradient condition to hold is that each term satisfies an appropriate *fraction* of the condition, i.e.,

$$\begin{aligned}
\mathcal{E}_1(\boldsymbol{\Phi}_k, \boldsymbol{\Psi}_k, \mathcal{I}_k, \boldsymbol{\mu}_k) &\leq \frac{\kappa_\varphi}{3\beta_1} \min\{\|\nabla m_k(\boldsymbol{\mu}_k)\|, \Delta_k\} \\
\mathcal{E}_3(\boldsymbol{\Phi}_k, \boldsymbol{\Psi}_k, \mathcal{I}_k, \boldsymbol{\mu}_k) &\leq \frac{\kappa_\varphi}{3\beta_2} \min\{\|\nabla m_k(\boldsymbol{\mu}_k)\|, \Delta_k\} \\
\mathcal{E}_4(\boldsymbol{\Phi}_k, \boldsymbol{\Psi}_k, \mathcal{I}_k, \boldsymbol{\mu}_k) &\leq \frac{\kappa_\varphi}{3\beta_3} \min\{\|\nabla m_k(\boldsymbol{\mu}_k)\|, \Delta_k\}.
\end{aligned} \tag{6.53}$$

The purpose of the positive weights β_1, β_2, β_3, introduced in the previous section, is to balance or scale the individual contributions of the error terms such the uniform split above is justified. This decomposition has reduced the monolithic task of constructing a sparse grid and reduced-order model such that the gradient condition in (3.15) holds to the individual tasks in (6.53). While the interplay between the three error terms in (6.53) and refinement of the reduced-order model and sparse grid is highly coupled and fairly complex, the following observations suggest an effective training strategy: (1) for a fixed sparse grid, $\mathcal{E}_1$ and $\mathcal{E}_3$ decrease (possibly non-monotonically) as the reduced-order model is hierarchically refined and (2) for a fixed reduced-order model, $\mathcal{E}_4$ decreases (possibly non-monotonically) as the sparse grid is refined. Therefore, the construction of the reduced-order model, for a fixed sparse grid, will proceed according to a variant of the classical greedy method [149, 173], to target the error terms $\mathcal{E}_1$ and $\mathcal{E}_3$. For a fixed reduced-order model, the sparse grid will be adapted using the anisotropic dimension-adaptive approach [67] to target $\mathcal{E}_4$. These steps will be performed iteratively until the conditions in (6.53) are met. Before discussing the combined algorithm in detail, the individual components, namely dimension-adaptive construction of a sparse grid and greedy construction of a reduced-order model, are introduced.

The construction of $\mathcal{I}$ will mimic the dimension-adaptive algorithm introduced in the seminal work by Gerstner and Griebel [67] for constructing a goal-oriented, anisotropic sparse grid. In this approach, the truncation error associated with the sparse grid $\mathcal{I}$ is approximated solely on the neighbors $\mathcal{N}(\mathcal{I})$, exactly as discussed in Section 6.1.3. If this truncation error approximation is larger than a specified tolerance, the multi-index in the set of neighbors that contributes most to the error is added to the index set, that is, $\mathcal{I} \leftarrow \mathcal{I} \cup \{\boldsymbol{i}^*\}$ where

$$\boldsymbol{i}^* = \underset{\boldsymbol{i} \in \mathcal{N}(\mathcal{I})}{\arg\min} \quad \left|\Delta^{\boldsymbol{i}}[g]\right|. \tag{6.54}$$

for the integrand $g : \mathbb{R}^{N_{\boldsymbol{y}}} \to \mathbb{R}$. In the context of the proposed two-level approximation, the dimension-adaptive algorithm is applied to the integrand

$$\|\nabla F_r(\boldsymbol{\mu}; \cdot, \boldsymbol{\Phi}, \boldsymbol{\Psi})\|$$

for a fixed reduced-order model $\boldsymbol{\Phi}$, $\boldsymbol{\Psi}$ and given $\boldsymbol{\mu}$. With this integrand, the dimension-adaptive algorithm decreases the error terms $\mathcal{E}_4(\boldsymbol{\Phi}, \boldsymbol{\Psi}, \cdot, \boldsymbol{\mu})$. While the convergence is not necessarily monotonic, this term approaches zero in the limiting case as $\mathcal{I} \to \mathbb{N}_+^{N_{\boldsymbol{y}}}$. In fact, since $\mathcal{E}_4(\boldsymbol{\Phi}, \boldsymbol{\Psi}, \mathcal{I}, \boldsymbol{\mu})$ is exactly the truncation error approximation, it is used for the convergence criteria in the algorithm.

The construction of the reduced-order basis follows the well-studied greedy algorithm [149, 173]. The original greedy algorithm improves the parametric robustness (usually over $\boldsymbol{\mu}$-space) of a reduced-order basis $\boldsymbol{\Phi}$ by adding snapshots of the high-dimensional model at the point where the reduced-order model error is largest. Regions of high error are found by evaluating the reduced-order model and an inexpensive error indicator at a (possibly large) set of candidate points (in the space where the ROM is being trained) and performs a direct search for maximum value of the error indicator over the candidate set. A weighted variant of the greedy algorithm was developed [42] for stochastic problems with non-uniform probability distributions to train a reduced-order model over the stochastic space $\boldsymbol{\Xi}$. This weighted greedy method uses the probability density $\rho(\boldsymbol{y})$ to weight the error indicator at a particular realization $\boldsymbol{y} \in \boldsymbol{\Xi}$ since regions with significant mass will amplify errors during the expectation computation. In the same work, the weighted greedy algorithm was coupled with sparse grids by using the sparse grid nodes as the candidate set; since the reduced-order model is only queried on the nodes of the sparse grid, it only needs to be trained at these points. In the present work, a similar weighted greedy algorithm is applied to train the reduced-order model over the stochastic collocation nodes (and neighbors) $\boldsymbol{\Xi}_{\mathcal{I} \cup \mathcal{N}(\mathcal{I})}$ for a fixed $\boldsymbol{\mu}$ and sparse grid $\mathcal{I}$. Since the gradient condition is only required to hold at the trust region center, the training is performed solely in stochastic space (with $\boldsymbol{\Xi}_{\mathcal{I} \cup \mathcal{N}(\mathcal{I})}$ as the candidate set) for $\boldsymbol{\mu} = \boldsymbol{\mu}_k$ fixed. Unlike the traditional greedy methods, the proposed method builds a reduced-order model that accurately represents primal *and* sensitivity or adjoint states over the training space. This is required since the greedy algorithm will be responsible for reducing the primal $\mathcal{E}_1$ and sensitivity/adjoint $\mathcal{E}_3$ error terms as both terms arise in the gradient error indicator $\varphi_k(\boldsymbol{\mu})$. This is achieved by adding sensitivity or adjoint snapshots to the reduced-order basis, in addition to the standard primal snapshots. From

the form of the gradient error indicator $\varphi_k(\boldsymbol{\mu})$ in (6.47), the primal error indicator is taken as the weighted primal residual norm, i.e., the integrand in $\mathcal{E}_1$,

$$\rho(\boldsymbol{y}) \left\| \boldsymbol{r}(\boldsymbol{\Phi}\boldsymbol{u}_r(\boldsymbol{\mu};\, \boldsymbol{y},\, \boldsymbol{\Phi},\, \boldsymbol{\Psi}), \boldsymbol{\mu},\, \boldsymbol{y}) \right\| \tag{6.55}$$

and the dual error indicator is taken as the weighted dual solution, i.e., the integrand in $\mathcal{E}_3$

$$\begin{aligned} &\rho(\boldsymbol{y}) \left\| \boldsymbol{r}^{\partial}(\boldsymbol{\Phi}\boldsymbol{u}_r(\boldsymbol{\mu};\, \boldsymbol{y},\, \boldsymbol{\Phi},\, \boldsymbol{\Psi}),\, \boldsymbol{\Phi}\frac{\partial \boldsymbol{u}_r}{\partial \boldsymbol{\mu}}(\boldsymbol{\mu};\, \boldsymbol{y},\, \boldsymbol{\Phi},\, \boldsymbol{\Psi}),\, \boldsymbol{\mu},\, \boldsymbol{y}) \right\| \\ &\rho(\boldsymbol{y}) \left\| \boldsymbol{r}^{\boldsymbol{\lambda}}(\boldsymbol{\Phi}\boldsymbol{u}_r(\boldsymbol{\mu};\, \boldsymbol{y},\, \boldsymbol{\Phi},\, \boldsymbol{\Psi}),\, \boldsymbol{\Psi}\boldsymbol{\lambda}_r(\boldsymbol{\mu};\, \boldsymbol{y},\, \boldsymbol{\Phi},\, \boldsymbol{\Psi}),\, \boldsymbol{\mu},\, \boldsymbol{y}) \right\|. \end{aligned} \tag{6.56}$$

Since the error terms $\mathcal{E}_1$ and $\mathcal{E}_3$ are integrated over $\mathcal{I} \cup \mathcal{N}(\mathcal{I})$, $\boldsymbol{\Xi}_{\mathcal{I}\cup\mathcal{N}(\mathcal{I})}$ is used as the candidate set. For a fixed point in parameter space $\boldsymbol{\mu}$ and sparse grid $\mathcal{I}$, the greedy algorithm builds up the reduced-order basis using primal and sensitivity/adjoint snapshots in the described manner, until $\mathcal{E}_1(\boldsymbol{\Phi},\, \boldsymbol{\Psi},\, \mathcal{I},\, \boldsymbol{\mu})$ and $\mathcal{E}_3(\boldsymbol{\Phi},\, \boldsymbol{\Psi},\, \mathcal{I},\, \boldsymbol{\mu})$ drop below user-defined tolerances. If a minimum-residual reduced-order model is employed, the algorithm is guaranteed to terminate due to the monotonicity property (Proposition 4.1). In the limiting case where snapshots have been added for each $\boldsymbol{y} \in \boldsymbol{\Xi}_{\mathcal{I}\cup\mathcal{N}(\mathcal{I})}$, the primal reduced-order model will be exact for each $\boldsymbol{y} \in \boldsymbol{\Xi}_{\mathcal{I}\cup\mathcal{N}(\mathcal{I})}$ and thus $\mathcal{E}_1$ is identically zero. If the reduced-order model is exact at these sparse grid nodes, the reduced sensitivity and adjoint method possess the minimum-residual property, which (Propositions 4.2 and 4.4) guarantees the reduced sensitivity/adjoint exactly reconstruct the corresponding high-dimensional quantity. Therefore $\mathcal{E}_3$ is identically zero. If a minimum-residual reduced-order model is employed, $\mathcal{E}_1$ (in the appropriate norm) will actually decrease monotonically since adding snapshots to the reduced-order basis can only improve the approximation quality (in terms of the residual norm in a particular metric). A similar argument cannot be made for $\mathcal{E}_3$, even if minimum-residual sensitivity/adjoint reduced-order models are used, since modification of $\boldsymbol{\Phi}$ alters the linearization point defining the sensitivity/adjoint residual and the objective function in successive minimum-residual optimization problems cannot be compared.

The final training algorithm combines the dimension-adaptive sparse grid construction with greedy sampling to build a reduced-order basis. For a fixed sparse grid $\mathcal{I}$, the primal-sensitivity/adjoint weighted greedy algorithm is used to build a reduced-order basis $\boldsymbol{\Phi}$ such that $\mathcal{E}_1(\boldsymbol{\Phi},\, \boldsymbol{\Psi},\, \mathcal{I},\, \boldsymbol{\mu})$ and $\mathcal{E}_3(\boldsymbol{\Phi},\, \boldsymbol{\Psi},\, \mathcal{I},\, \boldsymbol{\mu})$ satisfy (6.53). This reduced-order basis is fixed and a single step of the dimension-adaptive sparse grid is applied to updated $\mathcal{I}$ according to $\mathcal{I} \leftarrow \mathcal{I} \cup \{\boldsymbol{i}^*\}$ where $\boldsymbol{i}^*$ is defined in (6.54). Then the weighted greedy algorithm is applied with the new sparse grid. The algorithm proceeds in this manner until $\mathcal{E}_4(\boldsymbol{\Phi},\, \boldsymbol{\Psi},\, \mathcal{I},\, \boldsymbol{\mu})$ satisfies (6.53). Therefore the combined algorithm consists of an outer loop that refines the sparse grid (to reduced truncation error, $\mathcal{E}_4$) and an inner loop that builds an accurate reduced-order basis for a given sparse grid (to decrease the reduced-order model error, $\mathcal{E}_1$ and $\mathcal{E}_3$).

Algorithm 13 summarizes the combined dimension-adaptive greedy algorithm that proceeds according to the above two-level iteration to improve a given sparse grid and reduced-order basis such

that the gradient condition in (6.53) is satisfied. As stated, the algorithm implicitly requires initialization of each quantity. At iteration 0, the sparse grid is initialized as the uniform level-one sparse grid, $\mathcal{I} = \{(1, \ldots, 1)\}$, which consists of a single node $\boldsymbol{\Xi}_{\mathcal{I}} = \{\mathbf{0}\}$. The reduced-order basis is constructed from the primal and sensitivity/adjoint snapshot at this single sparse grid node at the first trust region center, i.e., $\boldsymbol{u}(\boldsymbol{\mu}_0, \mathbf{0})$ and $\frac{\partial \boldsymbol{u}}{\partial \boldsymbol{\mu}}(\boldsymbol{\mu}_0, \mathbf{0})$ or $\boldsymbol{\lambda}(\boldsymbol{\mu}_0, \mathbf{0})$. That is, the sparse grid $\mathcal{I}_0$ and reduced-order model $\boldsymbol{\Phi}_0$, $\boldsymbol{\Psi}_0$ are constructed as

$$\boldsymbol{\Phi}_0, \boldsymbol{\Psi}_0, \mathcal{I}_0 = \texttt{two-level-refine-grad}(\boldsymbol{\Phi}_{-1}, \boldsymbol{\Psi}_{-1}, \mathcal{I}_{-1}, \boldsymbol{\mu}_0) \tag{6.57}$$

where $\mathcal{I}_{-1} = \{(1, \ldots, 1)\}$, $\boldsymbol{\Phi}_{-1}$, $\boldsymbol{\Psi}_{-1}$ is the reduced-order model constructed with the aforementioned snapshots, and `two-level-refine-grad` is defined in Algorithm 13. For all subsequent iterations, the sparse grid and reduced-basis basis are initialized from the previous iteration, i.e., the construction of $\mathcal{I}_k$, $\boldsymbol{\Phi}_k$ is initialized with $\mathcal{I}_{k-1}$, $\boldsymbol{\Phi}_{k-1}$

$$\boldsymbol{\Phi}_k, \boldsymbol{\Psi}_k, \mathcal{I}_k = \texttt{two-level-refine-grad}(\boldsymbol{\Phi}_{k-1}, \boldsymbol{\Psi}_{k-1}, \mathcal{I}_{k-1}, \boldsymbol{\mu}_k). \tag{6.58}$$

Apart from being a natural way to initialize the dimension-adaptive greedy algorithm, it has the added benefit of only refining $\mathcal{I}_{k-1}$ and $\boldsymbol{\Phi}_{k-1}$ if the choice $\mathcal{I}_k = \mathcal{I}_{k-1}$, $\boldsymbol{\Phi}_k = \boldsymbol{\Phi}_{k-1}$ are not sufficient to guarantee convergence, i.e., the gradient condition in (6.53) does not hold.

This completes the discussion of the training algorithm to build $\mathcal{I}_k$ and $\boldsymbol{\Phi}_k$, $\boldsymbol{\Psi}_k$ to ensure the gradient condition holds and attention is shifted to construction of $\mathcal{I}'_k$, $\boldsymbol{\Phi}'_k$, $\boldsymbol{\Psi}'_k$ such that the inexact objective condition (3.22) holds in order to properly assess the trust region step without requiring queries to $F(\boldsymbol{\mu})$. The error indicator for the objective decrease is a weighted sum of two terms: the primal error $\mathcal{E}_1$ and QoI truncation error $\mathcal{E}_2$

$$\begin{aligned}\theta_k(\boldsymbol{\mu}) =& \alpha_1(\mathcal{E}_1(\boldsymbol{\Phi}'_k, \boldsymbol{\Psi}'_k, \mathcal{I}'_k, \boldsymbol{\mu}_k) + \mathcal{E}_1(\boldsymbol{\Phi}'_k, \boldsymbol{\Psi}'_k, \mathcal{I}'_k, \boldsymbol{\mu}))+ \\ & \alpha_2(\mathcal{E}_2(\boldsymbol{\Phi}'_k, \boldsymbol{\Psi}'_k, \mathcal{I}'_k, \boldsymbol{\mu}_k) + \mathcal{E}_2(\boldsymbol{\Phi}'_k, \boldsymbol{\Psi}'_k, \mathcal{I}'_k, \boldsymbol{\mu})),\end{aligned} \tag{6.59}$$

which involves error terms evaluated at $\boldsymbol{\mu}_k$ and $\hat{\boldsymbol{\mu}}_k$ since the pointwise version of the objective decrease bound is used. From Chapter 3, the error condition (3.22), i.e., $\theta_k(\hat{\boldsymbol{\mu}}_k)^\omega \leq \eta \min\{m_k(\boldsymbol{\mu}_k) - m_k(\hat{\boldsymbol{\mu}}_k), r_k\}$, is required to preserve global convergence of the trust region method when $\psi_k(\boldsymbol{\mu})$ is used in place of $F(\boldsymbol{\mu})$ in the computation of ρ_k. A sufficient condition for the objective condition to hold is that each term satisfies an appropriate fraction of the condition, i.e.,

$$\begin{aligned}\mathcal{E}_1(\boldsymbol{\Phi}'_k, \boldsymbol{\Psi}'_k, \mathcal{I}'_k, \boldsymbol{\mu}_k) + \mathcal{E}_1(\boldsymbol{\Phi}'_k, \boldsymbol{\Psi}'_k, \mathcal{I}'_k, \hat{\boldsymbol{\mu}}_k) &\leq \frac{1}{2\alpha_1}\left(\eta \min\{m_k(\boldsymbol{\mu}_k) - m_k(\hat{\boldsymbol{\mu}}_k), r_k\}\right)^{1/\omega} \\ \mathcal{E}_2(\boldsymbol{\Phi}'_k, \boldsymbol{\Psi}'_k, \mathcal{I}'_k, \boldsymbol{\mu}_k) + \mathcal{E}_2(\boldsymbol{\Phi}'_k, \boldsymbol{\Psi}'_k, \mathcal{I}'_k, \hat{\boldsymbol{\mu}}_k) &\leq \frac{1}{2\alpha_2}\left(\eta \min\{m_k(\boldsymbol{\mu}_k) - m_k(\hat{\boldsymbol{\mu}}_k), r_k\}\right)^{1/\omega}\end{aligned} \tag{6.60}$$

where the positive weights α_1, α_2 balance the contributions of $\mathcal{E}_1$ and $\mathcal{E}_2$ to justify this uniform split. Therefore the monolithic task of satisfying the objective condition has been broken into the modular tasks in (6.60). Similar to the approach taken to construct $\mathcal{I}_k$ and $\boldsymbol{\Phi}_k$, a weighted greedy algorithm

Algorithm 13 Refine reduced-order basis and sparse grid for gradient condition

$$\boldsymbol{\Phi},\, \boldsymbol{\Psi},\, \mathcal{I} = \texttt{two-level-refine-grad}(\boldsymbol{\Phi},\, \boldsymbol{\Psi},\, \mathcal{I},\, \boldsymbol{\mu},\, \delta)$$

1: **Initialization**: Given

$$\boldsymbol{\Phi},\, \boldsymbol{\Psi},\, \mathcal{I},\, \boldsymbol{\mu},\, \delta,\, \beta_1 > 0,\, \beta_2 > 0,\, \beta_3 > 0,\, \kappa_\varphi > 0$$

2: **while** $\mathcal{E}_4(\boldsymbol{\Phi},\, \boldsymbol{\Psi},\, \mathcal{I},\, \boldsymbol{\mu}) > \dfrac{\kappa_\varphi}{3\beta_3} \min\left\{\left\|\mathbb{E}_{\mathcal{I}}\left[\nabla F_r(\boldsymbol{\mu};\, \cdot\,,\, \boldsymbol{\Phi},\, \boldsymbol{\Psi})\right]\right\|,\, \delta\right\}$ **do**

3: **Refine index set**: Add index set with largest contribution to truncation error

$$\mathcal{I} \leftarrow \mathcal{I} \cup \{\boldsymbol{i}^*\} \qquad \text{where} \qquad \boldsymbol{i}^* = \underset{\boldsymbol{i} \in \mathcal{N}(\mathcal{I})}{\arg\max} \left|\Delta^{\boldsymbol{i}}\left[\left\|\nabla F_r(\boldsymbol{\mu};\, \cdot\,,\, \boldsymbol{\Phi},\, \boldsymbol{\Psi})\right\|\right]\right|$$

4: **while** $\mathcal{E}_1(\boldsymbol{\Phi},\, \boldsymbol{\Psi},\, \mathcal{I},\, \boldsymbol{\mu}) > \dfrac{\kappa_\varphi}{3\beta_1} \min\left\{\left\|\mathbb{E}_{\mathcal{I}}\left[\nabla F_r(\boldsymbol{\mu};\, \cdot\,,\, \boldsymbol{\Phi},\, \boldsymbol{\Psi})\right]\right\|,\, \delta\right\}$ **do**

5: **Evaluate primal error indicator**: Greedily select $\boldsymbol{y} \in \boldsymbol{\Xi}_{\boldsymbol{i}^*}$ with largest error

$$\boldsymbol{y}^* = \underset{\boldsymbol{y} \in \boldsymbol{\Xi}_{\boldsymbol{i}^*}}{\arg\max}\ \rho(\boldsymbol{y}) \left\|\boldsymbol{r}(\boldsymbol{\Phi}\boldsymbol{u}_r(\boldsymbol{\mu};\, \boldsymbol{y},\, \boldsymbol{\Phi},\, \boldsymbol{\Psi});\, \boldsymbol{\mu},\, \boldsymbol{y})\right\|$$

6: **Reduced-order model construction**: Update reduced basis with new snapshots

$$\boldsymbol{\Phi} = \begin{bmatrix} \boldsymbol{\Phi} & \boldsymbol{u}(\boldsymbol{\mu};\, \boldsymbol{y}^*) & \dfrac{\partial \boldsymbol{u}}{\partial \boldsymbol{\mu}}(\boldsymbol{\mu};\, \boldsymbol{y}^*) \end{bmatrix} \qquad \text{or} \qquad \begin{bmatrix} \boldsymbol{\Phi} & \boldsymbol{u}(\boldsymbol{\mu};\, \boldsymbol{y}^*) & \boldsymbol{\lambda}(\boldsymbol{\mu};\, \boldsymbol{y}^*) \end{bmatrix}$$

7: **end while**

8: **while** $\mathcal{E}_2(\boldsymbol{\Phi},\, \boldsymbol{\Psi},\, \mathcal{I},\, \boldsymbol{\mu}) > \dfrac{\kappa_\varphi}{3\beta_2} \min\left\{\left\|\mathbb{E}_{\mathcal{I}}\left[\nabla F_r(\boldsymbol{\mu};\, \cdot\,,\, \boldsymbol{\Phi},\, \boldsymbol{\Psi})\right]\right\|,\, \delta\right\}$ **do**

9: **Evaluate dual error indicator**: Greedily select $\boldsymbol{y} \in \boldsymbol{\Xi}_{\boldsymbol{i}^*}$ with largest error

$$\boldsymbol{y}^* = \underset{\boldsymbol{y} \in \boldsymbol{\Xi}_{\boldsymbol{i}^*}}{\arg\max}\ \rho(\boldsymbol{y}) \left\|\boldsymbol{r}^{\partial}\left(\boldsymbol{\Phi}\boldsymbol{u}_r(\boldsymbol{\mu};\, \boldsymbol{y},\, \boldsymbol{\Phi},\, \boldsymbol{\Psi});\, \boldsymbol{\Phi}\frac{\partial \boldsymbol{u}_r}{\partial \boldsymbol{\mu}}(\boldsymbol{\mu};\, \boldsymbol{y},\, \boldsymbol{\Phi},\, \boldsymbol{\Psi}),\, \boldsymbol{\mu},\, \boldsymbol{y}\right)\right\| \qquad \text{or}$$

$$= \underset{\boldsymbol{y} \in \boldsymbol{\Xi}_{\boldsymbol{i}^*}}{\arg\max}\ \rho(\boldsymbol{y}) \left\|\boldsymbol{r}^{\boldsymbol{\lambda}}\left(\boldsymbol{\Phi}\boldsymbol{u}_r(\boldsymbol{\mu};\, \boldsymbol{y},\, \boldsymbol{\Phi},\, \boldsymbol{\Psi});\, \boldsymbol{\Psi}\boldsymbol{\lambda}_r(\boldsymbol{\mu};\, \boldsymbol{y},\, \boldsymbol{\Phi},\, \boldsymbol{\Psi}),\, \boldsymbol{\mu},\, \boldsymbol{y}\right)\right\|$$

10: **Reduced-order model construction**: Update reduced basis with new snapshots

$$\boldsymbol{\Phi} = \begin{bmatrix} \boldsymbol{\Phi} & \boldsymbol{u}(\boldsymbol{\mu};\, \boldsymbol{y}^*) & \dfrac{\partial \boldsymbol{u}}{\partial \boldsymbol{\mu}}(\boldsymbol{\mu};\, \boldsymbol{y}^*) \end{bmatrix} \qquad \text{or} \qquad \begin{bmatrix} \boldsymbol{\Phi} & \boldsymbol{u}(\boldsymbol{\mu};\, \boldsymbol{y}^*) & \boldsymbol{\lambda}(\boldsymbol{\mu};\, \boldsymbol{y}^*) \end{bmatrix}$$

11: **end while**

12: **end while**

will be used to enforce the conditions on $\mathcal{E}_1$ and the dimension-adaptive sparse grid construction to satisfy the conditions on $\mathcal{E}_2$.

While the dimension-adaptive greedy algorithm to construct $\mathcal{I}'_k$, $\boldsymbol{\Phi}'_k$, $\boldsymbol{\Psi}'_k$ for the objective decrease condition will be very similar that used to construct $\mathcal{I}_k$, $\boldsymbol{\Phi}_k$, $\boldsymbol{\Psi}_k$, there will be two critical differences. First, the error terms in (6.60) involve *two* points in parameters space: the trust region center $\boldsymbol{\mu}_k$ and the candidate step $\hat{\boldsymbol{\mu}}_k$. In contrast, the gradient condition only imposed requirements at the trust region center. This has implications for both the dimension-adaptive sparse grid construction and greedy method. Second, the conditions in (6.60) only impose requirements on the primal reduced-order model accuracy and truncation error, whereas the gradient condition also placed requirements on the sensitivity/adjoint accuracy. This implies only *primal* snapshots are required during the greedy construction of the reduced-order model.

For a given $\mathcal{I}$, $\boldsymbol{\Phi}$, $\boldsymbol{\Psi}$, if the truncation error conditions in (6.60), i.e., requirements on $\mathcal{E}_2$, are not satisfied, the sparse grid is updated according to $\mathcal{I} \leftarrow \mathcal{I} \cup \{\boldsymbol{i}^*\}$, where

$$\boldsymbol{i}^* = \underset{\boldsymbol{i} \in \mathcal{N}(\mathcal{I})}{\arg\max} \left(\max \left\{ \left| \Delta^{\boldsymbol{i}} \left[F_r(\boldsymbol{\mu}_k;\, \boldsymbol{\Phi},\, \boldsymbol{\Psi}) \right] \right|, \left| \Delta^{\boldsymbol{i}} \left[F_r(\hat{\boldsymbol{\mu}}_k;\, \boldsymbol{\Phi},\, \boldsymbol{\Psi}) \right] \right| \right\} \right). \tag{6.61}$$

The integrands in each term is precisely the integrand of $\mathcal{E}_2$ at the two parameters of interest: $\boldsymbol{\mu}_k$ and $\hat{\boldsymbol{\mu}}_k$. Therefore this refinement process can be repeated iteratively until the conditions on $\mathcal{E}_2$ in (6.60) are satisfied. Following the combined dimension-adaptive greedy method introduced for the gradient condition, the sparse grid refinement steps are interwoven with greedy construction of the reduced-order model. For a fixed $\mathcal{I}$, $\boldsymbol{\Phi}$, $\boldsymbol{\Psi}$, define $\boldsymbol{\mu}^* \in \{\boldsymbol{\mu}_k, \hat{\boldsymbol{\mu}}_k\}$ and $\boldsymbol{y}^* \in \boldsymbol{\Xi}_{\mathcal{I}}$ as the quantities that maximize the weighted residual-based error indicator

$$\boldsymbol{\mu}^*, \boldsymbol{y}^* = \underset{\substack{\boldsymbol{\mu} \in \{\boldsymbol{\mu}_k, \hat{\boldsymbol{\mu}}_k\}, \\ \boldsymbol{y} \in \boldsymbol{\Xi}_{\mathcal{I}}}}{\arg\max} \rho(\boldsymbol{y}) \left\| \boldsymbol{r}(\boldsymbol{\Phi} \boldsymbol{u}_r(\boldsymbol{\mu};\, \boldsymbol{y},\, \boldsymbol{\Phi},\, \boldsymbol{\Psi}),\, \boldsymbol{\mu},\, \boldsymbol{y}) \right\|. \tag{6.62}$$

The reduced-order basis $\boldsymbol{\Phi}$ is updated according to $\boldsymbol{\Phi} \leftarrow \begin{bmatrix} \boldsymbol{\Phi} & \boldsymbol{u}(\boldsymbol{\mu}^*,\, \boldsymbol{y}^*) \end{bmatrix}$; an optional orthogonalization step is usually used to ensure the the reduced basis is full rank and the resulting reduced-order model is well-conditioned. As discussed, only primal snapshot are used since the conditions in (6.60) only places requirements on the primal accuracy of the reduced-order model. The argument of the maximization problem in (6.62) is exactly the integrand of $\mathcal{E}_1$. Assuming a minimum-residual reduced-order model is used, $\mathcal{E}_1(\boldsymbol{\Phi},\, \boldsymbol{\Psi},\, \mathcal{I},\, \boldsymbol{\mu}_k)$ and $\mathcal{E}_1(\boldsymbol{\Phi},\, \boldsymbol{\Psi},\, \mathcal{I},\, \hat{\boldsymbol{\mu}}_k)$ will monotonically decrease with each iteration of the greedy method and the iterations proceed until the conditions in (6.60) are satisfied. The combined training algorithm alternates between sparse grid and reduced basis construction exactly as that in Algorithm 14, namely, for a fixed sparse grid, the greedy method is applied to ensure the conditions on $\mathcal{E}_1$ in (6.60) hold, then the reduced-order model is fixed and the sparse grid is refined according to (6.62). The combined algorithm terminates when all conditions in (6.60) are satisfied.

Algorithm 14 summarizes the combined dimension-adaptive greedy algorithm that constructs a sparse grid and reduced-order model such that the objective decrease condition (6.60) holds. Similar

to Algorithm 13, this algorithm *refines* a given sparse grid and reduced basis and implicitly requires initialization of each quantity. At any iteration k, the sparse grid $\mathcal{I}_k$ and reduced-order model $\boldsymbol{\Phi}_k$, $\boldsymbol{\Psi}_k$ are used to initialize Algorithm 14, i.e.,

$$\boldsymbol{\Phi}_k', \boldsymbol{\Psi}_k', \mathcal{I}_k' = \texttt{two-level-refine-obj}(\boldsymbol{\Phi}_k, \boldsymbol{\Psi}_k, \mathcal{I}_k, \boldsymbol{\mu}_k, \hat{\boldsymbol{\mu}}_k, r_k) \tag{6.63}$$

since $\mathcal{I}_k$, $\boldsymbol{\Phi}_k$, $\boldsymbol{\Psi}_k$ have been constructed to satisfy the error condition in (3.15) at $\boldsymbol{\mu}_k$. If that requirements turns out to be more restrictive than that in (3.22), the algorithm will not modify the sparse grid or reduced-order basis, i.e., $\mathcal{I}_k' = \mathcal{I}_k$ and $\boldsymbol{\Phi}_k' = \boldsymbol{\Phi}_k$, and the actual-to-predicted ratio is unity and acceptance of the step is guaranteed.

Algorithm 14 Refine reduced-order basis and sparse grid for objective decrease condition

$$\boldsymbol{\Phi}, \boldsymbol{\Psi}, \mathcal{I} = \texttt{two-level-refine-obj}(\boldsymbol{\Phi}, \boldsymbol{\Psi}, \mathcal{I}, \boldsymbol{\mu}_1, \boldsymbol{\mu}_2, s)$$

1: **Initialization**: Given

$$\boldsymbol{\Phi}, \boldsymbol{\Psi}, \mathcal{I}, \boldsymbol{\mu}_1, \boldsymbol{\mu}_2, s > 0, \omega \in (0, 1)$$

2: **while**

$$\mathcal{E}_2(\boldsymbol{\Phi}, \boldsymbol{\Psi}, \mathcal{I}, \boldsymbol{\mu}_1) + \mathcal{E}_2(\boldsymbol{\Phi}, \boldsymbol{\Psi}, \mathcal{I}, \boldsymbol{\mu}_2) > \frac{1}{2\alpha_2} \left(\eta \min\left\{\mathbb{E}_{\mathcal{I}}\left[F_r(\boldsymbol{\mu}_1; \cdot, \boldsymbol{\Phi}, \boldsymbol{\Psi})\right] - \mathbb{E}_{\mathcal{I}}\left[F_r(\boldsymbol{\mu}_2; \cdot, \boldsymbol{\Phi}, \boldsymbol{\Psi})\right], s\right\}\right)^{1/\omega}$$

do

3: **Refine index set**: Add index set with largest contribution to truncation error

$$\mathcal{I} \leftarrow \mathcal{I} \cup \{\boldsymbol{i}^*\} \qquad \text{where} \qquad \boldsymbol{i}^* = \underset{\boldsymbol{i} \in \mathcal{N}(\mathcal{I})}{\arg\max} \left(\max\left\{\left|\Delta^{\boldsymbol{i}} F_r(\boldsymbol{\mu}_1; \cdot, \boldsymbol{\Phi}, \boldsymbol{\Psi})\right|, \left|\Delta^{\boldsymbol{i}} F_r(\boldsymbol{\mu}_2; \cdot, \boldsymbol{\Phi}, \boldsymbol{\Psi})\right|\right\}\right)$$

4: **while**

$$\mathcal{E}_1(\boldsymbol{\Phi}, \boldsymbol{\Psi}, \mathcal{I}, \boldsymbol{\mu}_1) + \mathcal{E}_1(\boldsymbol{\Phi}, \boldsymbol{\Psi}, \mathcal{I}, \boldsymbol{\mu}_2) > \frac{1}{2\alpha_1} \left(\eta \min\left\{\mathbb{E}_{\mathcal{I}}\left[F_r(\boldsymbol{\mu}_1; \cdot, \boldsymbol{\Phi}, \boldsymbol{\Psi})\right] - \mathbb{E}_{\mathcal{I}}\left[F_r(\boldsymbol{\mu}_2; \cdot, \boldsymbol{\Phi}, \boldsymbol{\Psi})\right], s\right\}\right)^{1/\omega}$$

do

5: **Evaluate error indicator**: Greedily select $\boldsymbol{\mu} \in \{\boldsymbol{\mu}_1, \boldsymbol{\mu}_2\}$, $\boldsymbol{y} \in \boldsymbol{\Xi}_{\boldsymbol{i}^*}$ with the largest error

$$\boldsymbol{\mu}^*, \boldsymbol{y}^* = \underset{\substack{\boldsymbol{\mu} \in \{\boldsymbol{\mu}_1, \boldsymbol{\mu}_2\} \\ \boldsymbol{y} \in \boldsymbol{\Xi}_{\boldsymbol{i}^*}}}{\arg\max}\ \rho(\boldsymbol{y}) \left\|\boldsymbol{r}(\boldsymbol{\Phi}\boldsymbol{u}_r(\boldsymbol{\mu}; \boldsymbol{y}, \boldsymbol{\Phi}, \boldsymbol{\Psi}); \boldsymbol{\mu}, \boldsymbol{y})\right\|$$

6: **Reduced-order model construction**: Update reduced basis with new snapshot

$$\boldsymbol{\Phi} = \begin{bmatrix}\boldsymbol{\Phi} & \boldsymbol{u}(\boldsymbol{\mu}^*; \boldsymbol{y}^*)\end{bmatrix}$$

7: **end while**

8: **end while**

6.3.3 Summary

The proposed multifidelity trust region method for efficient stochastic PDE-constrained optimization leverages the trust region framework of Chapter 3 and the two-level approximation of risk-averse measures of PDE quantities of interest using anisotropic sparse grids and projection-based model reduction. The ingredients required for the trust region algorithm in the present context were introduced in Section 6.3.1 and summarized below

$$\begin{aligned} m_k(\boldsymbol{\mu}) &= \mathbb{E}_{\mathcal{I}_k}\left[F_r(\boldsymbol{\mu};\,\cdot\,,\,\boldsymbol{\Phi}_k,\,\boldsymbol{\Psi}_k)\right] \\ \vartheta_k(\boldsymbol{\mu}) &= \|\boldsymbol{\mu}-\boldsymbol{\mu}_k\| \\ \varphi_k(\boldsymbol{\mu}) &= \beta_1\mathcal{E}_1(\boldsymbol{\Phi}_k,\,\boldsymbol{\Psi}_k,\,\mathcal{I}_k,\,\boldsymbol{\mu}) + \beta_2\mathcal{E}_3(\boldsymbol{\Phi}_k,\,\boldsymbol{\Psi}_k,\,\mathcal{I}_k,\,\boldsymbol{\mu}) + \\ &\quad \beta_3\mathcal{E}_4(\boldsymbol{\Phi}_k,\,\boldsymbol{\Psi}_k,\,\mathcal{I}_k,\,\boldsymbol{\mu}) \\ \psi_k(\boldsymbol{\mu}) &= \mathbb{E}_{\mathcal{I}'_k}\left[F_r(\boldsymbol{\mu};\,\cdot\,,\,\boldsymbol{\Phi}'_k,\,\boldsymbol{\Psi}'_k)\right] \\ \theta_k(\boldsymbol{\mu}) &= \alpha_1(\mathcal{E}_1(\boldsymbol{\Phi}'_k,\,\boldsymbol{\Psi}'_k,\,\mathcal{I}'_k,\,\boldsymbol{\mu}_k) + \mathcal{E}_1(\boldsymbol{\Phi}'_k,\,\boldsymbol{\Psi}'_k,\,\mathcal{I}'_k,\,\boldsymbol{\mu})) + \\ &\quad \alpha_2(\mathcal{E}_2(\boldsymbol{\Phi}'_k,\,\boldsymbol{\Psi}'_k,\,\mathcal{I}'_k,\,\boldsymbol{\mu}_k) + \mathcal{E}_2(\boldsymbol{\Phi}'_k,\,\boldsymbol{\Psi}'_k,\,\mathcal{I}'_k,\,\boldsymbol{\mu})). \end{aligned} \tag{6.64}$$

The sparse grid $\mathcal{I}_k$ and reduced-order model $\boldsymbol{\Phi}_k$, $\boldsymbol{\Psi}_k$ are constructed using the dimension-adaptive greedy algorithm (Algorithm 13) to ensure the gradient condition (6.53) is satisfied. The algorithm is initialized from the sparse grid and reduced-order model from the previous iteration

$$\boldsymbol{\Phi}_k,\,\boldsymbol{\Psi}_k,\,\mathcal{I}_k = \texttt{two-level-refine-grad}(\boldsymbol{\Phi}_{k-1},\,\boldsymbol{\Psi}_{k-1},\,\mathcal{I}_{k-1},\,\boldsymbol{\mu}_k) \tag{6.65}$$

in the event the quantities satisfy the gradient condition without refinement, e.g., if a small step is taken, which would save queries to the high-dimensional model. Once the dimension-adaptive greedy algorithm has been applied to satisfy the gradient condition (3.15), the objective decrease condition in (3.14) holds trivially since $\vartheta_k(\boldsymbol{\mu})$ is taken as the classical trust region constraint (Section 3.1.1). The sparse grid $\mathcal{I}'_k$ and reduced-order model $\boldsymbol{\Phi}'_k$, $\boldsymbol{\Psi}'_k$ defining the inexact objective decrease used in the computation of ρ_k are constructed using a similar dimension-adaptive greedy algorithm (Algorithm 13). In this case, primal high-dimensional model snapshots are used to reduce the error terms $\mathcal{E}_1$ and $\mathcal{E}_2$ at two parameters—the trust region center $\boldsymbol{\mu}_k$ and candidate step $\hat{\boldsymbol{\mu}}_k$—to satisfy the objective error condition (6.60)

$$\boldsymbol{\Phi}'_k,\,\boldsymbol{\Psi}'_k,\,\mathcal{I}'_k = \texttt{two-level-refine-obj}(\boldsymbol{\Phi}_k,\,\boldsymbol{\Psi}_k,\,\mathcal{I}_k,\,\boldsymbol{\mu}_k,\,\hat{\boldsymbol{\mu}}_k,\,r_k). \tag{6.66}$$

This initialization of the dimension-adaptive algorithm will take $\mathcal{I}'_k = \mathcal{I}_k$, $\boldsymbol{\Phi}'_k = \boldsymbol{\Phi}_k$, $\boldsymbol{\Psi}'_k = \boldsymbol{\Psi}_k$ if they satisfy the objective error condition, which may save substantial computational resources as it will eliminate (possibly many) queries to realizations of the high-dimensional model. With these choices, it is clear from (6.43) and (6.49) that $m_k(\boldsymbol{\mu}_k) = \psi_k(\boldsymbol{\mu}_k)$ and $m_k(\hat{\boldsymbol{\mu}}_k) = \psi_k(\hat{\boldsymbol{\mu}}_k)$. This implies ρ_k is unity and the step can be accepted with no additional work.

The complete multifidelity trust region algorithm, including inexact evaluation of the actual-to-predicted ratio using $\psi_k(\boldsymbol{\mu})$, is presented in Algorithm 16. Global convergence is not strictly guaranteed since the error indicators $\varphi_k(\boldsymbol{\mu})$ and $\theta_k(\boldsymbol{\mu})$ do not necessarily lead to bounds of the form (3.13) and (3.21) due to the approximation $\mathbb{E}_{\mathcal{I}^c} \approx \mathbb{E}_{\mathcal{N}(\mathcal{I})}$ in (6.35) and (6.36)-(6.37). However, even though the bounds cannot be rigorously established in the general case, the fact that the bounds are only required up to an arbitrary constant leaves hope they will hold in specific situations of interest. The numerical results in the next section provide evidence that this is the case since convergence is observed.

Future work will consider the incorporation of partially converged solutions as snapshots in the construction of $\boldsymbol{\Phi}_k$ and $\boldsymbol{\Phi}'_k$ in Algorithms 15 and 16 to further improve the efficiency of the proposed multifidelity trust region method. This will build on the idea introduced in Chapter 5; however, the implications on global convergence of the trust region framework will be more complicated to analyze since another layer of complexity is present, i.e., risk-averse measures of quantities of interest.

Algorithm 15 Trust region method based on reduced-order models and sparse grids

1: **Initialization**: Given

$$\boldsymbol{\mu}_0,\, \Delta_0,\, 0 < \gamma < 1,\, \Delta_{\max} > 0,\, 0 < \eta_1 < \eta_2 < 1,\, \kappa_\varphi > 0,$$
$$\mathcal{I}_{-1} = \{\mathbf{0}\},\, \boldsymbol{\Phi}_{-1} = \left[\boldsymbol{u}(\boldsymbol{\mu}_0;\, \mathbf{0})\right]$$

2: **Model and constraint update**: If the previous model is sufficient for convergence

$$\varphi_{k-1}(\boldsymbol{\mu}_k) \leq \kappa_\varphi \min\{\|\nabla m_{k-1}(\boldsymbol{\mu}_k)\|\,,\, \Delta_k\},$$

re-use for the current iteration: $m_k(\boldsymbol{\mu}) \coloneqq m_{k-1}(\boldsymbol{\mu})$ and $\vartheta_k(\boldsymbol{\mu}) = \|\boldsymbol{\mu} - \boldsymbol{\mu}_k\|$. Otherwise, refine the reduced-order model and sparse grid using two-level dimension adaptive greedy method

$$\boldsymbol{\Phi}_k,\, \boldsymbol{\Psi}_k,\, \mathcal{I}_k = \texttt{two-level-refine-grad}(\boldsymbol{\Phi}_{k-1},\, \boldsymbol{\Psi}_{k-1},\, \mathcal{I}_{k-1},\, \boldsymbol{\mu}_k,\, \Delta_k)$$

and define model and constraint as

$$m_k(\boldsymbol{\mu}) = \mathbb{E}_{\mathcal{I}_k}\left[f(\boldsymbol{\Phi}_k \boldsymbol{u}_r(\boldsymbol{\mu};\, \cdot\,,\, \boldsymbol{\Phi}_k,\, \boldsymbol{\Psi}_k),\, \boldsymbol{\mu},\, \cdot\,)\right]$$
$$\vartheta_k(\boldsymbol{\mu}) = \|\boldsymbol{\mu} - \boldsymbol{\mu}_k\|$$

3: **Step computation**: Approximately solve the trust region subproblem

$$\min_{\boldsymbol{\mu} \in \mathbb{R}^N} m_k(\boldsymbol{\mu}) \qquad \text{subject to} \qquad \vartheta_k(\boldsymbol{\mu}) \leq \Delta_k$$

for a candidate, $\hat{\boldsymbol{\mu}}_k$, to satisfy the fraction of Cauchy decrease

4: **Actual-to-predicted reduction**: Compute actual-to-predicted reduction ratio

$$\rho_k = \frac{F(\boldsymbol{\mu}_k) - F(\hat{\boldsymbol{\mu}}_k)}{m_k(\boldsymbol{\mu}_k) - m_k(\hat{\boldsymbol{\mu}}_k)}$$

5: **Step acceptance**:

$$\textbf{if} \quad \rho_k \geq \eta_1 \quad \textbf{then} \quad \boldsymbol{\mu}_{k+1} = \hat{\boldsymbol{\mu}}_k \quad \textbf{else} \quad \boldsymbol{\mu}_{k+1} = \boldsymbol{\mu}_k \quad \textbf{end if}$$

6: **Trust region update**:

$$\begin{array}{llll} \textbf{if} & \rho_k \leq \eta_1 & \textbf{then} & \Delta_{k+1} \in (0, \gamma\vartheta_k(\hat{\boldsymbol{\mu}}_k)] \quad \textbf{end if} \\ \textbf{if} & \rho_k \in (\eta_1, \eta_2) & \textbf{then} & \Delta_{k+1} \in [\gamma\vartheta_k(\hat{\boldsymbol{\mu}}_k), \Delta_k] \quad \textbf{end if} \\ \textbf{if} & \rho_k \geq \eta_2 & \textbf{then} & \Delta_{k+1} \in [\Delta_k, \Delta_{\max}] \quad \textbf{end if} \end{array}$$

Algorithm 16 Trust region method based on reduced-order models and sparse grids with inexact objective evaluations

1: **Initialization**: Given

$$\boldsymbol{\mu}_0,\, \Delta_0,\, 0 < \gamma < 1,\, \Delta_{\max} > 0,\, 0 < \eta_1 < \eta_2 < 1,\, \kappa_\varphi > 0,$$
$$\mathcal{I}_{-1} = \{\mathbf{0}\},\, \boldsymbol{\Phi}_{-1} = \left[\boldsymbol{u}(\boldsymbol{\mu}_0;\, \mathbf{0})\right],\, \omega \in (0,\, 1),\, \{r_k\}_{k=1}^{\infty} \text{ such that } r_k \to 0$$

2: **Model and constraint update**: If the previous model is sufficient for convergence

$$\varphi_{k-1}(\boldsymbol{\mu}_k) \leq \kappa_\varphi \min\{\|\nabla m_{k-1}(\boldsymbol{\mu}_k)\|\,,\, \Delta_k\},$$

re-use for the current iteration: $m_k(\boldsymbol{\mu}) \coloneqq m_{k-1}(\boldsymbol{\mu})$ and $\vartheta_k(\boldsymbol{\mu}) = \|\boldsymbol{\mu} - \boldsymbol{\mu}_k\|$. Otherwise, refine the reduced-order model and sparse grid using two-level dimension adaptive greedy method

$$\boldsymbol{\Phi}_k,\, \boldsymbol{\Psi}_k,\, \mathcal{I}_k = \texttt{two-level-refine-grad}(\boldsymbol{\Phi}_{k-1},\, \boldsymbol{\Psi}_{k-1},\, \mathcal{I}_{k-1},\, \boldsymbol{\mu}_k,\, \Delta_k)$$

and define model and constraint as

$$m_k(\boldsymbol{\mu}) = \mathbb{E}_{\mathcal{I}_k}\left[f(\boldsymbol{\Phi}_k \boldsymbol{u}_r(\boldsymbol{\mu};\, \cdot\,,\, \boldsymbol{\Phi}_k,\, \boldsymbol{\Psi}_k),\, \boldsymbol{\mu},\, \cdot\,)\right]$$
$$\vartheta_k(\boldsymbol{\mu}) = \|\boldsymbol{\mu} - \boldsymbol{\mu}_k\|$$

3: **Step computation**: Approximately solve the trust region subproblem

$$\min_{\boldsymbol{\mu} \in \mathbb{R}^N} m_k(\boldsymbol{\mu}) \qquad \text{subject to} \qquad \vartheta_k(\boldsymbol{\mu}) \leq \Delta_k$$

for a candidate, $\hat{\boldsymbol{\mu}}_k$, to satisfy the fraction of Cauchy decrease

4: **Computed-to-predicted reduction**: Compute computed-to-predicted reduction ratio

$$\rho_k = \begin{cases} 1 & \text{if} \quad \vartheta_k(\hat{\boldsymbol{\mu}}_k)^\omega \leq \eta \min\{m_k(\boldsymbol{\mu}_k) - m_k(\hat{\boldsymbol{\mu}}_k),\, r_k\} \\ \dfrac{\psi_k(\boldsymbol{\mu}_k) - \psi_k(\hat{\boldsymbol{\mu}}_k)}{m_k(\boldsymbol{\mu}_k) - m_k(\hat{\boldsymbol{\mu}}_k)} & \text{otherwise} \end{cases}$$

where

$$\psi_k(\boldsymbol{\mu}) \coloneqq \mathbb{E}_{\mathcal{I}'_k}\left[f(\boldsymbol{\Phi}'_k \boldsymbol{u}_r(\boldsymbol{\mu};\, \cdot\,,\, \boldsymbol{\Phi}'_k,\, \boldsymbol{\Psi}'_k),\, \boldsymbol{\mu},\, \cdot\,)\right]$$
$$\boldsymbol{\Phi}'_k,\, \boldsymbol{\Psi}'_k,\, \mathcal{I}'_k = \texttt{two-level-refine-obj}(\boldsymbol{\Phi}_k,\, \boldsymbol{\Psi}_k,\, \mathcal{I}_k,\, \boldsymbol{\mu}_k,\, \hat{\boldsymbol{\mu}}_k,\, r_k)$$

5: **Step acceptance**:

$$\textbf{if} \quad \rho_k \geq \eta_1 \quad \textbf{then} \quad \boldsymbol{\mu}_{k+1} = \hat{\boldsymbol{\mu}}_k \quad \textbf{else} \quad \boldsymbol{\mu}_{k+1} = \boldsymbol{\mu}_k \quad \textbf{end if}$$

6: **Trust region update**:

$$\begin{array}{llll} \textbf{if} & \rho_k \leq \eta_1 & \textbf{then} & \Delta_{k+1} \in (0, \gamma\vartheta_k(\hat{\boldsymbol{\mu}}_k)] \quad \textbf{end if} \\ \textbf{if} & \rho_k \in (\eta_1, \eta_2) & \textbf{then} & \Delta_{k+1} \in [\gamma\vartheta_k(\hat{\boldsymbol{\mu}}_k), \Delta_k] \quad \textbf{end if} \\ \textbf{if} & \rho_k \geq \eta_2 & \textbf{then} & \Delta_{k+1} \in [\Delta_k, \Delta_{\max}] \quad \textbf{end if} \end{array}$$

6.4 Numerical Experiment: Optimal Control of the Viscous Burgers' Equation with Uncertain Coefficients

This section studies the performance of the proposed algorithms (Algorithms 15 and 16) on a simple stochastic PDE-constrained optimization problem: optimal control of the one-dimensional viscous Burgers' equation with uncertain coefficients. This is precisely the stochastic counterpart to the problem in Section 5.5.2 used to study the deterministic trust region algorithm based on reduced-order models in Chapter 5. The optimization problem takes the form

$$\underset{\boldsymbol{\mu}\in\mathbb{R}^{n_{\boldsymbol{\mu}}}}{\text{minimize}} \quad \int_{\boldsymbol{\Xi}} \rho(\boldsymbol{y}) \left[\int_0^1 \frac{1}{2}(u(\boldsymbol{\mu}, \boldsymbol{y}, x) - \bar{u}(x))^2 \, dx + \frac{\alpha}{2} \int_0^1 z(\boldsymbol{\mu}, x)^2 \, dx \right] d\boldsymbol{y} \tag{6.67}$$

where $u(\boldsymbol{\mu}, \boldsymbol{y}, x)$ is the solution of the following parametrization of the one-dimensional viscous Burgers' equation

$$\begin{aligned} -\nu(\boldsymbol{y})\partial_{xx}u(\boldsymbol{\mu}, \boldsymbol{y}, x) + u(\boldsymbol{\mu}, \boldsymbol{y}, x)\partial_x u(\boldsymbol{\mu}, \boldsymbol{y}, x) = z(\boldsymbol{\mu}, x) \quad x \in (0, 1), \quad \boldsymbol{y} \in \boldsymbol{\Xi} \\ u(\boldsymbol{\mu}, \boldsymbol{y}, 0) = d_0(\boldsymbol{y}) \qquad u(\boldsymbol{\mu}, \boldsymbol{y}, 1) = d_1(\boldsymbol{y}). \end{aligned} \tag{6.68}$$

corresponding to the realization $\boldsymbol{y} \in \boldsymbol{\Xi}$. As in Section 5.5.2, the target state is $\bar{u}(x) \equiv 1$ and the regularization parameter is $\alpha = 10^{-3}$. This is the *risk-neutral* optimal control problem. A three-dimensional stochastic space, $\boldsymbol{\Xi} = [-1, 1]^3$, is chosen to introduce stochasticity into the viscosity and boundary conditions

$$\nu(\boldsymbol{y}) = 10^{y_1 - 2} \qquad d_0(\boldsymbol{y}) = 1 + \frac{y_2}{1000} \qquad d_1(\boldsymbol{y}) = \frac{y_3}{1000}.$$

A uniform probability distribution, $\rho(\boldsymbol{y})d\boldsymbol{y} = 2^{-3}d\boldsymbol{y}$, is chosen for simplicity, although any distribution could be used. The source term, or control, $z(\boldsymbol{\mu}, x)$ is defined by 50 cubic splines with clamped boundary conditions, which leads to 53 optimization variables. This stochastic optimal control problem is nearly identical to the one studied in [108, 109], with two exceptions being that the authors in [108, 109]: (1) considered one additional stochastic parameter governing a forcing term in (6.68) and (2) used a larger optimization parameter space consisting of the nodes of the underlying finite element shape functions. In all numerical experiments, the partial differential equation in (6.68) is discretized with 500 linear finite elements for a state space of dimension $N_{\boldsymbol{u}} = 499$, after application of the essential boundary conditions.

The initial guess for the optimal control problem taken in all numerical experiments is the constant: $z(\boldsymbol{\mu}_0, x) \equiv 1$. Figure 6.5 contains several different controls and the corresponding solution statistics of (6.68), including those corresponding to the optimal deterministic ($\boldsymbol{y} = 0$) and stochastic control. It is clear that the including stochasticity in the optimization formulation has a non-trivial impact on the optimal solution obtained. Furthermore, the stochastic formulation allows statistics of the solution and quantities of interest to be considered. The remainder of this section is devoted to studying the methods proposed in Algorithms 15 and 16 and comparing its performance to three

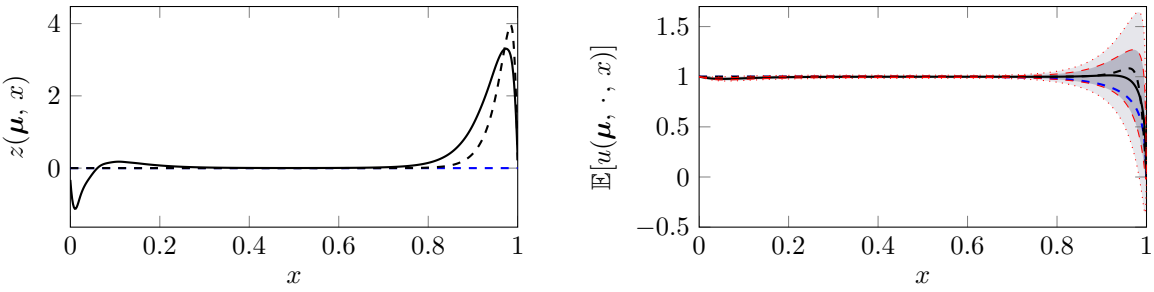


Figure 6.5: *Left*: the control defining the initial guess for the optimization problem (- - -), the solution of the deterministic optimal control problem, i.e., with the stochastic variables fixed at their mean value $\boldsymbol{y} = 0$ (- - -), and the solution of the stochastic optimal control problem (——). *Right*: the mean solution of the viscous Burgers' equation in (6.68) at the initial control (- - -), optimal deterministic control (- - -), and the optimal stochastic control. One (- - -) and two (·····) standard deviations about the mean solution corresponding to the optimal stochastic control are also included.

baseline methods.

The first method applied to the solve the stochastic optimization problem in (6.67) is Algorithm 15. Since the true function evaluations $F(\boldsymbol{\mu})$ are unavailable (the expectation cannot be computed exactly), it is approximated using a level 5 isotropic sparse grid. This amounts to a trust region method where inexactness is only used for the gradients, i.e., Algorithm 1 of Chapter 3. The work in [108] considers an identical trust region method for stochastic optimization, except the authors use an approximation model based solely on dimension-adaptive sparse grids, while the proposed method also employs projection-based reduced-order models. The reduced-order models considered in all numerical experiments use a Galerkin projection and, due to the large number of optimization variables, the adjoint method is used to compute gradients of reduced quantities of interest. To promote accuracy of the primal and adjoint solutions with respect to the HDM counterparts, the trial basis is constructed from primal and adjoint snapshots according. Such a reduced-order model does not guarantee the minimum-residual property (Definition 4.1) since the Jacobians of (6.68) are not SPD. However, the numerical experiments suggest that it is important for the reduced-order model gradients to possess discrete consistency to properly converge the trust region subproblem and ensure global convergence, particularly when there are a large number of optimization variables. This will be referred to a method **MI** in the remainder. The second stochastic optimization solver employed is exactly the method in Algorithm 16, including the approximation of the actual-to-predicted reduction ratio. This will be referred to as method **MII**. These two methods are expected to converge similarly since they are built on the same approximation framework; however, **MI** will require far more queries to the HDM since it employs a fine isotropic sparse grid to evaluate ρ_k. It is possible that method **MII** will generate an inaccurate approximation of ρ_k and will incorrectly accept or reject a step. It will be seen that does not occur in this numerical experiment.

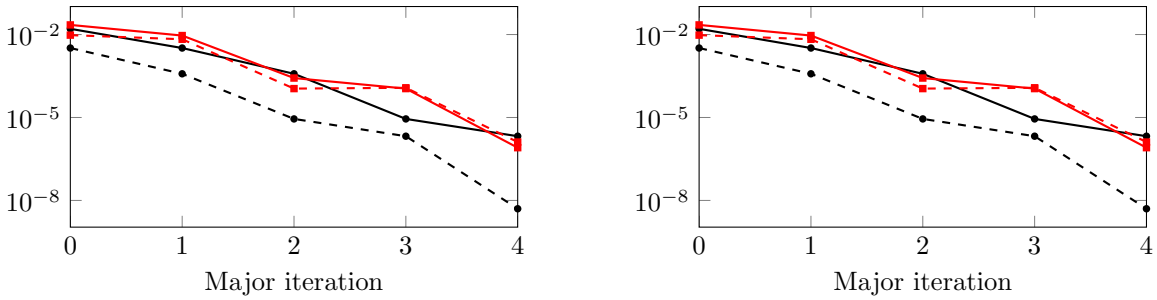


Figure 6.6: Convergence history of the objective error quantities using **MI** (left) and **MII** (right): $|F(\boldsymbol{\mu}_k) - F(\boldsymbol{\mu}^*)|$ (—•—), $|F(\hat{\boldsymbol{\mu}}_k) - F(\boldsymbol{\mu}^*)|$ (- •-), $|m_k(\boldsymbol{\mu}_k) - F(\boldsymbol{\mu}^*)|$ (—▪—), $|m_k(\hat{\boldsymbol{\mu}}_k) - F(\boldsymbol{\mu}^*)|$ (- ▪-). Rapid progress is made toward the optimal solution, despite poor agreement between the objective and model at early iterations.

To assess the performance of these proposed stochastic optimization solvers, three baseline methods will be used for comparison. The first method is the naive approach of using a fixed level 5 isotropic sparse grid to integrate the quantity of interest and the optimization problem is solved with the L-BFGS algorithm. This method will be denoted **BI**. The second, **BII**, and third, **BIII**, methods are the dimension-adaptive sparse grid approaches of [108] and [109], respectively. The stochastic optimization solvers **MI**, **MII**, **BII**, **BIII** are all trust region methods that use the Steihaug-Toint CG [48] method to approximate the solution of the trust region subproblem and use the parameters in (5.57).

The convergence history of the proposed trust region methods **MI** and **MII** are shown in Figure 6.6 and Tables 6.1–6.2. Both methods are converging to a first-order critical point ($\|\nabla F(\boldsymbol{\mu}_k)\| \to 0$); after only 5 trust region iterations the first-order optimality condition has reduced 4 orders of magnitude from the initial, sub-optimal control. At early iterations, the approximation model, $m_k(\boldsymbol{\mu})$, and true objective, $F(\boldsymbol{\mu})$, do not exhibit good agreement, even at trust region centers. In fact, from the tables, they do not even agree in the first digit. Despite this lack of agreement, the candidate step found by the approximation model leads to reasonable reduction in the true objective function. As a local minima is approached, the bound in (3.15) places more stringent requirements on the model error and, as a result, the approximation model shows excellent agreement with the objective function. The behavior of methods **MI** and **MII** are nearly identical since they rely on the same approximation model and error indicators in the trust region method. The only difference is the computation of ρ_k and, even though **MII** uses the approximation in (3.20) to compute ρ_k, it never falsely accepts or rejects a step; see Tables 6.1–6.2.

In contrast to the values of the approximation model and objective function, the gradient norms do show reasonable agreement, even at early iterations, which can be seen in Figure 6.7. This is to be expected since the refinement method in Algorithm 13 targets the gradient error. Both the values and gradients of the approximation model and objective do not show good agreement at the candidate step $\hat{\boldsymbol{\mu}}_k$, which is also expected since, at iteration k, the sparse grid and reduced-order

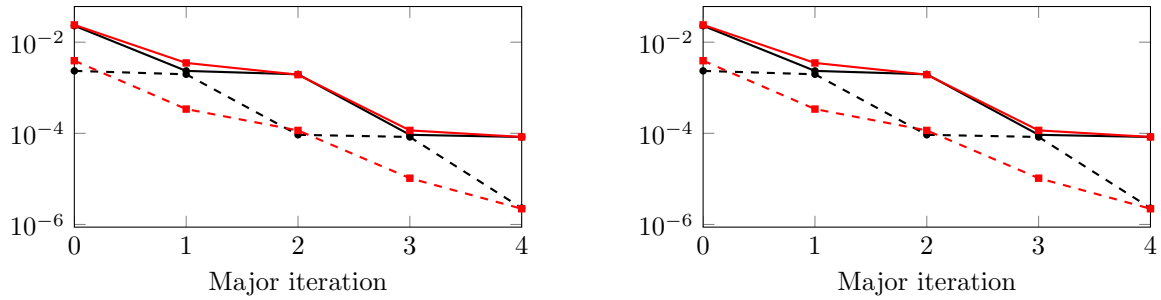


Figure 6.7: Convergence history of the gradient quantities using **MI** (left) and **MII** (right): $\|\nabla F(\boldsymbol{\mu}_k)\|$ (—•—), $\|\nabla F(\hat{\boldsymbol{\mu}}_k)\|$ (- -•- -), $\|\nabla m_k(\boldsymbol{\mu}_k)\|$ (—▪—), $\|\nabla m_k(\hat{\boldsymbol{\mu}}_k)\|$ (- -▪- -).

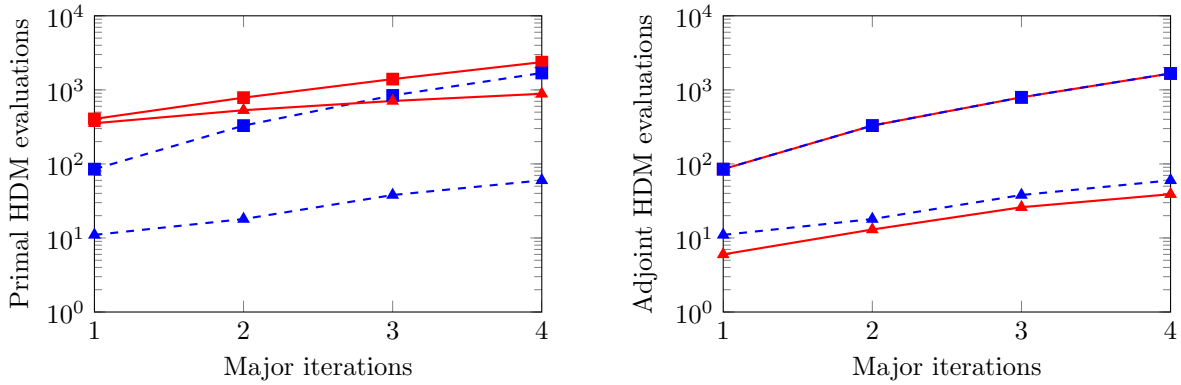


Figure 6.8: *Cumulative* number of HDM primal and adjoint evaluations as the major iterations in the various trust region algorithms progress: **BII** (—▪—), **BIII** (- -▪- -), **MI** (—▲—), **MII** (- -▲- -).

model were only trained *at the trust region center*.

