Goal-Oriented Model Reduction for Time-Dependent Nonlinear Parametrized Partial Differential Equations

by

Michael Kenneth Elliott Sleeman

A thesis submitted in conformity with the requirements
for the degree of Master of Applied Science
Graduate Department of University of Toronto Institute for Aerospace Studies
University of Toronto

Abstract

Goal-Oriented Model Reduction for Time-Dependent Nonlinear Parametrized Partial Differential Equations

Michael Kenneth Elliott Sleeman
Master of Applied Science
Graduate Department of University of Toronto Institute for Aerospace Studies
University of Toronto
2020

This thesis presents a goal-oriented projection-based model reduction framework for parameterized time-dependent nonlinear PDEs. The framework builds on the following ingredients: reduced basis spaces, which provide rapidly convergent approximations of the parameterized time-dependent solutions; the empirical quadrature procedure, which provides hyperreduction of nonlinear PDEs; and the dual weighted residual (DWR) method, which provides *a posteriori* error estimates for quantities of interest. In the offline stage, the method constructs reduced models that meet the user-prescribed output error tolerance using an automated and efficient training procedure. The online stage is rapid and reliable, meaning that the method provides rapid approximation of time-dependent outputs coupled with online-efficient error estimates that make the method reliable by indicating how much error is committed by the reduced model. We demonstrate the framework for an unsteady nonlinear reaction-diffusion equation and an unsteady lid-driven cavity flow.

Acknowledgements

First and foremost I would like to thank my academic adviser, Professor Masayuki Yano, for his guidance, motivation, and inspiration throughout my time at UTIAS. I would like to specially thank the other members of my research assessment committee, Professors David Zingg and Prasanth Nair, for providing valuable feedback on my research. I would also like to thank Eugene Du, who collaborated with me and Professor Yano on model reduction methods. Throughout my time at UTIAS, Eugene, Professor Yano, and I had many fruitful discussions on model reduction methods that were critical to the development of the ideas presented in this thesis. Finally I would like to thank my labmates and friends at UTIAS: Geoff, Aaron, Keishi, Eugene, Andrew, George, Anthony, Isaac, Sohini, and Nick for making my time at UTIAS more enjoyable.

Contents

Nomenclature

u	primal field variable
$\mathbf{u}$	algebraic primal variable
z	dual field variable
$\mathbf{z}$	algebraic dual variable
ψ	tangent field variable
$\boldsymbol{\psi}$	algebraic tangent variable
μ	parameter
t	time
Ω	spatial domain
$\mathcal{D}$	parameter domain
I	time domain
$\Xi_{N_{\text{train}}}$	a set of N_{train} training parameters
$\mathcal{V}$	infinite-dimensional Sobolev space
$\mathcal{V}_h$	$\mathcal{N}_h$-dimensional finite element space
$\mathcal{V}_N^{\text{pr}}$	N^{pr}-dimensional and $\mathcal{V}$-orthonormal primal reduced basis space
$\mathcal{V}_N^{\text{du}}$	N^{du}-dimensional and $\mathcal{V}$-orthonormal dual reduced basis space
$(\cdot)^{\text{pr}}$	quantity associated with the primal space
$(\cdot)^{\text{du}}$	quantity associated with the dual space
$\tilde{(\cdot)}$	the EQP approximation of $(\cdot)$
$\{\rho_i\}_{i=1}^{M^h}$	a set of M^h quadrature weights
δ	EQP error tolerance
$s(\mu)$	output (quantity of interest)
$\eta(\mu)$	DWR error estimate (to estimate output error)

Chapter 1

Introduction

1.1 Motivation

Computational Fluid Dynamics (CFD) is commonly used in both academic and industrial settings to simulate fluid flows. CFD simulations of unsteady aerodynamic flows have a wide range of practical applications, including real-time control of unsteady aerodynamic flows [2, 28] and *digital flight-envelope characterization*, where high-fidelity simulations are performed to characterize the capabilities of aircraft (i.e. their flight envelope) [32]. These simulations are important to engineering design because they can be used to supplement physical experiments to decrease both design time and cost [41]. Performing a CFD simulation involves solving a system of nonlinear partial differential equations (PDEs), known as the Navier-Stokes (NS) equations. Unfortunately, solving the NS equations is computationally expensive since they are nonlinear and have a wide range of spatial and temporal scales. Therefore, high-fidelity CFD simulations cannot be performed as frequently as desired due to their high computational costs [41].

This motivates the search for *model reduction* methods that can be used to reduce the computational complexity of these simulations while carefully controlling the approximation error. Model reduction employs offline-online decomposition, where expensive preprocessing is performed once in the offline stage to create a reduced-order model (ROM) to be evaluated for many new parameter values in the online stage at a substantially reduced computational cost [36, 39]. In model reduction, users accept a high training cost to create the ROM in the offline stage so that the marginal cost of performing new simulations is dramatically decreased in the online stage [36, 39, 7, 27].

1.2 Background

Systematic and mature projection-based model reduction methods exist for solving linear problems. These methods allow users to rapidly evaluate the ROM for new parameter values, while in many cases also providing rigorous and sharp upper error bounds. These error bounds indicate how much error the ROM commits when evaluated for new parameter values, ensuring that the ROM is reliable. The error bounds also allow users to train ROMs efficiently using greedy algorithms [39, 43]. We note that these methods have been applied to time-dependent problem. In particular, rigorous and sharp error bounds have been developed for time-dependent parabolic equations in [25], but these error bounds do not apply to hyperbolic problems or problems with general nonlinearities. Moreover, these methods are not rapid for problems with general nonlinearities [39].

With the above in mind, we introduce *hyperreduction* methods, which are model reduction methods developed for systems of nonlinear equations, such as the NS equations. Hyperreduction for time-dependent nonlinear PDEs is an active research area and several promising methods exist, including the Empirical Interpolation Method (EIM) [5], the Gauss-Newton with Approximated Tensors (GNAT) method [13, 14], and the Energy-Conserving Sampling and Weighting (ECSW) method [18, 19]. EIM has been demonstrated for time-dependent parametrized PDEs with general nonlinearities. Moreover, EIM has been applied to time-dependent parabolic problems and used in conjunction with error bounds [22]. However, these error bounds do not apply to problems with general nonlinearities. The GNAT method is specifically designed to solve time-dependent nonlinear PDEs, while the ECSW method can be applied to both steady and time-dependent problems. Neither the GNAT method nor the ECSW method have been demonstrated for time-dependent parametrized problems. Moreover, they have not been demonstrated in conjunction with error bounds.

It is critical that engineers be able to control and quantify the uncertainty in their numerical simulations. Therefore we will develop a hyperreduction method, coupled with error estimates (as opposed to upper error bounds), to be applied to time-dependent nonlinear parametrized PDEs. In many cases, engineers are most interested in estimating the value of a state-dependent scalar output (e.g., lift or drag) for a given input parameter (e.g., angle of attack or Mach number). If an engineer wishes to estimate the value of an output, they need not directly control the global solution error and can instead aim to directly control the output error. This motivates the *goal-oriented* context. In the goal-oriented context we seek to estimate the value of our output for a given parameter. Moreover, we seek to control and estimate the error in our output.

The hyperreduction methods listed above cannot directly quantify or control the error introduced by their respective hyperreduction approximations. The empirical quadrature procedure (EQP) is a more recent development that can directly control the hyperreduction error [49]. There also exists a goal-oriented version of the EQP framework that seeks to control and estimate the output error for steady nonlinear PDEs [48]. The primary goal of this project is to extend the goal-oriented EQP framework so that it can be applied to time-dependent PDEs.

Our goal is to train a ROM such that we can estimate an output and its associated error for any parameter value, even those for which the ROM was not trained. We use a physics-based approach whereby we numerically solve our PDE using the ROM to get an approximation to the field variable. Then we use the field variable to evaluate our output. We could alternatively explore black box methods (i.e., not physics-informed) to directly interpolate the output without solving for the field variable, but it is more challenging (if not impossible) to develop error estimates for an interpolated output. The goal-oriented context is advantageous because not all features of the field variable need to be well-resolved to accurately predict the output, leading to a reduced computational cost in many cases [39].

1.3 Objectives

The goal of this thesis is to develop a model reduction method for time-dependent nonlinear parametrized PDEs. We will develop this method in the context of finite element methods, which we will refer to as our full-order model. We wish for the model reduction method to be rapid, reliable, efficient, and automated. By this we mean,

- **rapid**: the computational cost to evaluate the ROM for any parameter value in the online stage is orders of magnitude lower than evaluating the full-order model.

- **reliable**: users have quantitative control of the output error coupled with an efficient and effective *a posteriori* error estimate in the online stage, while the training algorithm in the offline stage provides quantitative control of the output error arising from the finite element approximation, the reduced basis approximation, and hyperreduction.

- **efficient**: the adaptive training algorithm minimizes the offline training cost by carefully selecting which parameters values are evaluated using the full-order model.

- **automated**: training procedure is robust (i.e., it can be applied to a wide range of PDEs) and training requires minimal user intervention (i.e., relatively few tuning parameters so that users only need to adjust error tolerances).

1.4 Thesis Outline

The goal of this work is to develop a model reduction method for time-dependent nonlinear parametrized PDEs. In chapter 2 we introduce a model problem and then explain the relevant background information in terms of the model problem. We review finite element methods, time-marching methods, existing model reduction methods, and *a posteriori* error estimation techniques. In chapter 3 we review the existing empirical quadrature procedure, including the goal-oriented framework for steady problems. Then in chapter 4 we extend the theory for the goal-oriented empirical quadrature procedure so that it applies to time-dependent problems. In chapter 5 we present more details on how to implement our fully-adaptive training algorithm for time-dependent problems, then in chapter 6 we present numerical results to demonstrate our model reduction method for a nonlinear reaction-diffusion equation and a lid-driven cavity flow (incompressible Navier-Stokes). We will also assess (i) the offline training cost, (ii) the online cost, and (iii) the accuracy of the online model. Finally we conclude the thesis in chapter 7 with a summary of the main findings of this thesis and recommendations for future work.

Chapter 2

Background

Here we present the background required to understand this work. First we will explain how to discretize a general time-dependent PDE in space and time. Then we will outline existing model reduction methods and explain why we have developed a new method. We will also include a discussion of *a posteriori* error estimation methods that we will use to develop our new method.

2.1 Model Problem

Here we will describe how to discretize the time-dependent nonlinear reaction-diffusion equation presented in [23]. We note that we discretize this PDE in both space and time and that we can use similar techniques to discretize a general PDE in space and time. We state the problem: given any $\mu \in \mathcal{D}$, find $u(\mu) : \Omega \times I \to \mathbb{R}$ such that

$$\begin{aligned} \frac{\partial u(\mu)}{\partial t} - \nabla^2 u(\mu) + g(u(\mu);\mu) = f \quad &\text{on} \quad \Omega \times I, \\ u(\mu) = 0 \quad &\text{on} \quad \partial\Omega \times I, \\ u(\mu) = 0 \quad &\text{on} \quad \partial\Omega \times \{t = 0\}, \end{aligned} \tag{2.1}$$

where $\Omega \equiv (0,1)^2 \subset \mathbb{R}^2$ is the unit square spatial domain with the boundary $\partial\Omega$, $I \equiv (0,2]$ is the time interval, and $\mathcal{D} \equiv [0.01, 10]^2 \subset \mathbb{R}^2$ is the parameter domain. The parameter tuple μ has two scalar entries where $\mu = (\mu_{(1)}, \mu_{(2)})$. Throughout this work, $u(\mu) : \Omega \times I \to \mathbb{R}$ denotes a space-time function, and we use $u(\mu, t) : \Omega \to \mathbb{R}$ to denote the space-only function that results from the evaluation of $u(\mu)$ at time t. We further

note that the nonlinear reaction functional is defined as

$$g(u(\mu);\mu) = \mu_{(1)}\frac{e^{\mu_{(2)}u(\mu)}-1}{\mu_{(2)}}, \tag{2.2}$$

while the time-dependent forcing function is defined as

$$f((x_{(1)},x_{(2)}),t) = 100\sin(2\pi t)\sin(2\pi x_{(1)})\cos(2\pi x_{(2)}). \tag{2.3}$$

There is no steady-state because our forcing function is sinusoidal in time. We compute the output

$$s(\mu) = \int_I q(u(\mu,t);\mu,t)dt + q_T(u(\mu,T);\mu,T). \tag{2.4}$$

where

$$q(u(\mu,t);\mu,t) \equiv \int_\Omega u(\mu,t)dx \quad \forall t \in I,$$
$$q_T(u(\mu,T);\mu,T) \equiv \int_\Omega u(\mu,T)dx,$$

and we recall that $u(\mu,t) : \Omega \to \mathbb{R}$ denotes the space-time function $u(\mu) : \Omega \times I \to \mathbb{R}$ evaluated at time t.

2.2 Finite Element Approximation

We refer to equation (2.1) as the strong form of our PDE. We use the *weighted residual method* to convert our PDE to the more general variational (weak) form, which is more amenable to finite element and reduced basis approximations. We multiply the PDE by a test function v and then integrate over the spatial domain. We restrict v so that it lies in the space $\mathcal{V} \equiv H_0^1(\Omega) \subset H^1(\Omega)$, where $H^1(\Omega)$ is a Sobolev space of functions that are square integrable and whose weak first derivative is also square integrable, while the space $\mathcal{V} \equiv H_0^1(\Omega)$ is a subset of $H^1(\Omega)$ whose functions vanish on the boundary. Next we define the residual form $r : \mathcal{V} \times \mathcal{V} \times \mathcal{D} \times I \to \mathcal{V}$ and the bilinear form $m : \mathcal{V} \times \mathcal{V} \to \mathcal{V}$

$$r(w,v;\mu,t) \equiv \int_\Omega \nabla v \cdot \nabla w dx + \int_\Omega v g(w;\mu)dx - \int_\Omega v f(t)dx \quad \forall w,v \in \mathcal{V}, \tag{2.5}$$
$$m(w,v) \equiv \int_\Omega vwdx \quad \forall w,v \in \mathcal{V}, \tag{2.6}$$

where $f(t) : \Omega \to \mathbb{R}$ is the source function. Then we define the continuous-in-time unsteady residual $\bar{r} : \mathcal{V} \times \mathcal{V} \times \mathcal{D} \times I \to \mathcal{V}$

$$\bar{r}(w, v; \mu, t) \equiv m(\partial_t w, v) + r(w, v; \mu, t) \quad \forall w, v \in \mathcal{V}. \tag{2.7}$$

We state the weak problem: given $\mu \in \mathcal{D}$, find $u(\mu) \in C^0(I; \mathcal{V})$ such that

$$\bar{r}(u(\mu), v; \mu, t) = 0 \quad \forall v \in \mathcal{V}, \ \forall t \in I.$$

We note that (2.7) is the general weak form of unsteady PDEs and consists of (i) an unsteady mass term, and (ii) the weak form of the source and flux terms.

The space $\mathcal{V}$ is infinite-dimensional, so our current variational form is computationally intractable. Therefore we seek a finite-dimensional "truth" space $\mathcal{V}_h \subset \mathcal{V}$ of dimension $\mathcal{N}_h$ that approximates $\mathcal{V}$ and is amenable to computation. To this end, we triangulate the domain $\Omega \equiv (0, 1)^2$ into n_e non-overlapping elements such that we have the triangulation $\mathcal{T}_h \equiv \{J_i\}_{i=1}^{n_e}$. We can now introduce the "truth" space $\mathcal{V}_h$, which is a space of piecewise p-degree polynomials $\mathcal{V}_h \equiv \{v \in \mathcal{V} \mid v|_{J_i} \in \mathbb{P}^p(J_i), \ i = 1, 2, \ldots, n_e\}$. For our nonlinear reaction-diffusion equation, we choose to use third-degree polynomials ($p = 3$). We also introduce the "truth" quadrature rule $\{\xi_\kappa^h, \rho_\kappa^h\}_{\kappa=1}^{M^h}$, along with the residual form $r_h : \mathcal{V}_h \times \mathcal{V}_h \times \mathcal{D} \times I \to \mathcal{V}_h$ and the mass form $m_h : \mathcal{V}_h \times \mathcal{V}_h \to \mathcal{V}_h$

$$r_h(w, v; \mu, t) \equiv \sum_{\kappa=1}^{M^h} \rho_\kappa^h r_\kappa(w, v; \mu, t), \tag{2.8}$$

$$m_h(w, v) \equiv \sum_{\kappa=1}^{M^h} \rho_\kappa^h m_\kappa(w, v), \tag{2.9}$$

where ρ_κ^h is the quadrature weight associated with quadrature point κ, while $r_\kappa(\cdot, \cdot; \cdot, \cdot)$ and $m_\kappa(\cdot, \cdot)$ are the functions evaluated at quadrature point κ such that

$$r_\kappa(w, v; \mu, t) \equiv \nabla v(\xi_\kappa^h, t) \cdot \nabla w(\xi_\kappa^h, t) + v(\xi_\kappa^h, t) g(w(\xi_\kappa^h, t); \mu) - v(\xi_\kappa^h, t) f(\xi_\kappa^h, t), \tag{2.10}$$

$$m_\kappa(w, v) \equiv v(\xi_\kappa^h, t) w(\xi_\kappa^h, t). \tag{2.11}$$

We note that $r_h(w, v; \mu, t) \approx r(w, v; \mu, t)$ and $m_h(w, v) \approx m(w, v)$.

We discretize our PDE in time using the either the backward-Euler or Crank-Nicolson time-marching method. Here we use the backward Euler method ($\frac{\partial u_{h,k}}{\partial t} \approx \frac{u_{h,k} - u_{h,k-1}}{\Delta t}$) with the uniform time step $\Delta t = (T - t_0)/K = 2/K$ and the associated discrete times $t_k = k\Delta t$, $0 \leq k \leq K$. We distinguish between the semi-discrete solution $u_{h,k}(\mu)$ and the

continuous solution $u_h(\mu, t_k)$ by writing $u_{h,k}(\mu) \approx u_h(\mu, t_k)$. We state our fully-discrete problem: given $\mu \in \mathcal{D}$, find $u_{h,k}(\mu) \in \mathcal{V}_h$, $k = 1, 2, \ldots, K$ such that $u_{h,k=0}(\mu) = 0$ and

$$m_h(u_{h,k}(\mu) - u_{h,k-1}(\mu), v) + (t_k - t_{k-1})(r_h(u_{h,k}(\mu), v; \mu, t_k)) = 0 \quad \forall v \in \mathcal{V}_h. \tag{2.12}$$

Following the conventions of the RB community, we refer to the FE solutions as the "truth" solutions because they represent our best approximation to the true solutions and because they lie in the "truth" space.

To evaluate the output, we solve for the field variable at time steps $k = 1, 2, \ldots, K$ and use linear-interpolation combined with a two-point Gauss quadrature rule to numerically integrate the output in time. The Crank-Nicolson method is second-order accurate in time, while linear interpolation is second-order accurate and the two-point Gauss quadrature rule is fourth-order accurate. Therefore the linear-interpolation and the two-point Gauss quadrature rule do not introduce additional error in time.

2.2.1 Algebraic Form

We have introduced the variational form of the FE problem. However, we will employ the algebraic form of the FE problem during our discussion of some of the hyperreduction methods. Therefore we will now introduce the algebraic form. We have a basis $\{\phi_i\}_{i=1}^{\mathcal{N}_h}$ for $\mathcal{V}_h$, and given the the function $w \in \mathcal{V}_h$, we introduce the generalized coordinate $\mathbf{w} \in \mathbb{R}^{\mathcal{N}_h}$ where

$$w(x) \equiv \sum_{i=1}^{\mathcal{N}_h} \mathbf{w}_i \phi_i(x). \tag{2.13}$$

Similarly we can introduce the algebraic form of the residual form $\mathbf{r}_h : \mathbb{R}^{\mathcal{N}_h} \times \mathcal{D} \times I \to \mathbb{R}^{\mathcal{N}_h}$

$$\mathbf{r}_h(\mathbf{w}; \mu, t)_i \equiv r_h(w, \phi_i; \mu, t), \quad i = 1, 2, \ldots, \mathcal{N}_h, \tag{2.14}$$

and the mass matrix $\mathbf{M}_h \in \mathbb{R}^{\mathcal{N}_h \times \mathcal{N}_h}$

$$(\mathbf{M}_h)_{ij} \equiv m_h(\phi_j, \phi_i), \quad i, j = 1, 2, \ldots, \mathcal{N}_h. \tag{2.15}$$

We introduce the discrete-in-time residual form $\bar{\mathbf{R}}_h : \mathbb{R}^{\mathcal{N}_h} \times \mathbb{R}^{\mathcal{N}_h} \times \mathcal{D} \to \mathbb{R}^{\mathcal{N}_h}$

$$\bar{\mathbf{R}}_h(\mathbf{u}_{h,k}; \mathbf{u}_{h,k-1}; \mu) \equiv \mathbf{M}_h \mathbf{u}_{h,k} - \mathbf{M}_h \mathbf{u}_{h,k-1} + (t_k - t_{k-1}) \mathbf{r}_h(\mathbf{u}_{h,k}; \mu, t_k), \tag{2.16}$$

for $k = 1, 2, \ldots, K$ where we have applied backward Euler time-marching. We state the algebraic form of our problem: given $\mu \in \mathcal{D}$, find $\mathbf{u}_{h,k} \in \mathbb{R}^{\mathcal{N}_h}$, $k = 1, 2, \ldots, K$ such that

$$\bar{\mathbf{R}}_h(\mathbf{u}_{h,k}; \mathbf{u}_{h,k-1}; \mu) = 0 \quad \text{in} \quad \mathbb{R}^{\mathcal{N}_h}. \tag{2.17}$$

We introduce the algebraic form of the Jacobian $\mathbf{J}_h : \mathbb{R}^{\mathcal{N}_h} \times \mathcal{D} \times I \to \mathbb{R}^{\mathcal{N}_h \times \mathcal{N}_h}$

$$\mathbf{J}_h(\mathbf{w}; \mu, t)_{ij} \equiv \frac{\partial \mathbf{r}_{h,i}}{\partial \mathbf{w}_j}(\mathbf{w}; \mu, t), \quad i, j = 1, 2, \ldots, \mathcal{N}_h,$$

and the discrete-in-time Jacobian $\bar{\mathbf{J}}_h : \mathbb{R}^{\mathcal{N}_h} \times \mathcal{D} \times I \to \mathbb{R}^{\mathcal{N}_h \times \mathcal{N}_h}$

$$\bar{\mathbf{J}}_h(\mathbf{w}; \mu, t) \equiv \mathbf{M}_h + \mathbf{J}_h(\mathbf{w}; \mu, t).$$

We note that the Jacobian does not depend on $u_{h,k-1}$ because we do not take derivatives about $u_{h,k-1}$, so all of the terms associated with $u_{h,k-1}$ disappear.

2.2.2 Nonlinear Solver

We consider the the algebraic form of our model problem (2.17) at time-step k. We have a system of $\mathcal{N}_h$ equations with $\mathcal{N}_h$ unknowns (degrees of freedom). We will use Newton's method

$$\mathbf{u}_{k,n+1} - \mathbf{u}_{k,n} = -\bar{\mathbf{J}}_h^{-1}(\mathbf{u}_{k,n}; \mu, t_k)\bar{\mathbf{R}}_h(\mathbf{u}_{k,n},; \mu, t_k). \tag{2.18}$$

to iterate until the $\mathcal{V}_h$-norm of our residual converges to a sufficiently small value relative to the norm of the residual for the initial guess. The cost of each iteration of Newton's method depends on $\mathcal{N}_h$, which can be quite costly for large $\mathcal{N}_h$ and motivates our search for model reduction methods.

2.3 Model Reduction

Solving PDEs in a high-dimensional "truth" space is computationally expensive since the computational cost depends on the dimension of the "truth" space $\mathcal{N}_h \equiv \dim(\mathcal{V}_h)$. This motivates our search for model-reduction methods. We introduce the parametric manifold $\mathcal{M} = \{u(\mu, t)\}_{\mu \in \mathcal{D}, t \in I}$ and assume that this manifold is amenable to approximation by a low-dimensional linear space (a *reduced space*). We can project our problem onto this reduced space and solve it at a reduced cost. ROMs will ideally produce nearly the same input-output response as the high-dimensional "truth" approximation and we can estimate the error in our output using *a posteriori* error estimators. Model reduction

relies on offline-online decomposition, where expensive pre-processing is performed once in the offline stage to create a ROM that can be evaluated rapidly and reliably for new parameter values in the online stage [36, 39].

The key ingredients for a ROM are (i) a reduced basis (RB) space, (ii) a projection method, (iii) an efficient offline-online decomposition, (iv) hyperreduction (for nonlinear PDEs), and (v) an *a posteriori* error estimator. Model reduction methods do not replace FE methods. Instead, FE methods are used to train ROMs to enable faster and more efficient simulations. Moreover, the accuracy of a ROM is measured with respect to FE solutions (i.e., the "truth" solutions).

2.3.1 Reduced Basis Spaces

In many cases, the parametrically induced manifold $\mathcal{M} \equiv u(\mu, t)_{\mu \in \mathcal{D}, t \in I}$ is smooth and can be approximated using a low-dimensional linear space. The parametrically induced manifold represents the set of fields engendered as the input parameter and time vary over the parameter and time domain, respectively. The goal of reduced basis (RB) approximation is to construct a low-dimensional approximation space where solutions to the PDE can be approximated reasonably well. The manifold $\mathcal{M}$ is key to projection-based model reduction because if we can approximate our solutions using a low-dimensional RB space, it implies that the generic approximation spaces used by FE methods are unnecessarily rich [39].

We will consider constructing our RB spaces using two different approaches: (i) proper orthogonal decomposition (POD), and (ii) greedy sampling. Both of these approaches seek to minimize the difference between each snapshot and a linear combination of the RB space. We define the reduced basis $\zeta^n, 1 \leq n \leq N$, which spans the reduced basis space $\mathcal{V}_N \equiv \text{span}\{\zeta^n, 1 \leq n \leq N\}$. The reduced basis will always be hierarchical and orthogonal so that $\mathcal{V}_1 \subset \mathcal{V}_2 \subset \cdots \subset \mathcal{V}_{N-1} \subset \mathcal{V}_N$. The hierarchical condition of RB spaces is important to ensuring memory efficiency, while the orthogonality condition ensures that the reduced basis does not become colinear.

In the POD approach we collect solution "snapshots" from the FE solver and solve an optimization problem to compress these snapshots into an RB space. We introduce a set of N_{train} training parameters $\Xi_{N_{\text{train}}} \equiv \{\mu_n \in \mathcal{D}\}_{n=1}^{N_{\text{train}}}$ and then we collect solution snapshots

$$W_{N_{\text{train}}} = \text{span}\{u_{h,k}(\mu_n), 1 \leq n \leq N_{\text{train}}, 1 \leq k \leq K\},$$

by solving the time-dependent FE problem for each parameter in $\Xi_{N_{\text{train}}}$. This gives $N_{\text{sample}} = N_{\text{train}} \times K$ solution snapshots. POD is an optimization problem that can

be reduced to the solution of an $N_{\text{sample}} \times N_{\text{sample}}$ correlation-matrix eigenproblem or a singular value decomposition of $W_{N_{\text{train}}}$. Optimal POD spaces are given by

$$\mathcal{V}_N^{\text{POD}} = \arg \inf_{\substack{\text{spaces } \mathcal{V}_N \subset \\ \text{span}\{W_{N_{\text{train}}}\}}} \left(\sum_{\substack{\mu \in \Xi_{N_{\text{train}}} \\ k=1,2,\ldots,K}} \inf_{w_N \in \mathcal{V}_N} \|u_{h,k}(\mu) - w_N\|_{\mathcal{V}}^2 \right), \tag{2.19}$$

where $N \equiv \dim(\mathcal{V}_N)$.

POD is expensive for large N_{train} because performing the necessary FE solves entails a high computational cost. We can decrease the cost of POD by using a smaller $\Xi_{N_{\text{train}}}$ but we may construct a poor approximation space if N_{train} is too small [39]. We can alternatively consider a greedy procedure to select $\Xi_{N_{\text{train}}}$ so that we construct $\mathcal{V}_N$ such that it spans the parameter and time spaces using a smaller N_{train}. The greedy procedure chooses parameters according to *a posteriori* error estimators to assess how well a snapshot is approximated by the existing RB space [39, 43]. Greedy algorithms have been coupled with POD in previous work on model reduction for time-dependent problems [22, 25]. In section 5.1 we describe a greedy algorithm (algorithm 1) to be used with our newly developed hyperreduction framework.

2.3.2 Projection

After we construct the RB space $\mathcal{V}_N$, we can solve the weak form of our PDE using Galerkin projection onto the RB space. We state the variational form of our RB problem: given $\mu \in \mathcal{D}$ find $u_{N,k}(\mu) \in \mathcal{V}_N$, $k = 1, 2, \ldots, K$, such that $u_{N,k=0}(\mu) = 0$ and

$$m_h(u_{N,k}(\mu) - u_{N,k-1}(\mu), v; \mu, t) + (t_k - t_{k-1})(r_h(u_{N,k}(\mu), v; \mu, t)) = 0 \quad \forall v \in \mathcal{V}_N,$$

which we notice is identical to the "truth" problem (2.12), but projected onto the RB space. We can also introduce the algebraic form of the RB problem. We introduce the operator $V_N : \mathbb{R}^N \to \mathcal{V}_N$, which associates a generalized coordinate $\mathbf{w} \in \mathbb{R}^N$ to a function w as

$$w = V_N \mathbf{w} = \sum_{i=1}^{N} \mathbf{w}_i \zeta_i. \tag{2.20}$$

We introduce the algebraic residual $\mathbf{r}_N : \mathbb{R}^N \times \mathcal{D} \times I \to \mathbb{R}^N$

$$\mathbf{r}_{N,i}(\mathbf{w}; \mu, t) \equiv r_h(V_N \mathbf{w}, \zeta_i; \mu, t), \quad i = 1, 2, \ldots, N, \tag{2.21}$$

and the mass matrix $\mathbf{M}_N \in \mathbb{R}^{N\times N}$

$$(\mathbf{M}_N)_{ij} \equiv m_h(\zeta_j, \zeta_i), \quad i,j = 1,2,\ldots,N. \tag{2.22}$$

We introduce the discrete-in-time RB residual $\bar{\mathbf{R}}_N : \mathbb{R}^N \times \mathbb{R}^N \times \mathcal{D} \to \mathbb{R}^N$

$$\bar{\mathbf{R}}_N(\mathbf{u}_{N,k}; \mathbf{u}_{N,k-1}; \mu) \equiv \mathbf{M}_N\mathbf{u}_{N,k} - \mathbf{M}_N\mathbf{u}_{N,k-1} + (t_k - t_{k-1})\mathbf{r}_N(\mathbf{u}_{N,k}; \mu, t_k) \tag{2.23}$$

which we discretize in time using the backward Euler time-marching method for $k = 1,2,\ldots,K$. We state the algebraic form of our problem: given $\mu \in \mathcal{D}$, find $\mathbf{u}_{N,k} \in \mathbb{R}^N$, $k = 1,2,\ldots,K$ such that

$$\bar{\mathbf{R}}_{N,k}(\mathbf{u}_{N,k}; \mathbf{u}_{N,k-1}; \mu) = 0 \quad \text{in} \quad \mathbb{R}^N. \tag{2.24}$$

We need the Jacobian so that we can use Newton's method. We introduce the algebraic form of the RB Jacobian $\mathbf{J}_N : \mathbb{R}^N \times \mathcal{D} \times I \to \mathbb{R}^{N\times N}$

$$\mathbf{J}_N(\mathbf{w}; \mu, t)_{ij} \equiv \frac{\partial \mathbf{r}_{N,i}}{\partial \mathbf{w}_j}(\mathbf{w}; \mu, t), \quad i,j = 1,2,\ldots,N \tag{2.25}$$

and the discrete-in-time Jacobian $\bar{\mathbf{J}}_N : \mathbb{R}^N \times \mathcal{D} \times I \to \mathbb{R}^{N\times N}$

$$\bar{\mathbf{J}}_N(\mathbf{w}; \mu, t_k) \equiv \mathbf{M}_N + \mathbf{J}_N(\mathbf{w}; \mu, t). \tag{2.26}$$

Again we note that the Jacobian does not depend on $u_{N,k-1}$ because we do not take derivatives about $u_{N,k-1}$, so all of the terms associated with $u_{N,k-1}$ disappear.

The computational cost of solving the PDE in the low-dimensional RB space still depends on $\mathcal{N}_h$ because we must appeal to FE spaces each time we wish to evaluate our residual and Jacobian. However, the cost of the operation $\bar{\mathbf{J}}_N(\mathbf{u}_{N,k}; \mu)^{-1}\bar{\mathbf{R}}_N(\mathbf{u}_{N,k}; \mathbf{u}_{N,k-1}; \mu)$ has been reduced since we fewer matrix operations to perform. If we construct our RB space using POD, then this method is known as the *POD-Galerkin* method [39].

We will use the POD-Galerkin method, which we may also combine with greedy sampling (i.e., POD-in-time and greedy-in-parameters), throughout most of this work. However, we will also consider the minimum-residual method, which has been applied to steady problems in [29, 30, 9] and to time-dependent problems in [13]. The minimum-residual method is a popular alternative to the POD-Galerkin method that is used by

the GNAT method [13]. It involves solving the non-linear least-squares problem

$$\underset{u_{N,k}\in\mathcal{V}_N}{\text{minimize}}\|\bar{\mathbf{R}}_h(u_{N,k};u_{N,k-1};\mu)\|_2, \tag{2.27}$$

which seeks solutions that minimize the norm of the residual in the l^2-Euclidean norm. We apply the Gauss-Newton method, which is equivalent to performing a Petrov-Galerkin projection with the trial space $\mathcal{V}_N$ and the test space $\bar{\mathbf{J}}_N\mathcal{V}_N$. This method is also called the Least-Squares Petrov-Galerkin (LSPG) projection method. LSPG projection minimizes the norm of the FE residual at each time-step, so it can be said to be discrete-optimal. The POD-Galerkin method and the LSPG method are compared in detail in [12] and LSPG projection is shown using numerical experiments to have better time stability properties for certain cases. Numerical experiments for a limited number of test cases have shown that LSPG projection can generate stable and accurate solutions for certain large-scale turbulent, compressible flow problems where Galerkin projection has failed [12].

We further note that although projection onto the RB spaces decreases the number of equations and unknowns, we may still need to appeal to FE spaces to evaluate the residual and the Jacobian, meaning that the cost of solving for the field variable still depends on $\mathcal{N}_h$. We wish to solve for the field variable with a cost that is independent of $\mathcal{N}_h$, and the following sections will discuss how this can be achieved.

2.3.3 Offline-Online Decomposition

The goal of model reduction is to construct a ROM in the offline stage that can be evaluated with a reduced complexity for new parameter values in the online stage. When performed correctly, offline-online decomposition leads to dramatically reduced storage costs and computation time in the online stage as compared to the "truth" approximation [36, 39]. The complexity of the offline stage depends on $\mathcal{N}_h$ while ideally the complexity of the online stage is independent of $\mathcal{N}_h$ (i.e. decoupled). If (i) our residual is affine in functions of the parameters $\mu \in \mathcal{D}$ and the time $t \in I$, and (ii) the PDE is linear in the state, then we can express our residual as

$$r(w,v;\mu,t) = \sum_{q=1}^{Q_r} \theta_r^q(\mu,t) r^q(w,v) \quad \forall w,v \in \mathcal{V}, \quad \forall \mu \in \mathcal{D}, \quad \forall t \in I, \tag{2.28}$$

where $\theta_r^q(\mu,t)$ are parameter and time dependent functions, while $r^q(w,v)$ are continuous bilinear forms that are independent of the parameters and time. Each of the continuous bilinear forms $r^q(w,v)$ can be evaluated in the offline stage so that the computational cost

of the online stage depends only on N. The residuals of PDEs with general nonlinearities do not admit an affine decomposition, which means that users must appeal to the "truth" space each time they wish to evaluate their residual. This means that the POD-Galerkin approach does not typically admit an efficient online-offline decomposition for PDEs with general nonlinearities, which motivates our search for *hyperreduction* methods [7].

2.3.4 *A Posteriori* Error Estimation

Effective *a posteriori* error estimation is crucial to ensuring rapid, reliable, and automated reduced models. Error estimates enable rapid ROMs by identifying the RB space of lowest dimension N that achieves the requisite accuracy. Error estimates enable reliable ROMs by allowing users to quantify the error in their simulations, which is critical in engineering contexts [39]. Error estimates can be used to design greedy algorithms that permit automated and adaptive offline training by indicating which parameters need to be solved using FE methods and whether or not the ROM is currently accurate enough (i.e., do we need more FE solves for certain parameters associated with a larger error estimate?) [43]. In section 6.1.6 we demonstrate how greedy algorithms permit much larger training sets $\Xi_{N_{\text{train}}} \in \mathcal{D}$ at a greatly reduced offline computational cost. In summary, *a posteriori* error estimation not only allows users to estimate the error introduced by their ROM in the online stage, but also to control the error introduced by their ROM, and to minimize the computational cost of the offline stage.

Error estimates and error bounds should be rigorous so that they are valid for all N and for all parameters in $\mathcal{D}$. We also want to avoid error estimates and error bounds that are overly conservative so that the computational cost in the online stage does not grow too large to achieve the desired error tolerance (i.e., we seek sharp error estimates). The effectivity of an error estimate indicates how close the error estimate is to the true value of the error. Given an output error $s_N(\mu) - s_h(\mu)$ and an output error estimate $\eta_h(\mu)$, our effectivity is defined as $\text{eff}(\mu) \equiv |s_N(\mu) - s_h(\mu)|/|\eta_h(\mu)|$. An error estimate is more effective if its effectivity is closer to unity.

We will now briefly discuss "classical" RB error estimates, which are defined relative to the underlying FE approximation and require the residual to be a bilinear form. The RB error estimates are chosen to be an upper bound whose effectivity is as close to unity as possible. The error estimates also depend on the coercivity constant of the bilinear form and the Cauchy-Schwarz inequality. We do not present the "classical" RB error estimate here, and instead refer readers to [39]. These error bounds are sharp for coercive equations and apply to all N and $\mathcal{D}$. However, they do not apply to time-dependent PDEs with

general nonlinearities.

Previous work has developed rigorous and sharp upper error bounds for time-dependent parabolic problems, coupled with greedy training algorithms [25, 24, 22]. Previous work on EIM has also demonstrated the effectiveness of greedy training algorithms for time-dependent problems with general nonlinearities [22]. Space-time formulations that include *a posteriori* error bounds for parabolic problems were developed in [42] and demonstrated for Burgers' equation in [50]. The space-time formulation was also applied to the Boussinesq equations (i.e., the Navier-Stokes equations with heat transfer) in [44].

Although error bounds have previously been developed in the context of RB approximation for time-dependent problems, these error bounds do not apply to time-dependent problems with general nonlinearities. We will instead employ the dual-weighted residual (DWR) method to estimate the output error [6]. The DWR error estimate is not an error bound, but it has previously been used very effectively in the context of ROM for (i) time-dependent problems [33], (ii) h-refinement [11], and (iii) statistically modelling ROM output errors [31]. The major drawback of the the DWR method is that it does not provide an upper error bound. However, the major advantages of the DWR method are that it readily treats hyperbolic and convection-dominated PDEs with general nonlinearity and parametric dependence, while also not requiring the user to compute a stability constant [48]. We require error estimates to be online-efficient, so we will develop one ROM to solve our PDE and a second ROM to evaluate our DWR error estimate.

2.3.5 Hyperreduction

We wish to develop a framework that will allow users to automatically train projection-based ROMs with an efficient offline-online decomposition for time-dependent nonlinear parametrized PDEs. We will use hyperreduction to decouple the cost of performing a simulation in the online stage from $\mathcal{N}_h$. We will use hyperreduction to create approximations to our nonlinear residuals such that they admit an efficient offline-online decomposition where evaluation in the online stage is independent of $\mathcal{N}_h$. We will consider an arbitrary time-dependent nonlinear residual

$$r(w, v; \mu, t) \equiv \int_\Omega v\eta(w; \mu, t; x)dx, \quad \forall w, v \in \mathcal{V}, \tag{2.29}$$

that consists of two parts: (i) an integral over the domain Ω, and (ii) a nonlinear function $\eta : \mathcal{V} \times \mathcal{D} \times I \times \Omega$. We note that we consider a residual without boundary integration to simplify the presentation, but that the methods we present can equally be applied to residuals with boundary integration. To reduce the complexity of evaluat-

ing (2.29), we can choose to introduce additional approximations for either the integral or the nonlinear function. A popular approach is *interpolation-then-integration*, where a function is evaluated approximately using a sparse interpolation system over an RB space, while the integral is evaluated using the "truth" quadrature rules. Examples of interpolation-then-integration methods include Gappy POD [17], the Empirical Interpolation Method (EIM) [5], the Missing Point Estimation (MPE) method [4], Best Points Interpolation Method (BPIM) [35], and the Gauss-Newton with Approximated Tensors (GNAT) method [13, 14]. Alternatively we can employ *direct integration*, where we approximate the integral using a sparse quadrature rule while evaluating the nonlinear function exactly at the sparse set of quadrature points. Examples of direction integration methods include the Hyperreduction Method [40], Optimal Cubature [3], the Energy-Conserving Sampling and Weighting (ECSW) method [18, 19], Empirical Cubature [26], and the Empirical Quadrature Procedure (EQP) [37]. This work extends the existing EQP framework so that it applies to unsteady PDEs in the goal-oriented context. We will briefly outline EIM, the GNAT, ECSW, and EQP before moving onto a more detailed discussion of EQP in chapter 3.

