

A Goal-Oriented Adaptive Sampling Procedure for
Projection-Based Reduced-Order Models with
Hyperreduction

Calista Biondic

Department of Mechanical Engineering

McGill University, Montreal

August 2024

A thesis submitted to McGill University in partial fulfillment of the
requirements of the degree of

Master of Science

© Calista Biondic, 2024

Abstract

Projection-based reduced-order models (PROMs) have demonstrated accuracy, reliability, and robustness in approximating high-dimensional, differential equation-based computational models across many applications. For this reason, it has been proposed as a tool for high-querying parametric design problems like those arising in modern aircraft design. Since aerodynamic simulations can be computationally expensive, PROMs offer the potential for more rapid estimations of high-fidelity solutions. However, the efficiency can still be tied to the dimension of the full-order model (FOM), particularly when projected quantities must be frequently recomputed due to non-linearities or parameter dependence. In the case of Petrov-Galerkin models, the projected residual and Jacobian are re-evaluated at every Newton iteration, thereby limiting the anticipated cost improvements. Hyperreduction is one of the tools available to approximate these quantities and address this issue. This work tests the energy-conserving sampling and weighting (ECSW) method as a potential approach for hyperreduction. It will be incorporated into the work by Blais in his thesis, *Goal-Oriented Adaptive Sampling for Projection-Based Reduced-Order Models*, which had developed an adaptive sampling procedure for building a reduced-order model (ROM) with a controlled functional error. The impacts of hyperreduction on computational cost and accuracy will be studied using the NACA0012 airfoil.

Résumé

Les modèles d'ordre réduit par projection (PROM) se sont révélés, dans de nombreuses applications, précis, fiables et robustes lorsqu'il s'agit d'approximer des modèles numériques de calcul pour des systèmes d'équations différentielles à haute dimension. C'est pourquoi ils ont été proposés comme outil pour les problèmes de conception paramétrique itératifs, tels que ceux qui se posent dans la conception moderne des avions. Étant donné que les simulations aérodynamiques peuvent avoir un coût de calcul élevé, les PROM devraient permettre des estimations plus rapides de solutions à haute-fidélité. Toutefois, l'efficacité des PROM peut être liée à la dimension du modèle d'ordre complet (FOM), en particulier lorsque les quantités projetées doivent être fréquemment recalculées en raison de leurs non-linéarités ou de leur dépendance sur les paramètres de conception. Dans le cas des modèles utilisant la projection de Petrov-Galerkin, le résidu projeté et le jacobien sont réévalués à chaque itération de Newton, négligeant les avantages de coût de calcul attendus des modèles d'ordre réduit. L'hyper-réduction est l'un des outils disponibles pour approximer ces quantités. Dans ce travail, la méthode d'échantillonnage et de pondération conservant l'énergie (*energy-conserving sampling and weighting*, ECSW) sera évaluée en tant qu'approche potentielle pour l'hyper-réduction. Elle sera incorporée dans le travail de *Goal-Oriented Adaptive Sampling for Projection-Based Reduced-Order Models* dans lequel une procédure d'échantillonnage adaptative pour la construction d'un modèle d'ordre réduit (ROM) avec une erreur fonctionnelle contrôlée a été développée. Afin d'étudier les impacts de l'hyper-réduction sur le coût de calcul et la précision, l'écoulement autour d'un profil d'ail NACA0012 sera considéré.

Acknowledgements

First and foremost, I would like to thank my advisor, Professor Siva Nadarajah. His exceptional understanding and belief in me helped me persevere through some of the most challenging periods of my life, which I would not have been able to navigate without his unwavering support, empathy, and encouragement. His expertise and insights have been crucial in shaping my research and ensuring its success. I have learned many invaluable lessons from him which I hope to carry with me in my future academic and professional pursuits.

I am deeply thankful for my parents, Roberta and Renato. Their long-term commitment to supporting my academic journey and their constant encouragement have been a driving force in helping me pursue and achieve my dreams. To Savana and Amiel, thank you for making me feel at home and being my rock throughout this journey. A special thank you goes to my best friends, Willow and Mikayla. They've been there through thick and thin, helping me believe in myself and pushing me to achieve what I've done. I couldn't have made it this far without their love and support. To the wonderful girlfriends I have met in Montreal, you made this city a dream to live in and I am inspired by each and every one of you. I sincerely thank my colleagues from the McGill Computational Aerodynamics Group for their invaluable advice and for making my experience enjoyable. Thank you all for your contributions and support. This thesis would not have been possible without each and every one of you.

Table of Contents

Abstract	i
Résumé	ii
Acknowledgements	iii
List of Figures	x
List of Tables	xi
1 Introduction	1
1.1 Reduced-Order Models and Computational Bottlenecks	2
1.2 Hyperreduction	5
1.3 Error Bounds and Goal-Oriented Model Reduction	6
1.4 Thesis Overview	7
2 Fundamental Methods	10
2.1 Discontinuous Galerkin Method	10
2.2 Full-Order Model	12
3 Model Reduction	15
3.1 Proper Orthogonal Decomposition	15
3.2 Projection-Based Reduced-Order Model	17
3.3 Least-squares Petrov-Galerkin Projection	18

3.4	Hyperreduction	19
3.4.1	Approximate-then-Project	20
3.4.2	Project-then-Approximate	22
3.5	Energy-conserving Sampling and Weighting Method	24
3.5.1	Reduced Mesh Set Selection	25
3.5.2	Residual-based ECSW Training Process	26
3.5.3	Jacobian-based ECSW Training Process	28
3.5.4	Non-Negative Least-Squares Problem	29
3.6	Hyperreduced ROM Solution Evaluation with ECSW	30
4	ECSW Hyperreduction Verification	33
4.1	One Parameter One Dimensional Burgers' Equation	34
4.2	Two Parameter NACA0012 Airfoil in Transonic Flow	40
5	Adaptive Sampling Procedure	49
5.1	Overview of the Adaptive Sampling Procedure	49
5.2	Dual-Weighted Residual Errors	51
5.2.1	Error Estimate Between Coarse and Fine Hyperreduced Reduced-Order Models	52
5.3	Updated Goal-Oriented Adaptive Sampling	54
5.4	Analysis of Computational Cost Savings through Work Units	56
6	Results	61
6.1	One Parameter NACA0012 Airfoil in Inviscid Subsonic Flow	62
6.2	Two Parameter NACA0012 Airfoil in Transonic Flow	68
6.3	One Parameter NACA0012 Airfoil in Inviscid Subsonic Flow on a Fine Mesh	76
6.4	Two Parameter NACA0012 Airfoil in Inviscid Subsonic Flow on a Fine Mesh	81

7 Conclusion	86
7.1 Future Work	88
A Work Units Derivations	89
A.1 ROM Solution Non-linear Iteration Work Units	89
A.2 HROM Solution Non-linear Iteration Work Units	91
A.3 DWR Error ϵ_r Work Units	93
A.4 Hyperreduced DWR Error ϵ_r Work Units	96
Bibliography	97

List of Figures

4.1	Solutions of 1D Burgers' Equation with Exponential Source Term from Upwind Scheme	35
4.2	Error in the Functional \mathcal{J} over the Parameter Domain for the Final ROM and Three Residual-based ECSW HROM	36
4.3	Error in the Functional \mathcal{J} over the Parameter Domain for the Final ROM and Three Jacobian-based ECSW HROM	36
4.4	Solutions of 1D Burgers' Equation from the FOM and Projection-Based ROM with and without Residual-based Hyperreduction	38
4.5	Solutions of 1D Burgers' Equation from the FOM and Projection-Based ROM with and without Jacobian-based Hyperreduction	38
4.6	Impact of NNLS Tolerance on the Size of the Reduced Mesh and the Accuracy of the HROM for the Residual and Jacobian-based Training Approaches	40
4.7	Example Solution of NACA0012 Airfoil Test Case	41
4.8	Initial and Final Configuration of the Parameter Space for the 2 Parameter NACA0012 Airfoil in Transonic Flow	42
4.9	Error Estimate Across the Parameter Space for the ROM after Adaptive Sampling	42
4.10	True Error for the ROM and Zones of Tolerance Violation	43

4.11 True Error for the Residual-based HROM with $\epsilon_{\text{NNLS}} = 1E - 4$ and Zones of Tolerance Violation	44
4.12 True Error for the Residual-based HROM with $\epsilon_{\text{NNLS}} = 1E - 5$ and Zones of Tolerance Violation	45
4.13 True Error for the Residual-based HROM with $\epsilon_{\text{NNLS}} = 1E - 6$ and Zones of Tolerance Violation	45
4.14 True Error for the Jacobian-based HROM with $\epsilon_{\text{NNLS}} = 1E - 4$ and Zones of Tolerance Violation	46
4.15 True Error for the Jacobian-based HROM with $\epsilon_{\text{NNLS}} = 1E - 5$ and Zones of Tolerance Violation	46
4.16 True Error for the Jacobian-based HROM with $\epsilon_{\text{NNLS}} = 1E - 6$ and Zones of Tolerance Violation	47
4.17 Impact of NNLS Tolerance on the Size of the Reduced Mesh and the Accuracy of the HROM for the Residual and Jacobian-based Training Approaches . . .	48
6.1 Snapshot and ROM Points with Estimated and True Error for the ROM . . .	63
6.2 Snapshot and ROM Points with Estimated and True Error for the HROM w/o Hyper-DWR	64
6.3 Snapshot and ROM Points with Estimated and True Error for the HROM w/ Hyper-DWR	65
6.4 Maximum and Mean Error Estimate at the ROM Points during each Sampling Iteration for the Three Models	66
6.5 ECSW Reduced Mesh Set for the HROM w/o Hyper-DWR	67
6.6 ECSW Reduced Mesh Set for the HROM w/ Hyper-DWR	68
6.7 Final Configuration of the Parameter Space for the ROM and the Interpolated Estimated Error Distribution from the ROM Points	69

6.8	Final Configuration of the Parameter Space for the HROM w/o Hyper-DWR and the Interpolated Estimated Error Distribution from the ROM Points . . .	69
6.9	Final Configuration of the Parameter Space for the HROM w/ Hyper-DWR and the Interpolated Estimated Error Distribution from the ROM Points . . .	70
6.10	Maximum and Mean Error Estimate at the ROM Points during each Sampling Iteration for the Three Models	70
6.11	True Error for the ROM and Zones of Tolerance Violation	72
6.12	True Error for the HROM w/o Hyper-DWR and Zones of Tolerance Violation	72
6.13	True Error for the HROM w/ Hyper-DWR and Zones of Tolerance Violation	73
6.14	ECSW Reduced Mesh Set for the HROM w/o Hyper-DWR	74
6.15	ECSW Reduced Mesh Set for the HROM w/ Hyper-DWR	74
6.16	Work Units for the Adaptive Sampling Procedure to Build the Three Models	75
6.17	Snapshot and ROM placement with Estimated and True Error across the parameter space at the end of the Adaptive Sampling Procedure for the ROM	77
6.18	Snapshot and ROM placement with Estimated and True Error across the pa- rameter space at the end of the Adaptive Sampling Procedure for the HROM w/ Hyper-DWR	78
6.19	Maximum and Mean Error Estimate at the ROM Points during each Sampling Iteration for the ROM and the HROM with the Hyperreduced Error	79
6.20	ECSW Reduced Mesh set for the HROM w/ Hyper-DWR	80
6.21	Final Configuration of the Parameter Space for the ROM and the Interpolated Estimated Error Distribution from the ROM Points	81
6.22	Final Configuration of the Parameter Space for the HROM w/ Hyper-DWR and the Interpolated Estimated Error Distribution from the ROM Points . . .	82

6.23	Maximum and Mean Error Estimate at the ROM Points during each Sampling	
	Iteration for the ROM and the HROM with the Hyperreduced Error	82
6.24	ECSW Reduced Mesh set for the HROM w/ Hyper-DWR	84
6.25	True Error for the ROM and Zones of Tolerance Violation	84
6.26	True Error for the HROM w/ Hyper-DWR and Zones of Tolerance Violation	85
6.27	Work Units for the Adaptive Sampling Procedure to Build the Three Models	85

List of Tables

4.1	1D Burgers' ROM Results at $b = 0.044$ and Average Absolute ROM Point Error over the Parameter Domain	39
6.1	One Parameter NACA0012 Airfoil ROM and HROM Important Dimensions, Average ROM Point Error and Average Online Error	67
6.2	Two Parameter NACA0012 Airfoil in Transonic Flow ROM and HROM Important Dimensions, Average ROM Point Error and Average Online Error	73
6.3	One Parameter NACA0012 Airfoil in Subsonic Flow ROM and HROM Important Dimensions, Average ROM Point Error and Average Online Error	80
6.4	Two Parameter NACA0012 Airfoil in Subsonic Flow ROM and HROM Important Dimensions, Average ROM Point Error and Average Online Error	83

Chapter 1

Introduction

Many fields of design and engineering are increasingly reliant on computational models. In aircraft design, for instance, computational fluid dynamics (CFD) has become an essential tool for accelerating analysis and design processes, and also helps inform decisions on experimental testing. Advancements in higher-order methods, faster solution algorithms, and parallel high-performance computing have significantly improved the time and cost efficiency of CFD. These innovations have been instrumental in advancing the design of more fuel-efficient and environmentally friendly aircraft. Despite these advancements, in certain applications there remain challenges due to the high computational costs and storage requirements of these models, rendering them computationally intractable. This limitation has driven substantial interest in methods for reducing the complexity and cost of large-scale computational models.

1.1 Reduced-Order Models and Computational Bottlenecks

The cost of producing CFD solutions poses a significant challenge in high-querying problems, such as aerodynamic shape optimization (ASO), where the model must be evaluated at numerous design parameter combinations, and in online real-time scenarios, like active control, where solutions must be computed within seconds. The goal of reduced-order modeling is to decrease the solution time and storage requirements needed to compute new CFD solutions while preserving the fidelity achieved by higher-order methods.

The approximation of a high-fidelity model by a low-dimensional representation is a major area of research in model reduction methods. These methods focus on problems where the high-fidelity model is a high-dimensional system of ordinary differential equations or a system of equations derived from the discretization of partial differential equations, which capture the underlying physics of the system [1]. The belief is then that the physical processes and dynamics of the system can be represented by a smaller number of degrees of freedom (DOFs). The reduced-order models (ROMs) used to approximate these complex full-order models (FOMs) can be grouped into two categories: non-intrusive and intrusive models. Non-intrusive models treat the problem as a black box and are independent of the full-order equations. There are a multitude of challenges that arise when using these models, particularly in large-scale dynamic systems. These include violations of physical constraints, high costs associated with generating large training data sets, and lack of confidence indicators [1]. Examples of non-intrusive models include proper orthogonal decomposition coupled with interpolation [2], Isomap [3, 4], Kriging [5, 6], and neural networks [1, 7–9].

Intrusive models operate under the assumption that the solution of a large-scale system of dimension N can be represented on a much lower dimensional subspace of dimension n in-

duced by parameter variation. The low-dimensional representation is found by projecting the governing equations onto the reduced-order subspace, which is why these models are often referred to as projection-based reduced-order models (PROMs). The two most commonly used approaches are the Galerkin projection [10–12] and the least-squares Petrov-Galerkin projection (LSPG) [13–15]. These models are often preferable to non-intrusive models as they are physics-informed, making them more robust, and they offer accessible error metrics [16], which help predict the expected accuracy of the model online. In particular, Petrov-Galerkin PROMs have been shown to be ideal for parametric, nonlinear high-dimensional models. While it was previously thought that using PROMs in fluid dynamics would lead to numerical instabilities due to their inability to resolve the dissipative regime of the turbulent energy cascade, [17] demonstrated that in fact, the Galerkin framework was the true source of these instabilities. Specifically, for problems where the Jacobian is not symmetric positive definite, which is generally the case in CFD problems, PROMs constructed using the Galerkin framework can perform poorly or exhibit instabilities [12, 17]. However, for the same convection-dominated laminar and turbulent flows, PROMs constructed using the Petrov-Galerkin (PG) framework were found to be numerically stable and accurate [17]. Overall, PG ROMs have proven to be ideal candidates for approximating CFD problems and will therefore be used in this work.

The remaining major challenge is that, in practice, the computational cost of these techniques still scales with the dimension of the FOM, N . In parametric and/or non-linear problems, the parametric vectors and matrices—such as the residual and Jacobian—must be re-evaluated at each non-linear iteration and then projected onto the reduced-order basis. This can be computationally expensive, and the cost would scale with the dimension of the FOM. If not addressed, the expected time-saving benefits of the PROM may not be fully realized.

There are generally two approaches for addressing this computational bottleneck and approximating these quantities: exact and inexact [18]. Both share the fundamental strategy that underlies most PROM computations, of breaking down the evaluation of these quantities into offline and online components. Exact methods are applicable to specific classes of problems, two examples being parametric, linear FOMs admitting an efficient parameter-affine representation, and non-linear FOMs with low-order polynomial dependence of the internal force vector on the solution and a time-independent external force vector. These methods are often referred to as exact precomputation-based methods, as they compute the reduced matrices and vectors in two parts: the first part addresses the computational bottleneck and can be precomputed offline, while the second part, whose computational complexity scales with integer powers of the reduced-order subspace n , can be processed online and in real time [18].

The problems addressed in this work, however, do not fall into the classes suitable for pre-computation. Therefore, inexact or approximate reconstruction methodologies must be used. Hyperreduction is an inexact method which has been developed for addressing more general cases which can be both linear or non-linear, parametric or non-parametric [18]. These approaches introduce an additional layer of approximation to evaluate the high-dimensional quantities in a manner that is independent of the dimension N of the FOM. As a result, some accuracy in the ROM is traded for computational efficiency. Many methods have been developed in this field, most of which are related in some way, differing primarily by the level of theoretical support they have [19].

1.2 Hyperreduction

Current hyperreduction techniques can be classified into two groups: *approximate-then-project* methods and *project-then-approximate* methods. The former was developed first and, as the name suggests, begins by generating an approximation of the high-dimensional quantity and then computes the exact projection of the approximation onto the left reduced-order basis (ROB) or test basis \mathbf{W} associated with the PROM. Techniques in this group include the empirical interpolation method (EIM) [20, 21], and its associated discrete form the discrete empirical interpolation method (DEIM) [22], both of which have typically been applied in the context of Galerkin projections. In contrast, the Gauss–Newton with approximated tensors (GNAT) method is designed for Petrov–Galerkin projections [13]. [19] notes that this group of methods tend to emphasize accuracy but not numerical stability. The second group, project-then-approximate methods, was proposed more recently as a response to concerns about the stability of hyperreduction techniques. These methods have demonstrated greater robustness in certain cases [18]. Instead of approximating the high-dimensional quantities directly, these approaches approximate their projections onto the left ROB. One of the earlier examples of this method is the energy-conserving sampling and weighting (ECSW) method, which was initially developed for Galerkin PROMs [23] and later adapted for use in Petrov-Galerkin PROMs [15].

Farhat et al. [18] note that, at present, DEIM represents the state of the art among approximate-then-project methods, while ECSW represents the state of the art among project-then-approximate methods. Noting that this work focuses on solving fluid flows problems, least-squares Petrov-Galerkin (LSPG) PROMS have been demonstrated to be both numerically stable and accurate in the context of convection-dominated laminar flows and turbulent flow problems, where Galerkin PROMs often exhibit instability [17]. For this reason, the ECSW hyperreduction method emerges as a suitable choice for the goals of this

work. Moreover, it has both a substantial amount of theoretical and experimental support. The ECSW method is provably unconditionally stable for second-order hyperbolic models and has been shown to be both numerically stable and accurate in structural dynamics problems, where approximate-then-project methods have been known to fail [19]. Additionally, when applied to first-order hyperbolic problems, it was shown to be reliable, accurate, and computationally efficient [15].

1.3 Error Bounds and Goal-Oriented Model Reduction

A reduced-order model can always be constructed and optimized with hyperreduction; however, this does not guarantee that its outputs will be accurate. Without a reliable method to quantify the error in a ROM, little confidence can be placed in its estimations. Several approaches have been developed to address this issue. For example, the work in [24, 25] introduced rigorous *a posteriori* error bounds based on the residual for linear problems and problems with quadratic non-linearities. However, these bounds are not online-efficient and have been shown to be impractical for complex nonlinear systems [14]. More efficient error estimation methods have also been developed [26, 27]. These approaches estimate the error between the FOM and the ROM using the magnitude of the norm of the residual. While these methods are computationally cheaper, they do not guarantee that a specific bound on the error exists.

Most of these approaches focus on state or solution error, but this is not always predictive of the error in the output of interest [28]. In certain applications, the error in the output or functional of interest could be more important. For instance, when designing an airfoil, the lift or drag it produces might be of greater concern than the exact flow solution around it. This is one of the motivations for the use of the dual-weighted-residual (DWR) method. In

CFD problems, the flow solution is referred to as the primal solution, while the dual problem refers to a problem that solves for the sensitivity of the functional relative to the primal solution [16]. The DWR method provides a systematic way to relate the residual to the output error through the use of a dual (or adjoint) solution [29]. Since this approach is computationally inexpensive, DWR error estimates have also been widely applied to automate mesh adaptation for both finite volume and finite element discretizations [30–33].

While the DWR method does not provide a strict error bound but rather an estimate, it is applicable to general nonlinearities in predictive settings [29] and has proven effective in practice for aerodynamic applications [31]. This method has been utilized in various contexts within model reduction. Early adoptions include work by Meyer and Matthies, who applied DWRs to estimate functional errors in reduced-order models (ROMs) of wind turbine blades [34]. More recently, DWR error indicators have become key components in greedy and adaptive sampling strategies. Notable examples include works such as [29, 35, 36]. In adaptive sampling procedures, an error indicator is used to estimate the error between the ROM and FOM, guiding the ROM’s construction to ensure that this error estimate satisfies a specified tolerance. The DWR method has been shown to be both accurate and efficient in these settings [16].

1.4 Thesis Overview

This work builds on the methodologies developed in [16], [37], and [36], which introduced an adaptive sampling procedure designed to reduce the output error of an LSPG projection-based reduced-order model (ROM) to within a prescribed tolerance. The sampling procedure took advantage of dual-weighted residual (DWR) error indicators to estimate the error between the ROM and FOM at specific design parameter locations, which were then used

to estimate the output error across the parameter space. However, as noted by Blais [16], hyperreduction was not incorporated into the ROM, meaning the computational cost of obtaining reduced-order solutions still scales with the dimension of the FOM, N . In particular, during each of the non-linear Newton iterations required to solve the ROM, the residual and Jacobian must be re-evaluated, making the process computationally expensive—even when the ROM has significantly reduced the problem’s dimensionality (i.e. $n \ll N$).

The goal of this work is to update the sampling procedure to build a hyperreduced reduced-order model (HROM) which still achieves a prescribed error tolerance across the parameter space, while accounting for the additional error introduced by hyperreduction. The hyperreduction is implemented using the ECSW approach described in the previous section and is employed to approximate both the residual and the Jacobian in the HROM. Additionally, the second DWR error indicator introduced in [16], which quantifies the error between a coarse and fine ROM (referring to the number of columns in the reduced order basis), is modified to capture the error introduced by the hyperreduction.

The thesis is structured as follows: Chapter 2 introduces the fundamental methods used in this work. This includes a brief overview of the discontinuous Galerkin method used in the flow solver, as well as the implicit method utilized to solve the full-order system. Chapter 3 presents the components of the reduced-order model, providing brief introductions to proper orthogonal decomposition and the LSPG framework. This chapter also explores existing hyperreduction techniques and provides a detailed explanation of the selected ECSW hyperreduction method. Chapter 4 verifies the implementation of the ECSW method using two test cases: the 1D Burgers’ equation and the NACA0012 airfoil. Chapter 5 introduces the previously developed adaptive sampling method and outlines the changes necessary to incorporate hyperreduction. Updates to the DWR error indicators are discussed here, along with a methodology for comparing the computational savings of hyperreduction using approxi-

mate work units. Finally, Chapter 6 presents the results of HROMs constructed for various configurations of a two-dimensional inviscid flow around the NACA0012 airfoil, comparing them with non-hyperreduced ROMs in terms of both accuracy and efficiency.