Figure 6.8 provide further insight to behavior of the two proposed methods (**MI** and **MII**) in comparison to the trust region-based baseline methods (**BII** and **BIII**). All methods are based on the trust region framework in Algorithms 1 and 2 of Chapter 3 that are adapted from the work in [108, 109] and therefore possess the same concept of a *major iteration*. Figure 6.8 shows the cumulative number of queries to the high-dimensional model as the major iterations progress. The methods **BII** and **BIII** require more HDM queries (primal and adjoint) than their counterparts in **MI** and **MII** since their trust region model problems rely solely on HDM queries on an anisotropic sparse grid while **MI**, **MII** replace these with ROM queries. Another observation is that the **BIII** requires fewer primal HDM queries than **BII** and the same number of adjoint queries. This is expected since they both use the same approximation model in the trust region subproblem (implies same number of adjoint queries), but **BIII** uses inexact objective evaluations to evaluate the actual-to-predicted reduction ratio (implies fewer primal queries). A similar observation holds when comparing **MI** and **MII** for the same reason.

The reduction in the number of HDM queries realized by the proposed methods **MI** and **MII**

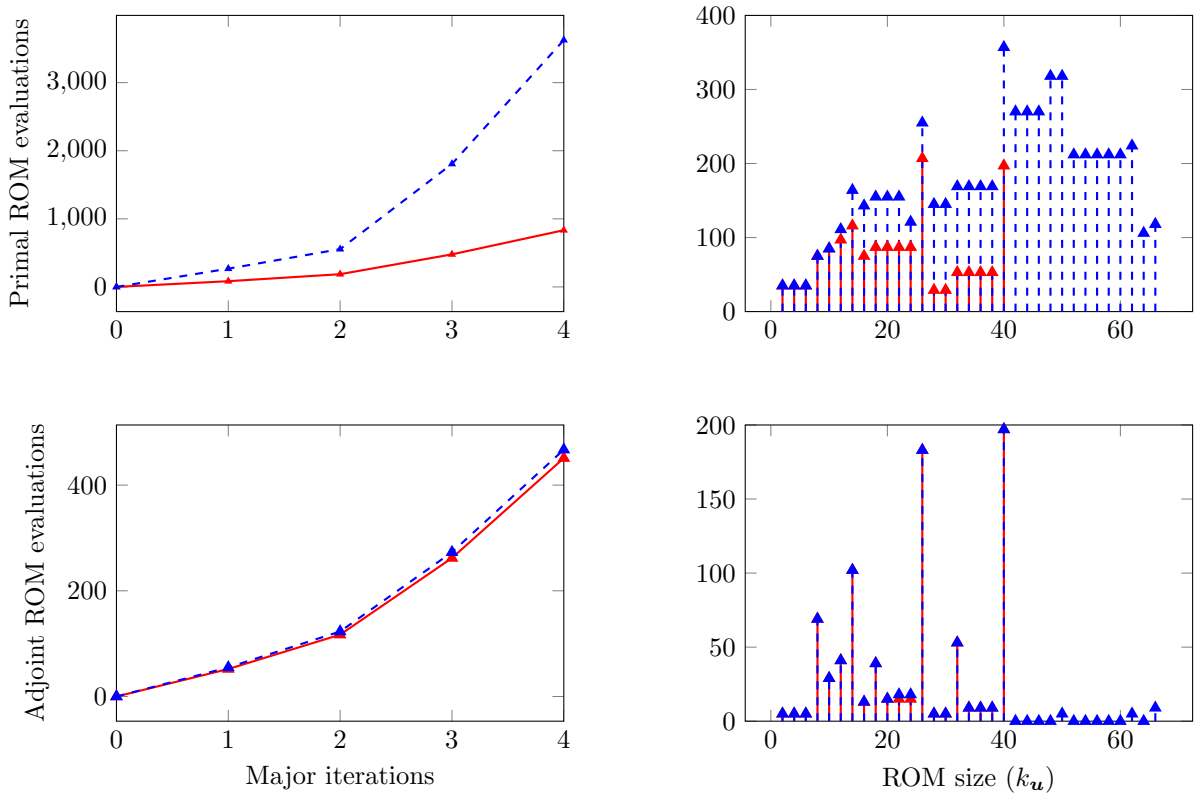


Figure 6.9: *Left*: *Cumulative* number of primal and adjoint ROM evaluations as the major iterations in the various trust region algorithms progress. *Right*: Number of primal and adjoint ROM queries organized according to the size of the reduced-order basis ($k_{\boldsymbol{u}}$). Trust region methods considered: MI (—▲—), MII (- -▲- -).

comes at the price of a large number of ROM queries. This can be seen from Figure 6.9 that includes the cumulative number of queries to the primal and adjoint ROM. Since the size of the reduced-order model constantly changes as these algorithms progress, the number of queries to a reduced-order model of a given size is also presented in Figure 6.9. Method **MII** requires nearly three times as many primal reduced-order queries as **MI**, but nearly the same number of adjoint queries. This comes from the fact that **MII** uses a (possibly) refined reduced-order model to approximation ρ_k, while **MI** uses high-dimensional model queries to compute it exactly. This also explains the fact that larger reduced-order models are required for **MII** and nearly all of these large reduced-order models are only called upon for a *primal* solve only, i.e., few adjoint solves for reduced-order models of size > 40.

Since the proposed method **MI** and **MII** and the baseline methods **BI**–**BIII** have different sources of cost, i.e., HDM and ROM evaluations versus only HDM evaluations, care must be taken when assessing the performance of the methods. The ultimate cost metric of interest is wall time; however, this one-dimensional model problem will not be representative of the speedups that can be

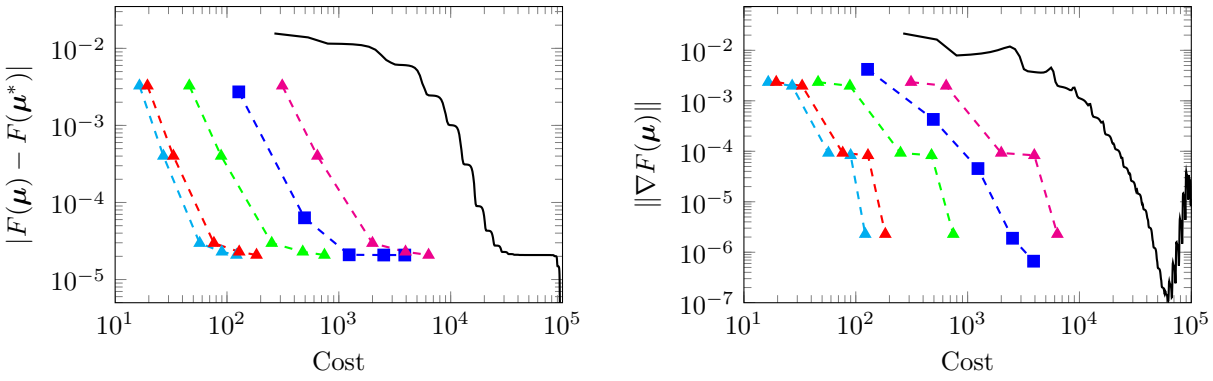


Figure 6.10: Convergence of the objective function (left) and gradient (right) as a function of the cost metric in (6.69) for method **MIII** for several values of the speedup factor of the reduced-order model: $\tau = 1$ (-▲-), $\tau = 10$ (-▲-), $\tau = 100$ (-▲-), $\tau = \infty$ (-▲-). The baseline methods used for comparison: **BI** (——) and **BIII** (-■-).

realized by methods **MI** and **MII**. Due to the small problem size and the fact that hyperreduction has not been included to reduce the complexity associated with the nonlinear term, queries to the reduced-order model are only marginally less expensive than the HDM queries. For larger problems that include hyperreduction, the motivation for this work, ROM queries have been shown [198] to be one to five orders of magnitude less expensive than HDM queries. To assess the speedups that can be realized by this method, the following simple cost model is introduced

$$C = n_{hp} + n_{ha}/2 + \tau^{-1}(n_{rp} + n_{ra}/2) \tag{6.69}$$

where C is the total cost associated with a particular method in the units of *equivalent number of primal HDM queries*, n_{hp} is the number of primal HDM queries, n_{ha} is the number of adjoint HDM queries, n_{rp} is the number of primal ROM queries, n_{ra} is the number of adjoint ROM queries, and τ is the ratio of the cost of a primal HDM query to a primal ROM query. This cost model assumes a primal solve is twice as expensive as an adjoint solve and a primal HDM solve is τ times as expensive as a primal ROM solve. Figure 6.10 shows the evolution of the objective function and gradient as a function of the cost metric in (6.69) for the baseline methods **BI**, **BIII** and the proposed method **MII** for $\tau = 1, 10, 100, \infty$. Even for slow reduced-order models ($\tau = 1$), **MII** exhibits faster convergence than the brute-force baseline method **BI**; however, it converges more slowly than the state-of-the-art method **BIII**. For a modest ROM speedup of $\tau = 10$, **MII** is more than 5× less expensive than **BIII**, i.e., for a given value of the objective function or gradient, the cost of **MII** is less than a fifth of **BIII**. For fast reduced-order models ($\tau = 100$), **MII** is an order of magnitude more efficient than **BIII**. An upper bound on the improvement attainable by **MII** compared to **BIII** is slightly greater than an order of magnitude, which is seen from the limiting case of *free* reduced-order model ($\tau = \infty$) in Figure 6.10.

Table 6.1: Convergence history of Algorithm 15 applied to the optimal control of the stochastic Burgers' equation in (6.67).

$F(\boldsymbol{\mu}_k)$	$m_k(\boldsymbol{\mu}_k)$	$F(\hat{\boldsymbol{\mu}}_k)$	$m_k(\hat{\boldsymbol{\mu}}_k)$	$\|\nabla F(\boldsymbol{\mu}_k)\|$	ρ_k	Δ_k	Success?
6.6506e-02	7.2694e-02	5.3655e-02	5.9922e-02	2.2959e-02	1.0062e+00	1.0000e+02	1.0000e+00
5.3655e-02	5.9593e-02	5.0783e-02	5.7152e-02	2.3424e-03	1.1765e+00	2.0000e+02	1.0000e+00
5.0783e-02	5.0670e-02	5.0412e-02	5.0292e-02	1.9724e-03	9.8344e-01	4.0000e+02	1.0000e+00
5.0412e-02	5.0292e-02	5.0405e-02	5.0284e-02	9.2654e-05	8.7408e-01	8.0000e+02	1.0000e+00
5.0405e-02	5.0404e-02	5.0403e-02	5.0401e-02	8.3139e-05	9.9873e-01	1.6000e+03	1.0000e+00
5.0403e-02	5.0401e-02	-	-	2.2846e-06	-	-	-

Table 6.2: Convergence history of Algorithm 16 applied to the optimal control of the stochastic Burgers' equation in (6.67).

$F(\boldsymbol{\mu}_k)$	$m_k(\boldsymbol{\mu}_k)$	$F(\hat{\boldsymbol{\mu}}_k)$	$m_k(\hat{\boldsymbol{\mu}}_k)$	$\|\nabla F(\boldsymbol{\mu}_k)\|$	ρ_k	Δ_k	Success?
6.6506e-02	7.2694e-02	5.3655e-02	5.9922e-02	2.2959e-02	1.0257e+00	1.0000e+02	1.0000e+00
5.3655e-02	5.9593e-02	5.0783e-02	5.7152e-02	2.3424e-03	9.7512e-01	2.0000e+02	1.0000e+00
5.0783e-02	5.0670e-02	5.0412e-02	5.0292e-02	1.9724e-03	9.8351e-01	4.0000e+02	1.0000e+00
5.0412e-02	5.0292e-02	5.0405e-02	5.0284e-02	9.2654e-05	8.7479e-01	8.0000e+02	1.0000e+00
5.0405e-02	5.0404e-02	5.0403e-02	5.0401e-02	8.3139e-05	9.9946e-01	1.6000e+03	1.0000e+00
5.0403e-02	5.0401e-02	-	-	2.2846e-06	-	-	-

Chapter 7

Conclusions

7.1 Summary and Conclusions

The primary contributions of this thesis are two-fold: (1) the development of an efficient solver for deterministic PDE-constrained optimization problems that leverages projection-based reduced-order models and partially converged PDE solutions and (2) the development of an efficient solver for stochastic PDE-constrained optimization problems that leverages projection-based reduced-order models and anisotropic sparse grids. These primary contributions were built on two independent auxiliary contributions that have applications that extend well beyond the scope of this thesis: (1) the introduction of a globally convergent, highly flexible generalized trust region method for managing efficient approximation models and (2) the generalization and extension of minimum-residual projection-based reduced-order models [115, 28, 31, 89] to sensitivity and adjoint PDEs.

The multifidelity trust region method introduced in Chapter 3 extends traditional trust region methods by allowing generalized trust region constraints to be used, provided the relationship in (3.12) between the approximation model decrease error and the trust region constraint can be established. This method is said to be a "generalized" trust region since the traditional trust region constraint, i.e., a ball in $\mathbb{R}^{N_{\boldsymbol{\mu}}}$, satisfies the required relationships and is therefore a valid constraint in the proposed method. The trust region method is closely based on the methods in [108, 109] that does not require first-order consistency with the objective function and allows an approximation model to be used in the computation of the actual-to-predicted reduction. This flexibility is significant since the resulting method in Algorithm 2 does not explicitly depend on the expensive objective function $F(\boldsymbol{\mu})$; however, construction of the approximation models $m_k(\boldsymbol{\mu})$ and $\psi_k(\boldsymbol{\mu})$ will likely require (inexact) evaluations of $F(\boldsymbol{\mu})$. Furthermore, the inexactness conditions adopted from [93, 108, 109] allow for asymptotic *error bounds* between the true and approximated quantities, which provides considerable flexibility in the approximation models that can be used. Even though the trust region framework was developed in the unconstrained setting, it can be embedded in an augmented Lagrangian framework to handle nonlinear equality constraints. This

multifidelity trust region method, or trust region model management framework, constitutes one of the pillars of this thesis from which the primary contributions regarding deterministic and stochastic PDE-constrained optimization in Chapters 5 and 6 follow. The second pillar is the primary PDE approximation technology employed in this work: projection-based model reduction.

While the concept of minimum-residual projection-based reduced-order models is not new [115, 28, 31, 89], this work contributes to the understanding of this technology and extends it to apply to sensitivity and adjoint PDEs. The primary factors that motivate the use of minimum-residual reduced-order models—optimality, monotonicity, and interpolation—are stated and proved in Proposition 4.1, 4.2, 4.4 for the primal, sensitivity, and adjoint PDEs. For the primal PDE, these concepts are well-known from previous work [31], but have only been sparingly explored [210] in the sensitivity/adjoint settings. These properties are crucial when reduced-order models are combined with the generalized trust region method of Chapter 3 as the trust region convergence theory places specific requirements on the accuracy of the approximation model at trust region centers, which is closely linked to these minimum-residual properties and the construction of the reduced-order bases. Propositions 4.3 and 4.5 are particularly important contributions of this thesis to the model reduction literature as they state conditions under which minimum-residual sensitivities/adjoints coincide with the true sensitivities/adjoints of the reduced-order model. These results provide insight into the construction of the reduced-order basis and ensures the true reduced-order model sensitivities/adjoints possess the minimum-residual properties (optimality, monotonicity, and interpolation). This is particularly important in the context of optimization since it guarantees the minimum-residual sensitivities/adjoints will lead to consistent gradients of QoIs based on the reduced-order model, which is extensively leveraged in the deterministic and stochastic PDE-constrained optimization methods of Chapters 5 and 6. Finally, the minimum-residual sensitivities/adjoints are much easier to implement and compute than their exact counterparts when a non-constant test basis is used since they do not require second derivatives of the governing equations. These results surrounding minimum-residual reduced-order models were extended to the case of collocation-based hyperreduction where the residual minimization occurs only over the subset of the degrees of freedom in the *mask*. Weaker versions of the crucial propositions mentioned above were established in this setting (Propositions 4.6 – 4.8).

These two technologies—the generalized trust region method and minimum-residual projection-based reduced-order models—serve as pillars for the primary contributions of the thesis: efficient optimization methods for deterministic and stochastic PDE-constrained optimization. The proposed method for deterministic PDE-constrained optimization uses projection-based reduced-order models as the approximation model in the generalized trust region method and residual-based error indicators (Appendix B justifies the use of residual-based error indicators as error *bounds*). The minimum-residual properties of the reduced-order models, as well as the compression algorithms in Section 4.3, are used to build a ROM that exactly satisfies the error conditions in (3.14), (3.15), which guarantees global convergence. The flexibility of the trust region framework is leveraged to use partially converged PDE solutions as snapshots for the reduced-order model and to approximate

the actual-to-predicted reduction. The proposed method is applied to a number of PDE-constrained optimization problems in fluid mechanics. The large-scale industrial example of aerodynamic shape optimization of the Common Research Model demonstrated the potential of the proposed method to be 1.6× faster than a state-of-the-art PDE-constrained optimization solver.

The multifidelity trust region method proposed as an efficient solver for stochastic PDE-constrained optimization problems in Chapter 6 requires a second level of inexactness to efficiently integrate quantities of interest over the stochastic space to form risk measures (Section 2.2.1). This lead to the development of the two-level approximation of risk measures of PDE quantities of interest that uses dimension-adaptive anisotropic sparse grids to perform efficient integration in the stochastic space and model reduction for efficient PDE queries at each collocation node. This two-level approximation was used to define the approximation model in the multifidelity trust region method and suitable error indicators were derived that take both the model reduction error and integral truncation error into account. Global convergence is established by employing a two-level dimension-adaptive greedy algorithm to simultaneously construct the sparse grid and reduced-order basis to satisfy the error conditions (3.14), (3.15). The proposed method directly extends the work in [108, 109] that only defines the approximation model using dimension-adaptive sparse grids with PDE queries at collocation nodes performed using the HDM. It is also similar to [44, 42, 43] that employs the same two-level approximation, but embeds it in an offline-online framework and claims regarding convergence only apply to simple PDEs. The numerical experiment in Chapter 6 demonstrate the promise of this method as a 500-fold reduction in the cost metric (6.69), compared to using a fine isotropic sparse grid without reduced-order models to perform the stochastic optimization, was realized. Even compared to the method in [109] that is considered state-of-the-art, a 10-fold reduction in the cost metric was realized.

7.2 Prospective Future Work

This thesis leaves a variety of research issues and spin-off projects that constitute promising avenues of future research. These research direction include:

- *Possible improvements to the proposed methods.* A number of possible improvements to the various methods proposed in this thesis are apparent. The first is a theoretical matter to extend the *liminf* statement on global convergence in Appendix A to the stronger *lim* convergence. Another independent issue that should be addressed is the complete formulation of the minimum-residual adjoint equations for the collocation-based hyperreduced models. As was pointed out in Section 4.2.6, this is delicate due to the differences between the mask and sample mesh that become a factor when considering the transpose of the Jacobian (as required by the adjoint residual). Another possible enhancement that would have a positive and widespread impact across the methods proposed in this thesis is the use of improved, faster, and possibly probabilistic, [15] error indicators. In the trust region framework, these can either be used as

the trust region constraint or as the gradient error indicator. Finally, in the context of optimization under uncertainty, the use of sparse grids implicitly assumes the risk measures are sufficiently smooth, which eliminates many of the most interesting and relevant risk measures in Section 2.2.1. To enable the use of these alternate risk measures, future work should focus on an alternative construction of collocation nodes that explicitly deals with non-smoothness.

- *Extension to problems with large-scale parameter spaces, $N_{\boldsymbol{\mu}} = \mathcal{O}(N_{\boldsymbol{u}})$.* All of the methods developed in this document assume there are few parameters compared to the dimension of the state vector, i.e., $N_{\boldsymbol{\mu}} \ll N_{\boldsymbol{u}}$, since reduction was only applied to the state vector. To handle the more complicated case where $N_{\boldsymbol{\mu}} = \mathcal{O}(N_{\boldsymbol{u}})$ that arises in applications such as topology optimization or inverse problems, reduction of some form must be applied to the parameter space as well. Evaluation of the reduced-order model will require at least $\mathcal{O}(N_{\boldsymbol{\mu}})$ operations, particularly if the parameters define coefficient of the underlying PDE, e.g., material properties, and this will constitute a major bottleneck when there are $\mathcal{O}(N_{\boldsymbol{u}})$ parameters. One possible method that exploits low-dimensional search spaces employed by individual iterations of linesearch and subspace methods is outlined in Appendix C.

- *Extension to time-dependent problems, possibly with periodicity constraints.* In this thesis, all problems considered in Chapters 5 and 6 were *static*. However, optimization problems governed by time-dependent PDEs (Appendix D) would benefit most from a multifidelity approach such as the ones proposed in this thesis due to their extreme computational cost and the plethora of training information generated, even after a single query to the HDM. Extension of this work to time-dependent problems will require the development of inexpensive error indicators for the primal and sensitivity/dual; however, the optimization methods themselves do not need to be modified since they are agnostic to the form of the underlying PDE (only work with quantities of interest and their gradients).

Appendix A

Global Convergence Proof: Error-Aware Trust Region Method

This section provides the global convergence theory for the error-aware, multifidelity trust region method in Algorithm 2 for the solution of the unconstrained optimization problem

$$\underset{\boldsymbol{\mu}\in\mathbb{R}^{N_{\boldsymbol{\mu}}}}{\text{minimize}} \quad F(\boldsymbol{\mu}).$$

It largely parallels the convergence theory in [133, 108, 109] with required changes to handle the error-aware trust regions. At iteration k, define the approximation model $m_k : \mathbb{R}^{N_{\boldsymbol{\mu}}} \to \mathbb{R}$ and the error indicators $\vartheta_k : \mathbb{R}^{N_{\boldsymbol{\mu}}} \to \mathbb{R}_+$ and $\varphi_k : \mathbb{R}^{N_{\boldsymbol{\mu}}} \to \mathbb{R}_+$ such that

$$|F(\boldsymbol{\mu}_k) - F(\boldsymbol{\mu}) + m_k(\boldsymbol{\mu}) - m_k(\boldsymbol{\mu}_k)| \leq \zeta \vartheta_k(\boldsymbol{\mu})^\nu \qquad \boldsymbol{\mu} \in \mathcal{R}_k \tag{A.1}$$

$$\|\nabla F(\boldsymbol{\mu}_k) - \nabla m_k(\boldsymbol{\mu}_k)\| \leq \xi \varphi_k(\boldsymbol{\mu}_k) \tag{A.2}$$

where ζ, $\xi > 0$ are *arbitrary* constants, $\nu > 1$, $\{\boldsymbol{\mu}_k\}$ is the sequences of iterates produced by the Algorithm 2, and $\mathcal{R}_k = \{\boldsymbol{\mu} \in \mathbb{R}^{N_{\boldsymbol{\mu}}} \mid \vartheta_k(\boldsymbol{\mu}) \leq \Delta_k\}$ are the sublevel sets of the error indicator $\vartheta_k(\boldsymbol{\mu})$. Furthermore, require the approximation is refined such that the error indicators satisfy the following conditions at trust region centers

$$\vartheta_k(\boldsymbol{\mu}_k) \leq \kappa_\vartheta \Delta_k \tag{A.3}$$

$$\varphi_k(\boldsymbol{\mu}_k) \leq \kappa_\varphi \min\{\|\nabla m_k(\boldsymbol{\mu}_k)\|, \Delta_k\}, \tag{A.4}$$

where $\kappa_\vartheta \in (0, 1)$ and $\kappa_\varphi > 0$ are algorithmic constants. Additionally, define an approximation model for the objective function $\psi_k : \mathbb{R}^{N_{\boldsymbol{\mu}}} \to \mathbb{R}$ and corresponding error indicator $\theta_k : \mathbb{R}^{N_{\boldsymbol{\mu}}} \to \mathbb{R}_+$ that satisfy

$$|F(\boldsymbol{\mu}_k) - F(\boldsymbol{\mu}) + \psi_k(\boldsymbol{\mu}) - \psi_k(\boldsymbol{\mu}_k)| \leq \sigma \theta_k(\boldsymbol{\mu}) \tag{A.5}$$

where $\sigma > 0$ is an arbitrary constant. Finally, require the objective approximation is refined such that the error indicators satisfies

$$\theta_k(\hat{\boldsymbol{\mu}}_k)^\omega \leq \eta \min\{m_k(\boldsymbol{\mu}_k) - m_k(\hat{\boldsymbol{\mu}}_k),\, r_k\} \tag{A.6}$$

where $\eta < \min\{\eta_1,\, 1 - \eta_2\}$, $\omega \in (0,\, 1)$ and $0 < \eta_1 < \eta_2 < 1$ are algorithmic constants, $\{r_k\}_{k=1}^{\infty}$ such that $r_k \to 0$ is a forcing sequence, and $\hat{\boldsymbol{\mu}}_k$ is the solution of the trust region subproblem at iteration k

$$\begin{aligned} \underset{\boldsymbol{\mu} \in \mathbb{R}^{N_{\boldsymbol{\mu}}}}{\text{minimize}} \quad & m_k(\boldsymbol{\mu}) \\ \text{subject to} \quad & \vartheta_k(\boldsymbol{\mu}) \leq \Delta_k. \end{aligned}$$

Before proceeding the main content of this section, the global convergence proof of Algorithm 2, additional assumptions are introduced on the regularity and boundedness of the objective function $F(\boldsymbol{\mu})$ and approximation model $m_k(\boldsymbol{\mu})$ in Assumptions A.1 and A.2.

Assumption A.1 (Objective function assumptions)**.**

(AF1) $F : \mathbb{R}^{N_{\boldsymbol{\mu}}} \to \mathbb{R}$ *is twice-continuously differentiable on* $\mathbb{R}^{N_{\boldsymbol{\mu}}}$

(AF2) $F(\boldsymbol{\mu})$ *is bounded below on* $\mathbb{R}^{N_{\boldsymbol{\mu}}}$*, i.e., there exists* $\kappa_{lbf} > 0$ *such that, for all* $\boldsymbol{\mu} \in \mathbb{R}^{N_{\boldsymbol{\mu}}}$*,*

$$F(\boldsymbol{\mu}) \geq \kappa_{lbf}$$

Assumption A.2 (Approximation model assumptions)**.**

(AM1) $m_k : \mathbb{R}^{N_{\boldsymbol{\mu}}} \to \mathbb{R}$ *is twice-continuously differentiable on* $\mathbb{R}^{N_{\boldsymbol{\mu}}}$

(AM2) The Hessian of the model remains bounded within the trust region, i.e.,

$$\beta_k \coloneqq 1 + \sup_{\boldsymbol{\mu} \in \mathcal{R}_k} \left\| \nabla^2 m_k(\boldsymbol{\mu}) \right\| \leq \kappa_{umh}$$

where $\kappa_{umh} \geq 1$

(AM3) $\vartheta_k : \mathbb{R}^{N_{\boldsymbol{\mu}}} \to \mathbb{R}$ *is continuously differentiable on* $\mathbb{R}^{N_{\boldsymbol{\mu}}}$

(AM4) The directional derivative of the constraint in any direction $\boldsymbol{p}_k$ *is bounded in the trust region, i.e.,*

$$\sup_{\boldsymbol{\mu} \in \mathcal{R}_k} \left| \frac{\partial \vartheta_k}{\partial \boldsymbol{\mu}}(\boldsymbol{\mu}) \boldsymbol{p}_k \right| \leq \kappa_{\nabla \vartheta}$$

where $\kappa_{\nabla \vartheta} > 0$

(AM5) The trust region subproblem

$$\begin{aligned} \underset{\boldsymbol{\mu} \in \mathbb{R}^{N_{\boldsymbol{\mu}}}}{\textit{minimize}} \quad & m_k(\boldsymbol{\mu}) \\ \textit{subject to} \quad & \vartheta_k(\boldsymbol{\mu}) \leq \Delta_k. \end{aligned}$$

has a local solution, which is guaranteed if $m_k(\boldsymbol{\mu})$ has a local minima in the interior of $\mathcal{R}_k$ or $\mathcal{R}_k$ is compact and a local solution lies on the boundary of the trust region $\partial\mathcal{R}_k \coloneqq \{\boldsymbol{\mu} \in \mathbb{R}^{N_\mu} \mid \vartheta_k(\boldsymbol{\mu}) = \Delta_k\}$.

Lemma A.1 (Circumscribe ball with radius proportional to Δ_k inside $\mathcal{R}_k$)**.** *Assume (AM3)–(AM4) hold. Then,*

$$\mathcal{D}_k \coloneqq \{\boldsymbol{\mu} \in \mathbb{R}^{N_\mu} \mid \|\boldsymbol{\mu} - \boldsymbol{\mu}_k\|_2 \leq (1 - \kappa_\vartheta)\kappa_{\nabla\vartheta}^{-1}\Delta_k\} \subseteq \mathcal{R}_k. \tag{A.7}$$

Proof. Let $\boldsymbol{p}_k$ be an arbitrary unit vector and take $\boldsymbol{\mu} \in \mathcal{D}_k$ such that $\boldsymbol{\mu} = \boldsymbol{\mu}_k + \alpha\boldsymbol{p}_k$. From the definition of $\mathcal{D}_k$ in (A.7), $\alpha \leq (1 - \kappa_\vartheta)\kappa_{\nabla\vartheta}^{-1}\Delta_k$. The mean value theorem, bound on $\vartheta_k(\boldsymbol{\mu}_k)$, and bound on the directional derivatives of $\vartheta_k(\boldsymbol{\mu})$ lead to

$$\vartheta_k(\boldsymbol{\mu}) = \vartheta_k(\boldsymbol{\mu}_k + \alpha\boldsymbol{p}_k) = \vartheta_k(\boldsymbol{\mu}_k) + \alpha\frac{\partial\vartheta_k}{\partial\boldsymbol{\mu}}(\boldsymbol{\zeta})\boldsymbol{p}_k \leq \kappa_\vartheta\Delta_k + \alpha\kappa_{\nabla\vartheta} \tag{A.8}$$

where $\boldsymbol{\zeta} = \boldsymbol{\mu}_k + \tau\alpha\boldsymbol{p}_k$ for some $\tau \in [0,\, 1]$. The bound on α that results from $\boldsymbol{\mu} \in \mathcal{D}_k$, along with the relation in (A.8) leads to

$$\vartheta_k(\boldsymbol{\mu}) \leq \Delta_k$$

and therefore $\boldsymbol{\mu} \in \mathcal{R}_k$. Thus, $\mathcal{D}_k \subseteq \mathcal{R}_k$. □

Lemma A.2 (Fraction of Cauchy Decrease)**.** *Assume (AM1) and (AM3)–(AM4) hold. Then there exists $\boldsymbol{\mu} \in \mathcal{R}_k$ such that*

$$m_k(\boldsymbol{\mu}_k) - m_k(\boldsymbol{\mu}) \geq \kappa_s \|\nabla m_k(\boldsymbol{\mu}_k)\| \min\left\{\frac{\|\nabla m_k(\boldsymbol{\mu}_k)\|}{\beta_k},\ (1 - \kappa_\vartheta)\kappa_{\nabla\vartheta}^{-1}\Delta_k\right\} \tag{A.9}$$

for $\kappa_s \in (0,\, 1)$.

Proof. From Theorem 6.3.3 of [48], there exists $\boldsymbol{\mu} \in \mathcal{D}_k$ such that holds

$$m_k(\boldsymbol{\mu}_k) - m_k(\boldsymbol{\mu}) \geq \kappa_s \|\nabla m_k(\boldsymbol{\mu}_k)\| \min\left\{\frac{\|\nabla m_k(\boldsymbol{\mu}_k)\|}{\beta_k},\ \delta\right\},$$

where $\kappa_s \in (0,\, 1)$ and $\delta = (1 - \kappa_\vartheta)\kappa_{\nabla\vartheta}^{-1}\Delta_k$ is the radius of $\mathcal{D}_k$. From Lemma A.1, $\mathcal{D}_k \subseteq \mathcal{R}_k$. Therefore there exists $\boldsymbol{\mu} \in \mathcal{R}_k$ such that (A.9) holds. □

Lemma A.3. *Assume (AM1), (AM3)–(AM4), (AM5) hold. Then the solution of the optimization problem*

$$\begin{aligned} \underset{\boldsymbol{\mu} \in \mathbb{R}^{N_\mu}}{\text{minimize}} \quad & m_k(\boldsymbol{\mu}) \\ \text{subject to} \quad & \vartheta_k(\boldsymbol{\mu}) \leq \Delta_k \end{aligned}$$

satisfies

$$m_k(\boldsymbol{\mu}_k) - m_k(\boldsymbol{\mu}) \geq \kappa_s \|\nabla m_k(\boldsymbol{\mu}_k)\| \min\left\{\frac{\|\nabla m_k(\boldsymbol{\mu}_k)\|}{\beta_k},\ (1 - \kappa_\vartheta)\kappa_{\nabla\vartheta}^{-1}\Delta_k\right\}$$

Proof. From Assumption (AM5), a solution of the optimization problem exists. By Lemma A.2, there exists a point in the feasible set of the optimization problem, i.e., $\mathcal{R}_k$, that satisfies (A.9). The (global) solution of the optimization problem must realize (at least) the same reduction in the objective function, which leads to the desired result. □

Lemma A.4. *If the objective approximation error bound (A.5) and accuracy condition (A.6) hold, then for k sufficiently large*

$$|F(\boldsymbol{\mu}_k) - F(\hat{\boldsymbol{\mu}}_k) + \psi_k(\hat{\boldsymbol{\mu}}_k) - \psi_k(\boldsymbol{\mu}_k)| \leq \eta \min\{m_k(\boldsymbol{\mu}_k) - m_k(\hat{\boldsymbol{\mu}}_k),\, r_k\}$$

Proof. The forcing sequence, $\{r_k\}$, in the bound on θ_k implies $\theta_k \to 0$. Therefore, for sufficiently large k, $\theta_k \leq \sigma^{-1/(1-\omega)}$. Then, (A.5), $\theta_k \leq \sigma^{-1/(1-\omega)}$, and (A.6) lead to the desired result

$$|F(\boldsymbol{\mu}_k) - F(\hat{\boldsymbol{\mu}}_k) + \psi_k(\hat{\boldsymbol{\mu}}_k) - \psi_k(\boldsymbol{\mu}_k)| \leq \sigma\theta_k = \sigma\theta_k^{\omega}\theta_k^{(1-\omega)} \leq \theta_k^{\omega} \leq \eta \min\{m_k(\boldsymbol{\mu}_k) - m_k(\hat{\boldsymbol{\mu}}_k),\, r_k\}.$$

□

Lemma A.5. *If the objective approximation error bound (A.5) and accuracy condition (A.6) hold, then for k sufficiently large*

$$\rho_k^* := \frac{F(\boldsymbol{\mu}_k) - F(\hat{\boldsymbol{\mu}}_k)}{m_k(\boldsymbol{\mu}_k) - m_k(\hat{\boldsymbol{\mu}}_k)} \in [\rho_k - \eta,\, \rho_k + \eta],$$

where

$$\rho_k = \frac{\psi_k(\boldsymbol{\mu}_k) - \psi_k(\hat{\boldsymbol{\mu}}_k)}{m_k(\boldsymbol{\mu}_k) - m_k(\hat{\boldsymbol{\mu}}_k)}.$$

Proof. For sufficiently large k,

$$\rho_k^* = \rho_k + (\rho_k^* - \rho_k) = \rho_k + \frac{F(\boldsymbol{\mu}_k) - F(\hat{\boldsymbol{\mu}}_k) + \psi_k(\hat{\boldsymbol{\mu}}_k) - \psi_k(\boldsymbol{\mu}_k)}{m_k(\boldsymbol{\mu}_k) - m_k(\hat{\boldsymbol{\mu}}_k)}.$$

Then, Lemma A.4 leads to the desired result

$$|\rho_k^* - \rho_k| \leq \frac{|F(\boldsymbol{\mu}_k) - F(\hat{\boldsymbol{\mu}}_k) + \psi_k(\hat{\boldsymbol{\mu}}_k) - \psi_k(\boldsymbol{\mu}_k)|}{m_k(\boldsymbol{\mu}_k) - m_k(\hat{\boldsymbol{\mu}}_k)} \leq \eta.$$

□

Lemma A.6. *Assume (AF2) and (AM1)–(AM4) hold and suppose there exists $\epsilon > 0$ such that $\|\nabla m_k(\boldsymbol{\mu}_k)\| \geq \epsilon$ for k sufficiently large. Then the sequence of trust region radii $\{\Delta_k\}$ produced by Algorithm 2 satisfies*

$$\sum_{k=1}^{\infty} \Delta_k < \infty.$$

Proof. If there are only a finite number of successful iterations, there exists $K > 0$ such that all iterations $k > K$ are unsuccessful. Then,

$$\begin{aligned}\sum_{k=1}^{\infty} \Delta_k &= \sum_{k=1}^{K} \Delta_k + \sum_{k=K+1}^{\infty} \Delta_k \\ &= C + \sum_{k=K+1}^{\infty} \Delta_k\end{aligned}$$

where $C = \sum_{k=1}^{K} \Delta_k < \infty$. Since iterations $k > K$ are unsuccessful, $\Delta_{k+1} \leq \gamma \Delta_k$ and $\sum_{k=K+1}^{\infty} \Delta_k$ is bounded above by a geometric series, implying the infinite sum is finite. Therefore, the result holds if there are a finite number of successful iterations. If there is an infinite sequence of successful iterations $\{k_i\}$, then for sufficiently large i

$$\begin{aligned}F(\boldsymbol{\mu}_{k_i}) - F(\hat{\boldsymbol{\mu}}_{k_i}) &\geq \psi_{k_i}(\boldsymbol{\mu}_{k_i}) - \psi_{k_i}(\hat{\boldsymbol{\mu}}_{k_i}) - \eta(m_{k_i}(\boldsymbol{\mu}_{k_i}) - m_{k_i}(\hat{\boldsymbol{\mu}}_{k_i})) \\ &\geq (\eta_1 - \eta)(m_{k_i}(\boldsymbol{\mu}_{k_i}) - m_{k_i}(\hat{\boldsymbol{\mu}}_{k_i})) \\ &\geq (\eta_1 - \eta)\kappa_s \left\|\nabla m_{k_i}(\boldsymbol{\mu}_{k_i})\right\| \min\left\{\frac{\left\|\nabla m_{k_i}(\boldsymbol{\mu}_{k_i})\right\|}{\beta_k},\ (1-\kappa_\vartheta)\kappa_{\nabla\vartheta}^{-1}\Delta_{k_i}\right\} \\ &\geq (\eta_1 - \eta)\kappa_s \epsilon \min\left\{\frac{\epsilon}{\kappa_{umh}},\ (1-\kappa_\vartheta)\kappa_{\nabla\vartheta}^{-1}\Delta_{k_i}\right\}.\end{aligned}$$

The first inequality follows from Lemma A.5, the second from the step acceptance condition in Algorithm 2, the third from the fraction of Cauchy decrease (A.9), and the fourth from the assumption that $\|\nabla m_k(\boldsymbol{\mu}_k)\| \geq \epsilon$. Summing over all i sufficiently large

$$(\eta_1 - \eta)\kappa_s \epsilon \sum_{i \geq I} \min\left\{\frac{\epsilon}{\kappa_{umh}},\ (1-\kappa_\vartheta)\kappa_{\nabla\vartheta}^{-1}\Delta_{k_i}\right\} \leq F(\boldsymbol{\mu}_{k_I}) - \lim_{i\to\infty} F(\boldsymbol{\mu}_{k_i}) < \infty$$

where the finiteness of the limit follows from F being bounded below. Since ϵ/κ_{umh} is bounded away from zero, the inequality above implies that $\sum_{i=1}^{\infty} \Delta_{k_i} < \infty$.

Let $\mathcal{S} \subset \mathbb{N}$ be the ordered set of indicies of successful iterations. For every $k \notin \mathcal{S}$, $\Delta_k \leq \gamma^{k-j(k)}\Delta_{j(k)}$ where $j(k) \in \mathcal{S}$ is the largest index such that $j(k) < k$, i.e. $j(k)$ represents the last successful iteration before the unsuccessful iteration k. Summing over all $k \notin \mathcal{S}$,

$$\sum_{k\notin\mathcal{S}} \Delta_k \leq \sum_{k\notin\mathcal{S}} \gamma^{k-j(k)}\Delta_{j(k)} = \sum_{i=1}^{\infty} \sum_{j(i)<k<j(i+1)} \gamma^{k-j(i)}\Delta_{j(i)} \leq \frac{1}{1-\gamma}\sum_{i=1}^{\infty}\Delta_{j(i)} = \frac{1}{1-\gamma}\sum_{k\in\mathcal{S}}^{\infty}\Delta_k < \infty.$$

Then,

$$\sum_{k=1}^{\infty}\Delta_k = \sum_{k\in\mathcal{S}}^{\infty}\Delta_k + \sum_{k\notin\mathcal{S}}\Delta_k \leq \left(1 + \frac{1}{1-\gamma}\right)\sum_{k\in\mathcal{S}}^{\infty}\Delta_k < \infty.$$

This proves the desired result. □

Lemma A.7. *Assume (AF2) and (AM1)–(AM4) hold and suppose there exists $\epsilon > 0$ such that $\|\nabla m_k(\boldsymbol{\mu}_k)\| \geq \epsilon$ for k sufficiently large. Then the ratios, $\{\rho_k\}$, produced by Algorithm 2, converge to one.*

Proof. From the asymptotic error bound on the approximation model in (A.1) and the fact that the candidate step lies within the trust region, i.e., $\hat{\boldsymbol{\mu}}_k \in \mathcal{R}_k$, it follows that

$$|F(\boldsymbol{\mu}_k) - F(\hat{\boldsymbol{\mu}}_k) + m_k(\hat{\boldsymbol{\mu}}_k) - m_k(\boldsymbol{\mu}_k)| \leq \zeta \vartheta_k(\hat{\boldsymbol{\mu}}_k)^\nu \leq \zeta \Delta_k^\nu. \tag{A.10}$$

From the Lemma A.2 and the convergence criteria on the trust region subproblem in Algorithm 2, we have

$$m_k(\boldsymbol{\mu}_k) - m_k(\hat{\boldsymbol{\mu}}_k) \geq \kappa_s \|\nabla m_k(\boldsymbol{\mu}_k)\| \min\left\{\frac{\|\nabla m_k(\boldsymbol{\mu}_k)\|}{\beta_k},\ (1-\kappa_\vartheta)\kappa_{\nabla\vartheta}^{-1}\Delta_k\right\}.$$

Then, for sufficiently large k,

$$m_k(\boldsymbol{\mu}_k) - m_k(\hat{\boldsymbol{\mu}}_k) \geq (1-\kappa_\vartheta)\kappa_{\nabla\vartheta}^{-1}\kappa_s\epsilon\Delta_k$$

due to Lemma A.6 and the assumption that $\|\nabla m_k(\boldsymbol{\mu})\| \geq \epsilon$. Combining these above inequalities leads to

$$|\rho_k - 1| = \left|\frac{F(\boldsymbol{\mu}_k) - F(\hat{\boldsymbol{\mu}}_k) + m_k(\hat{\boldsymbol{\mu}}_k) - m_k(\boldsymbol{\mu}_k)}{m_k(\boldsymbol{\mu}_k) - m_k(\hat{\boldsymbol{\mu}}_k)}\right| \leq \frac{\zeta\Delta_k^\nu}{(1-\kappa_\vartheta)\kappa_{\nabla\vartheta}^{-1}\kappa_s\epsilon\Delta_k} = \frac{\zeta}{(1-\kappa_\vartheta)\kappa_{\nabla\vartheta}^{-1}\kappa_s\epsilon}\Delta_k^{\nu-1}.$$

Therefore, $\rho_k \to 1$ since $\Delta_k \to 0$ (Lemma A.6) and $\nu > 1$. □

Theorem A.1. *Assume (AF1)–(AF2), (AM1)–(AM4) hold. Let $\{\boldsymbol{\mu}_k\}$ be the sequence of iterates produced by Algorithm 2 and $\{m_k\}$ the corresponding models. Then*

$$\liminf_{k\to\infty} \|\nabla m_k(\boldsymbol{\mu}_k)\| = \liminf_{k\to\infty} \|\nabla F(\boldsymbol{\mu}_k)\| = 0.$$

Proof. For contradiction, suppose there exists $\epsilon > 0$ such that $\|m_k(\boldsymbol{\mu}_k)\| \geq \epsilon$. By Lemma A.7, there exists $K > 0$ such that for all $k > K$, ρ_k is sufficiently close to 1 and the corresponding step is successful. From Algorithm 2, this implies $\Delta_K \leq \Delta_k \leq \Delta_{\max}$. This result contradicts Lemma A.6 and we must have

$$\liminf_{k\to\infty} \|\nabla m_k(\boldsymbol{\mu}_k)\| = 0.$$

From the triangle inequality and (A.2),

$$\|\nabla F(\boldsymbol{\mu}_k)\| \leq \|\nabla m_k(\boldsymbol{\mu}_k)\| + \|\nabla m_k(\boldsymbol{\mu}_k) - \nabla F(\boldsymbol{\mu}_k)\| \leq (1+\xi)\|\nabla m_k(\boldsymbol{\mu}_k)\|$$

which implies

$$\liminf_{k\to\infty} \|\nabla F(\boldsymbol{\mu}_k)\| = 0.$$

□

Appendix B

Residual-Based Error Bounds

In this section, computable error bounds on quantities of interest and their gradients are sought that will enable reduced-order models to be used in the multifidelity trust region framework introduced in Chapter 3. Global convergence (Appendix A) of the trust region method in Algorithms 1–2 requires asymptotic error bounds of the form

$$\begin{aligned} |F(\boldsymbol{\mu}_k) - F(\boldsymbol{\mu}) + m_k(\boldsymbol{\mu}) - m_k(\boldsymbol{\mu}_k)| &\leq \zeta \vartheta_k(\boldsymbol{\mu})^\nu \qquad && \boldsymbol{\mu} \in \mathcal{R}_k \\ \|\nabla F(\boldsymbol{\mu}) - \nabla m_k(\boldsymbol{\mu})\| &\leq \xi \varphi_k(\boldsymbol{\mu}) && \boldsymbol{\mu} \in \mathcal{N}_k, \end{aligned}$$

where $\zeta, \xi > 0$ are arbitrary constants, $\nu > 1$, $\mathcal{R}_k = \{\boldsymbol{\mu} \in \mathbb{R}^{N_\mu} \mid \vartheta_k(\boldsymbol{\mu}) \leq \Delta_k\}$, and $\mathcal{N}_k$ is any open neighborhood of $\boldsymbol{\mu}_k$. The constants ζ, ξ do not need to be small or even computable since they are never used in the trust region algorithm and global convergence is only predicated on their *existence*. Two key points about these error bounds that substantially reduces the burden of deriving error indicators $\vartheta_k(\boldsymbol{\mu})$ and $\varphi_k(\boldsymbol{\mu})$ are: (1) they do not need to have high effectivity to ensure global convergence and (2) they are not required to hold in the entire parameter space, only in the bounded sets $\mathcal{R}_k$ (assumed) and $\mathcal{N}_k$. Therefore, this section will consider general residual-based error bounds that hold in bounded subsets of the parameter space since they are easily derived and computed, even though they are known to have poor effectivity.

To facilitate the derivation of the residual-based error bounds, recall the following definition from (2.90) that uses an approximate primal solution $\boldsymbol{u} \in \mathbb{R}^{N_u}$ and sensitivity $\boldsymbol{w} \in \mathbb{R}^{N_u \times N_\mu}$ to reconstruct the gradient of the quantity of interest

$$\boldsymbol{g}^\partial(\boldsymbol{u}, \boldsymbol{w}, \boldsymbol{\mu}) \coloneqq \frac{\partial f}{\partial \boldsymbol{\mu}}(\boldsymbol{u}, \boldsymbol{\mu}) + \frac{\partial f}{\partial \boldsymbol{u}}(\boldsymbol{u}, \boldsymbol{\mu})\boldsymbol{w}. \tag{B.1}$$

Similarly, from (2.102), an approximate adjoint solution $\boldsymbol{z} \in \mathbb{R}^{N_u}$ can be used to approximate the gradient of the QoI as

$$\boldsymbol{g}^{\boldsymbol{\lambda}}(\boldsymbol{u}, \boldsymbol{z}, \boldsymbol{\mu}) \coloneqq \frac{\partial f}{\partial \boldsymbol{\mu}}(\boldsymbol{u}, \boldsymbol{\mu}) + \boldsymbol{z}^T \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{\mu}}(\boldsymbol{u}, \boldsymbol{\mu}). \tag{B.2}$$

Given these definitions, it is clear that

$$\nabla F(\boldsymbol{\mu}) = \boldsymbol{g}^{\partial}\left(\boldsymbol{u}(\boldsymbol{\mu}), \frac{\partial \boldsymbol{u}}{\partial \boldsymbol{\mu}}(\boldsymbol{\mu}), \boldsymbol{\mu}\right) = \boldsymbol{g}^{\boldsymbol{\lambda}}(\boldsymbol{u}(\boldsymbol{\mu}), \boldsymbol{\lambda}(\boldsymbol{\mu}), \boldsymbol{\mu}). \tag{B.3}$$

Before proceeding to the derivation of the residual-based error bounds, an operator $\boldsymbol{D}$ is defined in Definition B.1 that represents the Jacobian of the nonlinear residual integrated between states $\boldsymbol{u}_1$ and $\boldsymbol{u}_2$ for a fixed parameter $\boldsymbol{\mu}$.

Definition B.1. *Define* $\boldsymbol{D} : \mathbb{R}^{N_u} \times \mathbb{R}^{N_u} \times \mathbb{R}^{N_\mu} \to \mathbb{R}^{N_u} \times \mathbb{R}^{N_u}$ *as*

$$\boldsymbol{D}(\boldsymbol{u}_1, \boldsymbol{u}_2, \boldsymbol{\mu}) = \int_0^1 \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}(\boldsymbol{u}_2 + t(\boldsymbol{u}_1 - \boldsymbol{u}_2), \boldsymbol{\mu})\, dt. \tag{B.4}$$

Remark. *In the special case where* $\boldsymbol{r}(\boldsymbol{u}, \boldsymbol{\mu})$ *is linear in its first argument, i.e.,* $\boldsymbol{r}(\boldsymbol{u}, \boldsymbol{\mu}) = \boldsymbol{A}(\boldsymbol{\mu})\boldsymbol{u} + \boldsymbol{b}$*, then* $\boldsymbol{D}(\boldsymbol{u}_1, \boldsymbol{u}_2, \boldsymbol{\mu}) = \boldsymbol{A}(\boldsymbol{\mu})$.

The following assumptions are introduced on the nonlinear operator defining the system of equations and the quantity of interest.

Assumption B.1 (Nonlinear system assumptions)**.** *Let* $U \subset \mathbb{R}^{N_u}$ *and* $V \subset \mathbb{R}^{N_\mu}$ *be bounded subsets*

(AR1) $\boldsymbol{r} : U \times V \to \mathbb{R}^{N_u}$ *is continuously differentiable*

(AR2) The Jacobian, $\dfrac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}(\boldsymbol{u}, \boldsymbol{\mu})$*, is invertible for all* $\boldsymbol{u} \in U$ *and* $\boldsymbol{\mu} \in V$

(AR3) The inverse of the Jacobian, $\dfrac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}(\boldsymbol{u}, \boldsymbol{\mu})^{-1}$ *is bounded for all* $\boldsymbol{u} \in U$ *and* $\boldsymbol{\mu} \in V$

(AR4) The matrix $\boldsymbol{D}(\boldsymbol{u}_1, \boldsymbol{u}_2, \boldsymbol{\mu})$ *defined in (B.4) is invertible for all* $\boldsymbol{u}_1, \boldsymbol{u}_2 \in U$ *and* $\boldsymbol{\mu} \in V$

(AR5) The matrix $\boldsymbol{D}(\boldsymbol{u}_1, \boldsymbol{u}_2, \boldsymbol{\mu})^{-1}$ *is bounded for all* $\boldsymbol{u}_1, \boldsymbol{u}_2 \in U$ *and* $\boldsymbol{\mu} \in V$

(AR6) The parameter Jacobian, $\dfrac{\partial \boldsymbol{r}}{\partial \boldsymbol{\mu}}(\boldsymbol{u}, \boldsymbol{\mu})$*, is bounded for all* $\boldsymbol{u} \in U$ *and* $\boldsymbol{\mu} \in V$

(AR7) The Jacobian and its transpose, $\dfrac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}(\boldsymbol{u}, \boldsymbol{\mu})$*, are Lipschitz continuous in its first argument over* U*, i.e., there exists a constant* $c_{\partial_{\boldsymbol{u}}\boldsymbol{r}} > 0$ *such that*

$$\left\| \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}(\boldsymbol{u}_1, \boldsymbol{\mu}) - \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}(\boldsymbol{u}_2, \boldsymbol{\mu}) \right\| \leq c_{\partial_{\boldsymbol{u}}\boldsymbol{r}} \|\boldsymbol{u}_1 - \boldsymbol{u}_2\|$$

for all $\boldsymbol{u}_1, \boldsymbol{u}_2 \in U$ *and* $\boldsymbol{\mu} \in V$

(AR8) The parameter Jacobain, $\dfrac{\partial \boldsymbol{r}}{\partial \boldsymbol{\mu}}(\boldsymbol{u}, \boldsymbol{\mu})$*, is Lipschitz continuous in its first argument over* U*, i.e., there exists* $c_{\partial_{\boldsymbol{\mu}}\boldsymbol{r}} > 0$ *such that*

$$\left\| \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{\mu}}(\boldsymbol{u}_1, \boldsymbol{\mu}) - \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{\mu}}(\boldsymbol{u}_2, \boldsymbol{\mu}) \right\| \leq c_{\partial_{\boldsymbol{\mu}}\boldsymbol{r}} \|\boldsymbol{u}_1 - \boldsymbol{u}_2\|$$

for all $\boldsymbol{u}_1, \boldsymbol{u}_2 \in U$ *and* $\boldsymbol{\mu} \in V$

Assumption B.2 (Quantity of interest assumptions)**.** *Let $U \subset \mathbb{R}^{N_u}$ and $V \subset \mathbb{R}^{N_\mu}$ be bounded subsets*

(AQ1) $f : U \times V \to \mathbb{R}$ *is continuously differentiable*

(AQ2) $f : U \times V \to \mathbb{R}$ *is Lipschitz continuous with respect to its first argument, i.e., there exists a constant $c_f > 0$ such that*

$$|f(\boldsymbol{u}_1, \boldsymbol{\mu}) - f(\boldsymbol{u}_2, \boldsymbol{\mu})| \leq c_f \|\boldsymbol{u}_1 - \boldsymbol{u}_2\| \tag{B.5}$$

(AQ3) $\dfrac{\partial f}{\partial \boldsymbol{u}} : U \times V \to \mathbb{R}^{N_u}$ *is bounded and Lipschitz continuous with respect to its first argument, i.e., there exists a constant $c_{\partial_u f} > 0$ such that*

$$\left\| \frac{\partial f}{\partial \boldsymbol{u}}(\boldsymbol{u}_1, \boldsymbol{\mu}) - \frac{\partial f}{\partial \boldsymbol{u}}(\boldsymbol{u}_2, \boldsymbol{\mu}) \right\| \leq c_{\partial_u f} \|\boldsymbol{u}_1 - \boldsymbol{u}_2\| \tag{B.6}$$

(AQ4) $\dfrac{\partial f}{\partial \boldsymbol{\mu}} : U \times V \to \mathbb{R}^{N_\mu}$ *is Lipschitz continuous with respect to its first argument, i.e., there exists a constant $c_{\partial_\mu f} > 0$ such that*

$$\left\| \frac{\partial f}{\partial \boldsymbol{\mu}}(\boldsymbol{u}_1, \boldsymbol{\mu}) - \frac{\partial f}{\partial \boldsymbol{\mu}}(\boldsymbol{u}_2, \boldsymbol{\mu}) \right\| \leq c_{\partial_\mu f} \|\boldsymbol{u}_1 - \boldsymbol{u}_2\| \tag{B.7}$$

Finally, the sets U^*, W^*, Z^* are introduced as the set of primal, sensitivity, and adjoint solutions of the governing equations over a bounded set V of the parameter space.

Definition B.2. *Let $V \subseteq \mathbb{R}^{N_\mu}$ be a bounded set and define*

$$U^* = \{\boldsymbol{u}(\boldsymbol{\mu}) \mid \boldsymbol{\mu} \in V\}.$$

Furthermore, it is assumed that U^* is bounded.

Definition B.3. *Let $V \subseteq \mathbb{R}^{N_\mu}$ be a bounded set and define*

$$W^* = \left\{ \frac{\partial \boldsymbol{u}}{\partial \boldsymbol{\mu}}(\boldsymbol{\mu}) \mid \boldsymbol{\mu} \in V \right\}.$$

Boundedness of W^* is established in Lemma B.1.

Definition B.4. *Let $V \subseteq \mathbb{R}^{N_\mu}$ be a bounded set and define*

$$Z^* = \{\boldsymbol{\lambda}(\boldsymbol{\mu}) \mid \boldsymbol{\mu} \in V\}.$$

Boundedness of Z^* is established in Lemma B.2.

Lemma B.1. *Assume (AR1)–(AR3), (AR6) hold and $V \subseteq \mathbb{R}^{N_\mu}$ is a bounded subset. Then there exists a constant $\kappa > 0$ such that*

$$\sup_{\boldsymbol{\mu} \in V} \left\| \frac{\partial \boldsymbol{u}}{\partial \boldsymbol{\mu}}(\boldsymbol{\mu}) \right\| \leq \kappa \tag{B.8}$$

where $\dfrac{\partial \boldsymbol{u}}{\partial \boldsymbol{\mu}}(\boldsymbol{\mu})$ is the solution of $\boldsymbol{r}^{\partial}(\boldsymbol{u}(\boldsymbol{\mu}),\, \cdot\,,\, \boldsymbol{\mu}) = 0$ and $\boldsymbol{u}(\boldsymbol{\mu})$ is the solution of $\boldsymbol{r}(\,\cdot\,,\, \boldsymbol{\mu}) = 0$.

Proof. From the definition of $\dfrac{\partial \boldsymbol{u}}{\partial \boldsymbol{\mu}}(\boldsymbol{\mu})$ in (2.87), boundedness of U^*, and assumptions (AR2), (AR3), (AR6), there exists a constant $\kappa > 0$ such that

$$\left\| \frac{\partial \boldsymbol{u}}{\partial \boldsymbol{\mu}}(\boldsymbol{\mu}) \right\| \leq \left\| \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}(\boldsymbol{u}(\boldsymbol{\mu}), \boldsymbol{\mu})^{-1} \right\| \left\| \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{\mu}}(\boldsymbol{u}(\boldsymbol{\mu}), \boldsymbol{\mu}) \right\| \leq \kappa.$$

□

Lemma B.2. *Assume (AR1)–(AR3), (AQ3) hold and $V \subseteq \mathbb{R}^{N_\mu}$ is a bounded subset. Then there exists a constant $\kappa > 0$ such that*

$$\sup_{\boldsymbol{\mu} \in V} \|\boldsymbol{\lambda}(\boldsymbol{\mu})\| \leq \kappa \tag{B.9}$$

where $\boldsymbol{\lambda}(\boldsymbol{\mu})$ is the solution of $\boldsymbol{r}^{\boldsymbol{\lambda}}(\boldsymbol{u}(\boldsymbol{\mu}),\, \cdot\,,\, \boldsymbol{\mu}) = 0$ and $\boldsymbol{u}(\boldsymbol{\mu})$ is the solution of $\boldsymbol{r}(\,\cdot\,,\, \boldsymbol{\mu}) = 0$.

Proof. From the definition of $\boldsymbol{\lambda}(\boldsymbol{\mu})$ in (2.92), boundedness of U^*, and assumptions (AR2), (AR3), (AQ3), there exists a constant $\kappa > 0$ such that

$$\boldsymbol{\lambda}(\boldsymbol{\mu}) \leq \left\| \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}(\boldsymbol{u}(\boldsymbol{\mu}), \boldsymbol{\mu})^{-T} \right\| \left\| \frac{\partial f}{\partial \boldsymbol{u}}(\boldsymbol{u}(\boldsymbol{\mu}), \boldsymbol{\mu})^{T} \right\| \leq \kappa.$$

□

Lemma B.3. *Assume (AR1), (AR4), (AR5) hold and $U \subseteq \mathbb{R}^{N_u}$, $V \subseteq \mathbb{R}^{N_\mu}$ are bounded subsets. Then there exists a constant $\kappa > 0$ such that*

$$\|\boldsymbol{u}(\boldsymbol{\mu}) - \boldsymbol{u}\| \leq \kappa \, \|\boldsymbol{r}(\boldsymbol{u},\, \boldsymbol{\mu})\| \qquad \boldsymbol{\mu} \in V \tag{B.10}$$

for any $\boldsymbol{u} \in U$, where $\boldsymbol{u}(\boldsymbol{\mu})$ is the solution of $\boldsymbol{r}(\,\cdot\,,\, \boldsymbol{\mu}) = 0$.

Proof. Consider any $\boldsymbol{u} \in U$. A variant of the mean value theorem gives

$$\boldsymbol{r}(\boldsymbol{u}(\boldsymbol{\mu}),\, \boldsymbol{\mu}) - \boldsymbol{r}(\boldsymbol{u},\, \boldsymbol{\mu}) = \boldsymbol{D}(\boldsymbol{u}(\boldsymbol{\mu}),\, \boldsymbol{u},\, \boldsymbol{\mu})(\boldsymbol{u}(\boldsymbol{\mu}) - \boldsymbol{u}).$$

The boundedness of U^* and assumptions (AR4)–(AR5) imply the existence of a constant $\kappa > 0$ such that

$$\|\boldsymbol{u}(\boldsymbol{\mu}) - \boldsymbol{u}\| \leq \kappa \, \|\boldsymbol{r}(\boldsymbol{u},\, \boldsymbol{\mu})\| \,.$$

□

Lemma B.4. *Assume (AR1), (AR4), (AR5), (AQ2) hold and $U \subseteq \mathbb{R}^{N_u}$, $V \subseteq \mathbb{R}^{N_\mu}$ are bounded subsets. Then there exists a constant $\kappa > 0$ such that*

$$|f(\boldsymbol{u}(\boldsymbol{\mu}), \boldsymbol{\mu}) - f(\boldsymbol{u}, \boldsymbol{\mu})| \leq \kappa \left\|\boldsymbol{r}(\boldsymbol{u}, \boldsymbol{\mu})\right\| \qquad \boldsymbol{\mu} \in V \tag{B.11}$$

for any $\boldsymbol{u} \in U$, where $\boldsymbol{u}(\boldsymbol{\mu})$ is the solution of $\boldsymbol{r}(\,\cdot\,, \boldsymbol{\mu}) = 0$.

Proof. Consider any $\boldsymbol{u} \in U$. Lipschitz continuity of f (AQ2) gives

$$|f(\boldsymbol{u}(\boldsymbol{\mu}), \boldsymbol{\mu}) - f(\boldsymbol{u}, \boldsymbol{\mu})| \leq c_f \left\|\boldsymbol{u}(\boldsymbol{\mu}) - \boldsymbol{u}\right\|. \tag{B.12}$$

The bound in Lemma B.3 leads to the desired result. □

Lemma B.5. *Assume (AR1)–(AR8) hold and $U \subseteq \mathbb{R}^{N_u}$, $V \subseteq \mathbb{R}^{N_\mu}$, $W \subseteq \mathbb{R}^{N_u \times N_\mu}$ are bounded subsets. Then there exists constants κ, $\tau > 0$ such that*

$$\left\| \frac{\partial \boldsymbol{u}}{\partial \boldsymbol{\mu}}(\boldsymbol{\mu}) - \boldsymbol{w} \right\| \leq \kappa \left\|\boldsymbol{r}(\boldsymbol{u}, \boldsymbol{\mu})\right\| + \tau \left\|\boldsymbol{r}^{\partial}(\boldsymbol{u}, \boldsymbol{w}, \boldsymbol{\mu})\right\| \qquad \boldsymbol{\mu} \in V \tag{B.13}$$

for any $\boldsymbol{u} \in U$ and $\boldsymbol{w} \in W$, where $\dfrac{\partial \boldsymbol{u}}{\partial \boldsymbol{\mu}}(\boldsymbol{\mu})$ is the solution of $\boldsymbol{r}^{\partial}(\boldsymbol{u}(\boldsymbol{\mu}), \cdot\,, \boldsymbol{\mu}) = 0$ and $\boldsymbol{u}(\boldsymbol{\mu})$ is the solution of $\boldsymbol{r}(\,\cdot\,, \boldsymbol{\mu}) = 0$.