2.3.6 Empirical Interpolation Method

The empirical interpolation method (EIM) is an interpolation-then-integration method, while the discrete empirical interpolation method (DEIM) is a popular discrete version of EIM [15]. In EIM, users construct a collateral RB expansion $\eta_M(w, v; \mu, t; x) \approx \eta(w; \mu, t; x)$

$$\eta_M(w; \mu, t; x) = \sum_{m=1}^{M} \beta_m(\mu) q_m(x),$$

and an associated sparse interpolation system

$$\sum_{j=1}^{M} B_{ij}^M \beta_j(\mu) = \eta(w; \mu, t; p_i) \quad 1 \leq i \leq M,$$

where $B_{ij}^M = q_j(p_i)$, $1 \leq i, j \leq M$. The online cost of a Newton iteration is independent of $\mathcal{N}_h$ because we only need to evaluate our nonlinear function and its Jacobian at our M interpolation points $\{p_i\}_{i=1}^M$. The word "collateral" is used to distinguish between the RB expansion $u_N(\mu, t)$ for $u(\mu, t)$ and the collateral RB expansion $\eta_M(w; \mu, t; x)$ for $\eta(w; \mu, t; x)$. We perform two distinct tasks to train EIM in the offline stage. First we construct the collateral RB space $W_M^g = \text{span}\{q_1(x), \ldots, q_M(x)\}$, and second we identify our sparse set of interpolation points $\{p_i\}_{i=1}^M$. EIM was first presented in [5], while

additional details on how to implement EIM for time-dependent problems are presented in [23]. Furthermore, a greedy training algorithm that relies upon upper error bounds was developed for time-dependent nonlinear parabolic PDEs and was presented in [22].

EIM controls the error in the interpolation of $\eta_M(\cdot;\cdot;\cdot)$ and provides an online-efficient *a posteriori* error estimate for the error in $\eta_M(\cdot;\cdot;\cdot)$. Although EIM allows us to solve for the field variable $u(\mu,t)$ with a computational cost that is fully decoupled from $\mathcal{N}_h$, it does not allow us to directly control the output error. Moreover, EIM does not allow us to estimate the output error in an online-efficient manner, even though it is equipped with an online-efficient error estimate for $\eta_M(\cdot;\cdot;\cdot)$. The goal-oriented EQP framework, which we will develop for time-dependent problems in chapter 4, will allow us to directly control the output error and to estimate this error in an online-efficient manner.

2.3.7 Gauss-Newton with Approximated Tensors

The Gauss-Newton with approximated tensors (GNAT) method is an interpolation-then-integration method designed to operate at the fully discrete level (i.e., following discretization in both space and time) [13, 14]. The GNAT method approximates the tensors $r(w,\phi_i;\mu,t)$, $i=1,2,\ldots,\mathcal{N}_h$ (i.e., the FE residual defined in (2.7)) and $r'(w;\phi_j,\phi_i;\mu,t)$, $i,j=1,2,\ldots,\mathcal{N}_h$ (i.e., the FE Jacobian linearized about the state w). This section will present a brief overview of the GNAT method, while details on the implementation are presented in [13, 14].

First we collect solution snapshots using the FE solver which we compress into the RB space $\mathcal{V}_N \in \mathbb{R}^{\mathcal{N}_h \times N}$ by performing POD on the snapshots $\{u_k(\mu) - u_{k-1}(\mu) \mid k = 1,2,\ldots,K, \mu \in \mathcal{D}\}$. Then we solve the minimum residual problem, which we presented in section 2.3.2, to collect RB solutions and snapshots of our residual (we can choose between four different snapshot collection methods for the residual, as presented in [13, 14]). At this point the cost of evaluating our tensors still depends on $\mathcal{N}_h$ so we must introduce additional approximations. We decouple the cost of evaluating these tensors from $\mathcal{N}_h$ by only computing their exact values for a small subset of carefully selected rows, while we approximate the values of the remaining rows using gappy POD.

Error bounds for GNAT are developed in [14] and they apply to any semi-discrete ODE that has been discretized in time using the backward-Euler method. The error bounds allow users to bound the error in the approximation of the state $u(\mu,t)$ at time t. These error bounds may not always be computable because they require a Lipschitz constant to be computed or estimated. Moreover, it will not always be possible to time-march using the backward-Euler method, so we need an error estimate that applies to

any time-marching scheme.

Although we can use GNAT to solve for the field variable in an online-efficient manner, GNAT does not allow users to control the output error, nor does it allow users to estimate the output error in online-efficient manner. GNAT also requires that its ROMs use the same time-marching method as the FE solver because the GNAT method operates at the fully-discrete level. The framework we present in chapter 4 allows for explicit control of the output error and online-efficient output error estimation. The framework is also continuous-in-time so that the ROM need not use the same time-marching scheme as the FE solver.

2.3.8 Energy-Conserving Sampling and Weighting

The energy-conserving sampling and weighting (ECSW) method is a physics-based hyperreduction procedure based on mesh sampling and the principle of virtual work [18, 19]. A shortcoming of many existing reduced modelling methods is that the ROMs they provide lack numerical stability, ECSW focuses on providing ROMs that are numerically stable. ECSW is derived in the context of structural dynamics models. The residual consists of a sum of forces, while the RB space $\mathcal{V}_N$ leads to virtual displacement. The force coupled with the virtual displacement leads to virtual work and ECSW seeks to find a ROM that preserves the work performed by the force in the FE solver. In section 2.3.9 we present the empirical quadrature procedure, which is similar to ECSW in the sense that both methods introduce constraints based on the residual and seek a ROM based on these constraints. ECSW does not directly control the output error and it does not provide online-efficient error estimates.

2.3.9 Empirical Quadrature Procedure

The empirical quadrature procedure (EQP) was first presented in [37]. EQP is a direct integration method that seeks approximate the "truth" quadrature rule using a sparse empirical quadrature rule. We will present EQP in the context of RB methods, where it is used as a hyperreduction method to decouple the computational cost of the online stage from $\mathcal{N}_h$. However, EQP can also be used in many other cases where users wish to approximate an integral empirically (such as performing a Fourier transform) [37]. Work on EQP in the context of RB approximation using the continuous Galerkin method was presented in [49], while this framework was extended to discontinuous Galerkin methods in [47]. A goal-oriented version of EQP for steady problems was also presented in [48].

In the context of RB approximation, EQP reduces the computational cost of numerical

integration by using empirical quadrature rule with fewer quadrature points than the "truth" quadrature rule. We recall the M^h-point "truth" quadrature rule $\{\xi_\kappa^h, \rho_\kappa^h\}_{\kappa=1}^{M^h}$ and the residual form $r_h : \mathcal{V}_h \times \mathcal{V}_h \times \mathcal{D} \times I \to \mathbb{R}$

$$r_h(w, v; \mu, t) = \sum_{\kappa=1}^{M^h} \rho_\kappa^h r_\kappa(w, v; \mu, t), \tag{2.30}$$

where ρ_κ^h is the quadrature weight associated with quadrature point κ, while $r_\kappa(w, v; \mu, t)$ is the residual evaluated at quadrature point κ. We note that $r_h(w, v; \mu, t) \approx r(w, v; \mu, t)$ for the residual (2.29). Next we introduce the EQP weights $\{\rho_\kappa^* \in \mathbb{R}_{\geq 0}\}_{\kappa=1}^{M^h}$, which use the same quadrature points as the "truth" quadrature rule. Many of the EQP weights are identically zero so that $M^* \ll M^h$, where $M^* = \|\rho^*\|_0$ (the l_0 norm counts the number of non-zero entries in ρ^h and all $\{\rho_\kappa^h\}_{\kappa=1}^{M^h}$ are non-zero). We use the EQP rule to approximate the "truth" quadrature rule

$$\sum_{\kappa=1}^{M^h} \rho_\kappa^h r_\kappa(w, v; \mu, t) \approx \sum_{\kappa=1}^{M^h} \rho_\kappa^* r_\kappa(w, v; \mu, t). \tag{2.31}$$

The EQP rules are derived empirically for the given integrand following a linear programing (LP) procedure, which we present in definition 1 in section 3.1 [37].

In contrast to EIM, GNAT and ECSW, EQP allows users to train the ROM to achieve a user-specified error tolerance for the training data ($\Xi_{N_{\text{train}}} \subset \mathcal{D}$). If the training set is sufficiently dense, then the ROM will also achieve the error tolerance for the test data. EIM allows users to specify the number of basis functions in the collateral RB expansion of the nonlinear residual, but it is not clear how many basis functions are needed to achieve the desired error tolerance. GNAT allows users to specify a variety of tuning parameters, but it is not clear how these tuning parameters quantitatively affect the error. ECSW allows users to train a ROM that maintains numerical stability and preserves the virtual work performed by the system, but it is unclear how this affects the output error. None of the aforementioned methods provide robust online-efficient *a posteriori* error estimators, or a way to directly control the output error. In chapter 4 we develop an EQP framework for time-dependent problems that allows users to directly control the output error. Moreover, the framework allows users to compute and estimate the output error in an online-efficient manner. In chapter 3 we present the existing EQP frameworks for steady problems in more detail, while in chapter 4 we present the theory behind the new EQP framework for time-dependent problems.

2.4 Dual-Weighted Residual Method

As mentioned in section 2.3.4, we will use the DWR method to control and estimate the output error. The DWR was first presented in [6] and it is different from many of the other *a posteriori* error analysis techniques because it deals with the error in an output functional as opposed to error in global norms. In many cases the error estimates in global norms rely on unknown stability constants. As mentioned previously, global error does not always accurately reflect the error in an output. Further, the global stability constants may not always indicate the sensitivity of the global norm to local error sources. The DWR method replaces the unknown global stability constants with computationally obtained local sensitivity factors. The DWR method relies on Galerkin orthogonality, and the *a posteriori* error estimate is computed by multiplying the residual of the solution by the local sensitivity factors. It is important that we understand how the DWR method works to understand some of the methods and techniques outlined later in this work. We will begin by explaining the DWR method in the context of a steady problem and will later extend it to apply to unsteady problems.

Given a residual $r_{\text{ss}}(u_h(\mu), v; \mu)$ and an output functional $q_{\text{ss}}(u_h(\mu); \mu)$, both associated with a steady problem, we will consider an infinitesimal perturbation $\delta r_{\text{ss}}(v)$ added to the weak form of the residual. A dual solution $z_h \in \mathcal{V}_h$, sometimes called an adjoint solution, can be defined as the sensitivity of the output to the residual perturbation by the following relationship

$$\delta q_{\text{ss}} \equiv q_{\text{ss}}(u_h(\mu) + \delta u_h(\mu); \mu) - q_{\text{ss}}(u_h(\mu); \mu) = \delta r_{\text{ss}}(z_h). \tag{2.32}$$

For convection dominated problems, a small error in an upstream location could significantly affect the downstream solution. The DWR method is useful because outputs can be highly sensitive to discretization or residual errors that are not easy to identify *a priori*. This means that the DWR method can be used to facilitate mesh adaptation by quantifying how residual errors impact the output error [20]. We will use the DWR method in conjunction with EQP to control and estimate the output error.

Throughout the rest of this work we refer to the problem associated with the DWR method as the dual problem (i.e., the problem with field variable z), while we refer to the problem of interest (i.e., the problem with field variable u) as the primal problem. For the DWR method to work, we must solve the dual problem in a space that is richer than the space in which we solve the primal problem. For instance we could choose to solve our dual problem in a higher-order space (higher p) or we could solve on a finer mesh (larger n_e). In the reduced-basis context we could solve the primal problem in the

primal RB space, while we solve the dual problem in the FE space, linearized about the RB solutions.

2.4.1 Steady DWR in an FE Space

We can use the DWR method to estimate the error in the FE output relative to the exact PDE output, $|s(\mu) - s_h(\mu)|$. First we introduce the enriched space $\mathcal{V}_{\hat{h}} \supset \mathcal{V}_h$ in which we solve the adjoint problem. We recall that we have triangulated our domain Ω into n_e non-overlapping elements such that we have the triangulation $\mathcal{T}_h \equiv \{J_i\}_{i=1}^{n_e}$ and our FE space, which consists of piecewise p-order is defined as $\mathcal{V}_h \equiv \{v \in \mathcal{V} \mid v|_{J_i} \in \mathbb{P}^p(J_i), i = 1, 2, \ldots, n_e\}$. We can either enrich this space by globally increasing the polynomial degree by one (i.e., $\mathcal{V}_{\hat{h}} \equiv \{v \in \mathcal{V} \mid v \in \mathbb{P}^{p+1}(J_i), i = 1, 2, \ldots, n_e\}$) or by increasing the number of elements (i.e., $\mathcal{V}_{\hat{h}} \equiv \{v \in \mathcal{V} \mid v \in \mathbb{P}^p(J_i), i = 1, 2, \ldots, \hat{n}_e\}$, where $\hat{n}_e > n_e$). Regardless of which method we use to create the enriched space $\mathcal{V}_{\hat{h}}$ we can introduce the dual problem: given $\mu \in \mathcal{D}$ and $u_h(\mu) \in \mathcal{V}_h$, find $z_{\hat{h}} \in \mathcal{V}_{\hat{h}}$ such that

$$r'_{\mathrm{ss}}(u_h(\mu); w, z_{\hat{h}}(\mu); \mu) = q'_{\mathrm{ss}}(u_h(\mu); w; \mu) \quad \forall w \in \mathcal{V}_{\hat{h}},$$

where $r'_{\mathrm{ss}}(u_h(\mu); w, v; \mu)$ and $q'_{\mathrm{ss}}(u_h(\mu); w; \mu)$ are the Gateaux derivatives of $r_{\mathrm{ss}}(\cdot, v; \mu)$ and $q_{\mathrm{ss}}(\cdot; \mu)$ respectively, about $u_h(\mu)$ in the direction of w. We can subsequently estimate the error in the output

$$|s_{\mathrm{ss}}(\mu) - s_{\mathrm{ss,h}}(\mu)| \approx \eta_h \equiv | - r_{\mathrm{ss}}(u_h(\mu), z_{\hat{h}}(\mu); \mu)|.$$

There are two source of error in our error estimate: (i) the use of $u_h(\mu)$ as the linearization point of the adjoint problem instead of the mean-value linearization between $u_h(\mu)$ and $u(\mu)$, and (ii) the approximation of the adjoint in $\mathcal{V}_{\hat{h}}$ instead of the infinite-dimensional $\mathcal{V}$ [6]. This method has been used successfully in many aerodynamics applications and results are presented in a review paper [20].

2.5 Summary

In this chapter we presented projection-based model reduction methods in the context of the model problem presented in [23] (section 2.1). In particular, we outlined RB approximation and how to construct RB spaces using both POD and greedy approaches (sections 2.3.1 and 2.3.2). We also explained why hyperreduction is necessary for nonlinear PDEs when we discussed offline-online decomposition in section 2.3.3, while we

outlined four of the existing hyperreduction methods: EIM, GNAT, ECSW, and EQP following our discussion of hyperreduction in section 2.3.5. In section 2.3.4 we explained why error estimates are necessary to ensure that ROMs are reliable and how they can used in the context of greedy algorithms to sample larger training sets ($\Xi_{N_{\text{train}}} \subset \mathcal{D}$). Finally in section 2.4 we outlined the DWR method which we will use in chapter 4 to develop a goal-oriented hyperreduction method for time-dependent problems using EQP.

Chapter 3

Steady Empirical Quadrature Procedure

In section 2.1 we described projection-based model reduction and explained why hyperreduction is necessary for problems with general nonlinearities. We explained that we wish to develop a new hyperreduction framework using EQP (section 2.3.9) and the DWR method (section 2.4). We will use EQP to enable hyperreduction, while we will use the DWR method to control and estimate the output error. In this chapter we will describe the existing EQP frameworks for steady problems in more detail. In particular, we are interested in understanding goal-oriented EQP (section 3.3) so that we can extend this framework to apply to time-dependent problems in chapter 4. In this chapter we will also provide a new approach to steady goal-oriented EQP in section 3.5 which is important to developing the theory for time-dependent goal-oriented EQP.

Throughout our discussion of EQP we will denote EQP approximations of a quantity $(\cdot)$ by $\tilde{(\cdot)}$. If $(\cdot)$ is an operator, such as a residual, then $\tilde{(\cdot)}$ is the EQP approximation to this operator. If $(\cdot)$ is a variable, such as a field variable, then $\tilde{(\cdot)}$ is the EQP approximation to this variable obtained using the EQP operators. Quantities associated with the primal problem are denoted by $(\cdot)^{\mathrm{pr}}$, while quantities associated with the dual problem are denoted by $(\cdot)^{\mathrm{du}}$. We refer readers to the page on nomenclature immediately before the introduction for a reminder of what each symbol and variable means.

3.1 EQP for Residual Control

Here we will describe the general form of EQP which we can use to control the error in our residual evaluation. We introduce the steady residual form $r_{h,\mathrm{ss}} : \mathcal{V}_h \times \mathcal{V}_h \times \mathcal{D} \to \mathcal{V}_h$

$$r_{h,\mathrm{ss}}(w, v; \mu) \equiv \sum_{\kappa=1}^{M^h} \rho_\kappa^h r_{\mathrm{ss},\kappa}(w, v; \mu),$$

associated with the "truth" quadrature weights $\{\rho_\kappa^h \in \mathbb{R}_{\geq 0}\}_{\kappa=1}^{M^h}$, where $r_{\mathrm{ss},\kappa}(\cdot,\cdot;\cdot)$ is similar in form to (2.10). We state the FE problem: given $\mu \in \mathcal{D}$ find $u_h(\mu) \in \mathcal{V}_h$ such that $r_{h,\mathrm{ss}}(u_h(\mu), v; \mu) = 0 \ \forall v \in \mathcal{V}_h$. We solve the FE problem for all $\mu \in \Xi_{N_{\mathrm{train}}}$ and we use these snapshots to construct the $\mathcal{V}$-orthonormal basis $\{\phi_i^{\mathrm{pr}}\}_{i=1}^N$ of $\mathcal{V}_N^{\mathrm{pr}}$. Then we introduce the steady algebraic RB residual $\mathbf{r}_{N,\mathrm{ss}}^{\mathrm{pr}} : \mathbb{R}^N \times \mathcal{D} \to \mathbb{R}^N$ and its associated Jacobian $\mathbf{J}_{N,\mathrm{ss}}^{\mathrm{pr}} : \mathbb{R}^N \times \mathcal{D} \to \mathbb{R}^{N\times N}$

$$\begin{aligned}
\mathbf{r}_{N,\mathrm{ss}}^{\mathrm{pr}}(\mathbf{w}; \mu)_i \equiv & r_{h,\mathrm{ss}}(V_N^{\mathrm{pr}}\mathbf{w}, \phi_i^{\mathrm{pr}}; \mu), \quad i = 1, 2, \ldots, N, \\
\mathbf{J}_{N,\mathrm{ss}}^{\mathrm{pr}}(\mathbf{w}; \mu)_{ij} \equiv & \frac{\partial \mathbf{r}_{N,\mathrm{ss,i}}^{\mathrm{pr}}}{\partial \mathbf{w}_j}(\mathbf{w}; \mu, t), \quad i, j = 1, 2, \ldots, N,
\end{aligned}$$

where $w = V_N^{\mathrm{pr}}\mathbf{w}$. Using Galerkin projection onto the primal RB space ($\mathcal{V}_N^{\mathrm{pr}}$) we solve the RB problem using the "truth" quadrature rule: given $\mu \in \mathcal{D}$ find $\mathbf{u}_N(\mu) \in \mathcal{V}_N^{\mathrm{pr}}$, such that $\mathbf{r}_{N,\mathrm{ss}}^{\mathrm{pr}}(\mathbf{u}_N(\mu); \mu) = 0$. We will decouple the cost of the online stage from $\mathcal{N}_h$ by finding a sparse EQP rule. First we specify (i) an accuracy parameter $\delta \in \mathbb{R}_{\geq 0}$, (ii) a parameter training set $\Xi_{N_{\mathrm{train}}}$, and (iii) the associated solution snapshots $U_{N_{\mathrm{train}}} \equiv \{u_N(\mu) \in \mathcal{V}_N^{\mathrm{pr}}\}_{\mu\in\Xi_{N_{\mathrm{train}}}}$.

The "truth" quadrature weights are given by $\{\rho_\kappa^h \in \mathbb{R}_{\geq 0}\}_{\kappa=1}^{M^h}$ and its associated computational cost is characterized by $M^h = \|\rho^h\|_0$ (we recall that the l_0 norm counts the number of non-zero entries in ρ^h and all $\{\rho_\kappa^h\}_{\kappa=1}^{M^h}$ are non-zero), while the EQP weights are given by $\{\rho_\kappa^* \in \mathbb{R}_{\geq 0}\}_{\kappa=1}^{M^h}$, where $M^* \equiv \|\rho^*\|_0$. If a quadrature weight is identically zero, then we do not need to evaluate the residual at this quadrature point. If many quadrature weights are identically zero, then the cost to evaluate the residual is reduced and we have a sparse EQP rule where $M^* \ll M^h$. We replace the "truth" quadrature rule with the EQP rule to write the EQP residual form $\tilde{r}_{h,\mathrm{ss}} : \mathcal{V}_h \times \mathcal{V}_h \times \mathcal{D} \to \mathcal{V}_h$

$$\tilde{r}_{h,\mathrm{ss}}(w, v; \mu) \equiv \sum_{\kappa=1}^{M^h} \rho_\kappa^* r_{\mathrm{ss},\kappa}(w, v; \mu),$$

which allows us to write the algebraic RB-EQP residual $\tilde{\mathbf{r}}^{\mathrm{pr}}_{N,\mathrm{ss}} : \mathbb{R}^N \times \mathcal{D} \to \mathbb{R}^N$

$$\begin{aligned}\tilde{\mathbf{r}}^{\mathrm{pr}}_{N,\mathrm{ss}}(\mathbf{w};\mu) &\equiv \tilde{r}_{h,\mathrm{ss}}(V_N^{\mathrm{pr}}\mathbf{w}, \phi_i^{\mathrm{pr}};\mu) \quad i = 1, 2, \ldots, N, \\ &= \sum_{\kappa=1}^{M^h} \rho_\kappa^* \mathbf{r}^{\mathrm{pr}}_{N,\mathrm{ss},\kappa}(V_N^{\mathrm{pr}}\mathbf{w};\mu),\end{aligned}$$

where ρ_κ is the quadrature weight and $\mathbf{r}^{\mathrm{pr}}_{N,\mathrm{ss},\kappa}(\mathbf{w};\mu) = r_{\mathrm{ss},\kappa}(V_N^{\mathrm{pr}}\mathbf{w}, v;\mu)$ for $v \in \mathcal{V}_N^{\mathrm{pr}}$.

We now search for a sparse and accurate EQP rule. An EQP rule is sparse if $M^* \ll M^h$, while it is accurate if $|\mathbf{r}^{\mathrm{pr}}_{N,\mathrm{ss}}(\mathbf{u}_N(\mu);\mu) - \tilde{\mathbf{r}}^{\mathrm{pr}}_{N,\mathrm{ss}}(\mathbf{u}_N(\mu);\mu)| \leq \epsilon$, $\forall \mu \in \mathcal{D}$. We wish to minimize $\|\rho^*\|_0$, but we cannot consider an l_0 optimization procedure in the 'zero-norm' (i.e. $\rho^*_{\mathrm{opt}} = \arg\min_{\rho \in \mathbb{R}^{M^h}} \|\rho\|_0$) because this optimization problem is NP-hard [16, 34]. Instead we choose to solve this optimization problem in the weaker l_1 norm (i.e. $\rho^* = \arg\min_{\rho \in \mathbb{R}^{M^h}} \|\rho\|_1$) which minimizes the sum of the quadrature weights $\sum_{\kappa=1}^{M^h} \rho^*_\kappa$. Alternatively the method proposed in [19] solves a similar optimization problem in the l_2 norm, which is weaker than the l_1 norm and minimizes the sum of the squares of the quadrature weights $\sum_{\kappa=1}^{M^h} (\rho^*_\kappa)^2$. Here we define the linear programming (LP) procedure.

Definition 1. (LP procedure, $\mathrm{LP}_{\mathrm{EQP}}$) We define our LP procedure $\mathrm{LP}_{\mathrm{EQP}}$ as follows: find a basic feasible solution $\rho^* \in \mathbb{R}^{M^h}$ such that

$$\rho^* = \arg \min_{\rho \in \mathbb{R}^{M^h}} \sum_{\kappa=1}^{M^h} \rho_\kappa$$

subject to M^h non-negativity constraints

$$\{\rho^*_\kappa \geq 0\}_{\kappa=1}^{M^h}, \tag{3.1}$$

a constant-function accuracy constraint

$$\Big||\Omega| - \sum_{\kappa=1}^{M^h} \rho_\kappa\Big| < \delta, \tag{3.2}$$

and $N \times N_{\mathrm{train}}$ function accuracy constraints

$$\|\mathbf{r}^{\mathrm{pr}}_{N,\mathrm{ss}}(\mathbf{u}(\mu);\mu) - \sum_{\kappa=1}^{M^h} \rho_\kappa \mathbf{r}^{\mathrm{pr}}_{N,\mathrm{ss},\kappa}(\mathbf{u}(\mu);\mu)\|_\infty \leq \delta \quad \forall \mu \in \Xi_{N_{\mathrm{train}}}, \tag{3.3}$$

that are associated with the parametric manifold (manifold accuracy constraints). We refer to (3.3) as the generic manifold accuracy constraints, and we note that they control

the error in the residual evaluation. Users can change the manifold accuracy constraints to change which error they control (i.e., error in the residual evaluation, error in the output evaluation, etc.).

The size of the LP scales with the number of quadrature points in the full-order model (M^h), the number of basis functions in the RB space (N), and the size of the training set (N_{train}) where $N \leq N_{\text{train}}$. We note that LP_{EQP} is feasible because the "truth" quadrature rule satisfies all of the constraints. The constant function accuracy constraint (3.2) improves the robustness of EQP because values of the integrand might cancel with each other in the domain Ω. Moreover, without the constant function accuracy constraint (3.2), EQP produces a trivial quadrature rule when $\mathbf{r}^{\text{pr}}_{N,\text{ss}}(\mathbf{w};\mu,t) = 0$.

A major advantage of EQP is its precise and accurate error control: EQP can be used to control the global solution error [49], or to control the output error [48], depending on which constraints users choose to impose. Although we train our EQP rule to be accurate for $\mu \in \Xi_{N_{\text{train}}}$, if our training parameters are dense enough over $\mathcal{D}$, then our EQP rule should be predictive for other testing parameters $\Xi_{N_{\text{test}}} \in \mathcal{D}$. Moreover, for a sufficiently refined $\mathcal{N}_h$, increasing $\mathcal{N}_h$ should not increase M^* because the sparseness of the EQP rule should only depend on the parametric manifold.

3.2 EQP for Global Error Control

We can use EQP to control the global solution error (measured in the $\mathcal{V}$ norm) for steady problems [49]. Throughout this thesis, we refer to a constraint that can be written as a linear constraint as "LP-admissible." We introduce the LP procedure $\text{LP}^{\nu}_{\text{ss}}$ to control the global solution error.

Definition 2. ($\text{LP}^{\nu}_{\text{ss}}$) The LP procedure $\text{LP}^{\nu}_{\text{ss}}$ controls the global solution error due to the approximation of $\mathbf{r}^{\text{pr}}_{N,\text{ss}}$ by $\tilde{\mathbf{r}}^{\text{pr}}_{N,\text{ss}}$. We introduce the primal solution snapshots $U_{N_{\text{train}}} \equiv \{\mathbf{u}_N(\mu)\}_{\mu\in\Xi_{N_{\text{train}}}}$ and we define the LP procedure $\text{LP}^{\nu}_{\text{ss}}(\Xi_{N_{\text{train}}}, U_{N_{\text{train}}}, \delta_\nu)$. We replace the generic manifold accuracy constraints in Definition 1 from section 3.1 with the LP-admissible constraints

$$\|(\mathbf{J}^{\text{pr}}_{N,\text{ss}}(\mathbf{u}_N(\mu);\mu))^{-1}(\mathbf{r}^{\text{pr}}_{N,\text{ss}}(\mathbf{u}_N(\mu);\mu) - \sum_{\kappa=1}^{M^h} \rho_\kappa \mathbf{r}^{\text{pr}}_{N,\text{ss},\kappa}(\mathbf{u}_N(\mu);\mu))\|_\infty \leq \delta_\nu, \tag{3.4}$$

for all $\mu \in \Xi_{N_{\text{train}}}$, which directly control the global solution error.

We state the proposition on the global error that is proved in [49].

Proposition 3. *We first fix $\mu \in \mathcal{D}$. We then introduce $\hat{\mathbf{u}}_N(\mu) \in \mathbb{R}^N$ such that*

$$\|\mathbf{u}_N(\mu) - \hat{\mathbf{u}}_N(\mu)\|_2 \leq \epsilon^{\text{train}}$$

for some $\epsilon^{\text{train}} \in \mathbb{R}_{\geq 0}$ and such that $\mathbf{J}^{\text{pr}}_{N,\text{ss}}(\hat{\mathbf{u}}_N(\mu))$ is non-singular. Suppose the EQP-approximated residual form and Jacobian satisfy

$$\|\mathbf{J}^{\text{pr}}_{N,\text{ss}}(\hat{\mathbf{u}}_N(\mu);\mu)^{-1}\tilde{\mathbf{r}}^{\text{pr}}_{N,\text{ss}}(\hat{\mathbf{u}}_N(\mu);\mu)\|_\infty \leq \delta_\nu,$$
$$\|\mathbf{J}^{\text{pr}}_{N,\text{ss}}(\hat{\mathbf{u}}_N(\mu);\mu)^{-1}\tilde{\mathbf{J}}^{\text{pr}}_{N,\text{ss}}(\hat{\mathbf{u}}_N(\mu);\mu) - I\|_{\max} \leq \delta_J,$$

for some $\delta_r \in \mathbb{R}_{\geq 0}$ and $\delta_J \in [0, 1/N]$. Note that $\|A\|_{\max} \equiv \max_{i,j\in\{1,2,\ldots,N\}} |A_{ij}|$ for any $A \in \mathbb{R}^{N\times N}$. Then

$$\|\mathbf{u}_N(\mu) - \tilde{\mathbf{u}}_N\|_2 \leq \frac{2N^{1/2}\delta_r}{1 - N\delta_J} + \epsilon^{\text{train}} \tag{3.5}$$

Proof. The proof relies on the Brezzi-Rappaz-Raviart theorem; see [49]. □

3.3 Goal-Oriented EQP

In many cases engineers wish to evaluate an output, so we will consider the goal-oriented EQP framework presented in [48]. This framework allows users to evaluate the output and estimate the output error using the DWR method (presented in section 2.4) in an online-efficient manner [6]. It also allows for direct control of the output error. We will present all of the propositions, but we refer readers to [48] for proofs.

First we introduce the residual EQP weights $\{\rho^r_\kappa \in \mathbb{R}_{\geq 0}\}_{\kappa=1}^{M^h}$ and the residual form $\tilde{r}^{(r)}_{h,\text{ss}}(w, v; \mu) \equiv \sum_{\kappa=1}^{M^h} \rho^r_\kappa r_{\text{ss},\kappa}(w, v; \mu)$, which allows us to write the RB-EQP residual $\tilde{\mathbf{r}}^{\text{pr}}_{N,\text{ss}} : \mathbb{R}^N \times \mathcal{D} \to \mathbb{R}^N$

$$\tilde{\mathbf{r}}^{\text{pr}}_{N,\text{ss}}(\mathbf{w};\mu)_i \equiv \tilde{r}^{(r)}_{h,\text{ss}}(V^{\text{pr}}_N\mathbf{w}, \phi^{\text{pr}}_i;\mu) \quad i = 1, 2, \ldots, N,$$

so that we can state the primal RB-EQP problem: given $\mu \in \mathcal{D}$ find $\tilde{\mathbf{u}}_N(\mu) \in \mathcal{V}^{\text{pr}}_N$ such that $\tilde{\mathbf{r}}^{\text{pr}}_{N,\text{ss}}(\tilde{\mathbf{u}}_N(\mu);\mu) = 0$.

We introduce the output functional $q_{h,\text{ss}}(w;\mu) \equiv \sum_{\kappa=1}^{M^h} \rho^h_\kappa q_{\text{ss},\kappa}(w;\mu)$, which allows us to write the algebraic RB output functional $\mathbf{q}_{N,\text{ss}}(\mathbf{w};\mu) \equiv q_{h,\text{ss}}(V^{\text{pr}}_N\mathbf{w};\mu)$. We also introduce the output-functional EQP weights $\{\rho^q_\kappa \in \mathbb{R}_{\geq 0}\}_{\kappa=1}^{M^h}$ and the EQP output functional $\tilde{q}^{(q)}_{h,\text{ss}}(w;\mu) \equiv \sum_{\kappa=1}^{M^h} \rho^q_\kappa q_{\text{ss},\kappa}(w;\mu)$, which allows us to write the algebraic RB-EQP output functional $\tilde{\mathbf{q}}_{N,\text{ss}}(\mathbf{w};\mu) \equiv \tilde{q}^{(q)}_{h,\text{ss}}(V^{\text{pr}}_N\mathbf{w};\mu)$. Then we evaluate the outputs $s_{N,\text{ss}}(\mu) \equiv \mathbf{q}_{N,\text{ss}}(\mathbf{u}_N(\mu);\mu)$ and $\tilde{s}_{N,\text{ss}}(\mu) \equiv \tilde{\mathbf{q}}_{N,\text{ss}}(\tilde{\mathbf{u}}_N(\mu);\mu)$.

We wish to control the output error and we invoke the triangle inequality to write

$$|s_{N,\mathrm{ss}}(\mu) - \tilde{s}_{N,\mathrm{ss}}(\mu)| = |\mathbf{q}_{N,\mathrm{ss}}(\mathbf{u}_N(\mu);\mu) - \tilde{\mathbf{q}}_{N,\mathrm{ss}}(\tilde{\mathbf{u}}_N(\mu);\mu)| \\ \leq |\mathbf{q}_{N,\mathrm{ss}}(\mathbf{u}_N(\mu);\mu) - \mathbf{q}_{N,\mathrm{ss}}(\tilde{\mathbf{u}}_N(\mu);\mu| + |\mathbf{q}_{N,\mathrm{ss}}(\tilde{\mathbf{u}}_N(\mu);\mu) - \tilde{\mathbf{q}}_{N,\mathrm{ss}}(\tilde{\mathbf{u}}_N(\mu);\mu)|. \quad (3.6)$$

There are two sources of error in (3.6): (i) the error in the approximation of the solution $\mathbf{u}_N(\mu)$, and (ii) the error in the approximation of the functional $\mathbf{q}_{N,\mathrm{ss}}(\cdot;\mu)$. We will handle these two sources of error separately, starting with source (i).

The solution $\mathbf{u}_N(\mu)$ is the root of the RB residual $\mathbf{r}^{\mathrm{pr}}_{N,\mathrm{ss}}(\cdot;\mu)$, so error source (i) arises from the error in the approximation of the RB residual. We will control this error using the DWR method, approximated in the primal RB space. We introduce the Jacobian of the RB residual $\mathbf{J}^{\mathrm{pr}}_{N,\mathrm{ss}} : \mathbb{R}^N \times \mathcal{D} \to \mathbb{R}^{N\times N}$ and the gradient of the RB output functional $\mathbf{g}^{\mathrm{pr}}_{N,\mathrm{ss}} : \mathbb{R}^N \times \mathcal{D} \to \mathbb{R}^N$.

$$\mathbf{J}^{\mathrm{pr}}_{N,\mathrm{ss}}(\mathbf{w};\mu)_{ij} \equiv r'_{h,\mathrm{ss}}(V^{\mathrm{pr}}_N\mathbf{w};\phi^{\mathrm{pr}}_j,\phi^{\mathrm{pr}}_i), \quad i,j = 1,2,\ldots,N,$$
$$\mathbf{g}^{\mathrm{pr}}_{N,\mathrm{ss}}(\mathbf{w};\mu)_i \equiv q'_{h,\mathrm{ss}}(V^{\mathrm{pr}}_N\mathbf{w};\phi^{\mathrm{pr}}_i), \quad i = 1,2,\ldots,N.$$

We state the dual problem: given $\mu \in \mathcal{D}$ and $\hat{\mathbf{u}}_N(\mu) \in \mathbb{R}^N$, find $\mathbf{z}^{\mathrm{pr}}_N(\mu) \in \mathbb{R}^N$ such that

$$\mathbf{J}^{\mathrm{pr}}_{N,\mathrm{ss}}(\hat{\mathbf{u}}_N(\mu);\mu)^T\mathbf{z}^{\mathrm{pr}}_N(\mu) = \mathbf{g}^{\mathrm{pr}}_{N,\mathrm{ss}}(\hat{\mathbf{u}}_N(\mu);\mu) \quad \text{in} \quad \mathbb{R}^N.$$

Next we introduce a modified adjoint

$$\mathbf{z}^{\mathrm{pr,w}}_N(\mu)_i = \max\{|\mathbf{z}^{\mathrm{pr}}_N(\mu)|_i, z^{\mathrm{pr,min}}(\mu)\}, \quad i = 1,2,\ldots,N \quad (3.7)$$

where $z^{\mathrm{pr,min}}(\mu) \equiv N^{1/2}\sqrt{\delta^r}\|\mathbf{z}^{\mathrm{pr}}_N(\mu)\|_2$. We now introduce the LP procedure $\mathrm{LP}^r_{\mathrm{ss}}$ to control output error due to the error in the approximation of $\mathbf{r}^{\mathrm{pr}}_{N,\mathrm{ss}}$.

Definition 4. ($\mathrm{LP}^r_{\mathrm{ss}}$) The LP procedure $\mathrm{LP}^r_{\mathrm{ss}}$ identifies EQP weights $\{\rho^r_\kappa \in \mathbb{R}_{\geq 0}\}_{\kappa=1}^{M^h}$ and controls the output error due to the error in the approximation of $\mathbf{r}^{\mathrm{pr}}_{N,\mathrm{ss}}$. We introduce the primal solutions snapshots $\hat{U}_{N_{\mathrm{train}}} \equiv \{\hat{\mathbf{u}}_N(\mu)\}_{\mu\in\Xi_{N_{\mathrm{train}}}}$ and we define the LP procedure $\mathrm{LP}^r_{\mathrm{ss}}(\Xi_{N_{\mathrm{train}}}, \hat{U}_{N_{\mathrm{train}}}, \delta_r)$. We replace the generic manifold accuracy constraints in Definition 1 from section 3.1 with the LP-admissible constraints

$$\|\mathbf{z}^{\mathrm{pr,w}}_N(\mu) \circ (\mathbf{r}^{\mathrm{pr}}_{N,\mathrm{ss}}(\hat{\mathbf{u}}_N(\mu);\mu) - \sum_{\kappa=1}^{M^h}\rho^r_\kappa\mathbf{r}^{\mathrm{pr}}_{N,\mathrm{ss},\kappa}(\hat{\mathbf{u}}_N(\mu);\mu))\|_\infty \leq \frac{2\delta_r}{3N} \quad \forall\mu \in \Xi_{N_{\mathrm{train}}}, \quad (3.8)$$

where $\circ$ denotes the Hadamard (i.e., element-wise) product. Given vectors $\mathbf{w} \in \mathbb{R}^m$ and

$\mathbf{v} \in \mathbb{R}^m$, their dot product is given by $\mathbf{w} \cdot \mathbf{v} \equiv \sum_{i=1}^m \mathbf{w}_i \mathbf{v}_i \in \mathbb{R}^1$, and their Hadamard product is $\mathbf{w} \circ \mathbf{v} \in \mathbb{R}^m$ whose i-th entry is $\mathbf{w}_i \mathbf{v}_i$.

We provide a proposition from [48] to relate error source (i) of (3.6) to constraint (3.8).