Chapter 2

Fundamental Methods

This chapter outlines the fundamental methods that underpin this work. The first section provides a brief overview of the discontinuous Galerkin method and the second describes the approach used to solve for steady-state solutions of the full-order model.

2.1 Discontinuous Galerkin Method

The in-house CFD solver PHiLiP [38] employed in this work is based on the discontinuous Galerkin method, therefore a brief outline of this method will be given. This constitutes the full-order model (FOM) used in this work. The governing equations for physical flow phenomena can be expressed as the following conservation law:

$$\mathbf{R}(\mathbf{w}) = \frac{\partial \mathbf{w}}{\partial t} + \nabla \cdot \mathbf{F}(\mathbf{w}) - \mathbf{S}(\mathbf{w}) = 0, \quad (2.1)$$

where \mathbf{w} is the solution vector, $\mathbf{F}(\mathbf{w})$ is the flux vector, $\mathbf{S}(\mathbf{w})$ is a source term, and $\mathbf{R}(\mathbf{w})$ is the residual. CFD invokes the Navier-Stokes equations, which take the above form and are used

to solve problems involving fluid flows. These equations are an example of parameterized, time-dependent, nonlinear partial differential equations (PDEs).

Discontinuous Galerkin methods are a class of high-order finite element methods which were developed as a response to the shortcomings of the current methods for discretizing PDEs [39]. The goal is to combine the high-order accuracy enabled through local approximation by finite element methods with the stability provided by finite volume methods in wave-dominated problems enabled by the discontinuities at cell interfaces [16, 39]. The resulting framework is known as the discontinuous Galerkin finite element method (DG) [39].

A discrete numerical solution $\mathbf{w}_h \in \mathbb{R}^N$ is introduced which approximates the solution \mathbf{w} to Equation 2.1 over the computational domain Ω_h . This discrete solution has dimension N , which will represent the FOM dimension for the entirety of this work. The computational domain Ω_h approximates the physical domain Ω with boundary Γ using a collection of non-overlapping elements $\Omega_k \in \Omega_h$ with boundaries $\partial\Omega_k$. The discontinuous Galerkin method seeks a solution represented by the piecewise sum of local solutions $\mathbf{w}_h^k(\mathbf{x})$ over each element:

$$\mathbf{w}_h(\mathbf{x}) = \bigoplus_{\Omega_k \in \Omega_h} \mathbf{w}_h^k(\mathbf{x}). \quad (2.2)$$

The local solution on each element can be written in the form:

$$\mathbf{w}_h^k(\mathbf{x}) = \sum_{i=1}^{N_k(p_k)} \mathbf{w}_i^k \phi_{h,i}(\mathbf{x}), \quad (2.3)$$

where p_k is the degree of the polynomial approximation, $N_k(p_k)$ denotes the number of solution points within the element, and $\phi_{h,i}$ is the i^{th} polynomial shape function on element k . To obtain the local solutions, the weak formulation of Equation 2.1 is needed. Multiplying the equation by an arbitrary test function ψ , we then require the residual to be orthogonal to this function. In a Galerkin scheme, the test functions are from the same space as the

polynomial shape functions used in the solution, meaning $\boldsymbol{\psi} = \boldsymbol{\phi}$. Thus, multiplying equation 2.1 by $\boldsymbol{\phi}_h$ and integrating by parts over each element, we obtain:

$$\int_{\Omega_k} \boldsymbol{\phi}_h \frac{\partial \mathbf{w}_h}{\partial t} d\Omega - \int_{\Omega_k} \nabla \boldsymbol{\phi}_h \cdot \mathbf{F}(\mathbf{w}_h) d\Omega + \int_{\partial\Omega_k} \boldsymbol{\phi}_h \mathbf{F}^*(\mathbf{w}_h) \cdot \mathbf{n} d\Gamma - \int_{\Omega_k} \boldsymbol{\phi}_h \mathbf{S}(\mathbf{w}_h) d\Omega = 0, \quad (2.4)$$

where \mathbf{n} represents the normal vector to the element boundary and \mathbf{F}^* represents a numerical flux that ensures conservation of the governing equations across cell interfaces.

Substituting the discrete solution \mathbf{w}_h^k from Equation 2.3 into Equation 2.4 yields a local system of equations for each element. The local contributions from all the elements are then assembled to form a global discrete system of ordinary differential equations (ODEs). This system is written in semi-discrete form, along with an output functional of interest, \mathcal{J} , as:

$$\begin{aligned} \mathbf{M} \frac{d\mathbf{w}_h}{dt} + \mathbf{R}(\mathbf{w}_h(t), \boldsymbol{\mu}, t) &= 0, \\ \mathcal{J} &= \mathcal{J}(\mathbf{w}_h(t), \boldsymbol{\mu}, t), \end{aligned} \quad (2.5)$$

where \mathbf{M} is the global mass matrix, \mathbf{R} is the discretized spatial residual, and $\boldsymbol{\mu}$ is a set of parameters.

2.2 Full-Order Model

The current work focuses only on steady-state solutions. Therefore, a steady-state solution to the semi-discrete system of nonlinear ordinary differential equations given by Equation 2.5 is found using an implicit method with pseudo-time stepping. To begin, using the backward Euler method, we can linearize the residual at nonlinear iteration k about the solution $\mathbf{w}_h^{(k)}$.

For brevity $\mathbf{R}(\mathbf{w}_h^{(k)}) \equiv \mathbf{R}(\mathbf{w}_h(t^{(k)}), \boldsymbol{\mu}, t^{(k)})$, the linearization then reads:

$$\mathbf{R}(\mathbf{w}_h^{(k+1)}) = \mathbf{R}(\mathbf{w}_h^{(k)}) + \left. \frac{\partial \mathbf{R}}{\partial \mathbf{w}_h} \right|_{\mathbf{w}_h^{(k)}} (\mathbf{w}_h^{(k+1)} - \mathbf{w}_h^{(k)}). \quad (2.6)$$

Re-arranging Equation 2.5 for time step $(k + 1)$, we have:

$$\mathbf{M} \frac{d\mathbf{w}_h}{dt} = -\mathbf{R}(\mathbf{w}_h^{(k+1)}). \quad (2.7)$$

Using the definition for $\mathbf{R}(\mathbf{w}_h^{(k+1)})$ from Equation 2.6 and substituting it into Equation 2.7, we find:

$$\mathbf{M} \frac{d\mathbf{w}_h}{dt} = -\mathbf{R}(\mathbf{w}_h^{(k)}) - \left. \frac{\partial \mathbf{R}}{\partial \mathbf{w}_h} \right|_{\mathbf{w}_h^{(k)}} (\mathbf{w}_h^{(k+1)} - \mathbf{w}_h^{(k)}). \quad (2.8)$$

We can then find an approximation of the time derivative of the solution:

$$\frac{d\mathbf{w}_h}{dt} \approx \frac{(\mathbf{w}_h^{(k+1)} - \mathbf{w}_h^{(k)})}{\Delta\tau}. \quad (2.9)$$

Plugging this into Equation 2.8 and re-arranging, we obtain:

$$\left[\left. \frac{\partial \mathbf{R}}{\partial \mathbf{w}_h} \right|_{\mathbf{w}_h^{(k)}} + \frac{\mathbf{M}}{\Delta\tau} \right] (\mathbf{w}_h^{(k+1)} - \mathbf{w}_h^{(k)}) = -\mathbf{R}(\mathbf{w}_h^{(k)}). \quad (2.10)$$

The local CFL number, given by $\frac{c\Delta\tau}{\Delta x}$ where c is a constant, is used to determine the pseudo-time step τ on each element. The pseudo-time step begins small and is gradually increased as the solution approaches steady state, causing the second term in the square brackets ($\frac{\mathbf{M}}{\Delta\tau}$) to vanish. Therefore, for iterations $k = 1, \dots, K$ until convergence of the nonlinear residual,

the Newton steps can be written as:

$$\begin{aligned} \left[\frac{\partial \mathbf{R}^{(k)}}{\partial \mathbf{w}_h} \right] \mathbf{p}^{(k)} &= -\mathbf{R}^{(k)}, \\ \mathbf{w}_h^{(k+1)} &= \mathbf{w}_h^{(k)} + \mathbf{r}^{(k)} \mathbf{p}^{(k)}, \end{aligned} \tag{2.11}$$

where $\frac{\partial \mathbf{R}^{(k)}}{\partial \mathbf{w}_h}$ is the Jacobian, $\mathbf{p}^{(k)}$ is the search direction and $\mathbf{r}^{(k)}$ the step length, found using a line search algorithm. Finally, we formulate the complete full-order steady-state problem as:

$$\begin{aligned} \mathbf{R}(\mathbf{w}_h, \boldsymbol{\mu}) &= 0, \\ \mathcal{J} &= \mathcal{J}(\mathbf{w}_h, \boldsymbol{\mu}). \end{aligned} \tag{2.12}$$

In the rest of this work, the approximate numerical solution \mathbf{w}_h will be referred to as \mathbf{w} for brevity. It is important to remember that the FOM itself may have error introduced by the discrete approximation of the governing equations. For the discontinuous Galerkin method, this error could be related to both grid size and polynomial order [16]. All error in the reduced-order model will however only be measured with respect to the FOM solution \mathbf{w} , as the error introduced by the higher-order method prior to the model reduction cannot be controlled in any way by the reduced-order modelling procedure.

Chapter 3

Model Reduction

This chapter discusses the various components required to construct a hyperreduced reduced-order model (HROM). These include the creation of a trial basis, or reduced-order basis (ROB), via proper orthogonal decomposition, onto which the full-order model (FOM) will be projected. The basis is then used to derive an approximate representation $\tilde{\mathbf{w}}$ for FOM solution \mathbf{w} . By projecting the system of nonlinear equations onto the test basis \mathbf{W} , Newton's method can be employed to solve for the unknown reduced-order solution $\hat{\mathbf{w}}$. The least-squares Petrov-Galerkin (LSPG) framework is introduced. Then, a brief description of the two types of hyperreduction methods is given, followed by a more detailed description of the chosen energy-conserving sampling and weighting (ECSW) approach.

3.1 Proper Orthogonal Decomposition

As mentioned in the introduction, projection-based reduced order models take advantage of the lower dimensionality of the FOM solution manifold. They capture the physics of the problem by projecting the FOM onto a subspace of a smaller dimension, typically using a properly trained reduced-order basis $\mathbf{V} \in \mathbb{R}^{N \times n}$ [40]. This basis is a matrix whose columns,

often referred to as basis vectors or modes, span the lower dimensional subspace [16]. Among the many possible choices for basis vectors, the Karhunen-Loève basis has been shown to perform best in the context of model reduction [34]. In fluid dynamics, the expansion procedure conducted to build the basis is referred to as proper orthogonal decomposition (POD). POD can extract dominant features in data or decompose a function into its underlying modes, which can then be used to build a reduced-order basis (ROB). Further details on the POD procedure and its derivation can be found in [16].

The most common approach for identifying POD modes is singular value decomposition (SVD), a concept from linear algebra. To construct the ROB, a set of solution samples $\mathbf{w}^s \in \mathbb{R}^N$ at various parameter combinations must be collected. The way in which these samples are distributed in the parameter domain will be discussed further in Chapter 5. Using a reference solution $\mathbf{w}_{\text{ref}} \in \mathbb{R}^N$, which in the previous work [16] was chosen to be the mean of the solution samples, the snapshots used to determine the basis vectors are of the form $\mathbf{s} = \{\mathbf{w}^s - \mathbf{w}_{\text{ref}}\}_{s=1}^n$ to ensure consistency. These can be assembled into a matrix \mathbf{S} :

$$\mathbf{S} = \begin{bmatrix} \mathbf{s}_1 & \mathbf{s}_2 & \dots & \mathbf{s}_n \end{bmatrix} \in \mathbb{R}^{N \times n}. \quad (3.1)$$

The SVD of the above matrix would then be given by:

$$\mathbf{S} = \mathbf{U}\mathbf{\Sigma}\mathbf{Z}^T, \quad (3.2)$$

where $\mathbf{U} \in \mathbb{R}^{N \times N}$, $\mathbf{Z} \in \mathbb{R}^{n \times n}$ are orthogonal matrices and $\mathbf{\Sigma} = \text{diag}(\sigma_1, \dots, \sigma_k, 0, \dots, 0)$ is an $N \times n$ diagonal matrix containing singular values σ , where $k = \min(N, n)$. The columns of \mathbf{U} are equivalent to the POD modes identified using an eigenvalue problem. In this work, the number of snapshots n is assumed to be much smaller than the dimension of the FOM N . As a result, the last $N - n$ columns of \mathbf{U} are arbitrary, as they correspond to zero singular

values. Therefore, by employing a “thin” SVD without truncation, the first n columns of \mathbf{U} are selected as the basis vectors for the ROB \mathbf{V} which spans the reduced-order subspace, commonly referred to as the trial space.

3.2 Projection-Based Reduced-Order Model

Once a basis $\mathbf{V} \in \mathbb{R}^{N \times n}$ has been determined using POD, the next step is to find an approximation of the solution $\mathbf{w} \in \mathbb{R}^N$ to the FOM Equation 2.12. Projection-based ROMs operate under the assumption that these solutions lie in the reduced-order subspace. At a given parameter location $\boldsymbol{\mu}$, we would like to then find an approximate solution $\tilde{\mathbf{w}} \in \mathbb{R}^N$ of the form:

$$\tilde{\mathbf{w}} = \mathbf{w}_{\text{ref}} + \mathbf{V}\hat{\mathbf{w}}, \quad (3.3)$$

where $\mathbf{w}_{\text{ref}} \in \mathbb{R}^N$ is the reference state and $\hat{\mathbf{w}} \in \mathbb{R}^n$ is the unknown reduced-order solution. Substituting the approximate solution in Equation 3.3 into Equation 2.12 results in an overdetermined system as there are N equations and n unknowns in $\hat{\mathbf{w}}$. The system of equations is then projected onto a test subspace via the test basis $\mathbf{W} \in \mathbb{R}^{N \times n}$. This results in the following reduced-order system of nonlinear equations:

$$\mathbf{W}^T \mathbf{R}(\mathbf{w}_{\text{ref}} + \mathbf{V}\hat{\mathbf{w}}, \boldsymbol{\mu}) = \hat{\mathbf{R}}(\mathbf{w}_{\text{ref}} + \mathbf{V}\hat{\mathbf{w}}, \boldsymbol{\mu}) = \mathbf{0}^T, \quad (3.4)$$

where $\hat{\mathbf{R}} \in \mathbb{R}^n$ is the reduced-order residual. Using a Taylor series expansion to approximate the solution, we have:

$$\hat{\mathbf{R}}(\mathbf{w}_{\text{ref}} + \mathbf{V}\hat{\mathbf{w}}^{(k+1)}, \boldsymbol{\mu}) \approx \hat{\mathbf{R}}(\mathbf{w}_{\text{ref}} + \mathbf{V}\hat{\mathbf{w}}^{(k)}, \boldsymbol{\mu}) + \frac{\partial \hat{\mathbf{R}}^{(k)}}{\partial \hat{\mathbf{w}}} (\hat{\mathbf{w}}^{(k+1)} - \hat{\mathbf{w}}^{(k)}), \quad (3.5)$$

and given that $\hat{\mathbf{R}}(\mathbf{w}_{\text{ref}} + \mathbf{V}\hat{\mathbf{w}}, \boldsymbol{\mu}) = \mathbf{0}^T$, then:

$$\mathbf{0}^T = \mathbf{W}^T \mathbf{R}(\mathbf{w}_{\text{ref}} + \mathbf{V}\hat{\mathbf{w}}^{(k)}, \boldsymbol{\mu}) + \frac{\partial(\mathbf{W}^T \mathbf{R}(\mathbf{w}_{\text{ref}} + \mathbf{V}\hat{\mathbf{w}}^{(k)}, \boldsymbol{\mu}))}{\partial \hat{\mathbf{w}}} (\hat{\mathbf{w}}^{(k+1)} - \hat{\mathbf{w}}^{(k)}). \quad (3.6)$$

Ignoring the second-order sensitivities created by taking the derivative of the test basis and using the chain rule on the remaining derivative of the residual with respect to $\hat{\mathbf{w}}$, we find:

$$\mathbf{W}^T \frac{\partial \mathbf{R}^{(k)}}{\partial \mathbf{w}} \mathbf{V} (\hat{\mathbf{w}}^{(k+1)} - \hat{\mathbf{w}}^{(k)}) = -\mathbf{W}^T \mathbf{R}(\mathbf{w}_{\text{ref}} + \mathbf{V}\hat{\mathbf{w}}^{(k)}, \boldsymbol{\mu}). \quad (3.7)$$

Therefore, the Newton iteration for $k = 1, \dots, K$ are:

$$\begin{aligned} \left[\mathbf{W}^T \frac{\partial \mathbf{R}^{(k)}}{\partial \mathbf{w}} \quad \mathbf{V} \right] \mathbf{p}^{(k)} &= -\mathbf{W}^T \mathbf{R}^{(k)}, \\ \hat{\mathbf{w}}^{(k+1)} &= \hat{\mathbf{w}}^{(k)} + \mathbf{r}^{(k)} \mathbf{p}^{(k)}, \end{aligned} \quad (3.8)$$

where $\mathbf{p}^{(k)}$ is the search direction and $\mathbf{r}^{(k)}$ the step length.

3.3 Least-squares Petrov-Galerkin Projection

The final component of the PROM framework that must be selected is the test basis \mathbf{W} . Two options are available, the Galerkin and Petrov-Galerkin methods. As mentioned in the introduction, the Petrov-Galerkin framework has been shown to be more robust and accurate for CFD problems [14,17]. Specifically, the least-squares Petrov-Galerkin (LSPG) projection offers several attractive properties, including optimality [13] and monotonicity [41], which are discussed in detail in [16] and motivate its selection for use in this work.

In the case of the least-squares Petrov-Galerkin projection [42,43], the test basis is constructed such that $\mathbf{W} \equiv \frac{\partial \mathbf{R}}{\partial \mathbf{w}} \mathbf{V}$. Recalling 3.8, the Newton iterations for $k = 1, \dots, K$ in the

LSPG framework become:

$$\begin{aligned} \left[\mathbf{V}^T \frac{\partial \mathbf{R}^{(k)T}}{\partial \mathbf{w}} \quad \frac{\partial \mathbf{R}^{(k)}}{\partial \mathbf{w}} \quad \mathbf{V} \right] \mathbf{p}^{(k)} &= -\mathbf{V}^T \frac{\partial \mathbf{R}^{(k)T}}{\partial \mathbf{w}} \mathbf{R}^{(k)}, \\ \hat{\mathbf{w}}^{(k+1)} &= \hat{\mathbf{w}}^{(k)} + \mathbf{r}^{(k)} \mathbf{p}^{(k)}, \end{aligned} \quad (3.9)$$

where $\frac{\partial \mathbf{R}^{(k)}}{\partial \mathbf{w}} \in \mathbb{R}^{N \times N}$ is the Jacobian, $\mathbf{p}^{(k)}$ is the search direction and $\mathbf{r}^{(k)}$ the step length. It can be seen here that the test basis updates with each Newton iteration via the addition of the Jacobian, allowing it to better capture nonlinear effects [13].

It can be shown that an approximation will be consistent throughout a ROM simulation, provided it is consistent at the initial condition. This consistency can be enforced by setting $\tilde{\mathbf{w}}^{(0)} = \mathbf{w}_{\text{ref}} + \mathbf{V} \hat{\mathbf{w}}^{(0)}$ [44]. However, there is no guarantee that any arbitrary choice of $\tilde{\mathbf{w}}^{(0)}$ can be represented in this form. Therefore, a projection can be used to project an arbitrary initial condition $\mathbf{w}^{(0)}$ onto the reduced-order subspace. This is achieved by setting $\hat{\mathbf{w}}^{(0)} = \mathbf{V}^T (\mathbf{w}^{(0)} - \mathbf{w}_{\text{ref}})$. As a result, the initial condition is modified to a projected initial condition $\tilde{\mathbf{w}}_{\text{proj}}^{(0)}$:

$$\tilde{\mathbf{w}}_{\text{proj}}^{(0)} = \mathbf{w}_{\text{ref}} + \mathbf{V} (\mathbf{V}^T (\mathbf{w}^{(0)} - \mathbf{w}_{\text{ref}})). \quad (3.10)$$

3.4 Hyperreduction

In Newton's method shown in Equation 3.9 above, it can be seen that the residual and Jacobian must be re-computed at each iteration. This is one of the computational bottlenecks for a ROM of a non-linear, parametric model, as the cost of recomputing and projecting these quantities would scale with the dimension of the FOM N . Noting that the right-hand side of the first line in equation 3.9 is the reduced-order residual $\hat{\mathbf{R}}$, each iteration requires the evaluation of the high-dimensional residual \mathbf{R} , followed by multiplication with the test basis \mathbf{V}^T , which has asymptotic complexity $O(Nn)$ [45]. Therefore, while ROMs of the above

form can provide accurate approximations of the FOM, they may not be significantly less computationally expensive. For the class of problems discussed in this work, an inexact methodology is required for addressing this computational bottleneck. These methods are referred to as hyperreduction methods, despite the fact that many of the approaches were developed well before the use of the term “hyperreduction” was coined by Ryckelynck in [46]. These techniques can be grouped into two categories: *approximate-then-project* methods and *project-then-approximate* methods. This section will provide an overview of these two categories and examples of the more widely adopted versions of these methods.

3.4.1 Approximate-then-Project

Following their naming convention, these methods first approximate the nonlinear high-dimensional model quantity and then project the approximation onto the left reduced-order basis \mathbf{W} . [18] notes that the underlying concept of this approach finds its origin in the gappy POD method [47], which was originally developed for image reconstruction.

These methods share the following steps, outlined by Farhat in [18]. For a nonlinear, high-dimensional vector like the residual $\mathbf{R}(\mathbf{w}_{\text{ref}} + \mathbf{V}\hat{\mathbf{w}}, \boldsymbol{\mu}) \in \mathbb{R}^N$, a small number $m \ll N$ of empirically derived basis functions are used to approximate the operator. These basis functions are stored in the columns of the matrix $\mathbf{U} \in \mathbb{R}^{N \times m}$. The residual can then be written as:

$$\mathbf{R}(\mathbf{w}_{\text{ref}} + \mathbf{V}\hat{\mathbf{w}}, \boldsymbol{\mu}) \approx \hat{\mathbf{R}}_N(\mathbf{w}_{\text{ref}} + \mathbf{V}\hat{\mathbf{w}}, \boldsymbol{\mu}) = \mathbf{U}\mathbf{R}_m(\mathbf{w}_{\text{ref}} + \mathbf{V}\hat{\mathbf{w}}, \boldsymbol{\mu}), \quad (3.11)$$

where $\mathbf{R}_m \in \mathbb{R}^m$ is the vector of reduced-order coordinates. Using a minimization of the 2-norm, the optimal approximation of \mathbf{R} becomes $\hat{\mathbf{R}}_N$ if \mathbf{R}_m has the following form:

$$\begin{aligned}\hat{\mathbf{R}}_N(\mathbf{w}_{\text{ref}} + \mathbf{V}\hat{\mathbf{w}}, \boldsymbol{\mu}) &= \mathbf{U}\mathbf{R}_m^{\text{opt}}(\mathbf{w}_{\text{ref}} + \mathbf{V}\hat{\mathbf{w}}, \boldsymbol{\mu}) \\ &= \mathbf{U}(\mathbf{U}^{\mathcal{I}})^\dagger \mathbf{R}^{\mathcal{I}}(\mathbf{w}_{\text{ref}} + \mathbf{V}\hat{\mathbf{w}}, \boldsymbol{\mu}),\end{aligned}\tag{3.12}$$

where $(\cdot)^{\mathcal{I}}$ represents the restriction of a vector or matrix to its rows specified by the elements of \mathcal{I} and the superscript \dagger designates the Moore-Penrose pseudo-inverse [18]. In the above optimal case, the number of empirical basis functions m is equal to $|\mathcal{I}|$ and \mathcal{I} is the set of computed rows of the FOM residual \mathbf{R} at which the 2-norm error is minimized. Therefore, $\mathbf{R}^{\mathcal{I}} \in \mathbb{R}^{|\mathcal{I}|}$ and $\mathbf{U}^{\mathcal{I}} \in \mathbb{R}^{|\mathcal{I}| \times m}$. The above approximation must then be projected onto the test basis. The hyperreduced residual then becomes:

$$\begin{aligned}\tilde{\mathbf{R}}(\mathbf{w}_{\text{ref}} + \mathbf{V}\hat{\mathbf{w}}, \boldsymbol{\mu}) &= \mathbf{W}\hat{\mathbf{R}}_N(\mathbf{w}_{\text{ref}} + \mathbf{V}\hat{\mathbf{w}}, \boldsymbol{\mu}) \\ &= \mathbf{W}\mathbf{U}(\mathbf{U}^{\mathcal{I}})^\dagger \mathbf{R}^{\mathcal{I}}(\mathbf{w}_{\text{ref}} + \mathbf{V}\hat{\mathbf{w}}, \boldsymbol{\mu}).\end{aligned}\tag{3.13}$$

This splits the computation of the hyperreduced residual at a new parameter location $\boldsymbol{\mu}$ into two components. The first is the precomputation of the matrix-matrix product $\mathbf{Q} = \mathbf{W}\mathbf{U}(\mathbf{U}^{\mathcal{I}})^\dagger \in \mathbb{R}^{n \times |\mathcal{I}|}$, which can be conducted offline. The second is the online reconstruction of the approximation, given by $\tilde{\mathbf{R}}(\mathbf{w}_{\text{ref}} + \mathbf{V}\hat{\mathbf{w}}, \boldsymbol{\mu}) = \mathbf{Q}\mathbf{R}^{\mathcal{I}}(\mathbf{w}_{\text{ref}} + \mathbf{V}\hat{\mathbf{w}}, \boldsymbol{\mu}) \in \mathbb{R}^n$. This online computation requires $\mathcal{O}(n|\mathcal{I}|)$ operations, where $n \ll N$ and $|\mathcal{I}| \ll N$.