Proof. From the definition of $\dfrac{\partial \boldsymbol{u}}{\partial \boldsymbol{\mu}}(\boldsymbol{\mu})$ in (2.87) and $\boldsymbol{r}^{\partial}$ in (2.89), the following relation holds for any $\boldsymbol{w} \in \mathbb{R}^{N_u \times N_\mu}$

$$\begin{aligned} \frac{\partial \boldsymbol{u}}{\partial \boldsymbol{\mu}}(\boldsymbol{\mu}) - \boldsymbol{w} &= -\frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}(\boldsymbol{u}(\boldsymbol{\mu}), \boldsymbol{\mu})^{-1} \left[\frac{\partial \boldsymbol{r}}{\partial \boldsymbol{\mu}}(\boldsymbol{u}(\boldsymbol{\mu}), \boldsymbol{\mu}) + \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}(\boldsymbol{u}(\boldsymbol{\mu}), \boldsymbol{\mu}) \boldsymbol{w} \right] \\ &= -\frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}(\boldsymbol{u}(\boldsymbol{\mu}), \boldsymbol{\mu})^{-1} \boldsymbol{r}^{\partial}(\boldsymbol{u}(\boldsymbol{\mu}), \boldsymbol{w}, \boldsymbol{\mu}). \end{aligned} \tag{B.14}$$

Existence and boundedness of the Jacobian inverse over U^* leads to the bound

$$\left\| \frac{\partial \boldsymbol{u}}{\partial \boldsymbol{\mu}}(\boldsymbol{\mu}) - \boldsymbol{w} \right\| \leq \kappa_1 \left\|\boldsymbol{r}^{\partial}(\boldsymbol{u}(\boldsymbol{\mu}), \boldsymbol{w}, \boldsymbol{\mu})\right\| \tag{B.15}$$

for any $\boldsymbol{w} \in \mathbb{R}^{N_u \times N_\mu}$, where $\kappa_1 > 0$ is a constant. The desired bound will be obtained by bounding the sensitivity residual evaluated at the *exact* primal solution, $\boldsymbol{r}^{\partial}(\boldsymbol{u}(\boldsymbol{\mu}), \boldsymbol{w}, \boldsymbol{\mu})$, by a combination of the primal and sensitivity residuals at an approximate primal and sensitivity solution. For $\boldsymbol{w} \in W$

there exists a constant $\kappa_2 > 0$ such that

$$\begin{aligned}\left\|\boldsymbol{r}^{\partial}(\boldsymbol{u}(\boldsymbol{\mu}),\, \boldsymbol{w},\, \boldsymbol{\mu})\right\| &\leq \left\|\boldsymbol{r}^{\partial}(\boldsymbol{u},\, \boldsymbol{w},\, \boldsymbol{\mu})\right\| + \left\|\boldsymbol{r}^{\partial}(\boldsymbol{u}(\boldsymbol{\mu}),\, \boldsymbol{w},\, \boldsymbol{\mu}) - \boldsymbol{r}^{\partial}(\boldsymbol{u},\, \boldsymbol{w},\, \boldsymbol{\mu})\right\| \\ &\leq \left\|\boldsymbol{r}^{\partial}(\boldsymbol{u},\, \boldsymbol{w},\, \boldsymbol{\mu})\right\| + \left\|\frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}(\boldsymbol{u}(\boldsymbol{\mu}),\, \boldsymbol{\mu}) - \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}(\boldsymbol{u},\, \boldsymbol{\mu})\right\| \|\boldsymbol{w}\| + \left\|\frac{\partial \boldsymbol{r}}{\partial \boldsymbol{\mu}}(\boldsymbol{u}(\boldsymbol{\mu}),\, \boldsymbol{\mu}) - \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{\mu}}(\boldsymbol{u},\, \boldsymbol{\mu})\right\| \\ &\leq \left\|\boldsymbol{r}^{\partial}(\boldsymbol{u},\, \boldsymbol{w},\, \boldsymbol{\mu})\right\| + \left(c_{\partial_{\boldsymbol{\mu}}\boldsymbol{r}} + c_{\partial_{\boldsymbol{u}}\boldsymbol{r}} \|\boldsymbol{w}\|\right) \|\boldsymbol{u}(\boldsymbol{\mu}) - \boldsymbol{u}\| \\ &\leq \left\|\boldsymbol{r}^{\partial}(\boldsymbol{u},\, \boldsymbol{w},\, \boldsymbol{\mu})\right\| + \kappa_2 \|\boldsymbol{r}(\boldsymbol{u},\, \boldsymbol{\mu})\|\,.\end{aligned} \tag{B.16}$$

The first two inequalities follow from the triangle inequality and definition of the sensitivity residual $\boldsymbol{r}^{\partial}$ in (2.89). The third inequality follows from Lipschitz continuity of the partial derivatives of $\boldsymbol{r}$ (AR7)–(AR8). The final inequality follows from the boundedness of the subset W and Lemma B.3. Combining (B.15) and (B.16), the desired result follows. □

Lemma B.6. *Assume (AR1)–(AR7), (AQ1), (AQ3) hold and $U \subseteq \mathbb{R}^{N_{\boldsymbol{u}}}$, $V \subseteq \mathbb{R}^{N_{\boldsymbol{\mu}}}$, $Z \subseteq \mathbb{R}^{N_{\boldsymbol{u}}}$. Then there exists constants κ, $\tau > 0$ such that*

$$\|\boldsymbol{\lambda}(\boldsymbol{\mu}) - \boldsymbol{z}\| \leq \kappa \|\boldsymbol{r}(\boldsymbol{u},\, \boldsymbol{\mu})\| + \tau \left\|\boldsymbol{r}^{\boldsymbol{\lambda}}(\boldsymbol{u},\, \boldsymbol{z},\, \boldsymbol{\mu})\right\| \qquad \boldsymbol{\mu} \in V \tag{B.17}$$

for any $\boldsymbol{z} \in Z$, where $\boldsymbol{\lambda}(\boldsymbol{\mu})$ is the solution of $\boldsymbol{r}^{\boldsymbol{\lambda}}(\boldsymbol{u}(\boldsymbol{\mu}), \cdot\,, \boldsymbol{\mu}) = 0$ and $\boldsymbol{u}(\boldsymbol{\mu})$ is the solution of $\boldsymbol{r}(\cdot\,, \boldsymbol{\mu}) = 0$.

Proof. From the definition of $\boldsymbol{\lambda}(\boldsymbol{\mu})$ in (2.92) and $\boldsymbol{r}^{\boldsymbol{\lambda}}$ in (2.101), the following relation holds for any $\boldsymbol{z} \in \mathbb{R}^{N_{\boldsymbol{u}}}$

$$\begin{aligned}\boldsymbol{\lambda}(\boldsymbol{\mu}) - \boldsymbol{z} &= \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}(\boldsymbol{u}(\boldsymbol{\mu}),\, \boldsymbol{\mu})^{-T} \left[-\frac{\partial f}{\partial \boldsymbol{u}}(\boldsymbol{u}(\boldsymbol{\mu}),\, \boldsymbol{\mu})^T + \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}(\boldsymbol{u}(\boldsymbol{\mu}),\, \boldsymbol{\mu})^T \boldsymbol{z}\right] \\ &= \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}(\boldsymbol{u}(\boldsymbol{\mu}),\, \boldsymbol{\mu})^{-T} \boldsymbol{r}^{\boldsymbol{\lambda}}(\boldsymbol{u}(\boldsymbol{\mu}),\, \boldsymbol{z},\, \boldsymbol{\mu}).\end{aligned} \tag{B.18}$$

Existence and boundedness of the Jacobian inverse over U^* leads to the bound

$$\|\boldsymbol{\lambda}(\boldsymbol{\mu}) - \boldsymbol{z}\| \leq \kappa_1 \left\|\boldsymbol{r}^{\boldsymbol{\lambda}}(\boldsymbol{u}(\boldsymbol{\mu}),\, \boldsymbol{z},\, \boldsymbol{\mu})\right\| \tag{B.19}$$

for any $\boldsymbol{z} \in \mathbb{R}^{N_{\boldsymbol{u}}}$, where $\kappa_1 > 0$ is a constant. The desired bound will be obtained by bounding the adjoint residual evaluated at the *exact* primal solution, $\boldsymbol{r}^{\boldsymbol{\lambda}}(\boldsymbol{u}(\boldsymbol{\mu}),\, \boldsymbol{z},\, \boldsymbol{\mu})$, by a combination of the primal and adjoint residuals at an approximate primal and adjoint solution. Then for $\boldsymbol{z} \in Z$, there exists a constant $\kappa_2 > 0$ such that

$$\begin{aligned}\left\|\boldsymbol{r}^{\boldsymbol{\lambda}}(\boldsymbol{u}(\boldsymbol{\mu}),\, \boldsymbol{z},\, \boldsymbol{\mu})\right\| &\leq \left\|\boldsymbol{r}^{\boldsymbol{\lambda}}(\boldsymbol{u},\, \boldsymbol{z},\, \boldsymbol{\mu})\right\| + \left\|\boldsymbol{r}^{\boldsymbol{\lambda}}(\boldsymbol{u}(\boldsymbol{\mu}),\, \boldsymbol{z},\, \boldsymbol{\mu}) - \boldsymbol{r}^{\boldsymbol{\lambda}}(\boldsymbol{u},\, \boldsymbol{z},\, \boldsymbol{\mu})\right\| \\ &\leq \left\|\boldsymbol{r}^{\boldsymbol{\lambda}}(\boldsymbol{u},\, \boldsymbol{z},\, \boldsymbol{\mu})\right\| + \left\|\frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}(\boldsymbol{u}(\boldsymbol{\mu}),\, \boldsymbol{\mu})^T - \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}(\boldsymbol{u},\, \boldsymbol{\mu})^T\right\| \|\boldsymbol{z}\| + \left\|\frac{\partial f}{\partial \boldsymbol{u}}(\boldsymbol{u}(\boldsymbol{\mu}),\, \boldsymbol{\mu}) - \frac{\partial f}{\partial \boldsymbol{u}}(\boldsymbol{u},\, \boldsymbol{\mu})\right\| \\ &\leq \left\|\boldsymbol{r}^{\boldsymbol{\lambda}}(\boldsymbol{u},\, \boldsymbol{z},\, \boldsymbol{\mu})\right\| + \left(c_{\partial_{\boldsymbol{u}} f} + c_{\partial_{\boldsymbol{u}}\boldsymbol{r}} \|\boldsymbol{z}\|\right) \|\boldsymbol{u}(\boldsymbol{\mu}) - \boldsymbol{u}\| \\ &\leq \left\|\boldsymbol{r}^{\boldsymbol{\lambda}}(\boldsymbol{u},\, \boldsymbol{z},\, \boldsymbol{\mu})\right\| + \kappa_2 \|\boldsymbol{r}(\boldsymbol{u},\, \boldsymbol{\mu})\|\,.\end{aligned} \tag{B.20}$$

The first two inequalities follow from the triangle inequality and definition of the adjoint residual $\boldsymbol{r}^{\boldsymbol{\lambda}}$ in (2.101). The third inequality follows from Lipschitz continuity of the Jacobian of $\boldsymbol{r}$ (AR7) and f (AQ3). The final inequality follows from the boundedness of the set Z. Combining (B.18) and (B.20), the desired result follows. □

Lemma B.7. *Assume (AR1)–(AR8), (AQ1)–(AQ4) hold and $U \subseteq \mathbb{R}^{N_{\boldsymbol{u}}}$, $V \subseteq \mathbb{R}^{N_{\boldsymbol{\mu}}}$, $W \subseteq \mathbb{R}^{N_{\boldsymbol{u}} \times N_{\boldsymbol{\mu}}}$ are bounded subsets. Then there exists constant $\kappa, \tau > 0$ such that*

$$\left\| \boldsymbol{g}^{\partial}\left(\boldsymbol{u}(\boldsymbol{\mu}), \frac{\partial \boldsymbol{u}}{\partial \boldsymbol{\mu}}(\boldsymbol{\mu}), \boldsymbol{\mu}\right) - \boldsymbol{g}^{\partial}(\boldsymbol{u}, \boldsymbol{w}, \boldsymbol{\mu}) \right\| \leq \kappa \left\| \boldsymbol{r}(\boldsymbol{u}, \boldsymbol{\mu}) \right\| + \tau \left\| \boldsymbol{r}^{\partial}(\boldsymbol{u}, \boldsymbol{w}, \boldsymbol{\mu}) \right\| \qquad \boldsymbol{\mu} \in V \tag{B.21}$$

for any $\boldsymbol{u} \in U$, $\boldsymbol{w} \in W$, where $\boldsymbol{u}(\boldsymbol{\mu})$ is the solution of $\boldsymbol{r}(\,\cdot\,, \boldsymbol{\mu}) = 0$ and $\dfrac{\partial \boldsymbol{u}}{\partial \boldsymbol{\mu}}(\boldsymbol{\mu})$ is the solution of $\boldsymbol{r}^{\partial}(\boldsymbol{u}(\boldsymbol{\mu}), \cdot\,, \boldsymbol{\mu}) = 0$.

Proof. From the definition of $\boldsymbol{g}^{\partial}$ in (2.90) and the triangle inequality

$$\begin{aligned} \left\| \boldsymbol{g}^{\partial}\left(\boldsymbol{u}(\boldsymbol{\mu}), \frac{\partial \boldsymbol{u}}{\partial \boldsymbol{\mu}}(\boldsymbol{\mu}), \boldsymbol{\mu}\right) - \boldsymbol{g}^{\partial}(\boldsymbol{u}, \boldsymbol{w}, \boldsymbol{\mu}) \right\| \leq & \left\| \frac{\partial f}{\partial \boldsymbol{\mu}}(\boldsymbol{u}(\boldsymbol{\mu}), \boldsymbol{\mu}) - \frac{\partial f}{\partial \boldsymbol{\mu}}(\boldsymbol{u}, \boldsymbol{\mu}) \right\| + \\ & \left\| \frac{\partial f}{\partial \boldsymbol{u}}(\boldsymbol{u}(\boldsymbol{\mu}), \boldsymbol{\mu}) \frac{\partial \boldsymbol{u}}{\partial \boldsymbol{\mu}}(\boldsymbol{\mu}) - \frac{\partial f}{\partial \boldsymbol{u}}(\boldsymbol{u}, \boldsymbol{\mu}) \boldsymbol{w} \right\|. \end{aligned} \tag{B.22}$$

for any $\boldsymbol{u} \in U$ and $\boldsymbol{w} \in W$. Lipschitz continuity of $\dfrac{\partial f}{\partial \boldsymbol{\mu}}(\,\cdot\,, \boldsymbol{\mu})$ leads to

$$\begin{aligned} \left\| \boldsymbol{g}^{\partial}\left(\boldsymbol{u}(\boldsymbol{\mu}), \frac{\partial \boldsymbol{u}}{\partial \boldsymbol{\mu}}(\boldsymbol{\mu}), \boldsymbol{\mu}\right) - \boldsymbol{g}^{\partial}(\boldsymbol{u}, \boldsymbol{w}, \boldsymbol{\mu}) \right\| \leq & c_{\partial_{\boldsymbol{\mu}} f} \left\| \boldsymbol{u}(\boldsymbol{\mu}) - \boldsymbol{u} \right\| + \\ & \left\| \frac{\partial f}{\partial \boldsymbol{u}}(\boldsymbol{u}(\boldsymbol{\mu}), \boldsymbol{\mu}) \frac{\partial \boldsymbol{u}}{\partial \boldsymbol{\mu}}(\boldsymbol{\mu}) - \frac{\partial f}{\partial \boldsymbol{u}}(\boldsymbol{u}, \boldsymbol{\mu}) \boldsymbol{w} \right\|. \end{aligned} \tag{B.23}$$

Adding and subtracting $\dfrac{\partial f}{\partial \boldsymbol{u}}(\boldsymbol{u}, \boldsymbol{\mu}) \dfrac{\partial \boldsymbol{u}}{\partial \boldsymbol{\mu}}(\boldsymbol{\mu})$ leads to

$$\begin{aligned} \left\| \boldsymbol{g}^{\partial}\left(\boldsymbol{u}(\boldsymbol{\mu}), \frac{\partial \boldsymbol{u}}{\partial \boldsymbol{\mu}}(\boldsymbol{\mu}), \boldsymbol{\mu}\right) - \boldsymbol{g}^{\partial}(\boldsymbol{u}, \boldsymbol{w}, \boldsymbol{\mu}) \right\| \leq & c_{\partial_{\boldsymbol{\mu}} f} \left\| \boldsymbol{u}(\boldsymbol{\mu}) - \boldsymbol{u} \right\| + \\ & \left\| \left(\frac{\partial f}{\partial \boldsymbol{u}}(\boldsymbol{u}(\boldsymbol{\mu}), \boldsymbol{\mu}) - \frac{\partial f}{\partial \boldsymbol{u}}(\boldsymbol{u}, \boldsymbol{\mu}) \right) \frac{\partial \boldsymbol{u}}{\partial \boldsymbol{\mu}}(\boldsymbol{\mu}) \right\| + \\ & \left\| \frac{\partial f}{\partial \boldsymbol{u}}(\boldsymbol{u}, \boldsymbol{\mu}) \left(\frac{\partial \boldsymbol{u}}{\partial \boldsymbol{\mu}}(\boldsymbol{\mu}) - \boldsymbol{w} \right) \right\|. \end{aligned}$$

Lipschitz continuity and boundedness of $\dfrac{\partial f}{\partial \boldsymbol{u}}(\,\cdot\,, \boldsymbol{\mu})$ gives

$$\left\| \boldsymbol{g}^{\partial}\left(\boldsymbol{u}(\boldsymbol{\mu}), \frac{\partial \boldsymbol{u}}{\partial \boldsymbol{\mu}}(\boldsymbol{\mu}), \boldsymbol{\mu}\right) - \boldsymbol{g}^{\partial}(\boldsymbol{u}, \boldsymbol{w}, \boldsymbol{\mu}) \right\| \leq \left(c_{\partial_{\boldsymbol{\mu}} f} + c_{\partial_{\boldsymbol{u}} f} \left\| \frac{\partial \boldsymbol{u}}{\partial \boldsymbol{\mu}}(\boldsymbol{\mu}) \right\| \right) \left\| \boldsymbol{u}(\boldsymbol{\mu}) - \boldsymbol{u} \right\| + \tau_1 \left\| \frac{\partial \boldsymbol{u}}{\partial \boldsymbol{\mu}}(\boldsymbol{\mu}) - \boldsymbol{v} \right\|,$$

where $\tau_1 > 0$ is a constant. The boundedness of W^* leads to

$$\left\| \boldsymbol{g}^{\partial} \left(\boldsymbol{u}(\boldsymbol{\mu}), \frac{\partial \boldsymbol{u}}{\partial \boldsymbol{\mu}}(\boldsymbol{\mu}), \boldsymbol{\mu} \right) - \boldsymbol{g}^{\partial}(\boldsymbol{u}, \boldsymbol{w}, \boldsymbol{\mu}) \right\| \leq \kappa_1 \left\| \boldsymbol{u}(\boldsymbol{\mu}) - \boldsymbol{u} \right\| + \tau_1 \left\| \frac{\partial \boldsymbol{u}}{\partial \boldsymbol{\mu}}(\boldsymbol{\mu}) - \boldsymbol{w} \right\|$$

where $\kappa_1 > 0$ is a constant. Combining the above bound with Lemmas B.3 and B.5 gives the desired result. □

Lemma B.8. *Assume (AR1)–(AR8), (AQ1)–(AQ4) hold and $U \subseteq \mathbb{R}^{N_{\boldsymbol{u}}}$, $V \subseteq \mathbb{R}^{N_{\boldsymbol{\mu}}}$, $Z \subseteq \mathbb{R}^{N_{\boldsymbol{u}}}$ are bounded subsets. Then there exists constant $\kappa, \tau > 0$ such that*

$$\left\| \boldsymbol{g}^{\boldsymbol{\lambda}} \left(\boldsymbol{u}(\boldsymbol{\mu}), \boldsymbol{\lambda}(\boldsymbol{\mu}), \boldsymbol{\mu} \right) - \boldsymbol{g}^{\boldsymbol{\lambda}}(\boldsymbol{u}, \boldsymbol{z}, \boldsymbol{\mu}) \right\| \leq \kappa \left\| \boldsymbol{r}(\boldsymbol{u}, \boldsymbol{\mu}) \right\| + \tau \left\| \boldsymbol{r}^{\boldsymbol{\lambda}} \left(\boldsymbol{u}, \boldsymbol{z}, \boldsymbol{\mu} \right) \right\| \qquad \boldsymbol{\mu} \in V \tag{B.24}$$

for any $\boldsymbol{u} \in U$, $\boldsymbol{z} \in Z$, where $\boldsymbol{u}(\boldsymbol{\mu})$ is the solution of $\boldsymbol{r}(\,\cdot\,, \boldsymbol{\mu}) = 0$ and $\boldsymbol{\lambda}(\boldsymbol{\mu})$ is the solution of $\boldsymbol{r}^{\boldsymbol{\lambda}}(\boldsymbol{u}(\boldsymbol{\mu}), \cdot\,, \boldsymbol{\mu}) = 0$.

Proof. From the definition of $\boldsymbol{g}^{\boldsymbol{\lambda}}$ in (2.102) and the triangle inequality

$$\begin{aligned} \left\| \boldsymbol{g}^{\boldsymbol{\lambda}} \left(\boldsymbol{u}(\boldsymbol{\mu}), \boldsymbol{\lambda}(\boldsymbol{\mu}), \boldsymbol{\mu} \right) - \boldsymbol{g}^{\boldsymbol{\lambda}}(\boldsymbol{u}, \boldsymbol{z}, \boldsymbol{\mu}) \right\| \leq & \left\| \frac{\partial f}{\partial \boldsymbol{\mu}}(\boldsymbol{u}(\boldsymbol{\mu}), \boldsymbol{\mu}) - \frac{\partial f}{\partial \boldsymbol{\mu}}(\boldsymbol{u}, \boldsymbol{\mu}) \right\| + \\ & \left\| \boldsymbol{\lambda}(\boldsymbol{\mu})^T \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{\mu}}(\boldsymbol{u}(\boldsymbol{\mu}), \boldsymbol{\mu}) - \boldsymbol{z}^T \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{\mu}}(\boldsymbol{u}, \boldsymbol{\mu}) \right\|. \end{aligned} \tag{B.25}$$

for any $\boldsymbol{u} \in U$ and $\boldsymbol{z} \in Z$. Lipschitz continuity of $\dfrac{\partial f}{\partial \boldsymbol{\mu}}(\,\cdot\,, \boldsymbol{\mu})$ over U leads to

$$\begin{aligned} \left\| \boldsymbol{g}^{\boldsymbol{\lambda}} \left(\boldsymbol{u}(\boldsymbol{\mu}), \boldsymbol{\lambda}(\boldsymbol{\mu}), \boldsymbol{\mu} \right) - \boldsymbol{g}^{\boldsymbol{\lambda}}(\boldsymbol{u}, \boldsymbol{z}, \boldsymbol{\mu}) \right\| \leq & \, c_{\partial_{\boldsymbol{\mu}} f} \left\| \boldsymbol{u}(\boldsymbol{\mu}) - \boldsymbol{u} \right\| + \\ & \left\| \boldsymbol{\lambda}(\boldsymbol{\mu})^T \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{\mu}}(\boldsymbol{u}(\boldsymbol{\mu}), \boldsymbol{\mu}) - \boldsymbol{z}^T \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{\mu}}(\boldsymbol{u}, \boldsymbol{\mu}) \right\|. \end{aligned} \tag{B.26}$$

Adding and subtracting $\boldsymbol{\lambda}(\boldsymbol{\mu})^T \dfrac{\partial \boldsymbol{r}}{\partial \boldsymbol{\mu}}(\boldsymbol{u}, \boldsymbol{\mu})$ leads to

$$\begin{aligned} \left\| \boldsymbol{g}^{\partial} \left(\boldsymbol{u}(\boldsymbol{\mu}), \boldsymbol{\lambda}(\boldsymbol{\mu}), \boldsymbol{\mu} \right) - \boldsymbol{g}^{\partial}(\boldsymbol{u}, \boldsymbol{z}, \boldsymbol{\mu}) \right\| \leq & \, c_{\partial_{\boldsymbol{\mu}} f} \left\| \boldsymbol{u}(\boldsymbol{\mu}) - \boldsymbol{u} \right\| + \\ & \left\| \boldsymbol{\lambda}(\boldsymbol{\mu})^T \left(\frac{\partial \boldsymbol{r}}{\partial \boldsymbol{\mu}}(\boldsymbol{u}(\boldsymbol{\mu}), \boldsymbol{\mu}) - \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{\mu}}(\boldsymbol{u}, \boldsymbol{\mu}) \right) \right\| + \\ & \left\| (\boldsymbol{\lambda}(\boldsymbol{\mu}) - \boldsymbol{z})^T \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{\mu}}(\boldsymbol{u}, \boldsymbol{\mu}) \right\|. \end{aligned}$$

Lipschitz continuity and boundedness of $\dfrac{\partial \boldsymbol{r}}{\partial \boldsymbol{\mu}}(\,\cdot\,, \boldsymbol{\mu})$ gives

$$\left\| \boldsymbol{g}^{\partial} \left(\boldsymbol{u}(\boldsymbol{\mu}), \boldsymbol{\lambda}(\boldsymbol{\mu}), \boldsymbol{\mu} \right) - \boldsymbol{g}^{\partial}(\boldsymbol{u}, \boldsymbol{z}, \boldsymbol{\mu}) \right\| \leq \left(c_{\partial_{\boldsymbol{\mu}} f} + c_{\partial_{\boldsymbol{\mu}} \boldsymbol{r}} \left\| \boldsymbol{\lambda}(\boldsymbol{\mu}) \right\| \right) \left\| \boldsymbol{u}(\boldsymbol{\mu}) - \boldsymbol{u} \right\| + \tau_1 \left\| \boldsymbol{\lambda}(\boldsymbol{\mu}) - \boldsymbol{z} \right\|,$$

where $\tau_1 > 0$ is a constant. The boundedness of Z^* leads to

$$\left\| \boldsymbol{g}^{\partial} \left(\boldsymbol{u}(\boldsymbol{\mu}),\, \boldsymbol{\lambda}(\boldsymbol{\mu}),\, \boldsymbol{\mu} \right) - \boldsymbol{g}^{\partial}(\boldsymbol{u}, \boldsymbol{z}, \boldsymbol{\mu}) \right\| \leq \kappa_1 \left\| \boldsymbol{u}(\boldsymbol{\mu}) - \boldsymbol{u} \right\| + \tau_1 \left\| \boldsymbol{\lambda}(\boldsymbol{\mu}) - \boldsymbol{z} \right\|$$

where $\kappa_1 > 0$ is a constant. Combining the above bound with Lemmas B.3 and B.6 gives the desired result. □

Appendix C

Adaptive State and Parameter Space Reduction for Large-Scale Optimization

All of the optimization methods introduced in Chapters 5–6 were developed under the assumption that the number of optimization parameters is small compared to the size of the state vector ($N_{\boldsymbol{\mu}} \ll N_{\boldsymbol{u}}$) and therefore the dominant cost is attributed to the PDE solves. However, there are a large a number of relevant optimization problems, including topological optimization and inverse problems, where this is not the case. In these problems, the number of parameters is of the same order of magnitude as the number of degrees of freedom in the PDE, i.e., $N_{\boldsymbol{\mu}} = \mathcal{O}(N_{\boldsymbol{u}})$. In such settings, the cost of the optimization problem cannot be notably decreased if the state vector alone is reduced, e.g., with projection-based reduced-order models. This can be attributed to two main sources of computational cost. The first comes from the fact that the linear algebra involved in the optimization solver is non-negligible due to the large number of parameters. Therefore, even the reduced-space approach to PDE-constrained optimization (Section 2.3.2) will yield a *large-scale* optimization problem. Second, the evaluation of reduced-order model residual and Jacobian depend on $\mathcal{O}(N_{\boldsymbol{u}})$ parameters and will require *at least* $\mathcal{O}(N_{\boldsymbol{u}})$ operations and can not be expected to enjoy the dramatic reduction in computational resources that has been exploited in non-parametric or few-parameter settings [17, 171, 114, 125, 52].

The approach taken to eliminate the bottlenecks associated with large parameter spaces adaptively restricts the parameter space to a low-dimensional affine subspace of dimension $k_{\boldsymbol{\mu}}$, where $k_{\boldsymbol{\mu}} \ll N_{\boldsymbol{\mu}}$. While similar approaches have been taken in the past [168, 167, 117, 120], the proposed method focuses on establishing global convergence (not considered in [117, 120]) without requiring first-order consistency, a requirement in [168, 167], for increased efficiency. The proposed restriction converts the $N_{\boldsymbol{\mu}}$-parameter optimization problem to one in $k_{\boldsymbol{\mu}}$ parameters. The resulting optimization problem with few parameters is solved using the globally convergent multifidelity trust

region method that leverages projection-based reduced-order models (Chapter 5). This results in a *two-level, nested reduction* where, at the outermost level, the parameter space is restricted to a low-dimensional affine subspace to yield an optimization problem in few variables and, at the inner level, the projection-based model reduction reduces the dimensionality of the PDE itself. The inexactness introduced at the innermost level through the use of projection-based reduced-order models is managed using the multifidelity trust region of Chapter 5. Once the solution of the restricted optimization problem is found, the low-dimensional parameter subspace is adapted at the new point in $\boldsymbol{\mu}$-space. Such an approach to numerical optimization is called a *subspace method* [54, 119, 137, 143, 207]; the popular *linesearch* methods [143] correspond to the special case with $k_{\boldsymbol{\mu}} = 1$. Convergence theory from the subspace/linesearch optimization literature will be recycled to formulate a minimum requirement on the updated low-dimensional affine parameter subspace to ensure a globally convergent method. The proposed subspace update will satisfy this minimum requirement, thereby ensuring global convergence, while providing sufficient flexibility to incorporate generic optimization-based vectors (such as the steepest descent direction, quasi-Newton directions, and directions of negative curvature) as well as problem specific information. In applications such as topology optimization and inverse problems, the parameter vector has a strong connection to the geometry of the underlying PDE and its discretization, which can be exploited to yield a rapidly converging algorithm.

General subspace methods ($k_{\boldsymbol{\mu}} > 1$) have not been widely adopted by the optimization community because of the inherent difficulty/expense required to search a $k_{\boldsymbol{\mu}}$-dimensional subspace compared to a one-dimensional subspace as in linesearch methods. This is one reason linesearch methods have enjoyed considerable success. In contrast, trust region methods search the entire $N_{\boldsymbol{\mu}}$-dimensional space at each optimization iteration; however, the expensive objective function is usually replaced with a quadratic approximation that is inexpensive to query. The use of the more expensive subspace methods are justified in this work for two reasons. First, an efficient method has been developed in Chapter 5 to solve PDE-constrained optimization problems with few parameters and it is desirable to use this method to do as much work as possible before adapting the parameter space. Additionally, restricting the parameter space to few parameters will ensure evaluation of the reduced-order model does not involve operations that scale with $N_{\boldsymbol{\mu}} = \mathcal{O}(N_{\boldsymbol{u}})$. To develop the ideas of this section in a simple setting, only the deterministic case will be considered; future work will consider stochastic optimization problems with large-dimensional parameter spaces and a strategy that combines this approach with the method of Chapter 6, i.e., three-level approximation: reduction of the state space via model reduction, reduction of the parameter space via subspace and linesearch techniques, and approximation of integrals using dimension-adaptive sparse grids.

C.1 Two-Level Nested Reduction of Parametrized Partial Differential Equations

This section proposes a two-level, nested reduction strategy for parametrized partial differential equations. In the first level of reduction, the high-dimensional parameter space is restricted to a low-dimensional affine subspace. In the context of optimization, this amounts to a restriction of the search space to the chosen affine subspace; however, it does not introduce any error into the pointwise evaluation of the PDE. The second level of reduction uses projection-based model reduction (Chapter 4) to the reduce the number of degrees of freedom in the PDE, i.e., reduction of the state space. Unlike the reduction of the parameter space, the state space restriction *does* introduce error into the evaluation of the PDE, as seen previously in Chapter 4. These two types of reduction will be nested to efficiently solve a PDE-constrained optimization problem as follows: first, the parameter space restriction will be applied to reduce the optimization problem over $N_{\boldsymbol{\mu}}$ parameters to one over $k_{\boldsymbol{\mu}} \ll N_{\boldsymbol{\mu}}$ parameters and the trust region method of Chapter 5 that leverages projection-based reduced-order models will be applied to solve the reduced optimization problem. Adaptation of the parameter space, discussed in Section C.2.1, will be required to yield a globally convergent method. The remainder of this section will consider each layer of reduction, in isolation, which will be combined in Section C.2 to develop the nested optimization algorithm.

C.1.1 Outer Layer of Reduction: Restriction of Parameter Space

The PDE-constrained optimization problem that motivates this work takes the form (reduced-space formulation)

$$\underset{\boldsymbol{\mu}\in\mathbb{R}^{N_{\boldsymbol{\mu}}}}{\text{minimize}} \quad F(\boldsymbol{\mu}) \coloneqq f(\boldsymbol{u}(\boldsymbol{\mu}), \boldsymbol{\mu}) \tag{C.1}$$

where $\boldsymbol{u}(\boldsymbol{\mu})$ is the unique (Assumption 2.2), continuously differentiable (Theorem 2.1) solution of the fully discrete parametrized partial differential equation $\boldsymbol{r}(\,\cdot\,, \boldsymbol{\mu}) = 0$. Unlike previous chapters, here it is assumed that $N_{\boldsymbol{\mu}}$ is large, i.e., $N_{\boldsymbol{\mu}} = \mathcal{O}(N_{\boldsymbol{u}})$. The gradient of the objective function can be computed using either the sensitivity (Section 2.3.3) or adjoint (Section 2.3.4) method; however, due to the large number of parameters $N_{\boldsymbol{\mu}} = \mathcal{O}(N_{\boldsymbol{u}})$, the adjoint method is the only feasible approach.

The reduction of the parameter space proceeds in an identical manner to the state reduction in Chapter 4, i.e., with the ansatz that the parameter lies in a low-dimensional (affine) subspace

$$\boldsymbol{\mu} = \bar{\boldsymbol{\mu}} + \boldsymbol{\Upsilon}\boldsymbol{\eta} \tag{C.2}$$

where $\bar{\boldsymbol{\mu}} \in \mathbb{R}^{N_{\boldsymbol{\mu}}}$ is the affine offset, $\boldsymbol{\Upsilon} \in \mathbb{R}^{N_{\boldsymbol{\mu}} \times k_{\boldsymbol{\mu}}}$ a basis for the chosen low-dimensional subspace, $\boldsymbol{\eta} \in \mathbb{R}^{k_{\boldsymbol{\mu}}}$ are the reduced coordinates of $\boldsymbol{\mu}$ in the affine subspace $\mathcal{A}(\bar{\boldsymbol{\mu}}, \boldsymbol{\Upsilon}) \coloneqq \{\bar{\boldsymbol{\mu}} + \boldsymbol{\Upsilon}\boldsymbol{\eta} \mid \boldsymbol{\eta} \in \mathbb{R}^{k_{\boldsymbol{\mu}}}\}$, and $k_{\boldsymbol{\mu}} \ll N_{\boldsymbol{\mu}}$. For the remainder of this section, $\bar{\boldsymbol{\mu}}$ and $\boldsymbol{\Upsilon}$ will be assumed given and fixed; Section C.2 will provide details pertaining to their construction and adaptation. Substitution of the ansatz in (C.2) into the parametrized PDE $\boldsymbol{r}(\boldsymbol{u}, \boldsymbol{\mu}) = 0$ with $N_{\boldsymbol{\mu}}$ parameters leads to a parametrized PDE in

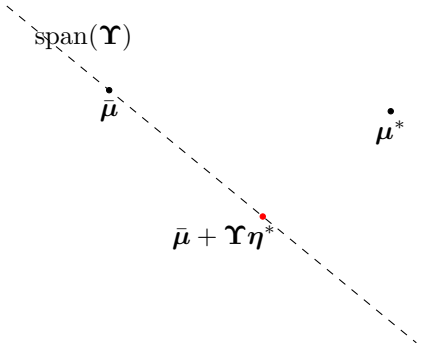


Figure C.1: Schematic of restriction of parameter space $\mathbb{R}^{N_\mu}$ to affine subspace $\mathcal{A}(\bar{\boldsymbol{\mu}}, \boldsymbol{\Upsilon})$ of dimension $k_{\boldsymbol{\mu}}$, in the special case where $N_{\boldsymbol{\mu}} = 2$ and $k_{\boldsymbol{\mu}} = 1$. The optimal solution $\boldsymbol{\mu}^*$ in the parameter space, as well as the optimal solution over $\mathcal{A}(\bar{\boldsymbol{\mu}}, \boldsymbol{\Upsilon})$ are also depicted.

$k_{\boldsymbol{\mu}}$ parameters

$$\boldsymbol{r}(\boldsymbol{u}, \bar{\boldsymbol{\mu}} + \boldsymbol{\Upsilon}\boldsymbol{\eta}) = 0. \tag{C.3}$$

In the above setting, the affine offset $\bar{\boldsymbol{\mu}}$ and basis $\boldsymbol{\Upsilon}$ are fixed and the PDE parameter is varied through variations in the reduced coordinates $\boldsymbol{\eta}$. For the remainder of this document, let $\boldsymbol{u}(\boldsymbol{\eta}; \bar{\boldsymbol{\mu}}, \boldsymbol{\Upsilon})$ be the solution of the restricted PDE in (C.3), i.e., $\boldsymbol{r}(\,\cdot\,, \bar{\boldsymbol{\mu}} + \boldsymbol{\Upsilon}\boldsymbol{\eta}) = 0$. Uniqueness and continuous differentiability of $\boldsymbol{u}(\boldsymbol{\eta}; \bar{\boldsymbol{\mu}}, \boldsymbol{\Upsilon})$ follow immediately from the corresponding properties of $\boldsymbol{u}(\boldsymbol{\mu})$ and the affine relationship between $\boldsymbol{\mu}$ and $\boldsymbol{\eta}$. Following the discussion at the beginning of this section, the approximation in (C.3) does not introduce error into the evaluation of the PDE since it is clear that $\boldsymbol{u}(\boldsymbol{\eta}; \bar{\boldsymbol{\mu}}, \boldsymbol{\Upsilon}) = \boldsymbol{u}(\bar{\boldsymbol{\mu}} + \boldsymbol{\Upsilon}\boldsymbol{\eta})$. Rather, it limits the possible variations of the parameter that can be realized, i.e., any $\boldsymbol{\mu} \in \mathbb{R}^{N_\mu}$ such that $\boldsymbol{\mu} \notin \mathcal{A}(\bar{\boldsymbol{\mu}}, \boldsymbol{\Upsilon})$ cannot be considered by the restricted PDE in (C.3). With the ansatz in (C.2), the PDE-constrained optimization problem in $N_{\boldsymbol{\mu}}$ parameters reduces to one in $k_{\boldsymbol{\mu}}$ parameters

$$\underset{\boldsymbol{\eta} \in \mathbb{R}^{k_\mu}}{\text{minimize}} \quad F(\boldsymbol{\eta}; \bar{\boldsymbol{\mu}}, \boldsymbol{\Upsilon}) \coloneqq f(\boldsymbol{u}(\boldsymbol{\eta}; \bar{\boldsymbol{\mu}}, \boldsymbol{\Upsilon}), \bar{\boldsymbol{\mu}} + \boldsymbol{\Upsilon}\boldsymbol{\eta}) \tag{C.4}$$

that amounts to a search for the optimal solution in the affine subspace $\mathcal{A}(\bar{\boldsymbol{\mu}}, \boldsymbol{\Upsilon})$, i.e., (C.4) is equivalent to

$$\underset{\boldsymbol{\mu} \in \mathcal{A}(\bar{\boldsymbol{\mu}}, \boldsymbol{\Upsilon})}{\text{minimize}} \quad F(\boldsymbol{\mu}). \tag{C.5}$$

This situation is illustrated in Figure C.1 for the case of $k_{\boldsymbol{\mu}} = 1$. In general, a local minima of (C.5), call it $\boldsymbol{\mu}^*$, will not be lie in $\mathcal{A}(\bar{\boldsymbol{\mu}}, \boldsymbol{\Upsilon})$ for an a priori selection of $\bar{\boldsymbol{\mu}}$ and $\boldsymbol{\Upsilon}$. This motivates the adaptation strategy for $\bar{\boldsymbol{\mu}}$ and $\boldsymbol{\Upsilon}$ that will be introduced in Section C.2.1.

Remark. *As previously discussed, the idea of restricting the optimization problem to a low-dimensional affine subspace generalizes linesearch methods that consider a one-dimensional affine search space. Such methods are known as* subspace methods*. In linesearch methods, the subspace is defined by any descent direction $\boldsymbol{p}_k$ (possibly the steepest descent or a quasi-Newton direction) and offset to include the current optimization iterate, $\boldsymbol{\mu}_k$, i.e., the search space is $\{\boldsymbol{\mu}_k + \alpha \boldsymbol{p}_k \mid \alpha > 0\}$. In the notation of*

this section, linesearch methods amount to the selection $\bar{\boldsymbol{\mu}} = \boldsymbol{\mu}_k$ *and* $\boldsymbol{\Upsilon} = [\boldsymbol{p}_k]$.

Since the number of parameters has been dramatically reduced, either the sensitivity or adjoint method are feasible approaches to compute the gradient of F. Following the procedure outlined in Section 2.3.3, the expression for $\nabla F(\boldsymbol{\eta};\, \bar{\boldsymbol{\mu}},\, \boldsymbol{\Upsilon})$ based on the sensitivity method is

$$\nabla F(\boldsymbol{\eta};\, \bar{\boldsymbol{\mu}},\, \boldsymbol{\Upsilon}) = \boldsymbol{g}^{\partial}\left(\boldsymbol{u}(\boldsymbol{\eta};\, \bar{\boldsymbol{\mu}},\, \boldsymbol{\Upsilon}),\, \frac{\partial \boldsymbol{u}}{\partial \boldsymbol{\eta}}(\boldsymbol{\eta};\, \bar{\boldsymbol{\mu}},\, \boldsymbol{\Upsilon}),\, \boldsymbol{\eta};\, \bar{\boldsymbol{\mu}},\, \boldsymbol{\Upsilon}\right), \tag{C.6}$$

where the definition of $\boldsymbol{g}^{\partial}$ varies slightly from that in (2.90)

$$\boldsymbol{g}^{\partial}\left(\boldsymbol{u},\, \boldsymbol{w}_r,\, \boldsymbol{\eta};\, \bar{\boldsymbol{\mu}},\, \boldsymbol{\Upsilon}\right) \coloneqq \frac{\partial f}{\partial \boldsymbol{\mu}}(\boldsymbol{u},\, \bar{\boldsymbol{\mu}} + \boldsymbol{\Upsilon}\boldsymbol{\eta})\boldsymbol{\Upsilon} + \frac{\partial f}{\partial \boldsymbol{u}}(\boldsymbol{u},\, \bar{\boldsymbol{\mu}} + \boldsymbol{\Upsilon}\boldsymbol{\eta})\boldsymbol{w}_r. \tag{C.7}$$

The sensitivity of $\boldsymbol{u}$ with respect to $\boldsymbol{\eta}$, i.e., $\dfrac{\partial \boldsymbol{u}}{\partial \boldsymbol{\eta}} = \dfrac{\partial \boldsymbol{u}}{\partial \boldsymbol{\eta}}(\boldsymbol{\eta};\, \bar{\boldsymbol{\mu}},\, \boldsymbol{\Upsilon})$ is defined as the solution of the sensitivity equations

$$\boldsymbol{r}^{\partial}\left(\boldsymbol{u}(\boldsymbol{\eta};\, \bar{\boldsymbol{\mu}},\, \boldsymbol{\Upsilon}),\, \cdot\,,\, \boldsymbol{\eta};\, \bar{\boldsymbol{\mu}},\, \boldsymbol{\Upsilon}\right) = 0, \tag{C.8}$$

where the sensitivity residual is defined as

$$\boldsymbol{r}^{\partial}(\boldsymbol{u},\, \boldsymbol{w}_r,\, \boldsymbol{\eta};\, \bar{\boldsymbol{\mu}},\, \boldsymbol{\Upsilon}) \coloneqq \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{\mu}}(\boldsymbol{u},\, \bar{\boldsymbol{\mu}} + \boldsymbol{\Upsilon}\boldsymbol{\eta})\boldsymbol{\Upsilon} + \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}(\boldsymbol{u},\, \bar{\boldsymbol{\mu}} + \boldsymbol{\Upsilon}\boldsymbol{\eta})\boldsymbol{w}_r. \tag{C.9}$$

Thus, the sensitivity computation requires the solution of $k_{\boldsymbol{\mu}}$ linear systems of equations defined by the Jacobian matrix with the kth right-hand side $\dfrac{\partial \boldsymbol{r}}{\partial \boldsymbol{\mu}}\boldsymbol{\Upsilon}\boldsymbol{e}_k$. For comparison, the sensitivity approach to compute $\nabla F(\boldsymbol{\mu})$ would require the solution of $N_{\boldsymbol{\mu}}$ linear systems defined by the Jacobian matrix and right-hand side $\dfrac{\partial \boldsymbol{r}}{\partial \boldsymbol{\mu}}\boldsymbol{e}_k$. From (C.2), the following relationship between the sensitivity of $\boldsymbol{u}$ with respect to $\boldsymbol{\mu}$ and $\boldsymbol{\eta}$ holds

$$\frac{\partial \boldsymbol{u}}{\partial \boldsymbol{\eta}}(\boldsymbol{\eta};\, \bar{\boldsymbol{\mu}},\, \boldsymbol{\Upsilon}) = \frac{\partial \boldsymbol{u}}{\partial \boldsymbol{\mu}}(\bar{\boldsymbol{\mu}} + \boldsymbol{\Upsilon}\boldsymbol{u}_r)\boldsymbol{\Upsilon} \tag{C.10}$$

Even though $k_{\boldsymbol{u}}$ is much smaller than $N_{\boldsymbol{u}}$, it may still be sufficiently large to prefer gradient computations via the adjoint method. Following any of the three procedures outlined in Section 2.3.4, the expression for $\nabla F(\boldsymbol{\eta};\, \bar{\boldsymbol{\mu}},\, \boldsymbol{\Upsilon})$ based on the adjoint method is

$$\nabla F(\boldsymbol{\eta};\, \bar{\boldsymbol{\mu}},\, \boldsymbol{\Upsilon}) = \boldsymbol{g}^{\boldsymbol{\lambda}}\left(\boldsymbol{u}(\boldsymbol{\eta};\, \bar{\boldsymbol{\mu}},\, \boldsymbol{\Upsilon}),\, \boldsymbol{\lambda}(\boldsymbol{\eta};\, \bar{\boldsymbol{\mu}},\, \boldsymbol{\Upsilon}),\, \boldsymbol{\eta};\, \bar{\boldsymbol{\mu}},\, \boldsymbol{\Upsilon}\right), \tag{C.11}$$

where the definition of $\boldsymbol{g}^{\boldsymbol{\lambda}}$ varies slightly from that in (2.102)

$$\boldsymbol{g}^{\boldsymbol{\lambda}}\left(\boldsymbol{u},\, \boldsymbol{z},\, \boldsymbol{\eta};\, \bar{\boldsymbol{\mu}},\, \boldsymbol{\Upsilon}\right) \coloneqq \frac{\partial f}{\partial \boldsymbol{\mu}}(\boldsymbol{u},\, \bar{\boldsymbol{\mu}} + \boldsymbol{\Upsilon}\boldsymbol{\eta})\boldsymbol{\Upsilon} + \boldsymbol{z}^T\frac{\partial \boldsymbol{r}}{\partial \boldsymbol{\mu}}(\boldsymbol{u},\, \bar{\boldsymbol{\mu}} + \boldsymbol{\Upsilon}\boldsymbol{\eta})\boldsymbol{\Upsilon}. \tag{C.12}$$

The adjoint state, $\boldsymbol{\lambda} = \boldsymbol{\lambda}(\boldsymbol{\eta};\, \bar{\boldsymbol{\mu}},\, \boldsymbol{\Upsilon})$ is defined as the solution of the adjoint equations

$$\boldsymbol{r}^{\boldsymbol{\lambda}}\left(\boldsymbol{u}(\boldsymbol{\eta};\, \bar{\boldsymbol{\mu}},\, \boldsymbol{\Upsilon}),\, \cdot\,,\, \boldsymbol{\eta};\, \bar{\boldsymbol{\mu}},\, \boldsymbol{\Upsilon}\right) = 0, \tag{C.13}$$

where the adjoint residual is defined as

$$\boldsymbol{r}^{\boldsymbol{\lambda}}(\boldsymbol{u},\, \boldsymbol{z},\, \boldsymbol{\eta};\, \bar{\boldsymbol{\mu}},\, \boldsymbol{\Upsilon}) \coloneqq \frac{\partial f}{\partial \boldsymbol{u}}(\boldsymbol{u},\, \bar{\boldsymbol{\mu}} + \boldsymbol{\Upsilon}\boldsymbol{\eta}) + \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}(\boldsymbol{u},\, \bar{\boldsymbol{\mu}} + \boldsymbol{\Upsilon}\boldsymbol{\eta})^T \boldsymbol{z}. \tag{C.14}$$

Thus, the adjoint computation requires the solution of one linear system of equations defined by the transpose of the Jacobian matrix, regardless of $k_{\boldsymbol{\mu}}$.

C.1.2 Inner Layer of Reduction: Projection-Based Model Reduction

While the first layer of reduction reduces the number of optimization variables, the large cost associated with solving the PDE for any $\boldsymbol{\mu} \in \mathcal{A}(\bar{\boldsymbol{\mu}},\, \boldsymbol{\Upsilon})$ remains since the dimensionality of the state space, i.e., number of equations and unknowns, is $N_{\boldsymbol{u}} \gg 1$. The second layer of reduction aims to address this source computational expense through the application of projection-based model reduction (Chapter 4).

Let $\boldsymbol{\Phi}$ and $\boldsymbol{\Psi}$ be a given trial and test basis defining a minimum-residual projection-based reduced-order model. Introduction of the model reduction ansatz $\boldsymbol{u} = \boldsymbol{\Phi}\boldsymbol{u}_r$ into the discretized PDE defined over the parameter space $\mathcal{A}(\bar{\boldsymbol{\mu}},\, \boldsymbol{\Upsilon})$ and subsequent projection onto the columnspace of $\boldsymbol{\Psi}$ leads to the projection-based reduced-order model

$$\boldsymbol{r}_r(\boldsymbol{u}_r,\, \boldsymbol{\eta};\, \bar{\boldsymbol{\mu}},\, \boldsymbol{\Upsilon},\, \boldsymbol{\Phi},\, \boldsymbol{\Psi}) \coloneqq \boldsymbol{\Psi}^T \boldsymbol{r}(\boldsymbol{\Phi}\boldsymbol{u}_r,\, \bar{\boldsymbol{\mu}} + \boldsymbol{\Upsilon}\boldsymbol{\eta}) = 0. \tag{C.15}$$

Denote the unique, continuously differentiable solution of the fully reduced model in (C.15) as $\boldsymbol{u}_r(\boldsymbol{\eta};\, \bar{\boldsymbol{\mu}},\, \boldsymbol{\Upsilon},\, \boldsymbol{\Phi},\, \boldsymbol{\Psi})$. Substitution of the reconstructed primal reduced-order model solution into the quantity of interest leads to its fully reduced form

$$F_r(\boldsymbol{\eta};\, \bar{\boldsymbol{\mu}},\, \boldsymbol{\Upsilon},\, \boldsymbol{\Phi},\, \boldsymbol{\Psi}) \coloneqq f(\boldsymbol{\Phi}\boldsymbol{u}_r(\boldsymbol{\eta};\, \bar{\boldsymbol{\mu}},\, \boldsymbol{\Upsilon},\, \boldsymbol{\Phi},\, \boldsymbol{\Psi}),\, \bar{\boldsymbol{\mu}} + \boldsymbol{\Upsilon}\boldsymbol{\eta}). \tag{C.16}$$

The gradient of the reduced quantity of interest is computed via the sensitivity (Section 2.3.3) or adjoint (Section 2.3.4) method as

$$\begin{aligned} \nabla F_r(\boldsymbol{\eta};\, \bar{\boldsymbol{\mu}},\, \boldsymbol{\Upsilon},\, \boldsymbol{\Phi},\, \boldsymbol{\Psi}) &= \boldsymbol{g}^{\partial}\left(\boldsymbol{u},\, \boldsymbol{\Phi}\frac{\partial \boldsymbol{u}_r}{\partial \boldsymbol{\eta}}(\boldsymbol{\eta};\, \bar{\boldsymbol{\mu}},\, \boldsymbol{\Upsilon},\, \boldsymbol{\Phi},\, \boldsymbol{\Psi}),\, \bar{\boldsymbol{\mu}} + \boldsymbol{\Upsilon}\boldsymbol{\eta}\right) \\ &= \boldsymbol{g}^{\boldsymbol{\lambda}}\left(\boldsymbol{u},\, \boldsymbol{\Psi}\boldsymbol{\lambda}_r(\boldsymbol{\eta};\, \bar{\boldsymbol{\mu}},\, \boldsymbol{\Upsilon},\, \boldsymbol{\Phi},\, \boldsymbol{\Psi}),\, \bar{\boldsymbol{\mu}} + \boldsymbol{\Upsilon}\boldsymbol{\eta}\right) \end{aligned} \tag{C.17}$$

where $\boldsymbol{u} = \boldsymbol{\Phi}\boldsymbol{u}_r(\boldsymbol{\eta};\, \bar{\boldsymbol{\mu}},\, \boldsymbol{\Upsilon},\, \boldsymbol{\Phi},\, \boldsymbol{\Psi})$ is the reconstructed primal solution, $\dfrac{\partial \boldsymbol{u}_r}{\partial \boldsymbol{\eta}}(\boldsymbol{\eta};\, \bar{\boldsymbol{\mu}},\, \boldsymbol{\Upsilon},\, \boldsymbol{\Phi},\, \boldsymbol{\Psi})$ is the solution of the reduced-order model sensitivity equations in (4.19), and $\boldsymbol{\lambda}_r(\boldsymbol{\eta};\, \bar{\boldsymbol{\mu}},\, \boldsymbol{\Upsilon},\, \boldsymbol{\Phi},\, \boldsymbol{\Psi})$ is the solution of the reduced-order model adjoint equations in (4.45). While the adjoint equations are identical to those in Section 4.1.3, the sensitivity equations require the following substitutions since we seek sensitivities with respect to $\boldsymbol{\eta}$ instead of $\boldsymbol{\mu}$:

$$\frac{\partial f}{\partial \boldsymbol{\mu}}(\boldsymbol{u},\, \boldsymbol{\mu}) \leftarrow \frac{\partial f}{\partial \boldsymbol{\mu}}(\boldsymbol{u},\, \boldsymbol{\mu})\boldsymbol{\Upsilon} \qquad \text{and} \qquad \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{\mu}}(\boldsymbol{u},\, \boldsymbol{\mu}) \leftarrow \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{\mu}}(\boldsymbol{u},\, \boldsymbol{\mu})\boldsymbol{\Upsilon} \tag{C.18}$$

for any $\boldsymbol{u} \in \mathbb{R}^{N_{\boldsymbol{u}}}$ and $\boldsymbol{\mu} \in \mathbb{R}^{N_{\boldsymbol{\mu}}}$. If the test basis is non-constant, following the developments of Sections 4.1.2 and 4.1.3, the minimum-residual approximation of the gradient $\widehat{\nabla F_r}(\boldsymbol{\mu})$ can be used to avoid computations involving second derivatives of $\boldsymbol{r}$

$$\begin{aligned}\widehat{\nabla F_r}(\boldsymbol{\eta};\, \bar{\boldsymbol{\mu}},\, \boldsymbol{\Upsilon},\, \boldsymbol{\Phi},\, \boldsymbol{\Psi}) &= \boldsymbol{g}^{\partial}\left(\boldsymbol{u},\, \boldsymbol{\Phi}^{\partial}\widehat{\frac{\partial \boldsymbol{u}_r}{\partial \boldsymbol{\eta}}}(\boldsymbol{\eta};\, \bar{\boldsymbol{\mu}},\, \boldsymbol{\Upsilon},\, \boldsymbol{\Phi}^{\partial},\, \boldsymbol{\Theta}^{\partial},\, \boldsymbol{u}),\, \bar{\boldsymbol{\mu}} + \boldsymbol{\Upsilon}\boldsymbol{\eta}\right) \\ &= \boldsymbol{g}^{\boldsymbol{\lambda}}\left(\boldsymbol{u}(\,\cdot\,),\, \boldsymbol{\Phi}^{\boldsymbol{\lambda}}\hat{\boldsymbol{\lambda}}_r(\boldsymbol{\eta};\, \bar{\boldsymbol{\mu}},\, \boldsymbol{\Upsilon},\, \boldsymbol{\Phi}^{\boldsymbol{\lambda}},\, \boldsymbol{\Theta}^{\boldsymbol{\lambda}},\, \boldsymbol{u}),\, \bar{\boldsymbol{\mu}} + \boldsymbol{\Upsilon}\boldsymbol{\eta}\right),\end{aligned} \tag{C.19}$$

where $\boldsymbol{u} = \boldsymbol{\Phi}\boldsymbol{u}_r(\boldsymbol{\eta};\, \bar{\boldsymbol{\mu}},\, \boldsymbol{\Upsilon},\, \boldsymbol{\Phi},\, \boldsymbol{\Psi})$ is the reconstructed primal solution, $\widehat{\dfrac{\partial \boldsymbol{u}_r}{\partial \boldsymbol{\eta}}}(\boldsymbol{\eta};\, \bar{\boldsymbol{\mu}},\, \boldsymbol{\Upsilon},\, \boldsymbol{\Phi}^{\partial},\, \boldsymbol{\Theta}^{\partial},\, \boldsymbol{u})$ is the solution of the minimum-residual sensitivity reduced-order model in (4.28), and $\hat{\boldsymbol{\lambda}}_r(\boldsymbol{\eta};\, \bar{\boldsymbol{\mu}},\, \boldsymbol{\Upsilon},\, \boldsymbol{\Phi}^{\boldsymbol{\lambda}},\, \boldsymbol{\Theta}^{\boldsymbol{\lambda}},\, \boldsymbol{u})$ is the solution of the minimum-residual adjoint reduced-order model in (4.56). For the minimum-residual sensitivity ROM in (4.28), the substitutions in (C.18) are required to directly compute sensitivities with respect to $\boldsymbol{\eta}$.

This section closes by stating the residual-based error bounds for the fully reduced quantity of interest, its gradient, and minimum-residual gradient approximation. The error bounds are given with respect to the first level of reduction as we are only concerned with the error for a fixed $\boldsymbol{\mu} \in \mathcal{A}(\bar{\boldsymbol{\mu}},\, \boldsymbol{\Upsilon})$. These will be used in the multifidelity trust region framework of Chapter 5 to solve (C.4), i.e., the optimization problem after the first layer of reduction. From Lemma B.4, the residual-based error bound on the quantity of interest takes the form

$$|F(\boldsymbol{\eta};\, \bar{\boldsymbol{\mu}},\, \boldsymbol{\Upsilon}) - F_r(\boldsymbol{\eta};\, \bar{\boldsymbol{\mu}},\, \boldsymbol{\Upsilon},\, \boldsymbol{\Phi},\, \boldsymbol{\Psi})| \leq \zeta \left\|\boldsymbol{r}(\boldsymbol{\Phi}\boldsymbol{u}_r(\boldsymbol{\eta};\, \bar{\boldsymbol{\mu}},\, \boldsymbol{\Upsilon},\, \boldsymbol{\Phi},\, \boldsymbol{\Psi}),\, \bar{\boldsymbol{\mu}} + \boldsymbol{\Upsilon}\boldsymbol{\eta})\right\| \tag{C.20}$$

for an arbitrary constant $\zeta > 0$. The residual-based error indicator for the gradient $\nabla F_r(\boldsymbol{\eta};\, \bar{\boldsymbol{\mu}},\, \boldsymbol{\Upsilon},\, \boldsymbol{\Phi},\, \boldsymbol{\Psi})$ computed with the sensitivity method is

$$\left\|\nabla F(\boldsymbol{\eta};\, \bar{\boldsymbol{\mu}},\, \boldsymbol{\Upsilon}) - \nabla F_r(\boldsymbol{\eta};\, \bar{\boldsymbol{\mu}},\, \boldsymbol{\Upsilon},\, \boldsymbol{\Phi},\, \boldsymbol{\Psi})\right\| \leq \kappa \left\|\boldsymbol{r}(\boldsymbol{u},\, \boldsymbol{\mu})\right\| + \tau \left\|\boldsymbol{r}^{\partial}(\boldsymbol{u},\, \boldsymbol{w},\, \boldsymbol{\mu})\right\| \tag{C.21}$$

where $\boldsymbol{u} = \boldsymbol{\Phi}\boldsymbol{u}_r(\boldsymbol{\eta};\, \bar{\boldsymbol{\mu}},\, \boldsymbol{\Upsilon},\, \boldsymbol{\Phi},\, \boldsymbol{\Psi}\boldsymbol{\Upsilon})$ is the reconstructed primal solution, $\boldsymbol{w} = \boldsymbol{\Phi}\dfrac{\partial \boldsymbol{u}_r}{\partial \boldsymbol{\eta}}(\boldsymbol{\eta};\, \bar{\boldsymbol{\mu}},\, \boldsymbol{\Upsilon},\, \boldsymbol{\Phi},\, \boldsymbol{\Psi})$, is the reconstructed sensitivity solution, and $\kappa,\, \tau > 0$ are arbitrary constants. The corresponding bound for gradients computed with the adjoint method is

$$\left\|\nabla F(\boldsymbol{\eta};\, \bar{\boldsymbol{\mu}},\, \boldsymbol{\Upsilon}) - \nabla F_r(\boldsymbol{\eta};\, \bar{\boldsymbol{\mu}},\, \boldsymbol{\Upsilon},\, \boldsymbol{\Phi},\, \boldsymbol{\Psi})\right\| \leq \kappa \left\|\boldsymbol{r}(\boldsymbol{u},\, \boldsymbol{\mu})\right\| + \tau \left\|\boldsymbol{r}^{\boldsymbol{\lambda}}(\boldsymbol{u},\, \boldsymbol{z},\, \boldsymbol{\mu})\right\| \tag{C.22}$$

where $\boldsymbol{u} = \boldsymbol{\Phi}\boldsymbol{u}_r(\boldsymbol{\eta};\, \bar{\boldsymbol{\mu}},\, \boldsymbol{\Upsilon},\, \boldsymbol{\Phi},\, \boldsymbol{\Psi}\boldsymbol{\Upsilon})$ is the reconstructed primal solution, $\boldsymbol{z} = \boldsymbol{\Psi}\boldsymbol{\lambda}_r(\boldsymbol{\eta};\, \bar{\boldsymbol{\mu}},\, \boldsymbol{\Upsilon},\, \boldsymbol{\Phi},\, \boldsymbol{\Psi})$ is the reconstructed adjoint solution, and $\kappa,\, \tau > 0$ are arbitrary constants. The residual-based error bounds for the minimum-residual approximation of the gradient of F_r are

$$\begin{aligned}\left\|\nabla F(\boldsymbol{\eta};\, \bar{\boldsymbol{\mu}},\, \boldsymbol{\Upsilon}) - \nabla F_r(\boldsymbol{\eta};\, \bar{\boldsymbol{\mu}},\, \boldsymbol{\Upsilon},\, \boldsymbol{\Phi},\, \boldsymbol{\Psi},\, \boldsymbol{\Phi}^{\partial},\, \boldsymbol{\Theta}^{\partial})\right\| &\leq \kappa \left\|\boldsymbol{r}(\boldsymbol{u},\, \boldsymbol{\mu})\right\| + \tau \left\|\boldsymbol{r}^{\partial}(\boldsymbol{u},\, \boldsymbol{w},\, \boldsymbol{\mu})\right\| \\ \left\|\nabla F(\boldsymbol{\eta};\, \bar{\boldsymbol{\mu}},\, \boldsymbol{\Upsilon}) - \nabla F_r(\boldsymbol{\eta};\, \bar{\boldsymbol{\mu}},\, \boldsymbol{\Upsilon},\, \boldsymbol{\Phi},\, \boldsymbol{\Psi},\, \boldsymbol{\Phi}^{\boldsymbol{\lambda}},\, \boldsymbol{\Theta}^{\boldsymbol{\lambda}})\right\| &\leq \kappa \left\|\boldsymbol{r}(\boldsymbol{u},\, \boldsymbol{\mu})\right\| + \tau \left\|\boldsymbol{r}^{\boldsymbol{\lambda}}(\boldsymbol{u},\, \boldsymbol{z},\, \boldsymbol{\mu})\right\|\end{aligned} \tag{C.23}$$

where $\boldsymbol{u} = \boldsymbol{\Phi}\boldsymbol{u}_r(\boldsymbol{\eta};\, \bar{\boldsymbol{\mu}},\, \boldsymbol{\Upsilon},\, \boldsymbol{\Phi},\, \boldsymbol{\Psi}\boldsymbol{\Upsilon})$ is the reconstructed primal solution, $\boldsymbol{w} = \boldsymbol{\Phi}^{\partial}\widehat{\frac{\partial \boldsymbol{u}_r}{\partial \boldsymbol{\eta}}}(\boldsymbol{\eta};\, \bar{\boldsymbol{\mu}},\, \boldsymbol{\Upsilon},\, \boldsymbol{\Phi},\, \boldsymbol{\Psi},\, \boldsymbol{\Phi}^{\partial},\, \boldsymbol{\Theta}^{\partial})$, is the reconstructed minimum-residual sensitivity solution, $\boldsymbol{z} = \boldsymbol{\Phi}^{\boldsymbol{\lambda}}\hat{\boldsymbol{\lambda}}_r(\boldsymbol{\eta};\, \bar{\boldsymbol{\mu}},\, \boldsymbol{\Upsilon},\, \boldsymbol{\Phi},\, \boldsymbol{\Psi},\, \boldsymbol{\Phi}^{\boldsymbol{\lambda}},\, \boldsymbol{\Theta}^{\boldsymbol{\lambda}})$ is the reconstructed minimum-residual adjoint solution, and $\kappa,\, \tau > 0$ are arbitrary constants.

Remark. *The $\boldsymbol{I}$-norm used in the error bounds (5.18) can be replaced with the minimum-residual metrics $\boldsymbol{\Theta}$, $\boldsymbol{\Theta}^{\partial}$, $\boldsymbol{\Theta}^{\boldsymbol{\lambda}}$ as done in Chapter 5 for greater consistency with the minimum-residual interpretation of the reduced-order model. This will be necessary if partially converged solutions are used as snapshots in construction of the reduced-order basis, as discussed in Section 5.2. This will be deferred to future work and the simpler (and less expensive) $\boldsymbol{I}$-norm will be used.*

C.2 Globally Convergent Multifidelity Trust Region Method

The two-level nested reduction of parametrized partial differential equations with a high-dimensional state and parameter space will serve as a pillar for an efficient method to solve optimization problems constrained by such PDEs. The first layer of reduction restricts the parameter space to the $k_{\boldsymbol{\mu}}$-dimensional affine subspace $\mathcal{A}(\bar{\boldsymbol{\mu}},\, \boldsymbol{\Upsilon})$ to yield an optimization problem in $k_{\boldsymbol{\mu}}$ variables. The second layer of reduction uses projection-based reduced-order models, embedded in the globally convergent multifidelity trust region framework of Chapter 3, i.e., the method developed in Chapter 5, to efficiently solve the $k_{\boldsymbol{\mu}}$-dimensional optimization problem. To ensure the method is globally convergent, the restricted parameter space $\mathcal{A}(\bar{\boldsymbol{\mu}},\, \boldsymbol{\Upsilon})$ is adapted using ideas from linesearch methods. The proposed optimization algorithm based on this nested reduction strategy consists of two types of iterations: (1) an inner iteration where the affine subspace for the parameter, $\mathcal{A}(\bar{\boldsymbol{\mu}},\, \boldsymbol{\Upsilon})$ is fixed and the multifidelity trust region method based on projection-based reduced-order models (Chapter 5) is applied to solve the optimization problem in (C.4) and (2) an outer iteration that adapts the parameter subspace $\mathcal{A}(\bar{\boldsymbol{\mu}},\, \boldsymbol{\Upsilon})$ to ensure global convergence of the complete algorithm. The inner iteration is guaranteed to converge to the solution of (C.4) since the multifidelity method introduced in Chapter 5 is globally convergent. The parameter subspace adaptation in the outer iteration will be constructed such that global convergence to the solution of (C.1) is guaranteed. The next two sections detail both the inner and outer iterations.