Proposition 5. *(Output error due to EQP approximation of $\mathbf{r}^{\text{pr}}_{N,\text{ss}}(\cdot;\cdot)$). Suppose*

$$\begin{aligned}\|\mathbf{z}_N^{\text{pr,w}} \circ (\mathbf{r}^{\text{pr}}_{N,\text{ss}}(\hat{\mathbf{u}}_N(\mu);\mu) - \tilde{\mathbf{r}}_{N,\text{ss}}(\hat{\mathbf{u}}_N(\mu);\mu))\|_\infty \leq & \frac{2\delta_r}{3N}, \\ \|\mathbf{I} - \mathbf{J}^{\text{pr}}_{N,\text{ss}}(\hat{\mathbf{u}}_N(\mu);\mu)\tilde{\mathbf{J}}^{\text{pr}}_{N,\text{ss}}(\hat{\mathbf{u}}_N(\mu);\mu)^{-1}\|_{\max} \leq & \delta_J,\end{aligned}$$

where $\|A\|_{\max}$ denotes the maximum entry of the matrix $A \in \mathbb{R}^{N\times N}$. Then

$$|\mathbf{q}_{N,\text{ss}}(\mathbf{u}_N(\mu);\mu) - \mathbf{q}_{N,\text{ss}}(\tilde{\mathbf{u}}_N(\mu);\mu)| \leq \delta_r + \mathcal{O}(\delta_J^2) + \mathcal{O}(\hat{\delta}^2) + \mathcal{O}(\tilde{\delta}^2),$$

where $\tilde{\delta} \equiv \|\mathbf{u}_N(\mu) - \tilde{\mathbf{u}}_N(\mu)\|_2$ and $\hat{\delta} \equiv \|\mathbf{u}_N(\mu) - \hat{\mathbf{u}}_N(\mu)\|_2$.

Proof. See [48]. □

We note that we do not directly enforce the Jacobian constraint because it would add N^2 manifold accuracy constraints per training point and we instead hope that we control the Jacobian error indirectly by controlling the residual error directly. We also note that the effect of Jacobian error on the output error is second-order (δ_J^2), as opposed to the first-order effect of the residual error (δ_r).

Next we introduce the LP procedure LP^q_{ss} to control error source (ii) of (3.6).

Definition 6. (LP^q_{ss}) The LP procedure LP^q_{ss} identifies EQP weights $\{\rho^q_\kappa \in \mathbb{R}_{\geq 0}\}_{\kappa=1}^{M^h}$ and controls the output error due to the error in the approximation of $\mathbf{q}_{N,\text{ss}}$. We introduce the primal solution snapshots $\tilde{U}_{N_{\text{train}}} \equiv \{\tilde{\mathbf{u}}_N(\mu)\}_{\mu\in\Xi_{N_{\text{train}}}}$ and we define the LP procedure $\text{LP}^q_{\text{ss}}(\Xi_{N_{\text{train}}}, \tilde{U}_{N_{\text{train}}}, \delta_q)$. We replace the generic manifold accuracy constraints in Definition 1 from section 3.1 with the LP-admissible constraints

$$|\mathbf{q}_N(\tilde{\mathbf{u}}_{N,\text{ss}}(\mu);\mu) - \sum_{\kappa=1}^{M^h} \rho^q_\kappa \mathbf{q}_{N,\text{ss},\kappa}(\tilde{\mathbf{u}}_N(\mu);\mu)| \leq \delta_q \quad \forall \mu \in \Xi_{N_{\text{train}}}.$$

3.4 DWR EQP: *A Posteriori* Error Estimation

In this section we describe how to estimate the output error in an online-efficient manner using the DWR method, as presented in [48]. We can compute the RB DWR error estimate $|\eta_N(\mu)| \approx |q_{h,\text{ss}}(u_h(\mu);\mu) - q_{h,\text{ss}}(\tilde{u}_N(\mu);\mu)|$, or the RB-EQP approximation $|\tilde{\eta}_N(\mu)|$.

First we compute dual FE solution snapshots $z \in \mathcal{V}_h$ (as described in section 2.4), linearized about the primal RB-EQP solutions $\tilde{u}_N \in \mathcal{V}_N^{\rm pr}$, to construct the $\mathcal{V}$-orthonormal basis $\{\phi_i^{\rm du}\}_{i=1}^N$ for $\mathcal{V}_N^{\rm du}$. Next we project the dual-problem onto the dual RB space, where we have the Jacobian $\mathbf{J}_{N,\rm ss}^{\rm du} : \mathbb{R}^N \times \mathcal{D} \to \mathbb{R}^{N\times N}$ and the gradient of the output $\mathbf{g}_{N,\rm ss}^{\rm du} : \mathbb{R}^N \times \mathcal{D} \to \mathbb{R}^N$

$$\begin{aligned}\mathbf{J}_{N,\rm ss}^{\rm du}(\mathbf{w};\mu)_{ij} &\equiv r'_{h,\rm ss}(V_N^{\rm pr}\mathbf{w};\phi_j^{\rm du},\phi_i^{\rm du}), \quad i,j=1,2,\ldots,N,\\ \mathbf{g}_{N,\rm ss}^{\rm du}(\mathbf{w};\mu)_i &\equiv q'_{h,\rm ss}(V_N^{\rm pr}\mathbf{w};\phi_i^{\rm du}), \quad i=1,2,\ldots,N.\end{aligned}$$

We state the dual RB problem: given $\mu \in \mathcal{D}$ and $\tilde{\mathbf{u}}_N(\mu) \in \mathbb{R}^N$, find $\mathbf{z}_N^{\rm du}(\mu) \in \mathbb{R}^N$ such that $\mathbf{J}_{N,\rm ss}^{\rm du}(\tilde{\mathbf{u}}_{N,\rm ss}(\mu);\mu)^T\mathbf{z}_N^{\rm du}(\mu) = \mathbf{g}_{N,\rm ss}^{\rm du}(\tilde{\mathbf{u}}_N(\mu);\mu)$, in $\mathbb{R}^N$. Then we introduce the primal residual projected onto the dual RB space $\mathbf{r}_{N,\rm ss}^{\rm du} : \mathbb{R}^N \times \mathcal{D} \to \mathbb{R}^{N\times N}$

$$\mathbf{r}_{N,\rm ss}^{\rm du}(\mathbf{w};\mu)_i \equiv r_{h,\rm ss}(V_N^{\rm pr}\mathbf{w};\phi_i^{\rm du}), \quad i=1,2,\ldots,N,$$

and we evaluate the RB DWR error estimate $\eta_N(\mu) \equiv \mathbf{z}_N^{\rm duT}\mathbf{r}_{N,\rm ss}^{\rm du}(\tilde{\mathbf{u}}_N(\mu);\mu)$.

We introduce the DWR EQP weights $\{\rho_\kappa^\eta \in \mathbb{R}_{\geq 0}\}_{\kappa=1}^{M^h}$. Then we introduce the EQP residual form $\tilde{r}_{h,\rm ss}^{(\eta)}(w,v;\mu) \equiv \sum_{\kappa=1}^{M^h}\rho_\kappa^\eta r_{h,\rm ss,\kappa}(w,v;\mu)$ and the EQP output form $\tilde{q}_{h,\rm ss}^{(\eta)}(w;\mu) \equiv \sum_{\kappa=1}^{M^h}\rho_\kappa^\eta q_{\rm ss,\kappa}(w;\mu)$. We use these forms to introduce the RB-EQP Jacobian and output gradient

$$\begin{aligned}\tilde{\mathbf{J}}_{N,\rm ss}^{\rm du}(\mathbf{w};\mu)_{ij} &\equiv \tilde{r}_{h,\rm ss}^{(\eta)\,\prime}(V_N^{\rm pr}\mathbf{w};\phi_j^{\rm du},\phi_i^{\rm du}), \quad i,j=1,2,\ldots,N,\\ \tilde{\mathbf{g}}_{N,\rm ss}^{\rm du}(\mathbf{w};\mu)_i &\equiv \tilde{q}_{h,\rm ss}^{(\eta)\,\prime}(V_N^{\rm pr}\mathbf{w};\phi_i^{\rm du}), \quad i=1,2,\ldots,N.\end{aligned}$$

Then we state the dual RB-EQP problem: given $\mu \in \mathcal{D}$ and $\tilde{\mathbf{u}}_N(\mu) \in \mathbb{R}^N$, find $\tilde{\mathbf{z}}_N^{\rm du}(\mu) \in \mathbb{R}^N$ such that $\tilde{\mathbf{J}}_{N,\rm ss}^{\rm du}(\tilde{\mathbf{u}}_{N,\rm ss}(\mu);\mu)^T\tilde{\mathbf{z}}_N^{\rm du}(\mu) = \tilde{\mathbf{g}}_{N,\rm ss}^{\rm du}(\tilde{\mathbf{u}}_N(\mu);\mu)$, in $\mathbb{R}^N$. We introduce the RB-EQP residual $\tilde{\mathbf{r}}_{N,\rm ss}^{\rm du} : \mathbb{R}^N \times \mathcal{D} \to \mathbb{R}^{N\times N}$

$$\tilde{\mathbf{r}}_{N,\rm ss}^{\rm du}(\mathbf{w};\mu)_i \equiv \tilde{r}_{h,\rm ss}^{(\eta)}(V_N^{\rm pr}\mathbf{w};\phi_i^{\rm du}), \quad i=1,2,\ldots,N,$$

and we evaluate the RB-EQP DWR error estimate $\tilde{\eta}_N(\mu) \equiv \tilde{\mathbf{z}}_N^{\rm duT}\tilde{\mathbf{r}}_{N,\rm ss}^{\rm du}(\tilde{\mathbf{u}}_N(\mu);\mu)$.

We wish to control the error $|\eta_N - \tilde{\eta}_N|$, which we write as

$$\begin{aligned}|\eta_N - \tilde{\eta}_N| &= |\mathbf{z}_N^{\rm duT}\mathbf{r}_{N,\rm ss}^{\rm du} - \tilde{\mathbf{z}}_N^{\rm duT}\tilde{\mathbf{r}}_{N,\rm ss}^{\rm du}|\\ &\leq |\underbrace{(\mathbf{z}_N^{\rm du} - \tilde{\mathbf{z}}_N^{\rm du})^T\mathbf{r}_{N,\rm ss}^{\rm du}}_{\text{source (i)}}| + |\underbrace{\mathbf{z}_N^{\rm duT}(\mathbf{r}_{N,\rm ss}^{\rm du} - \tilde{\mathbf{r}}_{N,\rm ss}^{\rm du})}_{\text{source (ii)}}| + |\underbrace{(\mathbf{z}_N^{\rm du} - \tilde{\mathbf{z}}_N^{\rm du})^T(\mathbf{r}_{N,\rm ss}^{\rm du} - \tilde{\mathbf{r}}_{N,\rm ss}^{\rm du})}_{\text{source (iii)}}|. \quad (3.9)\end{aligned}$$

We see that (3.9) has three source of error: (i) the error in the approximation of the dual solution $\mathbf{z}_N^{\mathrm{du}}(\cdot)$, (ii) the error in the residual evaluation $\mathbf{r}_{N,\mathrm{ss}}^{\mathrm{du}}(\cdot;\cdot)$, and (iii) the product of these two errors. We will introduce constraints to directly control error sources (i) and (ii) so that they indirectly control error source (iii).

We introduce scaling constants

$$\alpha(\mu) \equiv \frac{\|\mathbf{z}_N^{\mathrm{du}}(\mu)\|_2}{\|\mathbf{r}_{N,\mathrm{ss}}^{\mathrm{du}}(\tilde{\mathbf{u}}_N(\mu))\|_2}, \quad z^{\mathrm{du,min}}(\mu) \equiv \tfrac{1}{2}\sqrt{\tfrac{\alpha(\mu)\delta_\eta}{N}}, \quad r^{\mathrm{du,min}}(\mu) \equiv \frac{1}{2}\sqrt{\frac{\delta_\eta}{\alpha(\mu)N}},$$

and then we define

$$\begin{aligned}\mathbf{z}_N^{\mathrm{du,w}}(\mu) &= \max\{|\mathbf{z}_N^{\mathrm{du}}(\mu)|, z^{\mathrm{du,min}}(\mu)\} \\ \mathbf{r}_{N,\mathrm{ss}}^{\mathrm{du,w}}(\tilde{\mathbf{u}}_N(\mu);\mu) &= \max\{|\mathbf{r}_{N,\mathrm{ss}}^{\mathrm{du}}(\tilde{u}_N(\mu);\mu)|, r^{\mathrm{du,min}}(\mu)\},\end{aligned}$$

where the max operator between the vector and the scalar is applied to each element of the vector. Next we introduce the LP procedure $\mathrm{LP}_{\mathrm{ss}}^{\eta}$ to control the error in the DWR error estimate.

Definition 7. ($\mathrm{LP}_{\mathrm{ss}}^{\eta}$) The LP procedure $\mathrm{LP}_{\mathrm{ss}}^{\eta}$ identifies EQP weights $\{\rho_\kappa^\eta \in \mathbb{R}_{\geq 0}\}_{\kappa=1}^{M^h}$ and controls the error in the DWR error estimate. We introduce the primal and dual solution snapshots $\tilde{U}_{N_{\mathrm{train}}} \equiv \{\tilde{\mathbf{u}}_N(\mu)\}_{\mu\in\Xi_{N_{\mathrm{train}}}}$ and $Z_{N_{\mathrm{train}}} \equiv \{\mathbf{z}_N^{\mathrm{du}}(\mu)\}_{\mu\in\Xi_{N_{\mathrm{train}}}}$, then we define the LP procedure $\mathrm{LP}_{\mathrm{ss}}^{\eta}(\Xi_{N_{\mathrm{train}}}, \tilde{U}_{N_{\mathrm{train}}}, Z_{N_{\mathrm{train}}}, \delta_\eta)$. We replace the generic manifold accuracy constraints in Definition 1 from section 3.1 with constraints to control the error sources (i) and (ii) of (3.9). First we impose two sets of constraints to control error source (i) of (3.9) – the approximation of the dual solution $\mathbf{z}_N^{\mathrm{du}}(\mu)$:

$$\begin{aligned}\|\mathbf{r}_{N,\mathrm{ss}}^{\mathrm{du,w}}(\tilde{\mathbf{u}}_N(\mu);\mu) \circ \mathbf{J}_{N,\mathrm{ss}}^{\mathrm{du}}(\tilde{\mathbf{u}}_N(\mu);\mu)^{-T}(\mathbf{J}_{N,\mathrm{ss}}^{\mathrm{du}}(\tilde{\mathbf{u}}_N;\mu)^T\mathbf{z}^{\mathrm{du}}(\mu) \\ - \sum_{\kappa=1}^{M^h}\rho_\kappa^\eta \mathbf{J}_{N,\mathrm{ss},\kappa}^{\mathrm{du}}(\tilde{\mathbf{u}}_N(\mu);\mu)^T\mathbf{z}_N^{\mathrm{du}}(\mu))\|_\infty \leq \frac{\delta^\eta}{4N} \quad \forall\mu\in\Xi_{N_{\mathrm{train}}}, \quad (3.10)\end{aligned}$$

and

$$\begin{aligned}\|\mathbf{r}_{N,\mathrm{ss}}^{\mathrm{du,w}}(\tilde{\mathbf{u}}_N(\mu);\mu) \circ \mathbf{J}_{N,\mathrm{ss}}^{\mathrm{du}}(\tilde{\mathbf{u}}_N(\mu);\mu)^{-T}(\mathbf{g}_{N,\mathrm{ss}}^{\mathrm{du}}(\tilde{\mathbf{u}}_N(\mu);\mu) \\ - \sum_{\kappa=1}^{M^h}\rho_\kappa^\eta \mathbf{g}_{N,\mathrm{ss},\kappa}(\tilde{\mathbf{u}}_N(\mu);\mu))\|_\infty \leq \frac{\delta_\eta}{4N} \quad \forall\mu\in\Xi_{N_{\mathrm{train}}}. \quad (3.11)\end{aligned}$$

Next we impose constraints to control the error due to source (ii) of (3.9) – the approxi-

mation of the residual $\mathbf{r}_N^{\mathrm{du}}$:

$$\|\mathbf{z}_N^{\mathrm{du}}(\mu) \circ (\mathbf{r}_{N,\mathrm{ss}}^{\mathrm{du}}(\tilde{\mathbf{u}}_N(\mu);\mu) - \sum_{\kappa=1}^{M^h} \rho_\kappa^\eta \mathbf{r}_{N,\mathrm{ss},\kappa}^{\mathrm{du}}(\tilde{\mathbf{u}}_N(\mu);\mu))\|_\infty \leq \frac{\delta_\eta}{2N} \quad \forall \mu \in \Xi_{N_{\mathrm{train}}}. \tag{3.12}$$

We now provide propositions from [48] to relate constraints (3.10) and (3.11) to error source (i) of (3.9), and constraint (3.12) to error source (ii)

Proposition 8. *(DWR error due to approximation of $\mathbf{z}_N^{\mathrm{du}}(\mu)$). Suppose*

$$\|\mathbf{r}_{N,\mathrm{ss}}^{\mathrm{du,w}}(\tilde{\mathbf{u}}_N) \circ (\mathbf{J}_{N,\mathrm{ss}}^{\mathrm{du}}(\tilde{\mathbf{u}}_N)^{-T}(\mathbf{J}_{N,\mathrm{ss}}^{\mathrm{du}}(\tilde{\mathbf{u}}_N)^T \mathbf{z}_N^{\mathrm{du}} - \tilde{\mathbf{J}}_N^{\mathrm{du}}(\tilde{\mathbf{u}}_N)^T \mathbf{z}_N^{\mathrm{du}}))\|_\infty \leq \frac{\delta_\eta}{4N}, \tag{3.13}$$

$$\|\mathbf{r}_{N,\mathrm{ss}}^{\mathrm{du,w}}(\tilde{\mathbf{u}}_N) \circ (\mathbf{J}_{N,\mathrm{ss}}^{\mathrm{du}}(\tilde{\mathbf{u}}_N)^{-T}(\mathbf{g}_{N,\mathrm{ss}}^{\mathrm{du}}(\tilde{\mathbf{u}}_N) - \tilde{\mathbf{g}}_{N,\mathrm{ss}}^{\mathrm{du}})))\|_\infty \leq \frac{\delta_\eta}{4N}, \tag{3.14}$$

$$\|I - \tilde{\mathbf{J}}_{N,\mathrm{ss}}^{\mathrm{du}}(\hat{\mathbf{u}}_N)^{-T} \mathbf{J}_{N,\mathrm{ss}}^{\mathrm{du}}(\hat{\mathbf{u}}_N)^T\|_{\max} \leq \delta_J, \tag{3.15}$$

where $\|A\|_{\max}$ denotes the maximum entry of the matrix $A \in \mathbb{R}^{N\times N}$. Then

$$|\mathbf{r}_{N,\mathrm{ss}}^{\mathrm{du}}(\tilde{\mathbf{u}}_N)^T(\mathbf{z}_N^{\mathrm{du}} - \tilde{\mathbf{z}}_N^{\mathrm{du}})| \leq \frac{\delta_\eta}{2} + \mathcal{O}(\delta_\eta^2) + \mathcal{O}(\delta_J^2), \tag{3.16}$$

$$\|\mathbf{z}_N^{\mathrm{du}} - \tilde{\mathbf{z}}_N^{\mathrm{du}}\|_2 \leq \frac{1}{2}\sqrt{\alpha\delta_\eta} + \mathcal{O}(\delta_\eta) + \mathcal{O}(\delta_J^2). \tag{3.17}$$

Proof. See [48]. □

Proposition 9. *(DWR error due to approximation of $\mathbf{r}_{N,\mathrm{ss}}^{\mathrm{du}}(\cdot;\cdot)$). Suppose*

$$\|\mathbf{z}_N^{\mathrm{du}} \circ (\mathbf{r}_{N,\mathrm{ss}}^{\mathrm{du}}(\tilde{\mathbf{u}}_N) - \tilde{\mathbf{r}}_{N,\mathrm{ss}}^{\mathrm{du}}(\tilde{\mathbf{u}}_N))\|_\infty \leq \frac{\delta_\eta}{2N}. \tag{3.18}$$

Then

$$|\mathbf{z}_N^{\mathrm{du}T}(\mathbf{r}_{N,\mathrm{ss}}^{\mathrm{du}}(\tilde{\mathbf{u}}_N) - \tilde{\mathbf{r}}_{N,\mathrm{ss}}^{\mathrm{du}}(\tilde{\mathbf{u}}_N))| \leq \frac{\delta_\eta}{2}, \tag{3.19}$$

$$\|\mathbf{r}_{N,\mathrm{ss}}^{\mathrm{du}}(\tilde{\mathbf{u}}_N) - \tilde{\mathbf{r}}_N^{\mathrm{du}}(\tilde{\mathbf{u}}_N)\|_2 \leq \sqrt{\frac{\delta_\eta}{\alpha}}. \tag{3.20}$$

Proof. See [48]. □

Corollary 10. *The combination of propositions 8 and 9 implies that*

$$|(\mathbf{z}_N^{\mathrm{du}} - \tilde{\mathbf{z}}_N^{\mathrm{du}})^T(\mathbf{r}_{N,\mathrm{ss}}^{\mathrm{du}}(\tilde{\mathbf{u}}_N - \tilde{\mathbf{r}}_{N,\mathrm{ss}}^{\mathrm{du}}(\tilde{\mathbf{u}}_N))| \leq \|\mathbf{z}_N^{\mathrm{du}} - \tilde{\mathbf{z}}_N^{\mathrm{du}}\|_2 \|\mathbf{r}_{N,\mathrm{ss}}^{\mathrm{du}}(\tilde{\mathbf{u}}_N) - \tilde{\mathbf{r}}_{N,\mathrm{ss}}^{\mathrm{du}}(\tilde{\mathbf{u}}_N)\|_2 \leq \delta_\eta/2; \tag{3.21}$$

the highest-order error is at most $\delta_\eta/2$ (and is nominally δ_η^2).

We note that we do not directly control the Jacobian error for the same reasons that we described in section 3.3 (the Jacobian error is second-order and we hope that the control of the residual error will indirectly control the Jacobian error). We further note that by controlling error sources (i) and (ii) for the DWR error estimate, that we also indirectly control error source (iii).

3.5 DWR EQP: Alternate Form

We consider an alternate approach to DWR EQP where we apply the DWR method to the dual problem (dual-of-the-dual approach). First we introduce the tangent equation for steady problems, which is the equation that arises from the dual-of-the-dual approach. We will show how to estimate the error in the DWR error estimate, which in turn will allow us to control the error in the DWR error estimate. The dual-of-the-dual approach closely follows the goal-oriented EQP approach presented in section 3.3.

We will now present the RB tangent problem, which we solve in the dual RB space ($\mathcal{V}_N^{\mathrm{du}}$). We define the dual output as the RB DWR error estimate $\eta_N(\mu)$, while we treat the RB-EQP dual solution $\tilde{\mathbf{z}}_N^{\mathrm{du}}(\mu) \in \mathbb{R}^N$ as the primal variable (i.e., error in the approximation of $\mathbf{z}_N^{\mathrm{du}}(\mu)$ by $\tilde{\mathbf{z}}_N^{\mathrm{du}}(\mu)$ leads to error in the evaluation of $\eta_N(\mu)$). We state the tangent problem: given $\mu \in \mathcal{D}$ and $\tilde{\mathbf{u}}_N(\mu) \in \mathbb{R}^N$, find $\boldsymbol{\psi}_N^{\mathrm{du}}(\mu) \in \mathbb{R}^N$ such that

$$\mathbf{J}_{N,\mathrm{ss}}^{\mathrm{du}}(\tilde{\mathbf{u}}_N(\mu);\mu)\boldsymbol{\psi}_N^{\mathrm{du}}(\mu) - \mathbf{r}_{N,\mathrm{ss}}^{\mathrm{du}}(\tilde{\mathbf{u}}_N(\mu);\mu) = 0 \quad \text{in} \quad \mathbb{R}^N, \tag{3.22}$$

which means that $\boldsymbol{\psi}_N^{\mathrm{du}}(\mu) = \mathbf{J}_{N,\mathrm{ss}}^{\mathrm{du}}(\tilde{\mathbf{u}}_N(\mu);\mu)^{-1}\mathbf{r}_{N,\mathrm{ss}}^{\mathrm{du}}(\tilde{\mathbf{u}}_N(\mu);\mu)$. We can estimate the error in the DWR error estimate due to the error in the dual variable

$$\begin{aligned}-(\mathbf{z}_N^{\mathrm{du}} - \tilde{\mathbf{z}}_N^{\mathrm{du}})^T\mathbf{r}_{N,\mathrm{ss}}^{\mathrm{du}} \approx -\boldsymbol{\psi}_N^{\mathrm{du}T}(\mathbf{J}_{N,\mathrm{ss}}^{\mathrm{du}T}\tilde{\mathbf{z}}_N^{\mathrm{du}} + \mathbf{g}_N^{\mathrm{du}}) = \\ -(\mathbf{J}_{N,\mathrm{ss}}^{\mathrm{du}-1}\mathbf{r}_{N,\mathrm{ss}}^{\mathrm{du}})^T(\mathbf{J}_{N,\mathrm{ss}}^{\mathrm{du}T}\tilde{\mathbf{z}}_N^{\mathrm{du}} + \mathbf{g}_N^{\mathrm{du}}) = -\mathbf{r}_{N,\mathrm{ss}}^{\mathrm{du}T}\mathbf{J}_{N,\mathrm{ss}}^{\mathrm{du}-T}(\mathbf{J}_{N,\mathrm{ss}}^{\mathrm{du}T}\tilde{\mathbf{z}}_N^{\mathrm{du}} + \mathbf{g}_N^{\mathrm{du}}).\end{aligned}$$

We notice that the above expression is similar to constraints (3.10) and (3.11), which in proposition 8 from section 3.4 are used to control the error $|(\mathbf{z}_N^{\mathrm{du}} - \tilde{\mathbf{z}}_N^{\mathrm{du}})^T\mathbf{r}_{N,\mathrm{ss}}^{\mathrm{du}}|$.

We will now introduce and prove a new proposition that will allow us to control error source (i) of (3.9) using the tangent solution $\boldsymbol{\psi}_N^{\mathrm{du}}(\mu) \in \mathbb{R}^N$. We will continue to control error source (ii) using constraint (3.10) (see proposition 9). We will also split the LP procedure $\mathrm{LP}_{\mathrm{ss}}^{\eta}$ from definition 7 into two separate LP procedures. First we introduce the LP procedure $\mathrm{LP}^{\eta,2}$, associated with EQP weights $\{\rho_\kappa^{\eta,2} \in \mathbb{R}_{\geq 0}\}_{\kappa=1}^{M^h}$, to control error source (ii) of (3.9).

Definition 11. ($\text{LP}_{\text{ss}}^{\eta,2}$) The LP procedure $\text{LP}_{\text{ss}}^{\eta,2}$ identifies EQP weights $\{\rho_\kappa^{\eta,2} \in \mathbb{R}_{\geq 0}\}_{\kappa=1}^{M^h}$ and controls the error in the DWR error estimate due to the approximation of $\mathbf{r}_N^{\text{du}}(\cdot,\cdot)$ by $\tilde{\mathbf{r}}_N^{\text{du}}(\cdot,\cdot)$. We introduce the primal and dual solutions snapshots $\tilde{U}_{N_{\text{train}}} \equiv \{\tilde{\mathbf{u}}_N(\mu)\}_{\mu\in\Xi_{N_{\text{train}}}}$ and $Z_{N_{\text{train}}} \equiv \{\mathbf{z}_N^{\text{du}}(\mu)\}_{\mu\in\Xi_{N_{\text{train}}}}$, then we define the LP procedure $\text{LP}_{\text{ss}}^{\eta,2}(\Xi_{N_{\text{train}}}, \tilde{U}_{N_{\text{train}}}, Z_{N_{\text{train}}}, \delta_{\eta_2})$. We replace the generic manifold accuracy constraints in Definition 1 from section 3.1 with constraints

$$\|\mathbf{z}_N^{\text{du}}(\mu) \circ (\mathbf{r}_{N,\text{ss}}^{\text{du}}(\tilde{\mathbf{u}}_N(\mu);\mu) - \sum_{\kappa=1}^{M^h} \rho_\kappa^{\eta,2} \mathbf{r}_{N,\text{ss},\kappa}^{\text{du}}(\tilde{\mathbf{u}}_N(\mu);\mu))\|_\infty \leq \frac{\delta_{\eta_2}}{2N} \quad \forall \mu \in \Xi_{N_{\text{train}}}.$$

Next we introduce the scaling constant $\psi^{\text{du,min}}(\mu) \equiv \sqrt{\frac{\delta_{\eta_1}}{N}} \|\boldsymbol{\psi}_N^{\text{du}}(\mu)\|_2$ and we define

$$\boldsymbol{\psi}_N^{\text{du,w}}(\mu) \equiv \max\{|\boldsymbol{\psi}_N^{\text{du}}(\mu)|, \psi^{\text{du,min}}(\mu)\}.$$

We can now introduce the LP procedure $\text{LP}^{\eta,1}$, associated with EQP weights $\{\rho_\kappa^{\eta,1} \in \mathbb{R}_{\geq 0}\}_{\kappa=1}^{M^h}$, to control error source (i) of (3.9).

Definition 12. ($\text{LP}_{\text{ss}}^{\eta,1}$) The LP procedure $\text{LP}_{\text{ss}}^{\eta,1}$ identifies EQP weights $\{\rho_\kappa^{\eta,1} \in \mathbb{R}_{\geq 0}\}_{\kappa=1}^{M^h}$ and controls the error in the DWR error estimate due to the approximation of $\mathbf{z}_N^{\text{du}}$ by $\tilde{\mathbf{z}}_N^{\text{du}}$. We introduce the primal, dual, and tangent solution snapshots $\tilde{U}_{N_{\text{train}}} \equiv \{\tilde{\mathbf{u}}_N(\mu)\}_{\mu\in\Xi_{N_{\text{train}}}}$, $Z_{N_{\text{train}}} \equiv \{\mathbf{z}_N^{\text{du}}(\mu)\}_{\mu\in\Xi_{N_{\text{train}}}}$, and $\Psi_{N_{\text{train}}} \equiv \{\boldsymbol{\psi}_N^{\text{du}}(\mu)\}_{\mu\in\Xi_{N_{\text{train}}}}$. Then we define the LP procedure $\text{LP}_{\text{ss}}^{\eta,1}(\Xi_{N_{\text{train}}}, \tilde{U}_{N_{\text{train}}}, Z_{N_{\text{train}}}, \Psi_{N_{\text{train}}}, \delta_{\eta_1})$. We replace the generic manifold accuracy constraints in Definition 1 from section 3.1 with three sets of constraints $\forall \mu \in \Xi_{N_{\text{train}}}$:

$$\|\boldsymbol{\psi}_N^{\text{du,w}}(\mu) \circ (\mathbf{J}_{N,\text{ss}}^{\text{du}}(\tilde{\mathbf{u}}_N(\mu);\mu)^T \mathbf{z}_N^{\text{du}}(\mu) - \sum_{\kappa=1}^{M^h} \rho_\kappa^{\eta,1} \mathbf{J}_{N,\text{ss},\kappa}^{\text{du}}(\tilde{\mathbf{u}}_N(\mu);\mu)^T \mathbf{z}_N^{\text{du}}(\mu))\|_\infty \leq \frac{\delta_{\eta_1}}{4N}, \tag{3.23}$$

$$\|\boldsymbol{\psi}_N^{\text{du,w}}(\mu) \circ (\mathbf{g}_{N,\text{ss}}^{\text{du}}(\tilde{\mathbf{u}}_N(\mu);\mu) - \sum_{\kappa=1}^{M^h} \rho_\kappa^{\eta,1} \mathbf{g}_{N,\text{ss},\kappa}^{\text{du}}(\tilde{\mathbf{u}}_N(\mu);\mu))\|_\infty \leq \frac{\delta_{\eta_1}}{4N}. \tag{3.24}$$

$$\|\mathbf{J}_{N,\text{ss}}^{\text{du}}(\tilde{\mathbf{u}}_N(\mu);\mu)^{-T} (\sum_{\kappa=1}^{M^h} \rho_\kappa^{\eta,1} [\mathbf{J}_{N,\text{ss},\kappa}^{\text{du}}(\tilde{\mathbf{u}}_N(\mu);\mu)^T \mathbf{z}_N^{\text{du}}(\mu) + \mathbf{g}_{N,\text{ss},\kappa}^{\text{du}}(\tilde{\mathbf{u}}_N(\mu);\mu))]\|_\infty \leq \frac{1}{2}\sqrt{\frac{\delta_{\eta_1}}{N}} \tag{3.25}$$

We introduce a proposition to show that we can control error source (i) of (3.9) using the tangent solution $\boldsymbol{\psi}_N^{\text{du}}(\mu) \in \mathbb{R}^N$.

Proposition 13. *(DWR error due to approximation of $\mathbf{z}_N^{\mathrm{du}}(\cdot)$). Suppose*

$$|(\boldsymbol{\psi}_N^{\mathrm{du,w}})^T(\tilde{\mathbf{J}}_{N,\mathrm{ss}}^{\mathrm{du}}(\tilde{\mathbf{u}}_N)^T\mathbf{z}_N^{\mathrm{du}} + \tilde{\mathbf{g}}_{N,\mathrm{ss}}^{\mathrm{du}}(\tilde{\mathbf{u}}_N)| \leq \frac{\delta_{\eta_1}}{2}, \tag{3.26}$$

$$\|\mathbf{J}_{N,\mathrm{ss}}^{\mathrm{du}-T}(\tilde{\mathbf{J}}_{N,\mathrm{ss}}^{\mathrm{du}T}\mathbf{z}_N^{\mathrm{du}} + \tilde{\mathbf{g}}_{N,\mathrm{ss}}^{\mathrm{du}})\|_\infty \leq \frac{1}{2}\sqrt{\frac{\delta_{\eta_1}}{N}} \tag{3.27}$$

$$\|\mathbf{I} - \mathbf{J}_{N,\mathrm{ss}}^{\mathrm{du}}(\tilde{\mathbf{u}}_N)^T\tilde{\mathbf{J}}_{N,\mathrm{ss}}^{\mathrm{du}}(\tilde{\mathbf{u}}_N)^{-T}\|_{\max} \leq \delta_J, \tag{3.28}$$

$$\|\mathbf{I} - \tilde{\mathbf{J}}_{N,\mathrm{ss}}^{\mathrm{du}}(\tilde{\mathbf{u}}_N)^{-T}\mathbf{J}_{N,\mathrm{ss}}^{\mathrm{du}}(\tilde{\mathbf{u}}_N)^T\|_{\max} \leq \delta_J, \tag{3.29}$$

where $\|\mathbf{A}\|_{\max}$ denotes the maximum entry of matrix $\mathbf{A} \in \mathbb{R}^{N\times N}$. Then

$$|\mathbf{r}_{N,\mathrm{ss}}^{\mathrm{du}}(\tilde{\mathbf{u}}_N)^T(\mathbf{z}_N^{\mathrm{du}} - \tilde{\mathbf{z}}_N^{\mathrm{du}})| \leq \frac{\delta_\eta}{2} + \mathcal{O}(\delta_{\eta_1}^2) + \mathcal{O}(\delta_J^2), \tag{3.30}$$

$$\|\mathbf{z}_N^{\mathrm{du}} - \tilde{\mathbf{z}}_N^{\mathrm{du}}\|_2 \leq \frac{1}{2}\sqrt{\delta_{\eta_1}} + \mathcal{O}(\delta_{\eta_1}) + \mathcal{O}(\delta_J^2) \tag{3.31}$$

Proof. We drop $\tilde{\mathbf{u}}_N$ as an argument and assume that all three of $\mathbf{r}_{N,\mathrm{ss}}^{\mathrm{du}}$, $\mathbf{J}_{N,\mathrm{ss}}^{\mathrm{du}}$, and $\mathbf{g}_{N,\mathrm{ss}}^{\mathrm{du}}$ are evaluated about $\tilde{\mathbf{u}}_N$, unless specified otherwise. To prove (3.30), we recall that $\tilde{\mathbf{z}}_N^{\mathrm{du}} = -\tilde{\mathbf{J}}_{N,\mathrm{ss}}^{\mathrm{du}-T}\tilde{\mathbf{g}}_{N,\mathrm{ss}}^{\mathrm{du}}$ to get

$$\begin{aligned}\mathbf{z}_N^{\mathrm{du}} - \tilde{\mathbf{z}}_N^{\mathrm{du}} =& \mathbf{z}_N^{\mathrm{du}} + \tilde{\mathbf{J}}_{N,\mathrm{ss}}^{\mathrm{du}-T}\tilde{\mathbf{g}}_{N,\mathrm{ss}}^{\mathrm{du}} = \mathbf{J}_{N,\mathrm{ss}}^{\mathrm{du}-T}\mathbf{J}_{N,\mathrm{ss}}^{\mathrm{du}T}\tilde{\mathbf{J}}_{N,\mathrm{ss}}^{\mathrm{du}-T}(\tilde{\mathbf{J}}_{N,\mathrm{ss}}^{\mathrm{du}T}\mathbf{z}_N^{\mathrm{du}} + \tilde{\mathbf{g}}_{N,\mathrm{ss}}^{\mathrm{du}}), \\ =& \mathbf{J}_{N,\mathrm{ss}}^{\mathrm{du}-T}(I - (I - \mathbf{J}_{N,\mathrm{ss}}^{\mathrm{du}T}\tilde{\mathbf{J}}_{N,\mathrm{ss}}^{\mathrm{du}-T}))(\tilde{\mathbf{J}}_{N,\mathrm{ss}}^{\mathrm{du}T}\mathbf{z}_N^{\mathrm{du}} + \tilde{\mathbf{g}}_{N,\mathrm{ss}}^{\mathrm{du}}), \\ =& \mathbf{J}_{N,\mathrm{ss}}^{\mathrm{du}-T}(I - \mathbf{A}_1)(\tilde{\mathbf{J}}_{N,\mathrm{ss}}^{\mathrm{du}T}\mathbf{z}_N^{\mathrm{du}} + \tilde{\mathbf{g}}_{N,\mathrm{ss}}^{\mathrm{du}}),\end{aligned} \tag{3.32}$$

where we have defined $\mathbf{A}_1 \equiv I - \mathbf{J}_{N,\mathrm{ss}}^{\mathrm{du}T}\tilde{\mathbf{J}}_{N,\mathrm{ss}}^{\mathrm{du}-T}$. We apply the operator $\mathbf{r}_{N,\mathrm{ss}}^{\mathrm{du}T}$ to get

$$\mathbf{r}_{N,\mathrm{ss}}^{\mathrm{du}T}(\mathbf{z}_N^{\mathrm{du}} - \tilde{\mathbf{z}}_N^{\mathrm{du}}) = \mathbf{r}_{N,\mathrm{ss}}^{\mathrm{du}T}\mathbf{J}_{N,\mathrm{ss}}^{\mathrm{du}-T}(I - \mathbf{A}_1)(\tilde{\mathbf{J}}_{N,\mathrm{ss}}^{\mathrm{du}T}\mathbf{z}_N^{\mathrm{du}} + \tilde{\mathbf{g}}_{N,\mathrm{ss}}^{\mathrm{du}}). \tag{3.33}$$

Using the rules of matrix multiplication and the definition of our tangent problem we get $\mathbf{r}_{N,\mathrm{ss}}^{\mathrm{du}T}\mathbf{J}_{N,\mathrm{ss}}^{\mathrm{du}-T} = (\mathbf{J}_{N,\mathrm{ss}}^{\mathrm{du}-1}\mathbf{r}_{N,\mathrm{ss}}^{\mathrm{du}})^T = \boldsymbol{\psi}_N^{\mathrm{du}T}$. It follows from the triangle inequality that

$$|\mathbf{r}_{N,\mathrm{ss}}^{\mathrm{du}T}(\mathbf{z}_N^{\mathrm{du}} - \tilde{\mathbf{z}}_N^{\mathrm{du}})| \leq |\boldsymbol{\psi}_N^{\mathrm{du}T}(\tilde{\mathbf{J}}_{N,\mathrm{ss}}^{\mathrm{du}T}\mathbf{z}_N^{\mathrm{du}} + \tilde{\mathbf{g}}_{N,\mathrm{ss}}^{\mathrm{du}})| + |\boldsymbol{\psi}_N^{\mathrm{du}T}\mathbf{A}_1(\tilde{\mathbf{J}}_{N,\mathrm{ss}}^{\mathrm{du}T}\mathbf{z}_N^{\mathrm{du}} + \tilde{\mathbf{g}}_{N,\mathrm{ss}}^{\mathrm{du}})|. \tag{3.34}$$

We use the constraint (3.26) to bound the first term of (3.34)

$$|\boldsymbol{\psi}_N^{\mathrm{du}T}(\tilde{\mathbf{J}}_{N,\mathrm{ss}}^{\mathrm{du}T}\mathbf{z}_N^{\mathrm{du}} + \tilde{\mathbf{g}}_{N,\mathrm{ss}}^{\mathrm{du}})| \leq |(\boldsymbol{\psi}_N^{\mathrm{du,w}})^T(\tilde{\mathbf{J}}_{N,\mathrm{ss}}^{\mathrm{du}T}\mathbf{z}_N^{\mathrm{du}} + \tilde{\mathbf{g}}_{N,\mathrm{ss}}^{\mathrm{du}})| \leq \frac{\delta_{\eta_1}}{2} \tag{3.35}$$

and hence

$$|\boldsymbol{\psi}_N^{\mathrm{du}T}(\tilde{\mathbf{J}}_{N,\mathrm{ss}}^{\mathrm{du}T}\mathbf{z}_N^{\mathrm{du}} + \tilde{\mathbf{g}}_{N,\mathrm{ss}}^{\mathrm{du}})| \leq N\|\boldsymbol{\psi}_N^{\mathrm{du,w}} \circ (\tilde{\mathbf{J}}_{N,\mathrm{ss}}^{\mathrm{du}T}\mathbf{z}_N^{\mathrm{du}} + \tilde{\mathbf{g}}_{N,\mathrm{ss}}^{\mathrm{du}})\|_\infty \leq \frac{\delta_{\eta_1}}{2}.$$

We bound the second term of (3.34) by noting that

$$\psi^{\mathrm{du,min}}\|\tilde{\mathbf{J}}_{N,\mathrm{ss}}^{\mathrm{du}T}\mathbf{z}_N^{\mathrm{du}} + \tilde{\mathbf{g}}_{N,\mathrm{ss}}^{\mathrm{du}}\|_\infty \leq \|\boldsymbol{\psi}_N^{\mathrm{du,w}} \circ (\tilde{\mathbf{J}}_{N,\mathrm{ss}}^{\mathrm{du}T}\mathbf{z}_N^{\mathrm{du}} + \tilde{\mathbf{g}}_{N,\mathrm{ss}}^{\mathrm{du}})\|_\infty \leq \frac{\delta_{\eta_1}}{2N},$$

by $\psi^{\mathrm{du,min}} \leq \boldsymbol{\psi}_N^{\mathrm{du,w}}$ and (3.35). It follows that

$$\|\tilde{\mathbf{J}}_{N,\mathrm{ss}}^{\mathrm{du}T}\mathbf{z}_N^{\mathrm{du}} + \tilde{\mathbf{g}}_{N,\mathrm{ss}}^{\mathrm{du}}\|_\infty \leq \frac{\sqrt{N}}{\sqrt{\delta_{\eta_1}}\|\boldsymbol{\psi}_N^{\mathrm{du}}(\mu)\|_2}\frac{\delta_{\eta_1}}{2N} = \frac{\sqrt{\delta_{\eta_1}}}{2\sqrt{N}\|\boldsymbol{\psi}_N^{\mathrm{du}}(\mu)\|_2}, \tag{3.36}$$

by definition of $\psi^{\mathrm{du,min}}$, and hence

$$\begin{aligned}|\boldsymbol{\psi}_N^{\mathrm{du}}\mathbf{A}_1(\tilde{\mathbf{J}}_{N,\mathrm{ss}}^{\mathrm{du}T}\mathbf{z}_N^{\mathrm{du}} + \tilde{\mathbf{g}}_{N,\mathrm{ss}}^{\mathrm{du}})| \leq \|\boldsymbol{\psi}_N^{\mathrm{du}}\|_2\|\mathbf{A}_1\|_2\|\tilde{\mathbf{J}}_{N,\mathrm{ss}}^{\mathrm{du}T}\mathbf{z}_N^{\mathrm{du}} + \tilde{\mathbf{g}}_{N,\mathrm{ss}}^{\mathrm{du}}\|_2 \leq \\ N^{3/2}\|\boldsymbol{\psi}_N^{\mathrm{du}}\|_2\|\mathbf{A}_1\|_{\max}\|\tilde{\mathbf{J}}_{N,\mathrm{ss}}^{\mathrm{du}T}\mathbf{z}_N^{\mathrm{du}} + \tilde{\mathbf{g}}_{N,\mathrm{ss}}^{\mathrm{du}}\|_\infty \leq \frac{N}{2}\delta_J\sqrt{\delta_{\eta_1}} \leq \epsilon\delta_{\eta_1} + \mathcal{O}(\delta_J^2),\end{aligned} \tag{3.37}$$

where the third inequality follows from the definitions of our norms, while the fourth inequality follows from (3.28) and (3.36), and the final inequality follows from Young's inequality for arbitrary small ϵ. We combine (3.35) and (3.37) to get

$$|\mathbf{r}_{N,\mathrm{ss}}^{\mathrm{du}T}(\mathbf{z}_N^{\mathrm{du}} - \tilde{\mathbf{z}}_N^{\mathrm{du}})| \leq (\frac{1}{2} + \epsilon)\delta_{\eta_1} + \mathcal{O}(\delta_J^2),$$

for arbitrary small ϵ, which is the first desired inequality.