Some of the notable approximate-then-project hyperreduction methods include the empirical interpolation method (EIM) [20, 21] and its discrete version discrete EIM (DEIM) [22, 48], the missing point approach [49], and the Gauss-Newton with approximate tensors (GNAT) method [13, 50]. These approaches differ in terms of the theoretical support they have, the types of projection frameworks to which they have been applied, and their level

of success when applied to CFD problems. EIM was originally derived at the continuous level for PDEs and has some theoretical support for elliptic problems. The discrete version, DEIM, has become one of the most popular approaches. However, both methods have been applied only in Galerkin frameworks and have seen limited use in aerodynamics. When used, such as in Carlberg et al. [14, 50], they have been reported to exhibit temporal instabilities for turbulent unsteady flows [51]. The GNAT method, on the other hand, was formulated at the discrete level for Petrov-Galerkin frameworks and designed around approximations that satisfy consistency and discrete-optimality conditions [50]. It has proven highly effective for nonlinear structural dynamics problems [13], demonstrating robustness, accuracy, and excellent computational cost savings, as well as for benchmark turbulent viscous flow problems [50]. However, Washabaugh [43] notes that the GNAT method can be expensive for parameterized steady aerodynamic problems. Given the limitations of the approximate-then-project method, these approaches are not considered in this work.

3.4.2 Project-then-Approximate

Project-then-approximate methods first project the full-order model quantities onto the left ROB \mathbf{W} , then approximate the reduced-order vectors and matrices. These methods typically decompose the reduced-order quantities into a summation over the elements in the computational mesh and then seek to approximate the quantities by including only the contributions from a subset of the mesh elements, referred to as the reduced mesh. The way this set is determined varies, but all of the methods can be interpreted as generalized quadrature rules, where an empirical set of training data is collected, and a set of quadrature “points” and associated weights are learned in a supervised procedure [15]. All project-then-approximate methods conduct some form of mesh sampling in which the quadrature points are mesh elements. Some examples of this class of approaches include the empirical quadrature proce-

dure (EQP) [52,53], the energy-conserving sampling and weighting (ECSW) method [19,23], and the empirical cubature method (ECM) [54]. The EQP method is unique in that it can control the error introduced by hyperreduction in the norm of the solution [53], and was later extended to control the error in the quantity of interest [55]. This differs from other hyperreduction methods that control the error introduced into the quantity being approximated, such as the residual, which must then be “tuned” to achieve the desired functional error tolerance [29]. However, both the EQP and ECM methods have only been applied in Galerkin frameworks.

Farhat et al. [23] note that previous hyperreduction approaches have primarily focused on the accuracy of the approximation of the FOM quantities, but have given little consideration to important properties of the resulting HROM, such as preservation of symmetry or numerical stability [23]. The ECSW method was developed to preserve both symmetry and stability using the concepts of mesh sampling and the principle of virtual work [23]. It has been shown that, in second-order hyperbolic problems, this approach preserves the Lagrangian structure associated with Hamilton’s principle, which in turn enables the preservation of the numerical stability properties of the discrete system [19]. When applied to realistic structural dynamics problems, the use of the ECSW method results in stable and accurate HROMs, whereas HROMs built with DEIM fail due to numerical instability [19]. [15] extended the ECSW method to Petrov-Galerkin PROMs and demonstrated its success in constructing robust, accurate, and computationally efficient models for CFD applications, particularly those associated with convection-dominated viscous flows. Due to these properties, the ECSW method is the most suitable candidate for hyperreduction in this work.

3.5 Energy-conserving Sampling and Weighting Method

Consider the discretization of a spatial domain into N_e mesh entities making up the set \mathcal{E} , in the case of finite difference semi-discretization these would be vertices. The reduced-order residual for a steady-state problem can be written as:

$$\hat{\mathbf{R}}(\hat{\mathbf{w}}; \boldsymbol{\mu}) = \sum_{e \in \mathcal{E}} \mathbf{W}^T \mathbf{L}_e^T \mathbf{R}_e(\mathbf{L}_{e^+}(\mathbf{w}_{\text{ref}} + \mathbf{V}\hat{\mathbf{w}}); \boldsymbol{\mu}), \quad (3.14)$$

where $\mathbf{L}_e \in \{0, 1\}^{d_e \times N}$ is a boolean matrix localizing a FOM vector to the d_e degrees of freedom (DOFs) associated with the mesh element e . $\mathbf{R}_e \in \mathbb{R}^{d_e}$ is the contribution of this element to the global FOM residual; the spatial stencil determines the set of entities n_{e^+} required to evaluate \mathbf{R}_e , which can include the neighbours of e . Similar to \mathbf{L}_e , $\mathbf{L}_{e^+} \in \{0, 1\}^{(d_e n_{e^+}) \times N}$ is the boolean matrix that localizes a FOM vector to the DOFs associated with the entities n_{e^+} . It is assumed that there exists a subset of all the mesh entities $\tilde{\mathcal{E}} \subset \mathcal{E}$, with size $\tilde{N}_e = |\tilde{\mathcal{E}}| \ll N_e$, such that the reduced residual can be approximated with a smaller number of entities. The hyperreduced residual vector $\tilde{\mathbf{R}}$ can be written as:

$$\hat{\mathbf{R}}(\hat{\mathbf{w}}; \boldsymbol{\mu}) \approx \tilde{\mathbf{R}}(\hat{\mathbf{w}}; \boldsymbol{\mu}) = \sum_{e \in \tilde{\mathcal{E}}} \xi_e \mathbf{W}^T \mathbf{L}_e^T \mathbf{R}_e(\mathbf{L}_{e^+}(\mathbf{w}_{\text{ref}} + \mathbf{V}\hat{\mathbf{w}}); \boldsymbol{\mu}), \quad (3.15)$$

and interpreted as a generalized quadrature rule with a set of mesh element weights $\{\xi_e \mid e \in \tilde{\mathcal{E}}\}$. Note that this is an approximation of the projected residual $\hat{\mathbf{R}}$ not the FOM residual \mathbf{R} , therefore $\tilde{\mathbf{R}} \in \mathbb{R}^n$. With no additional approximation, the hyperreduced Jacobian $\tilde{\mathbf{J}}$ can be written as:

$$\frac{\partial \hat{\mathbf{R}}}{\partial \mathbf{w}}(\hat{\mathbf{w}}; \boldsymbol{\mu}) \approx \tilde{\mathbf{J}} = \sum_{e \in \tilde{\mathcal{E}}} \xi_e \mathbf{W}^T \mathbf{L}_e^T \mathbf{J}_e(\mathbf{L}_{e^+}(\mathbf{w}_{\text{ref}} + \mathbf{V}\hat{\mathbf{w}}); \boldsymbol{\mu}) \mathbf{L}_{e^+} \mathbf{V}, \quad (3.16)$$

where $\mathbf{J}_e \in \mathbb{R}^{d_e \times d_e^+}$ is the Jacobian matrix of \mathbf{R}_e with respect to the DOFs associated with the n_{e^+} entities involved in its evaluation [45].

3.5.1 Reduced Mesh Set Selection

[40] observed that, particularly in steady-state problems, residual values can often be very close to zero. This can result in poor training data and lead to inaccuracies in the reduced mesh and the associated weights. The challenges of applying the residual-based training strategy to steady-state problems are discussed in detail in [40]. A summary of these challenges is provided below:

The hyperreduction will be conducted in two stages: first, the offline computation of the reduced mesh and accompanying quadrature weights, followed by the online use of these approximations to evaluate the projected quantities within the LSPG framework. Two ECSW approaches were compared in the offline computation stage of this work. One approach trains the weights using the residual of a subset of the converged solution snapshots used to build the ROB, as in [15] and [23]. The other approach uses the Jacobian of the snapshots instead. This was motivated by our own initial results, which identified challenges with the conditioning of the matrices produced by the residual approach. [40] observed that, particularly in steady-state problems, the residual values can often be very close to zero. This can result in poor training data and lead to inaccuracies in the reduced mesh and the associated weights. The challenges of applying the residual-based training strategy to steady-state problems are discussed in detail in [40]. A shortened list is provided here:

- If the ROB is selected without truncation of the singular value decomposition, the snapshot solution will have an exact approximate representation in the reduced-order space. As a result, the residuals—and thus the training data—would nearly vanish for these computed solutions, which would be equivalent to training using numerical noise.

This will likely lead to inaccurate approximations. Furthermore, [40] notes that this scenario would invalidate the early stopping criterion in Equation 3.26 (introduced later in the non-negative least squares (NNLS) problem) that is used to build the reduced mesh.

- [40] also explains that parametric steady-state problems generally require fewer solution snapshots than unsteady ones, meaning that the available training data is more limited which can negatively impact the online performance of the hyperreduction approximations. This could create a challenge in the adaptive sampling procedure, as the purpose of the approach is to begin with as few snapshots as possible. If hyperreduction is then incorporated, there will be minimal training data available at earlier iterations. This could make the hyperreduction inaccurate, causing the error indicators at the ROM points to not capture the true error distribution, thereby affecting the placement of new FOM snapshots.

The next two sections outline the residual-based and Jacobian-based approaches for assembling the training data to determine the reduced mesh and associated weights. The final section describes the optimization problem arising from the training data.

3.5.2 Residual-based ECSW Training Process

To begin, each solution snapshot used to find the right ROB \mathbf{V} in equation 3.1 can be represented on the lower dimension with the following projection [15]:

$$\hat{\mathbf{w}}_s = \mathbf{V}^T(\mathbf{w}^s - \mathbf{w}_{\text{ref}}), \quad (3.17)$$

which can then be converted into an approximation of the FOM vector of dimension N through the equation:

$$\begin{aligned}\tilde{\mathbf{w}}_s &= \mathbf{w}_{\text{ref}} + \mathbf{V}\hat{\mathbf{w}}_s, \\ &= \mathbf{w}_{\text{ref}} + \mathbf{V}\mathbf{V}^T(\mathbf{w}^s - \mathbf{w}_{\text{ref}}).\end{aligned}\tag{3.18}$$

The original set of snapshots \mathcal{S} contains n FOM solutions as shown in the matrix in Equation 3.1. We create a new set \mathcal{S}_H of size N_s which contains all or a subset of the snapshots in \mathcal{S} , i.e. $\mathcal{S}_H \subseteq \mathcal{S}$. For each snapshot in \mathcal{S}_H , the contributions from each mesh element to the discrete ROM residual can be broken into:

$$\begin{aligned}\mathbf{c}_{se} &= \mathbf{W}^T \mathbf{L}_e^T \mathbf{R}_e(\mathbf{L}_e + (\mathbf{w}_{\text{ref}} + \mathbf{V}\mathbf{V}^T(\mathbf{w}^s - \mathbf{w}_{\text{ref}})); \boldsymbol{\mu}) \in \mathbb{R}^n, \\ \mathbf{d}_s &= \sum_{e \in \tilde{\mathcal{E}}} c_{se} \in \mathbb{R}^n \quad s = 1, \dots, N_s \quad e = 1, \dots, N_e.\end{aligned}\tag{3.19}$$

The training data can then be organized in block form and the exact assembly of this data will be as follows:

$$\mathbf{C}\mathbf{1} = \mathbf{d},\tag{3.20}$$

where,

$$\mathbf{C} = \begin{bmatrix} c_{11} & \dots & c_{1N_e} \\ \vdots & \ddots & \vdots \\ c_{N_s 1} & \dots & c_{N_s N_e} \end{bmatrix} \in \mathbb{R}^{(N_s n) \times N_e} \quad \text{and} \quad \mathbf{d} = \begin{bmatrix} d_1 \\ \vdots \\ d_{N_s} \end{bmatrix} \in \mathbb{R}^{(N_s n)}\tag{3.21}$$

and $\mathbf{1}$ is a N_e length vector of ones. The above matrix and vector will then be fed into the NNLS problem to solve for the weights and mesh entities used for the hyperreduction approximation, which will be described in more detail in an upcoming section.

3.5.3 Jacobian-based ECSW Training Process

Similar to the residual approach, a set \mathcal{S}_H of size N_s is created containing all or a subset of the snapshots in \mathcal{S} , i.e. $\mathcal{S}_H \subseteq \mathcal{S}$. For each snapshot in \mathcal{S}_H , the contributions from each mesh element to the discrete ROM Jacobian can be broken into:

$$\begin{aligned} \mathbf{c}_{se} &= \mathbf{Q}\mathbf{W}^T \mathbf{L}_e^T \mathbf{J}_e(\mathbf{L}_{e^+}(\mathbf{w}_{\text{ref}} + \mathbf{V}\hat{\mathbf{w}}); \boldsymbol{\mu}) \mathbf{L}_{e^+} \mathbf{V} \in \mathbb{R}^{n^2}, \\ \mathbf{d}_s &= \sum_{e \in \tilde{\mathcal{E}}} c_{se} \in \mathbb{R}^{n^2}, \quad s = 1, \dots, N_s, \quad e = 1, \dots, N_e, \end{aligned} \quad (3.22)$$

where \mathbf{Q} is the matrix that converts an $n \times n$ matrix into a column vector of dimension n^2 by stacking its columns on top of each other. The exact assembly of the training data can be written as:

$$\mathbf{C}\mathbf{1} = \mathbf{d}, \quad (3.23)$$

where,

$$\mathbf{C} = \begin{bmatrix} c_{11} & \dots & c_{1N_e} \\ \vdots & \ddots & \vdots \\ c_{N_s 1} & \dots & c_{N_s N_e} \end{bmatrix} \in \mathbb{R}^{(N_s n^2) \times N_e} \quad \text{and} \quad \mathbf{d} = \begin{bmatrix} d_1 \\ \vdots \\ d_{N_s} \end{bmatrix} \in \mathbb{R}^{(N_s n^2)} \quad (3.24)$$

and $\mathbf{1}$ is a N_e length vector of ones. It should be noted that while the number of rows in the \mathbf{C} matrix scales with the dimension n of the ROB \mathbf{V} for residual-based training, for the Jacobian-based training it scales with n^2 . This can lead to a significant increase in the training cost, which in this paper will be controlled by adjusting the number of snapshots in the set \mathcal{S}_H . Other methods for addressing this cost are also discussed in [40].

3.5.4 Non-Negative Least-Squares Problem

Once the training data has been assembled using one of the two approaches above, the goal is then to find a set of weights and accompanying mesh entities that can reproduce Equation 3.20 or 3.23 to a specified level of accuracy. Ideally, the vector of weights $\boldsymbol{\xi}$ replacing the vector of ones would be sparse, meaning the number of elements required to approximate the residual is much smaller than the total number of entities N_e . The result is an optimization problem of the following form [45]:

$$\begin{aligned} & \text{minimize } \|\boldsymbol{\xi}\|_0, \\ & \text{subject to } \|\mathbf{C}\boldsymbol{\xi} - \mathbf{d}\|_2 \leq \epsilon\|\mathbf{d}\|_2, \\ & \boldsymbol{\xi} \geq \mathbf{0}^T, \end{aligned} \tag{3.25}$$

where ϵ is an error tolerance used to control the accuracy of the hyperreduction. However, this is known to be an NP-hard problem. Using the results from [56], the non-convex problem above is replaced by a convex approximation known as a non-negative least-squares (NNLS) problem:

$$\begin{aligned} & \text{minimize } \|\mathbf{C}\boldsymbol{\xi} - \mathbf{d}\|_2^2, \\ & \text{subject to } \boldsymbol{\xi} \geq \mathbf{0}^T, \end{aligned} \tag{3.26}$$

with early stopping criteria $\|\mathbf{C}\boldsymbol{\xi} - \mathbf{d}\|_2 \leq \epsilon\|\mathbf{d}\|_2$ and $\boldsymbol{\xi} \geq \mathbf{0}^T$.

Note, in the NNLS problem, the closer ϵ is to unity, the more the solution approaches the results from the optimization problem in Equation 3.25 and the more sparse the resulting weights vector is. The closer ϵ is to zero the more accurate the approximation is [56]. For completeness, Algorithm 1 summarizes the full procedure to find the weights and reduced

mesh for the ECSW residual-based approach. The only difference for the Jacobian-based approach would be the entries of \mathbf{c}_{se} and \mathbf{d}_s .

Algorithm 1 Finding Weights for ECSW (Residual-based Training Data)

Inputs:

Set of Mesh Entities $\mathcal{E} = \{e_1, e_2, \dots, e_{n_e}\}$

POD Basis \mathbf{V} , Test Basis \mathbf{W}

Reference state \mathbf{w}_{ref}

Subset S_H of snapshots S (which were used to build the POD Basis)

NNLS Tolerance ϵ_{NNLS}

Outputs:

Reduced Mesh Element Set $\tilde{\mathcal{E}}$

Associate Set of Weights $\boldsymbol{\xi}_{\mathcal{E}} = \{\xi_e | e \in \tilde{\mathcal{E}}\}$

for $e \in \mathcal{E}$ **do**

Construct $\mathbf{L}_e \in \{0, 1\}^{d_e \times N}$ and $\mathbf{L}_{e^+} \in \{0, 1\}^{(d_e n_{e^+}) \times N}$

for $\mathbf{w}^s \in S_H$ **do**

Let $\mathbf{c}_{se} = \mathbf{W}^T \mathbf{L}_e^T \mathbf{R}_e(\mathbf{L}_{e^+}(\mathbf{w}_{\text{ref}} + \mathbf{V}\mathbf{V}^T(\mathbf{w}^s - \mathbf{w}_{\text{ref}})); \boldsymbol{\mu})$, where \mathbb{R}^{d_e} is the contribution of this element to the global FOM residual

Let $\mathbf{d}_s = \sum_{e \in \tilde{\mathcal{E}}} \mathbf{c}_{se}$

end for

end for

Solve the NNLS optimization problem for $\tilde{\mathcal{E}}$ and $\boldsymbol{\xi}_{\mathcal{E}}$:

minimize $\|\mathbf{C}\boldsymbol{\xi} - \mathbf{d}\|_2^2$

subject to $\boldsymbol{\xi} \geq \mathbf{0}^T$

with early termination criterion $\|\mathbf{C}\boldsymbol{\xi} - \mathbf{d}\|_2 \leq \epsilon_{\text{NNLS}} \|\mathbf{d}\|_2$

return $\boldsymbol{\xi}_{\mathcal{E}} = \{\xi_e | e \in \tilde{\mathcal{E}}\}$

3.6 Hyperreduced ROM Solution Evaluation with ECSW

With the resulting set of weights and mesh elements, the total number of elements on which the residual needs to be computed to approximate its reduced-order representation is expected to be much smaller than the total number of mesh elements N_e . The ECSW results can then be applied within the Newton iterations of the LSPG projection shown in Equation 3.9. At each new iteration, both the test basis \mathbf{W} , which depends on the Jacobian, and the residual \mathbf{R} must be re-evaluated. The process begins by solving for the test basis. Using

the weights and mesh entities derived from the NNLS optimization solution, the Jacobian can then be approximated with a reduced number of mesh entities. For the FOM, the Jacobian at a specific Newton iteration k can be broken down into the contributions from each element:

$$\mathbf{J}^{(k)} = \frac{\partial \mathbf{R}^{(k)}}{\partial \mathbf{w}} = \sum_{e \in \mathcal{E}} \mathbf{L}_e^T \mathbf{J}_e^{(k)} \mathbf{L}_{e^+}, \quad (3.27)$$

where $\mathbf{J}_e^{(k)}$ is the contribution of element e to the global, FOM Jacobian for the solution $\tilde{\mathbf{w}}^{(k)}$. Therefore, the hyperreduced FOM Jacobian would be represented by:

$$\mathbf{J}^{(k)} \approx \bar{\mathbf{J}}^{(k)} = \sum_{e \in \tilde{\mathcal{E}}} \xi_e \mathbf{L}_e^T \mathbf{J}_e^{(k)} \mathbf{L}_{e^+}. \quad (3.28)$$

Note that the above $\bar{\mathbf{J}}^{(k)}$ is the approximation of the FOM Jacobian, i.e. before projection, and differs from $\tilde{\mathbf{J}}^{(k)}$ which would be the hyperreduced reduced-order Jacobian at iteration k . This is required for the calculation of the test basis in the LSPG framework. The test basis at iteration k would be:

$$\mathbf{W}^{(k)} \approx \tilde{\mathbf{W}}^{(k)} = \bar{\mathbf{J}}^{(k)} \mathbf{V} = \left(\sum_{e \in \tilde{\mathcal{E}}} \xi_e \mathbf{L}_e^T \mathbf{J}_e^{(k)} \mathbf{L}_{e^+} \right) \mathbf{V}. \quad (3.29)$$

The right-hand-side of equation 3.9 is the reduced-order residual which can also be found using the weights and the test basis found above:

$$\hat{\mathbf{R}}^{(k)} \approx \tilde{\mathbf{R}}^{(k)} = \sum_{e \in \tilde{\mathcal{E}}} \xi_e \tilde{\mathbf{W}}^{(k)T} \mathbf{L}_e^T \mathbf{R}_e^{(k)}, \quad (3.30)$$

where $\mathbf{R}_e^{(k)}$ is the contribution of element e to the global, FOM residual for the solution $\tilde{\mathbf{w}}^{(k)}$. Thus the dependence on the high-dimensional model size N has been removed from

the LSPG projection approach and the cost should scale independently of this size. Below in Algorithm 2, the online computation of the reduced-order solution $\tilde{\mathbf{w}}_h$ is summarized.