C.2.1 Outer Iteration: Globally Convergent Parameter Space Adaptation

It is not reasonable to expect an a-prior selection of the restricted parameter subspace $\mathcal{A}(\bar{\boldsymbol{\mu}},\, \boldsymbol{\Upsilon})$ to lead to a globally convergent algorithm since, in general, $\boldsymbol{\mu}^* \notin \mathcal{A}(\bar{\boldsymbol{\mu}},\, \boldsymbol{\Upsilon})$ where $\boldsymbol{\mu}^*$ is a local minima of $F(\boldsymbol{\mu})$. Therefore, keeping with the theme of this document, this section develops an adaptation strategy for the affine offset $\bar{\boldsymbol{\mu}}$ and subspace $\boldsymbol{\Upsilon}$ defining the restricted parameter space. That is, an algorithm that constructs a sequence of affine subspaces $\{\mathcal{A}(\bar{\boldsymbol{\mu}}_j,\, \boldsymbol{\Upsilon}_j)\}$ of dimension $k^j_{\boldsymbol{\mu}} \ll N_{\boldsymbol{\mu}}$ such

that the iterates $\{\boldsymbol{\mu}_j\}$, computed as the solution of the restricted optimization problem

$$\boldsymbol{\mu}_{j+1} \coloneqq \underset{\boldsymbol{\mu}\in\mathcal{A}(\bar{\boldsymbol{\mu}}_j,\,\boldsymbol{\Upsilon}_j)}{\arg\min}\; F(\boldsymbol{\mu}), \tag{C.24}$$

converge to a stationary point of $F(\boldsymbol{\mu})$ over $\mathbb{R}^{N_{\boldsymbol{\mu}}}$, i.e., $\lim \left\|\nabla F(\boldsymbol{\mu}_j)\right\| = 0$. From the discussion in Section C.1, the definition in (C.24) is equivalent to

$$\boldsymbol{\mu}_{j+1} = \bar{\boldsymbol{\mu}}_j + \boldsymbol{\Upsilon}_j\boldsymbol{\eta}_{j+1} \qquad\qquad \boldsymbol{\eta}_{j+1} = \underset{\boldsymbol{\eta}\in\mathbb{R}^{k^j_{\boldsymbol{\mu}}}}{\arg\min}\; F(\boldsymbol{\eta};\,\bar{\boldsymbol{\mu}}_j,\,\boldsymbol{\Upsilon}_j), \tag{C.25}$$

i.e., the search in the $k^j_{\boldsymbol{\mu}}$-dimensional subspace embedded in $N_{\boldsymbol{\mu}}$ is equivalent to an optimization problem in $k^j_{\boldsymbol{\mu}}$ variables.

Before launching into the details of the proposed adaptation strategy, recall two standard results from optimization theory stated in Lemma C.1 and Theorem C.1. Theorem C.1 states that any iteration of the form $\boldsymbol{\mu}_{j+1} = \boldsymbol{\mu}_j + \alpha_j\boldsymbol{p}_j$, where $\boldsymbol{p}_j$ is a descent direction at $\boldsymbol{\mu}_j$ and $\alpha_j > 0$ satisfies the Wolfe conditions (C.28), constitutes a globally convergent optimization method and Lemma C.1 establishes the existence of a point satisfying the Wolfe conditions for any descent direction. These results are combined to arrive at the following conclusion: *if* $\bar{\boldsymbol{\mu}}_j = \boldsymbol{\mu}_j$ *and* $col(\boldsymbol{\Upsilon}_j)$ *contains a descent direction of* F *at* $\boldsymbol{\mu}_j$*, the sequence* $\{\boldsymbol{\mu}_j\}$ *produced by (C.24) will satisfy* $\lim_{j\to\infty}\left\|\nabla F(\boldsymbol{\mu}_j)\right\| = 0$. This claim is justified since $\boldsymbol{\mu}_{j+1}$ is the exact solution of the optimization problem restricted to $\mathcal{A}(\bar{\boldsymbol{\mu}}_j,\,\boldsymbol{\Upsilon}_j)$ (which contains $\boldsymbol{\mu}_j$ and a descent direction of $F(\boldsymbol{\mu}_j)$) and, since a point exists that satisfies the sufficient decrease conditions (Lemma C.1), $\boldsymbol{\mu}_{j+1}$ must also satisfy them and the iteration is globally convergent (Theorem C.1). This argument is justified rigorously by showing $\boldsymbol{\mu}_{j+1}$ satisfies the strong Wolfe conditions since Theorem C.1 guarantees global convergence if these conditions hold. The choice $\bar{\boldsymbol{\mu}}_j = \boldsymbol{\mu}_j$ implies the affine subspace $\mathcal{A}(\bar{\boldsymbol{\mu}}_j,\,\boldsymbol{\Upsilon}_j)$ contains points of the form $\boldsymbol{\mu} = \boldsymbol{\mu}_j + \boldsymbol{\Upsilon}_j\boldsymbol{\eta}$. Since $\mathrm{col}(\boldsymbol{\Upsilon}_j)$ contains a descent direction of F at $\boldsymbol{\mu}_j$, there must exist $\boldsymbol{\eta}\in\mathbb{R}^{k^j_{\boldsymbol{\mu}}}$ such that $\boldsymbol{p}_j = (1/\alpha_j)\boldsymbol{\Upsilon}_j\boldsymbol{\eta}$ is a descent direction of F at $\boldsymbol{\mu}_j$ for any $\alpha_j > 0$. Thus, the affine subspace $\mathcal{A}(\bar{\boldsymbol{\mu}}_j,\,\boldsymbol{\Upsilon}_j)$ contains vectors of the form $\boldsymbol{\mu} = \boldsymbol{\mu}_j + \alpha_j\boldsymbol{p}_j$ and Lemma C.1 guarantees the existence of an interval of step sizes (α_j) that satisfies the strong Wolfe conditions. Let α^*_j be any such step size. Then the following relations hold

$$F(\boldsymbol{\mu}_{j+1}) \le F(\boldsymbol{\mu}_j + \alpha^*_j\boldsymbol{p}_j) \le F(\boldsymbol{\mu}_j) + c_1\alpha^*_j\boldsymbol{p}_j^T\nabla F(\boldsymbol{\mu}_j), \tag{C.26}$$

where the first inequality follows from $\boldsymbol{\mu}_{j+1}$ being the solution of the optimization problem in (C.24) and the second holds since α^*_j satisfies the strong Wolfe conditions. This establishes the first strong Wolfe condition in (C.28). For the remaining Wolfe condition, observe that $\boldsymbol{p}_j^T\nabla F(\boldsymbol{\mu}_{j+1}) = 0$. This follows from the fact that $\boldsymbol{p}_j = (1/\alpha_j)\boldsymbol{\Upsilon}_j\boldsymbol{\eta}$ and the first-order optimality condition of (C.24), i.e., $\boldsymbol{\Upsilon}_j^T\nabla F(\boldsymbol{\mu}_{j+1}) = 0$. Therefore, the following relationships hold

$$\boldsymbol{p}_j^T\nabla F(\boldsymbol{\mu}_{j+1}) = 0 \le |\boldsymbol{p}_j^T\nabla F(\boldsymbol{\mu}_j + \alpha^*_j\boldsymbol{p}_j)| \le c_2|\boldsymbol{p}_j^T\nabla F(\boldsymbol{\mu}_j)|, \tag{C.27}$$

which establishes that $\boldsymbol{\mu}_{j+1}$ satisfies the second Wolfe condition. Therefore, by Theorem C.1, global convergence of the sequence $\{\boldsymbol{\mu}_j\}$ is guaranteed.

Lemma C.1. *Let $\{\boldsymbol{\mu}_j\}$ be a sequence of iterations that satisfy the update formula $\boldsymbol{\mu}_{j+1} = \boldsymbol{\mu}_j + \alpha_j \boldsymbol{p}_j$, where $\boldsymbol{p}_j$ is any descent direction at $\boldsymbol{\mu}_j$. Suppose $F(\boldsymbol{\mu})$ is continuously differentiable and bounded below along the ray $\{\boldsymbol{\mu}_j + \alpha \boldsymbol{p}_j \mid \alpha > 0\}$. Then, if $0 < c_1 < c_2 < 1$, there exist intervals of step lengths satisfying the strong Wolfe conditions*

$$\begin{aligned} F(\boldsymbol{\mu}_j + \alpha_j \boldsymbol{p}_j) &\leq F(\boldsymbol{\mu}_j) + c_1 \alpha_j \boldsymbol{p}_j^T \nabla F(\boldsymbol{\mu}_j) \\ |\boldsymbol{p}_j^T \nabla F(\boldsymbol{\mu}_j + \alpha_j \boldsymbol{p}_j)| &\leq c_2 |\boldsymbol{p}_j^T \nabla F(\boldsymbol{\mu}_j)|. \end{aligned} \tag{C.28}$$

Proof. Lemma 3.1 of [143]. □

Theorem C.1. *Let $\{\boldsymbol{\mu}_j\}$ be a sequence of iterations that satisfies the update formula $\boldsymbol{\mu}_{j+1} = \boldsymbol{\mu}_j + \alpha_j \boldsymbol{p}_j$, where $\boldsymbol{p}_j$ is any descent direction at $\boldsymbol{\mu}_j$ and α_j satisfies the strong Wolfe conditions (C.28) with $0 < c_1 < c_2 < 1$. Suppose the F is bounded below in $\mathbb{R}^{N_\mu}$ and continuously differentiable in an open set $\mathcal{N}$ containing the level set $\{\boldsymbol{\mu} \in \mathbb{R}^{N_\mu} \mid F(\boldsymbol{\mu}) \leq F(\boldsymbol{\mu}_0)\}$. Assume also its gradient is Lipschitz continuous on $\mathcal{N}$. Then*

$$\lim_{j \to \infty} \left\| \nabla F(\boldsymbol{\mu}_j) \right\| = 0. \tag{C.29}$$

Proof. Theorem 3.2 of [143]. □

Remark. *In linesearch and subspace methods, it is usually considered difficult or expensive to solve the low-dimensional optimization problem, e.g., (C.24), exactly. This lead to the introduction of the Wolfe conditions (C.28) that define a criteria for sufficient decrease in the objective function that will lead to global convergence (Theorem C.1). As a result, a slew of linesearch methods have been developed to locate points that satisfy the Wolfe conditions [143]. The proposed method deviates from this accepted strategy by solving the restricted optimization problem exactly to leverage the efficient method developed in Chapter 5 for solving PDE-constrained optimization problems in few variables using projection-based reduced-order models in the multifidelity trust region method of Chapter 3. To align with standard practices, the inner iteration can be terminated once the strong Wolfe conditions are satisfied without destroying global convergence.*

Let $\boldsymbol{p}_j$ be any descent direction to F at $\boldsymbol{\mu}_j$. From Theorem C.1, the following requirements on the affine subspace $\mathcal{A}(\bar{\boldsymbol{\mu}}_j, \boldsymbol{\Upsilon}_j)$ are sufficient to guarantee the iteration in (C.24) is globally convergent

$$\bar{\boldsymbol{\mu}}_j = \boldsymbol{\mu}_j \qquad \boldsymbol{p}_j \in \operatorname{col}(\boldsymbol{\Upsilon}_j). \tag{C.30}$$

The simplest affine subspace that fulfills these requirements is defined by

$$\bar{\boldsymbol{\mu}}_j = \boldsymbol{\mu}_j \qquad \boldsymbol{\Upsilon}_j = \boldsymbol{\Upsilon}_j^g \coloneqq \left[\nabla F(\boldsymbol{\mu}_j) \right], \tag{C.31}$$

which reduces the iteration in (C.24) to a steepest descent method with an exact linesearch. While this choice will result in a globally convergent iteration, steepest descent methods are well-known to suffer from *slow* convergence. The remainder of the section will construct a more sophisticated affine subspace such that the iteration in (C.24) quickly converges to a local minima.

From the requirements in (C.30), the affine offset will always be taken as the previous iterate $\bar{\boldsymbol{\mu}}_j = \boldsymbol{\mu}_j$. While the requirement in (C.30) provides considerable flexibility in the definition of $\boldsymbol{\Upsilon}_j$, we impose the stronger requirement that the affine subspace must contain the steepest descent space: $\mathcal{A}(\boldsymbol{\mu}_j, \boldsymbol{\Upsilon}_j^g) \subseteq \mathcal{A}(\bar{\boldsymbol{\mu}}_j, \boldsymbol{\Upsilon}_j)$. This is accomplished by taking the first column of $\boldsymbol{\Upsilon}_j$ to be $\nabla F(\boldsymbol{\mu}_j)$ and guarantees global convergence regardless of the other basis vectors that comprise $\boldsymbol{\Upsilon}_j$. These auxiliary basis vectors in $\boldsymbol{\Upsilon}_j$ will serve to improve the convergence *rate* of the iteration in (C.24). We will consider two types of auxiliary vectors: (1) optimization-based vectors that are defined for any optimization problem and (2) problem-specific information that exploits any knowledge or structure of the optimization variables $\boldsymbol{\mu}$.

The optimization-based vectors will consist of any variety of descent directions, i.e., Newton or quasi-Newton direction, or directions of negative curvature at the current iterate $\boldsymbol{\mu}_j$. Let $\boldsymbol{P}_j$ be a matrix consisting of such all optimization-based vectors and define

$$\boldsymbol{\Upsilon}_j = \begin{bmatrix} \nabla F(\boldsymbol{\mu}_j) & \boldsymbol{P}_j \end{bmatrix}. \tag{C.32}$$

This construction is general since the aforementioned directions can be constructed for *any* optimization problem.

In many applications, particularly those related to PDEs, it may be advantageous to incorporate problem-specific information in the affine subspace. This is particularly true for topology optimization and inverse problems where the optimization vectors have a strong connection to the underlying PDE mesh. The proposed framework is sufficiently flexible to incorporate such information without destroying global convergence by building $\boldsymbol{\Upsilon}_j$ according to

$$\boldsymbol{\Upsilon}_j = \begin{bmatrix} \nabla F(\boldsymbol{\mu}_j) & \boldsymbol{P}_j & \boldsymbol{Q}_j \end{bmatrix} \tag{C.33}$$

where $\boldsymbol{Q}_j$ is a matrix whose columns consist of problem-specific vectors. Future work will develop problem-specific information for various in structural and acoustic inverse problems. Algorithm 17 provides the complete outer iteration algorithm.

C.2.2 Inner Iteration: Multifidelity Optimization with Reduced-Order Models

Each iteration of the affine parameter space adaptation requires the solution of the PDE-constrained optimization problem (C.24), which can be written as an optimization problem in few variables ($k_{\boldsymbol{\mu}} \ll N_{\boldsymbol{\mu}}$). Even though the optimization problem contains few variables, it is still expensive to solve since each objective evaluation requires the solution of a potentially large-scale partial

Algorithm 17 Outer iteration: adaptive reduction of parameter space

1: **Initialization**: Given

$$\bar{\boldsymbol{\mu}}_0,\ \boldsymbol{\Upsilon}_0$$

2: **Inner iteration**: Solve restricted optimization problem (Algorithm 18)

$$\underset{\boldsymbol{\eta}\in\mathbb{R}^{k_{\boldsymbol{\mu}}^j}}{\text{minimize}}\ \ F(\bar{\boldsymbol{\mu}}_j + \boldsymbol{\Upsilon}_j\boldsymbol{\eta})$$

for $\boldsymbol{\eta}_j^*$, the optimal solution in the restricted parameter space and define $\boldsymbol{\mu}_j^* = \bar{\boldsymbol{\mu}}_j + \boldsymbol{\Upsilon}_j\boldsymbol{\eta}_j^*$

3: **Update search space**: Compute $\nabla F(\boldsymbol{\mu}_j^*)$, the optimization-based vectors $\boldsymbol{P}(\boldsymbol{\mu}_j^*)$, and the problem-specific vectors $\boldsymbol{Q}(\boldsymbol{\mu}_j^*)$ and update the restricted parameter space

$$\bar{\boldsymbol{\mu}}_{j+1} = \boldsymbol{\mu}_j^* \qquad \boldsymbol{\Upsilon}_{j+1} = \begin{bmatrix}\nabla F(\boldsymbol{\mu}_j^*) & \boldsymbol{P}(\boldsymbol{\mu}_j^*) & \boldsymbol{Q}(\boldsymbol{\mu}_j^*)\end{bmatrix}$$

differential equation, and the gradient requires a sensitivity or adjoint solution. The multifidelity trust region method based on projection-based model reduction proposed in Chapter 5 has been shown to be an efficient method to handle exactly these types of problems. This section will consider a special case of the method proposed in Chapter 5 to solve each $k_{\boldsymbol{\mu}}$-variable optimization problem encountered in the iteration (C.24).

Consider the optimization problem that arises at iteration j of (C.24)

$$\underset{\boldsymbol{\mu}\in\mathcal{A}(\bar{\boldsymbol{\mu}}_j,\,\boldsymbol{\Upsilon}_j)}{\text{minimize}}\ \ F(\boldsymbol{\mu}) \tag{C.34}$$

which, from the definition of F and $\mathcal{A}(\bar{\boldsymbol{\mu}},\,\boldsymbol{\Upsilon})$, is equivalent to the $k_{\boldsymbol{\mu}}^j$-dimensional optimization problem

$$\underset{\boldsymbol{\eta}\in\mathbb{R}^{k_{\boldsymbol{\mu}}^j}}{\text{minimize}}\ \ F(\boldsymbol{\eta};\,\bar{\boldsymbol{\mu}}_j,\,\boldsymbol{\Upsilon}_j). \tag{C.35}$$

We propose to solve this PDE-constrained optimization problem in few parameters using the method proposed in Chapter 5, i.e., the the multifidelity trust region method using reduced-order/hyperreduced approximation models. For a fixed outer iteration j, the approximation model at iteration k of the trust region method is defined as

$$m_{j,\,k}(\boldsymbol{\eta}) = F_r(\boldsymbol{\eta};\,\bar{\boldsymbol{\mu}}_j,\,\boldsymbol{\Upsilon}_j,\,\boldsymbol{\Phi}_{j,\,k},\,\boldsymbol{\Psi}_{j,\,k}), \tag{C.36}$$

where F_r is defined in (C.16) and $\boldsymbol{\Phi}_{j,\,k}$, $\boldsymbol{\Psi}_{j,\,k}$ are presumed given and define a projection-based reduced-order model that possesses the minimum-residual property. Details pertaining to the construction of the trial basis $\boldsymbol{\Phi}_{j,\,k}$ (and implicitly the test basis $\boldsymbol{\Psi}_{j,\,k}$ based on the minimum-residual requirement (4.14)) are provided at the end of this section. The gradient of the approximation model is computed exactly as

$$\nabla m_{j,\,k}(\boldsymbol{\eta}) = \nabla F_r(\boldsymbol{\eta};\,\bar{\boldsymbol{\mu}}_j,\,\boldsymbol{\Upsilon}_j,\,\boldsymbol{\Phi}_{j,\,k},\,\boldsymbol{\Psi}_{j,\,k}) \tag{C.37}$$

according to the sensitivity or adjoint method as defined in Section 2.3. In situations where the test basis is not constant, the exact gradient is cumbersome to compute and may be approximated with

$$\begin{aligned}\widehat{\nabla m}_{j,k}(\boldsymbol{\eta}) &= \widehat{\nabla F}_r(\boldsymbol{\eta};\, \bar{\boldsymbol{\mu}}_j,\, \boldsymbol{\Upsilon}_j,\, \boldsymbol{\Phi}_{j,k},\, \boldsymbol{\Psi}_{j,k},\, \boldsymbol{\Phi}^{\partial}_{j,k},\, \boldsymbol{\Theta}^{\partial}_{j,k}) \\ &= \widehat{\nabla F}_r(\boldsymbol{\eta};\, \bar{\boldsymbol{\mu}}_j,\, \boldsymbol{\Upsilon}_j,\, \boldsymbol{\Phi}_{j,k},\, \boldsymbol{\Psi}_{j,k},\, \boldsymbol{\Phi}^{\boldsymbol{\lambda}}_{j,k},\, \boldsymbol{\Theta}^{\boldsymbol{\lambda}}_{j,k})\end{aligned} \tag{C.38}$$

using minimum-residual sensitivity or adjoint reduced-order models.

A critical component of the multifidelity trust region method of Chapter 3 is the introduction of an objective decrease error indicator $\vartheta_k(\boldsymbol{\mu})$ and gradient error indicator $\varphi_k(\boldsymbol{\mu})$ that lead to the error bounds

$$\begin{aligned}|F(\boldsymbol{\eta}_{j,k};\, \bar{\boldsymbol{\mu}}_j,\, \boldsymbol{\Upsilon}_j) - F(\boldsymbol{\eta};\, \bar{\boldsymbol{\mu}}_j,\, \boldsymbol{\Upsilon}_j) + m_{j,k}(\boldsymbol{\eta}) - m_{j,k}(\boldsymbol{\eta}_{j,k})| &\leq \zeta \vartheta_{j,k}(\boldsymbol{\eta}) \\ \left\|\nabla F(\boldsymbol{\eta};\, \bar{\boldsymbol{\mu}}_j,\, \boldsymbol{\Upsilon}_j) - \nabla m_{j,k}(\boldsymbol{\eta})\right\| &\leq \xi \varphi_{j,k}(\boldsymbol{\eta}),\end{aligned} \tag{C.39}$$

where $\zeta,\, \xi > 0$ are arbitrary constants and $\boldsymbol{\eta}_{j,k}$ is the trust region center in the reduced parameter space. Two options are considered for the objective decrease error indicator: the classical trust region constraint $\vartheta_{j,k}(\boldsymbol{\eta}) = \left\|\boldsymbol{\eta} - \boldsymbol{\eta}_{j,k}\right\|$ and the residual-based error indicator introduced in Section 5.1.1

$$\begin{aligned}\vartheta_{j,k}(\boldsymbol{\eta}) = &\left\|\boldsymbol{r}(\boldsymbol{\Phi}_{j,k}\boldsymbol{u}_r(\boldsymbol{\eta}_{j,k};\, \bar{\boldsymbol{\mu}}_j,\, \boldsymbol{\Upsilon}_j,\, \boldsymbol{\Phi}_{j,k},\, \boldsymbol{\Psi}_{j,k}),\, \bar{\boldsymbol{\mu}}_j + \boldsymbol{\Upsilon}_j\boldsymbol{\eta}_{j,k})\right\| + \\ &\left\|\boldsymbol{r}(\boldsymbol{\Phi}_{j,k}\boldsymbol{u}_r(\boldsymbol{\eta};\, \bar{\boldsymbol{\mu}}_j,\, \boldsymbol{\Upsilon}_j,\, \boldsymbol{\Phi}_{j,k},\, \boldsymbol{\Psi}_{j,k}),\, \bar{\boldsymbol{\mu}}_j + \boldsymbol{\Upsilon}_j\boldsymbol{\eta})\right\|.\end{aligned} \tag{C.40}$$

From the discussion in Section 5.1.1 that refers to the proof in Appendix B, the residual-based error indicator satisfies the bound in (3.12). The classical trust region satisfies this bound, provided the gradient bound and condition hold (see Chapter 3 for a complete discussion). For simplicity, only the classical trust region constraint will be considered in the remainder; see Chapter 5 for a complete discussion regarding the use of the residual-based error indicator. From the bounds on the gradient error derived in Lemmas B.7 and B.8, the gradient error indicator is taken as

$$\begin{aligned}\varphi_{j,k}(\boldsymbol{\eta}) = &\alpha_1 \left\|\boldsymbol{r}(\boldsymbol{\Phi}_{j,k}\boldsymbol{u}_r(\boldsymbol{\eta};\, \bar{\boldsymbol{\mu}}_j,\, \boldsymbol{\Upsilon}_j,\, \boldsymbol{\Phi}_{j,k},\, \boldsymbol{\Psi}_{j,k}),\, \bar{\boldsymbol{\mu}}_j + \boldsymbol{\Upsilon}_j\boldsymbol{\eta})\right\| + \\ &\alpha_2 \left\|\boldsymbol{r}^{\partial}\left(\boldsymbol{\Phi}_{j,k}\boldsymbol{u}_r(\boldsymbol{\eta};\, \bar{\boldsymbol{\mu}}_j,\, \boldsymbol{\Upsilon}_j,\, \boldsymbol{\Phi}_{j,k},\, \boldsymbol{\Psi}_{j,k}),\, \boldsymbol{\Phi}_{j,k}\frac{\partial \boldsymbol{u}_r}{\partial \boldsymbol{\eta}}(\boldsymbol{\eta};\, \bar{\boldsymbol{\mu}}_j,\, \boldsymbol{\Upsilon}_j,\, \boldsymbol{\Phi}_{j,k},\, \boldsymbol{\Psi}_{j,k}),\, \hat{\boldsymbol{\mu}} + \boldsymbol{\Upsilon}\boldsymbol{\eta}\right)\right\|\end{aligned} \tag{C.41}$$

if the sensitivity approach is used to compute $\nabla m_{j,k}(\boldsymbol{\eta})$ and

$$\begin{aligned}\varphi_{j,k}(\boldsymbol{\eta}) = &\alpha_1 \left\|\boldsymbol{r}(\boldsymbol{\Phi}_{j,k}\boldsymbol{u}_r(\boldsymbol{\eta};\, \bar{\boldsymbol{\mu}}_j,\, \boldsymbol{\Upsilon}_j,\, \boldsymbol{\Phi}_{j,k},\, \boldsymbol{\Psi}_{j,k}),\, \bar{\boldsymbol{\mu}}_j + \boldsymbol{\Upsilon}_j\boldsymbol{\eta})\right\| + \\ &\alpha_2 \left\|\boldsymbol{r}^{\boldsymbol{\lambda}}\left(\boldsymbol{\Phi}_{j,k}\boldsymbol{u}_r(\boldsymbol{\eta};\, \bar{\boldsymbol{\mu}}_j,\, \boldsymbol{\Upsilon}_j,\, \boldsymbol{\Phi}_{j,k},\, \boldsymbol{\Psi}_{j,k}),\, \boldsymbol{\Psi}_{j,k}\boldsymbol{\lambda}_r(\boldsymbol{\eta};\, \bar{\boldsymbol{\mu}}_j,\, \boldsymbol{\Upsilon}_j,\, \boldsymbol{\Phi}_{j,k},\, \boldsymbol{\Psi}_{j,k}),\, \hat{\boldsymbol{\mu}} + \boldsymbol{\Upsilon}\boldsymbol{\eta}\right)\right\|\end{aligned} \tag{C.42}$$

if the adjoint approach is used. These indicators can be modified accordingly if the minimum-residual sensitivity or adjoint approach is used to compute the gradient approximation $\widehat{\nabla m}_{j,k}(\boldsymbol{\mu})$. Finally, the trust region method of Chapter 3 provides the flexibility to introduce an inexpensive approximation of the objective decrease $\psi_{j,k}(\boldsymbol{\eta})$ and corresponding error indicator $\theta_{j,k}(\boldsymbol{\eta})$ to mitigate

the computational burden of computing the actual-to-predicted reduction at each trust region step. Section 5.3 details an approach that defines $\psi_{j,k}(\boldsymbol{\eta})$ based on partially converged PDE solutions and $\theta_{j,k}(\boldsymbol{\eta})$ as the residual-based error indicator. This construction can be used in this context without modification and does not need to be discussed further.

With the definition of the necessary approximations and corresponding error indicators, the only remaining conditions that are left to satisfy are the error conditions in (3.14) and (3.15), restated here for convenience

$$\begin{aligned} \vartheta_{j,k}(\boldsymbol{\eta}_{j,k}) &\leq \kappa_\vartheta \Delta_{j,k} \\ \varphi_{j,k}(\boldsymbol{\eta}_{j,k}) &\leq \kappa_\varphi \min\{\left\|\nabla m_{j,k}(\boldsymbol{\eta}_{j,k})\right\|, \Delta_{j,k}\}. \end{aligned} \tag{C.43}$$

Since the classical trust region constraint is used to define $\vartheta_{j,k}(\boldsymbol{\eta})$, the first condition is always satisfied since $\vartheta_{j,k}(\boldsymbol{\eta}_{j,k}) = 0$. The second condition, called the gradient condition, is not always satisfied a priori and relies critically on the construction of the reduced-order model. The strategy taken constructs the reduced-order model such that the reconstructed primal and sensitivity/adjoint solutions exactly match the corresponding high-dimensional model quantity. This will obviously guarantee $\varphi_{j,k}(\boldsymbol{\eta}_{j,k}) = 0$ and the gradient condition will be satisfied. Without repeating the details from Section 5.1.2, the reduced-order model and its sensitivity/adjoint will be possess these interpolation properties provided primal and sensitivity/adjoint minimum-residual reduced-order models are used, the relationships between the reduced-order bases in (4.35) and (4.63) hold, and the trial basis possesses the following properties

$$\begin{aligned} \boldsymbol{u}(\boldsymbol{\eta}_{j,k};\, \bar{\boldsymbol{\mu}}_j,\, \boldsymbol{\Upsilon}_j) &\in \operatorname{col}(\boldsymbol{\Phi}_{j,k}) \\ \frac{\partial \boldsymbol{u}}{\partial \boldsymbol{\eta}}(\boldsymbol{\eta}_{j,k};\, \bar{\boldsymbol{\mu}}_j,\, \boldsymbol{\Upsilon}_j) &\in \operatorname{col}(\boldsymbol{\Phi}_{j,k}) \\ \boldsymbol{\Theta}^{\boldsymbol{\lambda}}_{j,k}(\boldsymbol{u},\, \boldsymbol{\mu}) \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}(\boldsymbol{u},\, \boldsymbol{\mu})^T \boldsymbol{\lambda}(\boldsymbol{\eta}_{j,k};\, \bar{\boldsymbol{\mu}}_j,\, \boldsymbol{\Upsilon}_j) &\in \operatorname{col}(\boldsymbol{\Phi}_{j,k}) \end{aligned} \tag{C.44}$$

where $\boldsymbol{\mu} = \bar{\boldsymbol{\mu}}_j + \boldsymbol{\Upsilon}_j \boldsymbol{\eta}$ and $\boldsymbol{u} = \boldsymbol{u}(\boldsymbol{\mu})$. The conditions in (4.14), (4.35), and (4.63) completely specify the test $\boldsymbol{\Psi}_{j,k}$, sensitivity $\boldsymbol{\Phi}^{\partial}_{j,k}$, and adjoint $\boldsymbol{\Phi}^{\boldsymbol{\lambda}}_{j,k}$ bases in terms of the trial basis $\boldsymbol{\Phi}_{j,k}$ and optimality metrics $\boldsymbol{\Theta}_{j,k}$, $\boldsymbol{\Theta}^{\partial}_{j,k}$, $\boldsymbol{\Theta}^{\boldsymbol{\lambda}}_{j,k}$. Therefore, the reduced-order model will possess the required interpolation properties provided the trial basis is constructed such that (C.44) holds.

Remark. *The requirement that the reduced-order model is exact at the trust region center leads to the stronger condition $\varphi_{j,k}(\boldsymbol{\eta}_{j,k}) = 0$ than required by (3.15) and may result in wasted effort. The weaker condition in (3.15) can be enforced directly using partially converged solutions in the construction of $\boldsymbol{\Phi}_{j,k}$ as detailed in Section 5.2; however, this is not considered in this section.*

Before continuing with the construction of $\boldsymbol{\Phi}_{j,k}$, the following notation is introduced to allow the sensitivity and adjoint method to be treated simultaneously and compactly:

$$\hat{\boldsymbol{v}}(\boldsymbol{\eta};\, \bar{\boldsymbol{\mu}},\, \boldsymbol{\Upsilon}) = \begin{cases} \dfrac{\partial \boldsymbol{u}}{\partial \boldsymbol{\eta}}(\boldsymbol{\eta};\, \bar{\boldsymbol{\mu}},\, \boldsymbol{\Upsilon}) & \text{sensitivity method} \\ \boldsymbol{\Theta}^{\boldsymbol{\lambda}}_{j,k}(\boldsymbol{u},\, \boldsymbol{\mu}) \dfrac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}(\boldsymbol{u},\, \boldsymbol{\mu})^T \boldsymbol{\lambda}(\boldsymbol{\eta};\, \bar{\boldsymbol{\mu}},\, \boldsymbol{\Upsilon}) & \text{adjoint method} \end{cases} \tag{C.45}$$

where $\boldsymbol{\mu} = \bar{\boldsymbol{\mu}}_j + \boldsymbol{\Upsilon}_j \boldsymbol{\eta}$ and $\boldsymbol{u} = \boldsymbol{u}(\boldsymbol{\mu})$. Since the sensitivity and adjoint method are rarely employed simultaneously the condition in (C.44) is weakened to

$$\begin{aligned} \boldsymbol{u}(\boldsymbol{\eta}_{j,k};\, \bar{\boldsymbol{\mu}}_j,\, \boldsymbol{\Upsilon}_j) &\in \operatorname{col}(\boldsymbol{\Phi}_{j,k}) \\ \hat{\boldsymbol{v}}(\boldsymbol{\eta}_{j,k};\, \bar{\boldsymbol{\mu}}_j,\, \boldsymbol{\Upsilon}_j) &\in \operatorname{col}(\boldsymbol{\Phi}_{j,k}). \end{aligned} \tag{C.46}$$

Next, define primal and dual snapshot matrices according to the recursive relationships

$$\begin{aligned} \mathbf{U}_{j,k} &= \begin{bmatrix} \mathbf{U}_{j-1,n_{j-1}} & \boldsymbol{u}(\boldsymbol{\eta}_{j,0},\, \bar{\boldsymbol{\mu}}_j,\, \boldsymbol{\Upsilon}_j) & \cdots & \boldsymbol{u}(\boldsymbol{\eta}_{j,k-1},\, \bar{\boldsymbol{\mu}}_j,\, \boldsymbol{\Upsilon}_j) \end{bmatrix} \\ \hat{\mathbf{V}}_{j,k} &= \begin{bmatrix} \hat{\mathbf{V}}_{j-1,n_{j-1}} & \hat{\boldsymbol{v}}(\boldsymbol{\eta}_{j,0},\, \bar{\boldsymbol{\mu}}_j,\, \boldsymbol{\Upsilon}_j) & \cdots & \hat{\boldsymbol{v}}(\boldsymbol{\eta}_{j,k-1},\, \bar{\boldsymbol{\mu}}_j,\, \boldsymbol{\Upsilon}_j) \end{bmatrix} \end{aligned} \tag{C.47}$$

where $\mathbf{U}_{-1,k} = \emptyset$, $\hat{\mathbf{V}}_{-1,k} = \emptyset$, and n_j is the number of inner iterations corresponding to outer iteration j. Then, the reduced-order basis is defined according to the heterogeneous, span-preserving variant of POD (Algorithm 7) as

$$\boldsymbol{\Phi}_{j,k} = \texttt{PODHSP}(\boldsymbol{u}(\boldsymbol{\eta}_{j,k},\, \bar{\boldsymbol{\mu}}_j,\, \boldsymbol{\Upsilon}_j),\, \mathbf{U}_{j,k},\, \hat{\boldsymbol{v}}(\boldsymbol{\eta}_{j,k},\, \bar{\boldsymbol{\mu}}_j,\, \boldsymbol{\Upsilon}_j),\, \hat{\mathbf{V}}_{j,k}). \tag{C.48}$$

By construction, the basis satisfies (C.44) and possesses additional information to improve the parametric robustness of the reduced-order model. The complete inner iteration algorithm is provided in Algorithm 18.

Algorithm 18 Inner iteration: trust region method based on reduced-order models in reduced parameter space

1: **Initialization**: Given

$$\bar{\boldsymbol{\mu}}_j,\ \boldsymbol{\Upsilon}_j,\ \boldsymbol{\eta}_{j,0},\ \mathbf{U}_{j-1,n_{j-1}},\ \hat{\mathbf{V}}_{j-1,n_{j-1}},\ \Delta_{j,0},\ 0<\gamma<1,\ \Delta_{\max}>0,\ 0<\eta_1<\eta_2<1,$$
$$0<\kappa_\vartheta<1,\ 0<\kappa_\varphi,\ 0<\omega<1,\ \{r_k\}_{k=0}^{\infty}\ \text{such that}\ r_k\to 0$$

2: **Model and constraint update**: If previous model and constraint are sufficient for convergence

$$\vartheta_{j,k-1}(\boldsymbol{\eta}_{j,k})\le\kappa_\vartheta\Delta_{j,k}\qquad\qquad \varphi_{j,k-1}(\boldsymbol{\eta}_{j,k})\le\kappa_\varphi\min\{\left\|\nabla m_{j,k-1}(\boldsymbol{\eta}_{j,k})\right\|,\Delta_{j,k}\},$$

re-use for the current iteration: $m_{j,k}(\boldsymbol{\eta}) \coloneqq m_{j,k-1}(\boldsymbol{\eta})$ and $\vartheta_{j,k}(\boldsymbol{\eta}) \coloneqq \vartheta_{j,k-1}(\boldsymbol{\eta})$. Otherwise, evaluate primal and sensitivity or adjoint solution of high-dimensional model

$$\boldsymbol{u}_{j,k}\coloneqq\boldsymbol{u}(\boldsymbol{\mu}_{j,k})\qquad \hat{\boldsymbol{v}}_{j,k}\coloneqq\frac{\partial\boldsymbol{u}}{\partial\boldsymbol{\mu}}(\boldsymbol{\mu}_{j,k})\ \text{ or }\ \boldsymbol{\Theta}^{\boldsymbol{\lambda}}_{j,k}(\boldsymbol{u}(\boldsymbol{\mu}_{j,k}),\boldsymbol{\mu}_{j,k})\frac{\partial\boldsymbol{r}}{\partial\boldsymbol{u}}(\boldsymbol{u}(\boldsymbol{\mu}_{j,k}),\boldsymbol{\mu}_{j,k})^T\boldsymbol{\lambda}(\boldsymbol{\mu}_{j,k})$$

where $\boldsymbol{\mu}_{j,k}=\bar{\boldsymbol{\mu}}_j+\boldsymbol{\Upsilon}_j\boldsymbol{\eta}_{j,k}$ and compute reduced-order basis via span-preserving variant of POD (Algorithm 7)

$$\boldsymbol{\Phi}_{j,k}=\texttt{PODHSP}(\boldsymbol{u}_{j,k},\mathbf{U}_{j,k},\hat{\boldsymbol{v}}_{j,k},\hat{\mathbf{V}}_{j,k}),$$

define model and constraint as

$$\begin{aligned}m_{j,k}(\boldsymbol{\eta})&=f(\boldsymbol{\Phi}_{j,k}\boldsymbol{u}_r(\bar{\boldsymbol{\mu}}_j+\boldsymbol{\Upsilon}_j\boldsymbol{\eta};\boldsymbol{\Phi}_{j,k},\boldsymbol{\Psi}_{j,k}),\bar{\boldsymbol{\mu}}_j+\boldsymbol{\Upsilon}_j\boldsymbol{\eta})\\ \vartheta_{j,k}(\boldsymbol{\eta})&=\left\|\boldsymbol{r}(\boldsymbol{\Phi}_{j,k}\boldsymbol{u}_r(\bar{\boldsymbol{\mu}}_j+\boldsymbol{\Upsilon}_j\boldsymbol{\eta}_{j,k};\boldsymbol{\Phi}_{j,k},\boldsymbol{\Psi}_{j,k}),\bar{\boldsymbol{\mu}}_j+\boldsymbol{\Upsilon}_j\boldsymbol{\eta}_{j,k})\right\|_{\boldsymbol{\Theta}_{j,k}}+\\ &\quad\left\|\boldsymbol{r}(\boldsymbol{\Phi}_{j,k}\boldsymbol{u}_r(\bar{\boldsymbol{\mu}}_j+\boldsymbol{\Upsilon}_j\boldsymbol{\eta};\boldsymbol{\Phi}_{j,k},\boldsymbol{\Psi}_{j,k}),\bar{\boldsymbol{\mu}}_j+\boldsymbol{\Upsilon}_j\boldsymbol{\eta})\right\|_{\boldsymbol{\Theta}_{j,k}},\end{aligned}$$

and update snapshot matrices

$$\mathbf{U}_{j,k+1}\leftarrow\begin{bmatrix}\mathbf{U}_{j-1,n_{j-1}} & \boldsymbol{u}_{j,0} & \cdots & \boldsymbol{u}_{j,k}\end{bmatrix}\qquad \hat{\mathbf{V}}_{j,k+1}\leftarrow\begin{bmatrix}\hat{\mathbf{V}}_{j-1,n_{j-1}} & \hat{\boldsymbol{v}}_{j,0} & \cdots & \hat{\boldsymbol{v}}_{j,k}\end{bmatrix}.$$

3: **Step computation**: Solve (exactly) the trust region subproblem

$$\min_{\boldsymbol{\eta}\in\mathbb{R}^{k_{\boldsymbol{\mu}}}} m_{j,k}(\boldsymbol{\eta})\qquad\text{subject to}\qquad \vartheta_{j,k}(\boldsymbol{\eta})\le\Delta_{j,k}$$

for a candidate, $\hat{\boldsymbol{\eta}}_{j,k}$, using interior-point method of Section 3.1.2.

4: **Actual-to-predicted reduction**: Compute actual-to-predicted reduction ratio

$$\rho_{j,k}=\begin{cases}1 & \text{if}\quad \vartheta_{j,k}(\hat{\boldsymbol{\eta}}_{j,k})^\omega\le\eta\min\{m_{j,k}(\boldsymbol{\eta}_{j,k})-m_{j,k}(\hat{\boldsymbol{\eta}}_{j,k}),r_k\}\\ \dfrac{F(\bar{\boldsymbol{\mu}}_j+\boldsymbol{\Upsilon}_j\boldsymbol{\eta}_{j,k})-F(\bar{\boldsymbol{\mu}}_j+\boldsymbol{\Upsilon}_j\hat{\boldsymbol{\eta}}_{j,k})}{m_{j,k}(\boldsymbol{\eta}_{j,k})-m_{j,k}(\hat{\boldsymbol{\eta}}_{j,k})} & \text{otherwise}\end{cases}$$

where $\eta<\min\{\eta_1,1-\eta_2\}$

5: **Step acceptance**:

$$\textbf{if}\quad \rho_{j,k}\ge\eta_1\quad\textbf{then}\quad \boldsymbol{\eta}_{j,k+1}=\hat{\boldsymbol{\eta}}_{j,k}\quad\textbf{else}\quad \boldsymbol{\eta}_{j,k+1}=\boldsymbol{\eta}_{j,k}\quad\textbf{end if}$$

6: **Trust region update**:

$$\begin{aligned}&\textbf{if}\quad \rho_{j,k}\le\eta_1\quad&&\textbf{then}\quad \Delta_{k+1}\in(0,\gamma\vartheta_{j,k}(\hat{\boldsymbol{\eta}}_{j,k})]&&\textbf{end if}\\ &\textbf{if}\quad \rho_{j,k}\in(\eta_1,\eta_2)\quad&&\textbf{then}\quad \Delta_{k+1}\in[\gamma\vartheta_{j,k}(\hat{\boldsymbol{\eta}}_{j,k}),\Delta_{j,k}]&&\textbf{end if}\\ &\textbf{if}\quad \rho_{j,k}\ge\eta_2\quad&&\textbf{then}\quad \Delta_{k+1}\in[\Delta_{j,k},\Delta_{\max}]&&\textbf{end if}\end{aligned}$$

Appendix D

Time-Dependent PDE-Constrained Optimization under Periodicity Constraints

This appendix summarizes the work in [211, 212].

D.1 Governing Equations and Discretization

This section is devoted to the treatment of conservation laws (2.9) on a *parametrized, deforming domain* using an Arbitrary Lagrangian-Eulerian (ALE) description of the governing equations and a brief discussion of a globally high-order numerical discretization of the ALE form of the system of conservation laws that closely parallels that in Chapter 2. Subsequently, Section D.2 will develop the corresponding fully discrete adjoint equations and the adjoint method for constructing gradients of quantities of interest.

The methods introduced in this work are not necessarily limited to Partial Differential Equations (PDE) that can be written as conservation laws (D.1). In Section D.1.2, the chosen spatial discretization (discontinuous Galerkin Arbitrary Lagrangian-Eulerian method) is applied to the PDE, resulting in a system of first-order ODEs, which is the point of departure for all adjoint-related derivations. Time-dependent PDEs that are not conservation laws can be written similarly at the semi-discrete level after application of an appropriate spatial discretization, e.g., a continuous finite element method for parabolic PDEs. In this work, the scope is limited to first-order temporal systems, or those which are recast as such.

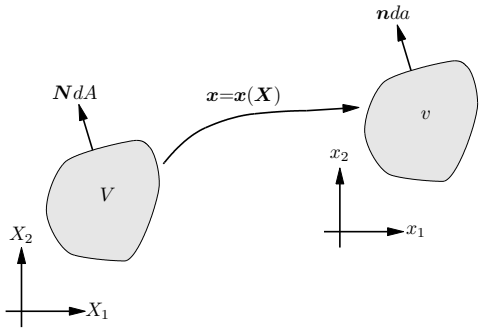


Figure D.1: Time-dependent mapping between reference and physical domains.

D.1.1 System of Conservation Laws on Deforming Domain: Arbitrary Lagrangian-Eulerian Description

Consider a general system of conservation laws, defined on a parametrized, deforming domain, $v(\boldsymbol{\mu}, t)$, written at the continuous level as

$$\frac{\partial \boldsymbol{U}}{\partial t} + \nabla \cdot \boldsymbol{F}(\boldsymbol{U},\ \nabla \boldsymbol{U}) = 0 \quad \text{in} \quad v(\boldsymbol{\mu}, t) \tag{D.1}$$

where the physical flux is decomposed into an inviscid and a viscous part $\boldsymbol{F}(\boldsymbol{U},\ \nabla \boldsymbol{U}) = \boldsymbol{F}^{inv}(\boldsymbol{U}) + \boldsymbol{F}^{vis}(\boldsymbol{U},\ \nabla \boldsymbol{U})$, $\boldsymbol{U}(\boldsymbol{x}, \boldsymbol{\mu}, t)$ is the solution of the system of conservation laws, $t \in (0, T)$ represents time, and $\boldsymbol{\mu} \in \mathbb{R}^{N_\mu}$ is a vector of parameters. This work will focus on the case where the *domain* is parametrized by $\boldsymbol{\mu}$, although extension to other types of parameters, e.g., constants defining the conservation law, is straightforward. The conservation law on a deforming domain is transformed into a conservation law on a *fixed* reference domain through the introduction of a time-dependent mapping between the physical and reference domains, resulting in an Arbitrary Lagrangian-Eulerian description of the governing equations.

Denote the physical domain by $v(\boldsymbol{\mu}, t) \subset \mathbb{R}^{n_{sd}}$ and the fixed, reference domain by $V \subset \mathbb{R}^{n_{sd}}$, where n_{sd} is the number of spatial dimensions. At each time t, let $\mathcal{G}$ be a time-dependent diffeomorphism between the reference domain and physical domain: $\boldsymbol{x}(\boldsymbol{X}, \boldsymbol{\mu}, t) = \mathcal{G}(\boldsymbol{X}, \boldsymbol{\mu}, t)$, where $\boldsymbol{X} \in V$ is a point in the reference domain and $\boldsymbol{x}(\boldsymbol{X}, \boldsymbol{\mu}, t) \in v(\boldsymbol{\mu}, t)$ is the corresponding point in the physical domain at time t and parameter configuration $\boldsymbol{\mu}$.

The transformed system of conservation laws from (D.1), under the mapping $\mathcal{G}$, defined on the reference domain takes the form

$$\left.\frac{\partial \boldsymbol{U}_{\boldsymbol{X}}}{\partial t}\right|_{\boldsymbol{X}} + \nabla_{\boldsymbol{X}} \cdot \boldsymbol{F}_{\boldsymbol{X}}(\boldsymbol{U}_{\boldsymbol{X}},\ \nabla_{\boldsymbol{X}} \boldsymbol{U}_{\boldsymbol{X}}) = 0 \tag{D.2}$$

where $\nabla_{\boldsymbol{X}}$ denotes spatial derivatives with respect to the reference variables, $\boldsymbol{X}$. The transformed state vector, $\boldsymbol{U}_{\boldsymbol{X}}$, and its corresponding spatial gradient with respect to the reference configuration

take the form

$$\boldsymbol{U}_{\boldsymbol{X}} = g\boldsymbol{U}, \qquad \nabla_{\boldsymbol{X}}\boldsymbol{U}_{\boldsymbol{X}} = g^{-1}\boldsymbol{U}_{\boldsymbol{X}}\frac{\partial g}{\partial \boldsymbol{X}} + g\nabla\boldsymbol{U}\cdot\boldsymbol{G}, \tag{D.3}$$

where $\boldsymbol{G} = \nabla_{\boldsymbol{X}}\mathcal{G}$, $g = \det(\boldsymbol{G})$, $\boldsymbol{v}_{\boldsymbol{G}} = \dfrac{\partial \boldsymbol{x}}{\partial t} = \dfrac{\partial \mathcal{G}}{\partial t}$, and the arguments have been dropped, for brevity. The transformed fluxes are

$$\begin{aligned}
\boldsymbol{F}_{\boldsymbol{X}}(\boldsymbol{U}_{\boldsymbol{X}}, \nabla_{\boldsymbol{X}}\boldsymbol{U}_{\boldsymbol{X}}) &= \boldsymbol{F}^{inv}_{\boldsymbol{X}}(\boldsymbol{U}_{\boldsymbol{X}}) + \boldsymbol{F}^{vis}_{\boldsymbol{X}}(\boldsymbol{U}_{\boldsymbol{X}}, \nabla_{\boldsymbol{X}}\boldsymbol{U}_{\boldsymbol{X}}),\\
\boldsymbol{F}^{inv}_{\boldsymbol{X}}(\boldsymbol{U}_{\boldsymbol{X}}) &= g\boldsymbol{F}^{inv}(g^{-1}\boldsymbol{U}_{\boldsymbol{X}})\boldsymbol{G}^{-T} - \boldsymbol{U}_{\boldsymbol{X}}\otimes\boldsymbol{G}^{-1}\boldsymbol{v}_{\boldsymbol{G}},\\
\boldsymbol{F}^{vis}_{\boldsymbol{X}}(\boldsymbol{U}_{\boldsymbol{X}}, \nabla_{\boldsymbol{X}}\boldsymbol{U}_{\boldsymbol{X}}) &= g\boldsymbol{F}^{vis}\left(g^{-1}\boldsymbol{U}_{\boldsymbol{X}}, g^{-1}\left[\nabla_{\boldsymbol{X}}\boldsymbol{U}_{\boldsymbol{X}} - g^{-1}\boldsymbol{U}_{\boldsymbol{X}}\frac{\partial g}{\partial \boldsymbol{X}}\right]\boldsymbol{G}^{-1}\right)\boldsymbol{G}^{-T}.
\end{aligned} \tag{D.4}$$

For details regarding the derivation of the transformed equations, the reader is referred to [152].

When integrated using inexact numerical schemes, this ALE formulation does not satisfy the Geometric Conservation Law (GCL) [60, 152]. This is overcome by introducing an auxiliary variable $\bar{g}$, defined as the solution of

$$\frac{\partial \bar{g}}{\partial t} - \nabla_{\boldsymbol{X}}\cdot\left(g\boldsymbol{G}^{-1}\boldsymbol{v}_{\boldsymbol{G}}\right) = 0. \tag{D.5}$$

The auxiliary variable, $\bar{g}$ is used to modify the *transformed* conservation law according to

$$\left.\frac{\partial \boldsymbol{U}_{\bar{\boldsymbol{X}}}}{\partial t}\right|_{\boldsymbol{X}} + \nabla_{\boldsymbol{X}}\cdot\boldsymbol{F}_{\bar{\boldsymbol{X}}}(\boldsymbol{U}_{\bar{\boldsymbol{X}}},\ \nabla_{\boldsymbol{X}}\boldsymbol{U}_{\bar{\boldsymbol{X}}}) = 0 \tag{D.6}$$

where the GCL-transformed state variables are

$$\boldsymbol{U}_{\bar{\boldsymbol{X}}} = \bar{g}\boldsymbol{U}, \qquad \nabla_{\boldsymbol{X}}\boldsymbol{U}_{\bar{\boldsymbol{X}}} = \bar{g}^{-1}\boldsymbol{U}_{\bar{\boldsymbol{X}}}\frac{\partial \bar{g}}{\partial \boldsymbol{X}} + \bar{g}\nabla\boldsymbol{U}\cdot\boldsymbol{G} \tag{D.7}$$

and the corresponding fluxes

$$\begin{aligned}
\boldsymbol{F}_{\bar{\boldsymbol{X}}}(\boldsymbol{U}_{\bar{\boldsymbol{X}}}, \nabla_{\boldsymbol{X}}\boldsymbol{U}_{\bar{\boldsymbol{X}}}) &= \boldsymbol{F}^{inv}_{\bar{\boldsymbol{X}}}(\boldsymbol{U}_{\bar{\boldsymbol{X}}}) + \boldsymbol{F}^{vis}_{\bar{\boldsymbol{X}}}(\boldsymbol{U}_{\bar{\boldsymbol{X}}}, \nabla_{\boldsymbol{X}}\boldsymbol{U}_{\bar{\boldsymbol{X}}}),\\
\boldsymbol{F}^{inv}_{\bar{\boldsymbol{X}}}(\boldsymbol{U}_{\bar{\boldsymbol{X}}}) &= g\boldsymbol{F}^{inv}(\bar{g}^{-1}\boldsymbol{U}_{\bar{\boldsymbol{X}}})\boldsymbol{G}^{-T} - \boldsymbol{U}_{\bar{\boldsymbol{X}}}\otimes\boldsymbol{G}^{-1}\boldsymbol{v}_{\boldsymbol{G}},\\
\boldsymbol{F}^{vis}_{\bar{\boldsymbol{X}}}(\boldsymbol{U}_{\bar{\boldsymbol{X}}}, \nabla_{\boldsymbol{X}}\boldsymbol{U}_{\bar{\boldsymbol{X}}}) &= g\boldsymbol{F}^{vis}\left(\bar{g}^{-1}\boldsymbol{U}_{\bar{\boldsymbol{X}}}, \bar{g}^{-1}\left[\nabla_{\boldsymbol{X}}\boldsymbol{U}_{\bar{\boldsymbol{X}}} - \bar{g}^{-1}\boldsymbol{U}_{\bar{\boldsymbol{X}}}\frac{\partial \bar{g}}{\partial \boldsymbol{X}}\right]\boldsymbol{G}^{-1}\right)\boldsymbol{G}^{-T}.
\end{aligned} \tag{D.8}$$

It was shown in [152] that the transformed equations (D.6) satisfy the GCL. In the next section, the ALE description of the governing equations (D.2) and (D.6) will be converted to first-order form and discretized via a high-order discontinuous Galerkin method.

D.1.2 Arbitrary Lagrangian-Eulerian Discontinuous Galerkin Method

The ALE description of the conservation law without GCL augmentation will be considered first. To proceed, the second-order system of partial differential equations in (D.2) is converted to first-order

form

$$\begin{aligned} \left.\frac{\partial \boldsymbol{U}_{\boldsymbol{X}}}{\partial t}\right|_{\boldsymbol{X}} + \nabla_{\boldsymbol{X}} \cdot \boldsymbol{F}_{\boldsymbol{X}}(\boldsymbol{U}_{\boldsymbol{X}},\ \boldsymbol{Q}_{\boldsymbol{X}}) &= 0 \\ \boldsymbol{Q}_{\boldsymbol{X}} - \nabla_{\boldsymbol{X}} \boldsymbol{U}_{\boldsymbol{X}} &= 0, \end{aligned} \tag{D.9}$$

where $\boldsymbol{Q}_{\boldsymbol{X}}$ is introduced as an auxiliary variable to represent the spatial gradient of the $\boldsymbol{U}_{\boldsymbol{X}}$. Equation (D.9) is discretized using a standard nodal discontinuous Galerkin finite element method [46], which, after local elimination of the auxiliary variables $\boldsymbol{Q}_{\boldsymbol{X}}$, leads to the following system of ODEs

$$\boldsymbol{M}_{\boldsymbol{X}} \frac{\partial \boldsymbol{u}_{\boldsymbol{X}}}{\partial t} = \boldsymbol{r}_{\boldsymbol{u}_{\boldsymbol{X}}}(\boldsymbol{u}_{\boldsymbol{X}}, \boldsymbol{\mu}, t), \tag{D.10}$$

where $\boldsymbol{M}_{\boldsymbol{X}}$ is the block-diagonal, symmetric, *fixed* mass matrix, $\boldsymbol{u}_{\boldsymbol{X}}$ is the vectorization of $\boldsymbol{U}_{\boldsymbol{X}}$ at all nodes in the high-order mesh, and $\boldsymbol{r}_{\boldsymbol{u}_{\boldsymbol{X}}}$ is the nonlinear function defining the DG discretization of the inviscid and viscous fluxes.

The GCL augmentation is treated identically, i.e., conversion to first-order form and subsequent application of the discontinuous Galerkin finite element method, where $\boldsymbol{U}_{\bar{\boldsymbol{X}}}$ is taken as the state variable. The result is a system of ODEs corresponding to a high-order ALE scheme that satisfies the GCL

$$\begin{aligned} \boldsymbol{M}_{\bar{\mathbf{g}}} \frac{\partial \bar{\mathbf{g}}}{\partial t} &= \boldsymbol{r}_{\bar{\mathbf{g}}}(\boldsymbol{\mu}, t) \\ \boldsymbol{M}_{\boldsymbol{X}} \frac{\partial \boldsymbol{u}_{\bar{\boldsymbol{X}}}}{\partial t} &= \boldsymbol{r}_{\boldsymbol{u}_{\bar{\boldsymbol{X}}}}(\boldsymbol{u}_{\bar{\boldsymbol{X}}}, \bar{\mathbf{g}}, \boldsymbol{\mu}, t) \end{aligned} \tag{D.11}$$

where each term is defined according to their counterparts in (D.10). From the conservation law defining $\bar{g}$ (D.5), the corresponding flux is continuous, implying the physical flux $g\boldsymbol{G}^{-1}\boldsymbol{v}_{\boldsymbol{G}}$ can be used as the numerical flux. This implies no information is required from neighboring elements and (D.5) can be solved at the element level, i.e., statically condensed. Furthermore, the $\bar{\mathbf{g}}$ residual, $\boldsymbol{r}_{\bar{\mathbf{g}}}$, does not depend on $\bar{\mathbf{g}}$ itself since the physical flux $g\boldsymbol{G}^{-1}\boldsymbol{v}_{\boldsymbol{G}}$ is independent of $\bar{g}$.

Since the equation for $\bar{\mathbf{g}}$ does not depend on $\boldsymbol{u}_{\bar{\boldsymbol{X}}}$, it can be solved independently of the equation for $\boldsymbol{u}_{\bar{\boldsymbol{X}}}$. This enables $\bar{\mathbf{g}}$ to be considered an implicit function of $\boldsymbol{\mu}$, i.e., $\bar{\mathbf{g}} = \bar{\mathbf{g}}(\boldsymbol{\mu}, t)$, through application of the implicit function theorem. Then, (D.11) reduces to

$$\boldsymbol{M}_{\boldsymbol{X}} \frac{\partial \boldsymbol{u}_{\bar{\boldsymbol{X}}}}{\partial t} = \boldsymbol{r}_{\boldsymbol{u}_{\bar{\boldsymbol{X}}}}(\boldsymbol{u}_{\bar{\boldsymbol{X}}}, \bar{\mathbf{g}}(\boldsymbol{\mu}, t), \boldsymbol{\mu}, t). \tag{D.12}$$

Equations (D.10) and (D.12) are abstracted into the following system of ODEs

$$\boldsymbol{M} \frac{\partial \boldsymbol{u}}{\partial t} = \boldsymbol{r}(\boldsymbol{u}, \boldsymbol{\mu}, t), \tag{D.13}$$

for convenience in the derivation of the fully discrete adjoint equations. Evaluation of the residual, $\boldsymbol{r}$, in (D.13) at parameter $\boldsymbol{\mu}$ and time t requires evaluation of the mapping, $\boldsymbol{x}(\boldsymbol{\mu}, t)$ and $\dot{\boldsymbol{x}}(\boldsymbol{\mu}, t)$, and $\bar{\mathbf{g}}(\boldsymbol{\mu}, t)$, if GCL augmentation is employed. The implicit dependence of $\bar{\mathbf{g}}$ on $\boldsymbol{\mu}$ requires special treatment when computing derivatives with respect to $\boldsymbol{\mu}$, which will be required in the adjoint method (Section D.2). Treatment of such terms will be deferred to Section D.2.4.

A convenient property of this DG-ALE scheme is that all computations are performed on the reference domain which is *independent* of time and parameter. This implies that the mass matrix of the ODE (D.13) is also time- and parameter-independent, which simplifies all adjoint computations introduced in Section D.2 as terms involving $\frac{\partial \boldsymbol{M}}{\partial \boldsymbol{u}}$ and $\frac{\partial \boldsymbol{M}}{\partial \boldsymbol{\mu}}$ are identically zero. This, in turn, simplifies the implementation of the adjoint method and translates to computational savings since contractions with these third-order tensor are not required; see [88] for a discretization with parameter-dependent mass matrices and the corresponding adjoint derivation. In subsequent sections, it will be assumed that the mass matrix is time- and parameter-independent.

The DG-ALE scheme outlined in this section constitutes a *spatial* discretization, which yields a system of ODEs when applied to the PDE in (D.1). The semi-discrete form of the conservation law is the point of departure for the remainder of this document. The subsequent development applies to any system of ODEs of the form (D.13) without relying on the specific spatial discretization scheme employed. The DG-ALE scheme was chosen to provide a high-order, stable spatial discretization of the conservation law (D.1).

The diagonally implicit Runge-Kutta scheme introduced in Section 2.1.3 is applied to the system of ODEs for a stable, high-order implicit discretization, repeated here for convenience

$$\begin{aligned} \boldsymbol{u}^{(0)} &= \boldsymbol{u}_0(\boldsymbol{\mu}) \\ \boldsymbol{u}^{(n)} &= \boldsymbol{u}^{(n-1)} + \sum_{i=1}^{s} b_i \boldsymbol{k}_i^{(n)} \\ \boldsymbol{M}\boldsymbol{k}_i^{(n)} &= \Delta t_n \boldsymbol{r}\left(\boldsymbol{u}_i^{(n)},\ \boldsymbol{\mu},\ t_{n-1} + c_i \Delta t_n\right), \end{aligned} \tag{D.14}$$

for $n = 1, \ldots, N_t$ and $i = 1, \ldots, s$, where N_t are the number of time steps in the temporal discretization and s is the number of stages in the DIRK scheme. The temporal domain, $[0,\ T]$ is discretized into N_t segments with endpoints $\{t_0, t_1, \ldots, t_{N_t}\}$, with the nth segment having length $\Delta t_n = t_n - t_{n-1}$ for $n = 1, \ldots, N_t$. Additionally, in (D.14), $\boldsymbol{u}_i^{(n)}$ is used to denote the approximation of $\boldsymbol{u}^{(n)}$ at the ith stage of time step n

$$\boldsymbol{u}_i^{(n)} = \boldsymbol{u}_i^{(n)}(\boldsymbol{u}^{(n-1)},\ \boldsymbol{k}_1^{(n)}, \ldots,\ \boldsymbol{k}_s^{(n)}) = \boldsymbol{u}^{(n-1)} + \sum_{j=1}^{i} a_{ij} \boldsymbol{k}_j^{(n)}. \tag{D.15}$$

From (D.14), a complete time step requires the solution of a sequence of s nonlinear systems of equation of size $N_{\boldsymbol{u}}$.

Finally, a *solver-consistent* discretization (Section 2.1.4) is applied to discretize output quantities of interest that take the form

$$\mathcal{F}(\boldsymbol{U}, \boldsymbol{\mu}, t) = \int_0^t \int_{\boldsymbol{\Gamma}} f(\boldsymbol{U}, \boldsymbol{\mu}, \tau)\ dS\ d\tau \tag{D.16}$$

to yield the update equations in (2.45) and the fully discrete quantity of interest

$$F(\boldsymbol{u}^{(0)}, \ldots, \boldsymbol{u}^{(N_t)}, \boldsymbol{k}_1^{(1)}, \ldots, \boldsymbol{k}_s^{(N_t)})$$

in (2.46). The generalization to other types of quantities of interest, such as volumetric integrals and instantaneous or pointwise quantities of interest, is immediate as the specific form of the quantity of interest will be abstracted away at the fully discrete level. The form in (D.16) will be used in the physical setup of the applications in Sections D.2.5–D.2.6.

D.2 Fully Discrete, Time-Dependent Adjoint Equations

The purpose of this section is to derive an expression for the total derivative of the discrete quantity of interest F in (2.46), which can be expanded as

$$\frac{\mathrm{d}F}{\mathrm{d}\boldsymbol{\mu}} = \frac{\partial F}{\partial \boldsymbol{\mu}} + \sum_{n=0}^{N_t} \frac{\partial F}{\partial \boldsymbol{u}^{(n)}} \frac{\partial \boldsymbol{u}^{(n)}}{\partial \boldsymbol{\mu}} + \sum_{n=1}^{N_t} \sum_{i=1}^{s} \frac{\partial F}{\partial \boldsymbol{k}_i^{(n)}} \frac{\partial \boldsymbol{k}_i^{(n)}}{\partial \boldsymbol{\mu}}, \tag{D.17}$$

that depends on the sensitivities of the state variables, $\frac{\partial \boldsymbol{u}^{(n)}}{\partial \boldsymbol{\mu}}$ and $\frac{\partial \boldsymbol{k}_i^{(n)}}{\partial \boldsymbol{\mu}}$. Each of the $N_{\boldsymbol{\mu}}$ state variable sensitivities is the solution of a linear evolution equation of the same dimension and number of steps as the primal equation (D.14), rendering these quantities intractable to compute when $N_{\boldsymbol{\mu}}$ is large. Elimination of the state variable sensitivities from (D.17) is accomplished through introduction of the adjoint equations corresponding to the functional F, and the corresponding dual variables. From the derivation of the adjoint equation in Section D.4.1, an expression for the reconstruction of the gradient of F, independent of the state variables sensitivities, follows naturally. At this point, it is emphasized that F represents *any* quantity of interest whose gradient is desired, such as the optimization objective function or a constraint. This section concludes with a discussion of the advantages of the fully discrete framework in the setting of the high-order numerical scheme.