Next we define $\mathbf{A}_2 \equiv \mathbf{I} - \tilde{\mathbf{J}}_{N,\mathrm{ss}}^{\mathrm{du}-T}\mathbf{J}_{N,\mathrm{ss}}^{\mathrm{du}T}$ so that we can write

$$\mathbf{z}_N^{\mathrm{du}} - \tilde{\mathbf{z}}_N^{\mathrm{du}} = (\mathbf{I} - \mathbf{A}_2)\mathbf{J}_{N,\mathrm{ss}}^{\mathrm{du}-T}(\tilde{\mathbf{J}}_{N,\mathrm{ss}}^{\mathrm{du}T}\mathbf{z}_N^{\mathrm{du}} + \tilde{\mathbf{g}}_{N,\mathrm{ss}}^{\mathrm{du}}).$$

We notice that $\tilde{\mathbf{J}}_{N,\mathrm{ss}}^{\mathrm{du}T}\mathbf{z}_N^{\mathrm{du}} + \tilde{\mathbf{g}}_{N,\mathrm{ss}}^{\mathrm{du}}$ is the residual for the dual RB-EQP evaluated about the dual RB state $\mathbf{z}_N^{\mathrm{du}}$ and the primal RB-EQP state $\tilde{\mathbf{u}}_N$. In proposition 3 from section 3.2 we state that we can control the global solution error for steady problems by enforcing the constraint $\mathbf{J}_{N,\mathrm{ss}}^{\mathrm{pr}}(\mathbf{u}_N(\mu);\mu)^{-1}\tilde{\mathbf{r}}_{N,\mathrm{ss}}^{\mathrm{pr}}(\mathbf{u}_N(\mu);\mu)$ [49]. We write the equivalent constraint for the dual problem $\mathbf{J}_{N,\mathrm{ss}}^{\mathrm{du}-T}(\tilde{\mathbf{J}}_{N,\mathrm{ss}}^{\mathrm{du}T}\mathbf{z}_N^{\mathrm{du}} + \tilde{\mathbf{g}}_{N,\mathrm{ss}}^{\mathrm{du}})$, which corresponds to constraint (3.27). We

can now prove the second inequality (3.31):

$$\|\mathbf{z}_N^{\mathrm{du}}-\tilde{\mathbf{z}}_N^{\mathrm{du}}\|_2 = \|(\mathbf{I}-\mathbf{A}_2)\mathbf{J}_{N,\mathrm{ss}}^{\mathrm{du}-T}(\tilde{\mathbf{J}}_{N,\mathrm{ss}}^{\mathrm{du}T}\mathbf{z}_N^{\mathrm{du}}+\tilde{\mathbf{g}}_{N,\mathrm{ss}}^{\mathrm{du}})\|_2 \leq (1+\|\mathbf{A}_2\|_2)\|\mathbf{J}_{N,\mathrm{ss}}^{\mathrm{du}-T}(\tilde{\mathbf{J}}_{N,\mathrm{ss}}^{\mathrm{du}T}\mathbf{z}_N^{\mathrm{du}}+\tilde{\mathbf{g}}_{N,\mathrm{ss}}^{\mathrm{du}})\|_2$$
$$\leq (1+N\|\mathbf{A}_2\|_{\max})\sqrt{N}\|\mathbf{J}_{N,\mathrm{ss}}^{\mathrm{du}-T}(\tilde{\mathbf{J}}_{N,\mathrm{ss}}^{\mathrm{du}T}\mathbf{z}_N^{\mathrm{du}}+\tilde{\mathbf{g}}_{N,\mathrm{ss}}^{\mathrm{du}})\|_\infty \leq \frac{1}{2}(1+N\delta_J)\sqrt{\delta_{\eta_1}} \leq \frac{1}{2}\sqrt{\delta_{\eta_1}}+\mathcal{O}(\delta_{\eta_1})+\mathcal{O}(\delta_J^2),$$

where the first inequality follows from $\|(\mathbf{I}-\mathbf{A})\mathbf{v}\|_2 \leq (1+\|\mathbf{A}\|_2)\|\mathbf{v}\|_2 \; \forall \mathbf{v} \in \mathbb{R}^N$, the second inequality follows from the norm inequalities, the third inequality follows from (3.27) and (3.29), and the last inequality follows from Young's inequality. This is the desired inequality. □

Corollary 14. *The combination of* (3.20) *from section 3.4 and* (3.31) *implies that*

$$|(\mathbf{z}_N^{\mathrm{du}} - \tilde{\mathbf{z}}_N^{\mathrm{du}})^T(\mathbf{r}_{N,\mathrm{ss}}^{\mathrm{du}}(\tilde{\mathbf{u}}_N) - \tilde{\mathbf{r}}_{N,\mathrm{ss}}^{\mathrm{du}}(\tilde{\mathbf{u}}))| \leq \|\mathbf{z}_N^{\mathrm{du}} - \tilde{\mathbf{z}}_N^{\mathrm{du}}\|_2\|\mathbf{r}_{N,\mathrm{ss}}^{\mathrm{du}}(\tilde{\mathbf{u}}_N) - \tilde{\mathbf{r}}_{N,\mathrm{ss}}^{\mathrm{du}}(\tilde{\mathbf{u}}_N)\|_2 \leq \delta_\eta/2,$$

so that error source (iii) is bounded by $\delta_\eta/2$*, where* $\delta_\eta = \max(\delta_1, \delta_2)$.

In the DWR EQP formulation presented in [48], the authors use the definition of $r^{\mathrm{du,min}}$ to bound $\|\mathbf{J}_{N,\mathrm{ss}}^{\mathrm{du}-T}(\tilde{\mathbf{J}}_{N,\mathrm{ss}}^{\mathrm{du}T}\mathbf{z}_N^{\mathrm{du}} + \tilde{\mathbf{g}}_{N,\mathrm{ss}}^{\mathrm{du}})\|_2$ implicitly, which in turn allows the formulation to control the error in $\|\mathbf{z}_N^{\mathrm{du}} - \tilde{\mathbf{z}}_N^{\mathrm{du}}\|_2$. However, in the new approach using the tangent solution $\boldsymbol{\psi}_N^{\mathrm{du}} = \mathbf{J}_{N,\mathrm{ss}}^{\mathrm{du}-T}\mathbf{r}_{N,\mathrm{ss}}^{\mathrm{du}}$, we are no longer able to use $r^{\mathrm{du,min}}$. We must instead introduce (3.27) as an explicit constraint, which corresponds to the constraint for global error control presented in [49].

3.6 Summary

In this chapter we presented a general form of EQP to control the error in the approximation of $\mathbf{r}_{N,\mathrm{ss}}^{\mathrm{pr}}$ by $\tilde{\mathbf{r}}_{N,\mathrm{ss}}^{\mathrm{pr}}$ using the $\mathrm{LP}_{\mathrm{EQP}}$ procedure (see definition 1 in section 3.1). In section 3.2 we showed how we can transform the general form of EQP to control the global solution error $\|\mathbf{u}_N - \tilde{\mathbf{u}}_N\|_{\mathcal{V}}$ using the LP^{ν} procedure (see definition 2). Then in section 3.3 we showed how to use the $\mathrm{LP}_{\mathrm{ss}}^{r}$ and $\mathrm{LP}_{\mathrm{ss}}^{q}$ procedures to control the output error (see definitions 4 and 6), while in section 3.4 we showed how to use the $\mathrm{LP}_{\mathrm{ss}}^{\eta}$ to control the error in the DWR error estimate (see definition 7). Finally in section 3.5, we introduced a new approach to combining EQP and DWR to control the error in the DWR error estimate using the $\mathrm{LP}_{\mathrm{ss}}^{\eta,1}$ and $\mathrm{LP}_{\mathrm{ss}}^{\eta,2}$ procedures (see definitions 11 and 12). We will use this new approach to extend the goal-oriented EQP framework to time-dependent problems in Chapter 4.

Chapter 4

EQP for Time-Dependent PDEs

In chapter 3 we demonstrated the versatility of the EQP framework for steady problems. In this chapter we will extend the goal-oriented EQP framework and its associated DWR error estimator to time-dependent problems. First we will present the relevant preliminary materials in section 4.1, followed by a discussion of the space-time DWR formulation in section 4.2. In section 3.5 we presented an alternate form of EQP DWR for steady problems that relies on the dual-of-the-dual (tangent) approach. In section 4.3 we will present the dual-of-the-dual approach for time-dependent problems, which we will use to extend DWR EQP to time-dependent problems in section 4.5. We will also extend goal-oriented EQP to time-dependent problems in section 4.4.

4.1 Preliminary Materials

Here we introduce the relevant preliminary materials to make this chapter as self-contained as possible. We will solve our primal and dual problems over the spatial domain Ω and the time domain $I \equiv (0, T]$, where T represents the terminal time. We project our problem onto the Sobolev space $\mathcal{V}$ (over Ω). Our unsteady residual has the form

$$\bar{r}(w, v; \mu, t) = \int_\Omega v \frac{\partial w}{\partial t} dx + r(w, v; \mu, t) \quad \forall w, v \in \mathcal{V},$$

while our output has the form

$$s(\mu) = \int_I q(u(\mu, t); \mu, t) dt + q_T(u(\mu, T); \mu, T).$$

4.2 Space-Time DWR

We presented the DWR method for steady problems in section 2.4, while the DWR method was first presented [6]. The DWR method has also previously been applied to time-dependent problems in the context of CFD [8, 21]. Here we will derive the DWR relationship for time-dependent problems. In this section we will include μ and t as arguments, but we will later omit these arguments for the sake of brevity.

Bochner spaces are often used in the functional analysis approach to study time-dependent PDEs [38]. We introduce (i) the space-time Bochner space $\mathscr{V} = C^0([0,T];L^2(\Omega))\cap L^2(I;\mathcal{V})$, (ii) the dual space $\mathscr{V}'$, and (iii) duality pairing $\langle\cdot,\cdot\rangle : \mathscr{V}'\times\mathscr{V}\to\mathbb{R}$. Then we define the space-time operator $A : \mathscr{V}\times\mathcal{D}\to\mathscr{V}'$ such that

$$\langle A(w;\mu),v\rangle \equiv \int_I\int_\Omega v\frac{\partial w}{\partial t}dxdt + \int_I r(w,v;\mu,t)dt - \int_\Omega v(t=0)(w_0 - w(t=0))dx \quad \forall w,v\in\mathscr{V}.$$

and the space-time output operator $J : \mathscr{V}\times\mathcal{D}\to\mathbb{R}$ such that

$$J(w;\mu) \equiv \int_I q(w;\mu,t)dt + q_T(w;\mu,T) \quad \forall w\in\mathscr{V}.$$

We also define the Lagrangian operator $L : \mathscr{V}\times\mathscr{V}\times\mathcal{D}\to\mathbb{R}$

$$L(w;v;\mu) \equiv \langle A(w;\mu),v\rangle + J(w;\mu) \quad \forall w,v\in\mathscr{V},$$

and require the primal and dual solutions $\{u,z\}\in\mathscr{V}\times\mathscr{V}$ to correspond to a stationary point of the Lagrangian. We seek solutions $\{u,z\}\in\mathscr{V}\times\mathscr{V}$ to the Euler-Lagrange system

$$\langle A(u;\mu),v\rangle = 0, \quad \forall v\in\mathscr{V}, \tag{4.1}$$

$$\langle DA(u;\mu)v,z\rangle + \langle DJ(u;\mu),v\rangle = 0, \quad \forall v\in\mathscr{V}, \tag{4.2}$$

where $DA : \mathscr{V}\times\mathcal{D}\to\mathscr{V}'$ and $DJ : \mathscr{V}\times\mathcal{D}\to\mathscr{V}'$ are the derivative operators such that

$$\langle DA(w;\mu)v,z\rangle \equiv \lim_{\epsilon\to 0}\frac{1}{\epsilon}(\langle A(w+\epsilon v;\mu),z\rangle - \langle A(w;\mu),z\rangle) \quad \forall w,z,v\in\mathscr{V}, \tag{4.3}$$

$$\langle DJ(w;\mu),v\rangle \equiv \lim_{\epsilon\to 0}\frac{1}{\epsilon}(J(w+\epsilon v;\mu) - J(w;\mu)) \quad \forall w,v\in\mathscr{V}. \tag{4.4}$$

We write two forms of the operator DA:

$$\langle DA(w;\mu)v, z\rangle = \int_I \int_\Omega z\frac{\partial v}{\partial t}dxdt + \int_I r'(w;v,z;\mu,t)dt - \int_\Omega [zv]_{t=0}dx, \tag{4.5}$$

$$= -\int_I \int_\Omega v\frac{\partial z}{\partial t}dxdt + \int_I r'(w;v,z;\mu,t)dt + \int_\Omega [zv]_{t=T}dx, \tag{4.6}$$

for all $w, z, v \in \mathscr{V}$, which are equivalent using integration by parts. We recall that $r'(w;v,z;\mu,t)$ is the Gateaux derivative of $r(\cdot,z;\mu,t)$ about w in the direction v. We also write the derivative operator DJ:

$$\langle DJ(w;\mu), v\rangle \equiv \int_I q'(w;v;\mu,t)dt + q'_T(w;v;\mu,T) \quad \forall w, v \in \mathscr{V}, \tag{4.7}$$

where $q'(w;v;\mu,t)$ and $q'_T(w;v;\mu,T)$ are the Gateaux derivatives of $q(\cdot;\mu,t)$ and $q_T(\cdot;\mu,T)$, respectively, about w in the direction v.

We write the adjoint equation (4.2) using the operators defined above

$$-\int_I \int_\Omega v\frac{\partial z}{\partial t}dxdt + \int_I r'(u;v,z;\mu,t)dt + \int_I q'(u;v;\mu,t)dt + q'_T(u(T);v;\mu,T) + \int_\Omega [zv]_{t=T}dx = 0 \quad \forall u, v, z \in \mathscr{V}. \tag{4.8}$$

We break this statement into two components: (i) the terminal condition, and (ii) the equation for all other times. We make this equation strong in time and write:

$$\int_\Omega [zv]_{t=T}dx + q'_T(u(T);v;\mu,T) = 0 \quad \forall u, v, z \in \mathscr{V}, \tag{4.9}$$

$$-\int_\Omega v\frac{\partial z}{\partial t}dx + r'(u;v,z;\mu,t) + q'(u;v;\mu,t) = 0 \quad \forall u, v, z \in \mathscr{V}. \tag{4.10}$$

We solve the primal problem in the space $\mathscr{V}_n \subset \mathscr{V}$, where n indicates a general sequence of approximations. The space $\mathscr{V}_n$ is meant to approximate the space $\mathscr{V}$, while the solution $u_n \in \mathscr{V}_n$ is meant to approximate the solution $u \in \mathscr{V}$. This leads to error in the evaluation of the output functional and we can estimate this error using the DWR method

$$J(u;\mu) - J(u_n;\mu) = -\langle A(u_n;\mu), z\rangle + \mathcal{O}(\|u - u_n\|^2_{\mathscr{V}}) \quad z \in \mathscr{V}. \tag{4.11}$$

We see that the truncation error is second-order, meaning that the DWR error estimate is first-order accurate. In section 3.4 we demonstrated how the DWR method can be used in the RB-EQP context to estimate the error between the output functional evaluated

using the true solution and the output functional evaluated using the RB-EQP solution.

4.3 Error in Space-Time DWR

In section 3.5 we discussed the dual-of-the-dual (tangent) approach for steady problems and we will now extend this discussion to time-dependent problems. We cannot solve for the exact dual solution $z \in \mathscr{V}$, so we instead seek the approximate dual solution $z_{\hat{n}} \in \mathscr{V}_{\hat{n}}$ for $u_n \in \mathscr{V}_n$, where $\mathscr{V}_n \subset \mathscr{V}_{\hat{n}} \subset \mathscr{V}$, such that

$$\langle DA(u_n)v, z_{\hat{n}}\rangle + \langle DJ(u_n), v\rangle = 0 \quad \forall v \in \mathscr{V}_{\hat{n}}.$$

We evaluate the DWR error estimate $\eta_{\hat{n}} = -\langle A(u_n), z_{\hat{n}}\rangle)$ and treat it as our "output". Then we apply the DWR method to the dual problem and find that the dual-of-the-dual $\psi \in \mathscr{V}$ satisfies

$$\langle DA(u_n)\psi, v\rangle - \langle A(u_n), v\rangle = 0 \quad \forall v \in \mathscr{V}, \tag{4.12}$$

so that we can estimate the error in the DWR error estimate as

$$\langle A(u_n), z\rangle - \langle A(u_n), z_{\hat{n}}\rangle = -\langle DA(u_n)\psi, z_{\hat{n}}\rangle - \langle DJ(u_n), \psi\rangle + \mathcal{O}(\|z - z_{\hat{n}}\|_{\mathscr{V}}^2). \tag{4.13}$$

We use the definitions $DA(w)$ and $DJ(w)$ from equations (4.6) and (4.7), respectively, to rewrite the dual-of-the-dual equation (4.12) as

$$\int_I \int_\Omega v\frac{d\psi}{dt}dxdt + \int_I r'(u;\psi,v)dt - \int_I \int_\Omega v\frac{\partial u}{\partial t}dxdt - \int_I r(u;v)dt = 0 \quad \forall v \in \mathscr{V},$$

while we write the estimate of the error in the DWR error estimate (4.13) as

$$\begin{aligned}
&\langle A(u), z\rangle - \langle A(u), z_h\rangle \approx -\langle DA(u)\psi, z_h\rangle - \langle DJ(u), \psi\rangle \\
&= -\int_I \int_\Omega z_h\frac{\partial \psi}{\partial t}dxdt - \int_I r'(u;\psi,z_h)dt - \int_I q'(u;\psi)dt - q_T'(u;\psi). \qquad (4.14)\\
&= \int_I \int_\Omega \psi\frac{\partial z_h}{\partial t}dxdt - \int_I r'(u;\psi,z_h)dt - \int_I q'(u;\psi)dt - q_T'(u;\psi) - \int_\Omega [z\psi]_T dx. \qquad (4.15)
\end{aligned}$$

In section 4.5 we introduce EQP constraints to control the error in the dual solution. We will choose our constraints based on (4.15) rather than (4.14) so that the dual residual used to compute the estimate of the error in the DWR error estimate is consistent with the dual residual used to solve the dual problem.

4.4 Goal-Oriented EQP

4.4.1 Preliminary Materials

We will now extend the goal-oriented EQP framework presented in section 3.3 so that it applies to time-dependent problems. First we solve the FE problem for the training parameters in $\Xi_{N_{\text{train}}}$ over time-steps $k = 1, 2, \ldots, K$, then we use these snapshots to construct the $\mathcal{V}$-orthonormal basis $\{\phi_i^{\text{pr}}\}_{i=1}^N$ of $\mathcal{V}_N^{\text{pr}}$. Next we introduce the RB space-time space $\mathscr{V}_N^{\text{pr}} = C^0([0,T];\mathcal{V}_N^{\text{pr}}) \cap L^2(I;\mathcal{V}_N^{\text{pr}})$. We also recall the operator $V_N^{\text{pr}} : \mathbb{R}^N \to \mathcal{V}_N$, which associates a generalized coordinate $\mathbf{w} \in \mathbb{R}^N$ to a function $w \in \mathcal{V}_N^{\text{pr}}$ as $w_N = V_N^{\text{pr}}\mathbf{w}_N = \sum_{i=1}^N \mathbf{w}_{N,i}\phi_{N,i}$. We reintroduce the residual form $r_h : \mathcal{V}_h \times \mathcal{V}_h \times \mathcal{D} \times I \to \mathcal{V}_h$ and the bilinear form $m_h : \mathcal{V}_h \times \mathcal{V}_h \to \mathcal{V}_h$

$$r_h(w, v; \mu, t) \equiv \sum_{\kappa=1}^{M^h} \rho_\kappa^h r_\kappa(w, v; \mu, t), \tag{4.16}$$

$$m_h(w, v) \equiv \sum_{\kappa=1}^{M^h} \rho_\kappa^h m_\kappa(w, v), \tag{4.17}$$

associated with the "truth" quadrature weights $\{\rho_\kappa^h \in \mathbb{R}_{\geq 0}\}_{\kappa=1}^{M^h}$. Then we introduce the algebraic residual $\mathbf{r}_N^{\text{pr}} : \mathbb{R}^N \times \mathcal{D} \times I \to \mathbb{R}^N$ such that

$$\mathbf{r}_N^{\text{pr}}(\mathbf{w}; \mu, t)_i \equiv r_h(V_N^{\text{pr}}\mathbf{w}; \phi_i^{\text{pr}}; \mu, t), \quad i = 1, 2, \ldots, N, \tag{4.18}$$

and the mass matrix $\mathbf{M}_N^{\text{pr}} \in \mathbb{R}^{N \times N}$ such that

$$(\mathbf{M}_N^{\text{pr}})_{ij} \equiv m_h(\phi_j^{\text{pr}}, \phi_i^{\text{pr}}) \quad i, j = 1, 2, \ldots, N. \tag{4.19}$$

We state the primal RB problem, discretized using the backward Euler time-marching method: given $\mu \in \mathcal{D}$ find $\mathbf{u}_{N,k}(\mu) \in \mathcal{V}_N^{\text{pr}}$ such that

$$\mathbf{M}_N^{\text{pr}}(\mathbf{u}_{N,k}(\mu) - \mathbf{u}_{N,k-1}(\mu)) + (t_k - t_{k-1})\mathbf{r}_N^{\text{pr}}(\mathbf{u}_{N,k}(\mu); \mu, t_k) = 0, \quad k = 1, 2, \ldots, K. \tag{4.20}$$

We introduce the output forms $q_h : \mathcal{V}_h \times \mathcal{D} \times I \to \mathbb{R}$ and $q_{h,T} : \mathcal{V}_h \times \mathcal{D} \times I \to \mathbb{R}$

$$q_h(w;\mu,t) \equiv \sum_{\kappa=1}^{M^h} \rho_\kappa^h q_\kappa(w;\mu,t), \tag{4.21}$$

$$q_{h,T}(w;\mu,t) \equiv \sum_{\kappa=1}^{M^h} \rho_\kappa^h q_{T,\kappa}(w;\mu,t). \tag{4.22}$$

We reintroduce the output functionals $\mathbf{q}_N : \mathbb{R}^N \times \mathcal{D} \times I \to \mathbb{R}$ and $\mathbf{q}_{N,T} : \mathbb{R}^N \times \mathcal{D} \times I \to \mathbb{R}$

$$\mathbf{q}_N(\mathbf{w};\mu,t) \equiv q_h(V_N^{\mathrm{pr}}\mathbf{w};\mu,t),$$
$$\mathbf{q}_{N,T}(\mathbf{w};\mu,T) \equiv q_{h,T}(V_N^{\mathrm{pr}}\mathbf{w};\mu,T).$$

and we evaluate the output

$$s_N(\mu) \equiv \int_I \mathbf{q}_N(\mathbf{u}_N(\mu,t);\mu,t)dt + \mathbf{q}_{N,T}(\mathbf{u}_N(\mu,T);\mu,T),$$

or equivalently $s_N(\mu) = J(u_N;\mu)$.

We introduce the residual EQP weights $\{\rho_\kappa^r \in \mathbb{R}_{\geq 0}\}_{\kappa=1}^{M^h}$ and the EQP forms $\tilde{r}_h^{(r)} : \mathcal{V}_h \times \mathcal{V}_h \times \mathcal{D} \times I \to \mathcal{V}_h$ and $\tilde{m}_h^{(r)} : \mathcal{V}_h \times \mathcal{V}_h$ such that

$$\tilde{r}_h^{(r)}(w,v;\mu,t) \equiv \sum_{\kappa=1}^{M^h} \rho_\kappa^r r_\kappa(w,v;\mu,t), \tag{4.23}$$

$$\tilde{m}_h^{(r)}(w,v) \equiv \sum_{\kappa=1}^{M^h} \rho_\kappa^r m_\kappa(w,v), \tag{4.24}$$

which allow us to introduce the RB-EQP residual $\tilde{\mathbf{r}}_N^{\mathrm{pr}} : \mathbb{R}^N \times \mathcal{D} \times I \to \mathbb{R}^N$ and mass matrix $\tilde{\mathbf{M}}_N^{\mathrm{pr}} \in \mathbb{R}^{N\times N}$ such that

$$\tilde{\mathbf{r}}_N^{\mathrm{pr}}(\mathbf{w};\mu,t) \equiv \tilde{r}_h^{(r)}(V_N^{\mathrm{pr}}\mathbf{w}, \phi_i^{\mathrm{pr}};\mu,t), \quad i = 1,2,\ldots,N$$
$$(\tilde{\mathbf{M}}_N^{\mathrm{pr}})_{ij} \equiv \tilde{m}_h^{(r)}(\phi_j^{\mathrm{pr}}, \phi_i^{\mathrm{pr}}), \quad i,j = 1,2,\ldots,N.$$

We state the primal RB-EQP problem, discretized using the backward Euler time-marching method: given $\mu \in \mathcal{D}$ find $\tilde{\mathbf{u}}_{N,k}(\mu) \in \mathcal{V}_N^{\mathrm{pr}}$ such that

$$\tilde{\mathbf{M}}_N^{\mathrm{pr}}(\tilde{\mathbf{u}}_{N,k}(\mu) - \tilde{\mathbf{u}}_{N,k-1}(\mu)) + (t_k - t_{k-1})\tilde{\mathbf{r}}_N^{\mathrm{pr}}(\tilde{\mathbf{u}}_{N,k}(\mu);\mu,t_k) = 0, \quad k = 1,2,\ldots,K. \tag{4.25}$$

We introduce the output functional EQP weights $\{\rho_\kappa^q \in \mathbb{R}_{\geq 0}\}_{\kappa=1}^{M^h}$ and introduce the EQP

output forms $\tilde{q}_h^{(q)}(w,v;\mu,t) \equiv \sum_{\kappa=1}^{M^h} \rho_\kappa^q q(w,v;\mu,t)$ and $\tilde{q}_{h,T}^{(q)}(w,v;\mu,T) \equiv \sum_{\kappa=1}^{M^h} \rho_\kappa^q q_T(w,v;\mu,T)$. This allows us to write the RB-EQP output functionals $\tilde{\mathbf{q}}_N(\mathbf{w};\mu,t) \equiv \tilde{q}_h^{(q)}(V_N^{\mathrm{pr}}\mathbf{w};\mu,t)$ and $\tilde{\mathbf{q}}_{N,T}(\mathbf{w};\mu,T) \equiv \tilde{q}_{h,T}^{(q)}(V_N^{\mathrm{pr}}\mathbf{w};\mu,T)$ and we evaluate the output

$$\tilde{s}_N(\mu) \equiv \int_I \tilde{\mathbf{q}}_N(\tilde{\mathbf{u}}_N(\mu,t);\mu,t)dt + \tilde{\mathbf{q}}_{N,T}(\tilde{\mathbf{u}}_N(\mu,T);\mu,T),$$

or equivalently $\tilde{s}_N(\mu) = \tilde{J}(\tilde{u}_N;\mu)$.

We wish to control the output error and we use the triangle inequality to write

$$|s_N(\mu) - \tilde{s}_N(\mu)| = |J(u_N;\mu) - \tilde{J}(\tilde{u}_N;\mu)| \leq \underbrace{|J(u_N;\mu) - J(\tilde{u}_N;\mu)|}_{\text{source (i)}} + \underbrace{|J(\tilde{u}_N;\mu) - \tilde{J}(\tilde{u}_N;\mu)|}_{\text{source (ii)}}. \tag{4.26}$$

We see that we have two sources of error: (i) the approximation of $\mathbf{u}_N$ by $\tilde{\mathbf{u}}_N$, and (ii) the approximation of $J(\cdot.\cdot)$ by $\tilde{J}(\cdot,\cdot)$. We will develop constraints to control these error sources using the DWR method for time-dependent problems presented in chapter 4.2.

4.4.2 EQP for Primal RB-EQP Problem

First we will control the error due to source (i) of (4.26). The solution $\mathbf{u}_{N,k}$ is the root of the discrete-in-time residual (2.16) at time-step k, so error source (i) arises from the error in the approximation of the primal space-time residual operator $A(\cdot;\cdot)$. We will control this error using the DWR method, approximated in the primal RB space. To further simplify the notation, we introduce the continuous-in-time residuals $\bar{\mathbf{r}}_N^{\mathrm{pr}}(\mathbf{w};\mu,t) \equiv \mathbf{M}_N^{\mathrm{pr}}\frac{d\mathbf{w}}{dt} + \mathbf{r}_N^{\mathrm{pr}}(\mathbf{w};\mu,t)$ and $\tilde{\bar{\mathbf{r}}}_N^{\mathrm{pr}}(\mathbf{w};\mu,t) \equiv \tilde{\mathbf{M}}_N^{\mathrm{pr}}\frac{d\mathbf{w}}{dt} + \tilde{\mathbf{r}}_N^{\mathrm{pr}}(\mathbf{w};\mu,t)$. We also note that $\tilde{A}(\cdot;\cdot)$ is the EQP approximation of operator $A(\cdot;\cdot)$, while $\tilde{J}(\cdot;\cdot)$ denotes the EQP approximation of operator $J(\cdot;\cdot)$. Moreover, $D\tilde{A}(\cdot;\cdot)$ denotes the EQP approximation to operator $DA(\cdot;\cdot)$, while $D\tilde{J}(\cdot;\cdot)$ denotes the EQP approximation to operator $DJ(\cdot;\cdot)$. We now introduce a proposition to control the output error due to source (i) of (4.26).

Proposition 15. *(Output error due to the EQP approximation of $A(\cdot,\cdot)$). Suppose*

$$|\int_I \mathbf{z}_N^{\mathrm{pr}T}(\bar{\mathbf{r}}_N^{\mathrm{pr}}(\hat{\mathbf{u}}_N) - \tilde{\bar{\mathbf{r}}}_N^{\mathrm{pr}}(\hat{\mathbf{u}}_N))dt| \leq \delta_r \tag{4.27}$$

$$\|B_N^{\mathrm{pr}}\| \leq \delta_J, \tag{4.28}$$

where $\hat{\mathbf{u}}_N$ is a surrogate state that may differ from both $\mathbf{u}_N$ and $\tilde{\mathbf{u}}_N$, and where

$$B_N^{\mathrm{pr}} \equiv I - DA(\hat{u}_N)D\tilde{A}(\hat{u}_N)^{-1}, \tag{4.29}$$

with $\|B_N^{\rm pr}\| \equiv \sup_{w\in\mathscr{V}_N'} \sup_{v\in\mathscr{V}_N} \frac{\langle B_N^{\rm pr} w, v\rangle}{\langle w, v\rangle}$. *Then*

$$|J(u_N) - J(\tilde{u}_N)| \le \delta_r + \mathcal{O}(\delta_r^2) + \mathcal{O}(\delta_J^2) + \mathcal{O}(\hat{\delta}^2),$$

where $\hat{\delta} \equiv \|u_N - \hat{u}_N\|_{\mathscr{V}} = (\int_I \|\mathbf{u}_N - \hat{\mathbf{u}}_N\|_2 dt)^2$.

Proof. We recall the DWR error estimate (4.11). We make the substitution $u = u_N$, $u_n = \tilde{u}_N$, and $z = z_N^{\rm pr}$, so that we treat $u_N, z_N^{\rm pr} \in \mathscr{V}_N^{\rm pr}$ as our true solutions, while we treat $\tilde{u}_N \in \mathscr{V}_N^{\rm pr}$ as our approximate solution. We write

$$J(u_N) - J(\tilde{u}_N) = -\langle A(\tilde{u}_N), z_N^{\rm pr}\rangle + \mathcal{O}(\tilde{\delta}^2) = -\langle A(\tilde{u}_N), z_N^{\rm pr}\rangle + \underbrace{\langle A(u_N), z_N^{\rm pr}\rangle}_{=0} + \mathcal{O}(\tilde{\delta}^2),$$

and use Taylor series expansions to write

$$\begin{aligned} J(u_N) - J(\tilde{u}_N) =& -\langle A(\hat{u}_N), z_N^{\rm pr}\rangle - \langle DA(\hat{u}_N)(\tilde{u}_N - \hat{u}_N), z_N^{\rm pr}\rangle + \mathcal{O}(\|\tilde{u}_N - \hat{u}_N\|_{\mathscr{V}}^2) \\ &+ \langle A(\hat{u}_N), z_N^{\rm pr}\rangle + \langle DA(\hat{u}_N)(u_N - \hat{u}_N), z_N^{\rm pr}\rangle + \mathcal{O}(\|u_N - \hat{u}_N\|_{\mathscr{V}}^2), \\ =& \langle DA(\hat{u}_N)(u_N - \hat{u}_N), z_N^{\rm pr}\rangle - \langle DA(\hat{u}_N), (\tilde{u}_N - \hat{u}_N), z_N^{\rm pr}\rangle \\ &+ \mathcal{O}(\|\tilde{u}_N - \hat{u}_N\|_{\mathscr{V}}^2) + \mathcal{O}(\|u_N - \hat{u}_N\|_{\mathscr{V}}^2) \\ =& \langle DA(\hat{u}_N)(\delta\tilde{u}_N), z_N^{\rm pr}\rangle + \mathcal{O}(\tilde{\delta}^2) + \mathcal{O}(\hat{\delta}^2), \end{aligned} \tag{4.30}$$

where $\mathcal{O}(\|\tilde{u}_N - \hat{u}_N\|_{\mathscr{V}}^2) \le \mathcal{O}(\tilde{\delta}^2) + \mathcal{O}(\hat{\delta}^2)$ results from the triangle inequality (note that we define $\delta\tilde{u} \equiv u_N - \tilde{u}_N$ and $\tilde{\delta} \equiv \|u_N - \tilde{u}_N\|_{\mathscr{V}} = (\int_I \|\mathbf{u}_N - \tilde{\mathbf{u}}_N\|_2 dt)^2$). We note that $\delta\tilde{u}_N$ depends on the EQP rule that we find following the LP procedure, so we cannot use $\delta\tilde{u}_N$ to construct the LP constraints because this would give a recursive LP procedure. Therefore we seek an expression for $\delta\tilde{u}_N$ that does not depend on the EQP rule.