Algorithm 2 Online Computation of the reduced-order solution by the LSPG projection with Hyperreduced Quantities

Inputs:

POD Basis \mathbf{V} , Reference state \mathbf{w}_{ref} , Convergence tolerance ϵ

Reduced Mesh Element Set $\tilde{\mathcal{E}}$

Associate Set of Weights $\boldsymbol{\xi}_{\mathcal{E}} = \{\xi_e | e \in \tilde{\mathcal{E}}\}$

Output:

Converged approximate solution $\tilde{\mathbf{w}}_h$

Ensure consistency of initial guess: $\tilde{\mathbf{w}}_{\text{proj}}^{(0)} = \mathbf{w}_{\text{ref}} + \mathbf{V}(\mathbf{V}^T(\mathbf{w}^{(0)} - \mathbf{w}_{\text{ref}}))$

Evaluate the hyperreduced Jacobian $\bar{\mathbf{J}}^{(0)}$ using Equation 3.28 and use this to find the test basis $\tilde{\mathbf{W}}^{(0)}$ using Equation 3.29

Evaluate the hyperreduced residual $\tilde{\mathbf{R}}^{(0)}$ using Equation 3.30 and compute the L_2 norm of the initial hyperreduced residual $r = \left\| \tilde{\mathbf{R}}^{(0)} \right\|_2$

while $r > \epsilon$ **do**

 Evaluate the hyperreduced Jacobian $\bar{\mathbf{J}}^{(k)}$ using Equation 3.28 and use this to find the test basis $\tilde{\mathbf{W}}^{(k)}$ using Equation 3.29

 Evaluate the hyperreduced residual $\tilde{\mathbf{R}}^{(k)}$ using Equation 3.30

 Solve $\left[\tilde{\mathbf{W}}^{(k)T} \tilde{\mathbf{W}}^{(k)} \right] \mathbf{p}^{(k)} = -\tilde{\mathbf{R}}^{(k)}$

 Compute the step length $a^{(k)}$ by a line-search procedure

 Update the approximate solution $\tilde{\mathbf{w}}^{(k+1)} = \tilde{\mathbf{w}}^{(k)} + \mathbf{V}(a^{(k)} \mathbf{p}^{(k)})$

 Update $\tilde{\mathbf{R}}^{(k)}$ and $r = \left\| \tilde{\mathbf{R}}^{(k)} \right\|_2$

end while

Chapter 4

ECSW Hyperreduction Verification

In this chapter, we will verify that the ECSW hyperreduction technique has been implemented correctly, before it is integrated it into the adaptive sampling procedure. First, we will use the method presented in [16] to generate a POD basis. With the FOM snapshots used to build the basis, a reduced mesh will be determined for multiple NNLS tolerance values, employing both residual- and Jacobian-based training data. Next, we will test a HROM that utilizes the same POD basis, the identified reduced set of elements, and the corresponding weights, at various parameter locations using Algorithm 2. The test cases include the one-dimensional Burgers' equation with one parameter and the NACA0012 airfoil in transonic flow with two design parameters.

4.1 One Parameter One Dimensional Burgers' Equation

The first test case is the one-dimensional Burgers' equation with a source function. The problem is described by the following differential equation [57]:

$$\mathbf{R}(\mathbf{w}) = \frac{\partial \mathbf{w}(x, t)}{\partial t} + 0.5 \frac{\partial \mathbf{w}^2(x, t)}{\partial x} - \mathbf{S}(\mu),$$

with initial and boundary conditions:

$$\mathbf{w}(x, 0) = 1, \forall x \in [0, 100], \tag{4.1}$$

$$\mathbf{w}(0, t) = 1, \forall t > 0.$$

Only steady-state solutions are considered in this case. The functional to optimize is the integral of the steady-state solution \mathbf{w} over the domain:

$$\mathcal{J}(\mathbf{w}) = \int_0^{100} \mathbf{w} \, dx. \tag{4.2}$$

The spatial domain is discretized using 1024 nodes (i.e. the FOM dimension N is 1024), the FOM solution is approximated using the upwind scheme:

$$\frac{w_j^{n+1} - w_j^n}{\Delta t} = \frac{F(w_{j+1}^n) - F(w_j^n)}{\Delta x} + S(\mu, w_j^n), \tag{4.3}$$

where F is the flux function $F(u) = \frac{u^2}{2}$, and S is the source term. An exponential source term will be used: $S(b, x) = e^{bx}$, where b is the one design parameter that can be varied to values between 0.01 and 0.1. Example solutions from the upwind scheme can be seen in Figure 4.1 at three evenly spaced parameter locations.

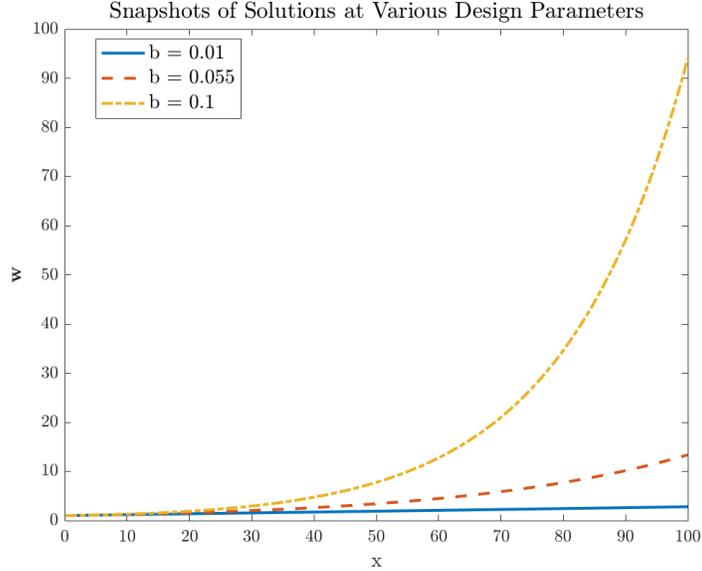


Figure 4.1: Solutions of 1D Burgers’ Equation with Exponential Source Term from Upwind Scheme

The adaptive sampling procedure was initialized with three solution snapshots, shown in Figure 4.1, which are uniformly distributed over the parameter space. After running to an error tolerance of $1E-4$, the procedure solved the FOM at 4 additional parameter locations, resulting in a ROM dimension n of 7. All 7 snapshots were then used as training data for both the residual and Jacobian-based ECSW approaches. Three values of ϵ in the NNLS problem were tested to evaluate the trade-off between computational/storage savings and accuracy in the ECSW method. For the residual-based approach, ϵ values of $1E-4$, $1E-6$ and $1E-8$ were tested. The original intention was to test the same values for the Jacobian-based approach; however, the algorithm used to solve the NNLS problem was only able to achieve a minimum tolerance of $1E-7$. This limitation highlights a potential shortcoming of the NNLS approach, as it does not guarantee convergence to a solution for all tolerance values. Ideally, as the tolerance approaches very low values, the solution vector should converge to a vector of ones. Even in the residual-based approach, there are practical limits on acceptable ϵ values. These limitations may stem from the conditioning of the \mathbf{C} matrix,

which could be affected by factors such as linear dependence among rows in the training data or subtractive cancellation errors. Furthermore, since one of the main objectives of the goal-oriented adaptive sampling procedure is to reduce the number of FOM solutions computed, the amount of training data available is inherently limited, which could further impact the performance of the ECSW method.

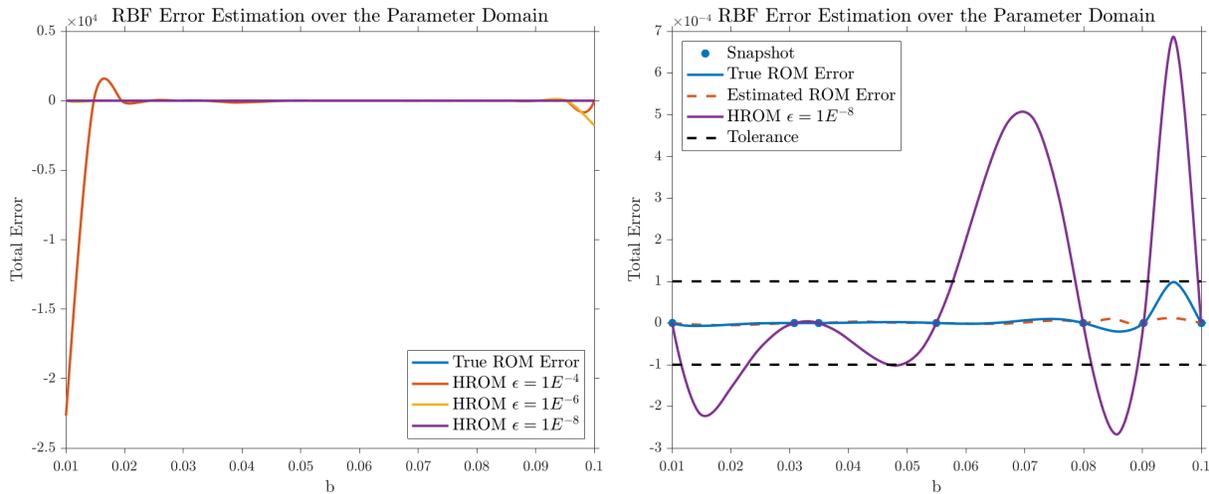


Figure 4.2: Error in the Functional \mathcal{J} over the Parameter Domain for the Final ROM and Three Residual-based ECSW HROM

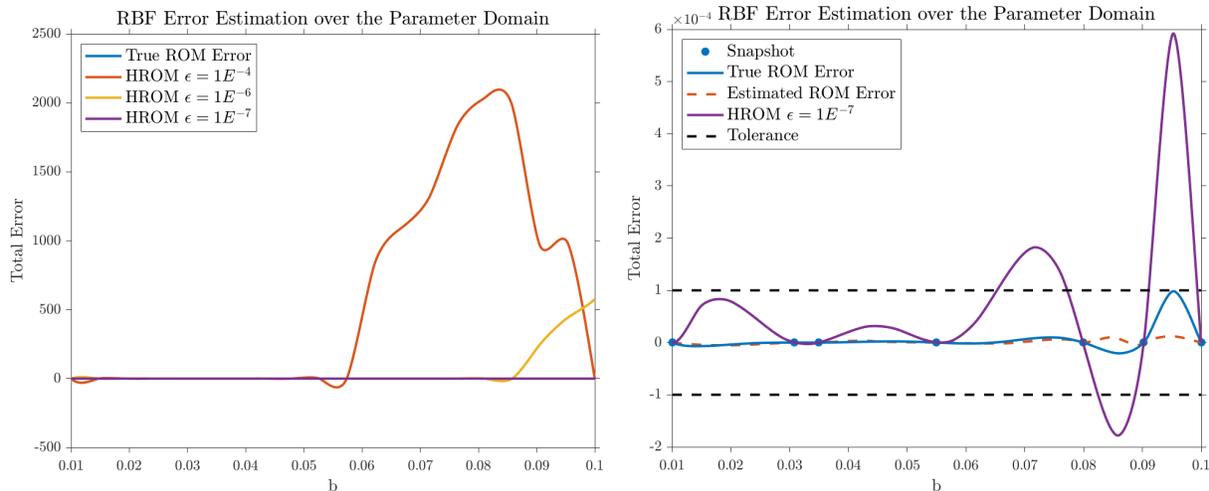


Figure 4.3: Error in the Functional \mathcal{J} over the Parameter Domain for the Final ROM and Three Jacobian-based ECSW HROM

Using the reduced mesh identified from each combination of training data type and NNLS tolerance value, six different hyperreduced ROMs (HROMs) were constructed. The error between the FOM and the HROMs was sampled at 20 evenly spaced points, and radial basis function (RBF) interpolation was employed to estimate the expected online or “true” error distribution across the parameter domain for each model. The same process was applied to the ROM without hyperreduction from the sampling procedure. Figures 4.2 and 4.3 show the functional error results for the HROMs built using residual and Jacobian-based training data, respectively. In both figures, the left image displays the “true” error estimate for the ROM and three HROMs with different ϵ values. The right image focuses on the HROM with the lowest NNLS tolerance value: $\epsilon = 1E - 8$ for the residual-based ECSW and $\epsilon = 1E - 7$ for the Jacobian-based ECSW. These plots also include the estimated error distribution from the final iteration of the sampling procedure, the “true” error for the ROM, and the functional error tolerance bound. The results reveal that for both approaches, higher NNLS tolerance values lead to significant spikes in the error distribution, indicating that the reduced mesh cannot accurately approximate the residual and Jacobian online. When the tolerance is reduced to its lowest value, the performance becomes comparable to the ROM without hyperreduction. Among the tested models, the Jacobian-based HROM with $\epsilon = 1E - 7$ performs best, though it only marginally outperforms the residual-based HROM with $\epsilon = 1E - 8$. At its worst, the Jacobian-based HROM violates the original tolerance bound by no more than an order of magnitude at the ROM points.

The solution vectors from the FOM, ROM, and HROMs at a parameter location of $b = 0.044$ are plotted together, Figure 4.4 displays the residual-based HROMs and Figure 4.5 displays the Jacobian-based HROMs. With the exception of the residual-based HROM with $\epsilon = 1E - 4$, the HROM and ROM solutions are nearly identical to the FOM solution, showing minimal error and appearing visually indistinguishable.

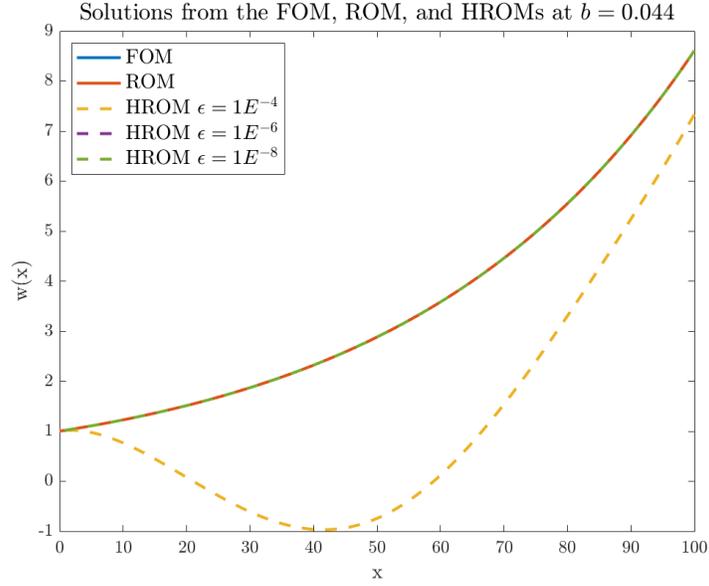


Figure 4.4: Solutions of 1D Burgers' Equation from the FOM and Projection-Based ROM with and without Residual-based Hyperreduction

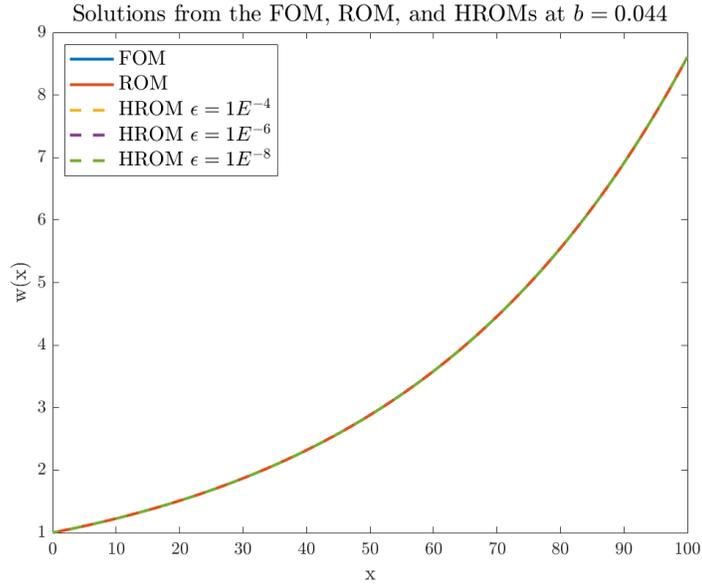


Figure 4.5: Solutions of 1D Burgers' Equation from the FOM and Projection-Based ROM with and without Jacobian-based Hyperreduction

The solution and functional error results at this parameter location, along with the average ROM point functional error for each model, are summarized in Table 4.1. Hyperreduction

has significantly decreased the number of elements required to compute the residual and Jacobian at each iteration, reducing the original 1024 elements to as few as 11 in some cases. The Jacobian-based HROMs outperform the residual-based HROMs in terms of errors introduced in both the solution vector \mathbf{w} and the functional \mathcal{J} , as seen in the third and fourth column of Table 4.1. However, even the best-performing HROMs introduce additional errors relative to the original ROM without hyperreduction, as expected due to the added layer of approximation.

Table 4.1: 1D Burgers’ ROM Results at $b = 0.044$ and Average Absolute ROM Point Error over the Parameter Domain

ROM/HROM	$\ \boldsymbol{\xi}\ _0$	$\ \mathbf{w}_* - \mathbf{w}_{FOM}\ _2$	$\ \mathcal{J}_* - \mathcal{J}_{FOM}\ _2$	Avg. ROM Point Error
ROM	-	$2.0431E - 8$	$-3.7066E - 6$	$8.3270E - 6$
Res. $\epsilon = 1E - 4$	18	$6.061E - 1$	$-2.224E2$	$1.1433E3$
Res. $\epsilon = 1E - 6$	35	$1.2079E - 6$	$4.5142E - 4$	$8.8868E1$
Res. $\epsilon = 1E - 8$	46	$2.1958E - 7$	$-8.0496E - 5$	$1.7082E - 4$
Jac. $\epsilon = 1E - 4$	11	$3.8299E - 6$	$1.4331E - 3$	$5.5215E2$
Jac. $\epsilon = 1E - 6$	25	$1.1953E - 6$	$-4.45416 - 4$	$6.3034E1$
Jac. $\epsilon = 1E - 7$	28	$8.8126E - 8$	$3.1820E - 5$	$8.0756E - 5$

The size of the reduced mesh and the average ROM point error for the HROMs are plotted against the NNLS tolerance in Figure 4.6. From the plot on the left, it is evident that the Jacobian-based training data leads to a smaller reduced mesh for equivalent NNLS tolerance values. This suggests that the Jacobian ECSW approach more efficiently approximates the projected FOM quantities online, requiring only 28 of the total 1024 elements when $\epsilon = 1E - 7$ to maintain the solution and functional error at $b = 0.044$, as well as the average ROM point error, within an order of magnitude of the ROM without hyperreduction. From the plot on the right, Jacobian-based ECSW exhibits lower average ROM point errors for the same or even higher NNLS tolerance values, indicating that it is both more accurate and computationally more efficient than residual-based ECSW.

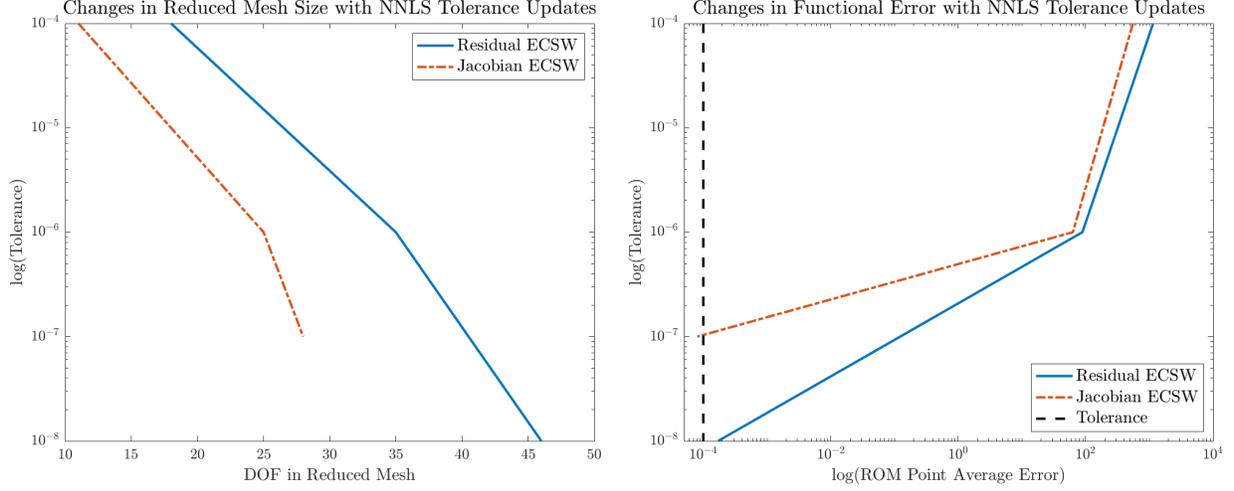


Figure 4.6: Impact of NNLS Tolerance on the Size of the Reduced Mesh and the Accuracy of the HROM for the Residual and Jacobian-based Training Approaches

4.2 Two Parameter NACA0012 Airfoil in Transonic Flow

The second test case is for a NACA0012 airfoil in inviscid flow. The Euler equations governing two-dimensional inviscid flow past an airfoil are:

$$\nabla_x \mathbf{F}_x + \nabla_y \mathbf{F}_y = 0, \quad (4.4)$$

where,

$$\mathbf{F}_x = \begin{bmatrix} \rho v_1 \\ \rho v_1 v_1 + p \\ \rho v_1 v_2 \\ v_1(\rho e + p) \end{bmatrix} \quad \text{and} \quad \mathbf{F}_y = \begin{bmatrix} \rho v_2 \\ \rho v_1 v_2 \\ \rho v_2 v_2 + p \\ v_2(\rho e + p) \end{bmatrix}, \quad (4.5)$$

where ρ is the density, v_1 and v_2 are velocity components, p is the pressure and e the internal energy. The far-field boundary conditions are specified by the Mach number, angle of attack

α , static pressure, and density. The airfoil surface is subject to no-slip boundary conditions and is adiabatic. A mesh with 560 cells and a polynomial order of 0 is used, resulting in a FOM dimension N of 2240 (total degrees of freedom). This test case involves two design parameters: Mach number in the transonic range $[0.5, 0.9]$ and angle of attack α between $[0, 5]^\circ$. The functional of interest \mathcal{J} is the lift coefficient. Figure 4.7 shows the pressure coefficient distribution of an example solution taken at an angle of attack of 2° .

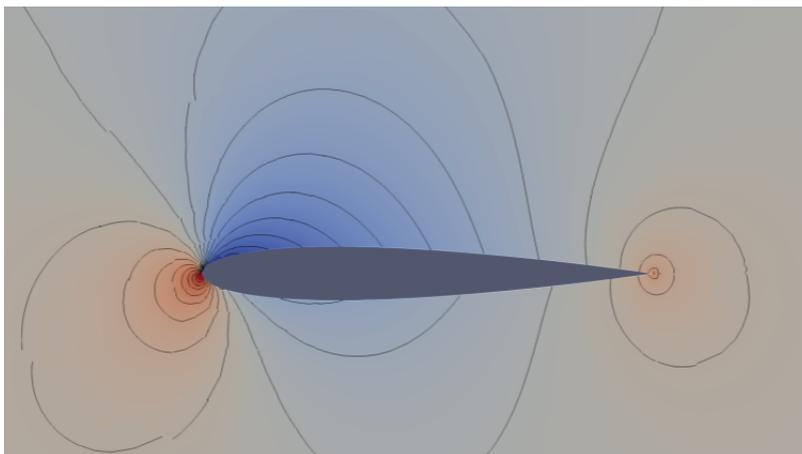


Figure 4.7: Example Solution of NACA0012 Airfoil Test Case

The adaptive sampling is initialized with 9 snapshots arranged in a grid pattern across the parameter space, as seen in the left plot of Figure 4.8. The functional error tolerance for the ROM points is set to $1E-4$. After running the sampling procedure, 27 additional FOM snapshot solutions are computed, bringing the total ROM dimension n to 36. The final RBF error estimate for the ROM, based on the snapshot and ROM locations, is shown in Figure 4.9.