Before proceeding to the derivation of the adjoint method, the following definitions are introduced for the Runge-Kutta stage equations and state updates

$$\begin{aligned}
\tilde{\boldsymbol{r}}^{(0)}(\boldsymbol{u}^{(0)},\ \boldsymbol{\mu}) &= \boldsymbol{u}^{(0)} - \boldsymbol{u}_0(\boldsymbol{\mu}) = 0 \\
\tilde{\boldsymbol{r}}^{(n)}(\boldsymbol{u}^{(n-1)},\ \boldsymbol{u}^{(n)},\ \boldsymbol{k}_1^{(n)}, \ldots, \boldsymbol{k}_s^{(n)},\ \boldsymbol{\mu}) &= \boldsymbol{u}^{(n)} - \boldsymbol{u}^{(n-1)} - \sum_{i=1}^{s} b_i \boldsymbol{k}_i^{(i)} = 0 \\
\boldsymbol{R}_i^{(n)}(\boldsymbol{u}^{(n-1)}, \boldsymbol{k}_1^{(n)}, \ldots, \boldsymbol{k}_i^{(n)}, \boldsymbol{\mu}) &= \boldsymbol{M}\boldsymbol{k}_i^{(n)} - \Delta t_n \boldsymbol{r}\left(\boldsymbol{u}_i^{(n)},\ \boldsymbol{\mu},\ t_{n-1} + c_i \Delta t_n\right) = 0
\end{aligned} \tag{D.18}$$

for $n = 1, \ldots, n$ and $i = 1, \ldots, s$. Differentiation of these expressions with respect to $\boldsymbol{\mu}$ gives rise to

the *fully discrete sensitivity equations*

$$\begin{aligned}
\frac{\partial \tilde{\boldsymbol{r}}^{(0)}}{\partial \boldsymbol{\mu}} + \frac{\partial \tilde{\boldsymbol{r}}^{(0)}}{\partial \boldsymbol{u}^{(0)}} \frac{\partial \boldsymbol{u}^{(0)}}{\partial \boldsymbol{\mu}} &= 0 \\
\frac{\partial \tilde{\boldsymbol{r}}^{(n)}}{\partial \boldsymbol{\mu}} + \frac{\partial \tilde{\boldsymbol{r}}^{(n)}}{\partial \boldsymbol{u}^{(n)}} \frac{\partial \boldsymbol{u}^{(n)}}{\partial \boldsymbol{\mu}} + \frac{\partial \tilde{\boldsymbol{r}}^{(n)}}{\partial \boldsymbol{u}^{(n-1)}} \frac{\partial \boldsymbol{u}^{(n-1)}}{\partial \boldsymbol{\mu}} + \sum_{p=1}^{s} \frac{\partial \tilde{\boldsymbol{r}}^{(n)}}{\partial \boldsymbol{k}_p^{(n)}} \frac{\partial \boldsymbol{k}_p^{(n)}}{\partial \boldsymbol{\mu}} &= 0 \\
\frac{\partial \boldsymbol{R}_i^{(n)}}{\partial \boldsymbol{\mu}} + \frac{\partial \boldsymbol{R}_i^{(n)}}{\partial \boldsymbol{u}^{(n-1)}} \frac{\partial \boldsymbol{u}^{(n-1)}}{\partial \boldsymbol{\mu}} + \sum_{j=1}^{i} \frac{\partial \boldsymbol{R}_i^{(n)}}{\partial \boldsymbol{k}_j^{(n)}} \frac{\partial \boldsymbol{k}_j^{(n)}}{\partial \boldsymbol{\mu}} &= 0
\end{aligned} \tag{D.19}$$

where $n = 1, \ldots, N_t$, $i = 1, \ldots, s$, and arguments have been dropped.

D.2.1 Derivation

The derivation of the fully discrete adjoint equations corresponding to the quantity of interest, F, begins with the introduction of test variables

$$\boldsymbol{\lambda}^{(0)},\ \boldsymbol{\lambda}^{(n)},\ \boldsymbol{\kappa}_i^{(n)} \in \mathbb{R}^{N_{\boldsymbol{u}}} \tag{D.20}$$

for $n = 1, \ldots, N_t$ and $i = 1, \ldots, s$. To eliminate the state sensitivities from the expression for $\dfrac{\mathrm{d}F}{\mathrm{d}\boldsymbol{\mu}}$ in (D.17), multiply the sensitivity equations (D.19) by the test variables, integrate (sum in the discrete setting) over the time domain, and subtract from the expression for the gradient in (D.17) to obtain

$$\begin{aligned}
\frac{\mathrm{d}F}{\mathrm{d}\boldsymbol{\mu}} = \frac{\partial F}{\partial \boldsymbol{\mu}} &+ \sum_{n=0}^{N_t} \frac{\partial F}{\partial \boldsymbol{u}^{(n)}} \frac{\partial \boldsymbol{u}^{(n)}}{\partial \boldsymbol{\mu}} + \sum_{n=1}^{N_t} \sum_{i=1}^{s} \frac{\partial F}{\partial \boldsymbol{k}_i^{(n)}} \frac{\partial \boldsymbol{k}_i^{(n)}}{\partial \boldsymbol{\mu}} - {\boldsymbol{\lambda}^{(0)}}^T \left[\frac{\partial \tilde{\boldsymbol{r}}^{(0)}}{\partial \boldsymbol{\mu}} + \frac{\partial \tilde{\boldsymbol{r}}^{(0)}}{\partial \boldsymbol{u}^{(0)}} \frac{\partial \boldsymbol{u}^{(0)}}{\partial \boldsymbol{\mu}} \right] \\
&- \sum_{n=1}^{N_t} {\boldsymbol{\lambda}^{(n)}}^T \left[\frac{\partial \tilde{\boldsymbol{r}}^{(n)}}{\partial \boldsymbol{\mu}} + \frac{\partial \tilde{\boldsymbol{r}}^{(n)}}{\partial \boldsymbol{u}^{(n)}} \frac{\partial \boldsymbol{u}^{(n)}}{\partial \boldsymbol{\mu}} + \frac{\partial \tilde{\boldsymbol{r}}^{(n)}}{\partial \boldsymbol{u}^{(n-1)}} \frac{\partial \boldsymbol{u}^{(n-1)}}{\partial \boldsymbol{\mu}} + \sum_{p=1}^{s} \frac{\partial \tilde{\boldsymbol{r}}^{(n)}}{\partial \boldsymbol{k}_p^{(n)}} \frac{\partial \boldsymbol{k}_p^{(n)}}{\partial \boldsymbol{\mu}} \right] \\
&- \sum_{n=1}^{N_t} \sum_{i=1}^{s} {\boldsymbol{\kappa}_i^{(n)}}^T \left[\frac{\partial \boldsymbol{R}_i^{(n)}}{\partial \boldsymbol{\mu}} + \frac{\partial \boldsymbol{R}_i^{(n)}}{\partial \boldsymbol{u}^{(n-1)}} \frac{\partial \boldsymbol{u}^{(n-1)}}{\partial \boldsymbol{\mu}} + \sum_{j=1}^{i} \frac{\partial \boldsymbol{R}_i^{(n)}}{\partial \boldsymbol{k}_j^{(n)}} \frac{\partial \boldsymbol{k}_j^{(n)}}{\partial \boldsymbol{\mu}} \right].
\end{aligned} \tag{D.21}$$

The right side of the equality in (D.21) is an equivalent expression for $\dfrac{\mathrm{d}F}{\mathrm{d}\boldsymbol{\mu}}$ *for any value of the test variables* since the terms in the brackets are zero, i.e., the sensitivity equations. Re-arrangement of terms in (D.21) leads to the following expression for $\dfrac{\mathrm{d}F}{\mathrm{d}\boldsymbol{\mu}}$, where the state variable sensitivities have

been isolated

$$\begin{aligned}\frac{\mathrm{d}F}{\mathrm{d}\boldsymbol{\mu}} = \frac{\partial F}{\partial \boldsymbol{\mu}} &+ \left[\frac{\partial F}{\partial \boldsymbol{u}^{(N_t)}} - \boldsymbol{\lambda}^{(N_t)^T}\frac{\partial \tilde{\boldsymbol{r}}^{(N_t)}}{\partial \boldsymbol{u}^{(N_t)}}\right]\frac{\partial \boldsymbol{u}^{(N_t)}}{\partial \boldsymbol{\mu}} - \sum_{n=0}^{N_t}\boldsymbol{\lambda}^{(n)^T}\frac{\partial \tilde{\boldsymbol{r}}^{(n)}}{\partial \boldsymbol{\mu}} - \sum_{n=1}^{N_t}\sum_{p=1}^{s}\boldsymbol{\kappa}_p^{(n)^T}\frac{\partial \boldsymbol{R}_p^{(n)}}{\partial \boldsymbol{\mu}} \\ &+ \sum_{n=1}^{N_t}\left[\frac{\partial F}{\partial \boldsymbol{u}^{(n-1)}} - \boldsymbol{\lambda}^{(n-1)^T}\frac{\partial \tilde{\boldsymbol{r}}^{(n-1)}}{\partial \boldsymbol{u}^{(n-1)}} - \boldsymbol{\lambda}^{(n)^T}\frac{\partial \tilde{\boldsymbol{r}}^{(n)}}{\partial \boldsymbol{u}^{(n-1)}} - \sum_{i=1}^{s}\boldsymbol{\kappa}_i^{(n)^T}\frac{\partial \boldsymbol{R}_i^{(n)}}{\partial \boldsymbol{u}^{(n-1)}}\right]\frac{\partial \boldsymbol{u}^{(n-1)}}{\partial \boldsymbol{\mu}} \\ &+ \sum_{n=1}^{N_t}\sum_{p=1}^{s}\left[\frac{\partial F}{\partial \boldsymbol{k}_p^{(n)}} - \boldsymbol{\lambda}^{(n)^T}\frac{\partial \tilde{\boldsymbol{r}}^{(n)}}{\partial \boldsymbol{k}_p^{(n)}} - \sum_{i=p}^{s}\boldsymbol{\kappa}_i^{(n)^T}\frac{\partial \boldsymbol{R}_i^{(n)}}{\partial \boldsymbol{k}_p^{(n)}}\right]\frac{\partial \boldsymbol{k}_p^{(n)}}{\partial \boldsymbol{\mu}}.\end{aligned} \tag{D.22}$$

The dual variables, $\boldsymbol{\lambda}^{(n)}$ and $\boldsymbol{\kappa}_i^{(n)}$, which have remained arbitrary to this point, are chosen as the solution to the following equations

$$\begin{aligned}\frac{\partial \tilde{\boldsymbol{r}}^{(N_t)}}{\partial \boldsymbol{u}^{(N_t)}}^T\boldsymbol{\lambda}^{(N_t)} &= \frac{\partial F}{\partial \boldsymbol{u}^{(N_t)}} \\ \frac{\partial \tilde{\boldsymbol{r}}^{(n)}}{\partial \boldsymbol{u}^{(n-1)}}^T\boldsymbol{\lambda}^{(n)} + \frac{\partial \tilde{\boldsymbol{r}}^{(n-1)}}{\partial \boldsymbol{u}^{(n-1)}}^T\boldsymbol{\lambda}^{(n-1)} &= \frac{\partial F}{\partial \boldsymbol{u}^{(n-1)}}^T - \sum_{i=1}^{s}\frac{\partial \boldsymbol{R}_i^{(n)}}{\partial \boldsymbol{u}^{(n-1)}}^T\boldsymbol{\kappa}_i^{(n)} \\ \sum_{j=i}^{s}\frac{\partial \boldsymbol{R}_j^{(n)}}{\partial \boldsymbol{k}_i^{(n)}}^T\boldsymbol{\kappa}_j^{(n)} &= \frac{\partial F}{\partial \boldsymbol{k}_i^{(n)}} - \frac{\partial \tilde{\boldsymbol{r}}^{(n)}}{\partial \boldsymbol{k}_i^{(n)}}^T\boldsymbol{\lambda}^{(n)}\end{aligned} \tag{D.23}$$

for $n = 1, \ldots, N_t$ and $i = 1, \ldots, s$. These are the *fully discrete adjoint equations* corresponding to the primal evolution equations in (D.18) and quantity of interest F. Defining the dual variables as the solution of the adjoint equations in (D.23), the expression for $\dfrac{\mathrm{d}F}{\mathrm{d}\boldsymbol{\mu}}$ in (D.22) reduces to

$$\frac{\mathrm{d}F}{\mathrm{d}\boldsymbol{\mu}} = \frac{\partial F}{\partial \boldsymbol{\mu}} - \sum_{n=0}^{N_t}\boldsymbol{\lambda}^{(n)^T}\frac{\partial \tilde{\boldsymbol{r}}^{(n)}}{\partial \boldsymbol{\mu}} - \sum_{n=1}^{N_t}\sum_{p=1}^{s}\boldsymbol{\kappa}_p^{(n)^T}\frac{\partial \boldsymbol{R}_p^{(n)}}{\partial \boldsymbol{\mu}}, \tag{D.24}$$

which is *independent* of the state sensitivities. Finally, elimination of the auxiliary variables, $\tilde{\boldsymbol{r}}^{(n)}$ and $\boldsymbol{R}_i^{(n)}$, in equations (D.23) and (D.24) through differentiation of their expressions in (D.18) gives rise to the adjoint equations

$$\begin{aligned}\boldsymbol{\lambda}^{(N_t)} &= \frac{\partial F}{\partial \boldsymbol{u}^{(N_t)}}^T \\ \boldsymbol{\lambda}^{(n-1)} &= \boldsymbol{\lambda}^{(n)} + \frac{\partial F}{\partial \boldsymbol{u}^{(n-1)}}^T + \sum_{i=1}^{s}\Delta t_n\frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}\left(\boldsymbol{u}_i^{(n)},\ \boldsymbol{\mu},\ t_{n-1} + c_i\Delta t_n\right)^T\boldsymbol{\kappa}_i^{(n)} \\ \boldsymbol{M}^T\boldsymbol{\kappa}_i^{(n)} &= \frac{\partial F}{\partial \boldsymbol{k}_i^{(n)}}^T + b_i\boldsymbol{\lambda}^{(n)} + \sum_{j=i}^{s}a_{ji}\Delta t_n\frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}\left(\boldsymbol{u}_j^{(n)},\ \boldsymbol{\mu},\ t_{n-1} + c_j\Delta t_n\right)^T\boldsymbol{\kappa}_j^{(n)}\end{aligned} \tag{D.25}$$

for $n = 1, \ldots, N_t$ and $i = 1, \ldots, s$ and the expression for gradient reconstruction, independent of state sensitivities,

$$\frac{\mathrm{d}F}{\mathrm{d}\boldsymbol{\mu}} = \frac{\partial F}{\partial \boldsymbol{\mu}} + \boldsymbol{\lambda}^{(0)^T} \frac{\partial \boldsymbol{u}_0}{\partial \boldsymbol{\mu}} + \sum_{n=1}^{N_t} \Delta t_n \sum_{i=1}^{s} \boldsymbol{\kappa}_i^{(n)^T} \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{\mu}}(\boldsymbol{u}_i^{(n)},\ \boldsymbol{\mu},\ t_{n-1} + c_i \Delta t_n), \tag{D.26}$$

specialized to the case of a DIRK temporal discretization. From inspection of (D.26), it is clear that the initial condition sensitivity $\frac{\partial \boldsymbol{u}_0}{\partial \boldsymbol{\mu}}$ is the only sensitivity term required to reconstruct $\frac{\mathrm{d}F}{\mathrm{d}\boldsymbol{\mu}}$. The presence of this term does not destroy the efficiency of the adjoint method for two reasons: (a) only matrix-vector products with $\frac{\partial \boldsymbol{u}_0}{\partial \boldsymbol{\mu}}^T$ are required and (b) the parametrization of the initial condition is either known analytically (uniform flow, zero freestream, independent of $\boldsymbol{\mu}$, etc) or is the solution of some nonlinear system of equations (most likely the steady-state equations). In the first case, $\boldsymbol{\lambda}^{(0)^T} \frac{\partial \boldsymbol{u}_0}{\partial \boldsymbol{\mu}}$ can be computed analytically once $\boldsymbol{\lambda}^{(0)}$ is known. The next section details efficient computation of $\boldsymbol{\lambda}^{(0)^T} \frac{\partial \boldsymbol{u}_0}{\partial \boldsymbol{\mu}}$ using the adjoint method of the steady-state problem.

D.2.2 Parametrization of Initial Condition

Suppose the initial condition $\boldsymbol{u}_0(\boldsymbol{\mu})$ is defined as the solution of the nonlinear system of equations—whose Jacobian is invertible at $\boldsymbol{u}_0(\boldsymbol{\mu})$—which is most likely the fully discrete steady-state form of the governing equations

$$\boldsymbol{R}(\boldsymbol{u}_0(\boldsymbol{\mu}), \boldsymbol{\mu}) = 0. \tag{D.27}$$

Differentiating with respect to the parameter $\boldsymbol{\mu}$ leads to the expansion

$$\frac{\mathrm{d}\boldsymbol{R}}{\mathrm{d}\boldsymbol{\mu}} = \frac{\partial \boldsymbol{R}}{\partial \boldsymbol{\mu}} + \frac{\partial \boldsymbol{R}}{\partial \boldsymbol{u}_0} \frac{\partial \boldsymbol{u}_0}{\partial \boldsymbol{\mu}} = 0, \tag{D.28}$$

where arguments have been dropped for brevity. Assuming the Jacobian matrix is invertible, multiply the preceding equation by the $\boldsymbol{\lambda}^{(0)}$ and rearrange to obtain

$$-\boldsymbol{\lambda}^{(0)^T} \frac{\partial \boldsymbol{u}_0}{\partial \boldsymbol{\mu}} = \left[\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{u}_0}^{-T} \boldsymbol{\lambda}^{(0)} \right]^T \frac{\partial \boldsymbol{R}}{\partial \boldsymbol{\mu}}. \tag{D.29}$$

This reveals the term $\boldsymbol{\lambda}^{(0)^T} \frac{\partial \boldsymbol{u}_0}{\partial \boldsymbol{\mu}}$ can be computed at the cost of one linear system solve of the form $\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{u}_0}^T \boldsymbol{v} = \boldsymbol{\lambda}^{(0)}$ and an inner product $\boldsymbol{v}^T \frac{\partial \boldsymbol{R}}{\partial \boldsymbol{\mu}}$. The only operation whose cost scales with the size of $\boldsymbol{\mu}$ is the evaluation of $\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{\mu}}$ and subsequent inner product. Given this exposition on the fully discrete, time-dependent adjoint method and the discrete adjoint method for computing $\boldsymbol{\lambda}^{(0)^T} \frac{\partial \boldsymbol{u}_0}{\partial \boldsymbol{\mu}}$, a discussion is provided detailing the advantages of the fully discrete framework when computing gradients of output quantities before discussing implementation details in Section D.2.4.

D.2.3 Benefits of Fully Discrete Framework

In the context of optimization, the fully discrete adjoint method is advantageous compared to the continuous or semi-discrete version as it is guaranteed that the resulting derivatives will be consistent with the quantity of interest, F. This emanates from the fact that in the fully discrete setting, the *discretization errors are also differentiated*. This property is practically relevant as convergence guarantees and convergence rates of many black-box optimizers are heavily dependent on consistent gradients of optimization functionals.

Additionally, when Runge-Kutta schemes are chosen for the temporal discretization, the fully discrete framework is particularly advantageous since the *stages* are rarely invariant with respect to the direction of time, that is to say,

$$\nexists i, j \in \{1, \dots, s\} \quad \text{such that} \quad t_{n-1} + c_i \Delta t_n = t_n - c_j \Delta t_n, \tag{D.30}$$

where c is from the Butcher tableau. Temporal invariance of an Runge-Kutta scheme, as defined in (D.30) is significant when computing adjoint variables. During the primal solve, $\boldsymbol{u}$ will be computed at t_n for $n = 1, \dots, N$ and its stage values at $t_{n-1} + c_i \Delta t_n$ for $n = 1, \dots, N$ and $i = 1, \dots, s$. If the same RK scheme is applied to integrate the *semi-discrete* adjoint equations backward in time, the primal solution will be required at $t_n - c_i \Delta t_n$ for $n = 1, \dots, N$ and $i = 1, \dots, s$. Due to condition (D.30), the solution to the primal problem was not computed during the forward solve. Obtaining the primal solution at this time requires interpolation, which complicates the implementation, degrades the accuracy of the computed adjoint variables, and destroys discrete consistency of the computed gradients. This issue does not arise in the fully discrete setting as only terms computed during the primal solve appear in the adjoint equations, by construction.

The next section is devoted to detailing an efficient and modular implementation of the fully discrete adjoint method on deforming domains.

D.2.4 Implementation

Implementation of the fully discrete adjoint method introduced in Section D.2 relies on the computation of the following terms from the spatial discretization

$$\boldsymbol{M}, \boldsymbol{r}, \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}, \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}^T, \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{\mu}}, f_h, \frac{\partial f_h}{\partial \boldsymbol{u}}, \frac{\partial f_h}{\partial \boldsymbol{\mu}}. \tag{D.31}$$

Here, $\boldsymbol{M}$ is the mass matrix of the semi-discrete conservation law, and $\boldsymbol{r}$ is the spatial residual vector with derivatives $\dfrac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}$ (Jacobian) and $\dfrac{\partial \boldsymbol{r}}{\partial \boldsymbol{\mu}}$. As in the previous section, f_h is the discretization of the spatial integral of the output quantity of interest with derivatives $\dfrac{\partial f_h}{\partial \boldsymbol{u}}$ and $\dfrac{\partial f_h}{\partial \boldsymbol{\mu}}$. The mass matrix, spatial flux, Jacobian of spatial flux, and output quantity are standard terms required by an implicit solver and will not be considered further. The Jacobian transpose is explicitly mentioned as additional implementational effort is required when performing parallel matrix transposition. The

derivatives with respect to $\boldsymbol{\mu}$ are rarely required outside adjoint method computations and will be considered further in subsequent sections. As indicated in Section D.1.2, all relevant derivatives of the mass matrix are zero since it is independent of time, parameter, and state variable, which is an artifact of the transformation to a fixed reference domain.

The parallel implementation of all semi-discrete quantities in (D.31) is performed using domain decomposition, where each processor contains a subset of the elements in the mesh, including a halo of elements to be communicated with neighbors [154]. Linear systems of the form

$$\frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}\mathbf{x} = \mathbf{b} \qquad \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}^T \mathbf{x} = \mathbf{b}$$

are solved in parallel using a GMRES solver with a block Incomplete-LU (ILU) preconditioner.

Given the availability of all terms in (D.31), the solution of the primal problem and integration of the output quantity F is given in Algorithm 19. The solution of the corresponding fully discrete adjoint equation, and reconstruction of the gradient of F, is given in Algorithm 20.

Algorithm 19 Primal Solution: Functional Evaluation

Input: Initial condition, $\boldsymbol{u}^{(0)}$; parameter configuration, $\boldsymbol{\mu}$
Output: Integrated output quantity, $F = \mathcal{F}_h^{(N_t)}$, and primal state quantities, $\boldsymbol{u}^{(n)}$ and $\boldsymbol{k}_i^{(n)}$ for $n = 1, \ldots, N_t$ and $i = 1, \ldots, s$

1: Initialize: $\mathcal{F}_h^{(0)} = 0$
2: **for** $n = 1, \ldots, N_t$ **do**
3: **for** $i = 1, \ldots, s$ **do**
4: Solve (D.14) for $\boldsymbol{k}_i^{(n)}$

$$\boldsymbol{M}\boldsymbol{k}_i^{(n)} = \Delta t \boldsymbol{r}\left(\boldsymbol{u}_i^{(n)}, \boldsymbol{\mu}, t_{n-1} + c_i \Delta t\right)$$

 where $\boldsymbol{u}_i^{(n)} = \boldsymbol{u}^{(n-1)} + \sum_{j=1}^{i} a_{ij}\boldsymbol{k}_j^{(n)}$
5: Write $\boldsymbol{k}_i^{(n)}$ to disk
6: **end for**
7: Update $\boldsymbol{u}$ according to (D.14)

$$\boldsymbol{u}^{(n)} = \boldsymbol{u}^{(n-1)} + \sum_{i=1}^{s} b_i \boldsymbol{k}_i^{(n)}$$

8: Update $\mathcal{F}_h$ according to (2.45)

$$\mathcal{F}_h^{(n)} = \mathcal{F}_h^{(n-1)} + \sum_{i=1}^{s} b_i f\left(\boldsymbol{u}_i^{(n)}, \boldsymbol{\mu}, t_{n-1} + c_i \Delta t_n\right)$$

9: Write $\boldsymbol{u}^{(n)}$ to disk
10: **end for**

A well-documented implementational issue corresponding to the unsteady adjoint method pertains to storage and I/O demands. The adjoint equations are solved backward in time and require

Algorithm 20 Dual Solution: Gradient Evaluation

Input: Primal state quantities, $\boldsymbol{u}^{(n)}$ and $\boldsymbol{k}_i^{(n)}$ for $n = 1, \ldots, N_t$ and $i = 1, \ldots, s$; initial condition sensitivity, $\dfrac{\partial \boldsymbol{u}^{(0)}}{\partial \boldsymbol{\mu}}$; parameter configuration, $\boldsymbol{\mu}$

Output: Gradient of integrated output quantity, $\dfrac{\mathrm{d}F}{\mathrm{d}\boldsymbol{\mu}}$, and dual state quantities, $\boldsymbol{\lambda}^{(n)}$ and $\boldsymbol{\kappa}_i^{(n)}$ for $n = 1, \ldots, N_t$ and $i = 1, \ldots, s$

Read primal solution $\boldsymbol{u}^{(N_t)}$ from disk

$$\boldsymbol{\lambda}^{(N_t)} = \frac{\partial F}{\partial \boldsymbol{u}^{(N_t)}}^T$$

Initial gradient of F with partial derivative and initial condition sensitivity

$$\frac{\mathrm{d}F}{\mathrm{d}\boldsymbol{\mu}} = \frac{\partial F}{\partial \boldsymbol{\mu}} + \boldsymbol{\lambda}^{(0)^T} \frac{\partial \boldsymbol{u}_0}{\partial \boldsymbol{\mu}}$$

for $n = N_t, \ldots, 1$ **do**
  Read primal solution $\boldsymbol{u}^{(n-1)}$ from disk
  for $i = s, \ldots, 1$ **do**
    Read primal solution $\boldsymbol{k}_i^{(n)}$ from disk
    Solve (D.25) for $\boldsymbol{\kappa}_i^{(n)}$

$$\boldsymbol{M}^T \boldsymbol{\kappa}_i^{(n)} = \frac{\partial F}{\partial \boldsymbol{k}_i^{(n)}}^T + b_i \boldsymbol{\lambda}^{(n)} + \sum_{j=i}^{s} a_{ji} \Delta t_n \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}} (\boldsymbol{u}_j^{(n)}, \boldsymbol{\mu}, t_{n-1} + c_j \Delta t_n)^T \boldsymbol{\kappa}_j^{(n)}$$

    Update $\dfrac{\mathrm{d}F}{\mathrm{d}\boldsymbol{\mu}}$ according to (D.26)

$$\frac{\mathrm{d}F}{\mathrm{d}\boldsymbol{\mu}} = \frac{\mathrm{d}F}{\mathrm{d}\boldsymbol{\mu}} + \Delta t_n \boldsymbol{\kappa}_i^{(n)^T} \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{\mu}} (\boldsymbol{u}_i^{(n)},\ \boldsymbol{\mu},\ t_{n-1} + c_i \Delta t_n)$$

  end for
  Update $\boldsymbol{\lambda}$ according to (D.25)

$$\boldsymbol{\lambda}^{(n-1)} = \boldsymbol{\lambda}^{(n)} + \frac{\partial F}{\partial \boldsymbol{u}^{(n-1)}}^T + \sum_{i=1}^{s} \Delta t_n \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}} (\boldsymbol{u}_i^{(n)}, \boldsymbol{\mu}, t_i + c_i \Delta t_n)^T \boldsymbol{\kappa}_i^{(n)}$$

end for

the solution of the primal problem at each of the corresponding steps/stages. Therefore, the adjoint computations cannot begin until all primal states have been computed. Additionally, this implies all primal states must be stored since they will be required in *reverse* order during the adjoint computation. For most problems, storing all primal states in memory will be infeasible, requiring disk I/O, which must be performed in parallel to ensure parallel scaling is maintained. There have been a number of strategies to minimize the required I/O operations, such as local-in-time adjoint strategies [206] and checkpointing [40, 90, 95]. For the DG-ALE method in this work, the cost of I/O was not significant compared to the cost of assembly and solving the linearized system of equations.

In this work, the 3DG software [150] was used for the high-order DG-ALE scheme. The temporal discretization and unsteady adjoint method were implemented in the Model Order Reduction Testbed (MORTestbed) [209, 210] code-base, which was used to wrap 3DG such that all data structures, and thus all parallel capabilities, were inherited.

Partial Derivatives of Residuals and Output Quantities

This section details computation of partial derivatives of the residual, $\boldsymbol{r}$, and the output quantity, f_h, with respect to the parameter $\boldsymbol{\mu}$. The DG-ALE discretizations of Section D.1.2, with and without GCL augmentation, are considered separately as the implicit dependence of $\bar{\mathbf{g}}$ on $\boldsymbol{\mu}$ requires special treatment.

Without GCL Augmentation

When the GCL augmentation is not considered, the dependence of $\boldsymbol{r}$ and f_h on the parameter $\boldsymbol{\mu}$ is solely due to the domain parametrization. Therefore, the following expansion of the partial derivatives with respect to $\boldsymbol{\mu}$ is exploited

$$\frac{\partial \boldsymbol{r}}{\partial \boldsymbol{\mu}} = \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{x}}\frac{\partial \boldsymbol{x}}{\partial \boldsymbol{\mu}} + \frac{\partial \boldsymbol{r}}{\partial \dot{\boldsymbol{x}}}\frac{\partial \dot{\boldsymbol{x}}}{\partial \boldsymbol{\mu}} \qquad \frac{\partial f_h}{\partial \boldsymbol{\mu}} = \frac{\partial f_h}{\partial \boldsymbol{x}}\frac{\partial \boldsymbol{x}}{\partial \boldsymbol{\mu}} + \frac{\partial f_h}{\partial \dot{\boldsymbol{x}}}\frac{\partial \dot{\boldsymbol{x}}}{\partial \boldsymbol{\mu}} \tag{D.32}$$

where $\dfrac{\partial \boldsymbol{x}}{\partial \boldsymbol{\mu}}$ and $\dfrac{\partial \dot{\boldsymbol{x}}}{\partial \boldsymbol{\mu}}$ are determined solely from the domain parametrization and the terms

$$\frac{\partial \boldsymbol{r}}{\partial \boldsymbol{x}}, \frac{\partial \boldsymbol{r}}{\partial \dot{\boldsymbol{x}}}, \frac{\partial f_h}{\partial \boldsymbol{x}}, \frac{\partial f_h}{\partial \dot{\boldsymbol{x}}} \tag{D.33}$$

are determined from the form of the governing equations and spatial discretization outlined in Section D.1. From the expressions in (D.32), the terms in (D.33) are not explicitly required in matrix form, rather matrix-vector products with $\dfrac{\partial \boldsymbol{x}}{\partial \boldsymbol{\mu}}$ and $\dfrac{\partial \dot{\boldsymbol{x}}}{\partial \boldsymbol{\mu}}$ from Section D.2.4 are required.

With GCL Augmentation

For the DG-ALE scheme with GCL augmentation, the dependence of $\boldsymbol{r}$ and f on the parameter $\boldsymbol{\mu}$ arises from two sources, the domain parametrization and the implicit dependence of $\bar{\mathbf{g}}$ on $\boldsymbol{\mu}$. Therefore, the chain rule expansions in (D.32) must include an additional term to account for the

dependence of $\bar{\mathbf{g}}$ on $\boldsymbol{\mu}$

$$\frac{\partial \boldsymbol{r}}{\partial \boldsymbol{\mu}} = \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{x}}\frac{\partial \boldsymbol{x}}{\partial \boldsymbol{\mu}} + \frac{\partial \boldsymbol{r}}{\partial \dot{\boldsymbol{x}}}\frac{\partial \dot{\boldsymbol{x}}}{\partial \boldsymbol{\mu}} + \frac{\partial \boldsymbol{r}}{\partial \bar{\mathbf{g}}}\frac{\partial \bar{\mathbf{g}}}{\partial \boldsymbol{\mu}} \qquad \frac{\partial f}{\partial \boldsymbol{\mu}} = \frac{\partial f}{\partial \boldsymbol{x}}\frac{\partial \boldsymbol{x}}{\partial \boldsymbol{\mu}} + \frac{\partial f}{\partial \dot{\boldsymbol{x}}}\frac{\partial \dot{\boldsymbol{x}}}{\partial \boldsymbol{\mu}} + \frac{\partial f}{\partial \bar{\mathbf{g}}}\frac{\partial \bar{\mathbf{g}}}{\partial \boldsymbol{\mu}}. \tag{D.34}$$

Similar to the previous section, the terms $\frac{\partial \boldsymbol{x}}{\partial \boldsymbol{\mu}}$ and $\frac{\partial \dot{\boldsymbol{x}}}{\partial \boldsymbol{\mu}}$ are determined solely from the domain parametrization and

$$\frac{\partial \boldsymbol{r}}{\partial \boldsymbol{x}}, \frac{\partial \boldsymbol{r}}{\partial \dot{\boldsymbol{x}}}, \frac{\partial \boldsymbol{r}}{\partial \bar{\mathbf{g}}}, \frac{\partial f}{\partial \boldsymbol{x}}, \frac{\partial f}{\partial \dot{\boldsymbol{x}}}, \frac{\partial f}{\partial \bar{\mathbf{g}}} \tag{D.35}$$

are determined from the form of the governing equations and spatial discretization in Section D.1. The only remaining term $\frac{\partial \bar{\mathbf{g}}}{\partial \boldsymbol{\mu}}$ is defined as the solution of the following ODE

$$\boldsymbol{M}_{\bar{\mathbf{g}}}\frac{\partial}{\partial t}\left(\frac{\partial \bar{\mathbf{g}}}{\partial \boldsymbol{\mu}}\right) = \frac{\partial \boldsymbol{r}_{\bar{\mathbf{g}}}}{\partial \boldsymbol{\mu}} + \frac{\partial \boldsymbol{r}_{\bar{\mathbf{g}}}}{\partial \bar{\mathbf{g}}}\frac{\partial \bar{\mathbf{g}}}{\partial \boldsymbol{\mu}} = \frac{\partial \boldsymbol{r}_{\bar{\mathbf{g}}}}{\partial \boldsymbol{\mu}}, \tag{D.36}$$

obtained by direct differentiation of (D.11). The last equality uses the fact that $\boldsymbol{r}_{\bar{\mathbf{g}}}$ is independent of $\bar{\mathbf{g}}$, which can be deduced from examination of the governing equation for $\bar{g}$ (D.5). Equation (D.36) is discretized with the same DIRK scheme used for the temporal discretization of the state equation.

Remark. *The special treatment of* $\bar{\mathbf{g}}$ *detailed in this section, including integration of the sensitivity equations (D.36), can be avoided by considering the ODEs in (D.11) directly without leveraging the fact that the* $\bar{\mathbf{g}}$ *equation is independent of* $\boldsymbol{u}_{\bar{\boldsymbol{X}}}$*. This implies the state vector will contain an additional unknown for* $\bar{g}$ *for each DG node. This increases the cost of a primal and dual solve, but simplifies the adjoint derivation and implementation.*

Time-Dependent, Parametrized Domain Deformation

A crucial component of the fully discrete adjoint method on deforming domains is a time-dependent parametrization of the domain, amenable to parallel implementation. A parallel implementation is required as domain deformation will involve operations on the entire computational mesh and will be queried at every stage of each time step of both the primal and dual solves, according to Algorithms 19 and 20. In this work, the domain parametrization is required to be sufficiently general to handle shape deformation, as well as kinematic motion. Additionally, the domain deformation must be sufficiently smooth to ensure sufficient regularity of the transformed solution, and the spatial and temporal derivatives must be analytically available for fast, accurate computation of the deformation gradient, $\boldsymbol{G}$, and velocity, $\boldsymbol{v}_{\boldsymbol{X}}$, of the mapping, $\mathcal{G}$.

The domain deformation will be defined by the superposition of a rigid body motion and a spatially varying deformation. To avoid large mesh velocities at the far-field, which could arise from rigid rotations of the body, the blending maps of [152] are used. First, define a spatial configuration consisting of a rigid body motion ($\boldsymbol{Q}(\boldsymbol{\mu}, t)$, $\boldsymbol{v}(\boldsymbol{\mu}, t)$) and deformation ($\boldsymbol{\varphi}(\boldsymbol{X}, \boldsymbol{\mu}, t)$) to the *reference* domain

$$\boldsymbol{X}' = \boldsymbol{Q}(\boldsymbol{\mu}, t)\boldsymbol{X} + \boldsymbol{v}(\boldsymbol{\mu}, t) + \boldsymbol{\varphi}(\boldsymbol{X}, \boldsymbol{\mu}, t), \tag{D.37}$$

which completely defines the *physical* motion of the body. This physical configuration is blended with the reference configuration according to

$$\boldsymbol{x} = (1 - b(d(\boldsymbol{X})))\boldsymbol{X}' + b(d(\boldsymbol{X}))\boldsymbol{X} \tag{D.38}$$

where $d(\boldsymbol{X}) = \|\boldsymbol{X} - \boldsymbol{X}_0\| - R_0$ is the signed distance from the origin $\boldsymbol{X}_0$ to the circle of radius R_0 centered at $\boldsymbol{X}_0$ and

$$b(s) = \begin{cases} 0, & \text{if } s < 0 \\ 1, & \text{if } s > R_1 \\ r(s/R_1), & \text{otherwise} \end{cases} \tag{D.39}$$

where $r(s) = 3s^2 - 2x^3$ for a cubic blending and $r(s) = 10s^3 - 15s^4 + 6s^5$ for a quintic blending. Spatial blending of this form ensures the desired physical motion of the body, $\boldsymbol{X}'$ is exactly achieved within a radius R_0 of the origin. Further, there is no deformation outside a radius $R_0 + R_1$ of the origin. In the annulus about the origin with inner radius R_0 and outer radius $R_0 + R_1$, the spatial configuration is blended smoothly between these two spatial configurations.

The specific form of $\boldsymbol{Q}(\boldsymbol{\mu}, t)$, $\boldsymbol{v}(\boldsymbol{\mu}, t)$, and $\boldsymbol{\varphi}(\boldsymbol{X}, \boldsymbol{\mu}, t)$ is problem-specific and will be deferred to Sections D.2.5, D.2.6, D.4.4, D.4.5. Assuming these terms are known analytically, the specific form of $\boldsymbol{G} = \dfrac{\partial \boldsymbol{x}}{\partial \boldsymbol{X}}$, $\boldsymbol{v}_{\boldsymbol{X}} = \dot{\boldsymbol{x}} = \dfrac{\partial \boldsymbol{x}}{\partial t}$, $\dfrac{\partial \boldsymbol{x}}{\partial \boldsymbol{\mu}}$, and $\dfrac{\partial \dot{\boldsymbol{x}}}{\partial \boldsymbol{\mu}}$ can be easily computed.

In the next two sections, the high-order numerical discretization of a system of conservation laws and corresponding adjoint method is applied to the isentropic compressible Navier-Stokes equations (2.24)-(2.25) to solve optimal control and shape optimization problems using gradient-based optimization techniques. The DG-ALE scheme introduced in Section D.1 is used for the spatial discretization of the system of conservation laws with polynomial order $p = 3$ and a diagonally implicit Runge-Kutta scheme for the temporal discretization. The DG-ALE scheme uses the Roe flux [169] for the inviscid numerical flux and the Compact DG flux [150] for the viscous numerical flux. The Butcher tableau for the three-stage, third-order DIRK scheme considered in this work is given in Table D.1. The instantaneous quantities of interest for a body, defined by the surface $\boldsymbol{\Gamma}$, take the

Table D.1: Butcher Tableau for 3-stage, 3rd order DIRK scheme [3]
$\alpha = 0.435866521508459$, $\gamma = -\frac{6\alpha^2 - 16\alpha + 1}{4}$, $\omega = \frac{6\alpha^2 - 20\alpha + 5}{4}$.

α	α		
$\frac{1+\alpha}{2}$	$\frac{1+\alpha}{2} - \alpha$	α	
1	γ	ω	α
	γ	ω	α

following form

$$\begin{aligned}
\mathcal{F}_x(\boldsymbol{U},\boldsymbol{\mu},t) &= \int_{\boldsymbol{\Gamma}} \boldsymbol{f}(\boldsymbol{U},\boldsymbol{\mu},t)\cdot\boldsymbol{e}_1 \; dS & \mathcal{F}_y(\boldsymbol{U},\boldsymbol{\mu},t) &= \int_{\boldsymbol{\Gamma}} \boldsymbol{f}(\boldsymbol{U},\boldsymbol{\mu},t)\cdot\boldsymbol{e}_2 \; dS \\
\mathcal{P}(\boldsymbol{U},\boldsymbol{\mu},t) &= \int_{\boldsymbol{\Gamma}} \boldsymbol{f}(\boldsymbol{U},\boldsymbol{\mu},t)\cdot\dot{\boldsymbol{x}} \; dS & \mathcal{P}_x(\boldsymbol{U},\boldsymbol{\mu},t) &= \int_{\boldsymbol{\Gamma}} \dot{x}\boldsymbol{f}(\boldsymbol{U},\boldsymbol{\mu},t)\cdot\boldsymbol{e}_1 \; dS \\
\mathcal{P}_y(\boldsymbol{U},\boldsymbol{\mu},t) &= \int_{\boldsymbol{\Gamma}} \dot{y}\boldsymbol{f}(\boldsymbol{U},\boldsymbol{\mu},t)\cdot\boldsymbol{e}_2 \; dS & \mathcal{P}_\theta(\boldsymbol{U},\boldsymbol{\mu},t) &= -\int_{\boldsymbol{\Gamma}} \dot{\theta}\boldsymbol{f}(\boldsymbol{U},\boldsymbol{\mu},t)\times(\boldsymbol{x}-\boldsymbol{x}_0) \; dS
\end{aligned} \tag{D.40}$$

where $\boldsymbol{f}\in\mathbb{R}^{n_{sd}}$ is the force imparted by the fluid on the body, $\boldsymbol{e}_i$ is the ith canonical basis vector in $\mathbb{R}^{n_{sd}}$, $\boldsymbol{x}$ and $\dot{\boldsymbol{x}}$ are the position and velocity of a point on the surface $\boldsymbol{\Gamma}$, and x, y, θ, $\dot{x}$, $\dot{y}$, $\dot{\theta}$ define the motion of the reference point, $\boldsymbol{x}_0$ (the 1/3-chord of the airfoil, in this case); see Figure D.2. The $\mathcal{F}_x$ and $\mathcal{F}_y$ terms correspond to the total x- and y-directed forces on the body and $\mathcal{P}$ is the total power exerted on the body by the fluid. The total power $\mathcal{P}$ is broken into its translational, $\mathcal{P}_x$ and $\mathcal{P}_y$, and rotational, $\mathcal{P}_\theta$, components. For a 2D rigid body motion, an additive relationship among these terms holds

$$\mathcal{P}(\boldsymbol{U},\boldsymbol{\mu},t) = \mathcal{P}_x(\boldsymbol{U},\boldsymbol{\mu},t) + \mathcal{P}_y(\boldsymbol{U},\boldsymbol{\mu},t) + \mathcal{P}_\theta(\boldsymbol{U},\boldsymbol{\mu},t). \tag{D.41}$$

The negative sign is included in the definition of $\mathcal{P}_\theta$ due to the clockwise definition of θ in Figure D.2. In the remainder of this document, a superscript h will be used to denote the high-order DG approximation to these spatial integrals that constitute the instantaneous quantities of interest, e.g., $\mathcal{P}^h(\boldsymbol{u},\boldsymbol{\mu},t)$ is the high-order approximation of $\mathcal{P}(\boldsymbol{U},\boldsymbol{\mu},t)$, where $\boldsymbol{u}$ is the semi-discrete approximation of $\boldsymbol{U}$. Temporal integration of the instantaneous quantities of interest leads to the integrated quantities of interest

$$\begin{aligned}
\mathcal{J}_x(\boldsymbol{U},\boldsymbol{\mu}) &= \int_0^T\int_{\boldsymbol{\Gamma}} \boldsymbol{f}(\boldsymbol{U},\boldsymbol{\mu},t)\cdot\boldsymbol{e}_1 \; dS\; dt & \mathcal{J}_y(\boldsymbol{U},\boldsymbol{\mu}) &= \int_0^T\int_{\boldsymbol{\Gamma}} \boldsymbol{f}(\boldsymbol{U},\boldsymbol{\mu},t)\cdot\boldsymbol{e}_2 \; dS\; dt \\
\mathcal{W}(\boldsymbol{U},\boldsymbol{\mu}) &= \int_0^T\int_{\boldsymbol{\Gamma}} \boldsymbol{f}(\boldsymbol{U},\boldsymbol{\mu},t)\cdot\dot{\boldsymbol{x}} \; dS\; dt & \mathcal{W}_x(\boldsymbol{U},\boldsymbol{\mu}) &= \int_0^T\int_{\boldsymbol{\Gamma}} \dot{x}\boldsymbol{f}(\boldsymbol{U},\boldsymbol{\mu},t)\cdot\boldsymbol{e}_1 \; dS\; dt \\
\mathcal{W}_y(\boldsymbol{U},\boldsymbol{\mu}) &= \int_0^T\int_{\boldsymbol{\Gamma}} \dot{y}\boldsymbol{f}(\boldsymbol{U},\boldsymbol{\mu},t)\cdot\boldsymbol{e}_2 \; dS\; dt & \mathcal{W}_\theta(\boldsymbol{U},\boldsymbol{\mu}) &= -\int_0^T\int_{\boldsymbol{\Gamma}} \dot{\theta}\boldsymbol{f}(\boldsymbol{U},\boldsymbol{\mu},t)\times(\boldsymbol{x}-\boldsymbol{x}_0) \; dS\; dt
\end{aligned} \tag{D.42}$$

which will be used as optimization functionals in subsequent sections. The terms $\mathcal{J}_x$ and $\mathcal{J}_y$ are the x- and y-directed impulse the fluid exerts on the airfoil, respectively, $\mathcal{W}$ is the total work done on the airfoil by the fluid, and $\mathcal{W}_x$, $\mathcal{W}_y$, and $\mathcal{W}_\theta$ are the translational and rotational components of the total work. The fully discrete, high-order approximation of the integrated quantities of interest (DG in space, DIRK in time) will be denoted with the corresponding Roman symbol, e.g., $W(\boldsymbol{u}^{(0)},\ldots,\boldsymbol{u}^{(N_t)},\boldsymbol{k}_1^{(n)},\ldots,\boldsymbol{k}_s^{(n)},\boldsymbol{\mu})$ is the fully discrete approximation of $\mathcal{W}(\boldsymbol{U},\boldsymbol{\mu})$.

D.2.5 Numerical Experiment: Energetically Optimal Trajectory of 2D Airfoil in Compressible, Viscous Flow

In this section, the high-order, time-dependent PDE-constrained optimization framework introduced in this document is applied to find the energetically optimal trajectory of a 2D NACA0012 airfoil with chord length $l = 1$ and zero-thickness trailing edge. The governing equations are the 2D compressible, isentropic Navier-Stokes equations. The mission of the airfoil is to move a distance of -1.5 units

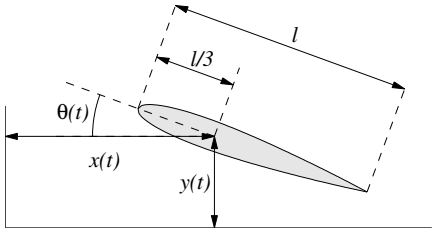


Figure D.2: Airfoil kinematics

horizontally and 1.5 units vertically in $T = 4$ units of time, with the restriction that $\theta(0) = \theta(T) = 0$, i.e., the angle of attack at the initial and final time is zero. Additionally, to ensure smoothness of the motion and avoid non-physical transients, $\dot{x}(0) = \dot{x}(T) = \dot{y}(0) = \dot{y}(T) = \dot{\theta}(0) = \dot{\theta}(T) = 0$ are enforced. The goal is to determine the trajectory $x(t)$, $y(t)$, $\theta(t)$ of the airfoil that minimizes the total energy required to complete the mission, i.e.,

$$\begin{aligned}
\underset{\boldsymbol{U},\,\boldsymbol{\mu}}{\text{minimize}} \quad & \mathcal{W}(\boldsymbol{U},\boldsymbol{\mu}) \\
\text{subject to} \quad & x(0) = \dot{x}(0) = \dot{x}(T) = 0,\ x(T) = -1.5 \\
& y(0) = \dot{y}(0) = \dot{y}(T) = 0,\ y(T) = 1.5 \\
& \theta(0) = \theta(T) = \dot{\theta}(0) = \dot{\theta}(T) = 0 \\
& \frac{\partial \boldsymbol{U}}{\partial t} + \nabla \cdot \boldsymbol{F}(\boldsymbol{U}, \nabla \boldsymbol{U}) = 0 \quad \text{in } v(\boldsymbol{\mu}, t).
\end{aligned} \tag{D.43}$$

The trajectory of the airfoil—$x(t)$, $y(t)$, and $\theta(t)$—is discretized via clamped cubic splines with $m_x + 1$, $m_y + 1$, and $m_\theta + 1$ knots, respectively. The knots are uniformly spaced between 0 and T in the t-dimension and the knot values are optimization parameters. Table D.2 summarizes two parametrizations considered in this section: (PI) the translational degrees of freedom—$x(t)$ and $y(t)$)—are frozen at their nominal value in Figure D.4 and the rotational degree of freedom—$\theta(t)$—is parametrized with a $m_\theta + 1$-knot clamped cubic spline and (PII) all rigid body modes are parametrized with clamped cubic splines. The 7 IDs in Table D.2 correspond to levels of refinement of the given parametrization with ID = 1 being the coarsest parametrization and ID = 7 the finest. With this parametrization of the airfoil kinematics, spatial and temporal discretization with the high-order scheme of Section D.1 leads to the fully discrete version of the optimization problem in

Table D.2: Summary of parametrizations considered in Section D.2.5. The number of clamped cubic spline knots used to discretize $x(t)$, $y(t)$, and $\theta(t)$ are $m_x + 1$, $m_y + 1$, and m_θ, respectively. PI freezes the rigid body translation ($m_x = m_y = 0$) and optimizes over only the rotation ($m_\theta \neq 0$). PII optimizes over all rigid body degrees of freedom ($m_x = m_y = m_\theta \neq 0$).

	PI				PII			
ID	m_x	m_y	m_θ	$N_{\boldsymbol{\mu}}$	m_x	m_y	m_θ	$N_{\boldsymbol{\mu}}$
1	0	0	2	3	2	2	2	9
2	0	0	6	7	6	6	6	21
3	0	0	10	11	10	10	10	33
4	0	0	15	16	15	15	15	48
5	0	0	25	26	25	25	25	78
6	0	0	50	51	50	50	50	153
7	0	0	100	101	100	100	100	303

(D.43)

$$\underset{\substack{\boldsymbol{u}^{(0)},\ \ldots,\ \boldsymbol{u}^{(N_t)} \in \mathbb{R}^{N_{\boldsymbol{u}}},\\ \boldsymbol{k}_1^{(1)},\ \ldots,\ \boldsymbol{k}_s^{(N_t)} \in \mathbb{R}^{N_{\boldsymbol{u}}},\\ \boldsymbol{\mu} \in \mathbb{R}^{N_{\boldsymbol{\mu}}}}}{\text{minimize}} \quad W(\boldsymbol{u}^{(0)},\ \ldots,\ \boldsymbol{u}^{(N_t)},\ \boldsymbol{k}_1^{(1)},\ \ldots,\ \boldsymbol{k}_s^{(N_t)},\ \boldsymbol{\mu})$$

$$\begin{aligned}
\text{subject to} \quad & x(0) = \dot{x}(0) = \dot{x}(T) = 0,\ x(T) = -1.5 \\
& y(0) = \dot{y}(0) = \dot{y}(T) = 0,\ y(T) = 1.5 \\
& \theta(0) = \theta(T) = \dot{\theta}(0) = \dot{\theta}(T) = 0 \\
& \boldsymbol{u}^{(0)} = \boldsymbol{u}_0 \\
& \boldsymbol{u}^{(n)} = \boldsymbol{u}^{(n-1)} + \sum_{i=1}^{s} b_i \boldsymbol{k}_i^{(n)} \\
& \boldsymbol{M}\boldsymbol{k}_i^{(n)} = \Delta t_n \boldsymbol{r}\left(\boldsymbol{u}_i^{(n)},\ \boldsymbol{\mu},\ t_{n-1} + c_i \Delta t_n\right).
\end{aligned} \tag{D.44}$$

Before considering the optimization problem (D.44), the proposed adjoint method for computing gradients of quantities of interest on the manifold of fully discrete, high-order solutions of the conservation law (D.14) is verified against a fourth-order finite difference approximation. The finite difference approximation to gradients on the aforementioned manifold requires finding the solution of the fully-discretized governing equations *at perturbations* about the nominal parameter configuration in Figure D.4. To mitigate round-offs errors as much as possible in the finite difference computation, the number of time steps was reduced to 10 and only half of a period was simulated. Figure D.3 shows the relative error between the gradients computed via the adjoint method and this finite difference approximation for a sweep of finite difference intervals, τ. A relative error on the order of 10^{-10} is observed for a finite difference step of $\tau = 10^{-4}$. As expected, the error starts to increase after τ drops too small due to the trade-off between finite difference accuracy and roundoff error.

With this verification of the adjoint-based gradients, attention is turned to the optimization problem in (D.44). The optimization solver used in this section is L-BFGS-B [215], a bound-constrained,

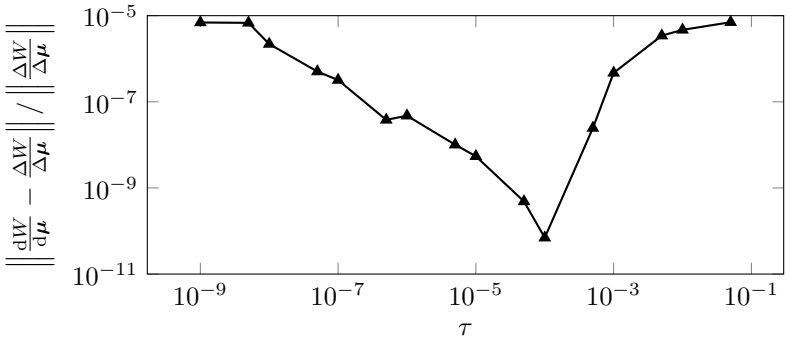


Figure D.3: Verification of adjoint-based gradient with fourth-order centered finite difference approximation, for a range of finite intervals, τ, for the total work W—the objective function in (D.44)—for parametrization PII (Table D.2). The computed gradient match the finite difference approximation to about 10 digits of accuracy before round-off errors degrade the accuracy.

limited-memory BFGS algorithm. Figure D.4 contains the initial guess for the optimization problem in (D.44) as well as its solution under both parametrization, PI and PII, at the finest level of refinement (ID = 7). The initial guess for the optimization problem is a pure translational motion with $\theta(t) = 0$. The solution under parametrization PI freezes the translational motion at its nominal value and incorporates rotational motion. The solution under parametrization PII increases the amplitude of the rotation, flattens the trajectory of $x(t)$, and incorporates an overshoot in $y(t)$ before settling to the required location, as compared to the optimal solution corresponding to PI.

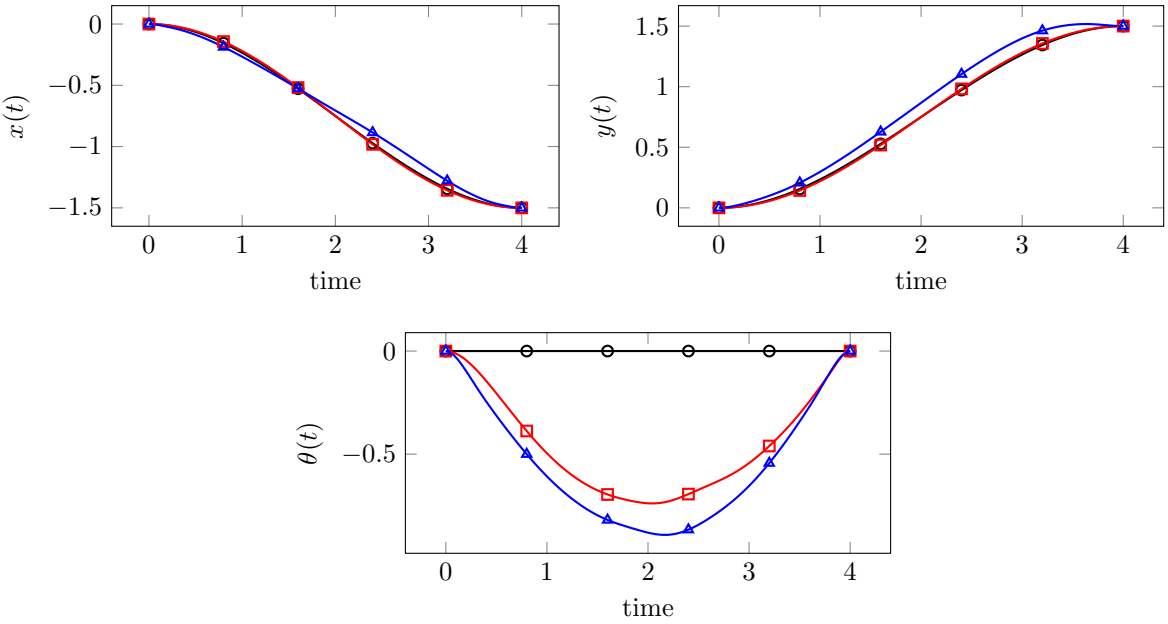


Figure D.4: Trajectories of $x(t)$, $y(t)$, and $\theta(t)$ at initial guess (—o—), solution of (D.44) under parametrization PI (—□—), and solution of (D.44) under parametrization PII (—△—) for ID = 7.

The instantaneous quantities of interest for the nominal trajectory and solution of (D.44) under parametrizations PI and PII are included in Figure D.5. It is clear that the optimal solution under both parametrizations result in a time history of the total power that is uniformly closer to 0 than that at the nominal trajectory, which is expected since W is the objective function. With the exception of the edges of the time interval, the total power time history for the optimal solution under parametrization PII is uniformly closer to 0 than that of PI. The same observation holds for the power due to the translational motion, $\mathcal{P}_x^h$ and $\mathcal{P}_y^h$. Whereas the total power corresponding to the nominal trajectory is due solely to the translational motion (since there is no rotation), the optimal solutions exchange large amounts of translational power for a small amount of rotational power. These observations can also be verified in Table D.3 which summarizes the optimal values of the integrated quantities of interest.

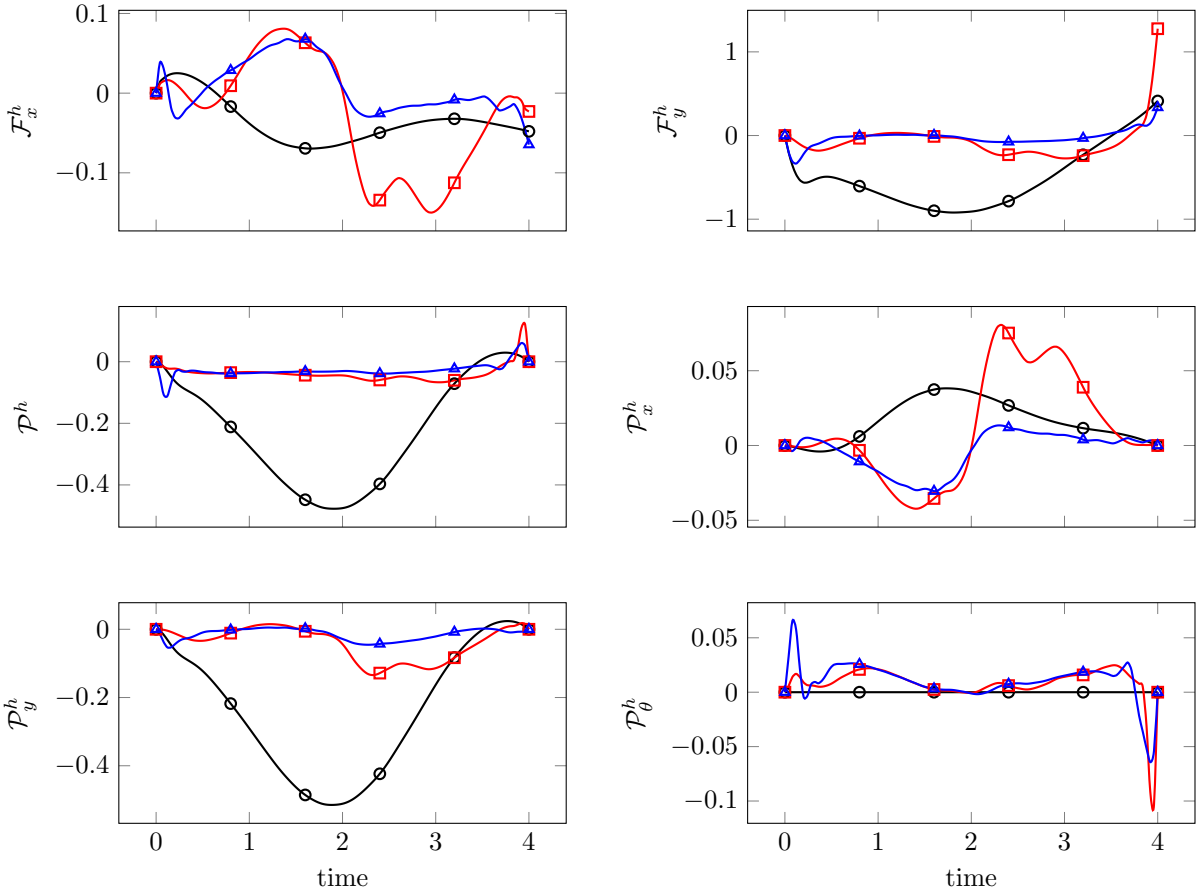


Figure D.5: Time history of instantaneous quantities of interest (x-directed force – $\mathcal{F}_x^h(\boldsymbol{u}, \boldsymbol{\mu}, t)$, y-directed force – $\mathcal{F}_y^h(\boldsymbol{u}, \boldsymbol{\mu}, t)$, total power – $\mathcal{P}^h(\boldsymbol{u}, \boldsymbol{\mu}, t)$, x-translational power – $\mathcal{P}_x^h(\boldsymbol{u}, \boldsymbol{\mu}, t)$, y-translational power – $\mathcal{P}_y^h(\boldsymbol{u}, \boldsymbol{\mu}, t)$, rotational power – $\mathcal{P}_\theta^h(\boldsymbol{u}, \boldsymbol{\mu}, t)$) at initial guess (—○—), solution of (D.44) under parametrization PI (—□—), and solution of (D.44) under parametrization PII (—△—) for ID = 7.

The convergence of the total work, i.e., the objective function of the optimization problem, with iterations of the optimization solver is summarized in Figure D.6 (left). Both parametrizations are included and iterations are agglomerated over all IDs. The first iteration corresponds to a steepest descent step, which causes an adverse jump in the objective value. The following iterations make rapid progress toward the optimal solution, which is slowed as convergence is approached. The solver requires additional iterations to converge the solution corresponding to parametrization PII, which is expected due to the larger parameter space.

Next, convergence of the total work as the parameter space is refined is considered in Figure D.6 (right) and Table D.3. This implies the optimal trajectory among all twice continuously differentiable functions is being approached. For both parametrizations, the optimal value of the total work agrees to 3 digits between IDs 6 and 7 (roughly a factor of 2 difference in dimension of parameter spaces) and 2 digits between IDs 3 and 7 (roughly a factor of 10 difference in dimension of parameter spaces).

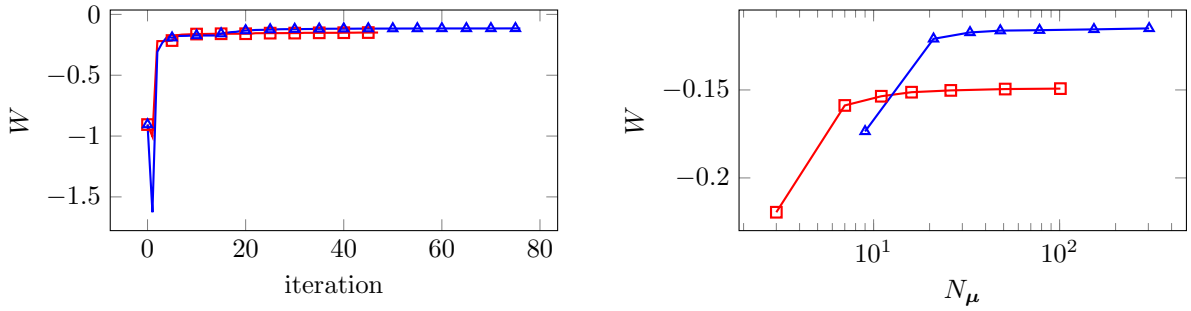


Figure D.6: *Left*: Convergence of total work W with optimization iteration for parametrization PI (—□—) and PII (—△—) for ID = 7. Both optimization problems converge to a motion with significantly lower required total work; PII finds a better motion than PI (in terms of total work) due to the enlarged search space, at the cost of additional iterations. Each optimization iteration requires a primal flow computation—to evaluate the quantities of interest—and its corresponding adjoint—to evaluate the gradient of the quantity of interest. *Right*: Convergence of optimal value of total work W as parameter space is refined for parametrization PI (—□—) and PII (—△—). This implies convergence to an optimal, smooth trajectory that is not polluted by its discrete parametrization.

The motion of the airfoil and vorticity of the surrounding flow are shown in Figure D.7 (nominal trajectory), Figure D.8 (optimal solution under parametrization PI), and Figure D.9 (optimal solution under parametrization PII). The flow corresponding to the nominal configuration experiences flow separation and vortex shedding, which results in the relatively large amount of total energy to complete the mission. Fixing the translational motion and optimizing over the rotation (PI) dramatically reduces the amount of shedding and consequently reduces the amount of work required. Optimizing the entire rigid body motion (PII) further reduces the shedding and required work.