We use two more Taylor series expansions to write

$$\begin{aligned} \langle A(\hat{u}_N), v\rangle - \langle \tilde{A}(\hat{u}_N), v\rangle =& -\langle D\tilde{A}(\hat{u}_N)(\delta\tilde{u}_N), v\rangle \\ &+ \langle D\tilde{A}(\hat{u}_N)(\delta\hat{u}_N), v\rangle - \langle DA(\hat{u}_N)(\delta\hat{u}_N), v\rangle + \mathcal{O}(\hat{\delta}^2) + \mathcal{O}(\tilde{\delta}^2) \quad \forall v \in \mathscr{V}_N^{\rm pr}, \end{aligned}$$

where $\delta\hat{u}_N \equiv u_N - \hat{u}_N$. We rearrange to write

$$\begin{aligned} \langle D\tilde{A}(\hat{u}_N)(\delta\tilde{u}_N), v\rangle =& -(\langle A(\hat{u}_N), v\rangle - \langle \tilde{A}(\hat{u}_N), v\rangle) \\ &+ \langle D\tilde{A}(\hat{u}_N)(\delta\hat{u}_N), v\rangle - \langle DA(\hat{u}_N)(\delta\hat{u}_N), v\rangle + \mathcal{O}(\hat{\delta}^2) + \mathcal{O}(\tilde{\delta}^2) \quad \forall v \in \mathscr{V}_N^{\rm pr}. \end{aligned} \tag{4.31}$$

We apply the operator $DA(\hat{u}_N)D\tilde{A}(\hat{u}_N)$ to (4.31) to get

$$\begin{aligned} DA(\hat{u}_N)\delta\tilde{u}_N =& - DA(\hat{u}_N)D\tilde{A}(\hat{u}_N)^{-1}(A(\hat{u}_N) - \tilde{A}(\hat{u}_N)) \\ &+ DA(\hat{u}_N)(\delta\hat{u} - D\tilde{A}(\hat{u}_N)^{-1}DA(\hat{u}_N)(\delta\hat{u}_N)) + \mathcal{O}(\hat{\delta}^2) + \mathcal{O}(\tilde{\delta}^2), \\ =& - DA(\hat{u}_N)D\tilde{A}(\hat{u}_N)^{-1}(A(\hat{u}_N) - \tilde{A}(\hat{u}_N)) \\ &+ (I - DA(\hat{u}_N)D\tilde{A}(\hat{u}_N)^{-1})DA(\hat{u}_N)(\delta\hat{u}_N) + \mathcal{O}(\hat{\delta}^2) + \mathcal{O}(\tilde{\delta}^2), \\ =& - (I - (I - DA(\hat{u}_N)D\tilde{A}(\hat{u}_N)^{-1}))(A(\hat{u}_N) - \tilde{A}(\hat{u}_N)) \\ &+ (I - DA(\hat{u}_N)D\tilde{A}(\hat{u}_N)^{-1})DA(\hat{u}_N)(\delta\hat{u}_N) + \mathcal{O}(\hat{\delta}^2) + \mathcal{O}(\tilde{\delta}^2), \\ =& - (I - B_N^{\mathrm{pr}})(A(\hat{u}_N) - \tilde{A}(\hat{u}_N)) + (B_N^{\mathrm{pr}})(DA(\hat{u}_N)(\delta\hat{u}_N)) \\ &+ \mathcal{O}(\hat{\delta}^2) + \mathcal{O}(\tilde{\delta}^2). \end{aligned} \tag{4.32}$$

using the operator B_N^{pr} defined in (4.29). We combine (4.30) and (4.32) to write

$$\begin{aligned} |J(u_N) - J(\tilde{u}_N)| =& | - \langle (I - B_N^{\mathrm{pr}})(A(\hat{u}_N) - \tilde{A}(\hat{u}_N)), z_N^{\mathrm{pr}} \rangle \\ &+ \langle (B_N^{\mathrm{pr}})(DA(\hat{u}_N)(\delta\hat{u}_N)), z_N^{\mathrm{pr}} \rangle + \mathcal{O}(\delta\tilde{u}_N^2) + \mathcal{O}(\delta\hat{u}_N^2)|, \\ \leq& |\langle A(\hat{u}_N) - \tilde{A}_N(\hat{u}_N), z_N^{\mathrm{pr}} \rangle| + |\langle (B_N^{\mathrm{pr}})(A(\hat{u}_N) - \tilde{A}(\hat{u}_N)), z_N^{\mathrm{pr}} \rangle| + \\ & |\langle (B_N^{\mathrm{pr}})(DA(\hat{u}_N)(\delta\hat{u}_N)), z_N^{\mathrm{pr}} \rangle| + (\delta\tilde{u}_N^2) + \mathcal{O}(\delta\hat{u}_N^2), \end{aligned}$$

or

$$\begin{aligned} |J(u_N) - J(\tilde{u}_N)| \leq & |\langle A(\hat{u}_N), z_N^{\mathrm{pr}} \rangle - \langle \tilde{A}(\hat{u}_N), z_N^{\mathrm{pr}} \rangle| \\ &+ |\langle (B_N^{\mathrm{pr}})(A(\hat{u}_N)), z_N^{\mathrm{pr}} \rangle - \langle (B_N^{\mathrm{pr}})(\tilde{A}(\hat{u}_N)), z_N^{\mathrm{pr}} \rangle| \\ &+ |\langle (B_N^{\mathrm{pr}})(DA(\hat{u}_N)(\delta\hat{u}_N)), z_N^{\mathrm{pr}} \rangle| + (\delta\tilde{u}_N^2) + \mathcal{O}(\delta\hat{u}_N^2). \end{aligned} \tag{4.33}$$

We see that in (4.33), $|J(u_N) - J(\tilde{u}_N)|$ is bounded by the sum of three terms. We apply constraint (4.27) to the first term

$$\text{(I)} = |\langle A(\hat{u}_N), z_N^{\mathrm{pr}} \rangle - \langle \tilde{A}(\hat{u}_N), z_N^{\mathrm{pr}} \rangle| = | \int_I \mathbf{z}_N^{\mathrm{pr}T}(\bar{\mathbf{r}}_N^{\mathrm{pr}}(\hat{\mathbf{u}}_N) - \tilde{\bar{\mathbf{r}}}_N^{\mathrm{pr}}(\hat{\mathbf{u}}_N))dt| \leq \delta_r. \tag{4.34}$$

We rewrite the second term of (4.33)

$$\text{(II)} = |\langle (B_N^{\mathrm{pr}})(A(\hat{u}_N)), z_N^{\mathrm{pr}} \rangle - \langle (B_N^{\mathrm{pr}})(\tilde{A}(\hat{u}_N)), z_N^{\mathrm{pr}} \rangle| \leq \|B_N^{\mathrm{pr}}\| |\langle A(\hat{u}_N), z_N^{\mathrm{pr}} \rangle - \langle \tilde{A}(\hat{u}_N), z_N^{\mathrm{pr}} \rangle|.$$

We use constraint (4.28) and equation (4.34) to bound the second term

$$\text{(II)} \leq \|B_N^{\mathrm{pr}}\| |\langle A(\hat{u}_N), z_N^{\mathrm{pr}} \rangle - \langle \tilde{A}(\hat{u}_N, z_N^{\mathrm{pr}} \rangle| \leq \delta_r \delta_J. \tag{4.35}$$

We rewrite the third term of (4.33)

$$\text{(III)} = |\langle (B_N^{\text{pr}})(DA(\hat{u}_N)(\delta\hat{u}_N)), z_N^{\text{pr}}\rangle| \leq \|B_N^{\text{pr}}\|\|DA(\hat{u}_N)(\delta\hat{u})\|(\delta\hat{u}, z_N^{\text{pr}})_{\mathscr{V}},$$

where we define the norm of $DA(\hat{u}_N) : \mathscr{V}_N \to \mathscr{V}_N'$ as

$$\|DA_N^{\text{pr}}\| \equiv \sup_{w\in\mathscr{V}_N}\sup_{v\in\mathscr{V}_N}\frac{\langle DAw, v\rangle}{(w,v)_{\mathscr{V}}}.$$

We use the Cauchy-Schwarz inequality to write $(\delta\hat{u}, z_N^{\text{pr}})_{\mathscr{V}} \leq \|\delta\hat{u}\|_{\mathscr{V}}\|z_N^{\text{pr}}\|_{\mathscr{V}} = \hat{\delta}\|z_N^{\text{pr}}\|_{\mathscr{V}}$, which we combine with constraint (4.28) to write

$$\text{(III)} \leq \|B_N^{\text{pr}}\|\|DA(\hat{u}_N)(\delta\hat{u})\|(\delta\hat{u}, z_N^{\text{pr}})_{\mathscr{V}} \leq \|DA(\hat{u}_N)(\delta\hat{u})\|\|z_N^{\text{pr}}\|_{\mathscr{V}}\delta_J\hat{\delta}. \tag{4.36}$$

We substitute (4.34), (4.35), and (4.36) into (4.33) to get

$$\begin{aligned}|J(u_N) - J(\tilde{u}_N)| \leq &\text{(I)}+ \leq \text{(II)} + \text{(III)} \leq \delta_r + \delta_r\delta_J + \|DA(\hat{u}_N)(\delta\hat{u})\|\|z_N^{\text{pr}}\|_{\mathscr{V}}\delta_J\hat{\delta}\\ \leq &\delta_r + \mathcal{O}(\delta_r^2) + \mathcal{O}(\delta_J^2) + \mathcal{O}(\hat{\delta}^2),\end{aligned}$$

where the second inequality follows from Young's inequality, resulting in the desired inequality. □

We now seek to make our constraints more conservative. We know that $\tilde{\mathbf{r}}_N^{\text{pr}}, \mathbf{z}_N^{\text{pr}} \in \mathbb{R}^N$ so we can enforce N constraints (one for each entry of $\tilde{\mathbf{r}}_N^{\text{pr}} \circ \mathbf{z}_N^{\text{pr}}$). We write

$$|\int_I \mathbf{z}_N^{\text{pr}T}(\bar{\mathbf{r}}_N^{\text{pr}}(\hat{\mathbf{u}}_N) - \tilde{\bar{\mathbf{r}}}_N^{\text{pr}}(\hat{\mathbf{u}}_N))dt| \leq N\|\int_I \mathbf{z}_N^{\text{pr}} \circ (\bar{\mathbf{r}}_N^{\text{pr}}(\hat{\mathbf{u}}_N) - \tilde{\bar{\mathbf{r}}}_N^{\text{pr}}(\hat{\mathbf{u}}_N))dt\|_\infty,$$

where the inequality follows from $|\mathbf{a}^T\mathbf{b}| \leq N\|\mathbf{a}\circ\mathbf{b}\|_\infty$. Next we split the time-domain I into K_{LP} sub-intervals so that $\{I_k^{\text{LP}} = (t_{k-1}^{\text{LP}}, t_k^{\text{LP}}]\}_{k=1}^{K_{\text{LP}}}$, where $t_0^{\text{LP}} = t_0$ and $t_{K_{\text{LP}}}^{\text{LP}} = T$. We further note that $\Delta t_k = t_k^{\text{LP}} - t_{k-1}^{\text{LP}}$. From the triangle inequality we know that $|\int_I f(t)dt| \leq \sum_{k=1}^{K_{\text{LP}}} |\int_{I_k^{\text{LP}}} f(t)dt|$. We introduce a more conservative version of (4.27)

$$\|\int_{I_k} \mathbf{z}_N^{\text{pr}} \circ (\bar{\mathbf{r}}_N^{\text{pr}}(\hat{\mathbf{u}}_N) - \tilde{\bar{\mathbf{r}}}_N^{\text{pr}}(\hat{\mathbf{u}}_N))dt\|_\infty \leq \frac{\delta_r}{N} \times \frac{\Delta t_k}{T} \quad k = 1, 2, \ldots, K_{\text{LP}}. \tag{4.37}$$

We know that

$$|\int_I \mathbf{z}_N^{\mathrm{pr}T}(\bar{\mathbf{r}}_N^{\mathrm{pr}}(\hat{\mathbf{u}}_N) - \tilde{\bar{\mathbf{r}}}_N^{\mathrm{pr}}(\hat{\mathbf{u}}_N))dt| \leq N\|\int_I \mathbf{z}_N^{\mathrm{pr}} \circ (\bar{\mathbf{r}}_N^{\mathrm{pr}}(\hat{\mathbf{u}}_N) - \tilde{\bar{\mathbf{r}}}_N^{\mathrm{pr}}(\hat{\mathbf{u}}_N))dt\|_\infty$$
$$\leq N \sum_{k=1}^{K_{\mathrm{LP}}} \|\int_{I_k} \mathbf{z}_N^{\mathrm{pr}} \circ (\bar{\mathbf{r}}_N^{\mathrm{pr}}(\hat{\mathbf{u}}_N) - \tilde{\bar{\mathbf{r}}}_N^{\mathrm{pr}}(\hat{\mathbf{u}}_N))dt\|_\infty = \delta_r \sum_{k=1}^{K_{\mathrm{LP}}} \frac{\Delta t_k}{T} = \delta_r,$$

where the first inequality follows from $\mathbf{a}^T\mathbf{b} \leq \|\mathbf{a}\circ\mathbf{b}\|_\infty$ for $\mathbf{a}, \mathbf{b} \in \mathbb{R}^N$, the second inequality follows from the triangle inequality, and the equalities yield from constraint (4.37). We now define the LP procedure LP^r to control the output error due to the approximation of $A(\cdot,\cdot)$ by $\tilde{A}(\cdot,\cdot)$.

Definition 16. (LP^r) The LP procedure LP^r identifies EQP weights $\{\rho_\kappa^r \in \mathbb{R}_{\geq 0}\}_{\kappa=1}^{M^h}$ and controls the output error due to the approximation of $A(\cdot,\cdot)$ by $\tilde{A}(\cdot,\cdot)$. We introduce the surrogate primal solution snapshots $U_{N_{\mathrm{train}}} \equiv \{\hat{\mathbf{u}}_{N,k}(\mu)\}_{\mu\in\Xi_{N_{\mathrm{train}}},0\leq k\leq K}$ and we define the LP procedure $\mathrm{LP}^r(\Xi_{N_{\mathrm{train}}}, U_{N_{\mathrm{train}}}, \delta_r)$. We replace the generic manifold accuracy constraints in Definition 1 from section 3.1 with the LP-admissible version of (4.37), which we write as

$$\|\int_{I_k} [\mathbf{z}_N^{\mathrm{pr}}(\mu,t) \circ (\bar{\mathbf{r}}_N^{\mathrm{pr}}(\hat{\mathbf{u}}_N(\mu,t);\mu,t) - \sum_{\kappa=1}^{M^h} \rho_\kappa^r \bar{\mathbf{r}}_{N,\kappa}^{\mathrm{pr}}(\hat{\mathbf{u}}_N(\mu,t);\mu,t))]dt\|_\infty \leq \frac{2\delta_r}{3N} \times \frac{\Delta t_k}{T}, \quad (4.38)$$

for $k = 1, 2, \ldots, K_{\mathrm{LP}}$ and $\mu \in \Xi_{N_{\mathrm{train}}}$. These constraints directly control the error in the approximation of $A(\cdot,\cdot)$, which indirectly controls the error in the approximation of $\mathbf{u}_N$.

Remark. (constraint (4.28) for B_N^{pr}) As in [48] (see section 3.3), we choose not to enforce the constraint (4.28) associated with B_N^{pr}. We do not have an algebraic expression for B_N^{pr} so the constraint (4.28) is itself quite abstract. We hope that by controlling the error in the space-time operator $A(\cdot,\cdot)$, we will indirectly control the error in $DA(\cdot,\cdot)$, and thus the operator B_N^{pr}. Moreover, the error associated with $DA(\cdot,\cdot)$ is second-order, as opposed to the first-order error associated with $A(\cdot,\cdot)$.

Remark. (modified adjoint) The steady goal-oriented EQP framework presented in [48] makes use of a modified adjoint (equation 3.7 in section 3.3). The modified adjoint is not necessary for the time-dependent problems that we have considered, nor is it LP admissible. We briefly explain why the modified adjoint is not LP admissible. First we introduce the modified adjoint for time-dependent problems

$$\mathbf{z}_N^{\mathrm{pr,w}}(\mu,t) = \max\{|\mathbf{z}_N^{\mathrm{pr}}(\mu,t)|, z_k^{\mathrm{pr,min}}(\mu)\} \quad \forall t \in I_k, \quad k = 1, 2, \ldots, K_{\mathrm{LP}},$$

where $z_k^{\text{pr,min}}(\mu) \equiv \max_{t\in I_k}\{N^{1/2}\sqrt{\delta_r}\|\mathbf{z}_N^{\text{pr}}(\mu,t)\|_2\}$ for $k = 1, 2, \ldots, K_{\text{LP}}$. We know that

$$\begin{aligned}\|\int_{I_k} \mathbf{z}_N^{\text{pr}} \circ (\bar{\mathbf{r}}_N^{\text{pr}}(\hat{\mathbf{u}}_N) - \tilde{\bar{\mathbf{r}}}_N^{\text{pr}}(\hat{\mathbf{u}}_N))dt\|_\infty &\le \int_{I_k} \|\mathbf{z}_N^{\text{pr}} \circ (\bar{\mathbf{r}}_N^{\text{pr}}(\hat{\mathbf{u}}_N) - \tilde{\bar{\mathbf{r}}}_N^{\text{pr}}(\hat{\mathbf{u}}_N))\|_\infty dt \\ &\le \int_{I_k} \|\mathbf{z}_N^{\text{pr,w}} \circ (\bar{\mathbf{r}}_N^{\text{pr}}(\hat{\mathbf{u}}_N) - \tilde{\bar{\mathbf{r}}}_N^{\text{pr}}(\hat{\mathbf{u}}_N))\|_\infty dt,\end{aligned}$$

where the second inequality follows from the definition of $\mathbf{z}_N^{\text{pr,w}}$. Integrals of the form $\int_{I_k} \|\mathbf{w}_N \circ (\mathbf{f}_N(t) - \tilde{\mathbf{f}}_N(t))\|_\infty dt$ are not LP admissible because they are not linear in their quadrature weights (the value of $\tilde{\mathbf{f}}_N(t)$ depends on the quadrature weights and affects the sign of $\mathbf{w}_N \circ (\mathbf{f}_N(t) - \tilde{\mathbf{f}}_N(t))$. However, we note that $\|\int_{I_k} \mathbf{w}_N \circ (\mathbf{f}_N(t) - \tilde{\mathbf{f}}_N(t))dt\|_\infty$ is LP admissible because the l^∞ norm is applied following integration.

4.4.3 EQP for Output

We now wish to control error source (ii) of (4.26), which corresponds to the error in the approximation of $J(\tilde{u}_N;\mu)$ by $\tilde{J}(\tilde{u}_N;\mu)$. We wish to achieve

$$\begin{aligned}|\int_I [\mathbf{q}_N(\tilde{\mathbf{u}}_N(\mu,t);\mu,t) - \tilde{\mathbf{q}}_N(\tilde{\mathbf{u}}_N(\mu,t);\mu,t)]dt \\ + [\mathbf{q}_{N,T}(\tilde{\mathbf{u}}_N(\mu,T);\mu,T) - \tilde{\mathbf{q}}_{N,T}(\tilde{\mathbf{u}}_N(\mu,T);\mu,T)]| \le \delta_q,\end{aligned}$$

so we introduce constraints in the LP procedure LP^q to control the error in the evaluation of the output functional

Definition 17. (LP^q) The LP procedure LP^q identifies EQP weights $\{\rho_\kappa^q \in \mathbb{R}_{\ge 0}\}_{\kappa=1}^{M^h}$ and controls the error in the evaluation of the output functional. We introduce the primal solution snapshots $\tilde{U}_{N_{\text{train}}} \equiv \{\tilde{\mathbf{u}}_{N,k}(\mu)\}_{\mu\in\Xi_{N_{\text{train}}},0\le k\le K}$ and we define the LP procedure $\text{LP}^q(\Xi_{N_{\text{train}}}, \tilde{U}_{N_{\text{train}}}, \delta_q)$. We replace the generic manifold accuracy constraints in Definition 1 from section 3.1 with the LP-admissible constraints

$$|\int_{I_k} \big(\mathbf{q}_N(\tilde{\mathbf{u}}_N(\mu,t);\mu,t) - \sum_{\kappa=1}^{M^h} \rho_\kappa^q \mathbf{q}_{N,\kappa}(\tilde{\mathbf{u}}_N(\mu,t);\mu,t)\big)dt| \le \frac{\delta_q}{2} \times \frac{\Delta t_k}{T}, \quad k = 1, 2, \ldots, K_{\text{LP}}, \tag{4.39}$$

$$|\mathbf{q}_{N,T}(\tilde{\mathbf{u}}_N(\mu,T);\mu,T) - \sum_{\kappa=1}^{M^h} \rho_\kappa^q \mathbf{q}_{N,T,\kappa}(\tilde{\mathbf{u}}_N(\mu,T);\mu,T)| \le \frac{\delta_q}{2}, \tag{4.40}$$

for $\mu \in \Xi_{N_{\text{train}}}$. It is immediately clear that these constraints control the error in the evaluation of the functional evaluation. We have also followed the same steps as in

section 4.4.2 to make our constraints more conservative.

4.5 *A Posteriori* Error Estimation: DWR EQP

Next we wish to develop an online-efficient error estimate, based on the DWR method, to estimate the error between $\tilde{J}(\tilde{u}_N;\mu)$ and $J(u_h;\mu)$. We estimate this error by solving the dual FE problem and evaluating the DWR error estimate

$$|\tilde{s}_N(\mu) - s_h(\mu)| \approx |\eta_h(\mu)| \equiv |-\int_I \bar{r}(\tilde{u}_N, z_h;\mu,t)dt|, \tag{4.41}$$

for $\tilde{u}_N \in \mathscr{V}_N^{\mathrm{pr}}$ and $z_h \in \mathscr{V}_h$. We take the dual FE snapshots $\{z_{h,k}(\mu) \in \mathcal{V}_h\}_{\mu\in\Xi_{N_{\mathrm{train}}},1\le k\le K}$ and compress them into the $\mathcal{V}$-orthonormal basis $\{\phi_i^{\mathrm{du}}\}_i$ of $\mathcal{V}_N^{\mathrm{du}}$. Then we introduce the dual RB Bochner space $\mathscr{V}_N^{\mathrm{du}} = C^0([0,T];\mathcal{V}_N^{\mathrm{du}}) \cap L^2(I;\mathcal{V}_N^{\mathrm{du}})$. We wish to solve the DWR problem projected onto the dual RB space. We introduce the Jacobian $\mathbf{J}_N^{\mathrm{du}} : \mathbb{R}^N \times \mathcal{D} \times I \to \mathbb{R}^{N\times N}$

$$\mathbf{J}_N^{\mathrm{du}}(\mathbf{w};\mu,t)_{ij} \equiv r_h'(V_N^{\mathrm{pr}}\mathbf{w};\phi_j^{\mathrm{du}},\phi_i^{\mathrm{du}};\mu,t), \quad i,j = 1,2,\ldots,N,$$

and the output gradients $\mathbf{g}_N^{\mathrm{du}} : \mathbb{R}^N \times \mathcal{D} \times I \to \mathbb{R}^N$ and $\mathbf{g}_{N,T}^{\mathrm{du}} : \mathbb{R}^N \times \mathcal{D} \times I \to \mathbb{R}^N$

$$\begin{aligned}\mathbf{g}_N^{\mathrm{du}}(\mathbf{w};\mu,t)_i &\equiv q_h'(V_N^{\mathrm{pr}}\mathbf{w};\phi_i^{\mathrm{du}};\mu,t), \quad i = 1,2,\ldots,N,\\ \mathbf{g}_{N,T}^{\mathrm{du}}(\mathbf{w};\mu,t)_i &\equiv q_{h,T}'(V_N^{\mathrm{pr}}\mathbf{w};\phi_i^{\mathrm{du}};\mu,t), \quad i = 1,2,\ldots,N,\end{aligned}$$

as well as the mass matrix $\mathbf{M}_N^{\mathrm{du}} \in \mathbb{R}^N$

$$(\mathbf{M}_N^{\mathrm{du}})_{ij} \equiv m_h(\phi_j^{\mathrm{du}},\phi_i^{\mathrm{du}}) \quad i,j = 1,2,\ldots,N,$$

We state the dual RB problem, discretized in time with the backward Euler time-marching method: given $\mu \in \mathcal{D}$ and $\tilde{\mathbf{u}}_{N,k} \in \mathbb{R}^N$, find $\mathbf{z}_{N,k}^{\mathrm{du}}(\mu) \in \mathbb{R}^N$ such that

$$\mathbf{M}_N^{\mathrm{du}}\mathbf{z}_{N,K}^{\mathrm{du}} + \mathbf{g}_{N,T}^{\mathrm{du}}(\tilde{\mathbf{u}}_{N,K};\mu,t_K) = 0,$$

and

$$\begin{aligned}\mathbf{M}_N^{\mathrm{du}}(\mathbf{z}_{N,k}^{\mathrm{du}} - \mathbf{z}_{N,k+1}^{\mathrm{du}}) + (t_{k+1}-t_k)\mathbf{J}_N^{\mathrm{du}}(\tilde{\mathbf{u}}_{N,k};\mu,t_k)^T\mathbf{z}_{N,k}^{\mathrm{du}} + \\ (t_{k+1}-t_k)\mathbf{g}_N^{\mathrm{du}}(\tilde{\mathbf{u}}_{N,k};\mu,t_k)) = 0, \quad k = K-1,K-2,\ldots,0.\end{aligned}$$

We introduce the primal residual projected onto the dual space $\mathbf{r}_N^{\mathrm{du}} : \mathbb{R}^N \times \mathcal{D} \times I \to \mathbb{R}^N$

$$\mathbf{r}_N^{\mathrm{du}}(\mathbf{w};\mu,t)_i \equiv r_h(V_N^{\mathrm{pr}}\mathbf{w};\phi_i^{\mathrm{du}};\mu,t), \quad i = 1,2,\ldots,N,$$

and the primal mass matrix projected onto the dual space $\mathbf{M}_N^{\mathrm{du,pr}} \in \mathbb{R}^{N\times N}$

$$(\mathbf{M}_N^{\mathrm{du,pr}})_{ij} \equiv m_h(\phi_j^{\mathrm{pr}},\phi_i^{\mathrm{du}}) \quad i,j = 1,2,\ldots,N.$$

We evaluate the RB DWR error estimate:

$$|\tilde{s}_N(\mu) - s_h(\mu)| \approx \eta_N(\mu) \equiv |-\int_I \mathbf{z}_N^{\mathrm{du}T}(\mathbf{M}_N^{\mathrm{du,pr}}\frac{d\tilde{\mathbf{u}}_N}{dt} + \mathbf{r}_N^{\mathrm{du}}(\tilde{\mathbf{u}}_N;\mu,t)dt|. \tag{4.42}$$

We will now introduce the dual RB-EQP problem. First we introduce the EQP weights to solve the dual problem $\{\rho_\kappa^{\eta,1} \in \mathbb{R}_{\geq 0}\}_{\kappa=1}^{M^h}$. Then we introduce the residual and bilinear forms $\tilde{r}_h^{(\eta,1)} : \mathcal{V}_h \times \mathcal{V}_h \times \mathcal{D} \times I$ and $\tilde{m}_h^{(\eta,1)}(w,v) : \mathcal{V}_h \times \mathcal{V}_h$ such that

$$\tilde{r}_h^{(\eta,1)}(w,v;\mu,t) \equiv \sum_{\kappa=1}^{M^h} \rho_\kappa^{\eta,1} r_\kappa(w,v;\mu,t),$$
$$\tilde{m}_h^{(\eta,1)}(w,v) \equiv \sum_{\kappa=1}^{M^h} \rho_\kappa^{\eta,1} m_\kappa(w,v),$$

and the output forms $\tilde{q}_h^{(\eta,1)} : \mathcal{V}_h \times \mathcal{D} \times I$ and $\tilde{q}_{h,T}^{(\eta,1)} : \mathcal{V}_h \times \mathcal{D} \times I$ such that

$$\tilde{q}_h^{(\eta,1)}(w;\mu,t) \equiv \sum_{\kappa=1}^{M^h} \rho_\kappa^{\eta,1} q_\kappa(w;\mu,t),$$
$$\tilde{q}_{h,T}^{(\eta,1)}(w;\mu,T) \equiv \sum_{\kappa=1}^{M^h} \rho_\kappa^{\eta,1} q_{T,\kappa}(w;\mu,T).$$

We use these forms to introduce the dual RB-EQP Jacobian $\tilde{\mathbf{J}}_N^{\mathrm{du}} : \mathbb{R}^N \times \mathcal{D} \times I \to \mathbb{R}^{N\times N}$ and the mass matrix $\tilde{\mathbf{M}}_N^{\mathrm{du}} \in \mathbb{R}^{N\times N}$, as well as the dual RB-EQP output gradients $\tilde{\mathbf{g}}_N^{\mathrm{du}} : \mathbb{R}^N \times \mathcal{D} \times I \to \mathbb{R}^N$, and $\tilde{\mathbf{g}}_{N,T}^{\mathrm{du}} : \mathbb{R}^N \times \mathcal{D} \times I \to \mathbb{R}^N$:

$$\tilde{\mathbf{J}}_N^{\mathrm{du}}(\mathbf{w};\mu,t)_{ij} \equiv \tilde{r}_h^{(\eta,1)\prime}(V_N^{\mathrm{pr}}\mathbf{w};\phi_j^{\mathrm{du}},\phi_i^{\mathrm{du}};\mu,t), \quad i,j = 1,2,\ldots,N,$$
$$(\tilde{\mathbf{M}}_N^{\mathrm{du}})_{ij} \equiv \tilde{m}_h^{(\eta,1)}(\phi_j^{\mathrm{du}},\phi_i^{\mathrm{du}}) \quad i,j = 1,2,\ldots,N,$$
$$\tilde{\mathbf{g}}_N^{\mathrm{du}}(\mathbf{w};\mu,t)_i \equiv \tilde{q}_h^{(\eta,1)\prime}(V_N^{\mathrm{pr}}\mathbf{w};\phi_i^{\mathrm{du}};\mu,t), \quad i = 1,2,\ldots,N,$$
$$\tilde{\mathbf{g}}_{N,T}^{\mathrm{du}}(\mathbf{w};\mu,t)_i \equiv \tilde{q}_{h,T}^{(\eta,1)\prime}(V_N^{\mathrm{pr}}\mathbf{w};\phi_i^{\mathrm{du}};\mu,t), \quad i = 1,2,\ldots,N.$$

We state the dual RB-EQP problem, discretized in time with the backward Euler time-marching method: given $\mu \in \mathcal{D}$ and $\tilde{\mathbf{u}}_{N,k} \in \mathbb{R}^N$, find $\tilde{\mathbf{z}}^{\mathrm{du}}_{N,k}(\mu) \in \mathbb{R}^N$ such that

$$\tilde{\mathbf{M}}^{\mathrm{du}}_N \tilde{\mathbf{z}}^{\mathrm{du}}_{N,K} + \tilde{\mathbf{g}}^{\mathrm{du}}_{N,T}(\tilde{\mathbf{u}}_{N,K}; \mu, t_K) = 0$$

and

$$\tilde{\mathbf{M}}^{\mathrm{du}}_N(\tilde{\mathbf{z}}^{\mathrm{du}}_{N,k} - \tilde{\mathbf{z}}^{\mathrm{du}}_{N,k+1}) + (t_{k+1} - t_k)\tilde{\mathbf{J}}^{\mathrm{du}}_N(\tilde{\mathbf{u}}_{N,k}; \mu, t_k)^T \tilde{\mathbf{z}}^{\mathrm{du}}_{N,k} + (t_{k+1} - t_k)\tilde{\mathbf{g}}^{\mathrm{du}}_N(\tilde{\mathbf{u}}_{N,k}; \mu, t_k)) = 0, \quad k = K-1, K-2, \ldots, 0.$$

We introduce EQP weights to evaluate the DWR error estimate $\{\rho^{\eta,2}_\kappa \in \mathbb{R}_{\geq 0}\}_{\kappa=1}^{M^h}$. Then we introduce the residual and bilinear forms $\tilde{r}^{(\eta,2)}_h : \mathcal{V}_h \times \mathcal{V}_h \times \mathcal{D} \times I$ and $\tilde{m}^{(\eta,2)}_h(w,v) : \mathcal{V}_h \times \mathcal{V}_h$ such that

$$\tilde{r}^{(\eta,2)}_h(w, v; \mu, t) \equiv \sum_{\kappa=1}^{M^h} \rho^{\eta,2}_\kappa r_\kappa(w, v; \mu, t),$$
$$\tilde{m}^{(\eta,2)}_h(w, v) \equiv \sum_{\kappa=1}^{M^h} \rho^{\eta,2}_\kappa m_\kappa(w, v).$$

We use these form to introduce the RB-EQP residual $\tilde{\mathbf{r}}^{\mathrm{du}}_N : \mathbb{R}^N \times \mathcal{D} \times I \to \mathbb{R}^N$ and the mass matrix $\tilde{\mathbf{M}}^{\mathrm{du,pr}}_N \in \mathbb{R}^{N \times N}$

$$\tilde{\mathbf{r}}^{\mathrm{du}}_N(\mathbf{w}; \mu, t) \equiv \tilde{r}^{(\eta,2)}_h(V^{\mathrm{pr}}_N \mathbf{w}; \mu, t), \quad i = 1, 2, \ldots, N,$$
$$(\tilde{\mathbf{M}}^{\mathrm{du,pr}}_N)_{ij} \equiv \tilde{m}^{(\eta,2)}_h(\phi^{\mathrm{pr}}_j, \phi^{\mathrm{du}}_i), \quad i, j = 1, 2, \ldots, N.$$

We evaluate the RB-EQP DWR error estimate:

$$|\tilde{s}_N(\mu) - s_h(\mu)| \approx \tilde{\eta}_N(\mu) \equiv |-\int_I \tilde{\mathbf{z}}^{\mathrm{du}T}_N (\tilde{\mathbf{M}}^{\mathrm{du,pr}}_N \frac{d\tilde{\mathbf{u}}_N}{dt} + \tilde{\mathbf{r}}^{\mathrm{du}}_N(\tilde{\mathbf{u}}_N; \mu, t) dt|. \tag{4.43}$$

We can fully decouple the cost of solving the dual problem and evaluating the DWR error estimate from $\mathcal{N}_h$ by introducing two additional EQP rules: (i) a rule to solve the dual problem ($\{\rho^{\eta,1}_\kappa \in \mathbb{R}_{\geq 0}\}_{\kappa=1}^{M^h}$), and (ii) a rule to evaluate the DWR error estimate ($\{\rho^{\eta,2}_\kappa \in \mathbb{R}_{\geq 0}\}_{\kappa=1}^{M^h}$). We wish to control the error $|\tilde{\eta}_N(\mu) - \eta_N(\mu)|$, so we take the general form of the DWR error estimate (4.11) and we replace u_n with $\tilde{u}_N \in \mathscr{V}^{\mathrm{pr}}_N$, z with $z^{\mathrm{du}}_N, \tilde{z}^{\mathrm{du}}_N \in \mathscr{V}^{\mathrm{du}}_N$, and $A(\cdot,\cdot)$ with $\tilde{A}(\cdot,\cdot)$ to write

$$\eta_N(\mu) - \tilde{\eta}_N(\mu) = -(\langle A(\tilde{u}_N), z^{\mathrm{du}}_N \rangle - \langle \tilde{A}(\tilde{u}_N), \tilde{z}^{\mathrm{du}}_N \rangle) = -\langle A(\tilde{u}_N), z^{\mathrm{du}}_N \rangle + \langle \tilde{A}(\tilde{u}_N), \tilde{z}^{\mathrm{du}}_N \rangle.$$

We decompose the above expression and use the triangle inequality to write

$$|\eta_N(\mu) - \tilde{\eta}_N(\mu)| = |\underbrace{\langle A(\tilde{u}_N), z_N^{\mathrm{du}} - \tilde{z}_N^{\mathrm{du}}\rangle}_{\text{source (i)}}| + |\underbrace{\langle A(\tilde{u}_N) - \tilde{A}(\tilde{u}_N), z_N^{\mathrm{du}}\rangle}_{\text{source (ii)}}| + |\underbrace{\langle A(\tilde{u}_N) - \tilde{A}(\tilde{u}_N), z_N^{\mathrm{du}} - \tilde{z}_N^{\mathrm{du}}\rangle}_{\text{source (iii)}}|. \quad (4.44)$$

We have three sources of error: (i) the error in the approximation of the adjoint solution $z_N^{\mathrm{du}} \in \mathscr{V}_N^{\mathrm{du}}$, (ii) the error in the approximation of the space-time operator $A(\cdot,\cdot)$, and (iii) the product of these two errors. We will introduce LP-admissible constraints to directly control the error sources (i) and (ii). We hope that by directly controlling error sources (i) and (ii), we will indirectly control error source (iii).

4.5.1 Error in the Dual Solution

We will use the dual-of-the-dual approach (section 4.3) to control error source (i) of (4.44). We recall that the tangent equation arises from applying the DWR method to the dual problem. We substitute the RB forms into the tangent equation (4.12) and we state the RB tangent problem, discretized in time using the backward Euler time-marching method: given $\mu \in \mathcal{D}$, $\tilde{\mathbf{u}}_{N,k} \in \mathbb{R}^N$, find $\boldsymbol{\psi}_{N,k}^{\mathrm{du}} \in \mathbb{R}^N$ such that $\boldsymbol{\psi}_{N,k=0}^{\mathrm{du}} = 0$ and

$$\mathbf{M}_N^{\mathrm{du}}(\boldsymbol{\psi}_{N,k}^{\mathrm{du}} - \boldsymbol{\psi}_{N,k-1}^{\mathrm{du}}) + (t_k - t_{k-1})\mathbf{J}_N^{\mathrm{du}}(\tilde{\mathbf{u}}_{N,k};\mu,t_k)\boldsymbol{\psi}_{N,k}^{\mathrm{du}} = \mathbf{M}_N^{\mathrm{du,pr}}(\tilde{\mathbf{u}}_{N,k} - \tilde{\mathbf{u}}_{N,k-1}) + (t_k - t_{k-1})\mathbf{r}_N^{\mathrm{du}}(\tilde{\mathbf{u}}_{N,k};\mu,t_k), \quad k = 1,2,\ldots,K, \quad (4.45)$$

We substitute RB forms into the equation (4.13) to estimate the error in the the DWR error estimate

$$|\eta_N(\mu) - \eta_h(\mu)| \approx -\int_I \boldsymbol{\psi}_N^{\mathrm{du}T}(-\mathbf{M}_N^{\mathrm{du}}\frac{d\tilde{\mathbf{z}}_N^{\mathrm{du}}}{dt} + \mathbf{J}_N^{\mathrm{du}}(\tilde{\mathbf{u}}_N;\mu,t)^T\tilde{\mathbf{z}}_N^{\mathrm{du}} + \mathbf{g}_N^{\mathrm{du}}(\tilde{\mathbf{u}}_N;\mu,t)dt - \boldsymbol{\psi}_N^{\mathrm{du}T}(\mathbf{M}_N^{\mathrm{du}}\tilde{\mathbf{z}}_N^{\mathrm{du}} + \mathbf{g}_{N,T}^{\mathrm{du}}(\tilde{\mathbf{u}}_N;\mu,t), \quad (4.46)$$

Throughout the rest of this section we drop $\tilde{u}_N$. We re-state the RB-EQP dual relationship in operator form:

$$\langle D\tilde{A}v, \tilde{z}_N^{\mathrm{du}}\rangle + \langle D\tilde{J}, v\rangle = 0 \quad \forall v \in \mathscr{V}_N^{\mathrm{du}},$$

or

$$\langle v, D\tilde{A}^*\tilde{z}_N^{\mathrm{du}}\rangle + \langle D\tilde{J}, v\rangle = 0 \quad \forall v \in \mathscr{V}_N^{\mathrm{du}},$$

where $D\tilde{A}^* : \mathscr{V}_N^{\mathrm{du}} \to \mathscr{V}'^{\mathrm{du}}_N$ is the adjoint operator of $D\tilde{A}$ such that

$$\langle D\tilde{A}v, w\rangle = \langle v, D\tilde{A}^* w\rangle \quad \forall v, w \in \mathscr{V}_N^{\mathrm{du}} \tag{4.47}$$

We introduce the continuous-in-time Jacobians $\bar{\mathbf{J}}_N^{\mathrm{du}}(\mathbf{w};\mu,t) \equiv \mathbf{M}_N^{\mathrm{du}} \frac{d\mathbf{w}}{dt} + \mathbf{J}_N^{\mathrm{du}}(\mathbf{w};\mu,t)$ and $\tilde{\bar{\mathbf{J}}}_N^{\mathrm{du}}(\mathbf{w};\mu,t) \equiv \tilde{\mathbf{M}}_N^{\mathrm{du}} \frac{d\mathbf{w}}{dt} + \tilde{\mathbf{J}}_N^{\mathrm{du}}(\mathbf{w};\mu,t)$, so that we can write

$$\begin{aligned}\langle \psi_N^{\mathrm{du}}, D\tilde{A}^* z_N^{\mathrm{du}}\rangle + \langle D\tilde{J}, \psi_N^{\mathrm{du}}\rangle = \int_I \boldsymbol{\psi}_N^{\mathrm{du}T}(\tilde{\bar{\mathbf{J}}}_N^{\mathrm{du}}(\tilde{\mathbf{u}}_N)^T \mathbf{z}_N^{\mathrm{du}} + \tilde{\mathbf{g}}_N^{\mathrm{du}}(\tilde{\mathbf{u}}_N))dt \\ + [\boldsymbol{\psi}_N^{\mathrm{du}T}(\tilde{\mathbf{M}}_N^{\mathrm{du}} \mathbf{z}_N^{\mathrm{du}} + \tilde{\mathbf{g}}_{N,T}^{\mathrm{du}}(\tilde{\mathbf{u}}_N))]_{t=T}.\end{aligned} \tag{4.48}$$

We introduce a proposition to show that we can control error source (i) of (4.44) by controlling (4.48).