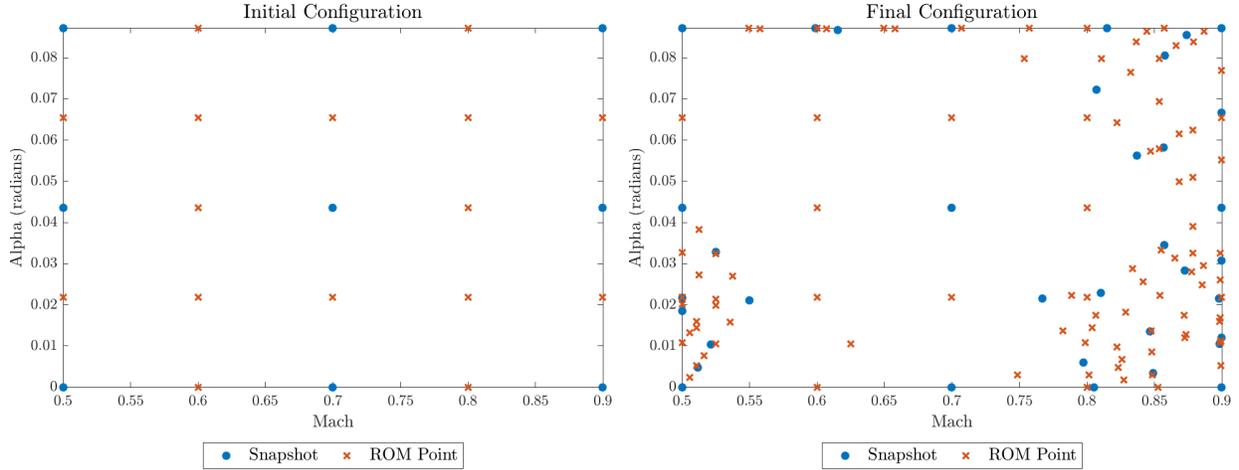


Figure 4.8: Initial and Final Configuration of the Parameter Space for the 2 Parameter NACA0012 Airfoil in Transonic Flow

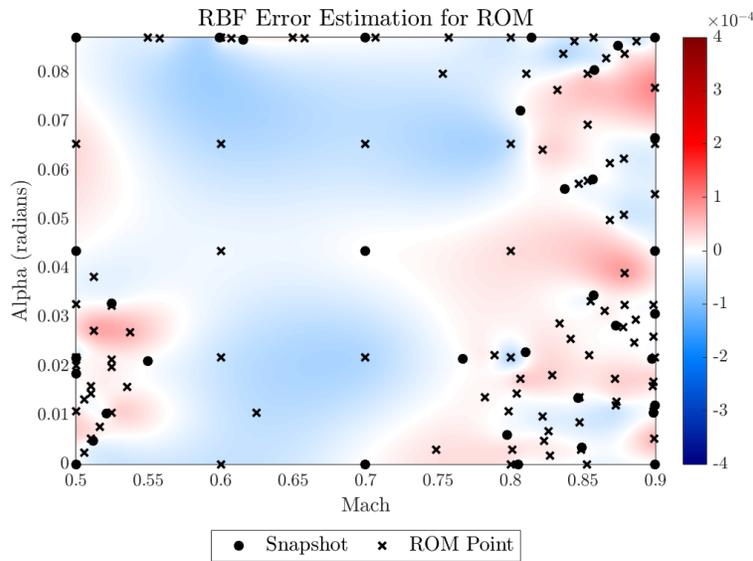


Figure 4.9: Error Estimate Across the Parameter Space for the ROM after Adaptive Sampling

Similar to the previous test case, three ϵ values of $1E - 4$, $1E - 5$, and $1E - 6$ were tested for both the residual and Jacobian-based ECSW approaches. In the Jacobian-based approach, including all snapshots as training data would result in a \mathbf{C} matrix with $N_s n^2$ rows, equating to $36 \times 36^2 = 46,656$ total rows. This proved to cause convergence issues in

the NNLS problem, likely due to poor matrix conditioning and potential linear dependence among the numerous rows in the training data. To address this, only six snapshots located on the boundaries of the parameter space were used during the training phase for both the residual-based and Jacobian-based hyperreduction. Developing more informed methods for selecting training snapshots is left for future work.

The online performance of each HROM with different training data approaches and hyperreduction tolerances can be compared to the ROM by evaluating their errors relative to the FOM at 400 evenly distributed locations across the parameter space. An RBF interpolation is then conducted to approximate the expected online accuracy of the models, referred to as the “true” or “actual” error. Figure 4.10 shows the online error distribution for the ROM. The results indicate that the goal-oriented adaptive sampling procedure was largely effective in predicting the ROM’s behavior and accuracy. The plot on the right highlights regions of the parameter space where the tolerance bound of $1E - 4$ from the adaptive sampling procedure is exceeded. Notably, at low Mach numbers and high angles of attack, higher functional errors are observed, likely due to the limited number of snapshots added in this region. Nevertheless, the ROM performs well across most of the design space, with deviations from the original tolerance bound being less than an order of magnitude.

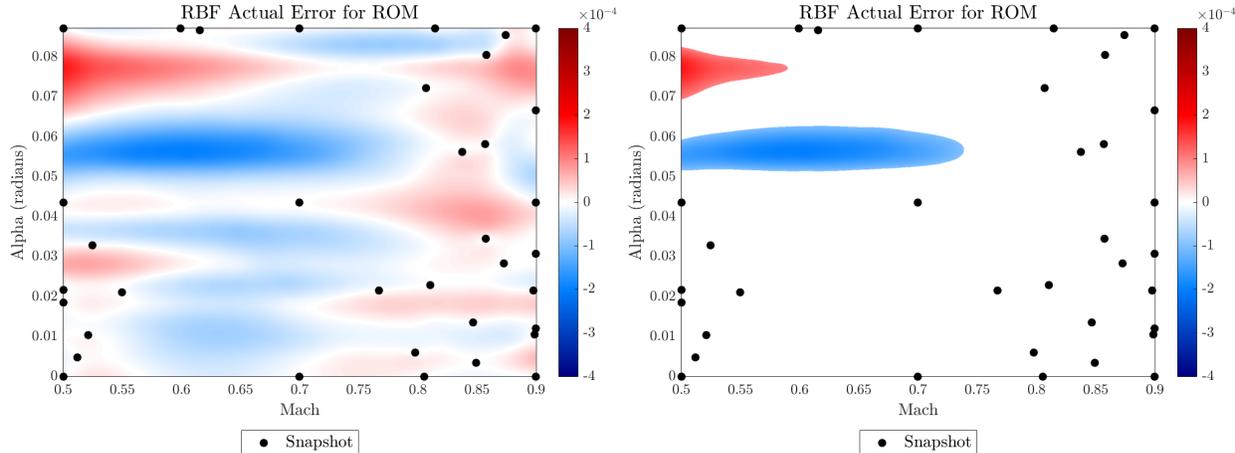


Figure 4.10: True Error for the ROM and Zones of Tolerance Violation

Figures 4.11, 4.12, and 4.13 show the RBF-interpolated error distributions for the HROMs built using residual-based training data with ϵ_{NNLS} tolerances of 1×10^{-4} , 1×10^{-5} , and 1×10^{-6} , respectively. Each figure includes a plot on the right that highlights regions of the parameter domain where the adaptive sampling tolerance of $1E - 4$, used to construct the ROM, is exceeded. Green circles mark the POD snapshots that were employed to train the ECSW hyperreduction. As expected, the hyperreduction introduces additional errors into the HROM solutions, as it adds another layer of approximation. Interestingly, the hyperreduction alters the regions where the tolerance bound is violated; however, as the ϵ_{NNLS} tolerance decreases, the error distribution converges towards the original ROM “true” error depicted in Figure 4.10.

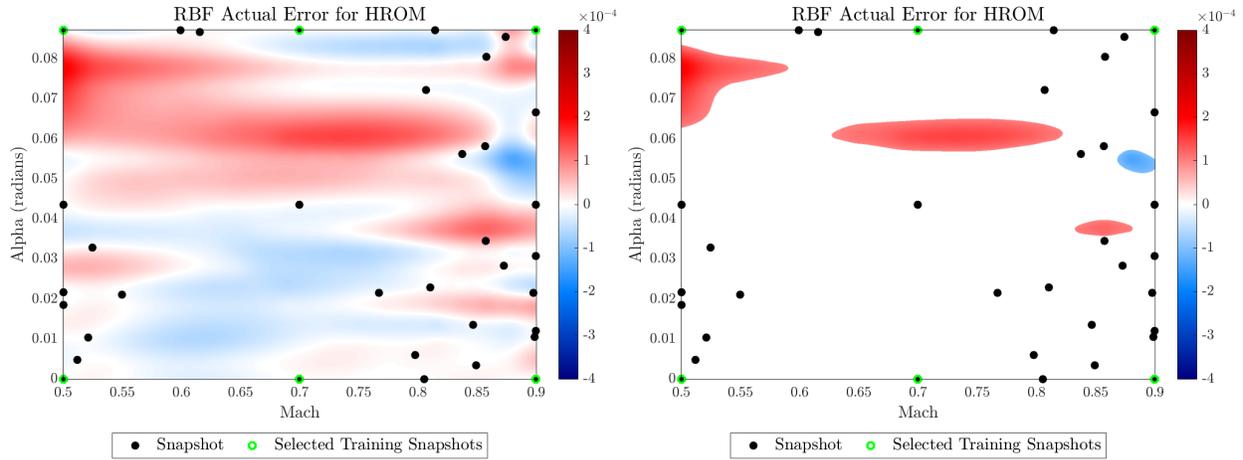


Figure 4.11: True Error for the Residual-based HROM with $\epsilon_{\text{NNLS}} = 1E - 4$ and Zones of Tolerance Violation

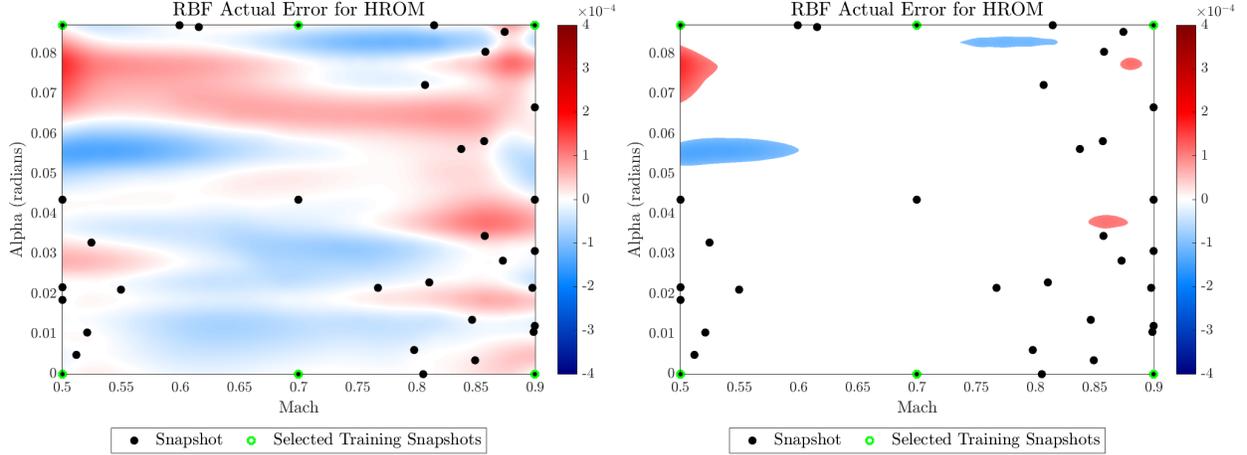


Figure 4.12: True Error for the Residual-based HROM with $\epsilon_{\text{NNLS}} = 1E - 5$ and Zones of Tolerance Violation

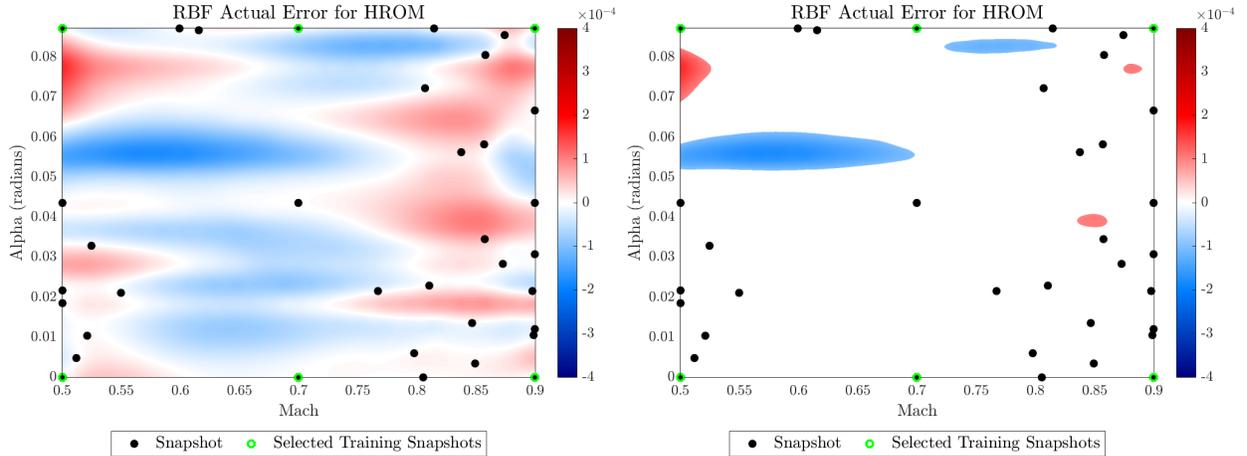


Figure 4.13: True Error for the Residual-based HROM with $\epsilon_{\text{NNLS}} = 1E - 6$ and Zones of Tolerance Violation

Figures 4.14, 4.15, and 4.16 present the RBF-interpolated error distributions for the HROMs constructed using Jacobian-based training data with ϵ_{NNLS} tolerances of 1×10^{-4} , 1×10^{-5} , and 1×10^{-6} , respectively. It is evident that the Jacobian-based ECSW introduces more error than the residual-based ECSW across all ϵ_{NNLS} tolerances. The largest errors occur in regions with fewer snapshots, suggesting that incorporating hyperreduction into the adaptive sampling procedure could help mitigate this issue. Despite these challenges,

the errors remain within an order of magnitude of the functional error tolerance used to construct the ROM.

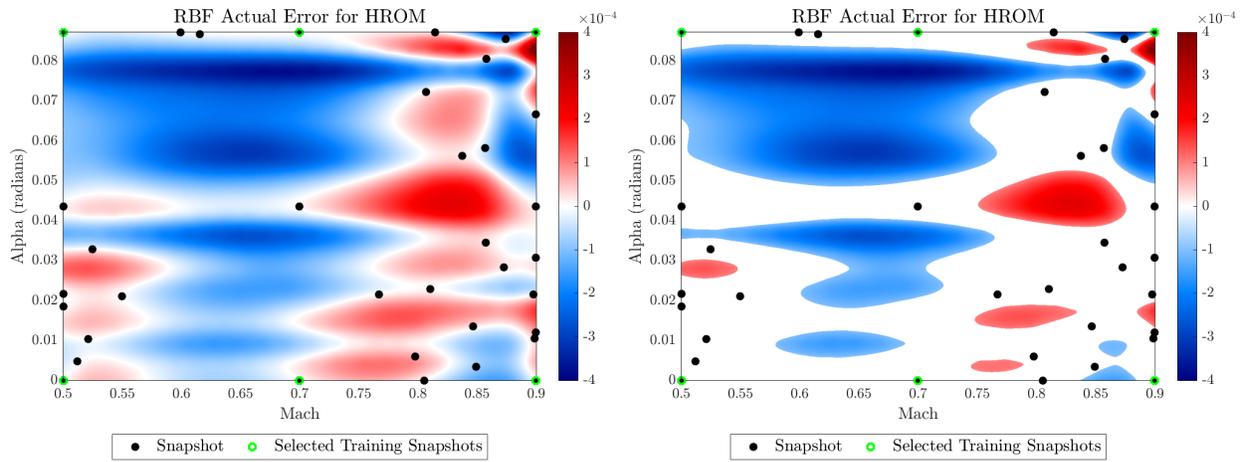


Figure 4.14: True Error for the Jacobian-based HROM with $\epsilon_{\text{NNLS}} = 1E - 4$ and Zones of Tolerance Violation

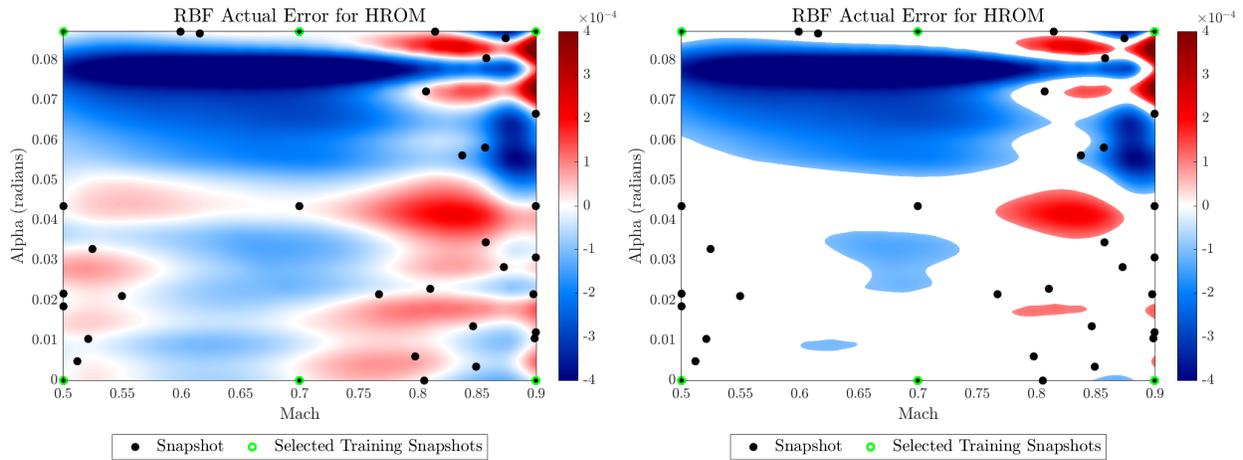


Figure 4.15: True Error for the Jacobian-based HROM with $\epsilon_{\text{NNLS}} = 1E - 5$ and Zones of Tolerance Violation

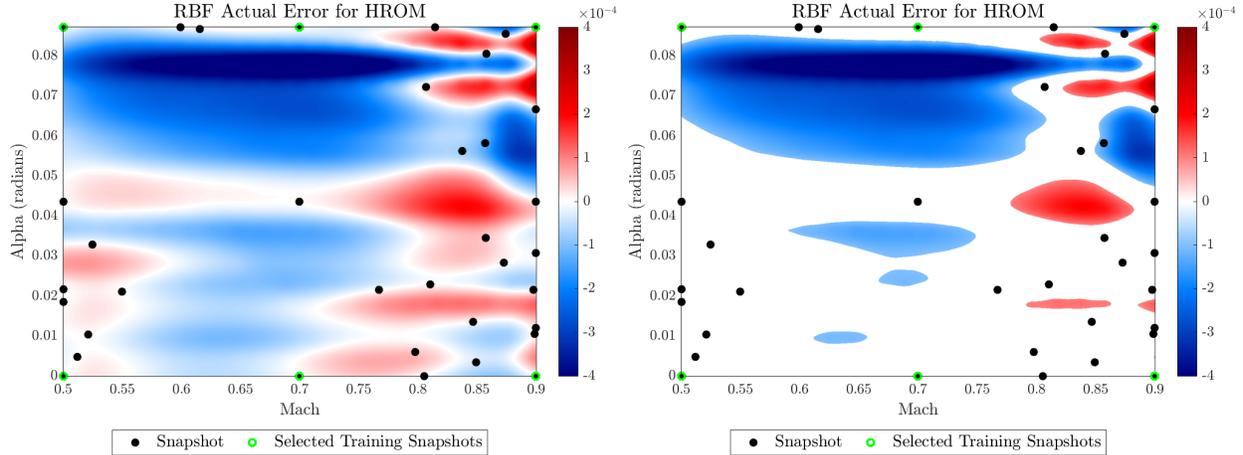


Figure 4.16: True Error for the Jacobian-based HROM with $\epsilon_{\text{NNLS}} = 1E - 6$ and Zones of Tolerance Violation

Figure 4.17 illustrates the impact of the NNLS tolerance on the size of the reduced mesh and the average functional error at the 400 test points used to evaluate the HROMs. The plot on the left shows that, as observed in the previous test case, Jacobian-based training data produces smaller reduced mesh sizes. In this test case, the plot on the right indicates that the different NNLS tolerance values tested had little effect on the average ROM point error. While the Jacobian-based ECSW introduced higher functional errors than the residual-based ECSW, these additional errors could likely be mitigated by incorporating hyperreduction into the adaptive sampling procedure. Therefore, if the primary goal is to maximize the efficiency of hyperreduction—by achieving the smallest possible reduced mesh—the Jacobian-based ECSW would be preferable, as the additional error can be addressed during the sampling procedure.

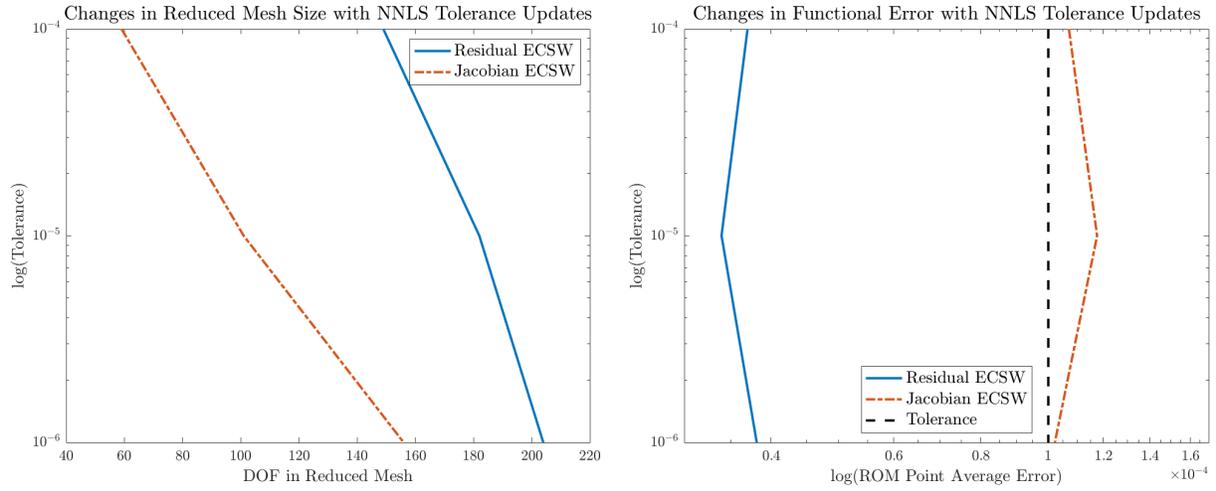


Figure 4.17: Impact of NNLS Tolerance on the Size of the Reduced Mesh and the Accuracy of the HROM for the Residual and Jacobian-based Training Approaches

Chapter 5

Adaptive Sampling Procedure

This chapter outlines the modifications required to integrate hyperreduction into the adaptive sampling procedure developed in [16]. First, the approximation of the residual and Jacobian in the Newton iterations, used to compute Reduced Order Model (ROM) solutions, must be updated. Additionally, the dual-weighted error indicators which approximate the error at ROM points require adjustments to account for the additional approximation introduced by the hyperreduction. Finally, a method for calculating the work units associated with the various sampling procedures will be presented.

5.1 Overview of the Adaptive Sampling Procedure

In this work, we focus on parameterized problems. For this class of problems, the ROM is built by sampling the FOM at various parameter combinations $\boldsymbol{\mu}$ in the parameter domain \mathcal{D} . It is then used to predict the solution at new parameter locations where the “true” solution is unknown. Without a reliable quantification of the error in the ROM, little value can be placed in these predictions [16]. In CFD problems, the focus is not only on the accuracy of the solution approximation but in particular on the error introduced in the functional \mathcal{J}

by the ROM. For example, in aerodynamic shape optimization, the iterative design of the aircraft or airfoil is guided by an output of interest, such as lift or drag, therefore confidence in the functional value is additionally important.

Building a ROM for parameterized problems involves two key challenges: identifying an *a posteriori* functional error estimate and determining optimal snapshot placement. The accuracy of the ROM depends heavily on the quality of the training data used to construct the reduced-order basis. Sampling too few points or placing them poorly within the parameter domain can lead to an inaccurate model. Conversely, oversampling the FOM incurs unnecessary computational costs. Therefore, it is essential to develop a method for selecting parameter values judiciously to collect snapshots efficiently.

In [16], a novel adaptive sampling procedure was developed with two major objectives. The first is to minimize the number of FOM solutions computed through efficient snapshot selection. The second is, through the use of an *a posteriori* error estimate based on dual-weighted residual error indicators, ensure that a prescribed output error tolerance is estimated to be satisfied across the entire parameter domain \mathcal{D} .

The following is a brief summary of the steps taken in the adaptive sampling procedure:

- An initial set of snapshots is evenly distributed across the parameter space, along with a set of ROM points placed between these snapshots. ROM points are used to probe the error between the ROM and FOM. Two dual-weighted residual errors are tracked at each ROM point.
- At each adaptive cycle, a radial basis function (RBF) interpolation is used to model the error across the parameter space. The next snapshot is placed at the extremum of the RBF. The error estimates at the ROM points are updated, and new ROM points are added near the newly selected snapshot.