Table D.3: Table summarizing integrated quantities of interest at optimal solution of (D.44) for each parametrization (PI, PII) for each level of refinement. The total work monotonically increases as $N_{\boldsymbol{\mu}}$ increases for a given parametrization, which is expected due to the nested search spaces. For a fixed ID, the optimal total work for parametrization PII is larger than that for PI since the search space for PI is a subset of that of PII. The other integrated quantities are included for completeness, but do not exhibit trends (except for converging to a fixed value as $N_{\boldsymbol{\mu}}$ increases) since they were not included in the optimization problem.

				PI			
ID	1	2	3	4	5	6	7
W	-2.1951e-01	-1.5881e-01	-1.5358e-01	-1.5128e-01	-1.5026e-01	-1.4950e-01	-1.4924e-01
W_x	8.1329e-02	5.6090e-02	4.9543e-02	4.5924e-02	4.5085e-02	4.4712e-02	4.4707e-02
W_y	-2.3460e-01	-1.8153e-01	-1.7122e-01	-1.6544e-01	-1.6374e-01	-1.6298e-01	-1.6294e-01
W_θ	6.6234e-02	3.3370e-02	3.1906e-02	3.1768e-02	3.1604e-02	3.1223e-02	3.1010e-02
F_x	-1.9234e-01	-1.3123e-01	-1.1886e-01	-1.1136e-01	-1.0912e-01	-1.0810e-01	-1.0800e-01
F_y	-5.1539e-01	-3.1711e-01	-3.1816e-01	-3.1877e-01	-3.2551e-01	-3.2959e-01	-3.3063e-01
				PII			
ID	1	2	3	4	5	6	7
W	-1.7357e-01	-1.2095e-01	-1.1733e-01	-1.1629e-01	-1.1603e-01	-1.1557e-01	-1.1502e-01
W_x	9.6487e-03	-1.4123e-02	-1.4328e-02	-1.4967e-02	-1.5021e-02	-1.5061e-02	-1.5027e-02
W_y	-1.1041e-01	-6.2238e-02	-6.1036e-02	-6.0425e-02	-6.0032e-02	-5.9489e-02	-5.9245e-02
W_θ	7.2807e-02	4.4585e-02	4.1963e-02	4.0895e-02	4.0980e-02	4.1023e-02	4.0749e-02
F_x	-4.1265e-02	2.8091e-02	2.7677e-02	2.9596e-02	2.9848e-02	3.0231e-02	3.0221e-02
F_y	-3.2231e-01	-1.064e-01	-1.0806e-01	-1.0890e-01	-1.1343e-01	-1.1626e-01	-1.1764e-01

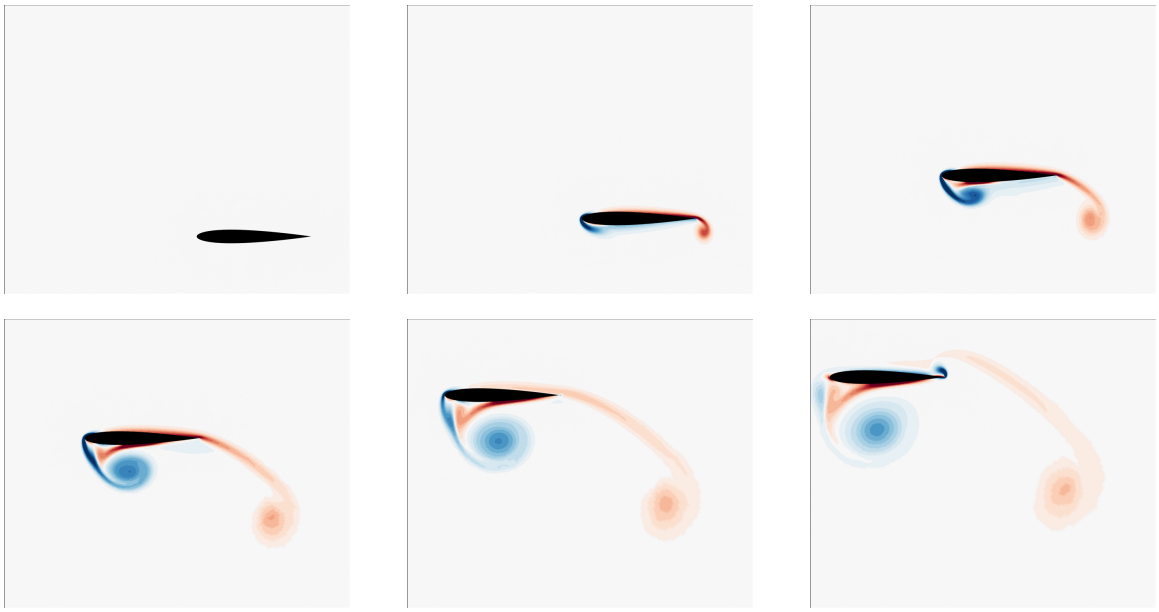


Figure D.7: Flow vorticity around airfoil undergoing motion corresponding to initial guess for optimization, i.e., pure heaving (—o—). Flow separation off leading edge implies a large amount of work required to complete mission. Snapshots taken at times $t = 0.0,\ 0.8,\ 1.6,\ 2.4,\ 3.2,\ 4.0$.

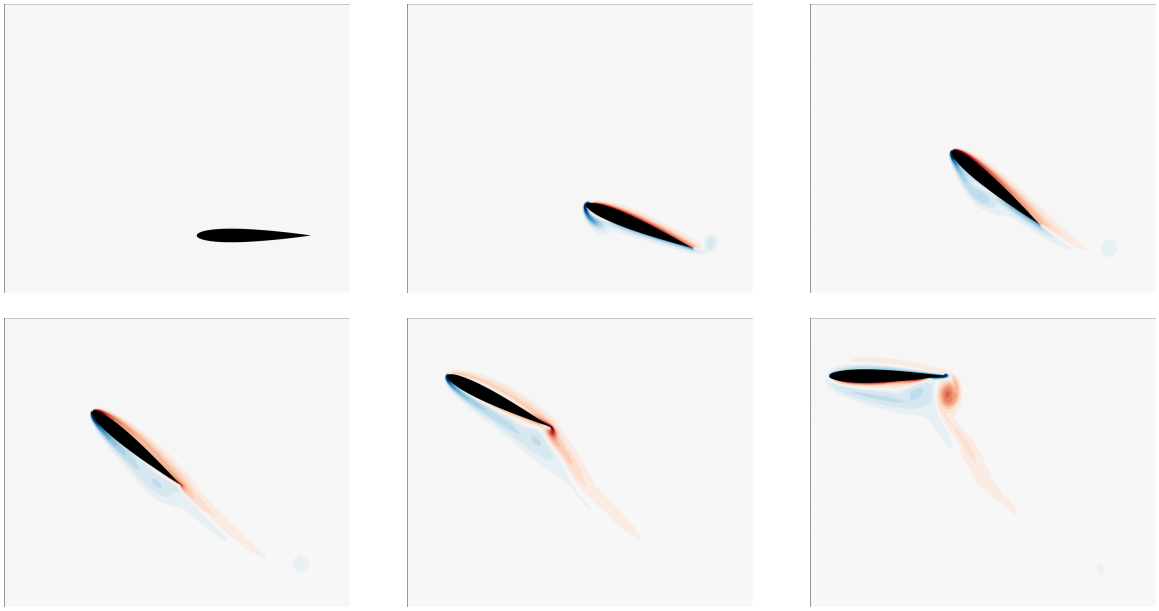


Figure D.8: Flow vorticity around airfoil undergoing motion corresponding to optimal pitching motion for fixed translational motion, i.e., solution of (D.44) under parametrization PI (—□—). The pitching motion greatly reduces the degree of flow separation and vortex shedding compared to the initial guess, and requires less work to complete the mission. Snapshots taken at times $t = 0.0,\ 0.8,\ 1.6,\ 2.4,\ 3.2,\ 4.0$.

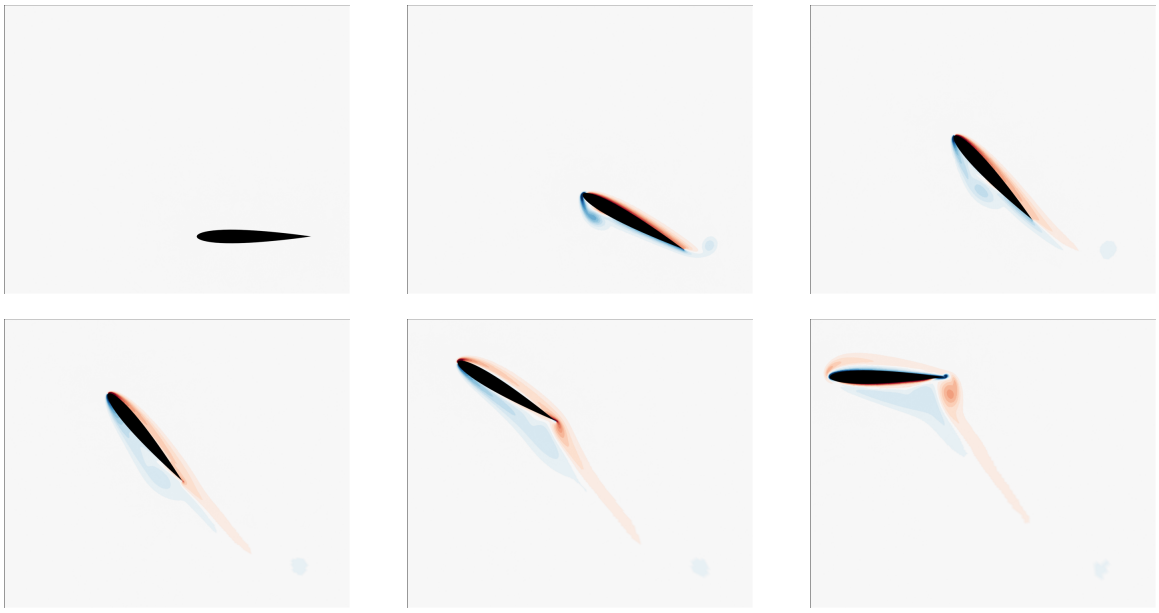


Figure D.9: Flow vorticity around airfoil undergoing motion corresponding to optimal rigid body motion, i.e., solution of (D.44) under parametrization PII (—▲—). This rigid body motion further reduces the degree of flow separation and required work to complete the mission. This motion differs from the solution of PI as it has a larger pitch amplitude and slightly overshoots the final vertical position before settling to the required position. Snapshots taken at times $t = 0.0,\ 0.8,\ 1.6,\ 2.4,\ 3.2,\ 4.0$.

D.2.6 Numerical Experiment: Energetically Optimal Shape and Flapping Motion of 2D Airfoil at Constant Impulse

In this section, the high-order, time-dependent PDE-constrained optimization framework introduced in this document is applied to find the energetically optimal flapping motion, under an impulse constraint, of a 2D NACA0012 airfoil (Figure D.10) with chord length $l = 1$ and zero-thickness trailing edge. The governing equations are the 2D compressible, isentropic Navier-Stokes equations.

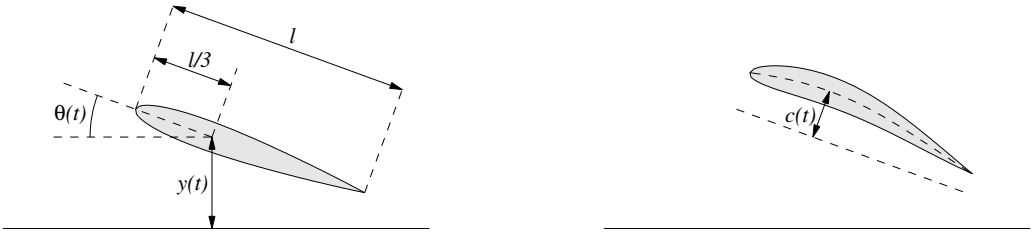


Figure D.10: Airfoil kinematics and deformation

The goal is to determine the flapping motion—$y(t)$ and $\theta(t)$—and shape—$c(t)$—of the airfoil

that minimizes the total energy such than a x-impulse of q is achieved, i.e.,

$$\begin{aligned} \underset{\boldsymbol{U},\ \boldsymbol{\mu}}{\text{minimize}} \quad & \mathcal{W}(\boldsymbol{U}, \boldsymbol{\mu}) \\ \text{subject to} \quad & \mathcal{J}_x(\boldsymbol{U}, \boldsymbol{\mu}) = q \\ & \frac{\partial \boldsymbol{U}}{\partial t} + \nabla \cdot \boldsymbol{F}(\boldsymbol{U},\ \nabla \boldsymbol{U}) = 0 \quad \text{ in } v(\boldsymbol{\mu}, t). \end{aligned} \tag{D.45}$$

The flapping frequency is fixed at 0.2, which corresponds to a period of $T = 5$. Proper initialization of the flow is the initial condition that results in a *time-periodic* flow [212] to completely avoid non-physical transients and simulate representative, *in-flight* conditions; this experiment uses a crude approximation that initializes the flow from the steady-state condition, simulates 3 periods of the flapping motion, and integrates the quantities of interest over the last period only. The deformation of the domain is determined from the value of $c(t)$ using the spatial blending map of Section D.2.4 with

$$\boldsymbol{\varphi}(\boldsymbol{X}, \boldsymbol{\mu}, t) = \begin{bmatrix} 0 \\ 2c(t)e^{-[(\boldsymbol{X} - \boldsymbol{x}_0)\cdot \boldsymbol{e}_1]^2} \end{bmatrix} \tag{D.46}$$

The trajectory of the airfoil—$y(t)$, and $\theta(t)$—and its shape – $c(t)$—are discretized via cubic splines with $m_y + 1$, $m_\theta + 1$, and $m_c + 1$ knots, respectively, with boundary conditions that enforce

$$y(t) = -y(t + T/2) \qquad \theta(t) = -\theta(t + T/2) \qquad c(t) = -c(t + T/2). \tag{D.47}$$

These boundary conditions[1] for $y(t)$, $\theta(t)$, and $c(t)$ correspond to a mirroring of the trajectory at $t = T/2$ and implicitly enforces periodicity with period T. The knots are uniformly spaced between 0 and T in the t-dimension and the knot values are optimization parameters. Since the unsteady simulation is initialized from the steady-state flow, non-zero velocities of the airfoil at $t = 0$ will result in non-physical transients. These transients are avoided by blending the periodic cubic spline smoothly to the zero function at the beginning of the time interval [193]. Let $s_y(t; \boldsymbol{\mu})$, $s_\theta(t; \boldsymbol{\mu})$, and $s_c(t; \boldsymbol{\mu})$ denote the periodic cubic spline approximations. Then, the flapping and shape trajectories are defined as

$$y(t) = b(t)s_y(t; \boldsymbol{\mu}) \qquad \theta(t) = b(t)s_\theta(t; \boldsymbol{\mu}) \qquad c(t) = b(t)s_c(t; \boldsymbol{\mu}), \tag{D.48}$$

where $b(t) = 1.0 - e^{-t^2}$. Table D.4 summarizes two parametrizations considered in this section: (FI) rigid body motion parametrized via cubic splines and shape fixed at nominal value and (FII) rigid body motion and shape of airfoil parametrized via cubic splines. With this parametrization of the airfoil kinematics and shape, spatial and temporal discretization with the high-order scheme of

[1] Periodic and mirrored cubic splines of this form with $m + 1$ knots only have m degrees of freedom since the boundary condition prescribes the value of the $m + 1$ knot from the values of the others m.

Table D.4: Summary of parametrizations considered in Section D.2.6. The number of periodic cubic spline knots used to discretize $y(t)$, $\theta(t)$, and (t) are $m_y + 1$, $m_\theta + 1$, and $m_c + 1$, respectively. FI freezes the airfoil shape and considers only rigid body motions ($m_y = m_\theta \neq 0, m_c = 0$). FII parametrizes both shape and kinematic motion ($m_y = m_\theta = m_c \neq 0$).

FI				FII			
m_y	m_θ	m_c	$N_{\boldsymbol{\mu}}$	m_y	m_θ	m_c	$N_{\boldsymbol{\mu}}$
4	4	0	6	4	4	4	9

Section D.1 leads to the fully discrete version of the optimization problem in (D.45)

$$
\begin{aligned}
\underset{\substack{\boldsymbol{u}^{(0)},\ \ldots,\ \boldsymbol{u}^{(N_t)} \in \mathbb{R}^{N_{\boldsymbol{u}}},\\ \boldsymbol{k}_1^{(1)},\ \ldots,\ \boldsymbol{k}_s^{(N_t)} \in \mathbb{R}^{N_{\boldsymbol{u}}},\\ \boldsymbol{\mu} \in \mathbb{R}^{N_{\boldsymbol{\mu}}}}}{\text{minimize}} \quad & W(\boldsymbol{u}^{(0)},\ \ldots,\ \boldsymbol{u}^{(N_t)},\ \boldsymbol{k}_1^{(1)},\ \ldots,\ \boldsymbol{k}_s^{(N_t)},\ \boldsymbol{\mu}) \\
\text{subject to} \quad & J_x(\boldsymbol{u}^{(0)},\ \ldots,\ \boldsymbol{u}^{(N_t)},\ \boldsymbol{k}_1^{(1)},\ \ldots,\ \boldsymbol{k}_s^{(N_t)},\ \boldsymbol{\mu}) = 0 \\
& \boldsymbol{u}^{(0)} = \boldsymbol{u}_0 \\
& \boldsymbol{u}^{(n)} = \boldsymbol{u}^{(n-1)} + \sum_{i=1}^{s} b_i \boldsymbol{k}_i^{(n)} \\
& \boldsymbol{M}\boldsymbol{k}_i^{(n)} = \Delta t_n \boldsymbol{r}\left(\boldsymbol{u}_i^{(n)},\ \boldsymbol{\mu},\ t_{n-1} + c_i \Delta t_n\right).
\end{aligned}
\tag{D.49}
$$

Given the gradient verification from the previous section, attention is turned directly to the optimization problem in (D.49) for various values of the impulse constraint, q. The optimization solver used in this section is SNOPT [70], a nonlinearly constrained SQP method. Figure D.11 contains the initial guess for the optimization problem in (D.44) as well as its solution under both parametrization, FI and FII. The initial guess for the optimization problem is a pure heaving motion at a fixed shape, i.e., $c(t) = \theta(t) = 0$. The solution under parametrization PI freezes the shape at its nominal configuration (NACA0012) and modifies the rigid body motion. Pitch is introduced for all values of the impulse constraint and the amplitude of the heaving motion is decreased for $q = 0.0, 1.0$ and increased for $q = 2.5$. The solution under parametrization PII reduces the heaving amplitude and slightly increases the pitch amplitude as compared to PI. It also introduces non-trivial camber.

The instantaneous quantities of interest—W and J_x in this case—for the nominal motion and shape and solution of (D.49) under parametrizations PI and PII are included in Figure D.12. It is clear that the optimal solution under both parametrizations result in a time history of the total power that is uniformly closer to 0 than that at the nominal trajectory, which is expected since W is the objective function. It is also clear that larger values of the impulse constraint require more power to complete the flapping motion. While it may not be clear from Figure D.12, the integration of $\mathcal{F}_x^h$ leads to an impulse that exactly conforms to the specified value of q. This can be seen more clearly in Figure D.13. These observations can also be verified in Figure D.13 and Table D.5 that summarizes the optimal values of the integrated quantities of interest.

Figure D.13 shows the convergence of the integrated quantities of interest with iterations in

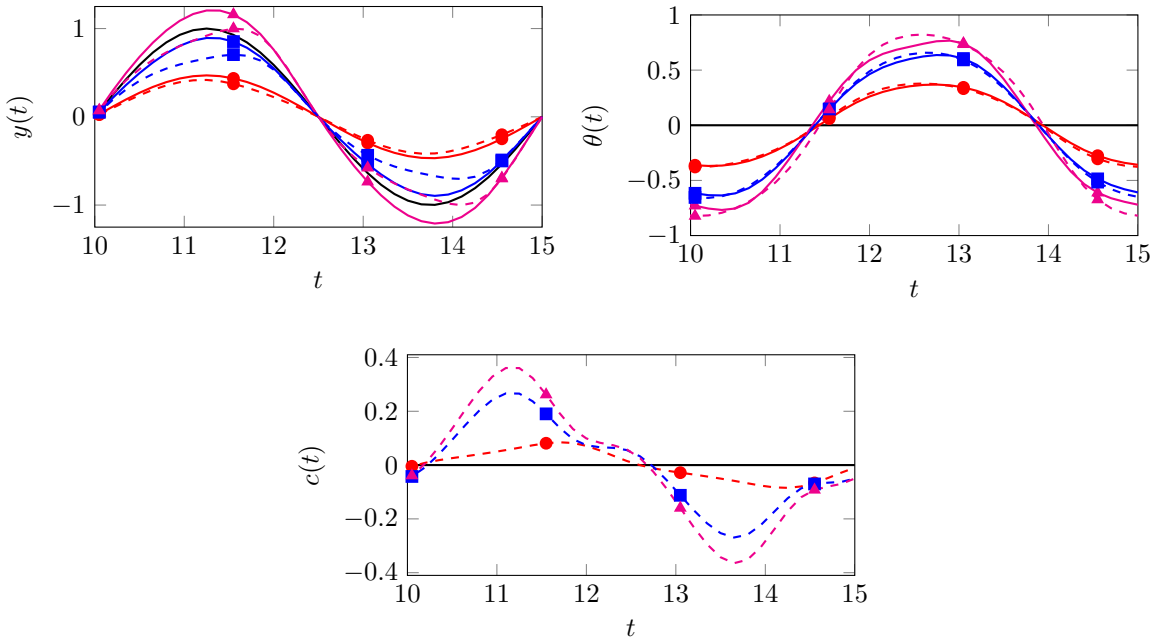


Figure D.11: Trajectories of $y(t)$, $\theta(t)$, and $c(t)$ at initial guess (——), solution of (D.49) under parametrization FI ($q = 0.0$: —●—, $q = 1.0$: —■—, $q = 2.5$: —▲—), and solution of (D.49) under parametrization FII ($q = 0.0$: - -●- -, $q = 1.0$: - -■- -, $q = 2.5$: - -▲- -) from Table D.4.

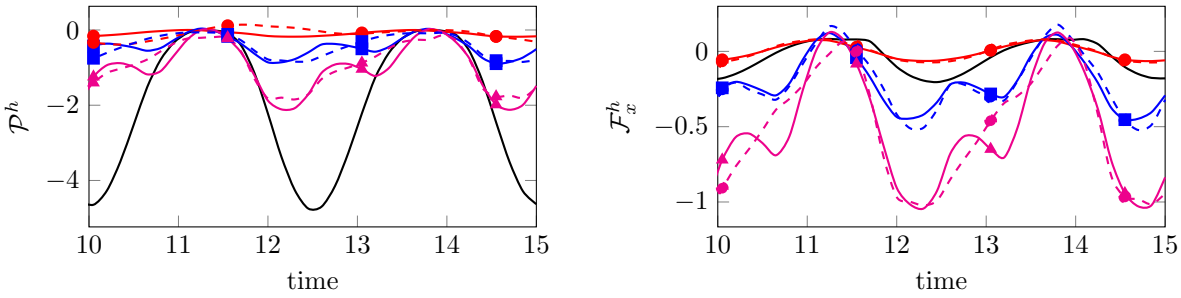


Figure D.12: Time history of total power, $\mathcal{P}^h(\boldsymbol{u}, \boldsymbol{\mu}, t)$, and x-directed force, $\mathcal{F}_x^h(\boldsymbol{u}, \boldsymbol{\mu}, t)$, imparted onto foil by fluid at initial guess (——), solution of (D.49) under parametrization FI ($q = 0.0$: —●—, $q = 1.0$: —■—, $q = 2.5$: —▲—), and solution of (D.49) under parametrization FII ($q = 0.0$: - -●- -, $q = 1.0$: - -■- -, $q = 2.5$: - -▲- -) from Table D.4.

the optimization solver. The aforementioned observations can be verified by inspection of the final iteration: all impulse constraints are satisfied, larger values of q require more work to achieve, and morphing the shape of the airfoil allows for a slight reduction in the required work. After 20 iterations, the impulse constraint is satisfied for $q = 0.0, 1.0$ and reduction of the work has essentially ceased, implying the optimization could have been terminated at that point. The case with $q = 2.5$ requires an additional 15 - 20 iterations to settle to a converged solution.

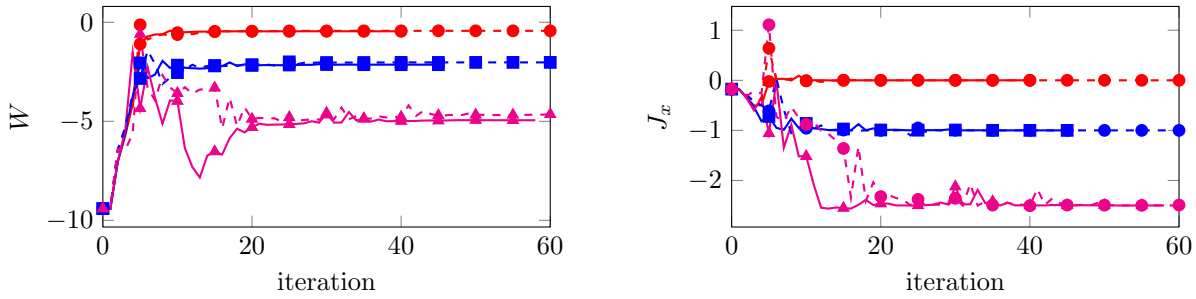


Figure D.13: Convergence of quantities of interest, W and J_x, with optimization iteration for parametrization FI ($q = 0.0$: —●—, $q = 1.0$: —■—, $q = 2.5$: —▲—) and FII ($q = 0.0$: - -●- - , $q = 1.0$: - -■- - , $q = 2.5$: - -▲- -) from Table D.4. Each optimization iteration requires the a primal flow computation—to evaluate quantities of interest—and its corresponding adjoint—to evaluate the gradient of quantities of interest.

The shape and motion of the airfoil and vorticity of the surrounding flow are shown in Figure D.14 (nominal), Figure D.15 (optimal solution under parametrization FI for $q = 2.5$), and Figure D.16 (optimal solution under parametrization FII for $q = 2.5$). The flow corresponding to the nominal configuration experiences flow separation and vortex shedding, which results in the relatively large amount of total energy to complete the flapping motion and *does not satisfy the impulse constraint*. Fixing the shape and optimizing over the heaving and pitching motion (FI) dramatically reduces the amount of shedding and consequently reduces the amount of work required. Optimizing the shape in addition to the pitching and heaving motion (FII) further reduces the shedding and required work. The solution of FI and FII both satisfy the impulse constraint to greater than 8 digits of accuracy.

To conclude this section, a brief comparison of the optimal flapping motions found in this work are compared to those found in the literature. From Figure D.11, the pitch of the foil leads its plunge by approximately 90° in all optimal flapping motions, a result that was found in several works that range from experimental and computational [191, 162, 158, 148]. The improved efficiency is largely due to a dramatic reduction in leading edge vortex shedding characteristic of pure heaving motions (Figure D.14) [191, 158]. The specific pitching and heaving amplitudes were determined by the optimizer such that the thrust constraint is satisfied; as the thrust requirement is increased, the magnitude of the pitch and plunge increase and leading edge shedding off the leading edge is induced (Figure D.15) [148]. The time-dependent shape deformation slightly reduces the magnitude of the vortices shedding off the leading edge, which can be seen by comparing Figures D.15 and D.16.

Table D.5: Table summarizing integrated quantities of interest at optimal solution of each optimization problem for each impulse level. In all cases, the desired value of J_x is achieved to greater than 4 digits of accuracy. The optimal solution for larger values of the impulse constraint require more total work to complete flapping motion, i.e., work monotonically increases in magnitude as value of impulse constraint increases. Smaller values of total work are achievable if airfoil is allowed to morph its shape in addition its rigid body motion. The other integrated quantities are included for completeness, but do not exhibit trends since they were not in the optimization problem.

	Initial	FI			FII		
q	0.0	1.0	2.5	0.0	1.0	2.5
W	-9.4096e+00	-4.5695e-01	-2.1419e+00	-4.9476e+00	-4.3252e-01	-2.0271e+00	-4.6110e+00
W_x	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00
W_y	-9.4096e+00	-4.2807e-01	-2.0642e+00	-4.7967e+00	-3.7363e-01	-1.5413e+00	-3.3712e+00
W_θ	0.0000e+00	2.8883e-02	7.7694e-02	1.5083e-01	1.7101e-02	6.7900e-03	1.8744e-01
F_x	-1.7660e-01	-4.0490e-11	-1.0000e+00	-2.5000e+00	1.6937e-10	-1.0000e+00	-2.5000e+00
F_y	3.5413e-01	1.5989e-02	5.0480e-02	9.7240e-02	-1.5292e-02	4.8657e-02	9.6440e-02

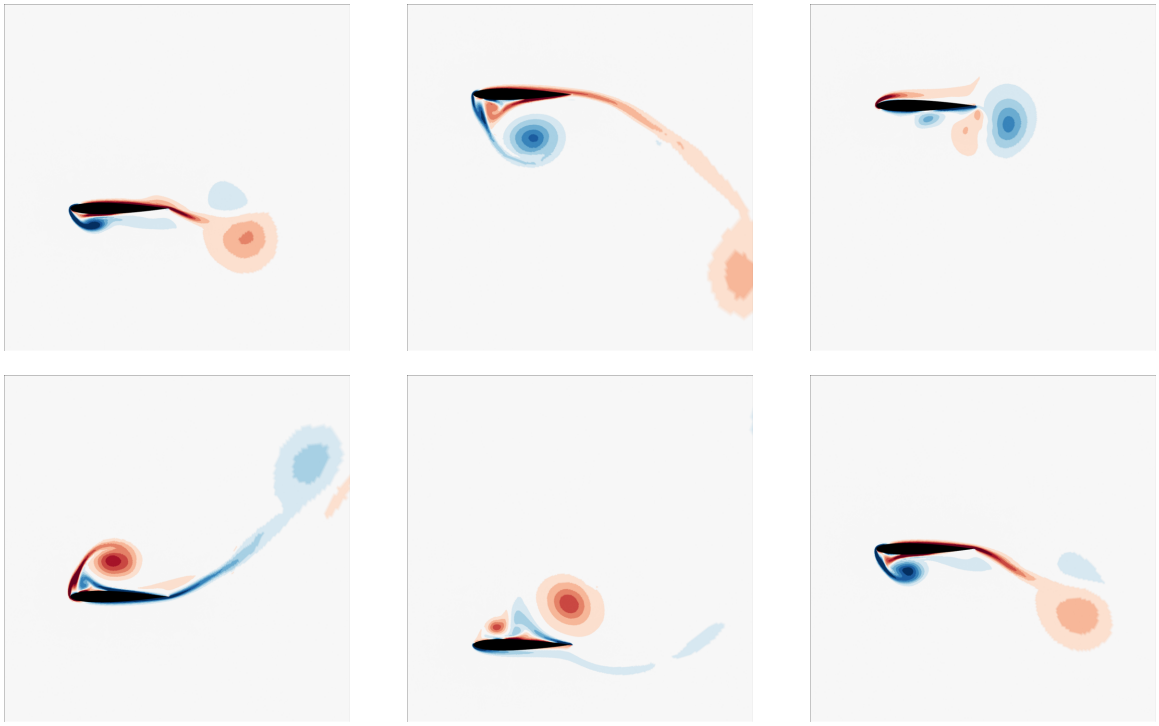


Figure D.14: Flow vorticity around flapping airfoil undergoing motion corresponding to initial guess for optimization problem (D.49), i.e., pure heaving (——). Flow separation off leading edge implies a large amount of work required for flapping motion. Snapshots taken at times $t = 9.75,\ 10.8,\ 11.85,\ 12.9,\ 13.95,\ 15.0$.

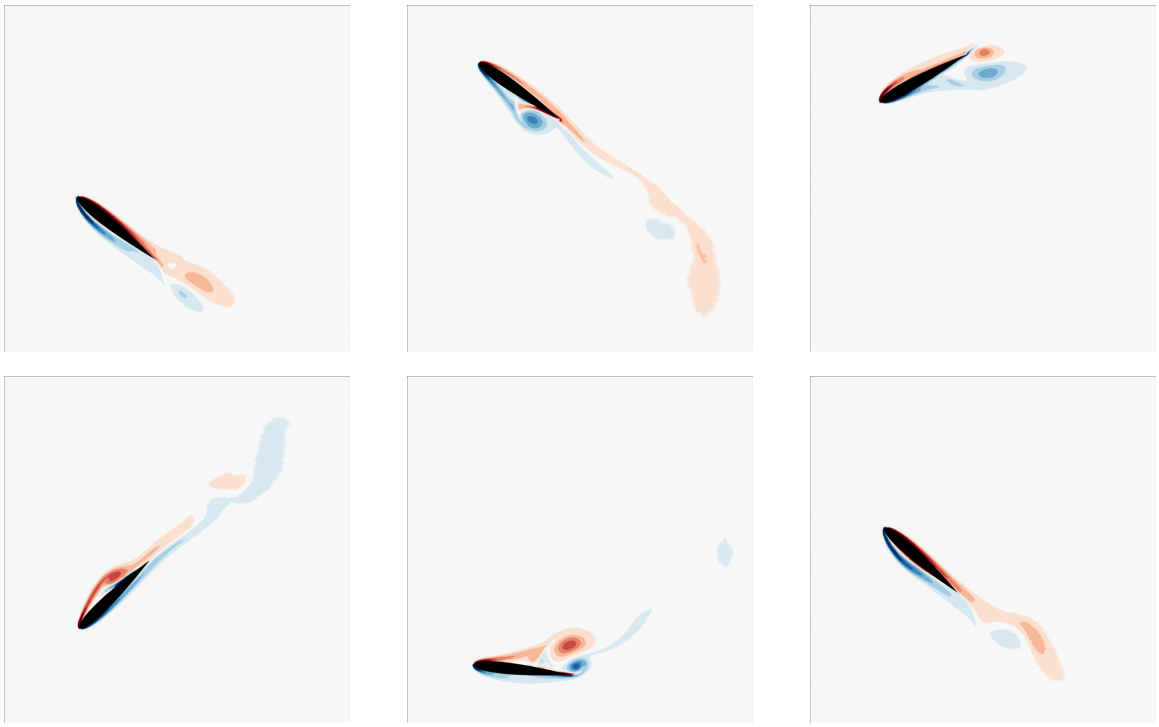


Figure D.15: Flow vorticity around flapping airfoil undergoing optimal rigid body motion corresponding to the solution of (D.49) under parametrization FI. The x-directed impulse is $J_x = 2.5$. The pitching motion greatly reduces the degree of flow separation and vortex shedding compared to the initial guess, and requires less work to complete the flapping motion and generate desired impulse. Snapshots taken at times $t = 9.75,\ 10.8,\ 11.85,\ 12.9,\ 13.95,\ 15.0$.

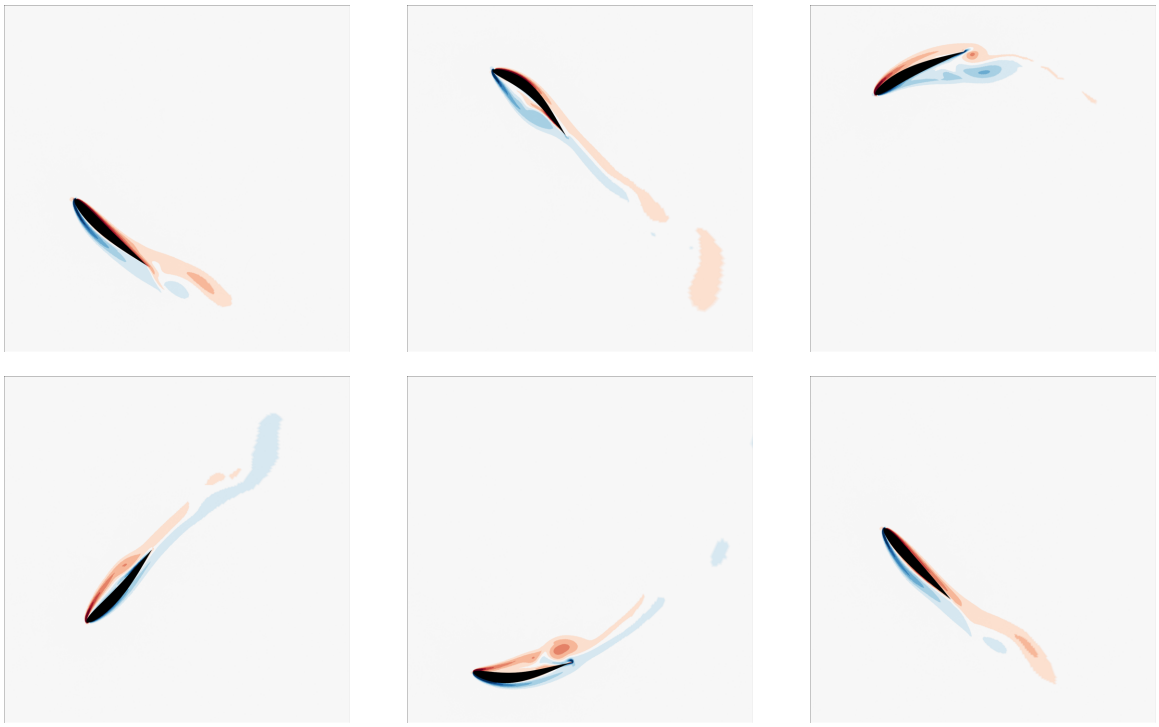


Figure D.16: Flow vorticity around flapping airfoil undergoing optimal deformation and kinematic motion, corresponding to the solution of (D.49) under parametrization FII. The x-directed impulse is $J_x = 2.5$. The morphing further reduces the flow separation and work required to complete the flapping motion and generate desired impulse. Snapshots taken at times $t = 9.75,\ 10.8,\ 11.85,\ 12.9,\ 13.95,\ 15.0$.

D.3 Computing Time-Periodic Solutions of Partial Differential Equations

This section is devoted to the solution of partial differential equations with time-periodicity constraints. This will largely be a review of existing work on the topic [131, 196, 7, 8, 201, 77], although emphasis will be placed on equations that are *parametrized* and *fully discretized*. This will lead to the main contribution of this work, the fully discrete adjoint equations corresponding to time-periodic solutions of partial differential equations and their use in computing gradients of quantities of interest along the manifold of time-periodic solutions.

Consider the general, nonlinear, time-periodically constrained system of partial differential equations, parametrized by the vector $\boldsymbol{\mu} \in \mathbb{R}^{N_{\boldsymbol{\mu}}}$,

$$\begin{aligned} \frac{\partial \boldsymbol{U}}{\partial t} &= \boldsymbol{\mathcal{L}}(\boldsymbol{U}, \boldsymbol{\mu}, t) \qquad \text{in} \quad \Omega(\boldsymbol{\mu}, t) \times (0, T] \\ \boldsymbol{U}(\boldsymbol{x}, 0) &= \boldsymbol{U}(\boldsymbol{x}, T), \end{aligned} \tag{D.50}$$

where $\boldsymbol{\mathcal{L}}(\cdot, \boldsymbol{\mu}, t)$ is a spatial differential operator on the parametrized, time-dependent domain $\Omega(\boldsymbol{\mu}, t) \subset \mathbb{R}^{n_{sd}}$. The boundary conditions have not been explicitly stated for brevity. This work will only consider temporally first-order partial differential equations, or those that have been recast as such. Without loss of generality, consider a quantity of interest of the form

$$\mathcal{F}(\boldsymbol{U}, \boldsymbol{\mu}) = \int_0^T \int_{\Gamma(\boldsymbol{\mu}, t)} f(\boldsymbol{U}, \boldsymbol{\mu}, t) \, dS \, dt, \tag{D.51}$$

where $\Gamma(\boldsymbol{\mu}, t) \subseteq \partial\Omega(\boldsymbol{\mu}, t)$. The generalization to other types of quantities of interest, such as volumetric integrals and instantaneous or pointwise quantities of interest, is immediate as the specific form of the quantity of interest will be abstracted away at the fully discrete level. The form in (D.51) will be used in the physical setup of the applications in subsequent sections. In subsequent sections, this quantity of interest will correspond to either the objective function or a constraint of an optimization problem governed by a partial differential equation and subject to a time-periodicity requirement. After space-time discretization of (D.50) via the DG-ALE-DIRK scheme discussed in Section D.1, the fully discrete equations are

$$\begin{aligned} \boldsymbol{u}^{(n)} &= \boldsymbol{u}^{(n-1)} + \sum_{i=1}^{s} b_i \boldsymbol{k}_i^{(n)} \\ \boldsymbol{M}\boldsymbol{k}_i^{(n)} &= \Delta t_n \boldsymbol{r}\left(\boldsymbol{u}_i^{(n)}, \ \boldsymbol{\mu}, \ t_{n-1} + c_i \Delta t_n\right), \end{aligned} \tag{D.52}$$

where $\boldsymbol{u}_i^{(n)}$ is defined in (D.15) and the fully discrete quantity of interest is

$$F(\boldsymbol{u}^{(0)}, \ldots, \boldsymbol{u}^{(N_t)}, \boldsymbol{k}_1^{(1)}, \ldots, \boldsymbol{k}_s^{(N_t)}). \tag{D.53}$$

Time-periodicity may then be expressed as the constraint

$$\boldsymbol{u}^{(0)} = \boldsymbol{u}^{(N_t)}, \tag{D.54}$$

where N_t is the time index of the cycle period.

The next section discusses methods for solving the fully discrete, time-periodically constrained partial differential equations. The periodicity constraint, i.e., $\boldsymbol{u}^{(0)} = \boldsymbol{u}^{(N_t)}$, turns the problem into a nonlinear two-point boundary value problem, which eliminates the possibility of using traditional evolution methods (since the initial conditions is unknown).

D.3.1 Numerical Solvers: Shooting Methods

This section provides a brief, non-exhaustive review of methods which have been introduced for solving time-periodic partial differential equations. A distinguishing feature of this work is that we directly consider the fully discrete form of the governing equations, whereas previous work has focused on the continuous [194] or semi-discrete [185] levels. The section will conclude with a discussion of a Newton-Krylov shooting method using a purely matrix-free Krylov solver to solve the linear systems of equations that arise, which extends the work in [77].

Define $\boldsymbol{u}^{(N_t)}(\boldsymbol{u}_0; \boldsymbol{\mu})$ as the solution of the following initial-value problem

$$\begin{aligned} \boldsymbol{u}^{(0)} &= \boldsymbol{u}_0 \\ \boldsymbol{u}^{(n)} &= \boldsymbol{u}^{(n-1)} + \sum_{i=1}^{s} b_i \boldsymbol{k}_i^{(n)} \\ \boldsymbol{M}\boldsymbol{k}_i^{(n)} &= \Delta t_n \boldsymbol{r}\left(\boldsymbol{u}_i^{(n)},\ \boldsymbol{\mu},\ t_{n-1} + c_i \Delta t_n\right), \end{aligned} \tag{D.55}$$

which can be solved using a traditional evolution algorithm that advances the solution from timestep n to $n+1$. Notice that this overloads the notation introduced in Section 2.1.3, which defines $\boldsymbol{u}^{(N_t)}$ as the discrete approximation of the time-periodic solution of the system of partial differential equations at the final time. Here, it is a nonlinear function that maps a state $\boldsymbol{u}_0 \in \mathbb{R}^{N_{\boldsymbol{u}}}$ to the state $\boldsymbol{u}^{(N_t)}(\boldsymbol{u}_0; \boldsymbol{\mu})$. It is clear that $\boldsymbol{u}_0$ is the time-periodic initial condition of the fully discrete partial differential equation if it is a fixed point of $\boldsymbol{u}^{(N_t)}(\cdot; \boldsymbol{\mu})$, namely

$$\boldsymbol{u}^{(N_t)}(\boldsymbol{u}_0; \boldsymbol{\mu}) = \boldsymbol{u}_0. \tag{D.56}$$

Then, provided the mapping $\boldsymbol{u}_0 \to \boldsymbol{u}^{(N_t)}(\boldsymbol{u}_0; \boldsymbol{\mu})$ is a contraction mapping, the Banach Fixed Point Theorem implies the existence of the fixed point and provides a convergent algorithm for finding it, see Algorithm 21. This is a convenient algorithm as it only relies on solution of the nonlinear evolution equation (D.55), but is known to suffer from poor convergence rates and lack of convergence if the mapping under consideration is not a contraction.

Another class of solvers for time-periodically constrained partial differential equations rely on

Algorithm 21 Fixed Point Iteration Time-Periodic Solutions of PDE

Input: Initial guess for periodic initial condition, $\boldsymbol{u}_0$; parameter configuration, $\boldsymbol{\mu}$
Output: Periodic initial condition, $\boldsymbol{u}^{(0)}$

1: **while** $\left\|\boldsymbol{u}^{(N_t)}(\boldsymbol{u}_0;\boldsymbol{\mu}) - \boldsymbol{u}_0\right\|_2 > \epsilon$ **do**
2: Update

$$\boldsymbol{u}_0 \leftarrow \boldsymbol{u}^{(N_t)}(\boldsymbol{u}_0;\boldsymbol{\mu})$$

3: **end while**
4: Define periodic initial condition

$$\boldsymbol{u}^{(0)} = \boldsymbol{u}_0$$

unconstrained, gradient-based optimization techniques. Define the function

$$j(\boldsymbol{u}_0) = \frac{1}{2}\left\|\boldsymbol{u}^{(N_t)}(\boldsymbol{u}_0;\boldsymbol{\mu}) - \boldsymbol{u}_0\right\|_2^2 \tag{D.57}$$

and consider the unconstrained optimization problem

$$\underset{\boldsymbol{u}_0 \in \mathbb{R}^{N_{\boldsymbol{u}}}}{\text{minimize}} \;\; j(\boldsymbol{u}_0), \tag{D.58}$$

which can be solved using gradient-based optimization techniques such as steepest descent, the Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm, or the limited-memory version of BFGS (L-BFGS) [71, 215, 143]. The gradient of (D.57), $\dfrac{\mathrm{d}j}{\mathrm{d}\boldsymbol{u}_0}$, is usually computed using the adjoint method since the large number of optimization variables, $N_{\boldsymbol{u}}$, renders the finite differences method or the linearized forward method impractical [78]. Throughout this work, the notation $\dfrac{\mathrm{d}(\cdot)}{\mathrm{d}\boldsymbol{\mu}}$ will be used to denote the total derivative of a quantity of interest with respect to parameters—including the explicit dependence as well as the implicit dependence through the solution of the governing equation—and the partial derivative notation $\dfrac{\partial(\cdot)}{\partial\boldsymbol{\mu}}$ will be used elsewhere. The adjoint equations for the fully discrete evolution equations in (D.55) corresponding to the quantity of interest, $j(\boldsymbol{u}_0)$, with parameter $\boldsymbol{u}_0$ are

$$\begin{aligned}
\boldsymbol{\lambda}^{(N_t)} &= \boldsymbol{u}^{(N_t)}(\boldsymbol{u}_0;\boldsymbol{\mu}) - \boldsymbol{u}_0 \\
\boldsymbol{\lambda}^{(n-1)} &= \boldsymbol{\lambda}^{(n)} + \sum_{i=1}^{s} \Delta t_n \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}\left(\boldsymbol{u}_i^{(n)},\ \boldsymbol{\mu},\ t_{n-1} + c_i \Delta t_n\right)^T \boldsymbol{\kappa}_i^{(n)} \\
\boldsymbol{M}^T \boldsymbol{\kappa}_i^{(n)} &= b_i \boldsymbol{\lambda}^{(n)} + \sum_{j=i}^{s} a_{ji} \Delta t_n \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}\left(\boldsymbol{u}_j^{(n)},\ \boldsymbol{\mu},\ t_{n-1} + c_j \Delta t_n\right)^T \boldsymbol{\kappa}_j^{(n)}
\end{aligned} \tag{D.59}$$

for $n = 1, \ldots, N_t$ and $i = 1, \ldots, s$. The gradient of $j(\boldsymbol{u}_0)$ is reconstructed from the dual variables as

$$\frac{\mathrm{d}F}{\mathrm{d}\boldsymbol{\mu}} = {\boldsymbol{\lambda}^{(0)}}^T + \boldsymbol{u}_0 - \boldsymbol{u}^{(N_t)}(\boldsymbol{u}_0;\boldsymbol{\mu}). \tag{D.60}$$

See [211] for the derivation. These methods have been used with considerable success to solve a variety of time-periodic partial differential equations, including the Benjamin-Ono equation [7], a

wave-guide array mode-locked laser system [202], and the vortex sheet with surface tension [8]. Unfortunately, the underlying optimization algorithms suffer from relatively slow convergence, requiring many line-searches before becoming superlinear, and never achieve quadratic convergence.

An attractive alternative is to recast the fixed point iteration as a nonlinear system of equations and use the Newton-Raphson method to reap the benefits of quadratic convergence. To this end, define the nonlinear system of equations

$$\boldsymbol{R}(\boldsymbol{u}_0) = \boldsymbol{u}^{(N_t)}(\boldsymbol{u}_0;\boldsymbol{\mu}) - \boldsymbol{u}_0 = 0 \tag{D.61}$$

with Jacobian matrix

$$\boldsymbol{J}(\boldsymbol{u}_0) = \frac{\partial \boldsymbol{R}}{\partial \boldsymbol{u}_0}(\boldsymbol{u}_0) = \frac{\partial \boldsymbol{u}^{(N_t)}}{\partial \boldsymbol{u}_0}(\boldsymbol{u}_0;\boldsymbol{\mu}) - \boldsymbol{I} \tag{D.62}$$

where $\boldsymbol{I}$ is the $N_{\boldsymbol{u}} \times N_{\boldsymbol{u}}$ identity matrix. The crucial component of the Newton-Raphson method is the solution of a linear system of equations with the Jacobian (D.62), i.e., the solution of $\boldsymbol{J}(\boldsymbol{u}_0)\boldsymbol{x} = \boldsymbol{b}$, given $\boldsymbol{u}_0 \in \mathbb{R}^{N_{\boldsymbol{u}}}$ and $\boldsymbol{b} \in \mathbb{R}^{N_{\boldsymbol{u}}}$. A *linear* evolution equation defining $\dfrac{\partial \boldsymbol{u}^{(N_t)}}{\partial \boldsymbol{u}_0}$, i.e., the sensitivity of the final state with respect to perturbations in the initial state, is introduced by linearizing the fully discrete evolution equation in (D.55) about the primal state $\boldsymbol{u}^{(n)}$, $\boldsymbol{k}_i^{(n)}$ with respect to the initial state $\boldsymbol{u}_0$. Direct differentiation of (D.55) with respect to $\boldsymbol{u}_0$ leads to the forward sensitivity equations

$$\begin{aligned}
\frac{\partial \boldsymbol{u}^{(0)}}{\partial \boldsymbol{u}_0} &= \boldsymbol{I} \\
\frac{\partial \boldsymbol{u}^{(n)}}{\partial \boldsymbol{u}_0} &= \frac{\partial \boldsymbol{u}^{(n-1)}}{\partial \boldsymbol{u}_0} + \sum_{i=1}^{s} b_i \frac{\partial \boldsymbol{k}_i^{(n)}}{\partial \boldsymbol{u}_0} \\
\boldsymbol{M}\frac{\partial \boldsymbol{k}_i^{(n)}}{\partial \boldsymbol{u}_0} &= \Delta t_n \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}\left(\boldsymbol{u}_i^{(n)},\ \boldsymbol{\mu},\ t_{n-1} + c_i \Delta t_n\right)\left[\frac{\partial \boldsymbol{u}^{(n-1)}}{\partial \boldsymbol{u}_0} + \sum_{j=1}^{i} a_{ij}\frac{\partial \boldsymbol{k}_j^{(n)}}{\partial \boldsymbol{u}_0}\right].
\end{aligned} \tag{D.63}$$

In general, $\dfrac{\partial \boldsymbol{u}^{(N_t)}}{\partial \boldsymbol{u}_0}$ is a large ($N_{\boldsymbol{u}} \times N_{\boldsymbol{u}}$), dense matrix that requires the solution of $N_{\boldsymbol{u}}$ linear evolution equations to form. While it is true that the columns of the matrix can be solved in parallel, formation and storage of this matrix may be impractical, particularly for the large-scale computational fluid dynamics problems that motivate this work. For non-dissipative problems such as standing waves in the free-surface Euler equations [201, 176], this is worth the expense since all perturbation directions have to be explored (as opposed to letting the evolution over a cycle damp out high frequency transients). But for viscous problems such as those studied in the numerical experiments, solving the Newton-Raphson equations by Krylov subspace methods requires many fewer iterations than there are columns of the Jacobian.

Formation and storage of $\dfrac{\partial \boldsymbol{u}^{(N_t)}}{\partial \boldsymbol{u}_0}$ can be completely avoided if a matrix-free Krylov method [106] is used to solve the linear systems arising in the Newton-Raphson method, i.e., $\boldsymbol{J}(\boldsymbol{u}_0)\boldsymbol{x} = \boldsymbol{b}$. In this

case, only matrix-vector products of the form

$$\boldsymbol{J}(\boldsymbol{u}_0)\boldsymbol{v} = \frac{\partial \boldsymbol{R}}{\partial \boldsymbol{u}_0}(\boldsymbol{u}_0)\boldsymbol{v} = \frac{\partial \boldsymbol{u}^{(N_t)}}{\partial \boldsymbol{u}_0}(\boldsymbol{u}_0;\boldsymbol{\mu})\boldsymbol{v} - \boldsymbol{v} \tag{D.64}$$

for any $\boldsymbol{v} \in \mathbb{R}^{N_{\boldsymbol{u}}}$, are required. For efficiency, these must be computed without explicitly forming the matrix $\frac{\partial \boldsymbol{u}^{(N_t)}}{\partial \boldsymbol{u}_0}$. This is accomplished by considering the forward sensitivity equations in (D.63) in the direction defined by $\boldsymbol{v}$. Multiplying (D.63) by the vector $\boldsymbol{v}$ leads to the system of linear evolution equations

$$\begin{aligned}
\frac{\partial \boldsymbol{u}^{(0)}}{\partial \boldsymbol{u}_0}\boldsymbol{v} &= \boldsymbol{v} \\
\frac{\partial \boldsymbol{u}^{(n)}}{\partial \boldsymbol{u}_0}\boldsymbol{v} &= \frac{\partial \boldsymbol{u}^{(n-1)}}{\partial \boldsymbol{u}_0}\boldsymbol{v} + \sum_{i=1}^{s} b_i \frac{\partial \boldsymbol{k}_i^{(n)}}{\partial \boldsymbol{u}_0}\boldsymbol{v} \\
\boldsymbol{M}\frac{\partial \boldsymbol{k}_i^{(n)}}{\partial \boldsymbol{u}_0}\boldsymbol{v} &= \Delta t_n \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}\left(\boldsymbol{u}_i^{(n)},\ \boldsymbol{\mu},\ t_{n-1} + c_i \Delta t_n\right)\left[\frac{\partial \boldsymbol{u}^{(n-1)}}{\partial \boldsymbol{u}_0}\boldsymbol{v} + \sum_{j=1}^{i} a_{ij}\frac{\partial \boldsymbol{k}_j^{(n)}}{\partial \boldsymbol{u}_0}\boldsymbol{v}\right].
\end{aligned} \tag{D.65}$$

These can be solved for $\frac{\partial \boldsymbol{u}^{(n)}}{\partial \boldsymbol{u}_0}\cdot\boldsymbol{v}$ and $\frac{\partial \boldsymbol{k}_i^{(n)}}{\partial \boldsymbol{u}_0}\cdot\boldsymbol{v}$ directly, only requiring *one* linear evolution for each $\boldsymbol{v}$. Since the equations in (D.65) are linear, the underlying linear solver must be converged to high accuracy if accurate sensitivities are to be obtained. This mitigates the speedup with respect to the nonlinear, primal solves whose linear systems are usually solved to low precision. For the problems considered in Section D.4.4–D.4.5, the primal equations were, on average, 2 times more expensive than the sensitivity equations, even though 5 nonlinear iterations were required for convergence. This implies the cost of evaluating $\boldsymbol{R}(\boldsymbol{u}_0)$ is approximately 2 times as expensive as a Jacobian-vector product $\boldsymbol{J}(\boldsymbol{u}_0)\boldsymbol{v}$. The Newton-Krylov method, with Jacobian-vector products computed as the solution of (D.65), is summarized in Algorithm 22. If the linear system of equations arising at each iteration is solved to sufficient accuracy, this algorithm will converge quadratically. The starting guess can be obtained by fixed point iteration (Algorithm 21), or, if solutions come in families parametrized by an amplitude, by numerical continuation [76, 103, 53, 7, 8, 200, 176]. The latter approach is particularly useful if the system is not dissipative and externally driven, as fixed point iteration relies on transient modes being damped by the evolution equations, i.e., on the periodic solution being stable and attracting.

Given this exposition on methods for computing time-periodic solutions of partial differential equations, we turn our attention to determining the *stability* of the corresponding periodic orbit.

D.3.2 Stability of Periodic Orbits of Fully Discrete Partial Differential Equations

In this section, the concept of stability of a periodic orbit of fully discrete partial differential equations is introduced and a method for determining the stability of a periodic solution presented. Recall

Algorithm 22 Newton-Krylov Shooting Method for Time-Periodic Solutions of PDE

Input: Initial guess for periodic initial condition, $\boldsymbol{u}_0$; parameter configuration, $\boldsymbol{\mu}$
Output: Periodic initial condition, $\boldsymbol{u}^{(0)}$

1: **while** $\left\|\boldsymbol{u}^{(N_t)}(\boldsymbol{u}_0;\boldsymbol{\mu}) - \boldsymbol{u}_0\right\|_2 > \epsilon$ **do**
2: Solve *unsymmetric* linear system of equations

$$\frac{\partial \boldsymbol{u}^{(N_t)}}{\partial \boldsymbol{u}_0}(\boldsymbol{u}_0;\boldsymbol{\mu}) \cdot \Delta\boldsymbol{u} = \boldsymbol{u}^{(N_t)}(\boldsymbol{u}_0;\boldsymbol{\mu}) - \boldsymbol{u}_0$$

using a *matrix-free* Krylov method with matrix-vector products

$$\frac{\partial \boldsymbol{u}^{(N_t)}}{\partial \boldsymbol{u}_0}(\boldsymbol{u}_0;\boldsymbol{\mu}) \cdot \boldsymbol{v}$$

computed as the solution of the directional sensitivity equations (D.65)
3: Update solution

$$\boldsymbol{u}_0 \leftarrow \boldsymbol{u}_0 - \Delta\boldsymbol{u}$$

4: **end while**
5: Define periodic initial condition

$$\boldsymbol{u}^{(0)} = \boldsymbol{u}_0$$

the interpretation of $\boldsymbol{u}^{(N_t)}$ as a function that propagates an initial condition $\boldsymbol{u}_0$ to its final state $\boldsymbol{u}^{(N_t)}(\boldsymbol{u}_0;\ \boldsymbol{\mu})$. Let $\boldsymbol{u}_0^*(\boldsymbol{\mu})$ be the time-periodic solution of the fully discrete partial differential equation in (D.52), (D.54) at parameter configuration $\boldsymbol{\mu}$, i.e., $\boldsymbol{u}_0^*(\boldsymbol{\mu}) = \boldsymbol{u}^{(N_t)}(\boldsymbol{u}_0^*;\ \boldsymbol{\mu})$. A periodic orbit is stable if there is a $\delta > 0$ such that

$$\lim_{n\to\infty} \left\|\boldsymbol{u}^{(n\cdot N_t)}(\boldsymbol{u}_0^*(\boldsymbol{\mu}) + \Delta\boldsymbol{u};\ \boldsymbol{\mu}) - \boldsymbol{u}_0^*(\boldsymbol{\mu})\right\| = 0 \tag{D.66}$$

if $\|\Delta\boldsymbol{u}\| < \delta$, where

$$\boldsymbol{u}^{(n\cdot N_t)}(\boldsymbol{u}_0;\ \boldsymbol{\mu}) = \boldsymbol{u}^{(N_t)}(\cdot;\ \boldsymbol{\mu}) \circ \cdots \circ \boldsymbol{u}^{(N_t)}(\boldsymbol{u}_0;\ \boldsymbol{\mu}). \tag{D.67}$$

A Taylor expansion of $\boldsymbol{u}^{(N_t)}$ about the periodic solution leads to

$$\boldsymbol{u}^{(N_t)}(\boldsymbol{u}_0^*(\boldsymbol{\mu});\ \boldsymbol{\mu}) = \boldsymbol{u}_0^*(\boldsymbol{\mu}) + \frac{\partial \boldsymbol{u}^{(N_t)}}{\partial \boldsymbol{u}_0}(\boldsymbol{u}_0^*(\boldsymbol{\mu});\ \boldsymbol{\mu}) \cdot \Delta\boldsymbol{u} + \mathcal{O}(\|\Delta\boldsymbol{u}\|^2) \tag{D.68}$$

where time-periodicity of $\boldsymbol{u}_0^*(\boldsymbol{\mu})$ was used. Repeated application of (D.68) leads to

$$\boldsymbol{u}^{(n\cdot N_t)}(\boldsymbol{u}_0^*(\boldsymbol{\mu}) + \Delta\boldsymbol{u};\ \boldsymbol{\mu}) = \boldsymbol{u}_0^*(\boldsymbol{\mu}) + \left[\frac{\partial \boldsymbol{u}^{(N_t)}}{\partial \boldsymbol{u}_0}(\boldsymbol{u}_0^*(\boldsymbol{\mu});\ \boldsymbol{\mu})\right]^n \Delta\boldsymbol{u} + \mathcal{O}(\|\Delta\boldsymbol{u}\|^{n+1}). \tag{D.69}$$

Taking $\delta < 1$, the stability criteria in (D.66) is satisfied if all eigenvalues of $\frac{\partial \boldsymbol{u}^{(N_t)}}{\partial \boldsymbol{u}_0}(\boldsymbol{u}_0^*(\boldsymbol{\mu});\ \boldsymbol{\mu})$ have modulus strictly less than 1. In Section D.4.4, the stability of the periodic flow around a flapping airfoil is verified using this method.

D.4 Fully Discrete, Time-Periodic Adjoint Method

In this section, the adjoint equations corresponding to the fully discrete time-periodically constrained partial differential equations (D.52), (D.54) and quantity of interest $F(\boldsymbol{u}^{(0)}, \dots, \boldsymbol{u}^{(N_t)}, \boldsymbol{k}_1^{(1)}, \dots, \boldsymbol{k}_s^{(N_t)}, \boldsymbol{\mu})$, will be derived. For the remainder of this section, $\boldsymbol{u}^{(0)}, \dots, \boldsymbol{u}^{(N_t)}, \boldsymbol{k}_1^{(1)}, \dots, \boldsymbol{k}_s^{(N_t)}$ will be taken as the time-periodic solution of the fully discrete partial differential equations (D.52), (D.54) at parameter $\boldsymbol{\mu}$. The adjoint equations will be derived by linearizing the fully discrete equations about this periodic solution. This highlights the importance of an efficient periodic solver—the subject of Section D.3.1—as it is a prerequisite for the adjoint method.

Before proceeding to the derivation of the adjoint equations, the following definitions are introduced for the fully discrete time-periodic constraint and Runge-Kutta stage equations and state updates

$$\begin{aligned}
\tilde{\boldsymbol{r}}^{(0)}(\boldsymbol{u}^{(0)},\ \boldsymbol{u}^{(N_t)}) &= \boldsymbol{u}^{(0)} - \boldsymbol{u}^{(N_t)} = 0 \\
\tilde{\boldsymbol{r}}^{(n)}(\boldsymbol{u}^{(n-1)},\ \boldsymbol{u}^{(n)},\ \boldsymbol{k}_1^{(n)}, \dots, \boldsymbol{k}_s^{(n)},\ \boldsymbol{\mu}) &= \boldsymbol{u}^{(n)} - \boldsymbol{u}^{(n-1)} - \sum_{i=1}^{s} b_i \boldsymbol{k}_i^{(i)} = 0 \\
\boldsymbol{R}_i^{(n)}(\boldsymbol{u}^{(n-1)}, \boldsymbol{k}_1^{(n)}, \dots, \boldsymbol{k}_i^{(n)}, \boldsymbol{\mu}) &= \boldsymbol{M}\boldsymbol{k}_i^{(n)} - \Delta t_n \boldsymbol{r}\left(\boldsymbol{u}_i^{(n)},\ \boldsymbol{\mu},\ t_{n-1} + c_i \Delta t_n\right) = 0
\end{aligned} \tag{D.70}$$

for $n = 1, \dots, n$ and $i = 1, \dots, s$.