Proposition 18. *(error control for dual solution, $\langle A(\tilde{u}_N), z_N^{\mathrm{du}} - \tilde{z}_N^{\mathrm{du}}\rangle$). Suppose*

$$|\int_I \boldsymbol{\psi}_N^{\mathrm{du}T}(\bar{\mathbf{J}}_N^{\mathrm{du}}(\tilde{\mathbf{u}}_N)^T \mathbf{z}_N^{\mathrm{du}} - \tilde{\bar{\mathbf{J}}}_N^{\mathrm{du}}(\tilde{\mathbf{u}}_N)^T \mathbf{z}_N^{\mathrm{du}} + \mathbf{g}_N^{\mathrm{du}}(\tilde{\mathbf{u}}_N) - \tilde{\mathbf{g}}_N^{\mathrm{du}}(\tilde{\mathbf{u}}_N))dt| \leq \frac{\delta_{\eta_1}}{4}, \tag{4.49}$$

$$|[\boldsymbol{\psi}_N^{\mathrm{du}T}(\mathbf{M}_N^{\mathrm{du}} \mathbf{z}_N^{\mathrm{du}} - \tilde{\mathbf{M}}_N^{\mathrm{du}} \mathbf{z}_N^{\mathrm{du}} + \mathbf{g}_{N,T}^{\mathrm{du}}(\tilde{\mathbf{u}}_N) - \tilde{\mathbf{g}}_{N,T}^{\mathrm{du}}(\tilde{\mathbf{u}}_N))]_{t=T}| \leq \frac{\delta_{\eta_1}}{4}, \tag{4.50}$$

$$\|B_N^{\mathrm{du}}\| \leq \delta_J, \tag{4.51}$$

where we have defined

$$B_N^{\mathrm{du}} \equiv I - DA^*(D\tilde{A}^*)^{-1}, \tag{4.52}$$

and $\|B_N^{\mathrm{du}}\| \equiv \sup_{w \in \mathscr{V}'^{\mathrm{du}}_N} \sup_{v \in \mathscr{V}_N^{\mathrm{du}}} \frac{\langle B_N^{\mathrm{du}} w, v\rangle}{\langle w, v\rangle}$. *Then*

$$|\langle A(\tilde{u}_N), z_N^{\mathrm{du}} - \tilde{z}_N^{\mathrm{du}}\rangle| \leq \delta_{\eta_1} + \mathcal{O}(\delta_{\eta_1}^2) + \mathcal{O}(\delta_J^2), \tag{4.53}$$

Proof. We use the RB-EQP adjoint relationship to solve for $\tilde{z}_N^{\mathrm{du}} = -(D\tilde{A}^*)^{-1} D\tilde{J}$, then for $z_N^{\mathrm{du}}, \tilde{z}_N^{\mathrm{du}} \in \mathscr{V}_N^{\mathrm{du}}$ we write

$$\begin{aligned} z_N^{\mathrm{du}} - \tilde{z}_N^{\mathrm{du}} =& z_N^{\mathrm{du}} + (D\tilde{A}^*)^{-1} D\tilde{J} = (D\tilde{A}^*)^{-1}(D\tilde{A}^* z_N^{\mathrm{du}} + D\tilde{J}), \\ =& (DA^*)^{-1} DA^* (D\tilde{A}^*)^{-1}(D\tilde{A}^* z_N^{\mathrm{du}} + D\tilde{J}), \\ =& (DA^*)^{-1}(I - (I - DA^*(D\tilde{A}^*)^{-1}))(D\tilde{A}^* z_N^{\mathrm{du}} + D\tilde{J}), \\ =& (DA^*)^{-1}(I - B_N^{\mathrm{du}})(D\tilde{A}^* z_N^{\mathrm{du}} + D\tilde{J}). \end{aligned} \tag{4.54}$$

using (4.52). We use the duality pairing to write

$$\begin{aligned}\langle A, z_N^{\mathrm{du}} - \tilde{z}_N^{\mathrm{du}} \rangle &= \langle A, (DA^*)^{-1}(I - B_N^{\mathrm{du}})(D\tilde{A}^* z_N^{\mathrm{du}} + D\tilde{J}) \rangle, \\ &= \langle (I - B_N^{\mathrm{du}})(D\tilde{A}^* z_N^{\mathrm{du}} + D\tilde{J}), DA^{-1}A \rangle,\end{aligned}$$

where the second equality results from the definition of the dual operator in (4.47). We recall the tangent relationship $\langle DA\psi_N^{\mathrm{du}}, v\rangle - \langle A, v\rangle = 0 \ \forall v \in \mathscr{V}_N^{\mathrm{du}}$ and we note that the operation $DA^{-1}A$ solves for $\psi_N^{\mathrm{du}} \in \mathscr{V}_N^{\mathrm{du}}$. This allows us to write

$$\langle A, z_N^{\mathrm{du}} - \tilde{z}_N^{\mathrm{du}} \rangle = \langle (I - B_N^{\mathrm{du}})(D\tilde{A}^* z_N^{\mathrm{du}} + D\tilde{J}), \psi_N^{\mathrm{du}} \rangle. \tag{4.55}$$

We use the triangle inequality and the definition of $\|B_N^{\mathrm{du}}\|$ to write

$$|\langle A(\tilde{u}_N), z_N^{\mathrm{du}} - \tilde{z}_N^{\mathrm{du}} \rangle| \leq (1 + \|B_N^{\mathrm{du}}\|) |\langle D\tilde{A}^* z_N^{\mathrm{du}} + D\tilde{J}, \psi_N^{\mathrm{du}} \rangle|. \tag{4.56}$$

We apply constraints (4.49) and (4.50) to $|\langle D\tilde{A}^* z_N^{\mathrm{du}} + D\tilde{J}, \psi_N^{\mathrm{du}} \rangle|$

$$\begin{aligned}|\langle D\tilde{A}^* z_N^{\mathrm{du}} + D\tilde{J}, \psi_N^{\mathrm{du}} \rangle| &= \underbrace{|\langle DA^* z_N^{\mathrm{du}} + DJ, \psi_N^{\mathrm{du}} \rangle}_{=0} + \langle D\tilde{A}^* z_N^{\mathrm{du}} + D\tilde{J}, \psi_N^{\mathrm{du}} \rangle| \\ &\leq |\int_I \boldsymbol{\psi}_N^{\mathrm{du}T} (\bar{\mathbf{J}}_N^{\mathrm{du}}(\tilde{\mathbf{u}}_N)^T \mathbf{z}_N^{\mathrm{du}} + \mathbf{g}_N^{\mathrm{du}}(\tilde{\mathbf{u}}_N)) dt| + |\boldsymbol{\psi}_N^{\mathrm{du}T} (\mathbf{M}_N^{\mathrm{du}} \mathbf{z}_N^{\mathrm{du}}) + \mathbf{g}_{N,T}^{\mathrm{du}}(\tilde{\mathbf{u}}_N)|_{t=T} \\ &\leq \frac{\delta_{\eta_1}}{4} + \frac{\delta_{\eta_1}}{4} = \frac{\delta_{\eta_1}}{2},\end{aligned} \tag{4.57}$$

where the first inequality follows from the triangle inequality, while the second inequality follows from the constraints we have imposed. We combine the constraint (4.51) with equation (4.57) to bound (4.56) and prove (4.53)

$$(1 + \|B_N^{\mathrm{du}}\|) |\langle D\tilde{A}^* z_N^{\mathrm{du}} + D\tilde{J}, \psi_N^{\mathrm{du}} \rangle| \leq (1 + \delta_J) \frac{\delta_{\eta_1}}{2} \leq \delta_{\eta_1} + \mathcal{O}(\delta_{\eta_1}^2) + \mathcal{O}(\delta_J^2), \tag{4.58}$$

where the final inequality follows from Young's inequality. □

We follow the same procedures as in section 4.4 to make constraints (4.49) and (4.50) more conservative by (i) enforcing a set of N constraints using the Hadamard product, (ii) splitting I into sub-intervals $\{I_k^{\mathrm{LP}}\}_{k=1}^{K_{\mathrm{LP}}}$, and (iii) splitting the constraints into constraints associated with the gradient of the residual and constraints associated with the gradient of the output (i.e., so that we have separate constraints for the residual gradient and the output gradient). With this in mind we define the LP procedure $\mathrm{LP}^{\eta,1}$, which controls the error $\langle A(\tilde{u}_N, z_N - \tilde{z}_N \rangle$.

Definition 19. ($\text{LP}^{\eta,1}$) The LP procedure $\text{LP}^{\eta,1}$ identifies EQP weights $\{\rho_\kappa^{\eta,1} \in \mathbb{R}_{\geq 0}\}_{\kappa=1}^{M^h}$ and controls the error in the DWR error estimate due to the error $\langle A(\tilde{u}_N), z_N - \tilde{z}_N \rangle$, which is associated with error source (i) of (4.44). We introduce the primal, dual, and tangent solution snapshots $\tilde{U}_{N_{\text{train}}} \equiv \{\tilde{\mathbf{u}}_{N,k}(\mu)\}_{\mu \in \Xi_{N_{\text{train}}}, 0 \leq k \leq K}$, $Z_{N_{\text{train}}} \equiv \{\mathbf{z}_{N,k}(\mu)\}_{\mu \in \Xi_{N_{\text{train}}}}, 0 \leq k \leq K$, and $\Psi_{N_{\text{train}}} \equiv \{\psi_{N,k}(\mu)\}_{\mu \in \Xi_{N_{\text{train}}}}, 0 \leq k \leq K$, as well as the LP procedure $\text{LP}^{\eta,1}(\Xi_{N_{\text{train}}}, \tilde{U}_{N_{\text{train}}}, Z_{N_{\text{train}}}, \Psi_{N_{\text{train}}}, \delta_{\eta_1})$. We replace the generic manifold accuracy constraints in Definition 1 from section 3.1 with the more conservative and LP-admissible versions of (4.49) and (4.50). We introduce the constraints associated with the gradient of the residual:

$$\| \int_{I_k} [\boldsymbol{\psi}_N^{\text{du,w}}(\mu,t) \circ (\bar{\mathbf{J}}_N^{\text{du}}(\tilde{\mathbf{u}}_N(\mu,t);\mu,t)^T \mathbf{z}_N^{\text{du}}(\mu,t) - \sum_{\kappa=1}^{M^h} \rho_\kappa^{\eta,1} \bar{\mathbf{J}}_{N,\kappa}^{\text{du}}(\tilde{\mathbf{u}}_N(\mu,t);\mu,t)^T \mathbf{z}_N^{\text{du}}(\mu,t))] dt \|_\infty \leq \frac{\delta_{\eta_1}}{8N} \times \frac{\Delta t_k}{T} \quad k = 1, 2, \ldots, K_{\text{LP}}, \quad (4.59)$$

and

$$\|\boldsymbol{\psi}_N^{\text{du,w}}(\mu,T) \circ (\mathbf{M}_N^{\text{du}} \mathbf{z}_N^{\text{du}}(\mu,T) - \sum_{\kappa=1}^{M^h} \rho_\kappa^{\eta,1} \mathbf{M}_{N,\kappa}^{\text{du}} \mathbf{z}_N^{\text{du}}(\mu,T))\|_\infty \leq \frac{\delta_{\eta_1}}{8N}. \quad (4.60)$$

We also introduce the constraints associated with the gradient of the output:

$$\| \int_{I_k} \boldsymbol{\psi}_N^{\text{du,w}}(\mu,t) \circ (\mathbf{g}_N^{\text{du}}(\tilde{\mathbf{u}}_N(\mu,t);\mu,t) - \sum_{\kappa=1}^{M^h} \rho_\kappa^{\eta,1} \mathbf{g}_{N,\kappa}^{\text{du}}(\tilde{\mathbf{u}}_N(\mu,t);\mu,t))\|_\infty dt \leq \frac{\delta_{\eta_1}}{8N} \times \frac{\Delta t_k}{T} \quad k = 1, 2, \ldots, K_{\text{LP}}, \quad (4.61)$$

and

$$\|\boldsymbol{\psi}_N^{\text{du,w}}(\mu,T) \circ (\mathbf{g}_{N,T}^{\text{du}}(\tilde{\mathbf{u}}_N(\mu,T);\mu,T) - \sum_{\kappa=1}^{M^h} \rho_\kappa^{\eta,1} \mathbf{g}_{N,T,\kappa}^{\text{du}}(\tilde{u}_N(\mu,T);\mu,T))\|_\infty \leq \frac{\delta_{\eta_1}}{8N}. \quad (4.62)$$

We enforce all constraints for $\mu \in \Xi_{N_{\text{train}}}$.

Remark. We do not enforce constraint (4.51) associated with B_N^{du} for the same reasons as explained previously in section 4.4.2 (i.e., its associated error is second-order and we do not have an algebraic expression for B_N^{du}).

4.5.2 Error in the Space-Time Residual Functional

Error source (ii) of (4.44) corresponds to the error in the approximation of the space-time functional. To simplify the notation, we introduce the continuous-in-time residuals $\bar{\mathbf{r}}_N^{\mathrm{du}}(\mathbf{w};\mu,t) \equiv \mathbf{M}_N^{\mathrm{du,pr}}\frac{d\mathbf{w}}{dt} + \mathbf{r}_N^{\mathrm{du}}(\mathbf{w};\mu,t)$ and $\tilde{\bar{\mathbf{r}}}_N^{\mathrm{du}}(\mathbf{w};\mu,t) \equiv \tilde{\mathbf{M}}_N^{\mathrm{du,pr}}\frac{d\mathbf{w}}{dt} + \tilde{\mathbf{r}}_N^{\mathrm{du}}(\mathbf{w};\mu,t)$. Then we write the error in the space-time residual functional

$$-\langle A(\tilde{u}_N) - \tilde{A}(\tilde{u}_N), z_N^{\mathrm{du}}\rangle = -\int_I \mathbf{z}_N^{\mathrm{du}T}(\bar{\mathbf{r}}_N^{\mathrm{du}}(\tilde{\mathbf{u}}_N) - \tilde{\bar{\mathbf{r}}}_N^{\mathrm{du}}(\tilde{\mathbf{u}}_N))dt.$$

We now introduce a proposition to control error source (ii) of (4.44).

Proposition 20. *(error control for dual residual $\langle A(\tilde{u}_N) - \tilde{A}(\tilde{u}_N), z_N^{\mathrm{du}}\rangle$). Suppose*

$$|\int_I \mathbf{z}_N^{\mathrm{du}T}(\bar{\mathbf{r}}_N^{\mathrm{du}}(\tilde{\mathbf{u}}_N) - \tilde{\bar{\mathbf{r}}}_N^{\mathrm{du}}(\tilde{\mathbf{u}}_N))dt| \leq \frac{\delta_{\eta_2}}{2}, \tag{4.63}$$

then for $\tilde{u}_N \in \mathscr{V}_N^{\mathrm{pr}}$ and $z_N^{\mathrm{du}} \in \mathscr{V}_N^{\mathrm{du}}$ we have

$$|\langle A(\tilde{u}_N) - \tilde{A}(\tilde{u}_N), z_N^{\mathrm{du}}\rangle| \leq \frac{\delta_{\eta_2}}{2}. \tag{4.64}$$

Proof. The inequality (4.64) follows from constraint (4.63)

$$|\langle A(\tilde{u}_N) - \tilde{A}(\tilde{u}_N), z_N\rangle| = |\int_I \mathbf{z}_N^{\mathrm{du}T}(\bar{\mathbf{r}}_N^{\mathrm{du}}(\tilde{\mathbf{u}}_N) - \tilde{\bar{\mathbf{r}}}_N^{\mathrm{du}}(\tilde{\mathbf{u}}_N))dt| \leq \frac{\delta_{\eta_2}}{2},$$

which is the desired result. □

We follow the same procedures as in section 4.4 to make constraint (4.63) more conservative by using the Hadamard product to turn it into N constraints, which we enforce over K_{LP} time-intervals $\{I_k^{\mathrm{LP}}\}_{k=1}^{K_{\mathrm{LP}}}$. We introduce the LP procedure $\mathrm{LP}^{\eta,2}$ to control the error $\langle A(\tilde{u}_N) - \tilde{A}(\tilde{u}_N), z_N^{\mathrm{du}}\rangle$.

Definition 21. ($\mathrm{LP}^{\eta,2}$) The LP procedure $\mathrm{LP}^{\eta,2}$ identifies EQP weights $\{\rho_\kappa^{\eta,2} \in \mathbb{R}_{\geq 0}\}_{\kappa=1}^{M^h}$ and controls the error in the DWR error estimate due to the error $\langle A(\tilde{u}_N) - \tilde{A}(\tilde{u}_N), z_N^{\mathrm{du}}\rangle$, which is associated with error source (ii) of (4.44). We introduce the primal and dual solution snapshots $\tilde{U}_{N_{\mathrm{train}}} \equiv \{\tilde{\mathbf{u}}_{N,k}(\mu)\}_{\mu\in\Xi_{N_{\mathrm{train}}},0\leq k\leq K}$ and $Z_{N_{\mathrm{train}}} \equiv \{\mathbf{z}_{N,k}^{\mathrm{du}}(\mu)\}_{\mu\in\Xi_{N_{\mathrm{train}}},0\leq k\leq K}$, as well as the LP procedure $\mathrm{LP}^{\eta,2}(\Xi_{N_{\mathrm{train}}}, \tilde{U}_{N_{\mathrm{train}}}, Z_{N_{\mathrm{train}}}, \delta_{\eta_2})$. We replace the generic manifold accuracy constraints in Definition 1 from section 3.1 with the more conservative and

LP-admissible version of (4.63), which we write as

$$\| \int_{I_k} [\mathbf{z}_N^{\mathrm{du,w}}(\mu,t) \circ (\bar{\mathbf{r}}_N^{\mathrm{du}}(\tilde{\mathbf{u}}_N(\mu,t);\mu,t) - \sum_{\kappa=1}^{M^h} \rho_\kappa^{\eta,2} \bar{\mathbf{r}}_{N,\kappa}^{\mathrm{du}}(\tilde{\mathbf{u}}_N(\mu,t);\mu,t)]dt\|_\infty \leq \frac{\delta_{\eta_2}}{2N} \times \frac{\Delta t_k}{T}, \tag{4.65}$$

for $k = 1, 2, \ldots, K_{\mathrm{LP}}$ and $\mu \in \Xi_{N_{\mathrm{train}}}$.

4.5.3 Product of DWR Error Sources

In proposition 18 we introduce constraints to control error source (i) of (4.44) ($|\langle A(\tilde{u}_N), z_N^{\mathrm{du}} - \tilde{z}_N^{\mathrm{du}} \rangle|$), while in definition 19 we introduce the LP procedure $\mathrm{LP}^{\eta,1}$ to find EQP weights $\{\rho_\kappa^{\eta,1} \in \mathbb{R}_{\geq 0}\}_{\kappa=1}^{M^h}$ to directly control this error. In proposition 20 we introduce constraints to control error source (ii) of (4.44) ($|\langle A(\tilde{u}_N) - \tilde{A}(\tilde{u}_N), z_N^{\mathrm{du}} \rangle|$), while in definition 21 we introduce the LP procedure $\mathrm{LP}^{\eta,2}$ to find EQP weights $\{\rho_\kappa^{\eta,2} \in \mathbb{R}_{\geq 0}\}_{\kappa=1}^{M^h}$ to directly control this error. We do not directly the errors $\|z_N^{\mathrm{du}} - \tilde{z}_N^{\mathrm{du}}\|_{\mathscr{Y}}$ and $\|A(\tilde{u}_N) - \tilde{A}(\tilde{u}_N)\|_{\mathscr{Y}'}$, whose product corresponds to error source (iii) of (4.44) ($|\langle A(\tilde{u}_N - \tilde{A}(\tilde{u}_N), z_N^{\mathrm{du}} - \tilde{z}_N^{\mathrm{du}}|$).

The steady goal-oriented EQP framework presented in [48] controls the errors in $\|\mathbf{z}_N^{\mathrm{du}} - \tilde{\mathbf{z}}_N^{\mathrm{du}}\|_2$ (see inequality (3.17) in proposition 8 from section 3.4) and $\|\mathbf{r}_N^{\mathrm{du}}(\tilde{\mathbf{u}}_N) - \tilde{\mathbf{r}}_N^{\mathrm{du}}(\tilde{\mathbf{u}}_N)\|_2$ (see inequality (3.20) in proposition 9 from section 3.4). Doing so indirectly controls error source (iii) of (3.9). Controlling error source (iii) is more complicated for time-dependent PDEs because the algebraic manipulations used to control $\|\mathbf{z}_N^{\mathrm{du}} - \tilde{\mathbf{z}}_N^{\mathrm{du}}\|_2$ in the steady context require time-marching in the unsteady context.

We could consider introducing constraints to directly control the global dual error $\|z_N^{\mathrm{du}} - \tilde{z}_N^{\mathrm{du}}\|_{\mathscr{Y}}$ and the error in the residual evaluation $\|A(\tilde{u}_N) - \tilde{A}(\tilde{u}_N)\|_{\mathscr{Y}'}$. We will instead *assume* that the existing constraints control the product of error sources (i) and (ii) reasonably well and that we can iteratively modify δ_{η_1} and δ_{η_2} to achieve the desired error tolerance. Experience tells us that solving the dual problem is more complicated than evaluating the DWR error estimate. Therefore if we wish to achieve $|\tilde{\eta}_N(\mu) - \eta_N(\mu)| \leq \delta_\eta$, then we choose $\delta_{\eta_1} = \delta_\eta$ and then decrease $\delta_{\eta_2} < \delta_\eta$ until we satisfy the desired error tolerance. We are now able to control all three error sources of the DWR error estimate in (4.44) using an iterative procedure.

4.6 Summary

In this chapter we presented a goal-oriented EQP framework for time-dependent problems. In section 4.2 we presented the DWR method for time-dependent problems, while in section 4.3 we demonstrated how we could apply the DWR method to the dual problem

to control the error in the DWR error estimate by solving the tangent problem. Afterward we extended goal-oriented EQP framework to apply to time-dependent problems in section 4.4, while we extended the DWR EQP framework to apply to time-dependent problems in section 4.5. We have presented a full goal-oriented EQP framework for time-dependent problems that includes direct control of the output error and the ability to evaluate the output and estimate its associated error in an online-efficient manner. In chapter 5 we will explain how to implement this framework for time-dependent nonlinear parametrized PDEs, then we will present numerical results to demonstrate this framework in chapter 6.

Chapter 5

Adaptive Offline Training

In chapter 4 we developed a goal-oriented EQP framework for time-dependent problems which we will briefly review here. We wish to evaluate an output and estimate the error in this output in an online-efficient manner. We follow the LP^r procedure in definition 16 from section 4.4.2 to control the error in the approximation of $\mathbf{u}_N$ by $\tilde{\mathbf{u}}_N$, while we follow the LP^q procedure in definition 17 from section 4.4.3 to control the error in the approximation of output functionals $\mathbf{q}_N$ and $\mathbf{q}_{N,T}$ by $\tilde{\mathbf{q}}_N$ and $\tilde{\mathbf{q}}_{N,T}$, respectively. After we solve the primal problem, we follow the $\text{LP}^{\eta,1}$ procedure in definition 19 from section 4.5.1 to control the error in the approximation of $\mathbf{z}_N^{\text{du}}$ by $\tilde{\mathbf{z}}_N^{\text{du}}$, while we follow the $\text{LP}^{\eta,2}$ procedure in definition 21 from section 4.5.2 to control the error in the approximation of $\tilde{\mathbf{r}}_N^{\text{du}}$ by $\tilde{\tilde{\mathbf{r}}}_N^{\text{du}}$.

We seek to develop model reduction methods with training procedures that are automated and efficient. With this in mind, we present a greedy algorithm in section 5.1 that enables us to train our ROMs in an automated and efficient manner. We also present (i) an automated adaptive mesh refinement (AMR) procedure and (ii) an automated method for selecting the uniform time-step Δt in section 5.2. The constraints associated with the LP procedures LP^r, LP^q, $\text{LP}^{\eta,1}$, and $\text{LP}^{\eta,2}$ are designed to be conservative. Since our constraints are conservative, the associated EQP rules may be denser than necessary. In section 5.3 we present an automated method for adaptively selecting the EQP tolerance δ to identify the sparsest quadrature rule possible while still achieving the desired error tolerance. The algorithms we present in this chapter are important to ensuring that our model reduction methods are efficient and automated.

5.1 Greedy Training Algorithm

In this section we present an efficient and automated greedy training algorithm (algorithm 1). The online-efficient DWR error estimate allows the algorithm to identify the

parameter associated with the largest error estimate ($\mu^* = \arg\max_{\mu \in \Xi_{N_{\text{train}}}} |\tilde{\eta}_N(\mu)|$) and is critical to ensuring the efficiency of the offline training procedure. As discussed in section 2.3.1, the greedy algorithm will allow us to reduce the total number of FE solves by identifying a reduced number of parameters required to construct an RB space that spans the parametric and temporal space associated with a given parametrized PDE. This work builds on the work of other researchers who have previously combined POD with greedy algorithms for time-dependent PDEs [22, 25].

In the N_{greedy}-th iteration of algorithm 1, we compute the FE solution at $\mu^{(N_{\text{greedy}})}$ using the adaptive finite element solver described in algorithm 2 (line 3). We construct the primal RB space $\mathcal{V}^{\text{pr}}_{N_{\text{pr}}}$ following POD and we follow an iterative procedure to select the smallest N_{pr} such that we satisfy we satisfy the desired output error tolerance $\max_{\mu \in \Xi_{N_{\text{greedy}}}} |s_N(\mu) - s_h(\mu)| \leq \delta^{\text{rb,pr}}$ (line 4). We follow the LP^r and LP^q procedures outlined in definitions 16 and 17 to select the new EQP weights for the primal problem (line 5). We solve the primal RB-EQP problem to get $\{u_{N,k}(\mu) \in \mathcal{V}^{\text{pr}}_{N_{\text{pr}}}\}_{k=1}^K$ for $\mu \in \Xi_{N_{\text{greedy}}}$, then we solve the dual FE problem linearized about the RB-EQP states to get $\{z_{h,k}(\mu^{(N_{\text{greedy}})}) \in \mathcal{V}_h\}_{k=1}^K$ which allows us to evaluate the FE DWR error estimate $\eta_h(\mu^{(N_{\text{greedy}})})$ (line 6). We use the dual FE snapshots to construct the RB space $\mathcal{V}^{\text{du}}_{N_{\text{du}}}$ using POD such that $\max_{\mu \in \Xi_{N_{\text{greedy}}}} |\eta_N(\mu) - \eta_h(\mu)| \leq \delta^{\text{rb,du}}$ (line 7). Then we select new EQP weights for the DWR error estimate following the LP$^{\eta,1}$ and LP$^{\eta,2}$ procedures from definitions 19 and 21 (line 8). We solve the primal RB-EQP problem, the dual RB-EQP problem, and we evaluate the DWR error estimate for the training data (line 9). We use the DWR error estimate to append parameter $\mu^* = \arg\sup_{\mu \in \Xi_{N_{\text{train}}}} |\tilde{\eta}_N(\mu)|$ to $\Xi_{N_{\text{greedy}}}$ (line 10). At each iteration we verify the convergence of the greedy algorithm (line 12). Once the greedy algorithm has converged, we randomly select N_{random} parameters to construct the parameter set $\Xi_{N_{\text{greedy,min}}}$, which combines the random parameters with the existing parameters in $\Xi_{N_{\text{greedy}}}$ (line 19). We use the $\Xi_{N_{\text{greedy,min}}}$ parameters to update the EQP rules (lines 21 and 24). Increasing from N_{greedy} to $N_{\text{greedy,min}}$ training points for the EQP rules improves the performance for a test set where $\Xi_{N_{\text{test}}} \not\subset \Xi_{N_{\text{train}}}$.

5.2 Adaptive Mesh Refinement

We use AMR to speed-up the convergence of our FE methods and to ensure that our FE solution meets the desired error tolerance so that our simulations are reliable. AMR allows us to find an "optimal" FE space that minimizes the number of degrees of freedom required to achieve the desired error tolerance. This is important to our adaptive training procedure because it ensures that we have "good" training snapshots (i.e., the FE error

Algorithm 1: Simultaneous RB-EQP greedy for time-dependent problems

input : training set $\Xi_{N_{\text{train}}} \subset \mathcal{D}$; output tol. δ^{fe}, $\delta^{\text{rb,pr}}$, and $\delta^{\text{rb,du}}$; number of random train params. N_{random}

output: RB spaces $\mathcal{V}^{\text{pr}}_{N_{\text{pr}}}$, $\mathcal{V}^{\text{du}}_{N_{\text{du}}}$; greedy params. $\Xi_{N_{\text{greedy}}} \subset \Xi_{N_{\text{train}}}$; EQP weights $\{\rho^r_\kappa, \rho^q_\kappa, \rho^{\eta,1}_\kappa, \rho^{\eta,2}_\kappa \in \mathbb{R}_{\geq 0}\}_{\kappa=1}^{M^h}$

set $N_{\text{greedy}} = 1$, *choose* $\mu^{(1)}$, *and set* $\Xi_{N_{\text{greedy}}} \equiv \mu^{(1)}$;

while $N_{\text{greedy}} < N_{\text{train}}$ **do**

- *Solve primal FE problem:* solve for $\{u_{h,k}(\mu^{(N_{\text{greedy}})}) \in \mathcal{V}_h\}_{k=1}^K$ so that $\eta_h(\mu^{(N_{\text{greedy}})}) \leq \delta^{\text{fe}}$, adapt mesh and time-step following algorithm 2
- *Construct primal RB space:* set $\mathcal{V}^{\text{pr}}_{N_{\text{pr}}} = \text{POD}_{N_{\text{pr}}}(\{\{u_{h,k}(\mu)\}_{k=1}^K\}_{\mu \in \Xi_{N_{\text{greedy}}}})$, where N_{pr} is chosen so that $\max_{\mu \in \Xi_{N_{\text{greedy}}}} |s_N(\mu) - s_h(\mu)| \leq \delta^{\text{rb,pr}}$
- *Update primal EQP:* update $\{\rho^r_\kappa, \rho^q_\kappa \in \mathbb{R}_{\geq 0}\}_{\kappa=1}^{M^h}$ using LP^r and LP^q
- *Evaluate* $\eta_h(\mu^{(N_{\text{greedy}})})$*:* solve for $\{\tilde{u}_{N,k}(\mu) \in \mathcal{V}^{\text{pr}}_{N_{\text{pr}}}\}_{k=1}^K$ for $\mu \in \Xi_{N_{\text{greedy}}}$, then solve for $\{z_{h,k}(\mu^{(N_{\text{greedy}})}) \in \mathcal{V}_h\}_{k=1}^K$ and eval. $\eta_h(\mu^{(N_{\text{greedy}})})$
- *Construct dual RB space:* set $\mathcal{V}^{\text{du}}_{N_{\text{du}}} = \text{POD}_{N_{\text{du}}}(\{\{z_{h,k}(\mu)\}_{k=1}^K\}_{\mu \in \Xi_{N_{\text{greedy}}}})$ so that $\max_{\mu \in \Xi_{N_{\text{greedy}}}} |\eta_N(\mu) - \eta_h(\mu)| \leq \delta^{\text{rb,du}}$
- *Update dual EQP:* update $\{\rho^{\eta,1}_\kappa, \rho^{\eta,2}_\kappa \in \mathbb{R}_{\geq 0}\}_{\kappa=1}^{M^h}$ using $\text{LP}^{\eta,1}$ and $\text{LP}^{\eta,2}$
- *Bootstrap:* solve for $\{\tilde{u}_{N,k} \in \mathcal{V}^{\text{pr}}_{N_{\text{pr}}}\}_{k=1}^K$, then solve for $\{\tilde{z}_{N,k} \in \mathcal{V}^{\text{du}}_{N_{\text{du}}}\}_{k=1}^K$ linearized about $\{\tilde{u}_{N,k} \in \mathcal{V}^{\text{pr}}_{N_{\text{pr}}}\}_{k=1}^K$, and evaluate $\tilde{\eta}_N(\mu)$ for $\mu \in \Xi_{N_{\text{train}}}$
- *Choose next parameter:* choose $\mu^* = \arg\sup_{\mu \in \Xi_{N_{\text{train}}}} |\tilde{\eta}_{N_{\text{du}}}(\mu)|$

Check convergence and update $\Xi_{N_{\text{greedy}}}$

if $\tilde{\eta}_{N_{\text{du}}}(\mu^*) \leq \delta^{\text{fe}}$ **then**

break

else

Set $N_{\text{greedy}} = N_{\text{greedy}} + 1$

Set $\mu^{(N_{\text{greedy}})} = \mu^*$ and $\Xi_{N_{\text{greedy}}} \equiv [\Xi_{N_{\text{greedy}}}, \mu^{(N_{\text{greedy}})}]$

end

end

- *Randomly select* $\Xi_{N_{\text{greedy min}}} \equiv [\Xi_{N_{\text{greedy}}}, \Xi_{N_{\text{random}}}]$ *for* N_{random}
- *Solve primal RB problem*: solve for $\{u_{N,k}(\mu)\}_{k=1}^K \in \mathcal{V}^{\text{pr}}_N$ for $\mu \in \Xi_{N_{\text{greedy,min}}}$
- *Update primal EQP*: get $\{\rho^r_\kappa, \rho^q_\kappa \in \mathbb{R}_{\geq 0}\}_{\kappa=1}^{M^h}$ using LP^r and LP^q for $\Xi_{N_{\text{greedy,min}}}$
- *Solve primal RB-EQP problem*: solve for $\{\tilde{u}_{N,k}(\mu)\}_{k=1}^K \in \mathcal{V}^{\text{pr}}_N$ for $\mu \in \Xi_{N_{\text{greedy,min}}}$
- *Solve dual RB problem*: solve for $\{z_{N,k}(\mu)\}_{k=1}^K \in \mathcal{V}^{\text{du}}_N$ for $\mu \in \Xi_{N_{\text{greedy,min}}}$
- *Update dual EQP*: get $\{\rho^{\eta,1}_\kappa, \rho^{\eta,2}_\kappa \in \mathbb{R}_{\geq 0}\}_{\kappa=1}^{M^h}$ using $\text{LP}^{\eta,1}$ and $\text{LP}^{\eta,2}$ for $\Xi_{N_{\text{greedy,min}}}$

level is low enough that we are confident that the error in the FE simulation does not introduce error in the RB and RB-EQP simulations). Moreover, AMR ensures that we achieve a high-accuracy simulation using fewer degrees of freedom because we refine the mesh in regions where the error indicators are the largest. The benefits of AMR for complex flows are demonstrated in [20]. AMR has also previously been used in the context of ROM and has been shown to significantly reduce the offline computational cost for RB generation in [45, 1, 46], which combine AMR with efficient *a posteriori* error estimates to enable greedy parameter selection.

We will use DWR as our *a posteriori* error estimator to enable AMR. We will first outline the DWR method for time-dependent FE problems, then we will explain how to use the DWR method in the context of AMR. We will conclude by discussing how we select our uniform time-step.

5.2.1 DWR Method for Time-Dependent Problems

We wish to control error due to the spatial and temporal discretizations. We will control and estimate the spatial error using the DWR method, while we will control the temporal error by iteratively adapting the size of our uniform time-step such that the temporal error is small compared to the spatial error. Since we wish to estimate the spatial error, we must enrich our problem in space. We solve our primal problem in the space $\mathcal{V}_h$ and solve our dual problem in the enriched space $\mathcal{V}_{\hat{h}}$. As described in section 2.4.1, we can either enrich this space by globally increasing the polynomial degree by one (i.e., $\mathcal{V}_{\hat{h}} \equiv \{v \in \mathcal{V} | v \in \mathbb{P}^{p+1}(J_i), i = 1, 2, \ldots, n_e\}$) or by increasing the number of elements (i.e., $\mathcal{V}_{\hat{h}} \equiv \{v \in \mathcal{V} | v \in \mathbb{P}^{p}(J_i), i = 1, 2, \ldots, \hat{n}_e\}$, where $\hat{n}_e > n_e$. We derived the space-time formulation of the DWR method in chapter 4.2. We state the continuous-in-time dual problem: given $\mu \in \mathcal{D}$, and $u_h(\mu, t) \in \mathcal{V}_h$ for $t \in I$, find $z_{\hat{h}}(\mu, t) \in \mathcal{V}_{\hat{h}}$ for $t \in I$ such that

$$-m(\partial_t z_{\hat{h}}(\mu, t), v) + r'(u_h(\mu, t); v, z_{\hat{h}}(\mu, t); \mu, t) + q'(u_h(\mu, t); v; \mu, t) = 0 \quad \forall v \in \mathcal{V}_{\hat{h}}, \tag{5.1}$$

where

$$m(z_{\hat{h}}(\mu, T), v) + q'_T(u_h(\mu, T); v; \mu, T) = 0 \quad \forall v \in \mathcal{V}_{\hat{h}}. \tag{5.2}$$

We discretize in time using the backward Euler time-marching method. We time-march from $t = T$ to $t = t_0$. We state the discrete-in-time dual problem: given $\mu \in \mathcal{D}$ and $u_{h,k}(\mu) \in \mathcal{V}_h$ for $k = 1, 2, \ldots, K$, find $z_{\hat{h},k} \in \mathcal{V}_{\hat{h}}$ such that

$$m(z_{\hat{h},K}(\mu), v) + q'_T(u_{h,K}(\mu); v; \mu, T) = 0 \quad \forall v \in \mathcal{V}_{\hat{h}}, \tag{5.3}$$

and

$$m(z_{\hat{h},k}(\mu) - z_{\hat{h},k+1}, v) + (t_{k+1} - t_k) r'(u_{h,k}(\mu); v, z_{\hat{h},k}; \mu, t_k) + (t_{k+1} - t_k) q'(u_{h,k}(\mu); v; \mu, t_k) = 0, \quad k = K-1, K-2, \ldots, 0 \quad (5.4)$$

We note that our "truth" solution is the solution to the semi-discrete equation that has already been discretized in time. If we choose a sufficiently small time-step, then we can expect the error in time to be small compared to the error in space. We compute the time-averaged error estimate by element

$$\eta^*_{\hat{h},i}(\mu) \equiv \int_I [m_i(\partial_t u_h(\mu,t), z_{\hat{h}}(\mu,t)) + r_i(u_h(\mu,t), z_{\hat{h}}(\mu,t))] dt \quad i = 1, 2, \ldots, \hat{n}_e$$

so that $\vec{\boldsymbol{\eta}}^*_{\hat{h}} \equiv \{\eta^*_{\hat{h},i}\}_{i=1}^{\hat{n}_e} \in \mathbb{R}^{\hat{n}_e}$ and $\eta_{\hat{h}} = \sum_{i=1}^{\hat{n}_e} \eta^*_{\hat{h},i}$.

5.2.2 AMR Algorithm

We present our AMR algorithm in algorithm 2, which will allow us to iteratively refine our mesh until our error estimate converges to the desired error tolerance. First we solve the primal problem on the coarse mesh $\mathcal{V}_h$. Then we prolong the coarse solutions to the uniformly refined meshes $\mathcal{V}_{\hat{h}}$ and solve the dual problem. We compute the DWR error estimate on the fine mesh (line 6) and then check if the error estimate has converged to the desired error tolerance (line 8). If further refinement is required, we restrict the DWR error estimate onto the coarse mesh (line 11) and then mark for refinement the α-fraction of elements (integer value of elements $\lfloor \alpha \times n_e \rfloor$) associated with the largest error estimates (line 12). We isotropically refine the marked elements (line 13) and then we verify that the error estimate is converging (line 14). If the error estimate is not converging, then we decrease the time-step to decrease the temporal error. Once the DWR error estimate has converged to the desired error tolerance, we verify that the spatial error is small by refining in time and solving the primal problem again (line 23). If the temporal error is small, then have properly selected our spatial and temporal discretizations. Otherwise we further decrease the time-step (line 31).

5.3 Adaptive EQP Tolerance Selection

In the goal-oriented EQP framework for time-dependent problems presented in chapter 4, we control only the first-order terms and ignore all higher-order terms. Moreover, we

Algorithm 2: Goal-Oriented AMR

input : coarse space $\mathcal{V}_h$; output tol. δ^{fe}; parameter μ; refinement fraction α; initial time-step Δt_0; max iter. count $\text{iter}_{\max}$

output: refined space $\mathcal{V}_h$; DWR error estimate $\eta_{\hat{h}}(\mu)$; time-step Δt

control the spatial error; set iter $= 0$ *and* $\Delta t = \Delta t_0$;
while iter $<$ iter$_{\max}$ **do**
  - Solve for $\{u_{h,k}(\mu) \in \mathcal{V}_h\}_{k=1}^{K}$.
  - Uniform mesh refinement (choose $\hat{n}_e > n_e$ to get $\mathcal{V}_h \subset \mathcal{V}_{\hat{h}}$)
  - Prolong $\{u_{h,k}(\mu) \in \mathcal{V}_h\}_{k=1}^{K}$ onto $\mathcal{V}_{\hat{h}}$ and solve for $\{z_{\hat{h},k}(\mu) \in \mathcal{V}_{\hat{h}}\}_{k=1}^{K}$
  - Compute the DWR error estimate by element to get the vector

$$\vec{\boldsymbol{\eta}}_{\hat{h}}^{*} \equiv \{\eta_{\hat{h},i}^{*}\}_{i=1}^{\hat{n}_e}$$

  - Evaluate the error estimate $\eta_{\hat{h}}(\mu) = \sum_{i=1}^{\hat{n}_e} \eta_{\hat{h},i}^{*}$
  if $|\eta_{\hat{h}}(\mu)| \leq \delta^{\text{fe}}$ **then**
    break
  end
  - Restrict the vector $\vec{\boldsymbol{\eta}}_{\hat{h}}^{*}$ onto $\mathcal{V}_h$ to get

$$\vec{\boldsymbol{\eta}}_{h}^{*} \equiv \{\eta_{h,i}^{*}\}_{i=1}^{n_e}$$

  - Mark elements: choose $\alpha \times n_e$ largest error indicators $\{|\eta_{h,i}|\}_{i=1}^{n_e}$
  - Refine the marked elements of $\mathcal{V}_h$ to get an adaptively refined mesh
  if iter > 3 **then**
    Compute convergence rate r as a function of $\mathcal{N}_h$
    if $r < 0.5$ **then**
      Decrease time-step so that $\Delta t = \Delta t/2$
    end
  end
  - Set iter $=$ iter $+ 1$
end
verify the temporal error, set iter $= 0$
while iter $<$ iter$_{\max}$ **do**
  - Solve for $u_h(\mu) \in \mathcal{V}_h$ and evaluate the output $s(\mu)$
  - Set $\Delta\hat{t} = \Delta t/2$
  - Solve for $\hat{u}_h(\mu) \in \mathcal{V}_h$ and evaluate the output $\hat{s}(\mu)$
  - Compute the error $\Delta s = s(\mu) - \hat{s}(\mu)$
  if $\Delta s \leq \delta_{\text{fe}}$ **then**
    break
  else
    $\Delta t = \Delta t/2$
  end
end

do not control the values of the operators B_N^{pr} and B_N^{du}. We also take steps to make our constraints more conservative, such as enforcing our constraints over sub-intervals $\{I_k\}_{k=1}^{K_{\text{LP}}}$ of I and by enforcing N constraints using the Hadamard product. With this in mind, we note that the EQP LP constraints enforce an upper bound on the error introduced by the EQP rule, but that this upper bound is not necessarily sharp. Therefore a given EQP rule may be denser than needed to achieve the desired error tolerance. In this section we present an iterative procedure to adaptively select an EQP tolerance to achieve the desired error tolerance.