- This process is repeated until the specified error tolerance is met at all ROM points across the parameter domain.

Further details on these steps, along with an algorithm describing the procedure, can be found in [16].

5.2 Dual-Weighted Residual Errors

Two distinct dual-weighted residual (DWR) errors are used to approximate the error at a ROM point: ϵ_f , which measures the error between the FOM and the ROM, and ϵ_r , which quantifies the error between a "coarse" ROM and a "fine" ROM. The coarseness of a ROM refers to the dimension of the basis and which iteration of the sampling cycle a ROM is from. Meaning, that a "coarse" ROM will have a coarse basis V_H which will have fewer columns than the fine basis V_h . We then say that for a given parameter location $\boldsymbol{\mu}$ with full-order solution \mathbf{w} , there are two ROM solutions, one computed on the fine basis $\mathbf{w}_{\text{ref}_h} + \mathbf{V}_h \hat{\mathbf{w}}_h$ and one computed on the coarse basis $\mathbf{w}_{\text{ref}_H} + \mathbf{V}_H \hat{\mathbf{w}}_H$. Therefore, the total error ϵ in the functional at this ROM is approximately [16]:

$$\begin{aligned}
\epsilon &= \mathcal{J}(\mathbf{w}_{\text{ref}_h} + \mathbf{V}_h \hat{\mathbf{w}}_h) - \mathcal{J}(\mathbf{w}) = (\mathcal{J}(\mathbf{w}_{\text{ref}_H} + \mathbf{V}_H \hat{\mathbf{w}}_H) - \mathcal{J}(\mathbf{w})) \\
&\quad - (\mathcal{J}(\mathbf{w}_{\text{ref}_H} + \mathbf{V}_H \hat{\mathbf{w}}_H) - \mathcal{J}(\mathbf{w}_{\text{ref}_h} + \mathbf{V}_h \hat{\mathbf{w}}_h)) \quad (5.1) \\
&\approx \epsilon_f - \epsilon_r.
\end{aligned}$$

ϵ_r will be the DWR error impacted by hyperreduction as it requires the evaluation of the reduced residual and Jacobian, therefore we will re-derive it in the context of a HROM.

5.2.1 Error Estimate Between Coarse and Fine Hyperreduced Reduced-Order Models

The error estimate between a coarse and fine ROM derived in [16] will be extended here to account for a hyperreduced ROM. As before, let the coarse ROM solution be denoted by $\mathbf{w}_{\text{ref}_H} + \mathbf{V}_H \hat{\mathbf{w}}_H$, where the subscript H indicates a coarse-space representation. Similarly, let the fine ROM solution be $\mathbf{w}_{\text{ref}_h} + \mathbf{V}_h \hat{\mathbf{w}}_h$, where the subscript h represents a fine-space representation. For brevity, these solutions will also be referred to as $\tilde{\mathbf{w}}_H$ and $\tilde{\mathbf{w}}_h$, respectively. Then the solutions must satisfy the following:

$$\begin{aligned}\tilde{\mathbf{R}}_h(\tilde{\mathbf{w}}_h) &= 0, \\ \tilde{\mathbf{R}}_H(\tilde{\mathbf{w}}_H) &= 0,\end{aligned}\tag{5.2}$$

where $\tilde{\mathbf{R}}_*$ represents the hyperreduced reduced-order residual from each HROM. Using a first-order Taylor series expansion of the fine-dimension hyperreduced reduced-order residual about the coarse reduced-order solution, we find:

$$\tilde{\mathbf{R}}_h(\tilde{\mathbf{w}}_h) = 0 \approx \tilde{\mathbf{R}}_h(\tilde{\mathbf{w}}_H) + \left[\frac{\partial \tilde{\mathbf{R}}_h}{\partial \hat{\mathbf{w}}_h} \Big|_{\tilde{\mathbf{w}}_H} \right] (\tilde{\mathbf{w}}_h - \tilde{\mathbf{w}}_H).\tag{5.3}$$

Note that the derivative is taken with respect to the reduced-order solution representation $\hat{\mathbf{w}}_h$ rather than $\tilde{\mathbf{w}}_h$. This choice simplifies the analysis and is justified because the errors arising from the derivative of the residual with respect to the reference state $\mathbf{w}_{\text{ref}_h}$ or the POD basis \mathbf{V}_h are primarily influenced by the accuracy of the DG approach not the ROM.

Conducting the chain rule on this derivative, we find:

$$\begin{aligned} \left. \frac{\partial \tilde{\mathbf{R}}_h}{\partial \hat{\mathbf{w}}_h} \right|_{\tilde{\mathbf{w}}_H} &= \left. \frac{\partial \tilde{\mathbf{R}}_h}{\partial \tilde{\mathbf{w}}_h} \right|_{\tilde{\mathbf{w}}_H} \frac{\partial \tilde{\mathbf{w}}_h}{\partial \hat{\mathbf{w}}_h} \\ &= \left. \frac{\partial \tilde{\mathbf{R}}_h}{\partial \tilde{\mathbf{w}}_h} \right|_{\tilde{\mathbf{w}}_H} \mathbf{V}_h. \end{aligned} \quad (5.4)$$

Using the definition of the hyperreduced residual in Equation 3.30, we can then write the remaining derivative as follows:

$$\begin{aligned} \frac{\partial \tilde{\mathbf{R}}_h}{\partial \tilde{\mathbf{w}}_h} &= \frac{\partial}{\partial \tilde{\mathbf{w}}_h} \left(\sum_{e \in \tilde{\mathcal{E}}} \xi_e \tilde{\mathbf{W}}_h^T \mathbf{L}_e^T \mathbf{R}_e \right) \\ &= \sum_{e \in \tilde{\mathcal{E}}} \xi_e \frac{\partial \tilde{\mathbf{W}}_h^T}{\partial \tilde{\mathbf{w}}_h} \mathbf{L}_e^T \mathbf{R}_e + \sum_{e \in \tilde{\mathcal{E}}} \xi_e \tilde{\mathbf{W}}_h^T \frac{\partial \tilde{\mathbf{w}}_h}{\partial \tilde{\mathbf{w}}_h} \mathbf{L}_e^T \frac{\partial \mathbf{R}_e}{\partial \tilde{\mathbf{w}}_h}. \end{aligned} \quad (5.5)$$

Given that a LSPG framework was used, second-order sensitivities will arise when taking the derivative of the test basis with respect to the solution, as it contains the Jacobian as well. [58] notes that this can be safely neglected with a minor loss in accuracy. Therefore, the first term on the right-hand side of Equation 5.5 will be ignored.

In a similar fashion, the first-order Taylor series expansions of the functional is:

$$\mathcal{J}(\tilde{\mathbf{w}}_h) \approx \mathcal{J}(\tilde{\mathbf{w}}_H) + \left[\left. \frac{\partial \mathcal{J}}{\partial \tilde{\mathbf{w}}_h} \right|_{\tilde{\mathbf{w}}_H} \right] (\tilde{\mathbf{w}}_h - \tilde{\mathbf{w}}_H). \quad (5.6)$$

We then define $\tilde{\psi}_h$ as the solution of the hyperreduced reduced-order dual problem:

$$\left[\left. \frac{\partial \tilde{\mathbf{R}}_h}{\partial \hat{\mathbf{w}}_h} \right|_{\tilde{\mathbf{w}}_H} \right]^T \tilde{\psi}_h = - \left[\left. \frac{\partial \mathcal{J}}{\partial \hat{\mathbf{w}}_h} \right|_{\tilde{\mathbf{w}}_H} \right]^T. \quad (5.7)$$

Using Equations 5.4 and 5.5, and applying the chain rule in a similar manner to the derivative of the functional, we find:

$$\left[\left\{ \sum_{e \in \tilde{\mathcal{E}}} \xi_e \mathbf{W}_h^T \mathbf{L}_e^T \frac{\partial \mathbf{R}_e}{\partial \tilde{\mathbf{w}}_h} \Big|_{\tilde{\mathbf{w}}_H} \right\} \mathbf{V}_h \right]^T \tilde{\psi}_h = - \left[\frac{\partial \mathcal{J}}{\partial \tilde{\mathbf{w}}_h} \Big|_{\tilde{\mathbf{w}}_H} \mathbf{V}_h \right]^T. \quad (5.8)$$

Note that the set of weights and the test basis will be from the fine ROM, meaning they are computed in the most recent iteration. Using the hyperreduced reduced-order adjoint in Equation 5.8 and the Taylor series expansion in Equation 5.3, an error metric between the fine and coarse space can be defined:

$$\begin{aligned} \epsilon_r &= \mathcal{J}(\tilde{\mathbf{w}}_H) - \mathcal{J}(\tilde{\mathbf{w}}_h) \\ &\approx - \left[\frac{\partial \mathcal{J}}{\partial \tilde{\mathbf{w}}_h} \Big|_{\tilde{\mathbf{w}}_H} \right] (\tilde{\mathbf{w}}_h - \tilde{\mathbf{w}}_H) \\ &\approx - \left[\frac{\partial \mathcal{J}}{\partial \tilde{\mathbf{w}}_h} \Big|_{\tilde{\mathbf{w}}_H} \right] \left[\frac{\partial \tilde{\mathbf{R}}_h}{\partial \tilde{\mathbf{w}}_h} \Big|_{\tilde{\mathbf{w}}_H} \right]^{-1} (\tilde{\mathbf{R}}_h(\tilde{\mathbf{w}}_h) - \tilde{\mathbf{R}}_h(\tilde{\mathbf{w}}_H)) \\ &\approx - \tilde{\psi}_h^T \tilde{\mathbf{R}}_h(\tilde{\mathbf{w}}_H). \end{aligned} \quad (5.9)$$

5.3 Updated Goal-Oriented Adaptive Sampling

The adaptive sampling procedure summarized in Section 5.5 and Algorithm 6 of [16] is updated here with modifications to accommodate the current work. While not all components are discussed in detail, we refer the reader to [16] for comprehensive descriptions. The major changes come from the need to train the ECSW weights at every iteration before solving for the ROM probing points, as well as the introduction of the updated dual-weighted residual (DWR) error indicator for measuring the error between coarse and fine ROMs. For clarity, algorithms from [16] are referenced as D-* (where * corresponds to the original numbering),

while algorithms introduced in this work are referred to by their sequence number in this document.

Algorithm 3 Goal-Oriented Adaptive Sampling

Inputs:

Error tolerance ϵ

Initial snapshot parameter locations $\mathcal{S}^\mu = \{\boldsymbol{\mu}_1^s, \boldsymbol{\mu}_2^s, \dots, \boldsymbol{\mu}_k^s\}$

NNLS tolerance ϵ_{NNLS}

Output:

POD basis \mathbf{V}

Maximum error ϵ_{max}

Compute initial set of k snapshot solutions $\mathcal{S}^\mathbf{w}$ at \mathcal{S}^μ

Compute initial POD basis \mathbf{V}_H from $\mathcal{S}^\mathbf{w}$ using Algorithm D-2

Determine initial set of ROM probing parameter locations $\mathcal{P}^\mu = \{\boldsymbol{\mu}_1^p, \boldsymbol{\mu}_2^p, \dots, \boldsymbol{\mu}_k^p\}$ using Algorithm D-3

Solve for the reduced mesh set and weights using the ECSW approach in Algorithm 1 or the Jacobian-based equivalent

for $\boldsymbol{\mu}^p \in \mathcal{P}^\mu$ **do**

 Compute the HROM solution $\tilde{\mathbf{w}}(\boldsymbol{\mu}^p)$ with POD basis \mathbf{V}_H using Algorithm 2

 Compute error between FOM and coarse ROM, $\epsilon_f(\boldsymbol{\mu}^p) = -\boldsymbol{\psi}^T \mathbf{R}(\tilde{\mathbf{w}})$

 Update total error, $\epsilon(\boldsymbol{\mu}^p) \leftarrow \epsilon_f(\boldsymbol{\mu}^p)$

end for

Compute RBF and obtain maximum error ϵ_{max} at $\boldsymbol{\mu}_{\text{max}}$ using Algorithm D-4

while $\epsilon_{\text{max}} > \epsilon$ **do**

 Compute new snapshot solution $\mathbf{w}(\boldsymbol{\mu}_{\text{max}})$, update $\mathcal{S}^\mathbf{w} = \mathcal{S}^\mathbf{w} \cup \{\mathbf{w}(\boldsymbol{\mu}_{\text{max}})\}$

 Compute \mathbf{V}_h from $\mathcal{S}^\mathbf{w}$ using Algorithm D-2

 Solve for the reduced mesh set and weights using the ECSW approach in Algorithm 1 or the Jacobian-based equivalent

for $\boldsymbol{\mu}^p \in \mathcal{P}^\mu$ **do**

 Compute error between coarse ROM and fine ROM, $\epsilon_r(\boldsymbol{\mu}^p) = -\tilde{\boldsymbol{\psi}}_h^T \tilde{\mathbf{R}}_h(\tilde{\mathbf{w}}_H)$

 Update total error, $\epsilon(\boldsymbol{\mu}^p) \leftarrow \epsilon_f(\boldsymbol{\mu}^p) - \epsilon_r(\boldsymbol{\mu}^p)$

end for

 Recompute solution at select ROM probing points in \mathcal{P}^μ using Algorithm D-5

 Add new ROM probing points \mathcal{P}_*^μ at midpoint of $\boldsymbol{\mu}_{\text{max}}$ and $n_p + 1$ nearest neighbours in \mathcal{S}^μ

for $\boldsymbol{\mu}_*^p \in \mathcal{P}_*^\mu$ **do**

 Update $\mathcal{P}^\mu = \mathcal{P}^\mu \cup \{\boldsymbol{\mu}_*^p\}$

 Let $\mathbf{V}_H = \mathbf{V}_h$

 Compute ROM solution $\tilde{\mathbf{w}}(\boldsymbol{\mu}_*^p)$ with POD basis \mathbf{V}_H using Algorithm 2

 Compute error between FOM and coarse ROM, $\epsilon_f(\boldsymbol{\mu}_*^p) = -\boldsymbol{\psi}^T \mathbf{R}(\tilde{\mathbf{w}})$

 Update total error, $\epsilon(\boldsymbol{\mu}_*^p) \leftarrow \epsilon_f(\boldsymbol{\mu}_*^p)$

end for

 Compute RBF and obtain maximum error ϵ_{max} at $\boldsymbol{\mu}_{\text{max}}$ using Algorithm D-4

end while

5.4 Analysis of Computational Cost Savings through Work Units

Currently, due to the implementation of the DG method in PHiLiP [38], it is not possible to compute the residual or Jacobian on a specific mesh element. As a result, a pseudo-implementation of the ECSW method is employed. In this approach, the residual and Jacobian are evaluated over the entire mesh, but only the elements in the reduced mesh are utilized in the HROMs. This means the CPU time cannot be used to compare the efficiency of the models. Instead, an approximation of work units, based on matrix dimensions and sparsity, is used to compare the cost of the different sampling procedures.

The number of work units at each cycle comes from two major contributions; the first is from the cost of solving for the reduced-order solutions at the new ROM points and the second is from re-evaluating the DWR error indicator between the coarse and fine ROM at all the previously added ROM points. The cost of solving for the FOM snapshots is not considered as it is not impacted by the hyperreduction. The cost of computing the FOM snapshots is excluded from these considerations as it is not impacted by the hyperreduction, and thus the cost incurred will be the same for any sampling procedure.

For ROM point solutions, the number of total non-linear iterations in a sampling iteration is multiplied by the cost of solving the linear system in Equation 3.9. Work units are estimated under the following assumptions: evaluating an entry in the residual or Jacobian costs one work unit; each floating-point operation (FLOP) in a matrix multiplication counts as one work unit; and the cost of solving the linear system is approximated using the worst-case operation count for the generalized minimal residual method (GMRES), which is $\mathcal{O}(n^3)$.

In the original adaptive sampling procedure which builds the ROM with no hyperreduction, the total number of work units to solve one non-linear iteration is given by:

$$\begin{aligned}
W_{\text{non lin}} = & (\text{units to evaluate the residual}) + \\
& (\text{units to evaluate the Jacobian}) + \\
& (\text{units to assemble LHS matrix and RHS vector in Equation 3.9}) + \quad (5.10) \\
& (\text{units to solve for solution update}) \\
= & (N) + (N^2) + [(2Nn_i + n_i^2 + N + n_i)(2N - 1)] + (n_i^3).
\end{aligned}$$

where N is the dimension of the FOM and n_i is the number of bases in the POD or reduced-order subspace dimension at adaptive sampling cycle i . In the sampling procedures where the hyperreduction is used to evaluate the residual and Jacobian to solve Equation 3.9 at the ROM points, each nonlinear iteration has work units determined by:

$$\begin{aligned}
W_{\text{non lin}} = & (\text{units to evaluate the hyperreduced residual}) + \\
& (\text{units to evaluate the hyperreduced test basis}) + \\
& (\text{units to assemble LHS matrix and RHS vector in Equation 3.9}) + \quad (5.11) \\
& (\text{units to solve for solution update}) \\
= & [n_{e_i}(d_e + 2n_i d_e + n_i)] + [2n_{e_i} d_e d_e^+ + 2d_e d_e^+ n_{e_i} n_i] + [n_i^2(2N - 1)] + (n_i^3).
\end{aligned}$$

where n_{e_i} the number of elements in the reduced mesh at adaptive sampling cycle i . Details of how both Equations 5.10 and 5.11 are derived are included in Appendix A.

The other contribution is from the evaluation of the second DWR error indicator ϵ_r at all of the ROM points from the previous sampling cycles. Similar to the previous case, the work units for this evaluation can be broken down into a couple of components. For the original adaptive sampling procedure which builds the ROM with no hyperreduction, the work units

required to solve for the error indicator are:

$$\begin{aligned}
W_{\text{DWR error}} = & (\text{units to evaluate the residual}) + \\
& (\text{units to evaluate the Jacobian}) + \\
& (\text{units to evaluate the derivative of the functional w.r.t. } \tilde{\mathbf{w}}) + \\
& (\text{units to assemble the adjoint problem}) + \\
& (\text{units to solve for the adjoint}) + \\
& (\text{units to solve } \epsilon_r) \\
= & (N) + (N^2) + (N) + [(2Nn_i + n_i^2 + n_i)(2N - 1)] + \\
& (n_i^3) + [(N + n_i)(2N - 1) + (2n_i - 1)].
\end{aligned} \tag{5.12}$$

In the sampling procedures where the hyperreduction is applied in the DWR error indicator, such as in Equation 5.8 and 5.9, the work unit required to solve for the error indicator are:

$$\begin{aligned}
W_{\text{DWR error}} = & (\text{units to evaluate the hyperreduced residual}) + \\
& (\text{units to evaluate the hyperreduced test basis}) + \\
& (\text{units to evaluate the derivative of the functional w.r.t. } \tilde{\mathbf{w}}) + \\
& (\text{units to assemble the adjoint problem}) + \\
& (\text{units to solve for the adjoint}) + \\
& (\text{units to solve } \epsilon_r) \\
= & [n_{e_i}(d_e + 2n_i d_e + n_i)] + [n_{e_i} d_e d_e^+ + 2d_e d_e^+ n_{e_i} n_i] + (N) \\
& + [n_{e_i}(d_e n_i + n_i) + (n_i^2)(2N - 1) + n_i(2N - 1)] + (n_i^3) + (2n_i - 1).
\end{aligned} \tag{5.13}$$

Details of how both Equations 5.12 and 5.13 are derived are included in Appendix A.

Once the sampling procedure is completed, we can gather the number of nonlinear iterations at each cycle. We will refer to the total number sampling of cycles as c . This, along with the knowledge of the dimension of the reduced-order subspace n_i and the number of previously added ROM points at each iteration, can be used to approximate the work units for a given adaptive sampling procedure. The algorithms for computing the work units for the three approaches used in the results section are included below. Algorithm 4 corresponds to the original sampling procedure, which does not include hyperreduction. Algorithm 5 represents the sampling procedure that incorporates hyperreduction in the ROM point solution evaluations but excludes the updated DWR error. Algorithm 6 represents the sampling procedure that includes both hyperreduction in the ROM point solution evaluations and the hyperreduced DWR error.

Algorithm 4 Evaluating Work Units for ROM

Inputs:

Number of sampling cycles c

Total number of nonlinear iterations in each sampling cycle ($\text{nnl} = \mathcal{N}_1, \mathcal{N}_2, \dots, \mathcal{N}_c$)

Number of design parameters n_p

Outputs:

Work Units for each sampling cycle $W_{\text{tot}}(i)$ for $i = 1, 2, \dots, c$

for $i = 1, 2, \dots, c$ **do**

 Evaluate $W_{\text{non lin}}$ using Equation 5.10

 Solve for the work units from ROM points solutions : $W_{\text{all ROM}} = (\mathcal{N}_i) * W_{\text{non lin}}$

 Evaluate $W_{\text{DWR error}}$ using 5.12

 Solve for the work units needed to find the DWR error at all previous ROM points:

$$W_{\text{all DWR}} = (n_p + 1)(i - 1) * W_{\text{DWR error}}$$

$$W_{\text{tot}}(i) = W_{\text{all ROM}} + W_{\text{all DWR}}$$

end for

Algorithm 5 Evaluating Work Units for HROM without Hyper-DWR

Inputs:Number of sampling cycles c Total number of nonlinear iterations in each sampling cycle ($\text{nnl} = \mathcal{N}_1, \mathcal{N}_2, \dots, \mathcal{N}_c$)Number of design parameters n_p **Outputs:**Work Units for each sampling cycle $W_{\text{tot}}(i)$ for $i = 1, 2, \dots, c$

for $i = 1, 2, \dots, c$ **do**Evaluate $W_{\text{non lin}}$ using Equation 5.11Solve for the work units from ROM points solutions : $W_{\text{all ROM}} = (\mathcal{N}_i) * W_{\text{non lin}}$ Evaluate $W_{\text{DWR error}}$ using 5.12

Solve for the work units needed to find the DWR error at all previous ROM points:

$$W_{\text{all DWR}} = (n_p + 1)(i - 1) * W_{\text{DWR error}}$$

$$W_{\text{tot}}(i) = W_{\text{all ROM}} + W_{\text{all DWR}}$$

end for

Algorithm 6 Evaluating Work Units for HROM with Hyper-DWR

Inputs:Number of sampling cycles c Total number of nonlinear iterations in each sampling cycle ($\text{nnl} = \mathcal{N}_1, \mathcal{N}_2, \dots, \mathcal{N}_c$)Number of design parameters n_p **Outputs:**Work Units for each sampling cycle $W_{\text{tot}}(i)$ for $i = 1, 2, \dots, c$

for $i = 1, 2, \dots, c$ **do**Evaluate $W_{\text{non lin}}$ using Equation 5.11Solve for the work units from ROM points solutions : $W_{\text{all ROM}} = (\mathcal{N}_i) * W_{\text{non lin}}$ Evaluate $W_{\text{DWR error}}$ using 5.13

Solve for the work units needed to find the DWR error at all previous ROM points:

$$W_{\text{all DWR}} = (n_p + 1)(i - 1) * W_{\text{DWR error}}$$

$$W_{\text{tot}}(i) = W_{\text{all ROM}} + W_{\text{all DWR}}$$

end for

Chapter 6

Results

This chapter presents results from the updated adaptive sampling framework. As noted in Chapter 3, [40] observed that in steady-state problems, residual values can often be very close to zero which cause issues when using it as training data for the ECSW hyperreduction approach. Furthermore, based on the findings in Chapter 4, Jacobian-based training data has been shown to produce smaller reduced mesh sizes, resulting in more computationally efficient approximations of the residual and Jacobian. For these reasons, all HROMs constructed in this section will exclusively use Jacobian-based training data with an NNLS tolerance of 1×10^{-6} . Various flow conditions and design parameter combinations for the NACA0012 airfoil will make up the different test cases. HROMs will be compared to ROMs built with the original sampling procedure presented in [16]. To evaluate the effectiveness of the updated DWR error indicator in capturing the additional approximation errors introduced by the ECSW hyperreduction approach, two types of HROMs are studied: those without the updated DWR error indicator (referred to as "HROM w/o Hyper-DWR") and those with the updated indicator (referred to as "HROM w/ Hyper-DWR").