D.4.1 Derivation

The derivation of the fully discrete adjoint equations corresponding to the output functional, F, begins with the introduction of test variables

$$\boldsymbol{\lambda}^{(0)},\ \boldsymbol{\lambda}^{(n)},\ \boldsymbol{\kappa}_i^{(n)} \in \mathbb{R}^{N_{\boldsymbol{u}}} \tag{D.71}$$

for $n = 1, \dots, N_t$ and $i = 1, \dots, s$. Since $\boldsymbol{u}^{(0)}, \dots, \boldsymbol{u}^{(N_t)}, \boldsymbol{k}_1^{(1)}, \dots, \boldsymbol{k}_s^{(N_t)}$ are taken as the solution of the fully discrete time-periodic problem in (D.70), the following identity holds, *for any* $\boldsymbol{\mu} \in \mathbb{R}^{N_{\boldsymbol{\mu}}}$,

$$F = F + 0 = F - {\boldsymbol{\lambda}^{(0)}}^T \tilde{\boldsymbol{r}}^{(0)} - \sum_{n=1}^{N_t} {\boldsymbol{\lambda}^{(n)}}^T \tilde{\boldsymbol{r}}^{(n)} - \sum_{n=1}^{N_t} \sum_{i=1}^{s} {\boldsymbol{\kappa}_i^{(n)}}^T \boldsymbol{R}_i^{(n)} \tag{D.72}$$

for *any* value of the test functions $\boldsymbol{\lambda}^{(n)}$ and $\boldsymbol{\kappa}_i^{(n)}$. In (D.72), arguments have been dropped for brevity; it is understood that all terms are evaluated at the periodic solution of (D.52), (D.54) at parameter $\boldsymbol{\mu}$. Since (D.70) holds for any $\boldsymbol{\mu} \in \mathbb{R}^{N_{\boldsymbol{\mu}}}$, provided $\boldsymbol{u}^{(0)}, \dots, \boldsymbol{u}^{(N_t)}, \boldsymbol{k}_1^{(1)}, \dots, \boldsymbol{k}_s^{(N_t)}$ is the

corresponding periodic solution, differentiation with respect to $\boldsymbol{\mu}$ leads to

$$\begin{aligned}\frac{\mathrm{d}F}{\mathrm{d}\boldsymbol{\mu}} = \frac{\partial F}{\partial \boldsymbol{\mu}} &+ \sum_{n=0}^{N_t} \frac{\partial F}{\partial \boldsymbol{u}^{(n)}} \frac{\partial \boldsymbol{u}^{(n)}}{\partial \boldsymbol{\mu}} + \sum_{n=1}^{N_t} \sum_{i=1}^{s} \frac{\partial F}{\partial \boldsymbol{k}_i^{(n)}} \frac{\partial \boldsymbol{k}_i^{(n)}}{\partial \boldsymbol{\mu}} - \boldsymbol{\lambda}^{(0)^T} \left[\frac{\partial \tilde{\boldsymbol{r}}^{(0)}}{\partial \boldsymbol{\mu}} + \frac{\partial \tilde{\boldsymbol{r}}^{(0)}}{\partial \boldsymbol{u}^{(0)}} \frac{\partial \boldsymbol{u}^{(0)}}{\partial \boldsymbol{\mu}} + \frac{\partial \tilde{\boldsymbol{r}}^{(0)}}{\partial \boldsymbol{u}^{(N_t)}} \frac{\partial \boldsymbol{u}^{(N_t)}}{\partial \boldsymbol{\mu}} \right] \\ &- \sum_{n=1}^{N_t} \boldsymbol{\lambda}^{(n)^T} \left[\frac{\partial \tilde{\boldsymbol{r}}^{(n)}}{\partial \boldsymbol{\mu}} + \frac{\partial \tilde{\boldsymbol{r}}^{(n)}}{\partial \boldsymbol{u}^{(n)}} \frac{\partial \boldsymbol{u}^{(n)}}{\partial \boldsymbol{\mu}} + \frac{\partial \tilde{\boldsymbol{r}}^{(n)}}{\partial \boldsymbol{u}^{(n-1)}} \frac{\partial \boldsymbol{u}^{(n-1)}}{\partial \boldsymbol{\mu}} + \sum_{p=1}^{s} \frac{\partial \tilde{\boldsymbol{r}}^{(n)}}{\partial \boldsymbol{k}_p^{(n)}} \frac{\partial \boldsymbol{k}_p^{(n)}}{\partial \boldsymbol{\mu}} \right] \\ &- \sum_{n=1}^{N_t} \sum_{i=1}^{s} \boldsymbol{\kappa}_i^{(n)^T} \left[\frac{\partial \boldsymbol{R}_i^{(n)}}{\partial \boldsymbol{\mu}} + \frac{\partial \boldsymbol{R}_i^{(n)}}{\partial \boldsymbol{u}^{(n-1)}} \frac{\partial \boldsymbol{u}^{(n-1)}}{\partial \boldsymbol{\mu}} + \sum_{j=1}^{i} \frac{\partial \boldsymbol{R}_i^{(n)}}{\partial \boldsymbol{k}_j^{(n)}} \frac{\partial \boldsymbol{k}_j^{(n)}}{\partial \boldsymbol{\mu}} \right].\end{aligned} \tag{D.73}$$

Re-arrangement of terms in (D.73) such that the state variable sensitivities are isolated leads to the following expression for $\dfrac{\mathrm{d}F}{\mathrm{d}\boldsymbol{\mu}}$

$$\begin{aligned}\frac{\mathrm{d}F}{\mathrm{d}\boldsymbol{\mu}} = \frac{\partial F}{\partial \boldsymbol{\mu}} &+ \left[\frac{\partial F}{\partial \boldsymbol{u}^{(N_t)}} - \boldsymbol{\lambda}^{(N_t)^T} \frac{\partial \tilde{\boldsymbol{r}}^{(N_t)}}{\partial \boldsymbol{u}^{(N_t)}} - \boldsymbol{\lambda}^{(0)^T} \frac{\partial \tilde{\boldsymbol{r}}^{(0)}}{\partial \boldsymbol{u}^{(N_t)}} \right] \frac{\partial \boldsymbol{u}^{(N_t)}}{\partial \boldsymbol{\mu}} - \sum_{n=0}^{N_t} \boldsymbol{\lambda}^{(n)^T} \frac{\partial \tilde{\boldsymbol{r}}^{(n)}}{\partial \boldsymbol{\mu}} - \sum_{n=1}^{N_t} \sum_{p=1}^{s} \boldsymbol{\kappa}_p^{(n)^T} \frac{\partial \boldsymbol{R}_p^{(n)}}{\partial \boldsymbol{\mu}} \\ &+ \sum_{n=1}^{N_t} \left[\frac{\partial F}{\partial \boldsymbol{u}^{(n-1)}} - \boldsymbol{\lambda}^{(n-1)^T} \frac{\partial \tilde{\boldsymbol{r}}^{(n-1)}}{\partial \boldsymbol{u}^{(n-1)}} - \boldsymbol{\lambda}^{(n)^T} \frac{\partial \tilde{\boldsymbol{r}}^{(n)}}{\partial \boldsymbol{u}^{(n-1)}} - \sum_{i=1}^{s} \boldsymbol{\kappa}_i^{(n)^T} \frac{\partial \boldsymbol{R}_i^{(n)}}{\partial \boldsymbol{u}^{(n-1)}} \right] \frac{\partial \boldsymbol{u}^{(n-1)}}{\partial \boldsymbol{\mu}} \\ &+ \sum_{n=1}^{N_t} \sum_{p=1}^{s} \left[\frac{\partial F}{\partial \boldsymbol{k}_i^{(n)}} - \boldsymbol{\lambda}^{(n)^T} \frac{\partial \tilde{\boldsymbol{r}}^{(n)}}{\partial \boldsymbol{k}_p^{(n)}} - \sum_{i=p}^{s} \boldsymbol{\kappa}_i^{(n)^T} \frac{\partial \boldsymbol{R}_i^{(n)}}{\partial \boldsymbol{k}_p^{(n)}} \right] \frac{\partial \boldsymbol{k}_p^{(n)}}{\partial \boldsymbol{\mu}}.\end{aligned} \tag{D.74}$$

The dual variables, $\boldsymbol{\lambda}^{(n)}$ and $\boldsymbol{\kappa}_i^{(n)}$, which have remained arbitrary to this point, are chosen such that the bracketed terms in (D.74) vanish

$$\begin{aligned}\frac{\partial \tilde{\boldsymbol{r}}^{(0)}}{\partial \boldsymbol{u}^{(N_t)}}^T \boldsymbol{\lambda}^{(0)} + \frac{\partial \tilde{\boldsymbol{r}}^{(N_t)}}{\partial \boldsymbol{u}^{(N_t)}}^T \boldsymbol{\lambda}^{(N_t)} &= \frac{\partial F}{\partial \boldsymbol{u}^{(N_t)}}^T \\ \frac{\partial \tilde{\boldsymbol{r}}^{(n)}}{\partial \boldsymbol{u}^{(n-1)}}^T \boldsymbol{\lambda}^{(n)} + \frac{\partial \tilde{\boldsymbol{r}}^{(n-1)}}{\partial \boldsymbol{u}^{(n-1)}}^T \boldsymbol{\lambda}^{(n-1)} &= \frac{\partial F}{\partial \boldsymbol{u}^{(n-1)}}^T - \sum_{i=1}^{s} \frac{\partial \boldsymbol{R}_i^{(n)}}{\partial \boldsymbol{u}^{(n-1)}}^T \boldsymbol{\kappa}_i^{(n)} \\ \sum_{j=i}^{s} \frac{\partial \boldsymbol{R}_j^{(n)}}{\partial \boldsymbol{k}_i^{(n)}}^T \boldsymbol{\kappa}_j^{(n)} &= \frac{\partial F}{\partial \boldsymbol{k}_i^{(n)}} - \frac{\partial \tilde{\boldsymbol{r}}^{(n)}}{\partial \boldsymbol{k}_i^{(n)}}^T \boldsymbol{\lambda}^{(n)}\end{aligned} \tag{D.75}$$

for $n = 1, \ldots, N_t$ and $i = 1, \ldots, s$. These are the *fully discrete adjoint equations* corresponding to the time-periodic primal evolution equations in (D.70), discrete quantity of interest F, and parameter $\boldsymbol{\mu}$. Defining the dual variables as the solution of the adjoint equations in (D.75), the expression for $\dfrac{\mathrm{d}F}{\mathrm{d}\boldsymbol{\mu}}$ in (D.74) reduces to

$$\frac{\mathrm{d}F}{\mathrm{d}\boldsymbol{\mu}} = \frac{\partial F}{\partial \boldsymbol{\mu}} - \sum_{n=0}^{N_t} \boldsymbol{\lambda}^{(n)^T} \frac{\partial \tilde{\boldsymbol{r}}^{(n)}}{\partial \boldsymbol{\mu}} - \sum_{n=1}^{N_t} \sum_{p=1}^{s} \boldsymbol{\kappa}_p^{(n)^T} \frac{\partial \boldsymbol{R}_p^{(n)}}{\partial \boldsymbol{\mu}}. \tag{D.76}$$

This provides a means of computing the total derivative $\dfrac{\mathrm{d}F}{\mathrm{d}\boldsymbol{\mu}}$ without explicitly computing the large, dense state sensitivities since the expression in (D.76) is independent of them. Direct differentiation of $\tilde{\boldsymbol{r}}^{(n)}$ and $\boldsymbol{R}_i^{(n)}$ from their definitions in (D.70) leads to the final form of the adjoint equations of the fully discrete, time-periodically constrained partial differential equations in (D.52), (D.54)

$$\begin{aligned}
\boldsymbol{\lambda}^{(N_t)} &= \boldsymbol{\lambda}^{(0)} + \frac{\partial F}{\partial \boldsymbol{u}^{(N_t)}}^T \\
\boldsymbol{\lambda}^{(n-1)} &= \boldsymbol{\lambda}^{(n)} + \frac{\partial F}{\partial \boldsymbol{u}^{(n-1)}}^T + \sum_{i=1}^{s} \Delta t_n \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}} \left(\boldsymbol{u}_i^{(n)},\ \boldsymbol{\mu},\ t_{n-1} + c_i \Delta t_n\right)^T \boldsymbol{\kappa}_i^{(n)} \\
\boldsymbol{M}^T \boldsymbol{\kappa}_i^{(n)} &= \frac{\partial F}{\partial \boldsymbol{k}_i^{(n)}}^T + b_i \boldsymbol{\lambda}^{(n)} + \sum_{j=i}^{s} a_{ji} \Delta t_n \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}} \left(\boldsymbol{u}_j^{(n)},\ \boldsymbol{\mu},\ t_{n-1} + c_j \Delta t_n\right)^T \boldsymbol{\kappa}_j^{(n)}
\end{aligned} \tag{D.77}$$

for $n = 1, \ldots, N_t$ and $i = 1, \ldots, s$. Similarly, the total derivative of F, independent of state sensitivities, takes the form

$$\frac{\mathrm{d}F}{\mathrm{d}\boldsymbol{\mu}} = \frac{\partial F}{\partial \boldsymbol{\mu}} + \sum_{n=1}^{N_t} \Delta t_n \sum_{i=1}^{s} \boldsymbol{\kappa}_i^{(n)^T} \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{\mu}} (\boldsymbol{u}_i^{(n)},\ \boldsymbol{\mu},\ t_{n-1} + c_i \Delta t_n). \tag{D.78}$$

From (D.77), it can be seen that the fully discrete adjoint equations take the form of a *linear, two-point boundary-value problem* and cannot be solved directly as an evolution equation. D.6 proves existence and uniqueness of solutions to (D.77). The next section will discuss solvers for the discrete time-periodic adjoint equations in (D.77).

D.4.2 Numerical Solver: Matrix-Free Krylov Method

As the adjoint equations corresponding to the fully discrete time-periodic partial differential equation are linear, this section will consider matrix-free Krylov methods to solve them. Alternatively, any of the methods discussed in Section D.3.1 could be used.

Define $\boldsymbol{\lambda}^{(0)}(\boldsymbol{\lambda}_{N_t}; \boldsymbol{\mu}, t)$ as the solution of the linear, backward evolution equations

$$\begin{aligned}
\boldsymbol{\lambda}^{(N_t)} &= \boldsymbol{\lambda}_{N_t} \\
\boldsymbol{\lambda}^{(n-1)} &= \boldsymbol{\lambda}^{(n)} + \frac{\partial F}{\partial \boldsymbol{u}^{(n-1)}}^T + \sum_{i=1}^{s} \Delta t_n \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}} \left(\boldsymbol{u}_i^{(n)},\ \boldsymbol{\mu},\ t_{n-1} + c_i \Delta t_n\right)^T \boldsymbol{\kappa}_i^{(n)} \\
\boldsymbol{M}^T \boldsymbol{\kappa}_i^{(n)} &= \frac{\partial F}{\partial \boldsymbol{k}_i^{(n)}} + b_i \boldsymbol{\lambda}^{(n)} + \sum_{j=i}^{s} a_{ji} \Delta t_n \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}} \left(\boldsymbol{u}_j^{(n)},\ \boldsymbol{\mu},\ t_{n-1} + c_j \Delta t_n\right)^T \boldsymbol{\kappa}_j^{(n)},
\end{aligned} \tag{D.79}$$

which can be directly evolved, backward-in-time. Similar to Section D.3.1 this constitutes a notation overload since $\boldsymbol{\lambda}^{(0)} \in \mathbb{R}^{N_{\boldsymbol{u}}}$ is the initial solution of the adjoint equations corresponding to the fully discrete periodic partial differential equations, as well as the linear function that takes a state $\boldsymbol{\lambda}_{N_t}$ to $\boldsymbol{\lambda}^{(0)}(\boldsymbol{\lambda}_{N_t}; \boldsymbol{\mu})$. Then, $\boldsymbol{\lambda}^{(0)}(\boldsymbol{\lambda}_{N_t}; \boldsymbol{\mu})$ is the initial solution of (D.77) if the following *linear* equation

is satisfied

$$\boldsymbol{\lambda}^{(0)}(\boldsymbol{\lambda}_{N_t};\boldsymbol{\mu},t) = \boldsymbol{\lambda}_{N_t} - \frac{\partial F}{\partial \boldsymbol{u}^{(N_t)}}^T. \tag{D.80}$$

This is a linear system of equations of the form, $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ where

$$\boldsymbol{A} = \frac{\partial \boldsymbol{\lambda}^{(0)}}{\partial \boldsymbol{\lambda}_{N_t}} - \boldsymbol{I}. \tag{D.81}$$

The columns of the linear operator $\boldsymbol{A}$ can be formed by considering perturbations of (D.79) with respect to the final state $\boldsymbol{\lambda}_{N_t}$. Differentiation of (D.79) with respect to $\boldsymbol{\lambda}_{N_t}$ leads to the adjoint sensitivity equations

$$\begin{aligned}
\frac{\partial \boldsymbol{\lambda}^{(N_t)}}{\partial \boldsymbol{\lambda}_{N_t}} &= \boldsymbol{I} \\
\frac{\partial \boldsymbol{\lambda}^{(n-1)}}{\partial \boldsymbol{\lambda}_{N_t}} &= \frac{\partial \boldsymbol{\lambda}^{(n)}}{\partial \boldsymbol{\lambda}_{N_t}} + \sum_{i=1}^{s} \Delta t_n \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}} \left(\boldsymbol{u}_i^{(n)},\ \boldsymbol{\mu},\ t_{n-1} + c_i \Delta t_n\right)^T \frac{\partial \boldsymbol{\kappa}_i^{(n)}}{\partial \boldsymbol{\lambda}_{N_t}} \\
\boldsymbol{M}^T \frac{\partial \boldsymbol{\kappa}_i^{(n)}}{\partial \boldsymbol{\lambda}_{N_t}} &= b_i \frac{\partial \boldsymbol{\lambda}^{(n)}}{\partial \boldsymbol{\lambda}_{N_t}} + \sum_{j=i}^{s} a_{ji} \Delta t_n \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}} \left(\boldsymbol{u}_j^{(n)},\ \boldsymbol{\mu},\ t_{n-1} + c_j \Delta t_n\right)^T \frac{\partial \boldsymbol{\kappa}_j^{(n)}}{\partial \boldsymbol{\lambda}_{N_t}}.
\end{aligned} \tag{D.82}$$

Similar to the situation for the primal problem, the matrix $\dfrac{\partial \boldsymbol{\lambda}^{(0)}}{\partial \boldsymbol{\lambda}_{N_t}}$ is an $N_{\boldsymbol{u}} \times N_{\boldsymbol{u}}$ dense matrix that requires $N_{\boldsymbol{u}}$ linear evolution equations to form. As this is impractical for large problems, a matrix-free Krylov method is used to solve (D.80), which only requires matrix-vector products of the form

$$\boldsymbol{A}\boldsymbol{v} = \frac{\partial \boldsymbol{\lambda}^{(0)}}{\partial \boldsymbol{\lambda}_{N_t}}\boldsymbol{v} - \boldsymbol{v}. \tag{D.83}$$

The first term in this matrix-vector product can be computed directly by considering the adjoint sensitivity equations in a given direction $\boldsymbol{v}$

$$\begin{aligned}
\frac{\partial \boldsymbol{\lambda}^{(N_t)}}{\partial \boldsymbol{\lambda}_{N_t}}\boldsymbol{v} &= \boldsymbol{v} \\
\frac{\partial \boldsymbol{\lambda}^{(n-1)}}{\partial \boldsymbol{\lambda}_{N_t}}\boldsymbol{v} &= \frac{\partial \boldsymbol{\lambda}^{(n)}}{\partial \boldsymbol{\lambda}_{N_t}}\boldsymbol{v} + \sum_{i=1}^{s} \Delta t_n \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}} \left(\boldsymbol{u}_i^{(n)},\ \boldsymbol{\mu},\ t_{n-1} + c_i \Delta t_n\right)^T \frac{\partial \boldsymbol{\kappa}_i^{(n)}}{\partial \boldsymbol{\lambda}_{N_t}}\boldsymbol{v} \\
\boldsymbol{M}^T \frac{\partial \boldsymbol{\kappa}_i^{(n)}}{\partial \boldsymbol{\lambda}_{N_t}}\boldsymbol{v} &= b_i \frac{\partial \boldsymbol{\lambda}^{(n)}}{\partial \boldsymbol{\lambda}_{N_t}}\boldsymbol{v} + \sum_{j=i}^{s} a_{ji} \Delta t_n \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}} \left(\boldsymbol{u}_j^{(n)},\ \boldsymbol{\mu},\ t_{n-1} + c_j \Delta t_n\right)^T \frac{\partial \boldsymbol{\kappa}_j^{(n)}}{\partial \boldsymbol{\lambda}_{N_t}}\boldsymbol{v}.
\end{aligned} \tag{D.84}$$

The equations in (D.84) can be solved for $\dfrac{\partial \boldsymbol{\lambda}^{(0)}}{\partial \boldsymbol{\lambda}_{N_t}} \cdot \boldsymbol{v}$ at the cost of one linear evolution solution for each $\boldsymbol{v}$. The adjoint sensitivity equations in (D.84) are *independent* of the quantity of interest, F. If there are multiple quantities of interest, fast multiple right-hand side solvers [182, 38, 79] could be used to solve $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ as the matrix $\boldsymbol{A}$ will be fixed and only the right-hand side varied. Furthermore, the adjoint sensitivity equations in (D.84) and the adjoint equations in (D.79) are identical, with the

exception of the terms $\frac{\partial F}{\partial \boldsymbol{u}^{(n-1)}}$ and $\frac{\partial F}{\partial \boldsymbol{k}_i^{(n)}}$. Therefore, the adjoint sensitivities are less expensive to compute than the adjoint states and the savings becomes substantial when the number of parameters in $\boldsymbol{\mu}$ is large since $\frac{\partial F}{\partial \boldsymbol{u}^{(n-1)}}$ and $\frac{\partial F}{\partial \boldsymbol{k}_i^{(n)}}$ become expensive to compute. Algorithm 23 below details the use of a matrix-free GMRES method to solve (D.80) with matrix-vector products defined by (D.84).

Algorithm 23 GMRES for Solution of Fully Discrete, Time-Periodic Adjoint PDE

Input: Initial guess for periodic adjoint final condition, $\boldsymbol{\lambda}_{N_t,0}$; parameter configuration, $\boldsymbol{\mu}$; periodic primal solution, $\boldsymbol{u}^{(0)}, \ldots, \boldsymbol{u}^{(N_t)}, \boldsymbol{k}_1^{(1)}, \ldots, \boldsymbol{k}_s^{(N_t)}$

Output: Periodic adjoint final condition, $\boldsymbol{\lambda}^{(N_t)}$

Compute

$$\boldsymbol{r}_0 = \boldsymbol{\lambda}^{(0)}(\boldsymbol{\lambda}_{N_t}, \boldsymbol{\mu}) + \frac{\partial F}{\partial \boldsymbol{u}^{(N_t)}}^T - \boldsymbol{\lambda}_{N_t,0}$$

Set $\beta = \|\boldsymbol{r}_0\|_2$, $\boldsymbol{v}_1 = \boldsymbol{r}_0/\beta$, and $\boldsymbol{\lambda}_{N_t} = \boldsymbol{\lambda}_{N_t,0}$

while $\left\| \boldsymbol{\lambda}^{(0)}(\boldsymbol{\lambda}_{N_t}, \boldsymbol{\mu}) + \frac{\partial F}{\partial \boldsymbol{u}^{(N_t)}}^T - \boldsymbol{\lambda}_{N_t} \right\|_2 > \epsilon$ **do**

  for j = 1, 2, . . . , m **do**

    Compute

$$\boldsymbol{w}_j = \frac{\partial \boldsymbol{\lambda}^{(0)}}{\partial \boldsymbol{\lambda}_{N_t}} \boldsymbol{v}_j - \boldsymbol{v}_j$$

    as the solution of the adjoint sensitivity equations (D.84)

    for i = 1, . . . , j **do**

      $h_{ij} = (\boldsymbol{w}_j, \boldsymbol{v}_i)$

      $\boldsymbol{w}_j = \boldsymbol{w}_j - h_{ij}\boldsymbol{v}_i$

    end for

    $h_{j+1,j} = \|\boldsymbol{w}_j\|_2$

    $\boldsymbol{v}_{j+1} = \boldsymbol{w}_j / h_{j+1,j}$

  end for

  Compute

$$\boldsymbol{y}_m = \arg\min \|\beta \boldsymbol{e}_1 - \boldsymbol{H}_m \boldsymbol{y}\|_2 ,$$

  where $\boldsymbol{e}_1$ is the first canonical until vector in $\mathbb{R}^{N_{\boldsymbol{u}}}$ and $\boldsymbol{H} = \{h_{ij}\}_{1 \le i \le m+1, 1 \le j \le m}$

  Update solution

$$\boldsymbol{\lambda}_{N_t} = \boldsymbol{\lambda}_{N_t,0} + \boldsymbol{V}_m \boldsymbol{y}_m$$

  where

$$\boldsymbol{V}_m = \begin{bmatrix} \boldsymbol{v}_1 & \cdots & \boldsymbol{v}_m \end{bmatrix}$$

end while

Define adjoint periodic final condition

$$\boldsymbol{\lambda}^{(N_t)} = \boldsymbol{\lambda}_{N_t}$$

With the solution of the fully discrete primal and dual time-periodic problems fully specified, from numerical discretization to solution algorithms, we close this section with an algorithm that uses the fully discrete adjoint method to compute the gradient of the quantity of interest *on the*

manifold of periodic solutions. First, the fully discrete time-periodic solution (D.52), (D.54) must be computed, e.g., using a matrix-free Newton-Krylov method, to yield $\boldsymbol{u}^{(0)}, \ldots, \boldsymbol{u}^{(N_t)}, \boldsymbol{k}_1^{(1)}, \ldots, \boldsymbol{k}_s^{(N_t)}$. Next, the corresponding fully discrete adjoint equations are defined about this periodic solution and solved, e.g., using a matrix-free Krylov method, for $\boldsymbol{\lambda}^{(0)}, \ldots, \boldsymbol{\lambda}^{(N_t)}, \boldsymbol{\kappa}_1^{(1)}, \ldots, \boldsymbol{\kappa}_s^{(N_t)}$. Finally, (D.78) is used to reconstruct the desired gradient $\dfrac{\mathrm{d}F}{\mathrm{d}\boldsymbol{\mu}}$. This procedure is summarized in Algorithm 24.

Algorithm 24 Gradients on Manifold of Time-Periodic Solutions of PDEs

Input: Parameter configuration, $\boldsymbol{\mu}$, and fully discrete quantity of interest, $F(\boldsymbol{u}^{(0)}, \ldots, \boldsymbol{u}^{(N_t)}, \boldsymbol{k}_1^{(1)}, \ldots, \boldsymbol{k}_s^{(N_t)})$

Output: Gradient, $\dfrac{\mathrm{d}F}{\mathrm{d}\boldsymbol{\mu}}$, on manifold of time-periodic solutions

1: For parameter $\boldsymbol{\mu}$, compute time-periodic solution of fully discrete PDE in (D.52), (D.54), e.g., using the Newton-Krylov shooting method in Algorithm 22

$$\boldsymbol{u}^{(0)}, \ldots, \boldsymbol{u}^{(N_t)}, \boldsymbol{k}_1^{(1)}, \ldots, \boldsymbol{k}_s^{(N_t)}$$

2: For fully discrete functional $F(\boldsymbol{u}^{(0)}, \ldots, \boldsymbol{u}^{(N_t)}, \boldsymbol{k}_1^{(1)}, \ldots, \boldsymbol{k}_s^{(N_t)})$, compute adjoint solution of fully discrete time-periodic PDE in (D.77), e.g., using GMRES shooting method in Algorithm 23 with matrix-vector products computed from the backward evolution of the adjoint sensitivity equations in (D.84)

$$\boldsymbol{\lambda}^{(0)}, \ldots, \boldsymbol{\lambda}^{(N_t)}, \boldsymbol{\kappa}_1^{(1)}, \ldots, \boldsymbol{\kappa}_s^{(N_t)}$$

3: Reconstruct $\dfrac{\mathrm{d}F}{\mathrm{d}\boldsymbol{\mu}}$ using dual variables according to (D.78)

D.4.3 Generalized Reduced-Gradient Method for PDE Optimization with Time-Periodicity Constraints

Consider the fully discrete time-dependent PDE-constrained optimization problem

$$\begin{aligned}
\underset{\substack{\boldsymbol{u}^{(0)},\ \ldots,\ \boldsymbol{u}^{(N_t)} \in \mathbb{R}^{N_{\boldsymbol{u}}},\\ \boldsymbol{k}_1^{(1)},\ \ldots,\ \boldsymbol{k}_s^{(N_t)} \in \mathbb{R}^{N_{\boldsymbol{u}}},\\ \boldsymbol{\mu} \in \mathbb{R}^{N_{\boldsymbol{\mu}}}}}{\text{minimize}} \quad & F(\boldsymbol{u}^{(0)},\ \ldots,\ \boldsymbol{u}^{(N_t)},\ \boldsymbol{k}_1^{(1)},\ \ldots,\ \boldsymbol{k}_s^{(N_t)},\ \boldsymbol{\mu}) \\
\text{subject to} \quad & \boldsymbol{c}(\boldsymbol{u}^{(0)},\ \ldots,\ \boldsymbol{u}^{(N_t)},\ \boldsymbol{k}_1^{(1)},\ \ldots,\ \boldsymbol{k}_s^{(N_t)},\ \boldsymbol{\mu}) \geq 0 \\
& \boldsymbol{u}^{(0)} = \boldsymbol{u}_0 \\
& \boldsymbol{u}^{(n)} = \boldsymbol{u}^{(n-1)} + \sum_{i=1}^{s} b_i \boldsymbol{k}_i^{(n)} \\
& \boldsymbol{M}\boldsymbol{k}_i^{(n)} = \Delta t_n \boldsymbol{r}\left(\boldsymbol{u}_i^{(n)},\ \boldsymbol{\mu},\ t_{n-1} + c_i \Delta t_n\right)
\end{aligned} \tag{D.85}$$

where F is a fully discrete output functional of the partial differential equation and $\boldsymbol{c}$ is a vector of such output functionals. The nested or Generalized Reduced-Gradient (GRG) approach to solve (D.85) explicitly enforces the PDE constraint at each optimization iteration. The implicit function theorem states that the solution of the discretized PDE, can be considered an implicit function of the

parameter $\boldsymbol{\mu}$, i.e., $\boldsymbol{u}^{(n)} = \boldsymbol{u}^{(n)}(\boldsymbol{\mu})$ and $\boldsymbol{k}_i^{(n)} = \boldsymbol{k}_i^{(n)}(\boldsymbol{\mu})$. Strict enforcement of the discretized partial differential equation allows the PDE variables and equations to be removed from the optimization problem

$$\begin{aligned} \underset{\boldsymbol{\mu}\in\mathbb{R}^{N_{\boldsymbol{\mu}}}}{\text{minimize}} \quad & F(\boldsymbol{u}^{(0)}(\boldsymbol{\mu}),\ \ldots,\ \boldsymbol{u}^{(N_t)}(\boldsymbol{\mu}),\ \boldsymbol{k}_1^{(1)}(\boldsymbol{\mu}),\ \ldots,\ \boldsymbol{k}_s^{(N_t)}(\boldsymbol{\mu}),\ \boldsymbol{\mu}) \\ \text{subject to} \quad & \boldsymbol{c}(\boldsymbol{u}^{(0)}(\boldsymbol{\mu}),\ \ldots,\ \boldsymbol{u}^{(N_t)}(\boldsymbol{\mu}),\ \boldsymbol{k}_1^{(1)}(\boldsymbol{\mu}),\ \ldots,\ \boldsymbol{k}_s^{(N_t)}(\boldsymbol{\mu}),\ \boldsymbol{\mu}) \geq 0. \end{aligned} \tag{D.86}$$

To solve this optimization problem using gradient-based techniques, the terms $\dfrac{\mathrm{d}F}{\mathrm{d}\boldsymbol{\mu}}$ and $\dfrac{\mathrm{d}\boldsymbol{c}}{\mathrm{d}\boldsymbol{\mu}}$—gradients of quantities of interest along the manifold of solutions of the PDE—are required. Depending on the relative number of variables in $\boldsymbol{\mu}$ to the number of constraints in $\boldsymbol{c}$, the direct or adjoint method can be efficiently used to compute these gradients *without relying on finite differences*.

Now consider the optimization problem in (D.85) with the time-periodicity constraint added

$$\begin{aligned} \underset{\substack{\boldsymbol{u}^{(0)},\ \ldots,\ \boldsymbol{u}^{(N_t)}\in\mathbb{R}^{N_{\boldsymbol{u}}},\\ \boldsymbol{k}_1^{(1)},\ \ldots,\ \boldsymbol{k}_s^{(N_t)}\in\mathbb{R}^{N_{\boldsymbol{u}}},\\ \boldsymbol{\mu}\in\mathbb{R}^{N_{\boldsymbol{\mu}}}}}{\text{minimize}} \quad & F(\boldsymbol{u}^{(0)},\ \ldots,\ \boldsymbol{u}^{(N_t)},\ \boldsymbol{k}_1^{(1)},\ \ldots,\ \boldsymbol{k}_s^{(N_t)},\ \boldsymbol{\mu}) \\ \text{subject to} \quad & \boldsymbol{c}(\boldsymbol{u}^{(0)},\ \ldots,\ \boldsymbol{u}^{(N_t)},\ \boldsymbol{k}_1^{(1)},\ \ldots,\ \boldsymbol{k}_s^{(N_t)},\ \boldsymbol{\mu}) \geq 0 \\ & \boldsymbol{u}^{(0)} = \boldsymbol{u}^{(N_t)} \\ & \boldsymbol{u}^{(n)} = \boldsymbol{u}^{(n-1)} + \sum_{i=1}^{s} b_i \boldsymbol{k}_i^{(n)} \\ & \boldsymbol{M}\boldsymbol{k}_i^{(n)} = \Delta t_n \boldsymbol{r}\left(\boldsymbol{u}_i^{(n)},\ \boldsymbol{\mu},\ t_{n-1} + c_i \Delta t_n\right). \end{aligned} \tag{D.87}$$

Strict enforcement of the time-periodic partial differential equations leads to an application of the implicit function theorem, similar to that above, i.e., $\boldsymbol{u}^{(n)} = \boldsymbol{u}^{(n)}(\boldsymbol{\mu})$ and $\boldsymbol{k}_i^{(n)} = \boldsymbol{k}_i^{(n)}(\boldsymbol{\mu})$, where $\boldsymbol{u}^{(n)}$ and $\boldsymbol{k}_i^{(n)}$ are the time-periodic solution of the discrete partial differential equations. This results in an optimization problem identical to that in (D.86) with this new definition of $\boldsymbol{u}^{(n)}(\boldsymbol{\mu})$ and $\boldsymbol{k}_i^{(n)}(\boldsymbol{\mu})$. The novel periodic adjoint method, derived in Section D.4.1, can be used to compute gradients along the manifold of time-periodic solutions of the fully discrete PDE, i.e. $\dfrac{\mathrm{d}F}{\mathrm{d}\boldsymbol{\mu}}$ and $\dfrac{\mathrm{d}\boldsymbol{c}}{\mathrm{d}\boldsymbol{\mu}}$, for the use in gradient-based optimization.

D.4.4 Numerical Experiment: Time-Periodic Solutions of the Compressible Navier-Stokes Equations

In this section, the various solvers discussed in this document for determining primal and dual time-periodic solutions of partial differential equations are compared for a flapping airfoil in an isentropic, viscous flow. The stability of the periodic orbit is verified by performing an eigenvalue analysis of $\dfrac{\partial \boldsymbol{u}^{(N_t)}}{\partial \boldsymbol{u}_0}$. The section closes with validation of the adjoint method, introduced for efficient gradient computation of quantities of interest, against a second-order finite difference approximation.

Consider the NACA0012 airfoil in Figure D.27 immersed in an isentropic, viscous flow with Reynolds and Mach number set to 1000 and 0.2, respectively. The kinematic motion of the foil is

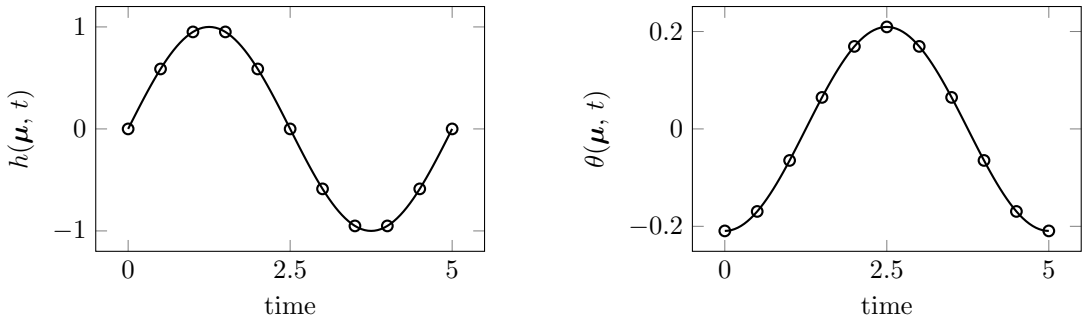


Figure D.17: Trajectories of $h(\boldsymbol{\mu}, t)$ and $\theta(\boldsymbol{\mu}, t)$ that define the motion of the airfoil in Figure D.27 and will be used to study primal and dual time-periodic solvers.

parametrized with a single Fourier mode, i.e.,

$$\begin{aligned} h(\boldsymbol{\mu}, t) &= A_h \sin(\omega_h t + \phi_h) + c_h \\ \theta(\boldsymbol{\mu}, t) &= A_\theta \sin(\omega_\theta t + \phi_\theta) + c_\theta. \end{aligned} \tag{D.88}$$

The vector of parameters is fixed for the remainder of this section

$$\boldsymbol{\mu} = \begin{bmatrix} A_h & \omega_h & \phi_h & c_h & A_\theta & \omega_\theta & \phi_\theta & c_\theta \end{bmatrix} = \begin{bmatrix} 1.0 & 0.4\pi & 0.0 & 0.0 & \frac{\pi}{15} & 0.4\pi & \frac{\pi}{2} & 0.0 \end{bmatrix}, \tag{D.89}$$

and corresponds to the motion in Figure D.17 with period $T = 5$. The mapping $\mathcal{G}(\boldsymbol{X}, t)$ from the fixed reference domain V to the physical domain $\Omega(\boldsymbol{\mu}, t)$ takes the form of a parametrized rigid body motion

$$\mathcal{G}(\boldsymbol{X}, t) = \boldsymbol{v}(\boldsymbol{\mu}, t) + \boldsymbol{Q}(\boldsymbol{\mu}, t)(\boldsymbol{X} - \boldsymbol{x}_0) + \boldsymbol{x}_0, \tag{D.90}$$

where $\boldsymbol{x}_0$ is the location of pitching axis in the reference configuration (the 1/3 chord) and

$$\boldsymbol{Q}(\boldsymbol{\mu}, t) = \begin{bmatrix} \cos\theta(\boldsymbol{\mu}, t) & \sin\theta(\boldsymbol{\mu}, t) \\ -\sin\theta(\boldsymbol{\mu}, t) & \cos\theta(\boldsymbol{\mu}, t) \end{bmatrix} \qquad \boldsymbol{v}(\boldsymbol{\mu}, t) = \begin{bmatrix} 0 \\ h(\boldsymbol{\mu}, t) \end{bmatrix}.$$

The isentropic Navier-Stokes equations are discretized with the discontinuous Galerkin scheme of Section D.1.2 using 978 triangular $p = 3$ elements. No-slip boundary conditions are imposed on the airfoil wall and characteristic free-stream boundary conditions at the far-field. The temporal discretization uses a third-order diagonally implicit Runge-Kutta solver with 100 equally spaced steps to discretize a single period of the motion. The airfoil and surrounding fluid vorticity field are shown in Figures D.18 and D.19 with the flow field initialized from steady-state flow and the time-periodic initial condition, respectively. It is clear that the flow in Figure D.19 will seamlessly transition between periods. The initialization from the steady-state solution in Figure D.18 will introduce non-physcial transients into the flow as discussed in the next section.

First, the solvers introduced in Section D.3.1 are compared for different initial guesses for the time-periodic initial condition. In the absence of any a priori information regarding the time-periodic

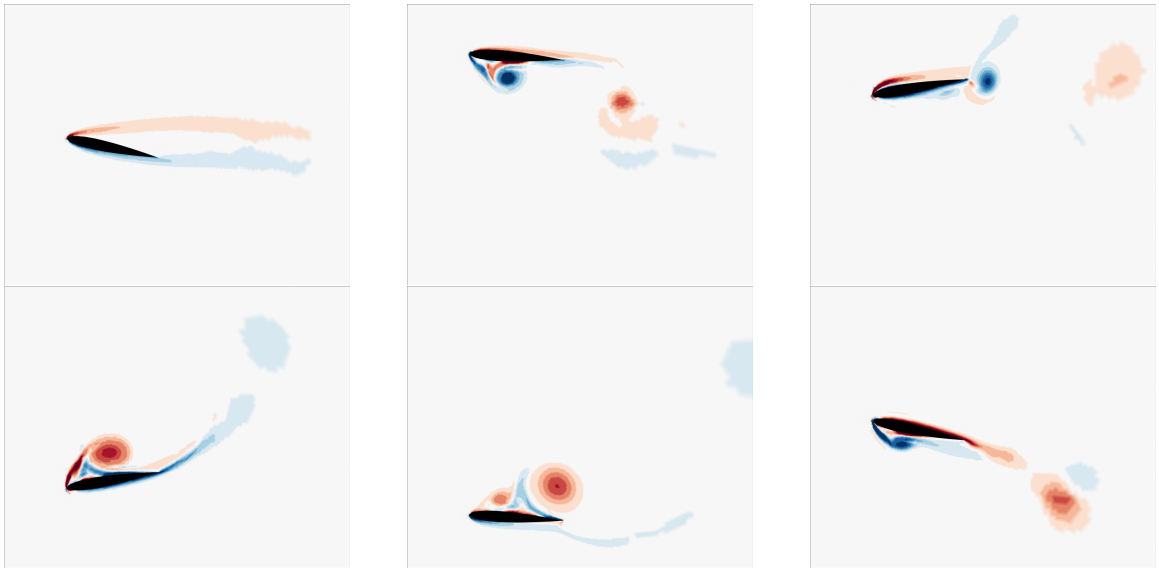


Figure D.18: Flow vorticity around heaving/pitching airfoil for simulation initialized from steady state flow. Non-physical transients are introduced at the beginning of the time interval that result in non-trivial errors in integrated quantities of interests. Snapshots taken at times $t = 0.0,\ 1.0,\ 2.0,\ 3.0,\ 4.0,\ 5.0$.

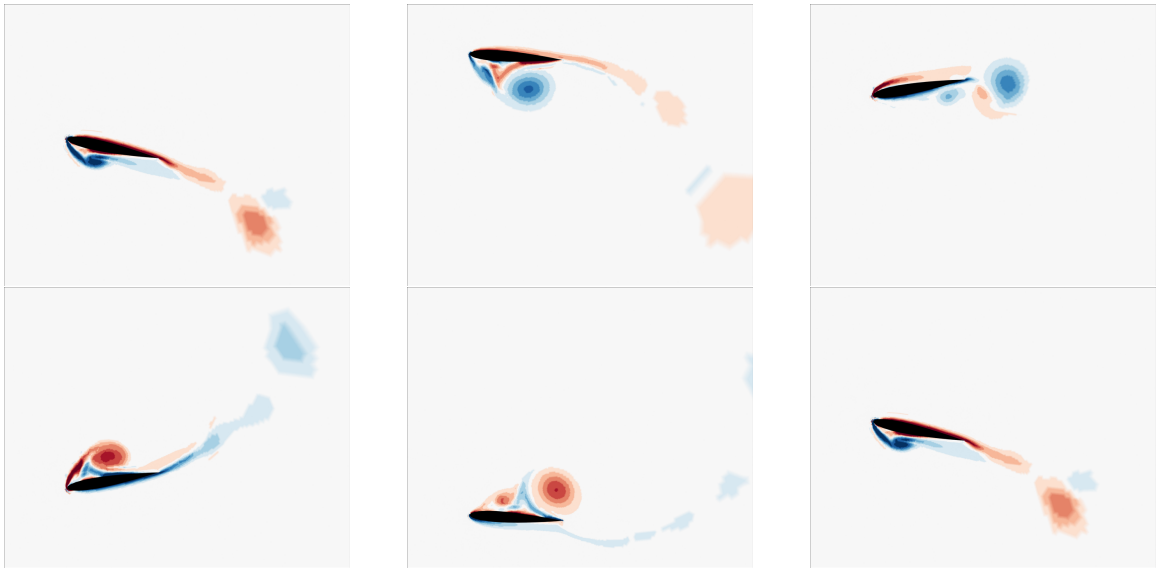


Figure D.19: Time-periodic flow vorticity around heaving/pitching airfoil, i.e., initialized from periodic initial condition. The time-periodic initial condition ensures transients are not introduced at the beginning of the simulation; the result is a seamless transition between periods, as would be experienced in-flight, and trusted integrated quantities of interest. Snapshots taken at times $t = 0.0,\ 1.0,\ 2.0,\ 3.0,\ 4.0,\ 5.0$.

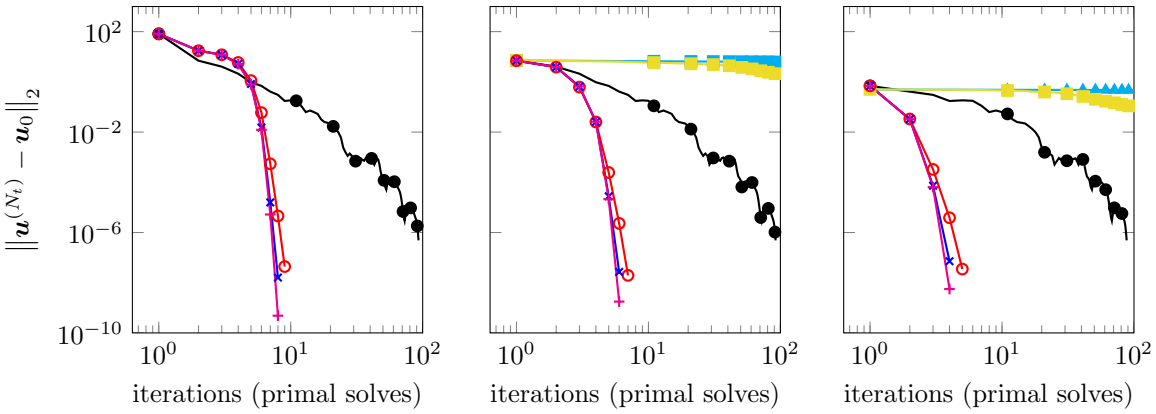


Figure D.20: Convergence comparison for numerical solvers for fully discrete time-periodically constrained partial differential equations (D.52), (D.54), nonlinearly preconditioned with m fixed point iterations. Left: $m = 0$, middle: $m = 1$, right: $m = 5$. Solvers: fixed point iteration (—●—), steepest decent (—▲—), L-BFGS (—■—), Newton-GMRES: $\Delta = 10^{-2}$ (—○—), $\Delta = 10^{-3}$ (—×—), $\Delta = 10^{-4}$ (—+—), where Δ is the GMRES convergence tolerance. The optimization algorithms (steepest decent and L-BFGS) were not included in the $m = 0$ study due to lack of convergence issues.

solution, a reasonable initial guess is the steady-state flow. Since the problem under consideration is being forced by an input—the periodic motion of the foil—a mechanism for improving the initial guess is to simulate the flow field for m periods of the foil motion and use the final state of the final period as the initial guess. This corresponds to using m iterations of fixed point iteration (Algorithm 21) as a nonlinear preconditioner for the nonlinear system of equations (D.56) that enforces time-periodicity of the flow.

Figure D.20 and Table D.6 compare the solvers under consideration for different levels of nonlinear preconditioning. Regardless of nonlinear preconditioning, the Newton-GMRES solver converges most rapidly for a range of linear system tolerances from 10^{-2} to 10^{-4} and the optimization algorithms (steepest decent and L-BFGS) converge most slowly. In fact, without any nonlinear preconditioning the optimization algorithms fail to make progress toward the optimal solution and were not included in the figure. Nonlinear preconditioning helps the Newton-GMRES algorithm most substantially, particularly with $m = 5$, as this appears to place the initial guess close enough to the solution that quadratic convergence is obtained from the outset. This causes the number of Newton iterations to be reduced from 8 or 9 to 3 or 4. From Table D.6, this does not save many primal solvers—since the nonlinear preconditioning requires primal solves—but requires far fewer linear system solves and therefore fewer sensitivity solutions. Figure D.21 isolates the Newton-GMRES solver (for $m = 0$, i.e., the case without preconditioning) to highlight convergence rates for different GMRES tolerances. It also shows the convergence of GMRES for each nonlinear iteration and each tolerance considered. As expected, more GMRES iterations are required near convergence as it becomes more difficult to reduce the linear residual the prescribed orders of magnitude.

Table D.6: Table summarizing performance of numerical solvers for fully discrete time-periodic partial differential equations, considering nonlinear preconditioning via m fixed point iterations.

$m = 0$	$\left\|\boldsymbol{u}^{(N_t)} - \boldsymbol{u}_0\right\|_2$	Primal Solves (D.55)	Sensitivity Solves (D.63)	Adjoint Solves (D.25)
Fixed Point Iteration	8.10e-07	90	0	0
Newton-Krylov (10^{-2})	4.41e-08	9	128	0
Newton-Krylov (10^{-3})	1.60e-08	8	156	0
Newton-Krylov (10^{-4})	4.85e-10	8	220	0
$m = 1$	$\left\|\boldsymbol{u}^{(N_t)} - \boldsymbol{u}_0\right\|_2$	Primal Solves (D.55)	Sensitivity Solves (D.63)	Adjoint Solves (D.25)
Fixed Point Iteration	8.10e-07	90	0	0
Steepest Decent	6.09e+00	121	0	121
L-BFGS	1.36e+00	121	0	121
Newton-Krylov (10^{-2})	1.96e-08	8	104	0
Newton-Krylov (10^{-3})	2.69e-08	7	116	0
Newton-Krylov (10^{-4})	1.77e-09	7	149	0
$m = 5$	$\left\|\boldsymbol{u}^{(N_t)} - \boldsymbol{u}_0\right\|_2$	Primal Solves (D.55)	Sensitivity Solves (D.63)	Adjoint Solves (D.25)
Fixed Point Iteration	8.10e-07	90	0	0
Steepest Decent	4.65e-01	125	0	125
L-BFGS	7.40e-02	125	0	125
Newton-Krylov (10^{-2})	3.50e-08	10	92	0
Newton-Krylov (10^{-3})	7.18e-08	9	88	0
Newton-Krylov (10^{-4})	5.61e-09	9	121	0

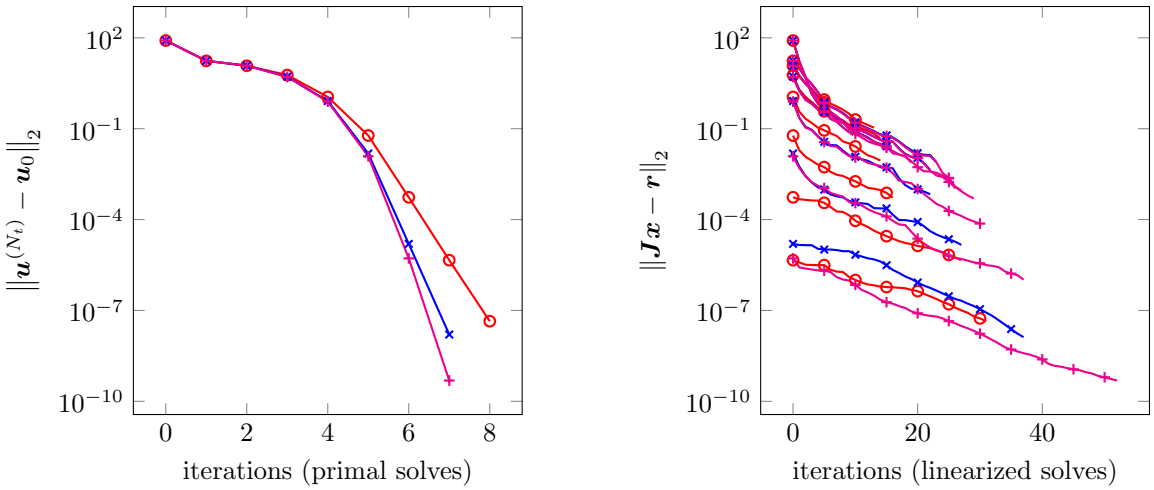


Figure D.21: Linear and nonlinear convergence of Newton-GMRES method for determining fully discrete time-periodic solutions with various linear system tolerances, Δ, i.e., $\|\boldsymbol{J}\boldsymbol{x} - \boldsymbol{R}\| < \Delta$, where $\boldsymbol{r}$ and $\boldsymbol{J}$ are defined in (D.61) and (D.62). Tolerances considered: $\Delta = 10^{-2}$ (—o—), $\Delta = 10^{-3}$ (—x—), $\Delta = 10^{-4}$ (—+—).

The time history of the instantaneous quantities of interest in Figure D.22 illustrate the non-physical transients that result from initializing the flow with the steady-state solution. While the transients mostly vanish after a single Newton iteration, the trajectories of these quantities of interest do not coincide with those of the true time-periodic solution. The error between the integrated quantities of interest—W and J_x—at the time-periodic flow versus intermediate iterations is shown in Figure D.23. Comparing Figures D.20 and D.23, it can be seen that a tolerance of 10^{-8} on $\left\|\boldsymbol{u}^{(N_t)} - \boldsymbol{u}^{(0)}\right\|_2$ leads to an accuracy of 10^{-6} in the integrated quantities of the time-periodic solution.

Next, the stability of the periodic orbit is verified by considering the eigenvalues of $\dfrac{\partial \boldsymbol{u}^{(N_t)}}{\partial \boldsymbol{u}_0}$,

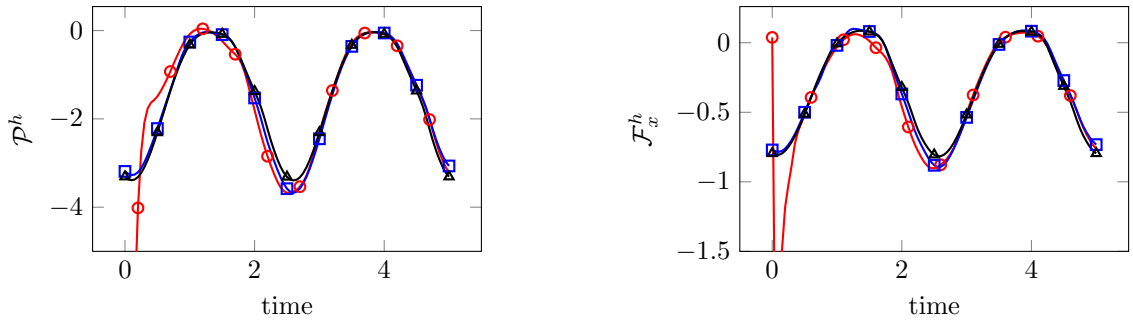


Figure D.22: Time history of power, $\mathcal{F}_x^h(\boldsymbol{u}, \boldsymbol{\mu}, t)$, and x-directed force, $\mathcal{P}^h(\boldsymbol{u}, \boldsymbol{\mu}, t)$, after k Newton-GMRES iterations (linear system convergence tolerance $\Delta = 10^{-2}$) starting from steady-state. Values of k: 0 (—o—), 1 (—□—), and 8 (—△—).

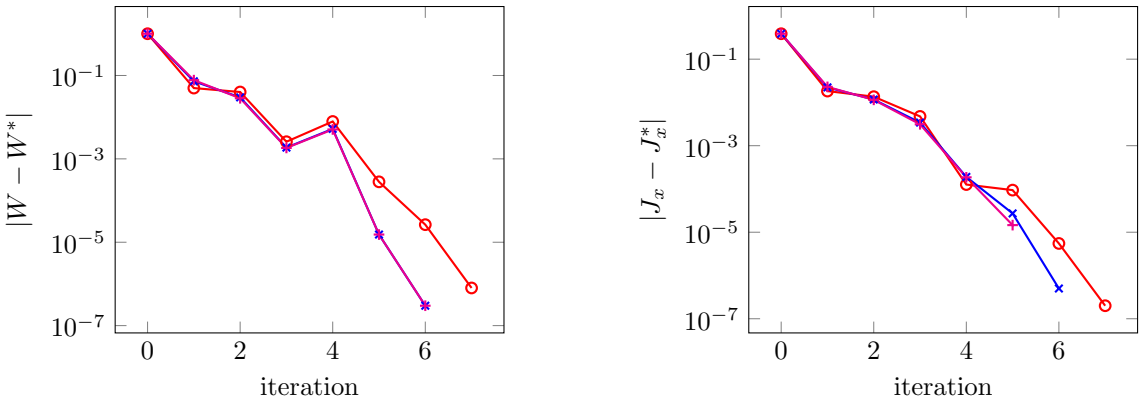


Figure D.23: Convergence of fully discrete quantities of interest to their values at the time-periodic solution, W^* and J_x^*, for various solvers, without nonlinear preconditioning. Solvers: Newton-GMRES: $\Delta = 10^{-2}$ (—o—), $\Delta = 10^{-3}$ (—x—), $\Delta = 10^{-4}$ (—+—), where Δ is the GMRES convergence tolerance.

evaluated at the time-periodic solution. As discussed in Section D.3.2 and many prior works [47, 112], the periodic orbit is stable if all eigenvalues of this matrix have modulus less than unity. Figure D.24 shows that the 200 eigenvalues of largest modulus lie within the unit circle in the complex plane; thus, the periodic orbit is stable for this problem.

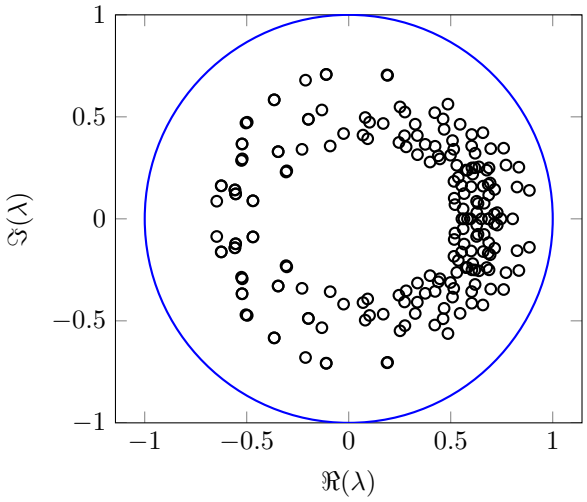


Figure D.24: First 200 eigenvalues (o) of $\frac{\partial \boldsymbol{u}^{(N_t)}}{\partial \boldsymbol{u}_0}$—evaluated at periodic solution—with largest magnitude. All eigenvalues lie in unit circle, thus the periodic orbit is stable.

This completes the discussion of the primal time-periodic problem and attention is turned to the dual, or adjoint, problem. First, a brief comparison of two potential solvers—fixed point iteration and GMRES—for the periodic adjoint equation is provided. In contrast to the primal problem, there is a less pronounced difference between the convergence of fixed point iteration and the Krylov solver

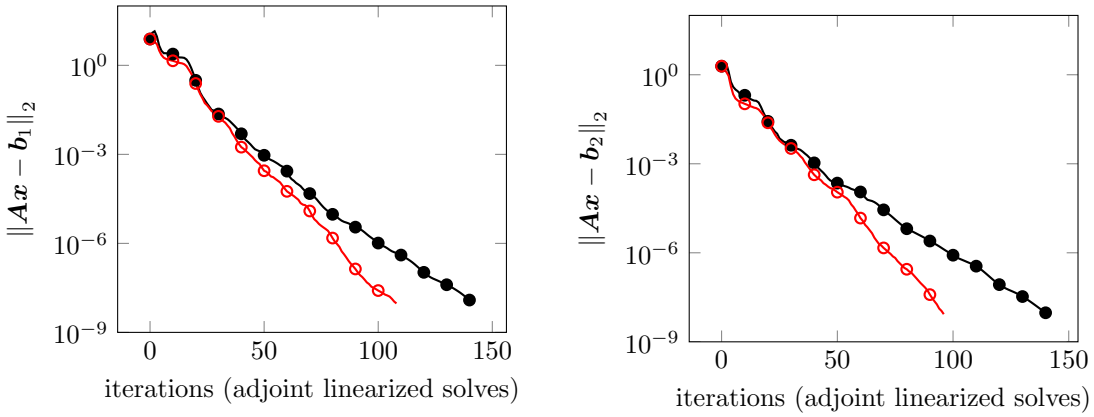


Figure D.25: GMRES convergence for determining solution of adjoint equations corresponding to fully discrete time-periodic partial differential equation, i.e., a linear two-point boundary value problem. $\boldsymbol{A}$ defined in (D.81), $\boldsymbol{b}_1 = \dfrac{\partial W}{\partial \boldsymbol{u}^{(N_t)}}$, and $\boldsymbol{b}_2 = \dfrac{\partial J_x}{\partial \boldsymbol{u}^{(N_t)}}$ from (D.80), where W is fully discrete approximation of the total work done by fluid on airfoil and J_x is the x-directed impulse. Solvers: fixed point iteration (—●—) and GMRES (—○—). The linearization is performed about the time-periodic solution obtained with Newton-Krylov ($\Delta = 10^{-4}$) method.

in the dual problem. Figure D.25 shows the convergence history for two different right-hand sides of $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$, each corresponding to the adjoint method for a different quantity of interest. However, it should be noted that the iterations for the GMRES solver are cheaper than those of the fixed point solver as the terms $\dfrac{\partial F}{\partial \boldsymbol{\mu}}$—which may be expensive if $\boldsymbol{\mu}$ is a large vector—are not computed. Therefore, the GMRES algorithm is superior to fixed point iterations as there are fewer required iterations, each of which is cheaper.

Finally, the adjoint method for computing gradients of quantities of interest on the manifold of time-periodic solutions of the partial differential equations is verified against a second-order finite difference approximations. The finite difference approximation to gradients on the aforementioned manifold requires finding the *time-periodic* solution of the governing equations *at perturbations* about the nominal parameter configuration in (D.89). Figure D.26 shows the relative error between the gradients computed via the adjoint method in Algorithm 24 to this finite difference approximation for a sweep of finite difference intervals, τ. To realize the sub-10^{-6} finite difference errors in the time-periodic gradient, tolerances of 10^{-12} were used for the primal and dual time-periodic solutions. As expected, the error starts to increase after τ drops too small due to the trade-off between finite difference accuracy and round-off error.

Given this exposition on solvers for time-periodically constrained partial differential equations, we turn our attention to deriving the corresponding fully discrete adjoint equations.

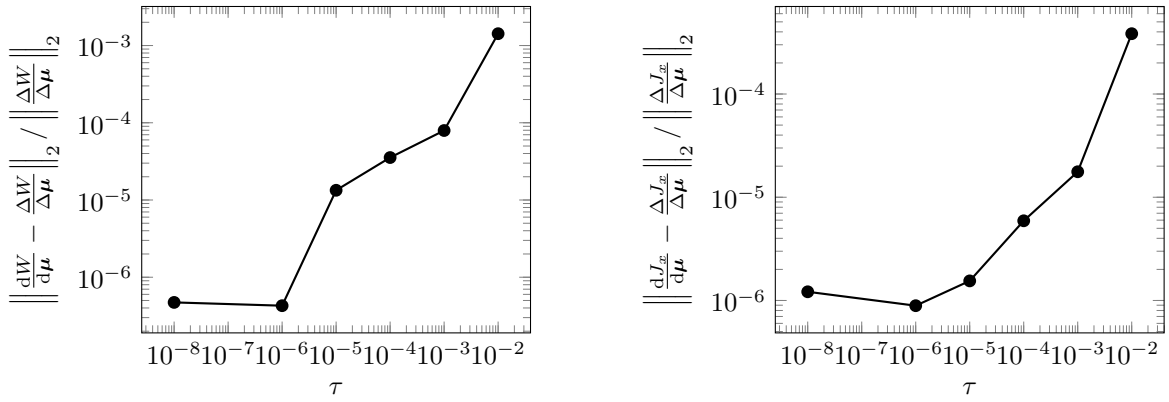


Figure D.26: Verification of periodic adjoint-based gradient with second-order centered finite difference approximation, for a range of finite intervals, τ. The computed gradient match the finite difference approximation to nearly 7 digits before round-off errors degrade the accuracy.

Table D.7: Comparison of non-zero derivatives of total energy, W, and x-impulse, J_x, computed with the adjoint method and a second-order finite difference approximation with step size $\tau = 10^{-6}$.

	$\frac{\partial W}{\partial A_h}$	$\frac{\partial W}{\partial \omega_h}$	$\frac{\partial W}{\partial A_\theta}$	$\frac{\partial W}{\partial c_\theta}$
Adjoint	-2.30919016e+01	-2.593579090e+01	-7.99568107e+00	5.881595017e-01
Finite difference	-2.30919013e+01	-2.593579395e+01	-7.99568151e+00	5.881594917e-01
	$\frac{\partial J_x}{\partial A_h}$	$\frac{\partial J_x}{\partial \omega_h}$	$\frac{\partial J_x}{\partial A_\theta}$	$\frac{\partial J_x}{\partial c_\theta}$
Adjoint	-1.85436790e-01	-1.029830753e-01	6.72970822e+00	1.270106907e-02
Finite difference	-1.85436774e-01	-1.029834126e-01	6.72970891e+00	1.270112956e-02

D.4.5 Numerical Experiment: Energetically Optimal Flapping with Thrust and Time-Periodicity Constraints

This section will apply the novel, fully discrete, periodic adjoint method to solve an optimal control problem governed by the time-periodically constrained isentropic compressible Naiver-Stokes equations. The system of PDEs is discretized using a nodal discontinuous Galerkin (DG) method on unstructured meshes of triangles, with polynomial degrees 3 within each element. The viscous fluxes are chosen according to the compact DG method [150] method, and our implementation is fully implicit with exact Jacobian matrices and a range of parallel iterative solvers [153]. The resulting semi-discrete system has the form of our general system of ODEs (D.13). All partial derivatives of the semi-discrete governing equations and corresponding quantities of interest, namely $\frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}$, $\frac{\partial \boldsymbol{r}}{\partial \boldsymbol{\mu}}$, $\frac{\partial f_h}{\partial \boldsymbol{u}}$, $\frac{\partial f_h}{\partial \boldsymbol{\mu}}$ are computed via automatic symbolic differentiation at the element-level with the MAPLE software [126] and subsequent assembly. The semi-discrete quantity of interest f_h is defined as the approximation of $\int_{\boldsymbol{\Gamma}(\boldsymbol{\mu},\, t)} f(\boldsymbol{U},\, \boldsymbol{\mu},\, t)\, dS$ in (D.51) using the DG shape functions and required, along with the temporal discretization scheme, to compute the discrete output functional F in (D.53). Additional details regarding computation of the partial derivatives with respect to $\boldsymbol{\mu}$ in the case of a parametrized, deforming domain are provided in Section D.2.4 and [211].

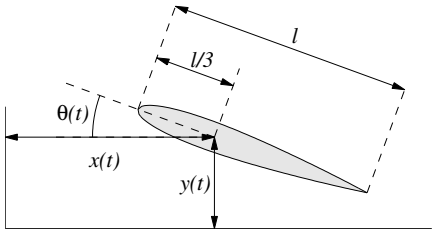


Figure D.27: Kinematic description of body under consideration, NACA0012 airfoil (right).