We wish for our EQP rules to sparse and accurate. An EQP rule is sparse if it has few quadrature points compared to the "truth" quadrature rule, while it is accurate if the error introduced by the EQP rule is small. We wish to find the sparsest EQP rule possible that still achieves the desired error tolerance. Given an error tolerance $\delta^* \in \mathbb{R}_{\geq 0}$, we can measure the error $\delta_{\text{true}}(\delta^*)$. For LP^r we measure the error

$$\delta_{\text{true}}^r(\delta^*) \equiv \max_{\mu \in \Xi_{N_{\text{train}}}} |J(u_N;\mu) - J(\tilde{u}_N;\mu)|,$$

while for LP$^{\eta,1}$ we measure the error

$$\delta_{\text{true}}^{\eta,1}(\delta^*) \equiv \max_{\mu \in \Xi_{N_{\text{train}}}} |\langle A(\tilde{u}_N;\mu), z_N - \tilde{z}_N \rangle|,$$

and for LP$^{\eta,2}$ we measure the error

$$\delta_{\text{true}}^{\eta,2}(\delta^*) \equiv \max_{\mu \in \Xi_{N_{\text{train}}}} |\langle A(\tilde{u}_N;\mu) - \tilde{A}(\tilde{u}_N;\mu), z_N \rangle|.$$

We do not apply the adaptive error tolerance selection procedure to LP^q because these quadrature rules are already very sparse, so it is not important to make them sparser.

Here we will demonstrate how to iteratively solve the LP system to find sparser quadrature rules that still achieve the desired error tolerance. We wish to find a quadrature rule such that $\delta_{\text{true}}(\delta^*) \leq \delta_0$, where δ_0 is the desired error tolerance and δ^* is the error tolerance used to solve the initial LP system. We consider a quadrature rule to be "better" if it has fewer quadrature points while still achieving the desired error tolerance. In general, we expect the error associated with an EQP rule to be larger if it has fewer quadrature points. We consider two error tolerances $0 < \delta_1 < \delta_2$ and their associated quadrature weights $\{\rho_\kappa^1, \rho_\kappa^2 \in \mathbb{R}_{\geq 0}\}_{\kappa=1}^{M^h}$. We assume that if it is possible to achieve a tight error tolerance $\delta_{\text{true}}(\delta_1) \ll \delta_1$ using some set of quadrature points, then it is possible to achieve a looser error tolerance $\delta_{\text{true}}(\delta_2) \leq \delta_2$ using a subset of these

quadrature points. We note that this LP procedure is feasible if $\delta_2 \geq \delta_1$ since we already know that $\delta_{\text{true}}(\delta_1) \leq \delta_2$, meaning that we can achieve the desired error tolerance if $\{\rho_\kappa^2\}_{\kappa=1}^{M^h} = \{\rho_\kappa^1\}_{\kappa=1}^{M^h}$. Given the previous assumptions we expect $\{\rho_\kappa^2\}_{\kappa=1}^{M^h}$ to a subset of $\{\rho_\kappa^1\}_{\kappa=1}^{M^h}$. We take advantage of this nesting property by reconstructing our LP constraints using the EQP rule as opposed to "truth" quadrature rule. Not only does this decrease the cost of constructing the LP constraints, but it also decreases the cost of solving the LP because we have fewer variables.

In algorithm 3 we present the procedure to adaptively select the error tolerance δ. First we verify that the initial quadrature rule satisfies the desired error tolerance so that $\delta_{\text{true}}(\delta_0) < \delta_0$ (line 2). Then we identify an error tolerance $\delta_{\text{tight}}^{\text{bisect}}$ such that $\delta_{\text{true}}(\delta_{\text{tight}}^{\text{bisect}}) > \delta_{\text{low}}$ (line 7). Next we identify an error tolerance $\delta_{\text{loose}}^{\text{bisect}}$ such that $\delta_{\text{true}}(\delta_{\text{loose}}^{\text{bisect}}) < \delta_0$ (line 11). We use the error tolerances $0 < \delta_{\text{tight}}^{\text{bisect}} < \delta_0 < \delta_{\text{loose}}^{\text{bisect}}$ to in conjunction with the bisection method to iteratively identify the loosest δ^* that satisfies our error tolerance $\delta_{\text{true}}(\delta^*)$ so that we achieve the sparsest quadrature rule possible (line 20). Once the algorithm converges we get the sparse quadrature rule $\{\rho_\kappa^* \in \mathbb{R}_{\geq 0}\}_{\kappa=1}^{M^h}$ associated with δ^* such that $\delta_{\text{true}}(\delta^*) < \delta_0$.

5.4 Summary

We have presented a greedy training procedure (algorithm 1 in section 5.1), an AMR scheme coupled with a scheme to adaptively select our uniform time-step (algorithm 2 in section 5.2), and an iterative procedure to adaptively select the EQP error tolerance (algorithm 3 in section 5.3). The greedy training procedure allows us to improve the efficiency of the offline training stage so that we can achieve the desired error tolerance using fewer FE solvers. The AMR scheme allows us to automatically construct a mesh that satisfies a user-specified error tolerance, while the adaptive EQP tolerance selection procedure allows us to identify the sparsest EQP rule possible that still achieves the desired error tolerance. Our offline training procedure is fully automatic and it is adaptive in both space and time. We demonstrate our offline training procedure in chapter 6.

Algorithm 3: Adaptive EQP Tolerance Selection

input : error bounds δ_0 and δ_{low}, where $0 < \delta_{\text{low}} < \delta_0$;
EQP weights $\{\rho_\kappa \in \mathbb{R}_{\geq 0}\}_{\kappa=1}^{M^h}$ (associated with δ_0); train set $\Xi_{N_{\text{train}}} \subset \mathcal{D}$;
output: EQP weights $\{\rho_m^* \in \mathbb{R}_{\geq 0}\}_{\kappa=1}^{M^h}$ (associated with δ^*); error tol. δ^*;
error measure $\delta_{\text{true}}(\delta^*)$

set $\delta^* = \delta^0$, iter $= 0$, $\text{iter}_{\max} = 100$;
verify that system is feasible ;
Compute error measure $\delta_{\text{true}}(\delta_0)$ for $\Xi_{N_{\text{train}}}$ using $\{\rho_\kappa\}_{\kappa=1}^{M^h}$
if $\delta_{\text{true}}(\delta_0) > \delta_0$ **then**
  Error: system is not feasible
else if $\delta_{\text{true}}(\delta_0) > \delta_{\text{low}}$ **then**
  $\delta_{\text{tight}}^{\text{bisect}} = \delta_0$
else
  break: error bounds are satisfied
end
find loose error tolerance $\delta_{\text{loose}}^{\text{bisect}}$;
Find EQP weights $\{\rho_\kappa^* \in \mathbb{R}_{\geq 0}\}_{\kappa=1}^{M^h}$ for δ^*, use $\{\rho_\kappa\}_{\kappa=1}^{M^h}$ and $\Xi_{N_{\text{train}}}$ to construct LP
Compute error measure $\delta_{\text{true}}(\delta^*)$, use $\{\rho_\kappa^*\}_{\kappa=1}^{M^h}$ and $\Xi_{N_{\text{train}}}$
while $\delta_{\text{true}}(\delta^*) < \delta_0$ **do**
  - Increase δ^* (i.e., $\delta^* = 10 \times \delta^*$)
  - Update EQP weights $\{\rho_\kappa^* \in \mathbb{R}_{\geq 0}\}_{\kappa=1}^{M^h}$ for δ^*, use $\{\rho_\kappa\}_{\kappa=1}^{M^h}$ and $\Xi_{N_{\text{train}}}$ to construct LP
  - Compute error measure $\delta_{\text{true}}(\delta^*)$, use $\{\rho_\kappa^*\}_{\kappa=1}^{M^h}$ and $\Xi_{N_{\text{train}}}$
end
set $\delta_{\text{loose}}^{\text{bisect}} = \delta^*$;
find "best" quadrature rule;
while iter $<$ $\text{iter}_{\max}$ **do**
  - Update $\delta^* = (\delta_{\text{tight}}^{\text{bisect}} + \delta_{\text{loose}}^{\text{bisect}})/2$
  - Update EQP weights $\{\rho_\kappa^* \in \mathbb{R}_{\geq 0}\}_{\kappa=1}^{M^h}$ for δ^*, use $\{\rho_\kappa\}_{\kappa=1}^{M^h}$ and $\Xi_{N_{\text{train}}}$ to construct LP
  - Compute error measure $\delta_{\text{true}}(\delta^*)$, use $\{\rho_\kappa^*\}_{\kappa=1}^{M^h}$ and $\Xi_{N_{\text{train}}}$
  if $\delta_{\text{true}}(\delta^*) < \delta_{\text{low}}$ **then**
    $\delta_{\text{tight}}^{\text{bisect}} = \delta^*$
  else if $\delta_{\text{true}}(\delta^*) > \delta_0$ **then**
    $\delta_{\text{loose}}^{\text{bisect}} = \delta^*$
  else
    break
  end
  - Update iteration counter: iter $=$ iter $+ 1$
end

Chapter 6

Numerical Results

In this chapter we present numerical results demonstrating that the model reduction method we have designed is: (i) rapid, (ii) reliable, (iii) efficient, and (iv) automated. We present numerical results for two unsteady problems: (i) a nonlinear reaction diffusion equation in section 6.1, and (ii) a lid-driven cavity flow problem in section 6.2.

6.1 Nonlinear Reaction Diffusion Equation

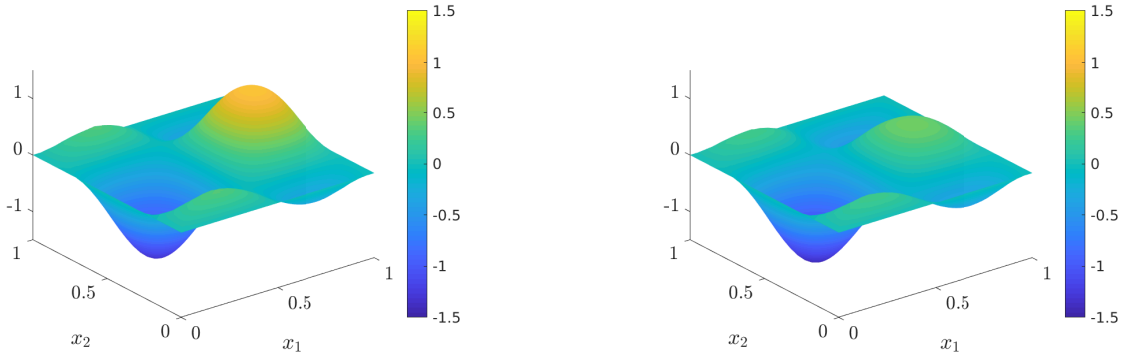
6.1.1 Problem Description

We will now apply EQP to the nonlinear reaction diffusion equation that we presented in section 2.1 and that was originally presented in [23]. For completeness we restate the problem: given any $\mu \in \mathcal{D}$, find $u(\mu) : \Omega \times I \to \mathbb{R}$ that satisfies

$$\begin{aligned} \frac{\partial u(\mu)}{\partial t} - \nabla^2 u(\mu) + g(u(\mu);\mu) = f \quad \text{on} \quad \Omega \times I, \\ u(\mu) = 0 \quad \text{on} \quad \partial\Omega \times I, \\ u(\mu) = 0 \quad \text{on} \quad \partial\Omega \times \{t = 0\}, \end{aligned}$$

where $\Omega \equiv (0,1)^2 \subset \mathbb{R}^2$ is the unit square spatial domain with the boundary $\partial\Omega$, $I \equiv (0, 1.25]$ is the *modified* time interval, and $\mathcal{D} \equiv [0.01, 10]^2 \subset \mathbb{R}^2$ is the parameter domain. The parameter tuple μ has two scalar entries where $\mu = (\mu_{(1)}, \mu_{(2)})$. Moreover, the nonlinear reaction function is defined as

$$g(u(\mu);\mu) = \mu_{(1)} \frac{e^{\mu_{(2)} u(\mu)} - 1}{\mu_{(2)}},$$



(a) $\mu = (10, 0.01)$ (b) $\mu = (10, 10)$

Figure 6.1: Plot of solution to Grepl problem at $t = 1.25$

while the time-dependent forcing function is defined as

$$f((x_{(1)}, x_{(2)}), t) = 100 \sin(2\pi t) \sin(2\pi x_{(1)}) \cos(2\pi x_{(2)}).$$

We compute the *modified* output $s(\mu) = q_T(u(\mu, T); \mu, T)$, where the output functional is the mean value of the solution weighted by a Gaussian distribution, centred about $x = (0.75, 0.05)$ with a standard deviation of $\sigma = 0.05$, so that we localize the output further:

$$q_T(u(\mu, T); \mu, T) \equiv \frac{1}{0.05 \times 2\pi} \int_\Omega \exp\Big(- \Big(\frac{(x_1 - 0.75)^2}{2 \times (0.05)^2} + \frac{(x_2 - 0.5)^2}{2 \times (0.05)^2} \Big) \Big) u(\mu, T) dx.$$

We see that $\mu_{(1)}$ controls the magnitude of the nonlinear reaction term, while $\mu_{(2)}$ controls nonlinearity of the nonlinear reaction term. Small magnitudes of either $\mu_{(1)}$ or $\mu_{(2)}$ (especially $\mu_{(2)}$) make the problem more linear so that the average value of the solution integrated in time approaches zero, which in turn means that the magnitude of the output approaches zero In figure 6.1 we plot the solution at $t = 1.25$ for $\mu = (10, 0.01)$ (slightly nonlinear) and $\mu = (10, 10)$ (highly nonlinear). We notice that the nonlinear reaction term damps the peaks but not the troughs. We also notice that the damping behaviour is more prominent for the most nonlinear case.

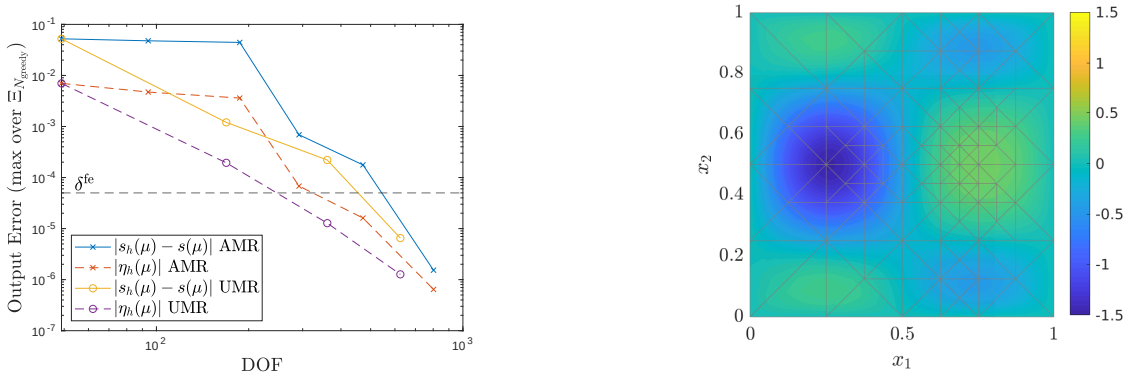
We will train our ROM following the greedy algorithm (algorithm 1) outlined in chapter 5.1. We will demonstrate the we have direct numerical control of the error introduce by EQP and that the online stage is rapid and reliable. We will further demonstrate that the greedy algorithm is efficient by showing that it reduces the offline training costs by comparing the performance of greedy and random parameter selection.

Following [23], we choose a uniform 12×12 grid of training points $\Xi_{N_{\text{train}}}$ and a uniform 15×15 grid of test points $\Xi_{N_{\text{test}}}$. We choose our first parameter to be $\mu^{(1)} = \arg\max_{\mu \in \Xi_{N_{\text{train}}}} (\mu_{(1)}^2 + \mu_{(2)}^2) = [10, 10]$, which is the most nonlinear $\mu \in \mathcal{D}$. We further note that we chose the FE error tolerance $\delta^{\text{fe}} = 5 \times 10^{-5}$, the RB error tolerances $\delta^{\text{rb,pr}} = 10 \times \delta^{\text{fe}} = 5 \times 10^{-4}$ and $\delta^{\text{rb,du}} = 0.25 \times \delta^{\text{rb,pr}} = 1.25 \times 10^{-4}$. We also note that we wish for the EQP rules to introduce only a small additional error. Therefore we choose $\delta^r = 5 \times 10^{-5}$, $\delta^q = 10^{-8}$, $\delta^{\eta_1} = 1.25 \times 10^{-5}$, and $\delta^{\eta_2} = 1.25 \times 10^{-6}$. Finally we note that we choose an initial mesh with third-order polynomials and $\mathcal{N}_h = 49$, while our initial time-step is $\Delta t = 0.0025$. After the greedy algorithm has converged we seek to improve the accuracy of the EQP rule by randomly select $N_{\text{random}} = 10$ parameters to create $\Xi_{N_{\text{greedy,min}}} \equiv [\Xi_{N_{\text{greedy}}}, \Xi_{N_{\text{random}}}]$, then we use these parameters to retrain our EQP rules.

6.1.2 Output Prediction

Here we present numerical results for the output error. We note that the greedy algorithm converges after four iterations so that $N_{\text{greedy}} = 4$. The largest magnitude of the output is associated with $\mu^{(2)} \equiv [0.01, 0.01]$ (the most linear case) where $s(\mu^{(2)}) = 0.156$. We define the relative output error $\text{err}_{s,\text{rel}}(\mu) \equiv \frac{|s_h(\mu) - s(\mu)|}{|s(\mu)|} \approx |\frac{\eta_h(\mu)}{s_h(\mu)}|$. Our choice of δ^{fe} means that the largest relative output error satisfies $\max_{\mu \in \mathcal{D}} \text{err}_{s,\text{rel}}(\mu) \leq \frac{5 \times 10^{-5}}{0.156} = 2.07 \times 10^{-4} = 0.032\% < 0.1\%$. We invoke AMR (algorithm 2 in section 5.2) to automatically refine our initial mesh and uniform time-step to get a mesh whose "truth" quadrature rule has $M^h = 2064$ points and the uniform time-step $\Delta t = 0.0025$. We verify that the FE solver achieves the desired error tolerance for all parameters $\mu \in \Xi_{N_{\text{greedy}}}$ and we find that $\max_{\mu \in \Xi_{N_{\text{greedy}}}} |\eta_h(\mu)| = 5.40 \times 10^{-6} < \delta^{\text{fe}} = 5 \times 10^{-5}$. In figure 6.2a we compare the convergence of AMR to uniform mesh refinement. We overlay the error estimate $|\eta_h(\mu)|$ with the "true" error $|s_h(\mu) - s_{\text{ref}}(\mu)|$, where $s_{\text{ref}}(\mu)$ is the reference output computed on the reference mesh, which we get by uniformly uniformly refining the finest mesh from AMR twice. We see that the DWR error estimate closely tracks the true error as $\mathcal{N}_h$ increases (i.e., it's effectivity increases as $\mathcal{N}_h$ increases). In figure 6.2b we plot the AMR mesh overlayed with the solution for $\mu = (10, 10)$ at $t = 1.25$. We see that the mesh is more refined towards $x = (0.25, 0.5)$ and $x = (0.75, 0.5)$, which have large peaks and troughs, and thus higher solution gradients. In particular, the mesh is even more refined towards $x = (0.75, 0.5)$, which is the centre of the Gaussian distribution of the output and corresponds to the centre of one of the peaks / troughs.

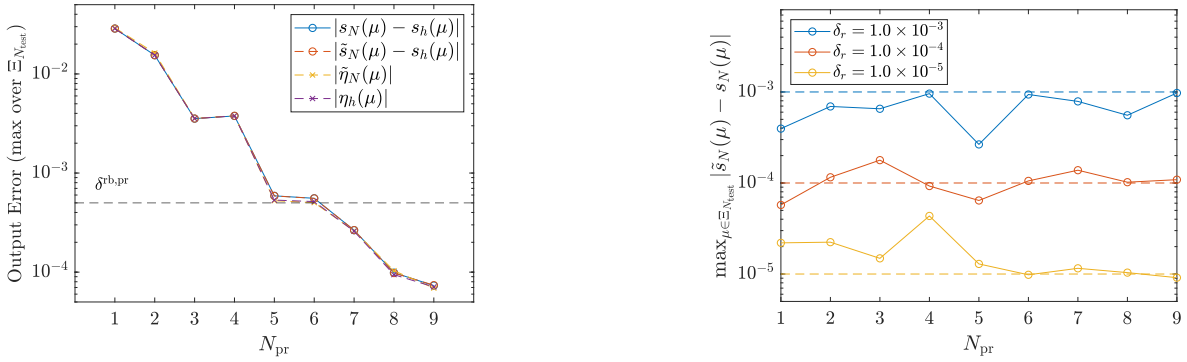
Next we demonstrate direct numerical control over the RB and RB-EQP output



(a) FE convergence error

(b) Mesh

Figure 6.2: AMR for Grepl problem



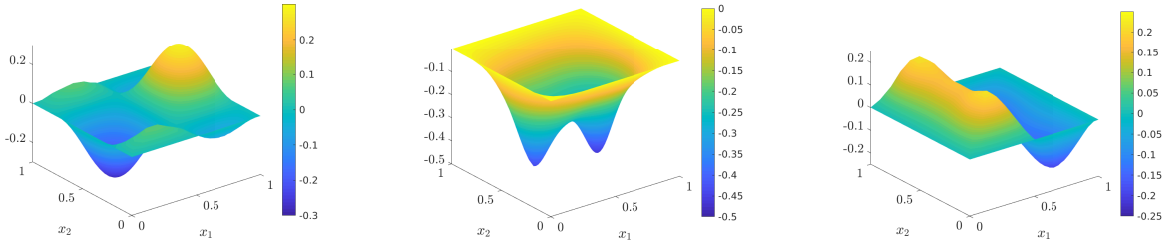
(a) RB convergence

(b) RB-EQP error control

Figure 6.3: Convergence of output error (train with $\Xi_{N_{\text{greedy,min}}}$ and test over $\Xi_{N_{\text{test}}}$)

errors. We demonstrate the RB convergence of the output error and the associated DWR error estimate in figure 6.3a, while we demonstrate direct numerical control of the error due to EQP in 6.3b. We see that the error between the RB and FE outputs decreases as we add more primal basis functions and that the error between the RB and RB-EQP outputs decreases as we decrease δ_r. The nonlinear reaction diffusion equation becomes linear if we set $\mu_{(1)} = \mu_{(2)} = 0$. We can solve the linear problem using only one sinusoidal RB function, but we require $\{\phi_i\}_{i>1}$ to resolve the nonlinearity when $\mu_{(1)} > 0$ and $\mu_{(2)} > 0$. In figure 6.4 we plot the first three primal RB functions and we notice that the first RB function is similar in shape to the solutions plotted in figure 6.1.

Using the greedy algorithm we accurately approximate the primal RB problem using $N_{\text{pr}} = 9$ with $M^r = 79$ and $M^q = 10$. We measure the output error in the test data



(a) First RB function (b) Second RB function (c) Third RB function

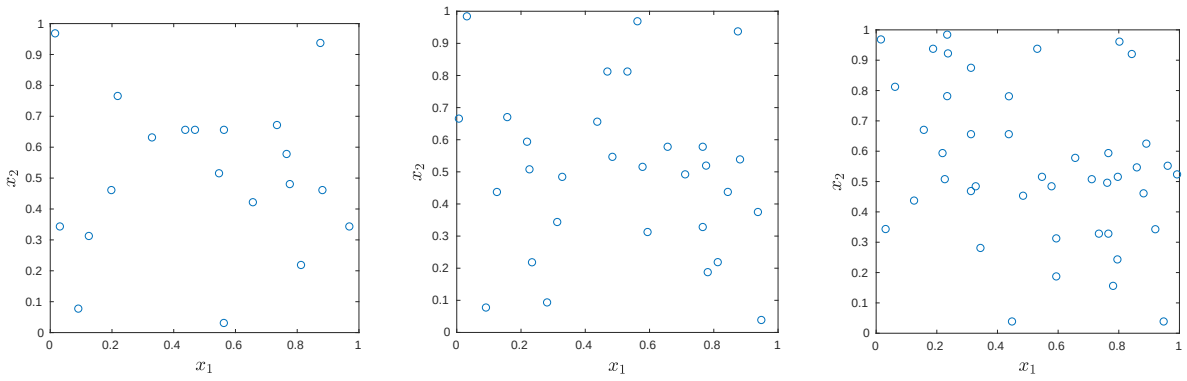
Figure 6.4: Primal RB functions for Grepl problem

δ_0	M_0^r	$\text{err}_{r,0}$	δ^*	M^r	err_r
10^{-3}	37	3.62×10^{-5}	1.00×10^{-2}	20	9.74×10^{-4}
10^{-4}	55	4.47×10^{-6}	1.78×10^{-3}	30	9.20×10^{-5}
10^{-5}	79	8.27×10^{-7}	1.78×10^{-4}	43	9.74×10^{-6}

Table 6.1: Adaptive EQP tolerance selection for LP^r - Grepl problem

$\max_{\mu \in \Xi_{N_{\text{test}}}} |\tilde{s}_N(\mu) - s_h(\mu)| = 7.35 \times 10^{-5} < \delta^{\text{rb,pr}} = 5 \times 10^{-4}$ and $\max_{\mu \in \Xi_{N_{\text{test}}}} |\tilde{s}_N(\mu) - s_N(\mu)| = 1.88 \times 10^{-6} < \delta^r = 1 \times 10^{-5}$. We note that we are most concerned with controlling the error between the RB-EQP and the FE outputs. We are not concerned with the magnitude of the error between the RB-EQP and RB outputs as long as this error is small compared to the error between the RB-EQP and FE outputs.

Here we demonstrate the adaptive EQP tolerance selection algorithm (algorithm 3 in section 5.3). In figure 6.5 we look at how changing $\delta_{r,0}$ affects the number of quadrature points associated with LP^r for $N_{\text{pr}} = 9$. We measure the error $\text{err}_r \equiv \max_{\mu \in \Xi_{N_{\text{greedy}}}} |\tilde{s}_N(\mu) - s_N(\mu)|$, while we set the lower bound $\delta_{\text{low}} = 0.8 \times \delta_0$. In table 6.1 we show that given δ_0, we can find $\delta^* > \delta_0$ such that the associated quadrature rule is sparser, yet still achieves the desired error tolerance. This again demonstrates quantitative control of the EQP error. We also note that the initial EQP rules without the adaptive tolerance selection are conservative. This is consistent with the theory in section 4.4 because we have intentionally chosen to impose constraints that are more conservative than necessary. These results also emphasize the adaptivity of our training procedure.



(a) $\delta_{r,0} = 10^{-3}$, $M^r = 20$ (b) $\delta_{r,0} = 10^{-4}$, $M^r = 30$ (c) $\delta_{r,0} = 10^{-5}$, $M^r = 43$

Figure 6.5: LP^r EQP points for Grepl problem ($N_{\text{pr}} = 9$ and $\Xi_{N_{\text{greedy,min}}}$)

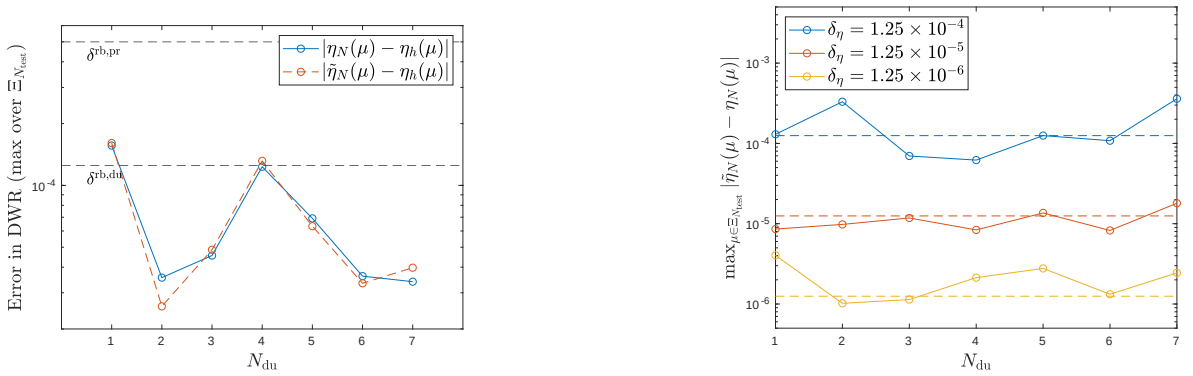
6.1.3 Error Estimation

Here we present results for the DWR error estimate. In figure 6.6a we see that the error in the RB and RB-EQP DWR error estimates, measured with respect to the FE DWR error estimate, converges with N_{du}. We combine the LP procedures presented in definitions 19 and 21 and enforce the constraints simultaneously for δ_η. In figure 6.6b, we demonstrate that decreasing δ_η decreases the error in the DWR error estimate, meaning that we have direct numerical control over the error in the DWR error estimate due to EQP. Following the greedy algorithm, we achieve the desired error tolerance using $N_{\text{du}} = 7$ with $M^{\eta,1} = 27$ and $M^{\eta,2} = 81$. We measure the error in the DWR error estimate for the test data $\max_{\mu\in\Xi_{N_{\text{test}}}} |\tilde{\eta}_N(\mu) - \eta_h(\mu)| = 3.40 \times 10^{-5} < \delta^{\text{rb,du}} = 1.25 \times 10^{-4}$ and $\max_{\mu\in\Xi_{N_{\text{test}}}} |\tilde{\eta}_N(\mu) - \eta_N(\mu)| = 4.43 \times 10^{-7} < \max(\delta^{\eta_1}, \delta^{\eta_2}) = 1.25 \times 10^{-5}$. We see that we have precise control of the error in the DWR error estimate.

Here we demonstrate the adaptive EQP tolerance selection algorithm (algorithm 3 in section 5.3) for LP$^{\eta_1}$. In figure 6.7 we see that using larger $\delta_{\eta_1,0}$ leads to sparser quadrature rules. We measure the error in LP$^{\eta_1}$

$$\text{err}_{\eta_1} \equiv \max_{\mu\in\Xi_{N_{\text{greedy}}}} \| \int_I \bar{\mathbf{r}}_N^{\text{du}}(\tilde{\mathbf{u}}_N(\mu);\mu,t)^T(\tilde{\mathbf{z}}_N^{\text{du}}(\mu) - \mathbf{z}_N^{\text{du}}(\mu))dt\|_2,$$

while we set the lower bound $\delta_{\text{low}} = \delta_{\eta_1,0}/5$. In table 6.2 we see that we get sparser quadrature rules using the adaptive EQP tolerance selection procedure. In figure 6.7 we notice that the EQP points are clustered around $x = (0.75, 0.5)$, which is the centre of the Gaussian distribution for the output.



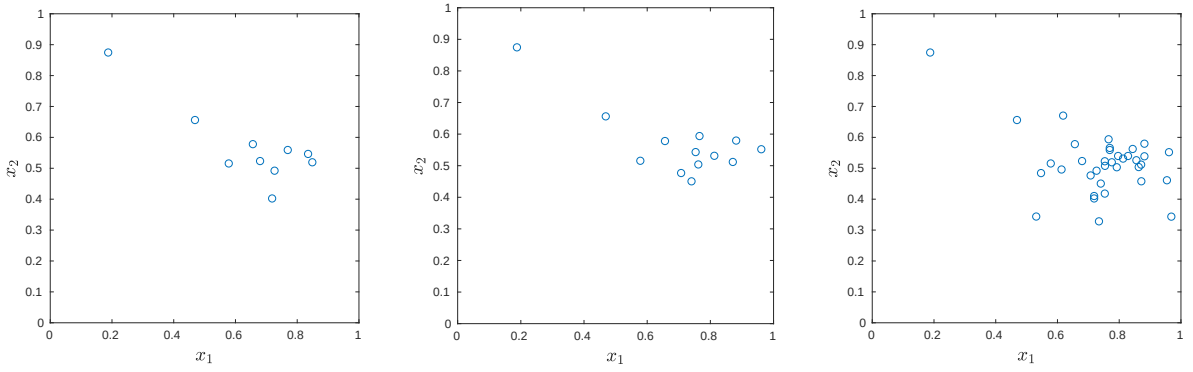
(a) RB convergence

(b) RB-EQP error control

Figure 6.6: Convergence of the error in the DWR error estimate (train with $\Xi_{N_{\text{greedy,min}}}$ and test over Ξ_{test})

δ_0	M_0^r	$\text{err}_{r,0}$	δ^*	M^r	err_r
1.25×10^{-4}	17	4.97×10^{-6}	2.00×10^{-3}	10	6.78×10^{-5}
1.25×10^{-5}	27	1.522×10^{-6}	4.61×10^{-4}	13	1.37×10^{-6}
1.25×10^{-6}	36	4.65×10^{-7}	1.25×10^{-6}	36	4.65×10^{-7}

Table 6.2: Adaptive EQP tolerance selection for LP^{η_1} - Grepl problem



(a) $\delta_{\eta_1,0} = 1.25 \times 10^{-4}$, $M^{\eta_1} = 10$

(b) $\delta_{\eta_1,0} = 1.25 \times 10^{-5}$, $M^{\eta_1} = 22$

(c) $\delta_{\eta_1,0} = 1.25 \times 10^{-6}$, $M^{\eta_1} = 38$

Figure 6.7: $\text{LP}^{\eta,1}$ EQP points for Grepl problem ($N_{\text{du}} = 7$, and $\Xi_{N_{\text{greedy,min}}}$)

	LP^r	LP^q	LP^{η_1}	LP^{η_2}	total
Construct	0.989	0.256	0.930	0.981	3.16
Solve	0.0388	0.0052	0.0023	0.0022	0.0485

Table 6.3: Normalized offline time for LP_{EQP} for $\Xi_{N_{\text{greedy}}}$

6.1.4 Offline Computational Cost

We will now discuss the cost of the greedy offline training. The primary training costs are associated with (i) solving the primal and dual FE problems, (ii) following the LP procedures to construct the EQP rules for $\Xi_{N_{\text{greedy}}}$, (iii) evaluating the ROMs for $\mu \in \Xi_{N_{\text{train}}}$ in the bootstrapping stage, and (iv) following the LP procedures to construct the EQP rules for $\Xi_{N_{\text{greedy,min}}}$. We also note that the initial AMR procedure to determine the spatial and temporal discretizations is more costly than subsequent FE solves, so we will separate this cost from the cost of the other FE solves. We normalize all times using the time for one primal FE solve for $\mu = (10, 10)$.

The initial AMR procedure entails a normalized cost of $\hat{t}_{\text{fe,amr}} = 9.68$, while all subsequent FE solves have a combined cost of (i) $\hat{t}_{\text{fe,pr}} = 3.00$ to solve the primal problem, (ii) $\hat{t}_{\text{fe,dwr}} = 5.42$ to verify that the desired error tolerance is satisfied (i.e., solve dual problem on a uniformly refined mesh and evaluate the DWR error estimate), and (iii) $\hat{t}_{\text{fe,du}} = 1.97$ to collect snapshots for the dual RB space by solving the dual FE problem (without uniform mesh refinement). Next we perform POD and choose both N_{pr} ($\hat{t}_{N_{\text{pr}}} = 0.124$) and N_{du} ($\hat{t}_{N_{\text{du}}} = 0.386$). After choosing N_{pr} and N_{du} we must solve the dual problem projected onto the primal RB space ($\hat{t}_{\text{rb},\beta,\text{pr}} = 0.714$) and the tangent problem projected onto the dual RB space ($\hat{t}_{\text{rb},\chi,\text{du}} = 1.28$). A break-down of the costs to construct the constraints and solve the LP for all LP procedures is provided table 6.3, and their total cost is $\hat{t}_{\text{eqp,total}} = 3.20$ for $\Xi_{N_{\text{greedy}}}$. After we construct the EQP rules, we solve the primal RB-EQP problem ($\hat{t}_{\text{eqp,boot,pr}} = 14.8$) and evaluate the RB-EQP DWR error estimate ($\hat{t}_{\text{eqp,boot,du}} = 9.51$) for $\mu \in \Xi_{N_{\text{train}}}$, which entails a total cost of $\hat{t}_{\text{eqp,boot,total}} = 15.7$, where $N_{\text{train}} = 144$. The total cost of the greedy training algorithm for four iterations is $\hat{t}_{\text{greedy,total,iter}} = 50.1$, which is substantially lower than performing $N_{\text{train}} = 144$ primal solves, applying POD to these snapshots, and then constructing EQP rules for $N_{\text{train}} = 144$. Moreover, the approach of the greedy algorithm is more reliable than the aforementioned brute force method because it also provides error estimates. This demonstrates the efficiency and reliability of the greedy algorithm.

Once the greedy algorithm has converged, we randomly select $N_{\text{random}} = 10$ parameters to be combined with $\Xi_{N_{\text{greedy}}}$ to create $\Xi_{N_{\text{greedy,min}}}$. We use the RB-EQP states from

	primal	output	DWR (dual)	DWR (eval)	total
FE	1	0.0351	0.477	0.104	1.62
RB	0.215	0.0128	0.0761	0.0625	0.367
RB-EQP	0.0266	0.0036	0.0117	0.0064	0.0483

Table 6.4: Normalized online computational time for Grepl problem

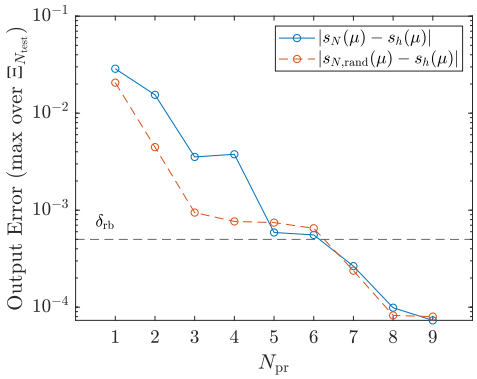
the bootstrapping stage so that we solve the dual problem projected onto the primal RB space ($\hat{t}_{\text{rb,pr,min}} = 0.83$) and the tangent problem projected onto the dual RB space ($\hat{t}_{\text{rb,du,min}} = 1.79$). Then we construct the new EQP rules for a total cost of $\hat{t}_{\text{eqp,total,min}} = 4.92$. The total cost of the steps involving $\Xi_{N_{\text{greedy,min}}}$ is $\hat{t}_{\text{greedy,total,min}} = 7.54$. This brings the total cost of the greedy algorithm to $\hat{t}_{\text{greedy,total}} = \hat{t}_{\text{greedy,total,iter}} + \hat{t}_{\text{greedy,total,min}} = 57.6$, which is still much lower than performing $N_{\text{train}} = 144$ primal solves.

6.1.5 Online Computational Cost

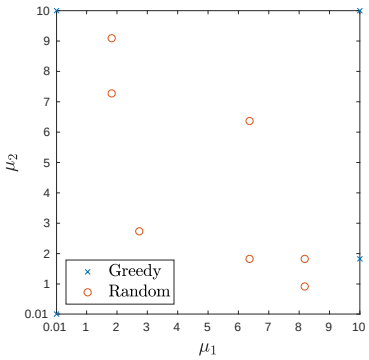
We will now discuss the online computational cost of our ROMs. We will normalize all computational times using the time to perform a one primal finite element solve and present the normalized times in table 6.4. The RB solver the computational cost with respect to the FE solver because the linear solve at each Newton iteration depends on $\mathcal{O}(N)$ rather than $\mathcal{O}(\mathcal{N}_h)$. The FE solver entails a high computational cost because we must perform a large number of sparse operations that cannot be efficiently performed on modern computers. The RB and RB-EQP solvers entail lower costs because we need only perform dense operations. The RB-EQP solver further increases the cost savings of the RB solver by decoupling the cost to evaluate the operators from $\mathcal{O}(\mathcal{N}_h)$. The RB-EQP solver leads to a speed-up of 34.3 to solve the primal problem and evaluate the output, while it leads to a speed-up of 33.1 to solve the dual problem and evaluate the DWR error estimate, with respect to the FE solver. The RB-EQP solver leads to a total speed-up of 33.5 with respect to the FE solver.

6.1.6 Greedy vs. Random Sampling

Greedy sampling converges in four iterations, while random sampling converges in seven iterations. This demonstrates that greedy sampling allows users to construct their ROMs more efficiently. In figure 6.8b we plot the parameters chosen by both greedy and random sampling. We see that greedy sampling preferentially selects parameters at the boundary of $\mathcal{D}$ ($\mu^{(1)} = (10, 10)$, $\mu^{(2)} = (0.01, 0.01)$, $\mu^{(3)} = (10, 1.83)$, and $\mu^{(4)} = (0.01, 10)$).