6.1 One Parameter NACA0012 Airfoil in Inviscid Subsonic Flow

This test will use the NACA0012 airfoil in inviscid flow. It will have the same set up as the test in section 4.2, however the Mach number will be fixed at 0.5 and the angle of attack is varied between $[0, 4]^\circ$. The functional of interest will still be the lift coefficient. The full-order model is run on a grid with 560 cells and a polynomial of order 0, resulting in 2240 DOFs. The adaptive sampling tolerance is set to $1E - 4$. As stated previously, three ROMs will be built and compared. The first will use the unchanged adaptive sampling procedure without hyperreduction incorporated, referred to in plots as the “ROM”. The second will be a hyperreduced ROM which uses the ECSW hyperreduction method to evaluate the residual and Jacobian to find reduced-order solutions but does include the update to the DWR error indicator discussed in Chapter 5. This HROM is referred to as the “HROM w/o Hyper-DWR”. The final HROM is built using the update adaptive sampling procedure shown in Algorithm 3 which includes the new DWR error indicator which incorporates hyperreduction. This HROM is referred to as the “HROM w/ Hyper-DWR”.

Figures 6.1 to 6.3 show the results of the three models in the order they were just introduced. Each figure includes a plot of the final configuration of the parameter domain which shows the placement of the FOM snapshots and ROM points. They also include a plot of the “estimated” and “true” error distributions for each ROM. The estimated error is produced via an RBF interpolation using the snapshot locations and ROM points from the adaptive sampling cycle. The true error is found by sampling the error between each ROM and the FOM at 20 evenly distributed points across the parameter space at the end of the sampling procedure. The functional error tolerance bound is also plotted. It can be

seen that all three processes successfully predicted the online behaviour of the ROMs, as the “true” error remained inside of the tolerance bound for all of the plots.

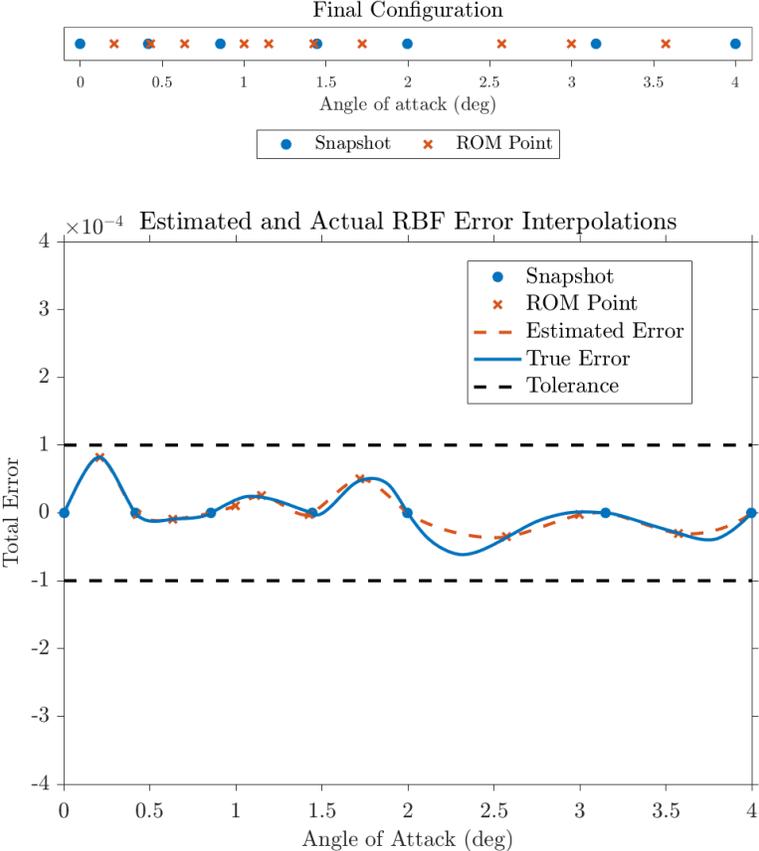


Figure 6.1: Snapshot and ROM Points with Estimated and True Error for the ROM

From Figure 6.2, we can see that for the HROM without the hyperreduced error indicator the distribution of the snapshots is similar to the ROM, with one additional point added at 0.02° . This is likely because the error indicator is what dictates the placement of the snapshot locations, so without consideration of the hyperreduction in the DWR error indicator, the placement of the snapshots will remain similar unless the ROM point solutions are also inaccurate due to hyperreduction of the residual and Jacobian in the Newton iterations in Equation 3.9.

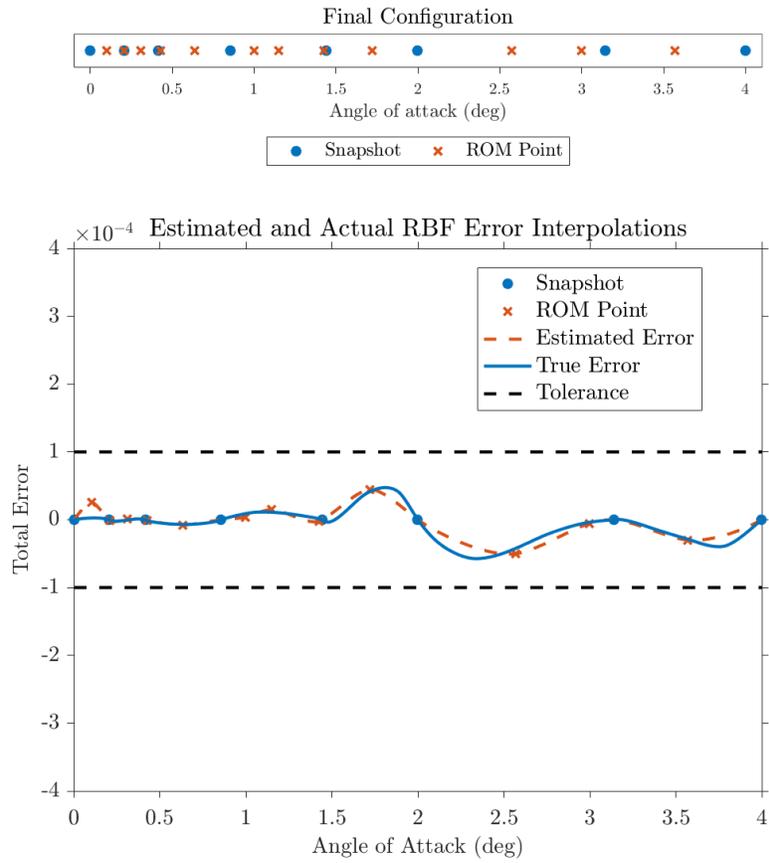


Figure 6.2: Snapshot and ROM Points with Estimated and True Error for the HROM w/o Hyper-DWR

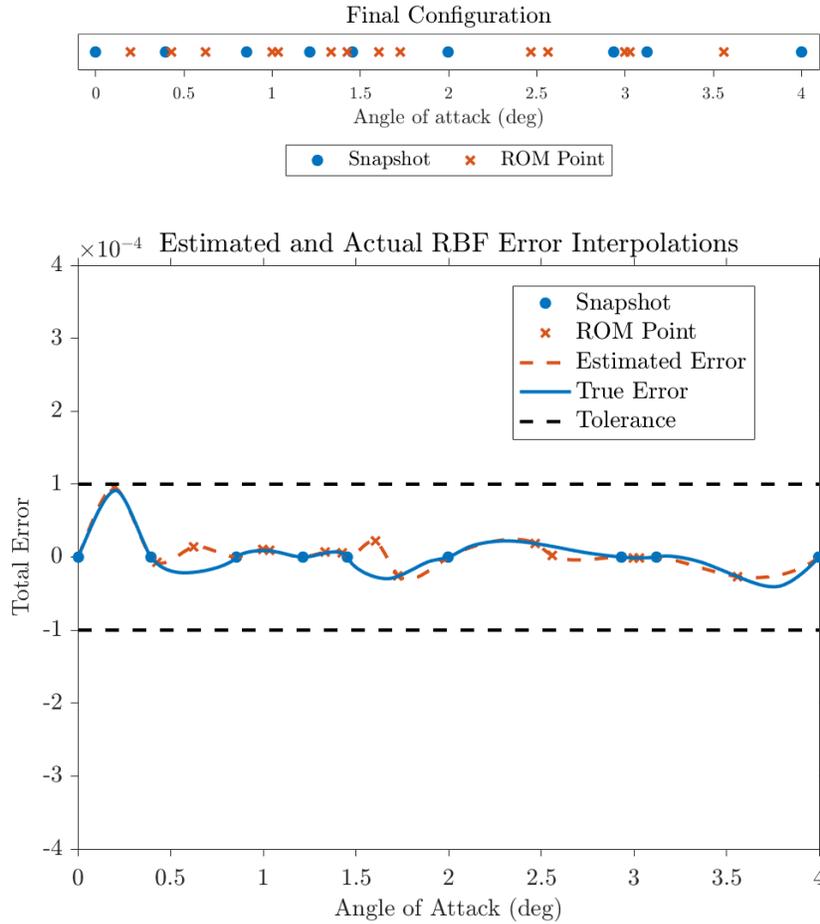


Figure 6.3: Snapshot and ROM Points with Estimated and True Error for the HROM w/ Hyper-DWR

Figure 6.3 shows that once hyperreduction is introduced into the error indicator, more snapshots are required to achieve the same error tolerance as in the ROM. In Figure 6.4, the maximum and average error at all of the ROM points for each model are plotted over the adaptive sampling iterations. With the introduction of hyperreduction and the additional layer of approximation, one additional adaptive sampling cycle is required for the HROM w/o hyper-DWR and two additional sampling cycles are required for the HROM w/ hyper-DWR to achieve the same tolerance that the ROM reached in 5 cycles.

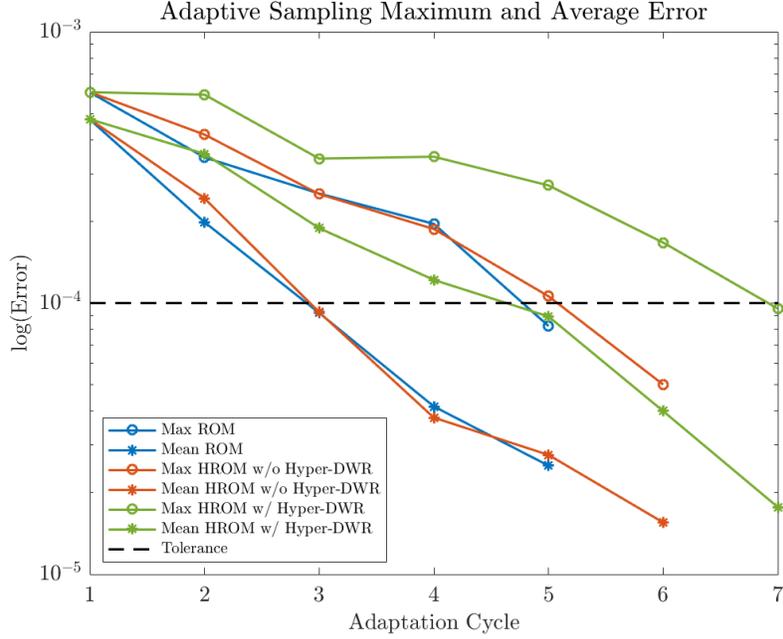


Figure 6.4: Maximum and Mean Error Estimate at the ROM Points during each Sampling Iteration for the Three Models

In Table 6.1, key characteristics and quantities for each of the models are summarized. We can see that the dimension of the POD n increases as hyperreduction is introduced first in the ROM solutions and then into the DWR error indicators, however, all are still significantly smaller than the FOM dimension N of 2240. The reduction in the number of elements used to evaluate the residual and Jacobian is also shown, for the HROM w/o hyper-DWR only 39 of the 560 mesh elements are required and 45 are used in the HROM w/ hyper-DWR. Figures 6.5 and 6.6 show the selected elements that are included in the reduced mesh for the HROM w/o and w/ Hyper-DWR, respectively. It can be seen that both approaches include some of the far field mesh elements ahead of the airfoil as well as some of the elements aft. Both include a majority of the cell directly behind the trailing edge of the airfoil. All three models have accuracy within the same orders of magnitude at the ROM points from the adaptive sampling cycle as well as the 20 points used to evaluate the models “online”. Both HROMs also outperform the ROM online, likely due to their larger POD dimension.

Table 6.1: One Parameter NACA0012 Airfoil ROM and HROM Important Dimensions, Average ROM Point Error and Average Online Error

Model	POD Dim. n	$\ \xi\ _0$	Avg. ROM Error	Avg. Online Error
ROM	7	-	$2.5162E - 5$	$2.2850E - 5$
HROM w/o Hyper-DWR	8	39	$1.5532E - 5$	$1.7671E - 5$
HROM w/ Hyper-DWR	9	45	$1.7692E - 5$	$1.5497E - 5$

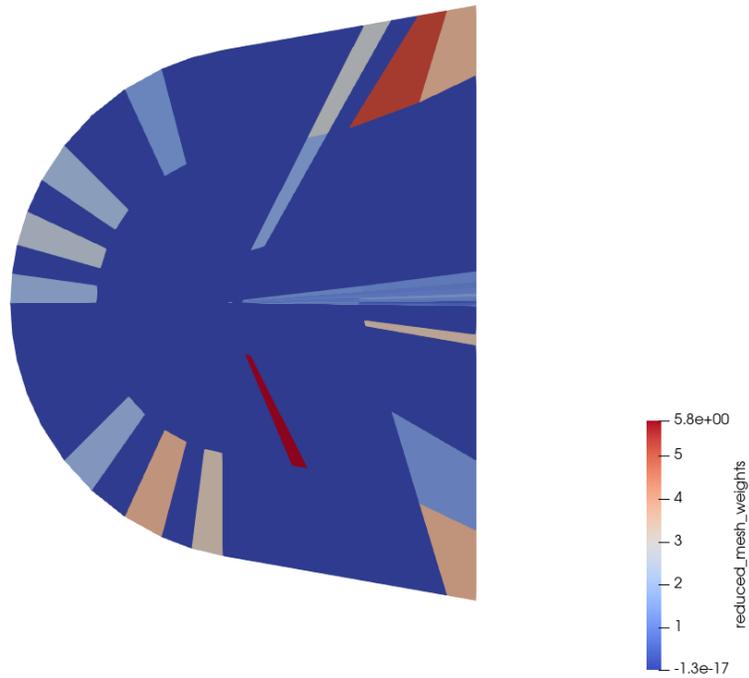


Figure 6.5: ECSW Reduced Mesh Set for the HROM w/o Hyper-DWR

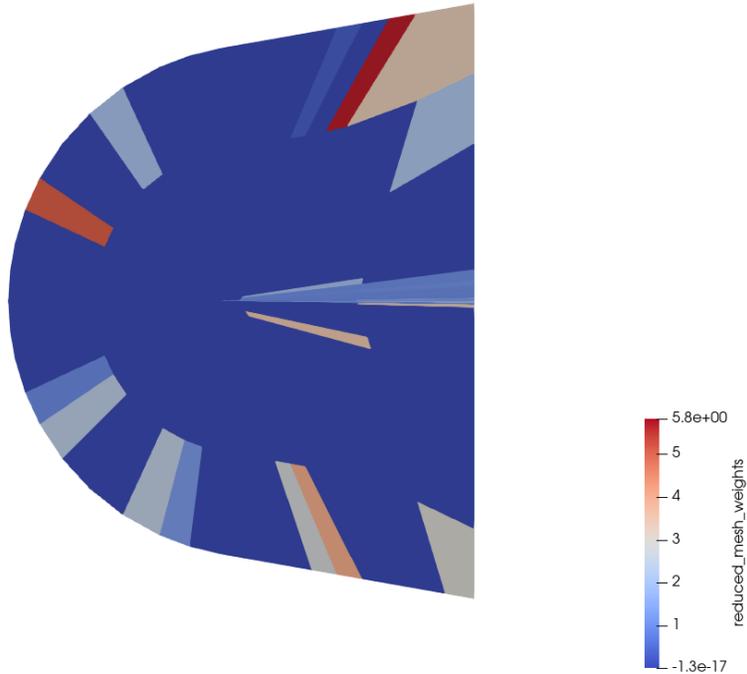


Figure 6.6: ECSW Reduced Mesh Set for the HROM w/ Hyper-DWR

6.2 Two Parameter NACA0012 Airfoil in Transonic Flow

This case uses the same setup as section 4.2, however, the adaptive sampling tolerance is increased to $3E - 4$. The design parameters remain Mach number in the transonic range $[0.5, 0.9]$ and angle of attack α between $[0, 5]^\circ$, and the functional of interest is the lift coefficient. Figures 6.7 to 6.9 show the final configuration of the parameter space for each ROM as well as the error distribution found from the ROM points at the last adaptive sampling iteration. For the ROMs which included hyperreduction in some capacity, the FOM snapshots used to train the ECSW hyperreduction approach are highlighted using green circles. These were selected for training as they are solved for at the very first iteration of the sampling procedure and can therefore be used to train the reduced mesh at every iteration,

and are evenly distributed in the parameter domain. Further investigation into how to select these points is left as future work.

It can be seen that both hyperreduction in the ROM solution and the DWR error indicator change the distribution of the interpolated error, and therefore the final snapshot placements for all three models differ. The results show that both the HROM w/o Hyper-DWR and HROM w/ Hyper-DWR sampling procedures tend to favor placing snapshot locations at higher Mach numbers and on the boundaries of the parameter space.

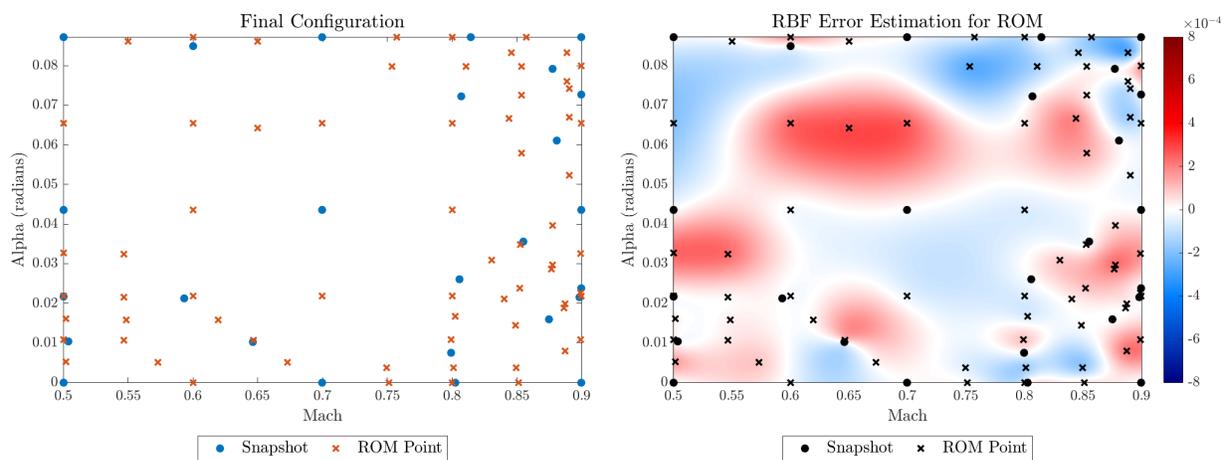


Figure 6.7: Final Configuration of the Parameter Space for the ROM and the Interpolated Estimated Error Distribution from the ROM Points

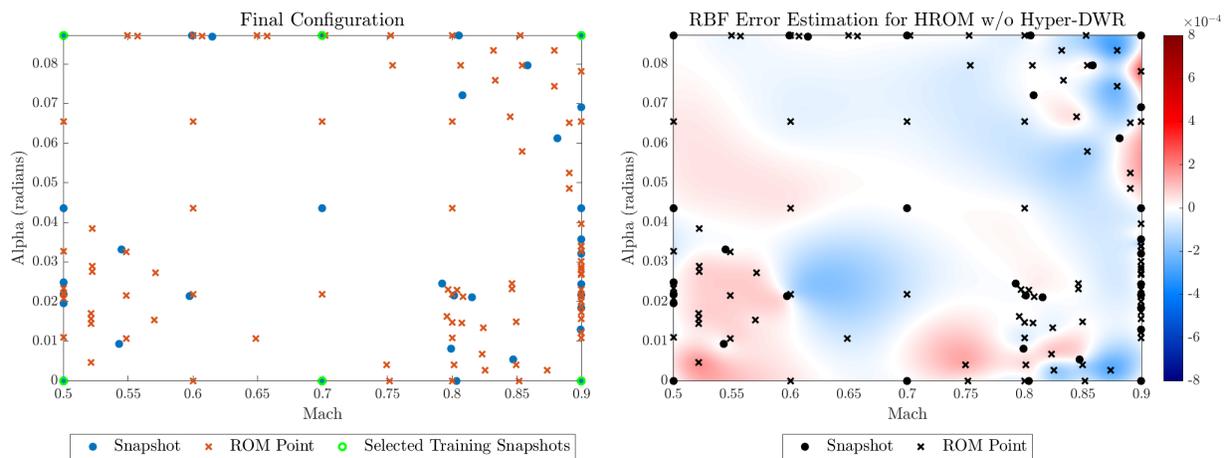


Figure 6.8: Final Configuration of the Parameter Space for the HROM w/o Hyper-DWR and the Interpolated Estimated Error Distribution from the ROM Points

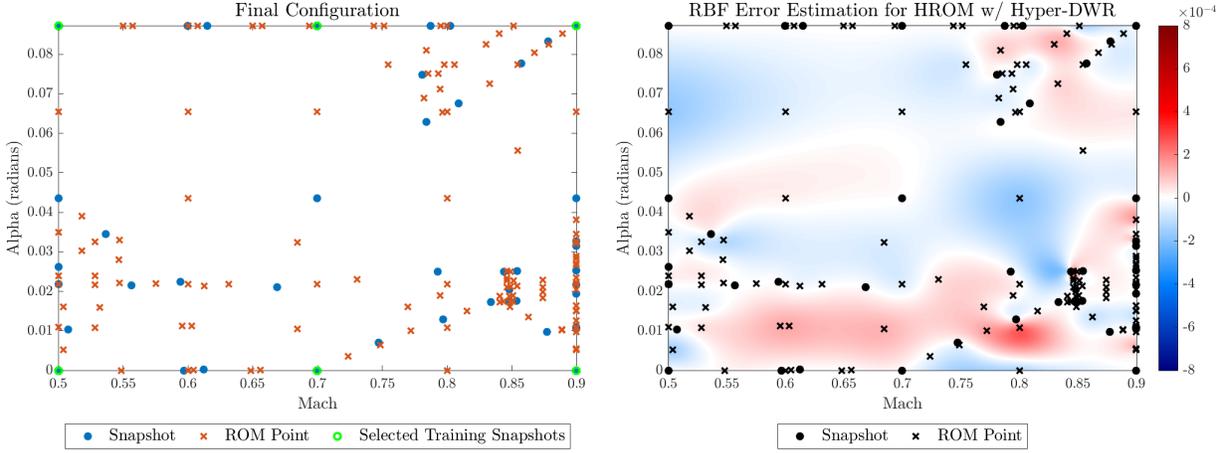


Figure 6.9: Final Configuration of the Parameter Space for the HROM w/ Hyper-DWR and the Interpolated Estimated Error Distribution from the ROM Points

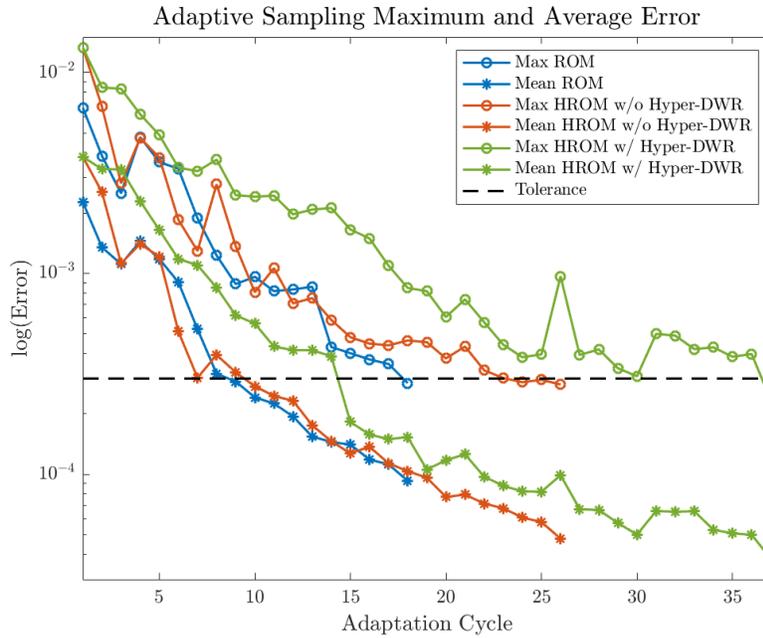


Figure 6.10: Maximum and Mean Error Estimate at the ROM Points during each Sampling Iteration for the Three Models

Figure 6.10 shows the maximum and average error at the ROM points during each adaptation cycle for the three models. It can be seen that the HROM w/ Hyper-DWR tends to have higher errors, which is expected as the hyperreduction introduces an additional approx-

imation to the model. This is not captured in the HROM w/o Hyper-DWR, and therefore it terminates earlier than the other HROM but still later than the ROM.