The remainder of this section will consider the time-periodic solution and optimization of a flapping NACA0012 airfoil, shown in Figure D.27. Two quantities of interest that will be considered are the total work exerted by the fluid on the airfoil, $\mathcal{W}$, and the impulse in the x-direction imparted on the airfoil by the fluid, $\mathcal{J}_x$, which take the form

$$\mathcal{W}(\boldsymbol{U}, \boldsymbol{\mu}) = \int_0^T \int_{\boldsymbol{\Gamma}} \boldsymbol{f}(\boldsymbol{U}, \boldsymbol{\mu}, t) \cdot \dot{\boldsymbol{x}}\, dS\, dt \qquad \text{and} \qquad \mathcal{J}_x(\boldsymbol{U}, \boldsymbol{\mu}) = \int_0^T \int_{\boldsymbol{\Gamma}} \boldsymbol{f}(\boldsymbol{U}, \boldsymbol{\mu}, t) \cdot \mathbf{e}_1\, dS\, dt \quad \text{(D.91)}$$

In this case, $\boldsymbol{\Gamma}$ is the surface of the airfoil, $\boldsymbol{e}_1 \in \mathbb{R}^{n_{sd}}$ is the 1st canonical unit vector, $\boldsymbol{f}(\boldsymbol{U}, \boldsymbol{\mu}, t) \in \mathbb{R}^{n_{sd}}$ is the instantaneous force that the fluid exerts on the airfoil, and $\dot{\boldsymbol{x}}$ is the pointwise velocity of airfoil. The solver-consistent discretization, discussed in Section 2.1.4 and [211], of these quantities results in the fully discrete approximations $W(\boldsymbol{u}^{(0)}, \ldots, \boldsymbol{u}^{(N_t)}, \boldsymbol{k}_1^{(1)}, \ldots, \boldsymbol{k}_s^{(N_t)}, \boldsymbol{\mu})$ and $J_x(\boldsymbol{u}^{(0)}, \ldots, \boldsymbol{u}^{(N_t)}, \boldsymbol{k}_1^{(1)}, \ldots, \boldsymbol{k}_s^{(N_t)}, \boldsymbol{\mu})$.

The *instantaneous* quantities of interest corresponding to those in (D.91) are the power and x-directed force the fluid exerts on the airfoil, which take the form

$$\mathcal{P}(\boldsymbol{U},\boldsymbol{\mu},t) = \int_{\boldsymbol{\Gamma}} \boldsymbol{f}(\boldsymbol{U},\boldsymbol{\mu},t)\cdot\dot{\boldsymbol{x}}\; dS \qquad \text{and} \qquad \mathcal{F}_x(\boldsymbol{U},\boldsymbol{\mu},t) = \int_{\boldsymbol{\Gamma}} \boldsymbol{f}(\boldsymbol{U},\boldsymbol{\mu},t)\cdot\mathbf{e}_1\; dS.$$

Define $\mathcal{P}^h(\boldsymbol{u},\boldsymbol{\mu},t)$ and $\mathcal{F}_x^h(\boldsymbol{u},\boldsymbol{\mu},t)$ as the solver-consistent semi-discretization of these instantaneous quantities of interest.

In this section, the periodic adjoint method is used to solve an optimal control problem with *time-periodicity constraints* using gradient-based optimization techniques. The optimization problem is to determine the *energetically optimal* flapping motion of the NACA0012 airfoil in isentropic, viscous flow—over a single *representative*, in-flight period—such that the x-directed impulse on the body is identically 0. The continuous form of the optimal control problem is given as

$$\begin{aligned}
\underset{\boldsymbol{U},\;\boldsymbol{\mu}}{\text{minimize}} \quad & \mathcal{W}(\boldsymbol{U},\boldsymbol{\mu}) \\
\text{subject to} \quad & \mathcal{J}_x(\boldsymbol{U},\boldsymbol{\mu}) = 0 \\
& \boldsymbol{U}(\boldsymbol{x},\;0) = \boldsymbol{U}(\boldsymbol{x},\;T) \\
& \frac{\partial \boldsymbol{U}}{\partial t} + \nabla\cdot\boldsymbol{F}(\boldsymbol{U},\;\nabla\boldsymbol{U}) = 0 \quad \text{in } \Omega(\boldsymbol{\mu},t).
\end{aligned} \tag{D.92}$$

After spatial and temporal discretization via the high-order discontinuous Galerkin and diagonally implicit Runge-Kutta schemes in Section 2.1.3, the continuous optimization problem in (D.92) is replaced with its fully discrete counterpart

$$\begin{aligned}
\underset{\substack{\boldsymbol{u}^{(0)},\;\ldots,\;\boldsymbol{u}^{(N_t)}\in\mathbb{R}^{N_{\boldsymbol{u}}},\\ \boldsymbol{k}_1^{(1)},\;\ldots,\;\boldsymbol{k}_s^{(N_t)}\in\mathbb{R}^{N_{\boldsymbol{u}}},\\ \boldsymbol{\mu}\in\mathbb{R}^{N_{\boldsymbol{\mu}}}}}{\text{minimize}} \quad & W(\boldsymbol{u}^{(0)},\;\ldots,\;\boldsymbol{u}^{(N_t)},\;\boldsymbol{k}_1^{(1)},\;\ldots,\;\boldsymbol{k}_s^{(N_t)},\;\boldsymbol{\mu}) \\
\text{subject to} \quad & J_x(\boldsymbol{u}^{(0)},\;\ldots,\;\boldsymbol{u}^{(N_t)},\;\boldsymbol{k}_1^{(1)},\;\ldots,\;\boldsymbol{k}_s^{(N_t)},\;\boldsymbol{\mu}) = 0 \\
& \boldsymbol{u}^{(0)} = \boldsymbol{u}^{(N_t)} \\
& \boldsymbol{u}^{(n)} = \boldsymbol{u}^{(n-1)} + \sum_{i=1}^{s} b_i \boldsymbol{k}_i^{(n)} \\
& \boldsymbol{M}\boldsymbol{k}_i^{(n)} = \Delta t_n \boldsymbol{r}\left(\boldsymbol{u}_i^{(n)},\;\boldsymbol{\mu},\;t_{n-1}+c_i\Delta t_n\right).
\end{aligned} \tag{D.93}$$

The physical and numerical setup are identical to that in Section D.4.4 with the exception of the kinematic parametrization. Instead of a single Fourier mode, the kinematic motion is parametrized by cubic splines with 5 equally spaced knots and boundary conditions that enforce

$$\begin{aligned}
h(\boldsymbol{\mu},\,t) &= -h(\boldsymbol{\mu},\,t+T/2) \\
\theta(\boldsymbol{\mu},\,t) &= -\theta(\boldsymbol{\mu},\,t+T/2)
\end{aligned} \tag{D.94}$$

where t is time and $T = 5$ is the fixed period of the flapping motion. The vector of parameters,

$\boldsymbol{\mu}$—used as optimization parameters—are the *knots* of the cubic splines. This leads to $N_{\boldsymbol{\mu}} = 8$ parameters; 4 knots for the motion of $h(\boldsymbol{\mu}, t)$ and $\theta(\boldsymbol{\mu}, t)$[2]. Notice that (D.94) enforces the trajectories of $h(\boldsymbol{\mu}, t)$ and $\theta(\boldsymbol{\mu}, t)$ in $[T/2, T]$ to be the mirror of those in $[0, T/2]$, which implicitly enforces periodicity with period T. The mapping $\mathcal{G}$ from the reference to physical domain required for the DG-ALE formulation is defined in (D.90) with the new definition of $h(\boldsymbol{\mu}, t)$ and $\theta(\boldsymbol{\mu}, t)$ with periodic cubic splines.

The optimization problem in (D.93) is solved using the extension of the nested framework for PDE-constrained optimization, or generalized reduced-gradient method, introduced in Section D.4.3. The solvers introduced in Section D.3.1 will be used to determine the time-periodic flow around the airfoil. Given the results in the previous section, the Newton-GMRES method with a tolerance of $\Delta = 10^{-3}$, warm-started from $m = 5$ fixed-point iterations is employed. The flow is deemed to be periodic if

$$\left\| \boldsymbol{u}^{(0)} - \boldsymbol{u}^{(N_t)} \right\|_2 \leq 10^{-10}. \tag{D.95}$$

The periodic flow is used to compute quantities of interest—the total work and x-impulse. Then, the periodic adjoint method will be used to compute gradients of the quantities of interest along the manifold of *time-periodic* solutions of the governing equation. GMRES is used to solve the dual linear, periodic adjoint equations with a tolerance of $\Delta = 10^{-4}$. Since there are two quantities of interest, two periodic adjoint solves must be performed at each optimization iteration. Finally, the quantities of interest and their gradients are passed to an optimization solver—SNOPT [70] is used in this work—and progress is made toward a local minimum.

The initial condition for the optimization solver is shown in Figure D.28; the heaving motion is a sinusoid with amplitude 1 and there is no pitch—pure heaving motion. The vorticity snapshots in Figure D.31 show this motion induces a fairly violent flow with shedding vortices. The corresponding time history of the power, $\mathcal{P}^h(\boldsymbol{u}, \boldsymbol{\mu}, t)$, and x-directed force, $\mathcal{F}_x^h(\boldsymbol{u}, \boldsymbol{\mu}, t)$, imparted onto the airfoil by the fluid are shown in Figure D.29. After 16 periodic optimization iterations, the first-order optimality conditions have been reduced by two orders of magnitude. From Figure D.28, the optimal airfoil motion is a combination of heaving and pitching. From the initial guess, the amplitude of the heaving motion has been reduced by more than a factor of two and the pitching amplitude increased to 18.7°. The convergence history for the optimization solver is given in Figure D.30. At the optimal solution, the total work required to perform the flapping motion is more than an order of magnitude smaller than at the initial guess (pure heaving). Figures D.31 and D.32 show snapshots of the flow in time at the initial, purely heaving motion and the optimal flapping motion. From these figures, it is clear that the flow corresponding to the optimal motion is relatively benign with no shedding vortices, which explains the reduction in required work. The efficiency of combined pitching and heaving has been repeatedly observed experimentally [191, 162, 158] and the phase angle of approximately 90° between pitching and heaving, as observed in Figure D.28, has also been observed in experiments [191, 162, 158, 148]. The specific pitching and heaving amplitudes were

[2]There are only 4 degrees of freedom since the mirror boundary condition in (D.94) prescribes the value of one of the knots given the other four.

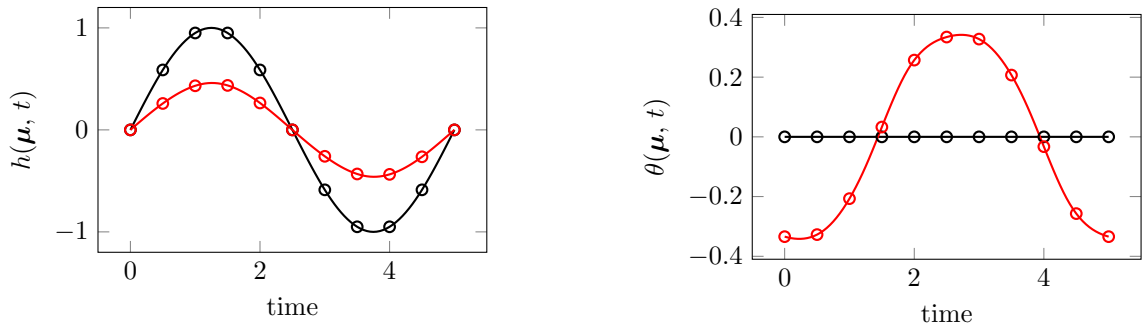


Figure D.28: Trajectories of $h(\boldsymbol{\mu}, t)$ and $\theta(\boldsymbol{\mu}, t)$ at initial guess (—o—) and optimal solution (—o—) for optimization problem in (D.93).

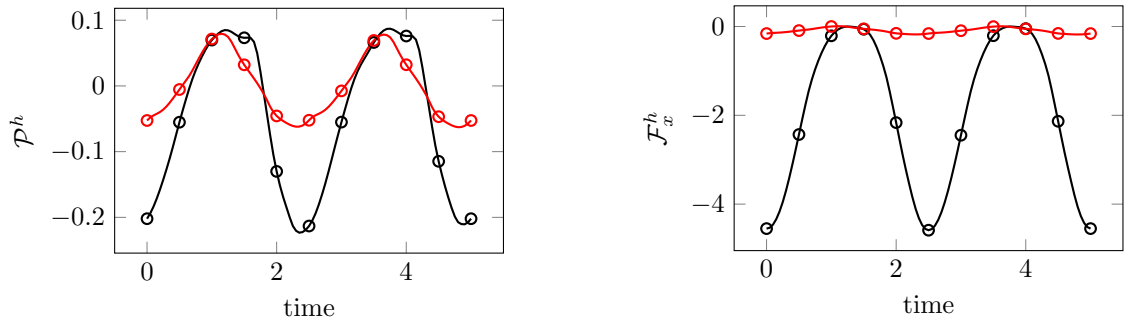


Figure D.29: Time history of the power, $\mathcal{P}^h(\boldsymbol{u}, \boldsymbol{\mu}, t)$, and x-directed force, $\mathcal{F}_x^h(\boldsymbol{u}, \boldsymbol{\mu}, t)$, imparted onto foil by fluid at initial guess (—o—) and optimal solution (—o—) for optimization problem in (D.93).

determined by the optimizer such that the thrust constraint is satisfied; if the thrust requirement was increased, these magnitudes would increase and result in a more violent flow field, eventually leading to vortex shedding [191, 211].

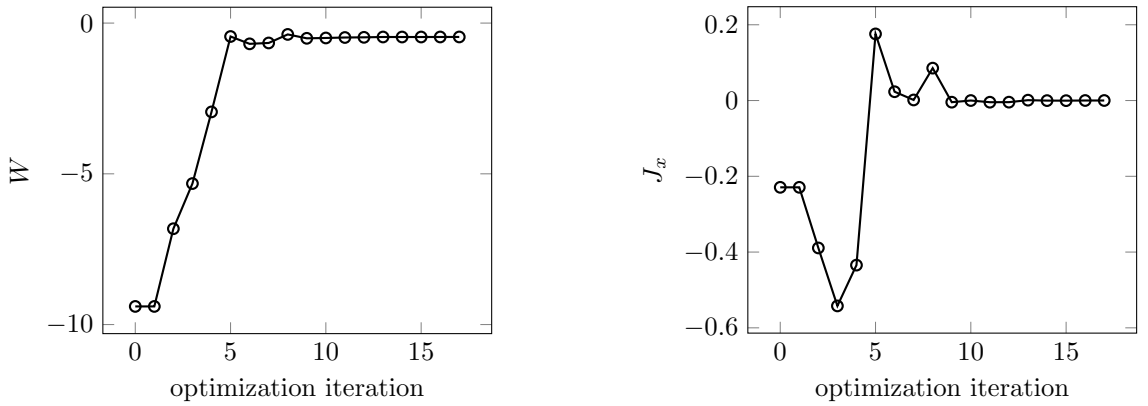


Figure D.30: Convergence of quantities of interest, W and J_x, with optimization iteration. Each optimization iteration requires a periodic flow computation and its corresponding adjoint to evaluate the quantities of interest and their gradients.

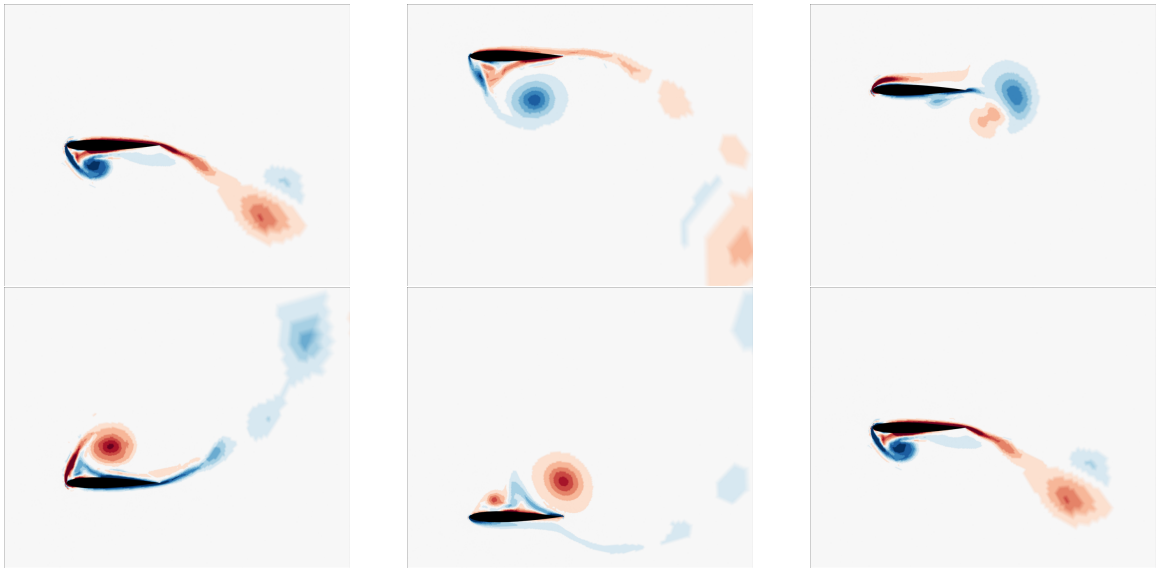


Figure D.31: Trajectory of airfoil and flow vorticity at initial guess for optimization (pure heaving motion, see Figure D.28). Snapshots taken at times $t = 0.0,\ 1.0,\ 2.0,\ 3.0,\ 4.0,\ 5.0$.

Figure D.32: Trajectory of airfoil and flow vorticity at energetically optimal, zero-impulse flapping motion (see Figure D.28). Snapshots taken at times $t = 0.0,\ 1.0,\ 2.0,\ 3.0,\ 4.0,\ 5.0$.

D.5 Conclusion

This appendix discussed a fully discrete framework for computing time-periodic solutions of partial differential equations. The discussion included the spatio-temporal discretization of the governing equations and a slew of time-periodic shooting solvers, including optimization-based and Newton-Krylov methods. These shooting methods consider the state at the final time to be a nonlinear function of the initial condition and solve $\boldsymbol{u}^{(N_t)}(\boldsymbol{u}_0) = \boldsymbol{u}_0$ using Newton-Raphson iterations or optimization techniques to minimize its norm. The linear system of equations, arising in the Newton-Raphson iterations, were solved using matrix-free GMRES with matrix-vector products computed as the solution of the linearized, sensitivity equations (with appropriate initial condition). The adjoint method was used to compute the gradients in the gradient-based optimization solvers. These periodic solvers were used to compute the time-periodic flow around a flapping airfoil in isentropic, compressible, viscous flow, and their performance compared. The Newton-Krylov solver exhibits superior convergence to the optimization-based shooting methods, even when inexact tolerances were used on the linear system solves, and fully leverages quality starting guesses. An eigenvalue analysis is provided to show the periodic orbit of the flapping problem is stable.

The main contribution of the document is the derivation of the adjoint equations corresponding to the fully discrete time-periodically constraint partial differential equations. As opposed to the backward-in-time evolution equations, these equations constitute a linear, *two-point boundary value problem* that is provably solvable. The corresponding adjoint method was introduced for computing *exact* gradients of quantities of interest along the manifold of time-periodic solutions of the discrete conservation law. The gradients were verified against a second-order finite difference approximation. These quantities of interest and their gradients were used in the context of gradient-based

optimization to solve an optimal control problem with time-periodicity constraints, among others. In particular, the energetically optimal flapping motion of a 2D airfoil in *time-periodic*, isentropic, compressible, viscous flow that generates a prescribed time-averaged thrust is sought. The proposed framework improves the nominal flapping motion by reducing the flapping energy nearly an order of magnitude and exactly satisfies the thrust constraint.

While this work is an initial step toward problems of engineering and scientific relevance, additional development will be required to solve truly impactful problems. One extension of this work is the development of robust solvers for determining *nearly* time-periodic solutions of problems where a time-periodic solution does not exist, but exhibits quasi-cyclic behavior. An example of such a problem is the 3D turbulent flow around periodically driven bodies such as helicopter and windmill blades. Another extension will be the development of faster numerical solvers to reduce the cost of computing time-periodic solutions or solving optimization problems with time-periodicity constraints. For example, economical, matrix-free preconditioners could result in non-trivial speedups for the Newton-Krylov time-periodicity solver and Krylov solver for the periodic adjoint equations. Model order reduction techniques could dramatically reduce the cost of computing the solution of the primal partial differential equations, and consequently the entire time-periodic solver.

D.6 Existence and Uniqueness of Solutions of the Adjoint Equations of the Fully Discrete, Time-Periodically Constrained Partial Differential Equations

This section proves existence and uniqueness of solutions of the adjoint equations of the fully discrete, time-periodically constrained partial differential equation. The strategy is to show the linear operator that encapsulates them is the transpose of the linear operator that defines the fully discrete, sensitivity equations, which is assumed non-singular at a time-periodic solution.

Consider the initial-value problem (D.55), with the initial condition parametrized by $\boldsymbol{\mu}$,

$$\begin{aligned} \boldsymbol{u}^{(0)} &= \boldsymbol{u}_0(\boldsymbol{\mu}) \\ \boldsymbol{u}^{(n)} &= \boldsymbol{u}^{(n-1)} + \sum_{i=1}^{s} b_i \boldsymbol{k}_i^{(n)} \\ \boldsymbol{M}\boldsymbol{k}_i^{(n)} &= \Delta t_n \boldsymbol{r}\left(\boldsymbol{u}_i^{(n)},\ \boldsymbol{\mu},\ t_{n-1} + c_i \Delta t_n\right). \end{aligned} \tag{D.96}$$

The fully discrete adjoint equations corresponding to the primal equation in (D.96) and the discrete

quantity of interest, $F(\boldsymbol{u}^{(0)},\dots,\boldsymbol{u}^{(N_t)},\boldsymbol{k}_1^{(1)},\dots,\boldsymbol{k}_s^{(N_t)},\boldsymbol{\mu})$ are

$$\begin{aligned}
\boldsymbol{\nu}^{(N_t)} &= \frac{\partial F}{\partial \boldsymbol{u}^{(N_t)}}^T \\
\boldsymbol{\nu}^{(n-1)} &= \boldsymbol{\nu}^{(n)} + \frac{\partial F}{\partial \boldsymbol{u}^{(n-1)}}^T + \sum_{i=1}^{s} \Delta t_n \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}\left(\boldsymbol{u}_i^{(n)},\ \boldsymbol{\mu},\ t_{n-1}+c_i\Delta t_n\right)^T \boldsymbol{\tau}_i^{(n)} \\
\boldsymbol{M}^T\boldsymbol{\tau}_i^{(n)} &= \frac{\partial F}{\partial \boldsymbol{k}_i^{(n)}}^T + b_i\boldsymbol{\nu}^{(n)} + \sum_{j=i}^{s} a_{ji}\Delta t_n \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}\left(\boldsymbol{u}_j^{(n)},\ \boldsymbol{\mu},\ t_{n-1}+c_j\Delta t_n\right)^T \boldsymbol{\tau}_j^{(n)},
\end{aligned} \tag{D.97}$$

and the gradient of the quantity of interest can be reconstructed as

$$\frac{\mathrm{d}F}{\mathrm{d}\boldsymbol{\mu}} = \frac{\partial F}{\partial \boldsymbol{\mu}} + \boldsymbol{\nu}^{(0)^T}\frac{\partial \boldsymbol{u}_0}{\partial \boldsymbol{\mu}} + \sum_{n=1}^{N_t}\Delta t_n \sum_{i=1}^{s} \boldsymbol{\tau}_i^{(n)^T}\frac{\partial \boldsymbol{r}}{\partial \boldsymbol{\mu}}(\boldsymbol{u}_i^{(n)},\ \boldsymbol{\mu},\ t_{n-1}+c_i\Delta t_n), \tag{D.98}$$

where $\boldsymbol{\nu}^{(n)}$ and $\boldsymbol{\tau}_i^{(n)}$ are the Lagrange multipliers. These equations can be obtained using an identical derivation to that in Section D.4.1; see [211]. At this point, take $F = \boldsymbol{v}^T\boldsymbol{u}^{(N_t)}$ and $\boldsymbol{\mu} = \boldsymbol{u}_0$ for a fixed, arbitrary vector $\boldsymbol{v} \in \mathbb{R}^{N_{\boldsymbol{u}}}$. For this selection of F and $\boldsymbol{\mu}$, the above equations reduce to

$$\begin{aligned}
\boldsymbol{\nu}^{(N_t)} &= \boldsymbol{v} \\
\boldsymbol{\nu}^{(n-1)} &= \boldsymbol{\nu}^{(n)} + \sum_{i=1}^{s} \Delta t_n \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}\left(\boldsymbol{u}_i^{(n)},\ \boldsymbol{\mu},\ t_{n-1}+c_i\Delta t_n\right)^T \boldsymbol{\tau}_i^{(n)} \\
\boldsymbol{M}^T\boldsymbol{\tau}_i^{(n)} &= b_i\boldsymbol{\nu}^{(n)} + \sum_{j=i}^{s} a_{ji}\Delta t_n \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}\left(\boldsymbol{u}_j^{(n)},\ \boldsymbol{\mu},\ t_{n-1}+c_j\Delta t_n\right)^T \boldsymbol{\tau}_j^{(n)}
\end{aligned} \tag{D.99}$$

and

$$\frac{\mathrm{d}F}{\mathrm{d}\boldsymbol{\mu}}^T = \frac{\partial \boldsymbol{u}^{(N_t)}}{\partial \boldsymbol{u}_0}^T \boldsymbol{v} = \boldsymbol{\nu}^{(0)}. \tag{D.100}$$

The equations in (D.99) defining $\boldsymbol{\nu}^{(0)}$ are *identical* to those in (D.79) defining $\dfrac{\partial \boldsymbol{\lambda}^{(0)}}{\partial \boldsymbol{\lambda}_{N_t}}$, which leads to the relation

$$\frac{\partial \boldsymbol{u}^{(N_t)}}{\partial \boldsymbol{u}_0}^T \boldsymbol{v} = \frac{\partial \boldsymbol{\lambda}^{(0)}}{\partial \boldsymbol{\lambda}_{N_t}}\boldsymbol{v} \tag{D.101}$$

for any $\boldsymbol{v}$. Thus, it can be concluded that

$$\frac{\partial \boldsymbol{\lambda}^{(0)}}{\partial \boldsymbol{\lambda}_{N_t}} = \frac{\partial \boldsymbol{u}^{(N_t)}}{\partial \boldsymbol{u}_0}^T. \tag{D.102}$$

Since the Jacobian of the time-periodic residual, $\dfrac{\partial \boldsymbol{u}^{(N_t)}}{\partial \boldsymbol{u}_0} - \boldsymbol{I}$, is non-singular at a time-periodic solution, the matrix defining the linear, two-point boundary value problem, $\dfrac{\partial \boldsymbol{\lambda}^{(0)}}{\partial \boldsymbol{\lambda}_{N_t}} - \boldsymbol{I}$ must also be non-singular. Thus, a solution of the linear, two-point boundary value problem exists and is unique.

Bibliography

[1] Anshul Agarwal and Lorenz T Biegler. A trust-region framework for constrained optimization using reduced order modeling. *Optimization and Engineering*, 14(1):3–35, 2013.

[2] Volkan Akcelik, George Biros, Omar Ghattas, Judith Hill, David Keyes, and Bart van Bloemen Waanders. Parallel algorithms for PDE-constrained optimization. *Parallel Processing for Scientific Computing*, 20:291, 2006.

[3] Roger Alexander. Diagonally implicit Runge-Kutta methods for stiff ODE's. *SIAM Journal on Numerical Analysis*, 14(6):1006–1021, 1977.

[4] Natalia M Alexandrov, John E Dennis Jr, Robert Michael Lewis, and Virginia Torczon. A trust-region framework for managing the use of approximation models in optimization. *Structural Optimization*, 15(1):16–23, 1998.

[5] Natalia M Alexandrov and Robert Michael Lewis. An overview of first-order model management for engineering optimization. *Optimization and Engineering*, 2(4):413–430, 2001.

[6] Natalia M Alexandrov, Robert Michael Lewis, Clyde R Gumbert, Lawrence L Green, and Perry A Newman. Approximation and model management in aerodynamic optimization with variable-fidelity models. *Journal of Aircraft*, 38(6):1093–1101, 2001.

[7] David M Ambrose and Jon Wilkening. Computation of time-periodic solutions of the Benjamin-Ono equation. *arXiv preprint arXiv:0804.3623*, 2008.

[8] David M Ambrose and Jon Wilkening. Computation of symmetric, time-periodic solutions of the vortex sheet with surface tension. *Proceedings of the National Academy of Sciences*, 107(8):3361–3366, 2010.

[9] George R Anderson, Michael J Aftosmis, and Marian Nemec. Parametric deformation of discrete geometry for aerodynamic shape design. *AIAA Paper*, 965, 2012.

[10] Eyal Arian, Marco Fahl, and Ekkehard W Sachs. Trust-region proper orthogonal decomposition for flow control. Technical report, DTIC Document, 2000.

[11] Douglas N Arnold, Franco Brezzi, Bernardo Cockburn, and L Donatella Marini. Unified analysis of discontinuous Galerkin methods for elliptic problems. *SIAM Journal on Numerical Analysis*, 39(5):1749–1779, 2002.

[12] Ivo Babuška, Fabio Nobile, and Raúl Tempone. A stochastic collocation method for elliptic partial differential equations with random input data. *SIAM Review*, 52(2):317–355, 2010.

[13] Ivo Babuska, Raúl Tempone, and Georgios E Zouraris. Galerkin finite element approximations of stochastic elliptic partial differential equations. *SIAM Journal on Numerical Analysis*, 42(2):800–825, 2004.

[14] Maciej Balajewicz and Earl H Dowell. Stabilization of projection-based reduced order models of the navier–stokes. *Nonlinear Dynamics*, 70(2):1619–1632, 2012.

[15] Afonso S Bandeira, Katya Scheinberg, and Luís N Vicente. Convergence of trust-region methods based on probabilistic models. *SIAM Journal on Optimization*, 24(3):1238–1264, 2014.

[16] Jernej Barbič and Doug L James. Real-time subspace integration for St. Venant-Kirchhoff deformable models. In *ACM transactions on graphics (TOG)*, volume 24, pages 982–990. ACM, 2005.

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

[18] Volker Barthelmann, Erich Novak, and Klaus Ritter. High dimensional polynomial interpolation on sparse grids. *Advances in Computational Mathematics*, 12(4):273–288, 2000.

[19] Ted Belytschko, Wing Kam Liu, Brian Moran, and Khalil Elkhodary. *Nonlinear Finite Elements for Continua and Structures*. John wiley & sons, 2013.

[20] Martin Philip Bendsoe and Ole Sigmund. *Topology Optimization: Theory, Methods, and Applications*. Springer Science & Business Media, 2013.

[21] Gal Berkooz, Philip Holmes, and John L Lumley. The proper orthogonal decomposition in the analysis of turbulent flows. *Annual Review of Fluid Mechanics*, 25(1):539–575, 1993.

[22] A Borzì, V Schulz, C Schillings, and G Von Winckel. On the treatment of distributed uncertainties in PDE-constrained optimization. *GAMM-Mitteilungen*, 33(2):230–246, 2010.

[23] Alfio Borzì. Multigrid and sparse-grid schemes for elliptic control problems with random coefficients. *Computing and Visualization in Science*, 13(4):153–160, 2010.

[24] Alfio Borzì and G von Winckel. Multigrid methods and sparse-grid collocation techniques for parabolic optimal control problems with random coefficients. *SIAM Journal on Scientific Computing*, 31(3):2172–2192, 2009.

[25] Matthew Brand. Incremental singular value decomposition of uncertain data with missing values. In *European Conference on Computer Vision*, pages 707–720. Springer, 2002.

[26] Matthew Brand. Fast low-rank modifications of the thin singular value decomposition. *Linear Algebra and its Applications*, 415(1):20–30, 2006.

[27] Martin Dietrich Buhmann. Radial basis functions. *Acta Numerica 2000*, 9:1–38, 2000.

[28] T. Bui-Thanh, K. Willcox, and O. Ghattas. Model reduction for large-scale systems with high-dimensional parametric input space. *SIAM Journal on Scientific Computing*, 30(6):3270–3288, 2008.

[29] Hans-Joachim Bungartz and Michael Griebel. Sparse grids. *Acta Numerica*, 13:147–269, 2004.

[30] Yanzhao Cao, MY Hussaini, and HONGTAO Yang. Numerical optimization of radiated engine noise with uncertain wavenumbers. *International Journal of Numerical Analysis and Modeling*, 4(3-4):392–401, 2007.

[31] Kevin Carlberg, Charbel Bou-Mosleh, and Charbel Farhat. Efficient non-linear model reduction via a least-squares petrov–galerkin projection and compressive tensor approximations. *International Journal for Numerical Methods in Engineering*, 86(2):155–181, 2011.

[32] Kevin Carlberg and Charbel Farhat. A low-cost, goal-oriented compact proper orthogonal decompositionbasis for model reduction of static systems. *International Journal for Numerical Methods in Engineering*, 86(3):381–402, 2011.

[33] Kevin Thomas Carlberg. *Model reduction of nonlinear mechanical systems via optimal projection and tensor approximation*. PhD thesis, Stanford University, 2011.

[34] Richard G Carter. Numerical optimization in Hilbert space using inexact function and gradient evaluations. 1989.

[35] Richard G Carter. On the global convergence of trust region algorithms using inexact gradient information. *SIAM Journal on Numerical Analysis*, 28(1):251–265, 1991.

[36] Richard G Carter. Numerical experience with a class of algorithms for nonlinear optimization using inexact function and gradient information. *SIAM Journal on Scientific Computing*, 14(2):368–388, 1993.

[37] ASL Chan. The design of Michell optimum structures. Technical report, College of Aeronautics Cranfield, 1960.

[38] Tony F Chan and Wing Lok Wan. Analysis of projection methods for solving linear systems with multiple right-hand sides. *SIAM Journal on Scientific Computing*, 18(6):1698–1721, 1997.

[39] Peter C Chang and S Chi Liu. Recent research in nondestructive evaluation of civil infrastructures. *Journal of Materials in Civil Engineering*, 15(3):298–304, 2003.

[40] I Charpentier. Checkpointing schemes for adjoint codes: Application to the meteorological model Meso-NH. *SIAM Journal on Scientific Computing*, 22(6):2135–2151, 2001.

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

[42] Peng Chen and Alfio Quarteroni. Weighted reduced basis method for stochastic optimal control problems with elliptic PDE constraint. *SIAM/ASA Journal on Uncertainty Quantification*, 2(1):364–396, 2014.

[43] Peng Chen and Alfio Quarteroni. A new algorithm for high-dimensional uncertainty quantification based on dimension-adaptive sparse grid approximation and reduced basis methods. *Journal of Computational Physics*, 298:176–193, 2015.

[44] Peng Chen, Alfio Quarteroni, and Gianluigi Rozza. Multilevel and weighted reduced basis method for stochastic optimal control problems constrained by Stokes equations. *Numerische Mathematik*, pages 1–36, 2013.

[45] Jintai Chung and GM Hulbert. A time integration algorithm for structural dynamics with improved numerical dissipation: the generalized-α method. *Journal of applied mechanics*, 60(2):371–375, 1993.

[46] Bernardo Cockburn and Chi-Wang Shu. Runge–Kutta discontinuous Galerkin methods for convection-dominated problems. *Journal of Scientific Computing*, 16(3):173–261, 2001.

[47] Earl A Coddington and Norman Levinson. *Theory of Ordinary Differential Equations*. Tata McGraw-Hill Education, 1955.

[48] Andrew R Conn, Nicholas IM Gould, and Ph L Toint. *Trust Region Methods*, volume 1. SIAM, 2000.

[49] Arnaud Debussche, Marco Fuhrman, and Gianmario Tessitore. Optimal control of a stochastic heat equation with boundary-noise and boundary-control. *ESAIM: Control, Optimisation and Calculus of Variations*, 13(01):178–205, 2007.

[50] Jean-Antoine Désidéri and Ales Janka. Multilevel shape parameterization for aerodynamic optimization: Application to drag and noise reduction of transonic/supersonic business jet. In *Proceedings of the European Congress on Computational Methods in Applied Sciences and Engineering, ECCOMAS 2004*, 2004.

[51] Benoît Desjardins, Emmanuel Grenier, P-L Lions, and Nader Masmoudi. Incompressible limit for solutions of the isentropic Navier–Stokes equations with dirichlet boundary conditions. *Journal de Mathématiques Pures et Appliquées*, 78(5):461–471, 1999.

[52] Markus A Dihlmann and Bernard Haasdonk. Certified PDE-constrained parameter optimization using reduced basis surrogate models for evolution problems. *Computational Optimization and Applications*, 60(3):753–787, 2015.

[53] Eusebius Doedel, Herbert B Keller, and Jean Pierre Kernevez. Numerical analysis and control of bifurcation problems (II): Bifurcation in infinite dimensions. *International Journal of Bifurcation and Chaos*, 1(04):745–772, 1991.

[54] Arne Drud. CONOPT: A GRG code for large sparse dynamic nonlinear optimization problems. *Mathematical Programming*, 31(2):153–191, 1985.

[55] Thomas D Economon, Francisco Palacios, and Juan J Alonso. Unsteady continuous adjoint approach for aerodynamic design on dynamic meshes. *AIAA Journal*, 53(9):2437–2453, 2015.

[56] R. Everson and L. Sirovich. Karhunen–Loève procedure for gappy data. *JOSA A*, 12(8):1657–1664, 1995.

[57] Marco Fahl and Ekkehard W Sachs. Reduced order modelling approaches to PDE-constrained optimization based on proper orthogonal decomposition. In *Large-scale PDE-constrained optimization*, pages 268–280. Springer, 2003.

[58] C. Farhat, C. Degand, B. Koobus, and M. Lesoinne. Torsional springs for two-dimensional dynamic unstructured fluid meshes. *Computer Methods in Applied Mechanics and Engineering*, 163(1–4):231–245, 1998.

[59] Charbel Farhat, Philip Avery, Todd Chapman, and Julien Cortial. Dimensional reduction of nonlinear finite element dynamic models with finite rotations and energy-based mesh sampling and weighting for computational efficiency. *International Journal for Numerical Methods in Engineering*, 98(9):625–662, 2014.

[60] Charbel Farhat, Philippe Geuzaine, and Céline Grandmont. The discrete geometric conservation law and the nonlinear stability of ALE schemes for the solution of flow problems on moving grids. *Journal of Computational Physics*, 174(2):669–694, 2001.

[61] Gerald Farin. *Curves and Surfaces for Computer-Aided Geometric Design: A Practical Guide*. Elsevier, 2014.

[62] Alexander IJ Forrester, Neil W Bressloff, and Andy J Keane. Optimization using surrogate models and partially converged computational fluid dynamics simulations. In *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, volume 462, pages 2177–2204. The Royal Society, 2006.

[63] Alexander IJ Forrester and Andy J Keane. Recent advances in surrogate-based optimization. *Progress in Aerospace Sciences*, 45(1):50–79, 2009.

[64] Bradley M Froehle. *High-order discontinuous Galerkin fluid-structure interaction methods*. PhD thesis, University of California, Berkeley, 2013.

[65] Michel Géradin and Daniel J Rixen. *Mechanical Vibrations: Rheory and Application to Structural Dynamics*. John Wiley & Sons, 2014.

[66] Thomas Gerstner and Michael Griebel. Numerical integration using sparse grids. *Numerical Algorithms*, 18(3-4):209–232, 1998.

[67] Thomas Gerstner and Michael Griebel. Dimension–adaptive tensor–product quadrature. *Computing*, 71(1):65–87, 2003.

[68] P. Geuzaine, G. Brown, C. Harris, and C. Farhat. Aeroelastic dynamic analysis of a full F-16 configuration for various flight conditions. *AIAA Journal*, 41:363–371, 2003.

[69] Omar Ghattas and Jai-Hyeong Bark. Optimal control of two-and three-dimensional incompressible Navier–Stokes flows. *Journal of Computational Physics*, 136(2):231–244, 1997.

[70] Philip E Gill, Walter Murray, and Michael A Saunders. SNOPT: An SQP algorithm for large-scale constrained optimization. *SIAM Review*, 47(1):99–131, 2005.

[71] Philip E Gill, Walter Murray, and Margaret H Wright. Practical optimization. 1981.

[72] Victor Giurgiutiu and Adrian Cuc. Embedded non-destructive evaluation for structural health monitoring, damage detection, and failure prevention. *Shock and Vibration Digest*, 37(2):83, 2005.

[73] Tuhfe Göçmen and Barış Özerdem. Airfoil optimization for noise emission problem and aerodynamic performance criterion on small scale wind turbines. *Energy*, 46(1):62–71, 2012.

[74] Jedidiah Gohlke. *Reduced Order Modeling for Optimization of Large Scale Dynamical Systems*. PhD thesis, Rice University, 2013.

[75] Gene H Golub and Charles F Van Loan. *Matrix Computations*, volume 3. JHU Press, 2012.

[76] Willy JF Govaerts. *Numerical Methods for Bifurcations of Dynamical Equilibria*, volume 66. Siam, 2000.

[77] Sanjay Govindjee, Trevor Potter, and Jon Wilkening. Cyclic steady states of treaded rolling bodies. *International Journal for Numerical Methods in Engineering*, 99(3):203–220, 2014.

[78] Max D Gunzburger. *Perspectives in Flow Control and Optimization*, volume 5. SIAM, 2003.

[79] Martin H Gutknecht. Block Krylov space methods for linear systems with multiple right-hand sides: an introduction. 2006.

[80] Raphael T Haftka and Z Mroz. First-and second-order sensitivity analysis of linear and nonlinear structures. *AIAA Journal*, 24(7):1187–1192, 1986.

[81] Nathan Halko, Per-Gunnar Martinsson, and Joel A Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Review*, 53(2):217–288, 2011.

[82] Kathryn Harriman, DJ Gavaghan, and Endre Suli. The importance of adjoint consistency in the approximation of linear functionals using the discontinuous Galerkin finite element method. Technical report, 2004.

[83] Kathryn Harriman, Paul Houston, Bill Senior, and Endre Suli. hp-version discontinuous Galerkin methods with interior penalty for partial differential equations with nonnegative characteristic form. Technical report, 2002.

[84] Ralf Hartmann. Adjoint consistency analysis of discontinuous Galerkin discretizations. *SIAM Journal on Numerical Analysis*, 45(6):2671–2696, 2007.

[85] Alexander Hay, Imran Akhtar, and Jeff T Borggaard. On the use of sensitivity analysis in model reduction to predict flows for varying inflow conditions. *International Journal for Numerical Methods in Fluids*, 68(1):122–134, 2012.

[86] Alexander Hay, Jeff Borggaard, Imran Akhtar, and Dominique Pelletier. Reduced-order models for parameter dependent geometries based on shape sensitivity analysis. *Journal of Computational Physics*, 229(4):1327–1352, 2010.

[87] Alexander Hay, Jeffrey T Borggaard, and Dominique Pelletier. Local improvements to reduced-order models using sensitivity analysis of the proper orthogonal decomposition. *Journal of Fluid Mechanics*, 629:41–72, 2009.

[88] Beichang He, Omar Ghattas, and James F Antaki. Computational strategies for shape optimization of time–dependent Navier–Stokes flows. Technical report, Carnegie Mellon University, 1997.

[89] J He and LJ Durlofsky. Constraint reduction procedures for reduced-order subsurface flow models based on pod–tpwl. *International Journal for Numerical Methods in Engineering*, 103(1):1–30, 2015.

[90] Patrick Heimbach, Chris Hill, and Ralf Giering. An efficient exact adjoint of the parallel MIT general circulation model, generated via automatic differentiation. *Future Generation Computer Systems*, 21(8):1356–1371, 2005.

[91] Matthias Heinkenschloss. Formulation and analysis of a sequential quadratic programming method for the optimal dirichlet boundary control of Navier-Stokes flow. In *Optimal Control*, pages 178–203. Springer, 1998.

[92] Matthias Heinkenschloss and Denis Ridzal. A matrix-free trust-region SQP method for equality constrained optimization. *SIAM Journal on Optimization*, 24(3):1507–1541, 2014.

[93] Matthias Heinkenschloss and Luis N Vicente. Analysis of inexact trust-region SQP algorithms. *SIAM Journal on Optimization*, 12(2):283–302, 2002.

[94] William S Hemp. *Optimum Structures*. Clarendon Press, 1973.

[95] Vincent Heuveline and Andrea Walther. Online checkpointing for parallel adjoint computation in PDEs: Application to goal-oriented adaptivity and flow control. In *Euro-Par 2006 Parallel Processing*, pages 689–699. Springer, 2006.

[96] Michael Hinze, René Pinnau, Michael Ulbrich, and Stefan Ulbrich. *Optimization with PDE Constraints*, volume 23. Springer Science & Business Media, 2008.

[97] Paul Houston and Endre Süli. hp-adaptive discontinuous galerkin finite element methods for first-order hyperbolic problems. *SIAM Journal on Scientific Computing*, 23(4):1226–1252, 2001.

[98] Sergio R Idelsohn and Alberto Cardona. A reduction method for nonlinear structural dynamic analysis. *Computer Methods in Applied Mechanics and Engineering*, 49(3):253–279, 1985.

[99] M Hasan Imam. Three-dimensional shape optimization. *International Journal for Numerical Methods in Engineering*, 18(5):661–673, 1982.

[100] Antony Jameson. Aerodynamic design via control theory. *Journal of Scientific Computing*, 3(3):233–260, 1988.

[101] Ian Jolliffe. *Principal Component Analysis*. Wiley Online Library, 2002.

[102] Martin Jones and Nail K Yamaleev. Adjoint based shape and kinematics optimization of flapping wing propulsive efficiency. In *43rd AIAA Fluid Dynamics Conference. San Diego, CA*, 2013.

[103] HB KELLER. *Numerical Methods in Bifurcation Problems*. Springer-Verlag, 1987.

[104] CT Kelley and David E Keyes. Convergence analysis of pseudo-transient continuation. *SIAM Journal on Numerical Analysis*, 35(2):508–523, 1998.

[105] CT Kelley, Li-Zhi Liao, Liqun Qi, Moody T Chu, JP Reese, and C Winton. Projected pseudo-transient continuation. 2007.

[106] Dana A Knoll and David E Keyes. Jacobian-free Newton–Krylov methods: a survey of approaches and applications. *Journal of Computational Physics*, 193(2):357–397, 2004.

[107] Drew P Kouri. An approach for the adaptive solution of optimization problems governed by partial differential equations with uncertain coefficients. Technical report, DTIC Document, 2012.

[108] Drew P Kouri, Matthias Heinkenschloss, Denis Ridzal, and Bart G van Bloemen Waanders. A trust-region algorithm with adaptive stochastic collocation for PDE optimization under uncertainty. *SIAM Journal on Scientific Computing*, 35(4):A1847–A1879, 2013.

[109] Drew P Kouri, Matthias Heinkenschloss, Denis Ridzal, and Bart G van Bloemen Waanders. Inexact objective function evaluations in a trust-region algorithm for PDE-constrained optimization under uncertainty. *SIAM Journal on Scientific Computing*, 36(6):A3011–A3029, 2014.

[110] Drew Philip Kouri and Thomas M Surowiec. Risk-averse PDE-constrained optimization using the conditional value-at-risk. Technical report, Sandia National Laboratories (SNL-NM), Albuquerque, NM (United States), 2014.

[111] J-P Kruth, Ming-Chuan Leu, and T Nakagawa. Progress in additive manufacturing and rapid prototyping. *CIRP Annals-Manufacturing Technology*, 47(2):525–540, 1998.

[112] Peter A Kuchment. *Floquet Theory for Partial Differential Equations*, volume 60. Birkhäuser, 2012.

[113] Karl Kunisch and Stefan Volkwein. Proper orthogonal decomposition for optimality systems. *ESAIM: Mathematical Modelling and Numerical Analysis*, 42(01):1–23, 2008.

[114] Toni Lassila and Gianluigi Rozza. Parametric free-form shape design with PDE models and reduced basis method. *Computer Methods in Applied Mechanics and Engineering*, 199(23):1583–1592, 2010.

[115] Patrick Allen LeGresley. *Application of proper orthogonal decomposition (POD) to design decomposition methods*. PhD thesis, Citeseer, 2005.

[116] Friedemann Leibfritz and Ekkehard W Sachs. Inexact SQP interior point methods and large scale optimal control problems. *SIAM Journal on Control and Optimization*, 38(1):272–293, 1999.

[117] Chad Lieberman, Karen Willcox, and Omar Ghattas. Parameter and state model reduction for large-scale statistical inverse problems. *SIAM Journal on Scientific Computing*, 32(5):2523–2542, 2010.

[118] Chi-Kun Lin. On the incompressible limit of the compressible Navier-Stokes equations. *Communications in Partial Differential Equations*, 20(3-4):677–707, 1995.

[119] Dong C Liu and Jorge Nocedal. On the limited memory BFGS method for large scale optimization. *Mathematical Programming*, 45(1-3):503–528, 1989.

[120] Trent Lukaczyk, Francisco Palacios, Juan J Alonso, and P Constantine. Active subspaces for shape optimization. In *Proceedings of the 10th AIAA Multidisciplinary Design Optimization Conference*, pages 1–18, 2014.

[121] L Machiels, Y Maday, and AT Patera. Output bounds for reduced-order approximations of elliptic partial differential equations. *Computer Methods in Applied Mechanics and Engineering*, 190(26):3413–3426, 2001.

[122] Yvon Maday and Einar M Rønquist. A reduced-basis element method. *Journal of scientific computing*, 17(1-4):447–459, 2002.

[123] Charles A Mader, JR RA Martins, Juan J Alonso, and E Van Der Weide. ADjoint: An approach for the rapid development of discrete adjoint solvers. *AIAA Journal*, 46(4):863–873, 2008.

[124] Karthik Mani and Dimitri J Mavriplis. Unsteady discrete adjoint formulation for two-dimensional flow problems with deforming meshes. *AIAA Journal*, 46(6):1351–1364, 2008.

[125] Andrea Manzoni, Alfio Quarteroni, and Gianluigi Rozza. Shape optimization for viscous flows by reduced basis methods and free-form deformation. *International Journal for Numerical Methods in Fluids*, 70(5):646–670, 2012.

[126] V Maple. Waterloo MAPLE software. *University of Waterloo, Version*, 5, 1994.

[127] K Maute, M Nikbay, and C Farhat. Sensitivity analysis and design optimization of three-dimensional non-linear aeroelastic systems by the adjoint method. *International Journal for Numerical Methods in Engineering*, 56(6):911–933, 2003.

[128] K Maute and M Raulli. FEM—optimization module and SDESIGN user guides, 2006.

[129] Kurt Maute, Melike Nikbay, and Charbel Farhat. Coupled analytical sensitivity analysis and optimization of three-dimensional nonlinear aeroelastic systems. *AIAA Journal*, 39(11):2051–2061, 2001.

[130] Dimitri J Mavriplis. Discrete adjoint-based approach for optimization problems on three-dimensional unstructured meshes. *AIAA Journal*, 45(4):741–750, 2007.

[131] GN Mercer and AJ Roberts. Standing waves in deep water: Their stability and extreme form. *Physics of Fluids A: Fluid Dynamics (1989-1993)*, 4(2):259–269, 1992.

[132] Asitav Mishra, Karthik Mani, Dimitri Mavriplis, and Jay Sitaraman. Time dependent adjoint-based optimization for coupled fluid–structure problems. *Journal of Computational Physics*, 292:253–271, 2015.

[133] Jorge J Moré. *Recent developments in algorithms and software for trust region methods*. Springer, 1983.

[134] Matthias Morzfeld, Xuemin Tu, Jon Wilkening, and Alexandre Chorin. Parameter estimation by implicit sampling. *Communications in Applied Mathematics and Computational Science*, 10(2):205–225, 2015.

[135] Siva Nadarajah and Antony Jameson. A comparison of the continuous and discrete adjoint approach to automatic aerodynamic optimization. *AIAA Paper*, 667:2000, 2000.

[136] Siva K Nadarajah and Antony Jameson. Optimum shape design for unsteady flows with time-accurate continuous and discrete adjoint method. *AIAA Journal*, 45(7):1478–1491, 2007.

[137] Guy Narkiss and Michael Zibulevsky. *Sequential Subspace Optimization Method for Large-Scale Unconstrained Problems*. Technion-IIT, Department of Electrical Engineering, 2005.

[138] James C Newman III, Arthur C Taylor III, Richard W Barnwell, Perry A Newman, and Gene J-W Hou. Overview of sensitivity analysis and shape optimization for complex aerodynamic configurations. *Journal of Aircraft*, 36(1):87–96, 1999.

[139] Nathan Mortimore Newmark. A method of computation for structural dynamics. In *Proc. ASCE*, volume 85, pages 67–94, 1959.

[140] Eric J Nielsen, Boris Diskin, and Nail K Yamaleev. Discrete adjoint-based design optimization of unsteady turbulent flows on dynamic unstructured grids. *AIAA Journal*, 48(6):1195–1206, 2010.

[141] Fabio Nobile, Raul Tempone, and CG Webster. The analysis of a sparse grid stochastic collocation method for partial differential equations with high-dimensional random input data. Technical report, Technical report, Sandia National Laboratories, 2007. SAND REPORT, 2007.

[142] Fabio Nobile, Raúl Tempone, and Clayton G Webster. A sparse grid stochastic collocation method for partial differential equations with random input data. *SIAM Journal on Numerical Analysis*, 46(5):2309–2345, 2008.

[143] Jorge Nocedal and Stephen Wright. *Numerical Optimization*. Springer Science & Business Media, 2006.

[144] Erich Novak and Klaus Ritter. High dimensional integration of smooth functions over cubes. *Numerische Mathematik*, 75(1):79–97, 1996.

[145] Erich Novak and Klaus Ritter. Simple cubature formulas with high polynomial exactness. *Constructive Approximation*, 15(4):499–522, 1999.

[146] Erich Novak and Henryk Woźniakowski. *Tractability of Multivariate Problems: Standard information for Functionals*, volume 12. European Mathematical Society, 2010.

[147] Carlos E Orozco and ON Ghattas. Massively parallel aerodynamic shape optimization. *Computing Systems in Engineering*, 3(1):311–320, 1992.

[148] Akira Oyama, Yoshiyuki Okabe, Koji Shimoyama, and Kozo Fujii. Aerodynamic multiobjective design exploration of a flapping airfoil using a Navier-Stokes solver. *Journal of Aerospace Computing, Information, and Communication*, 6(3):256–270, 2009.

[149] Anthony T Patera and Gianluigi Rozza. Reduced basis approximation and a posteriori error estimation for parametrized partial differential equations. Technical report, (C) MIT, Massachusetts Institute of Technology, 2007.

[150] Jaime Peraire and P-O Persson. The Compact Discontinuous Galerkin (CDG) method for elliptic problems. *SIAM Journal on Scientific Computing*, 30(4):1806–1824, 2008.

[151] Ruben E Perez, Peter W Jansen, and Joaquim RRA Martins. pyOpt: a Python-based object-oriented framework for nonlinear constrained optimization. *Structural and Multidisciplinary Optimization*, 45(1):101–118, 2012.

[152] P-O Persson, J Bonet, and J Peraire. Discontinuous Galerkin solution of the Navier–Stokes equations on deformable domains. *Computer Methods in Applied Mechanics and Engineering*, 198(17):1585–1595, 2009.

[153] P-O Persson and Jaime Peraire. Newton-GMRES preconditioning for discontinuous Galerkin discretizations of the Navier-Stokes equations. *SIAM Journal on Scientific Computing*, 30(6):2709–2733, 2008.

[154] Per-Olof Persson. Scalable parallel Newton-Krylov solvers for discontinuous Galerkin discretizations. *AIAA Paper*, 606:2009, 2009.

[155] Per-Olof Persson and Jaime Peraire. Curved mesh generation and mesh refinement using Lagrangian solid mechanics. In *Proceedings of the 47th AIAA Aerospace Sciences Meeting and Exhibit*, volume 204, 2009.

[156] Knut Petras. On the smolyak cubature error for analytic functions. *Advances in Computational Mathematics*, 12(1):71–93, 2000.

[157] Knut Petras. Smolyak cubature of given polynomial degree with few nodes for increasing dimension. *Numerische Mathematik*, 93(4):729–753, 2003.

[158] Max F Platzer, Kevin D Jones, John Young, and JC S. Lai. Flapping wing aerodynamics: progress and challenges. *AIAA Journal*, 46(9):2136–2149, 2008.

[159] MJD Powell. Convergence properties of a class of minimization algorithms. *Nonlinear Programming*, 2(0):1–27, 1975.

[160] Alfio Quarteroni and Gianluigi Rozza. Optimal control and shape optimization of aorto-coronaric bypass anastomoses. *Mathematical Models and Methods in Applied Sciences*, 13(12):1801–1823, 2003.

[161] Louis B Rall. Automatic differentiation: Techniques and applications. 1981.

[162] Ravi Ramamurti and William Sandberg. Simulation of flow about flapping airfoils using finite element incompressible flow solver. *AIAA Journal*, 39(2):253–260, 2001.

[163] James Reuther, Juan Jose Alonso, Mark J Rimlinger, and Antony Jameson. Aerodynamic shape optimization of supersonic aircraft configurations via an adjoint formulation on distributed memory parallel computers. *Computers & Fluids*, 28(4):675–700, 1999.

[164] James Reuther, Antony Jameson, James Farmer, Luigi Martinelli, and David Saunders. Aerodynamic shape optimization of complex aircraft configurations via an adjoint formulation. *AIAA paper*, 94, 1996.

[165] Michal Rewienski and Jacob White. A trajectory piecewise-linear approach to model order reduction and fast simulation of nonlinear circuits and micromachined devices. *IEEE Transactions on computer-aided design of integrated circuits and systems*, 22(2):155–170, 2003.

[166] Denis Ridzal. *Trust-region SQP methods with inexact linear system solves for large-scale optimization*. PhD thesis, Citeseer, 2006.

[167] TD Robinson, MS Eldred, KE Willcox, and R Haimes. Surrogate-based optimization using multifidelity models with variable parameterization and corrected space mapping. *AIAA Journal*, 46(11):2814–2822, 2008.

[168] Theresa Dawn Robinson. *Surrogate-based optimization using multifidelity models with variable parameterization*. PhD thesis, Massachusetts Institute of Technology, 2007.

[169] Philip L Roe. Approximate Riemann solvers, parameter vectors, and difference schemes. *Journal of Computational Physics*, 43(2):357–372, 1981.

[170] Sabrina Rogg. *Trust Region POD for Optimal Boundary Control of a Semilinear Heat Equation*. PhD thesis, University of Trier, 2014.

[171] Gianluigi Rozza. On optimization, control and shape design of an arterial bypass. *International Journal for Numerical Methods in Fluids*, 47(10-11):1411–1419, 2005.

[172] Gianluigi Rozza. *Shape design by optimal flow control and reduced basis techniques*. PhD thesis, EPFL, 2005.

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

[174] Gianluigi Rozza and Andrea Manzoni. Model order reduction by geometrical parametrization for shape optimization in computational fluid dynamics. In *Proceedings of the ECCOMAS CFD 2010, V European Conference on Computational Fluid Dynamics*, number EPFL-CONF-148535, 2010.

[175] D. Ryckelynck. A priori hyperreduction method: an adaptive approach. *Journal of Computational Physics*, 202(1):346–366, 2005.

[176] Chris H Rycroft and Jon Wilkening. Computation of three-dimensional standing water waves. *Journal of Computational Physics*, 255:612–638, 2013.

[177] Jamshid A Samareh. A survey of shape parameterization techniques. In *NASA Conference Publication*, pages 333–344. Citeseer, 1999.

[178] Claudia Schillings. *Optimal aerodynamic design under uncertainties*. PhD thesis, PhD thesis, Fb–IV, Mathematik, Universität Trier, D–54286 Trier, Germany, 2010.

[179] Thomas W Sederberg and Scott R Parry. Free-form deformation of solid geometric models. *ACM SIGGRAPH computer graphics*, 20(4):151–160, 1986.

[180] Ole Sigmund. Design of multiphysics actuators using topology optimization–part I: One-material structures. *Computer Methods in Applied Mechanics and Engineering*, 190(49):6577–6604, 2001.

[181] Ole Sigmund and Kurt Maute. Topology optimization approaches. *Structural and Multidisciplinary Optimization*, 48(6):1031–1055, 2013.

[182] Valeria Simoncini and Efstratios Gallopoulos. An iterative method for nonsymmetric systems with multiple right-hand sides. *SIAM Journal on Scientific Computing*, 16(4):917–933, 1995.

[183] L. Sirovich. Turbulence and the dynamics of coherent structures. I-coherent structures. II-symmetries and transformations. III-dynamics and scaling. *Quarterly of Applied Mathematics*, 45:561–571, 1987.

[184] Sergey A Smolyak. Quadrature and interpolation formulas for tensor products of certain classes of functions. In *Dokl. Akad. Nauk SSSR*, volume 4, page 123, 1963.

[185] Roman Srzednicki. Periodic and bounded solutions in blocks for time-periodic nonautonomous ordinary differential equations. *Nonlinear Analysis: Theory, Methods & Applications*, 22(6):707–737, 1994.

[186] Eka Suwartadi, Stein Krogstad, and Bjarne Foss. Adjoint-based surrogate optimization of oil reservoir water flooding. *Optimization and Engineering*, 16(2):441–481, 2015.

[187] Jeffrey P Thomas, Kenneth C Hall, and Earl H Dowell. Discrete adjoint approach for modeling unsteady aerodynamic design sensitivities. *AIAA Journal*, 43(9):1931–1936, 2005.

[188] Hanne Tiesler, Robert M Kirby, Dongbin Xiu, and Tobias Preusser. Stochastic collocation for optimal control problems with stochastic PDE constraints. *SIAM Journal on Control and Optimization*, 50(5):2659–2682, 2012.

[189] Ph L Toint. Global convergence of a class of trust-region methods for nonconvex minimization in Hilbert space. *IMA Journal of Numerical Analysis*, 8(2):231–252, 1988.

[190] Fredi Tröltzsch. Optimal control of partial differential equations. *Graduate studies in mathematics*, 112, 2010.

[191] Ismail H Tuncer and Mustafa Kaya. Optimization of flapping airfoils for maximum thrust and propulsive efficiency. *AIAA Journal*, 43(11):2329–2336, 2005.

[192] Nico P van Dijk, K Maute, M Langelaar, and F Van Keulen. Level-set methods for structural topology optimization: a review. *Structural and Multidisciplinary Optimization*, 48(3):437–472, 2013.

[193] Marnix P van Schrojenstein Lantman and Krzysztof Fidkowski. Adjoint-based optimization of flapping kinematics in viscous flows. In *21st AIAA Computaional Fluid Dynamics Conference*, 2013.

[194] Stefan Vandewalle and Robert Piessens. Efficient parallel algorithms for solving initial-boundary value and time-periodic parabolic partial differential equations. *SIAM Journal on Scientific and Statistical Computing*, 13(6):1330–1346, 1992.

[195] Jean Virieux and Stéphane Operto. An overview of full-waveform inversion in exploration geophysics. *Geophysics*, 74(6):WCC1–WCC26, 2009.

[196] Divakar Viswanath. Recurrent motions within plane Couette turbulence. *Journal of Fluid Mechanics*, 580:339–358, 2007.

[197] Zhi Wang, IM Navon, FX Le Dimet, and X Zou. The second order adjoint analysis: theory and applications. *Meteorology and Atmospheric Physics*, 50(1-3):3–20, 1992.

[198] Kyle Washabaugh. *Faster Fidelity For Better Design: A Scalable Model Order Reduction Framework For Steady Aerodynamic Design Applications*. PhD thesis, Stanford University, 2016.

[199] Clayton G Webster. *Sparse grid stochastic collocation techniques for the numerical solution of partial differential equations with random input data*. Flordia State University, 2007.

[200] Jon Wilkening. Breakdown of self-similarity at the crests of large-amplitude standing water waves. *Physical Review Letters*, 107(18):184501, 2011.

[201] Jon Wilkening and Jia Yu. Overdetermined shooting methods for computing standing water waves with spectral accuracy. *Computational Science & Discovery*, 5(1):014017, 2012.

[202] Matthew O Williams, Jon Wilkening, Eli Shlizerman, and J Nathan Kutz. Continuation of periodic solutions in the waveguide array mode-locked laser. *Physica D: Nonlinear Phenomena*, 240(22):1791–1804, 2011.

[203] Kaufui V Wong and Aldo Hernandez. A review of additive manufacturing. *ISRN Mechanical Engineering*, 2012, 2012.

[204] Dongbin Xiu and Jan S Hesthaven. High-order collocation methods for differential equations with random inputs. *SIAM Journal on Scientific Computing*, 27(3):1118–1139, 2005.

[205] Nail K Yamaleev, Boris Diskin, and Eric J Nielsen. Adjoint-based methodology for time-dependent optimization. *AIAA Paper*, 5857:2008, 2008.

[206] Nail K Yamaleev, Boris Diskin, and Eric J Nielsen. Local-in-time adjoint-based method for design optimization of unsteady flows. *Journal of Computational Physics*, 229(14):5394–5407, 2010.

[207] Ya-Xiang Yuan. Subspace methods for large scale nonlinear equations and nonlinear least squares. *Optimization and Engineering*, 10(2):207–218, 2009.

[208] Yao Yue and Karl Meerbergen. Accelerating optimization of parametric linear systems by model order reduction. *SIAM Journal on Optimization*, 23(2):1344–1370, 2013.

[209] Matthew J. Zahr, Kevin Carlberg, David Amsallem, and Charbel Farhat. Comparison of model reduction techniques on high-fidelity linear and nonlinear electrical, mechanical, and biological systems. Technical report, University of California, Berkeley, 2010.

[210] Matthew J. Zahr and Charbel Farhat. Progressive construction of a parametric reduced-order model for PDE-constrained optimization. *International Journal for Numerical Methods in Engineering*, 102(5):1111–1135, 2015.

[211] Matthew J. Zahr and Per-Olof Persson. An adjoint method for a high-order discretization of deforming domain conservation laws for optimization of flow problems. *Journal of Computational Physics*, In review, 2016.

[212] Matthew J. Zahr, Per-Olof Persson, and John Wilkening. A fully discrete adjoint method for optimization of flow problems on deforming domains with time-periodicity constraints. *Computers & Fluids*, 2016.

[213] Tomás Zegard and Glaucio H Paulino. Bridging topology optimization and additive manufacturing. *Structural and Multidisciplinary Optimization*, pages 1–18, 2015.

[214] Kemin Zhou, John Comstock Doyle, Keith Glover, et al. *Robust and Optimal Control*, volume 40. Prentice Hall, New Jersey, 1996.

[215] Ciyou Zhu, Richard H Byrd, Peihuang Lu, and Jorge Nocedal. Algorithm 778: L-BFGS-B: Fortran subroutines for large-scale bound-constrained optimization. *ACM Transactions on Mathematical Software (TOMS)*, 23(4):550–560, 1997.

[216] J Carsten Ziems and Stefan Ulbrich. Adaptive multilevel inexact SQP methods for PDE-constrained optimization. *SIAM Journal on Optimization*, 21(1):1–40, 2011.