(a) Primal RB convergence, $N_{\text{greedy}} = 4$



(b) Selected parameters

Figure 6.8: Greedy vs. random parameter selection for Grepl problem

From random sampling we get $N_{\text{pr}} = 7$ with $M^r = 63$ and $M^q = 8$ for the primal problem, while we get $N_{\text{du}} = 2$ with $M^{\eta_1} = 9$ and $M^{\eta_2} = 21$. We measure the output error for random sampling $\max_{\mu \in \Xi_{N_{\text{test}}}} |\tilde{s}_N(\mu) - s_h(\mu)| = 7.03 \times 10^{-5} < \delta^{\text{rb,pr}} = 5 \times 10^{-4}$ and $\max_{\mu \in \Xi_{N_{\text{test}}}} |\tilde{s}_N(\mu) - s_N(\mu)| = 9.66 \times 10^{-6} < \delta^r = 10^{-5}$, while we measure the error in the DWR error estimate $\max_{\mu \in \Xi_{N_{\text{test}}}} |\tilde{\eta}_N(\mu) - \eta_h(\mu)| = 2.35 \times 10^{-5} < \delta^{\text{rb,du}} = 1.25 \times 10^{-4}$ and $\max_{\mu \in \Xi_{N_{\text{test}}}} |\tilde{\eta}_N(\mu) - \eta_N(\mu)| = 1.10 \times 10^{-6} < \max(\delta^{\eta_1}, \delta^{\eta_2}) = 1.25 \times 10^{-5}$. The notice that despite taking more iterations to converge, random sampling constructs a ROM that is just as accurate as greedy sampling. We also notice that greedy sampling gives a ROM with fewer basis functions.

In figure 6.8a we compare the convergence of the primal RB problem using $\mathcal{V}^{\text{pr}}_{N_{\text{pr}}}$ constructed using greedy parameter selection to $\mathcal{V}^{\text{pr}}_{N_{\text{pr}}}$ constructed using random parameter selection. The greedy algorithm converges after four iterations, so we construct the RB space $\mathcal{V}^{\text{pr}}_{N_{\text{pr}}}$ using four parameters in both cases. We observe that random parameter selection initially performs better than greedy parameter selection, but that they have similar performance as N_{pr} increases.

Random parameter sampling requires three more iterations to converge than greedy sampling. Therefore random parameter sampling entails: (i) three additional FE solves, (ii) an EQP training set ($\Xi_{N_{\text{greedy}}}$) with three additional parameters, and (iii) three additional iterations of bootstrapping. Here we will quantify the additional costs associated with each of these three items. The FE solves entail a total cost of $\hat{t}_{\text{fe,amr}} = 5.12$ (first primal FE solve), $\hat{t}_{\text{fe,pr}} = 6.00$ (subsequent primal FE solves), $\hat{t}_{\text{fe,dwr}} = 11.3$ (to verify DWR error estimate for subsequent FE solves), and $\hat{t}_{\text{fe,du}} = 3.43$ (to collect dual FE snapshots). The total cost associated with constructing EQP rules for $\Xi_{N_{\text{greedy}}}$ is $\hat{t}_{\text{eqp,total}} = 6.98$, while the total cost of bootstrapping is $\hat{t}_{\text{eqp,boot,total}} = 42.2$, where $\hat{t}_{\text{eqp,boot,pr}} = 27.0$ and

$\hat{t}_{\text{eqp,boot,du}} = 15.2$. The total cost of the greedy training algorithm is $\hat{t}_{\text{random,total,iter}} = 79.7$, while the cost of EQP for $\Xi_{N_{\text{greedy,min}}}$ following convergence of the greedy algorithm entails a cost of $\hat{t}_{\text{random,total,min}} = 5.33$, bringing the total cost to $\hat{t}_{\text{random,total}} = 85.0$. The cost of random parameter selection ($\hat{t}_{\text{random,total}} = 85.0$) is higher than the cost of greedy parameter selection ($\hat{t}_{\text{greedy,total}} = 57.6$), which demonstrates the efficiency of the greedy training algorithm. The problem considered here is relatively low-dimensional and we expect the cost-savings provided by greedy sampling to be even larger for high-dimensional problems.

6.2 Unsteady Lid-Driven Cavity Flow

6.2.1 Problem Description

We will now consider a two-dimensional lid-driven cavity flow. This problem is of real-world interest and it is vectorized, which increases the computationally complexity. The lid-driven cavity flow problem is governed by the incompressible Navier-Stokes equations. We write the strong form of the unsteady incompressible Navier-Stokes equations specialized to our lid-driven cavity flow problem

$$\frac{\partial v(\mu)}{\partial t} - \nu \Delta v(\mu) + (v(\mu) \cdot \nabla) v(\mu) + \nabla p(\mu) = 0 \quad \text{on} \quad \Omega \times I, \tag{6.1}$$

$$\nabla \cdot v(\mu) = 0 \quad \text{on} \quad \Omega \times I, \tag{6.2}$$

$$v(\mu) = \sin(\omega t) v_b \quad \text{on} \quad \Gamma_{\text{lid}} \times I, \tag{6.3}$$

$$v(\mu) = 0 \quad \text{on} \quad \partial\Omega \setminus \Gamma_{\text{lid}} \times I, \tag{6.4}$$

$$v(\mu) = 0 \quad \text{on} \quad \partial\Omega \times \{t = 0\}, \tag{6.5}$$

where $\Omega \subseteq \mathbb{R}^2$ is the unit square spatial domain, $\Gamma_{\text{lid}} \equiv (0,1) \times \{x_2 = 1\}$ is the upper boundary corresponding to the lid, $I = (0,T)$ is the time interval, $\nu \in \mathbb{R}_{>0}$ is the kinematic viscosity, and ω is the frequency of the lid's velocity. If we take the limit as $\nu \to \infty$, then we approach Stokes flow. We introduce the Reynolds number $\text{Re} = UL/\nu$ and set the characteristic velocity and characteristic length to unity so that $\text{Re} = 1/\nu$. We define our parameter $\mu \equiv [\text{Re}, \omega]$ and the parameter domain $\mathcal{D} \equiv [0, 100] \times [\pi/2, 2\pi]$. We further note that v is the vector of velocity fields with entries v_1 and v_2, while p is the pressure field. We enforce time-varying Dirichlet boundary conditions $\sin(\omega t) v_b$, which correspond to the velocity of the lid, where $v_{b,1} = e^{-\frac{0.05}{0.25-(x_1-0.5)^2}}$ and $v_{b,2} = 0$. We also

define the output functional

$$q(v(\mu);\mu,t) = \frac{1}{2}\int_\Omega [v(\mu)\cdot v(\mu)]dx, \tag{6.6}$$

so that our output is the mean kinetic energy: $s(\mu) \equiv \int_I q(v(\mu);\mu,t)dt$.

We introduce the function spaces $\mathcal{V} = [H_0^1(\Omega)]^2$ and $Q = L^2(\Omega)$. Then we introduce the test function space $X \equiv \mathcal{V}\times Q\times\mathbb{R}$ and the test function $\varphi \equiv (\psi,\chi,\Lambda)^T \in X$. We enforce inhomogeneous Dirichlet boundary conditions on the velocity field on Γ_{lid} and homogeneous Dirichlet boundary conditions on $\partial\Omega\setminus\Gamma_{\text{lid}}$. We introduce the lifting function $v_{\text{ss}} \in H^1(\Omega)^2$ such that $v_{\text{ss}}|_{\Gamma_{\text{lid}}} = v_b$ and $v_{\text{ss}}|_{\partial\Omega\setminus\Gamma_{\text{lid}}} = 0$. Then we introduce the time-dependent essential spaces $\mathcal{V}^{E,t} \equiv \sin(\omega t)v_{\text{ss}} + \mathcal{V}$ and $X^{E,t} \equiv \mathcal{V}^{E,t}\times Q\times\mathbb{R}$, which we use to introduce the lifting function $u_{\text{ss}} = (v_{\text{ss}},0,0) \in X^{E,t}$ such that we can also write $X^{E,t} \equiv \sin(\omega t)u_{\text{ss}} + X$. We further note that $u \equiv (v,p,\lambda)$ and that given $u \in X^{E,t}$, the velocity field satisfies the inhomogeneous Dirichlet boundary conditions (i.e., $v \in \mathcal{V}^{E,t}$). In practice we identify our lifting function by solving the steady version of the lid-driven cavity flow problem with $\nu = 1$, $\partial_t(v(\mu)) = 0$, and Dirichlet boundary condition $v = v_b$ on Γ_{lid}.

We cannot uniquely determine the value of pressure without imposing a constraint, so we enforce the zero-mean pressure condition using the Lagrange multiplier λ, which we add to the divergence-free condition (6.2). We state the weak form of our problem: given $\mu \in \mathcal{D}$, find $v \in \mathcal{V}^{E,t}$, $p \in Q$, and $\lambda \in \mathbb{R}$ such that

$$\int_\Omega [\psi\cdot\partial_t v + \psi\cdot(v\cdot\nabla)v - (\nabla\cdot\psi)p + \nu(\nabla\psi\cdot\nabla v)]dx = 0 \quad \forall\psi\in\mathcal{V}, \tag{6.7}$$

$$\int_\Omega \chi(\nabla\cdot v - \lambda)dx = 0 \quad \forall\chi\in Q, \tag{6.8}$$

$$\int_\Omega p dx = 0, \tag{6.9}$$

where (6.7) is the weak form of (6.1), (6.8) corresponds to the divergence-free condition (6.2), and (6.9) corresponds to the zero-mean pressure condition.

We use Taylor-Hood elements to solve the FE problem. We use a p-order polynomial space $\mathcal{V}_h$ for the space associated with the velocity field, while we use a $p-1$-order polynomial space Q_h for the space associated with the pressure field. Our essential space varies in time, so we look for $v_h \in \mathcal{V}_h^{E,t}$ at time t. For this work we choose $p = 3$.

We solve the primal FE problem to collect solution snapshots of the velocity fields $v_h \in \mathcal{V}_h^{E,t}$, the pressure field $p_h \in Q_h$, and the Lagrange multiplier $\lambda_h \in \mathbb{R}$. We have $\mathcal{N}_{h_1}$ degrees of freedom associated with the pressure space Q_h, while we have $2\times\mathcal{N}_{h_2}$

degrees of freedom associated with the velocity space $\mathcal{V}_h$ ($\mathcal{N}_{h_2}$ degrees of freedom for each dimension). We introduce the generalized coordinates $\mathbf{w}_1, \mathbf{w}_2 \in \mathbb{R}^{\mathcal{N}_{h_2}}$ which we associate to the function $w \in \mathcal{V}_h^E$ as

$$w_j(x) \equiv \sum_{i=1}^{\mathcal{N}_{h_2}} \mathbf{w}_{j,i}\psi_i(x) \quad j = 1, 2.$$

We introduce the generalized coordinate $\mathbf{z} \in \mathbb{R}^{\mathcal{N}_{h,1}}$ which we associate to the function $z \in Q_h$ where

$$z(x) \equiv \sum_{i=1}^{\mathcal{N}_{h_1}} \mathbf{w}_i\chi_i(x).$$

The generalized state coordinates $(\mathbf{v}_{h,1}, \mathbf{v}_{h,2})^T \in \mathbb{R}^{2\mathcal{N}_{h_2}}$ correspond to $v_h \in \mathcal{V}_h^{E,t}$, while the generalized state coordinate $\mathbf{p}_h \in \mathbb{R}^{\mathcal{N}_{h_1}}$ corresponds to $p_h \in Q_h$. We combine these generalized state coordinates into the vector $\mathbf{u}_h \equiv (\mathbf{v}_{h,1}, \mathbf{v}_{h,2}, \mathbf{p}_h, \lambda_h)^T \in \mathbb{R}^{\mathcal{N}_h}$, where $\mathcal{N}_h \equiv 2 \times \mathcal{N}_{h,2} + \mathcal{N}_{h_1} + 1$, which corresponds to $u_h \equiv (v_h, p_h, \lambda_h)^T \in X_h^{E,t}$. We further note that the generalized state $\mathbf{u}_{h,\text{ss}} \in \mathbb{R}^{\mathcal{N}_h}$ corresponds to the FE approximation of the lifting function u_{ss}.

We introduce the snapshots $\{\mathbf{u}_{h,k}(\mu)\}_{k=1}^K$ for $\mu \in \Xi_{N_{\text{train}}}$, where $\mathbf{u}_{h,k}(\mu) \in \mathbb{R}^{\mathcal{N}_h}$. We modify these snapshots to get $\{\mathbf{S}_k^0(\mu) \equiv \mathbf{u}_{h,k}(\mu) - \sin(\omega t_k)\mathbf{u}_{h,\text{ss}}\}_{k=1}^K$, where the snapshots $\mathbf{S}_k^0 \in \mathbb{R}^{\mathcal{N}_h}$ satisfy homogeneous (as opposed to inhomogeneous) Dirichlet BCs. We use POD to compress the snapshots $\{\mathbf{S}_k^0(\mu)\}_{k=1}^K$ into an X-orthonormal basis $\{\phi_i^{\text{pr}}\}_{i=1}^N$ of X_N^{pr}. Then we introduce generalized coordinate $\mathbf{w} \in \mathbb{R}^N$ that we associate to the function

$$w = \sum_{j=1}^N \phi_j^{\text{pr}}\mathbf{w}_j \in X_N^{\text{pr}}$$

We introduce the essential space $X_N^{E,t,\text{pr}} \equiv \sin(\omega t)u_{\text{ss}} + X_N^{\text{pr}}$, and we write

$$w = \sum_{j=1}^N \phi_j^{\text{pr}}\mathbf{w}_j + \sin(\omega t)\mathbf{u}_{h,\text{ss}} \in X_N^{E,t,\text{pr}},$$

which satisfies the inhomogeneous Dirichlet boundary conditions. We note that this lifting technique was first presented in [10].

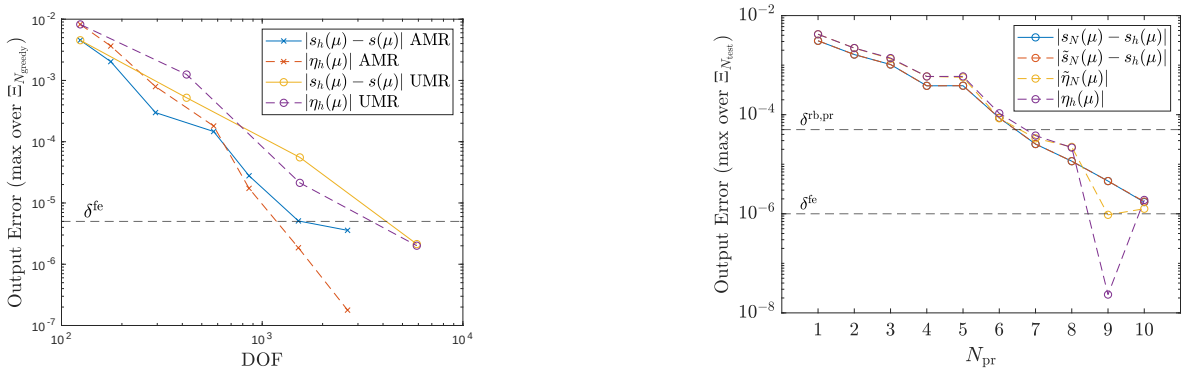
We conclude this section with a description of the discretization parameters and error tolerances. We use AMR to control the FE error. We time-march using the Crank-Nicolson method over the interval $I \equiv (0, 1]$ with the time-step $\Delta t = 0.005$ so that we have $K = 201$ time-steps. We choose $\delta^{\text{fe}} = 5\times10^{-6}$, $\delta^{\text{rb,pr}} = 5\times10^{-5}$, and $\delta^{\text{rb,du}} = 0.25\times\delta^{\text{rb,pr}} =$

1.25×10^{-5}. We also choose the EQP error tolerances $\delta^r = 0.1 \times \delta^{\text{rb,pr}} = 5 \times 10^{-6}$, $\delta^q = 10^{-8}$, $\delta^{\eta_1} = 0.1 \times \delta^{\text{rb,du}} = 1.25 \times 10^{-6}$, and $\delta^{\eta_2} = 0.1 \times \delta^{\eta_1} = 1.25 \times 10^{-7}$. We also choose $N_{\text{random}} = 5$ for when we construct $\Xi_{N_{\text{greedy,min}}}$. We have two parameters to choose: (i) the Reynolds number $\text{Re} \in [10, 100]$, and (ii) the frequency $\omega \in [\pi/4, 2\pi]$. We choose a uniform 10×8 grid of train points $\Xi_{N_{\text{train}}}$ with 10 Reynolds numbers and 8 frequencies, while we randomly select $N_{\text{test}} = 20$ test points $\Xi_{N_{\text{test}}}$.

6.2.2 Results

We invoke the greedy algorithm (algorithm 1 in section 5.1) which converges after three iterations ($N_{\text{greedy}} = 3$) with $N_{\text{pr}} = 10$ and $N_{\text{du}} = 3$. Figure 6.9a shows the FE DWR error estimate as a function of the number of FE degrees of freedom for $\mu \in \Xi_{N_{\text{greedy}}}$. We observe that the error associated with AMR converges more rapidly than uniform refinement. We identify $\mu^* \equiv \arg\max_{\mu \in \Xi_{N_{\text{greedy}}}} |\eta_h(\mu)|$ and plot $|\eta_h(\mu^*)|$. We also see that our DWR error estimate is reasonably effective. Figure 6.9b shows the RB and RB-EQP output error as a function of N_{pr}, overlayed with the FE and RB-EQP DWR error estimates. We see that the DWR error estimates are effective, but that the effectivity is worse for $N_{\text{pr}} = 9$. For $N_{\text{pr}} = 9, 10$ the output error is low and is close to the FE error tolerance, so the resolution of the dual FE solver may not be high enough to properly estimate the output error for $N_{\text{pr}} > 8$. The DWR method results from a linearization, so we expect its effectivity to improve as the error decreases as long as $\mathcal{V}_h$ is sufficiently refined. When we construct our RB space, we are able to measure the exact error for the parameters in $\Xi_{N_{\text{greedy}}}$, so we know that our RB-EQP model is accurate for these parameters. The AMR algorithm is important because it ensures that our FE space is rich enough that (i) the output FE output error is much lower than our RB output error tolerance, and (ii) the FE DWR error estimate is effective.

We will now discuss the error introduced by the RB and RB-EQP approximations. The greedy algorithm converges after three iterations so that $N_{\text{greedy}} = 3$. For the primal problem we get $N_{\text{pr}} = 10$ with $M^r = 148$ and $M^q = 61$, while for the dual problem we get $N_{\text{du}} = 3$ with $M^{\eta_1} = 28$ and $M^{\eta_2} = 87$. We measure the output error in the test data $\max_{\mu \in \Xi_{N_{\text{test}}}} |\tilde{s}_N(\mu) - s_h(\mu)| = 2.55 \times 10^{-6} < \delta^{\text{rb,pr}} = 5 \times 10^{-5}$ and $\max_{\mu \in \Xi_{N_{\text{test}}}} |\tilde{s}_N(\mu) - s_N(\mu)| = 1.96 \times 10^{-6} < \delta^r = 5 \times 10^{-6}$. We see that we have precise control of the output error. We measure the error in the DWR error estimate $\max_{\mu \in \Xi_{N_{\text{test}}}} |\tilde{\eta}_N(\mu) - \eta_h(\mu)| = 4.05 \times 10^{-6} < \delta^{\text{rb,du}} = 1.25 \times 10^{-5}$ and $\max_{\mu \in \Xi_{N_{\text{test}}}} |\tilde{\eta}_N(\mu) - \eta_N(\mu)| = 4.52 \times 10^{-8} < \max(\delta^{\eta_1}, \delta^{\eta_2}) = 1.25 \times 10^{-6}$. Here we notice that the error control for the DWR error estimate is also tight. We note that our DWR error estimate is also



(a) AMR Convergence

(b) RB Convergence

Figure 6.9: Convergence for lid-driven cavity flow

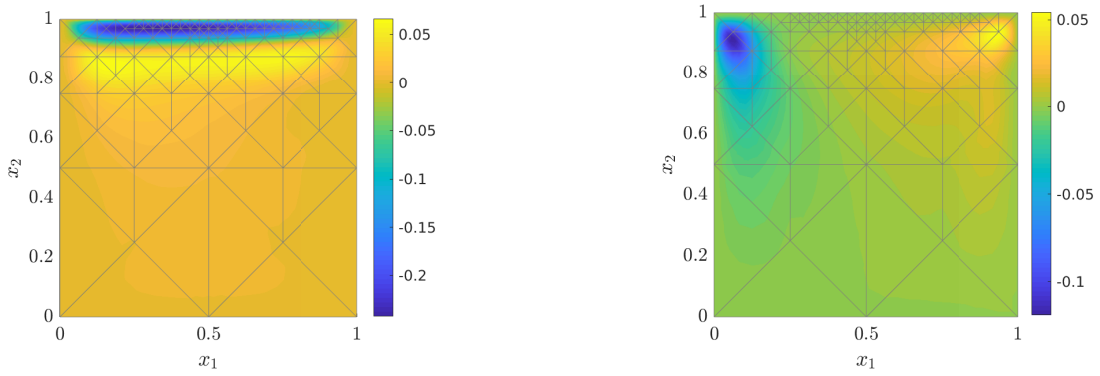
(a) Primal EQP

(b) Dual EQP

Figure 6.10: EQP points for lid-driven cavity flow

effective because the DWR error estimate closes matches the output error, as shown in figure 6.9b.

We plot the EQP points for the lid-driven cavity flow problem in figure 6.10 to make EQP more concrete. We see that the quadrature points concentrate to the upper boundary, especially in the corners, where the boundary conditions are inhomogeneous and vary in time. We note that the quadrature points in the upper corners are likely chosen because this is where y-velocity has the largest gradients. We further note that the rules associated with $\text{LP}^{\eta,1}$ and $\text{LP}^{\eta,2}$ both select some of the same quadrature points and that we separate $\text{LP}^{\eta,1}$ and $\text{LP}^{\eta,2}$ for precisely this reason. We also plot the solution, overlayed with the adaptively refined mesh in figure 6.11 and we see that the mesh is more refined toward the upper boundary, which corresponds to the thin boundary layer in the v_1 field.



(a) x_1 velocity

(b) x_2 velocity

Figure 6.11: Velocity field for lid-driven cavity flow for $\mathrm{Re} = 100$ and $\omega = 2\pi$ at $t = 1$

	LP^r	LP^q	LP^{η_1}	LP^{η_2}	total
Construct	1.28	0.120	0.513	0.956	2.87
Solve	0.034	0.0080	0.0011	0.0018	0.0451

Table 6.5: Normalized offline time for $\mathrm{LP}_{\mathrm{EQP}}$ for $\Xi_{N_{\mathrm{greedy}}}$

6.2.3 Offline Computational Cost

We will now discuss the cost of the greedy offline training. The primary training costs are associated with (i) solving the primal and dual FE problems, (ii) following the LP procedures to construct the EQP rules, and (iii) evaluating the ROMs for $\mu \in \Xi_{N_{\mathrm{train}}}$ in the bootstrapping stage. We also note that the initial AMR procedure to determine the spatial and temporal discretizations is more costly than subsequent FE solves, so we will separate this cost from the cost of the other FE solves. We normalize all times using the time for one primal FE solve for $\mu = (100, 2\pi)$.

The initial AMR procedure entails a normalized cost of $\hat{t}_{\mathrm{fe,amr}} = 10.2$, while all subsequent FE solves have a combined cost of (i) $\hat{t}_{\mathrm{fe,pr}} = 2.00$ to solve the primal problem, (ii) $\hat{t}_{\mathrm{fe,dwr}} = 5.04$ to verify that the desired error tolerance is satisfied (i.e., solve dual problem on a uniformly refined mesh and evaluate the DWR error estimate), and (iii) $\hat{t}_{\mathrm{fe,du}} = 1.50$ to collect snapshots for the dual RB space by solving the dual FE problem (without uniform mesh refinement). Next we perform POD and choose both N_{pr} ($\hat{t}_{N_{\mathrm{pr}}} = 0.182$) and N_{du} ($\hat{t}_{N_{\mathrm{du}}} = 0.155$). After choosing N_{pr} and N_{du}, we must solve the dual problem projected onto the primal RB space ($\hat{t}_{\mathrm{rb},\beta,\mathrm{pr}} = 1.11$) and the tangent problem projected onto the dual RB space ($\hat{t}_{\mathrm{rb},\chi,\mathrm{du}} = 0.824$). A break-down of the costs to construct the

	primal	output	DWR (dual)	DWR (eval)	total
FE	1	0.0122	0.504	0.0344	1.55
RB	0.523	0.0196	0.0997	0.0257	0.668
RB-EQP	0.0340	5.79×10^{-4}	0.0091	0.0013	0.0449

Table 6.6: Normalized online computational time for lid-driven cavity flow

constraints and solve the LP for all LP procedures for $\Xi_{N_{\text{greedy}}}$ is provided table 6.5, and their total cost is $\hat{t}_{\text{eqp,total}} = 2.92$. After we construct the EQP rules, we solve the primal RB-EQP problem ($\hat{t}_{\text{eqp,boot,pr}} = 5.78$) and evaluate the RB-EQP DWR error estimate ($\hat{t}_{\text{eqp,boot,du}} = 2.41$) for $\mu \in \Xi_{N_{\text{train}}}$, which entails a total cost of $\hat{t}_{\text{eqp,boot,total}} = 8.19$, where $N_{\text{train}} = 80$. The total cost of the greedy training algorithm for three iterations is $\hat{t}_{\text{greedy,total,iter}} = 32.1$, which is much lower than performing $N_{\text{train}} = 80$ primal solves. This demonstrates the efficiency and reliability of the greedy algorithm.

We also choose $N_{\text{random}} = 5$ so that $N_{\text{greedy,min}} = 8$. The cost to solve the primal RB problem is $\hat{t}_{\text{rb},\alpha,\text{pr,min}} = 4.10$, while the cost to solve the dual problem in the primal RB space is $\hat{t}_{\text{rb},\beta,\text{pr,min}} = 1.14$. After we construct the LP^r EQP rule for $\Xi_{N_{\text{greedy min}}}$, we solve the dual RB problem ($\hat{t}_{\text{rb},\beta,\text{du,min}} = 0.772$) and we solve the tangent problem projected onto the dual RB space ($\hat{t}_{\text{rb},\chi,\text{du,min}} = 1.21$). The total cost of EQP for $\Xi_{N_{\text{greedy,min}}}$ is $\hat{t}_{\text{eqp,min}} = 4.49$. The total cost of the training procedure associated with $\Xi_{N_{\text{greedy,min}}}$ is $\hat{t}_{\text{greedy,total,min}} = 10.9$. This brings the total cost of the greedy algorithm to $\hat{t}_{\text{greedy,total}} = \hat{t}_{\text{greedy,total,iter}} + \hat{t}_{\text{greedy,total,min}} = 43.0$, which is still much lower than performing $N_{\text{train}} = 80$ FE solves.

6.2.4 Online Computational Cost

We will now discuss the online computational cost of our ROMs. We normalize all computational times using the time to perform one primal FE solve for $\mu = (100, 2\pi)$. We present the normalized times in table 6.6. We notice that the speed-up introduced by the RB model is larger when solving the primal and dual problems as compared to evaluating the output or the DWR error estimate. The cost of evaluating functionals using the RB model still depends on $\mathcal{N}_h$, but the RB model decreases the cost of the nonlinear solvers because the systems are smaller (N instead of $\mathcal{N}_h$). The RB-EQP solver leads to a speed-up of 29.3 to solve the primal problem and evaluate the output, while it leads to a speed-up of 51.9 to solve the dual problem and evaluate the DWR error estimate. The RB-EQP solver leads to a total speed-up of 34.5 with respect to the FE solver.

Chapter 7

Conclusion

7.1 Summary

This thesis presents work on goal-oriented model reduction for time-dependent nonlinear parametrized PDEs. In chapter 2 we present an overview of existing model reduction methods, while in section 2.3.3 we outline why hyperreduction methods are necessary for nonlinear problems, then we briefly discuss four existing hyperreduction methods (EIM, GNAT, ECSW, and EQP) following our discussion of hyperreduction in section 2.3.5. In chapter 3 we present the existing EQP framework in much more detail. In particular we present the goal-oriented EQP and DWR EQP frameworks for steady problems in sections 3.3 and 3.4, respectively. In section 3.5 we present an alternate approach to DWR EQP using the dual-of-the-dual approach, which solves the tangent equation (3.22), that we use to extend DWR EQP to time-dependent problems in section 4.5.1. In chapter 4 we present a goal-oriented EQP framework and a DWR EQP framework for time-dependent problems, then in chapter 5 we present an efficient and automated offline training procedure (algorithm 1 in section 5.1). In chapter 6, we present numerical results to demonstrate that our model reduction method is (i) rapid, (ii) reliable, (iii) efficient, and (iv) automated for a nonlinear reaction-diffusion equation and a lid-driven cavity flow problem, both of which are unsteady. We examine the computational cost of both the offline and online stages and find that our offline stage is efficient, while our online stage is rapid. We also find that our output evaluation is rapid and reliable in the online stage, and that we are able to evaluate our DWR error estimates rapidly for any parameter, meaning that we have reliable simulations. In the next section we will discuss directions for future work.

7.2 Future Work

In the future, we wish to extend our work on goal-oriented model reduction for time-dependent problem in several directions:

- *Application to larger-scale problems with more complex nonlinearities.* We wish to apply the goal-oriented EQP framework to time-dependent aerodynamic flows and to multi-component problems with a nonlinearity that is higher than second-order (i.e., to a lid-driven cavity flow with a turbulence model at higher Reynolds numbers, as the lid-driven cavity flow presented in section 6.2 has a second-order nonlinearity).

- *Application to many-query time-dependent problems.* We are also interested in demonstrating the framework in the context of uncertainty quantification, optimization, and digital flight-envelope characterization so that we can further demonstrate its value. The FE simulations needed to enable these many-query applications are computationally expensive, but this framework will allow users to perform simulations for a larger number of parameters more rapidly than would otherwise be possible, which will further enhance the performance of applications in the many-query context.

- *Stability analysis.* We wish to analyze the nonlinear stability of the RB-EQP method, particularly for convection-dominated problems. While we did not encounter major convergence problems for the problems tested, both the theoretical and practical understanding of the stability of the method is important to address more challenging problems discussed above. It is shown in [47] that a ROM based on the RB-EQP hyperreduction procedure inherits the energy stability of a discontinuous Galerkin discretization scheme for linear convection-diffusion systems. We wish to extend the formulation to provide guaranteed nonlinear (or entropy) stability.

- *Improving the autonomy of the greedy algorithm.* While the proposed simultaneous RB-EQP offline training algorithm eliminates much user intervention in constructing a reliable ROM, some aspects of the algorithm could be further improved. For instance the current algorithm requires the selection of N_{random} parameters for EQP training to improve the parametric robustness of the error estimate of the EQP rules. We would like to avoid using tuning parameters, such as N_{random}, so that users need only specify the desired error tolerances.

Bibliography

[1] M. Ali, K. Steih, and K. Urban. Reduced basis methods based upon adaptive snapshot computations. *Advances in Computational Mathematics*, 43:257–294, 2017.

[2] D. Amsallem, S. Deolalikar, F. Gurrola, and C. Farhat. Model predictive control under coupled fluid-structure constraints using a database of reduced-order models on a tablet. In *21st AIAA Computational Fluid Dynamics Conference*. American Institute of Aeronautics and Astronautics, June 2013. paper 2013-2588.

[3] S. S. An, T. Kim, and D. L. James. Optimizing cubature for efficient integration of subspace deformations. *ACM Transactions on Graphics*, 27(5):165:1–165:10, Dec. 2008.

[4] P. Astrid, S. Weiland, K. Willcox, and T. Backx. Missing point estimation in models described by proper orthogonal decomposition. *IEEE Transactions on Automatic Control*, 53(10):2237–2251, 2008.

[5] M. Barrault, Y. Maday, N. C. Nguyen, and A. T. Patera. An "empirical interpolation" method: application to efficient reduced-basis discretization of partial differential equations. *Comptes rendus de l'Académie des Sciences, Serie I*, 339:667–672, 2004.

[6] R. Becker and R. Rannacher. An optimal control approach to a posteriori error estimation in finite element methods. In A. Iserles, editor, *Acta Numerica*. Cambridge University Press, 2001.

[7] P. Benner, S. Gugercin, and K. Willcox. A survey of projection-based model reduction methods for parametric dynamical systems. *SIAM Review*, 57(4):483–531, 2015.

[8] M. Besier and R. Rannacher. Goal-oriented space-time adaptivity in the finite element Galerkin method for the computation of nonstationary incompressible flow. *International Journal for Numerical Methods in Fluids*, 70(9):1139–1166, jan 2012.

[9] T. Bui-Thanh, K. Willcox, and O. Ghattas. Parametric reduced-order models for probabilistic analysis of unsteady aerodynamic applications. *AIAA Journal*, 46(10):2520–2529, 2008.

[10] J. Burkardt, M. Gunzburger, and H. Lee. Centroidal voronoi tessellation-based reduced-order modeling of complex systems. *SIAM Journal on Scientific Computing*, 28(2):459–484, 2006.

[11] K. Carlberg. Adaptive h-refinement for reduced-order models. *International Journal for Numerical Methods in Engineering*, 102(5):1192–1210, Nov. 2014.

[12] K. Carlberg, M. Barone, and H. Antil. Galerkin v. least-squares Petrov-Galerkin projection in nonlinear model reduction. *Journal of Computational Physics*, 330:693–734, Feb. 2017.

[13] K. Carlberg, C. Bou-Mosleh, and C. 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.

[14] K. Carlberg, C. Farhat, J. Cortial, and D. Amsallem. The GNAT method for nonlinear model reduction: effective implementation and application to computational fluid dynamics and turbulent flows. *Journal of Computational Physics*, 242:623–647, 2013.

[15] S. Chaturantabut and D. C. Sorensen. Nonlinear model reduction via Discrete Empirical Interpolation. *SIAM Journal on Scientific Computing*, 32(5):2737–2764, 2010.

[16] V. K. Edoardo Amaldi. On the approximability of minimizing nonzero variables or unsatisfied relations in linear systems. *Theoretical Computer Science*, 209(7):237–260, Dec. 1998.

[17] R. Everson and L. Sirovich. Karhunen-Loève procedure for gappy data. *Journal of the Optical Society of America A, Optics and Image Science*, 12(8):1657–1664, 1995.

[18] C. Farhat, P. Avery, T. Chapman, and J. 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.

[19] C. Farhat, T. Chapman, and P. Avery. Structure-preserving, stability, and accuracy properties of the energy-conserving sampling and weighting method for the hyper reduction of nonlinear finite element dynamic models. *International Journal for Numerical Methods in Engineering*, 102(5):1077–1110, 2015. nme.4820.

[20] K. Fidkowski and D. Darmofal. Review of output-based error estimation and mesh adaptation in computational fluid dynamics. *AIAA Journal*, 49(4):673–694, 2011.

[21] K. J. Fidkowski. Output-based space-time mesh optimization for unsteady flows using continuous-in-time adjoints. *Journal of Computational Physics*, 341:258–277, 2017.

[22] M. A. Grepl. Certified reduced basis methods for nonaffine linear time-varying and nonlinear parabolic partial differential equations. *Mathematical Models & Methods in Applied Sciences*, 22(3):1150015–1–1150015–40, 2012.

[23] M. A. Grepl, Y. Maday, N. C. Nguyen, and A. T. Patera. Efficient reduced-basis treatment of nonaffine and nonlinear partial differential equations. *ESAIM: M2AN*, 41(3):575–605, 2007.

[24] B. Haasdonk. Convergence rates of the POD-Greedy method. *ESAIM: Mathematical Modelling and Numerical Analysis*, 47(3):859–873, Apr. 2013.

[25] B. Haasdonk and M. Ohlberger. Reduced basis method for finite volume approximations of parametrized linear evolution equations. *Mathematical Modelling and Numerical Analysis*, 42(2):277–302, 2008.

[26] J. A. Hernández, M. A. Caicedo, and A. Ferrer. Dimensional hyper-reduction of nonlinear finite element models via empirical cubature. *Computer Methods in Applied Mechanics and Engineering*, 313:687–722, 2017.

[27] J. S. Hesthaven, G. Rozza, and B. Stamm. *Certified reduced basis methods for parametrized partial differential equations*. Springer, 2016.

[28] S. Hovland, J. T. Gravdahl, and K. E. Willcox. Explicit model predictive control for large-scale systems via model reduction. *Journal of Guidance, Control, and Dynamics*, 31(4):918–926, July 2008.

[29] P. LeGresley and J. Alonso. Airfoil design optimization using reduced order models based on proper orthogonal decomposition. In *Fluids 2000 Conference and Exhibit*. American Institute of Aeronautics and Astronautics, June 2000. paper 2000-2545.

[30] P. A. LeGresley and J. J. Alonso. Investigation of non-linear projection for POD based reduced order models for aerodynamics. In *39th Aerospace Sciences Meeting and Exhibit*, number 2001-0926. American Institute of Aeronautics and Astronautics, 2001.

[31] K. C. Martin Drohmann. The romes method for statistical modeling of reduced-order-model error. *SIAM/ASA Journal on Uncertainty Quantification*, 3:115–145, 2015.

[32] D. Mavriplis, D. Darmofal, D. Keyes, and M. Turner. Petaflops opportunities for the NASA fundamental aeronautics program. In *18th AIAA Computational Fluid Dynamics Conference*. American Institute of Aeronautics and Astronautics, June 2007. paper 2007-4084.

[33] M. Meyer and H. G. Matthies. Efficient model reduction in non-linear dynamics using the Karhunen-Loève expansion and dual-weighted-residual methods. *Computational Mechanics*, 31(1-2):179–191, May 2003.

[34] B. K. Natarajan. Sparse approximate solutions to linear systems. *SIAM Journal on Computing*, 24(8):227–234, 1995.

[35] N. C. Nguyen, A. T. Patera, and J. Peraire. A 'best points' interpolation method for efficient approximation of parametrized functions. *International Journal for Numerical Methods in Engineering*, 73(4):521–543, 2008.

[36] A. T. Patera and G. Rozza. Reduced basis approximation and a posteriori error estimation for parametrized partial differential equations. MIT Pappalardo graduate monographs in mechanical engineering, MIT, 2006.

[37] A. T. Patera and M. Yano. An LP empirical quadrature procedure for parametrized functions. *Comptes Rendus Mathematique*, 355(11):1161–1167, Nov. 2017.

[38] A. Quarteroni and A. Valli. *Numerical approximation of partial differential equations*. Springer, New York, 1997.

[39] G. Rozza, D. B. P. Huynh, and A. T. Patera. Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations — Application to transport and continuum mechanics. *Archives of Computational Methods in Engineering*, 15(3):229–275, 2008.

[40] D. Ryckelynck. A priori hyperreduction method: an adaptive approach. *Journal of Computational Physics*, 2005.

[41] J. Slotnick, A. Khodadoust, J. Alonso, D. Darmofal, W. Gropp, E. Lurie, and D. Mavriplis. CFD vision 2030 study: a path to revolutionary computational aerosciences. Nasa/cr-2014-218178, NASA, 2014.

[42] K. Urban and A. T. Patera. An improved error bound for reduced basis approximation of linear parabolic problems. *Mathematics of Computation*, 83:1599–1615, 2014.

[43] K. Veroy, C. Prud'homme, D. Rovas, and A. Patera. A posteriori error bounds for reduced-basis approximation of parametrized noncoercive and nonlinear elliptic partial differential equations. In *16th AIAA Computational Fluid Dynamics Conference*. American Institute of Aeronautics and Astronautics, June 2003. paper 2003-3847.

[44] M. Yano. A space-time Petrov-Galerkin certified reduced basis method: application to the Boussinesq equations. *SIAM Journal on Scientific Computing*, 36(1):A232–A266, Jan. 2014.

[45] M. Yano. A minimum-residual mixed reduced basis method: exact residual certification and simultaneous finite-element reduced-basis refinement. *ESAIM: Mathematical Modelling and Numerical Analysis*, 50(1):163–185, Jan. 2016.

[46] M. Yano. A reduced basis method for coercive equations with an exact solution certificate and spatio-parameter adaptivity: energy-norm and output error bounds. *SIAM Journal on Scientific Computing*, 40(1):A388–A420, 2018.

[47] M. Yano. Discontinuous Galerkin reduced basis empirical quadrature procedure for model reduction of parametrized nonlinear conservation laws. *Advances in Computational Mathematics*, 45(5-6):2287–2320, June 2019.

[48] M. Yano. Goal-oriented model reduction of parametrized nonlinear PDEs; application to aerodynamics. *International Journal for Numerical Methods in Engineering*, accepted, 2020.

[49] M. Yano and A. T. Patera. An LP empirical quadrature procedure for reduced basis treatment of parametrized nonlinear PDEs. *Computer Methods in Applied Mechanics and Engineering*, 344:1104–1123, Feb. 2019.

[50] M. Yano, A. T. Patera, and K. Urban. A space-time hp-interpolation-based certified reduced basis method for Burgers' equation. *Mathematical Models and Methods in Applied Sciences*, 24(09):1903–1935, Aug. 2014.