Figures 6.11 to 6.13 show the expected online behaviour or “true” error for each of the models as well as the areas of the domain where the functional error tolerance of $3E - 4$ is not met. This distribution is found by evenly distributing 400 points over the parameter space and using an RBF interpolation to approximate the error everywhere. Here we can see the impact of updating the DWR error indicator to include the approximations introduced by hyperreduction. In Figure 6.11, we can see the ROM meets the tolerance bound over almost the entire domain, except for a very small region of high Mach numbers and high angles of attack. This suggests the adaptive sampling procedure and error estimates closely predicted the online behaviour of the model. In contrast, in Figure 6.12, the HROM w/o Hyper-DWR has multiple larger regions where the tolerance is not met online. The error behaviours in these regions were not predicted by the RBF interpolation from the sampling procedure in Figure 6.8 and were therefore not considered during the sampling procedure. Once the hyperreduced DWR is used, the tolerance is met over the entire parameter space, as seen in Figure 6.13. This suggests the updated error indicator is correctly capturing the additional error introduced by the hyperreduction, allowing the adaptive sampling procedure to build an accurate HROM with a controllable functional error prediction.

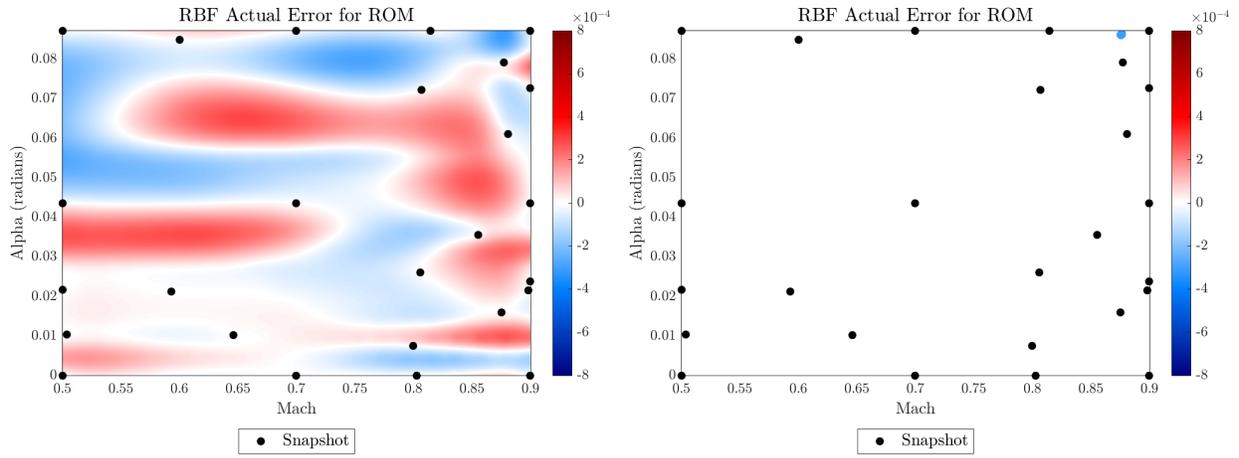


Figure 6.11: True Error for the ROM and Zones of Tolerance Violation

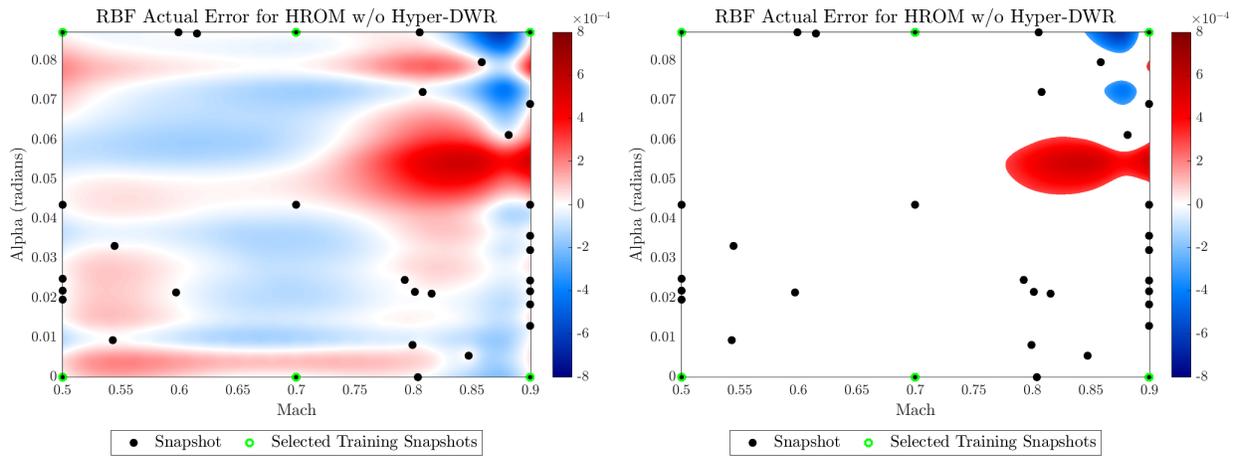


Figure 6.12: True Error for the HROM w/o Hyper-DWR and Zones of Tolerance Violation

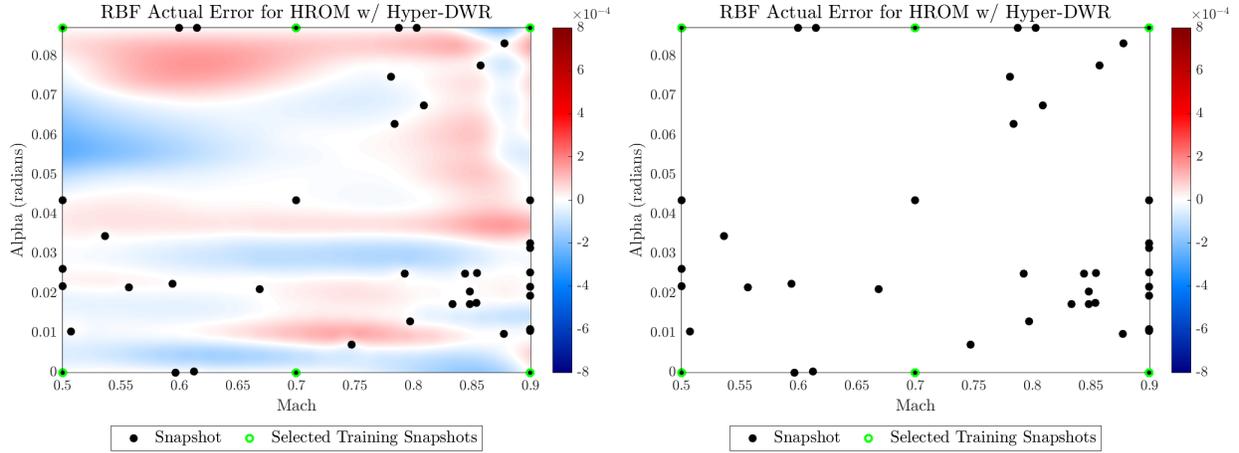


Figure 6.13: True Error for the HROM w/ Hyper-DWR and Zones of Tolerance Violation

Table 6.2 summarizes some of the key dimensions of the three models as well as the average error at the ROM points and at the 400 points used to estimate the online behaviour. The HROM w/o Hyper-DWR has reduced-space dimension n of 34 which is 9 larger than the ROM dimension, and 152 elements in the reduced mesh. The HROM w/ Hyper-DWR has a larger n of 40, note however it is still true that $n \ll N$ given that N is 2240, and a reduced mesh with 179 elements down from the 560 FOM mesh. This results in a 68 percent reduction in the number of elements used to evaluate the residual and Jacobian. The reduced mesh of the two HROMs is shown in Figures 6.14 and 6.15. It can be seen that in comparison to the one parameter subsonic case, these reduced meshes tend to weigh more heavily elements closer to the airfoil, which may be because in transonic flow these elements are more relevant to changes in flow solution, and in turn the residual and Jacobian.

Table 6.2: Two Parameter NACA0012 Airfoil in Transonic Flow ROM and HROM Important Dimensions, Average ROM Point Error and Average Online Error

Model	POD Dim. n	$\ \xi\ _0$	Avg. ROM Error	Avg. Online Error
ROM	26	-	$9.2964E - 5$	$9.4169E - 5$
HROM w/o Hyper-DWR	34	152	$4.7918E - 5$	$8.7406E - 5$
HROM w/ Hyper-DWR	45	179	$3.8079E - 5$	$4.8941E - 5$

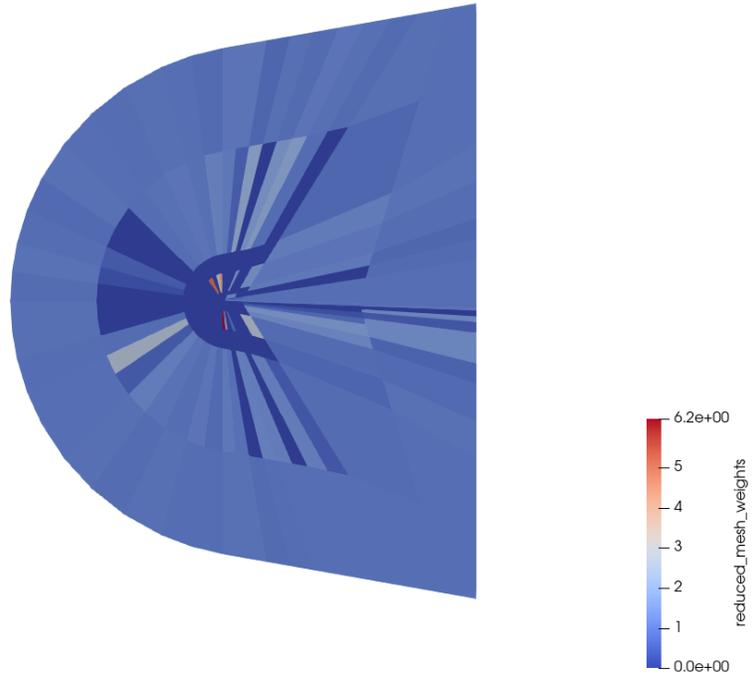


Figure 6.14: ECSW Reduced Mesh Set for the HROM w/o Hyper-DWR

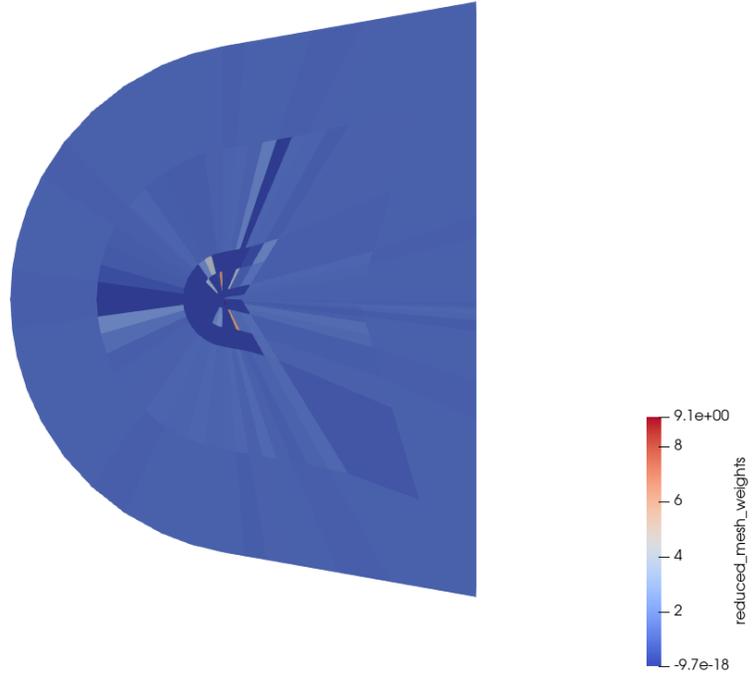


Figure 6.15: ECSW Reduced Mesh Set for the HROM w/ Hyper-DWR

Next, the impact of the hyperreduction on the computational cost of the adaptive sampling procedure can be studied. The work units discussed in section 5.4 for each of the models are plotted against the average ROM point error, as seen in Figure 6.16. It can be seen that although the POD dimension n and reduced mesh are slightly larger for the HROM w/ Hyper-DWR, each iteration is much less computationally expensive than the other two models due to the savings in evaluating the residual and Jacobian. The addition of hyperreduction into the second DWR indicator results in significant savings as well, since this must be evaluated at all the ROM points from previous iterations, the number of which only grows as the adaptive sampling cycle proceeds. Overall, it appears that by incorporating hyperreduction into the ROM the true computational savings can be realized and with the updated error indicator the online behaviour can be accurately predicted.

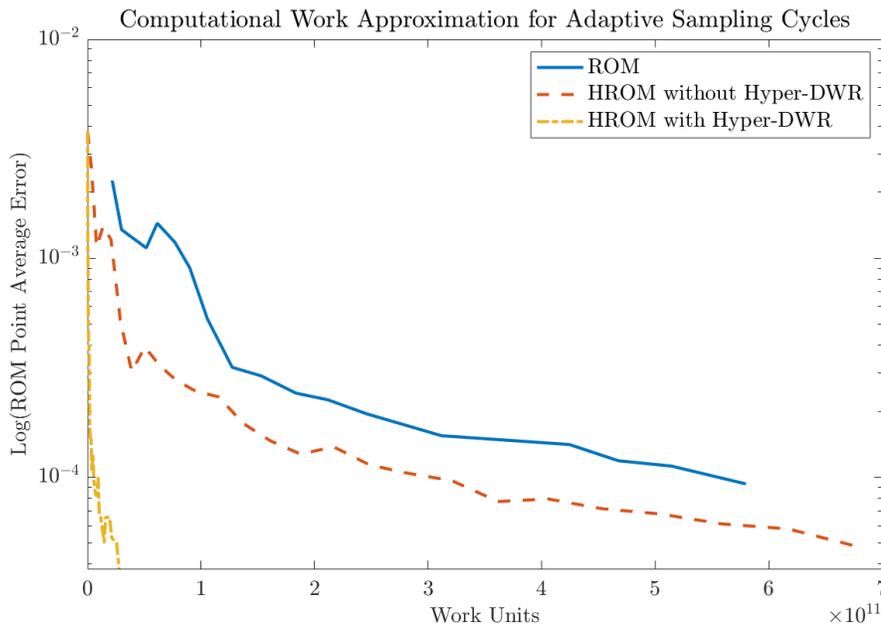


Figure 6.16: Work Units for the Adaptive Sampling Procedure to Build the Three Models

6.3 One Parameter NACA0012 Airfoil in Inviscid Subsonic Flow on a Fine Mesh

The following test will use the same setup as section 6.1, however, a finer mesh will be used which contains 2240 elements rather than 540, resulting in 8960 DOFs. Going forward, we will no longer build the HROM without the updated DWR as we have seen that it is necessary to predict the online behaviour of the model and is more computationally efficient. Figures 6.17 and 6.18 show the final distribution of FOM snapshots in the parameter space and the error distributions at the end of the adaptive sampling cycle and “online”. The HROM tends to place new snapshots quite close together. This may be because the error in the DWR error indicators is growing too quickly between cycles, as they are only approximations of the error at each ROM. Further investigation into this is left as future work.

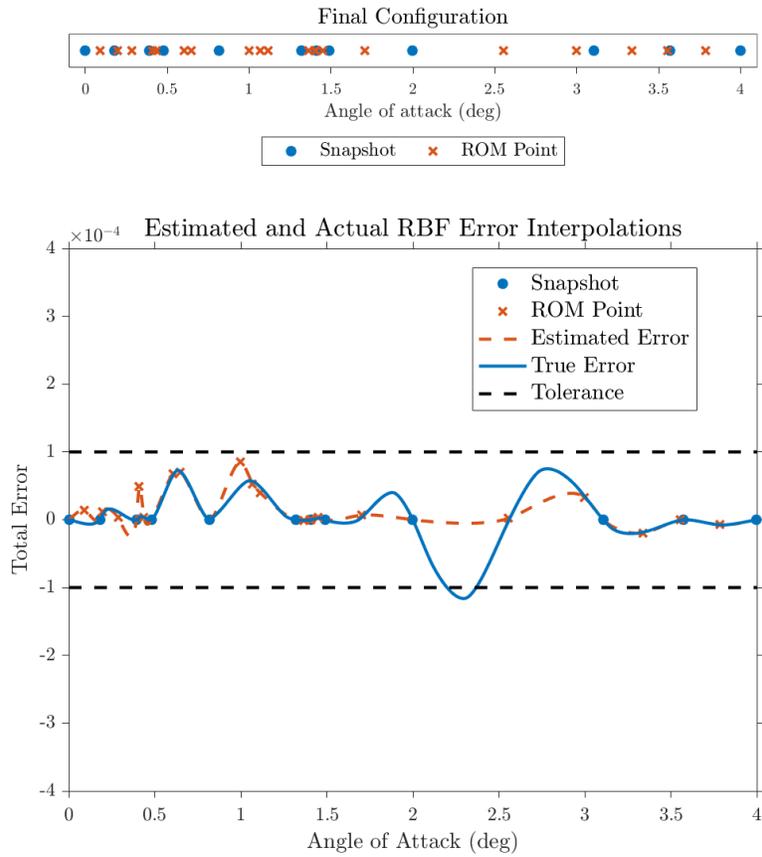


Figure 6.17: Snapshot and ROM placement with Estimated and True Error across the parameter space at the end of the Adaptive Sampling Procedure for the ROM

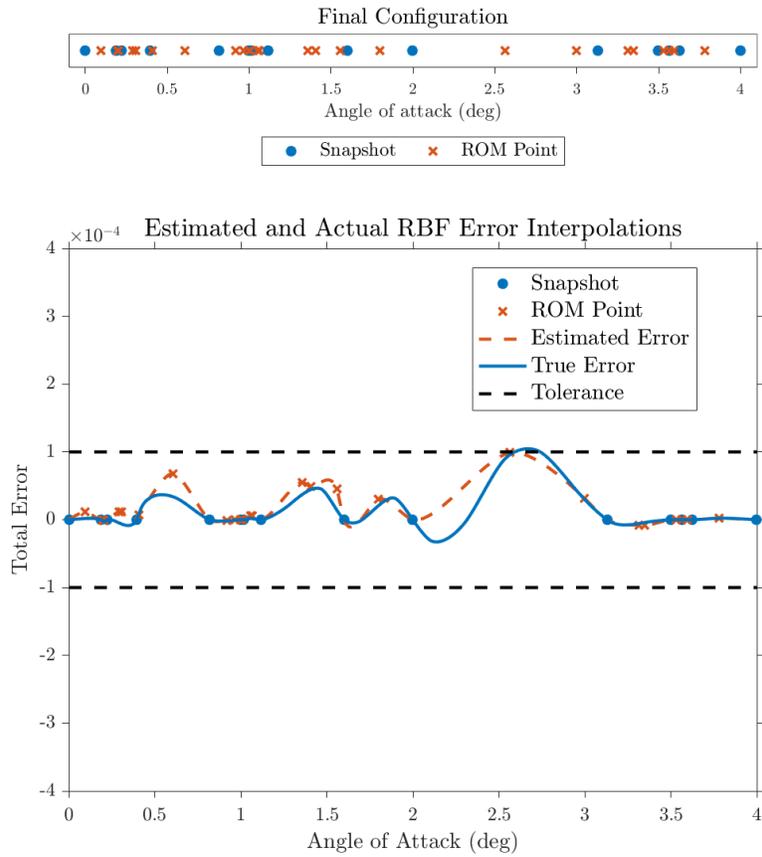


Figure 6.18: Snapshot and ROM placement with Estimated and True Error across the parameter space at the end of the Adaptive Sampling Procedure for the HROM w/ Hyper-DWR

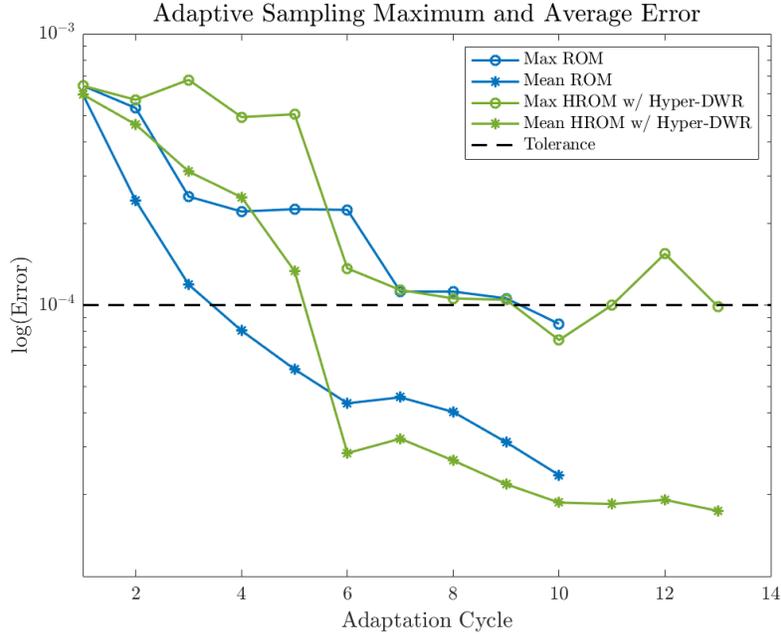


Figure 6.19: Maximum and Mean Error Estimate at the ROM Points during each Sampling Iteration for the ROM and the HROM with the Hyperreduced Error

Figure 6.19 shows the average and maximum ROM point error for each model over the adaptive sampling process. We can see that the HROM takes 3 additional cycles to achieve the same tolerance, which is expected as it introduces an extra approximation layer and a new source of error. Table 6.3 summarizes some of the key dimensions of the models as well as the average error from the ROM points at the end of the adaptive sampling cycle and the points used to find the “true” online error. Although the reduced-subspace dimension n is slightly larger for the HROM, significant computational savings will come from the reduced mesh. The original FOM mesh has 2240 elements and the hyperreduction technique only requires 72 of these elements to accurately approximate the variation in the residual and Jacobian. This is a 96.8% reduction in the number of elements that need to be evaluated for these quantities, which is even larger than in the case of the 560-element mesh in section 6.1. We suspect that the finer the mesh gets or the larger the number of total elements, the greater the savings from the ECSW hyperreduction approach. Figure 6.20 shows the reduced mesh

set selected for the ECSW hyperreduction technique. Comparing this to the coarser mesh in the transonic case in Figure 6.15, we suspect that having a mesh with smaller elements of similar sizes allows the ECSW method to more accurately select important elements as we can see a clearer pattern in this finer mesh. A majority appear to be placed aft of the airfoil, specifically in the wake above and below the airfoil. Future work can be conducted on problems with larger meshes to study this relationship further.

Table 6.3: One Parameter NACA0012 Airfoil in Subsonic Flow ROM and HROM Important Dimensions, Average ROM Point Error and Average Online Error

Model	POD Dim. n	$\ \xi\ _0$	Avg. ROM Error	Avg. Online Error
ROM	12	-	$2.3611E - 5$	$2.7680E - 5$
HROM w/ Hyper-DWR	15	72	$1.7420E - 5$	$2.1243E - 5$

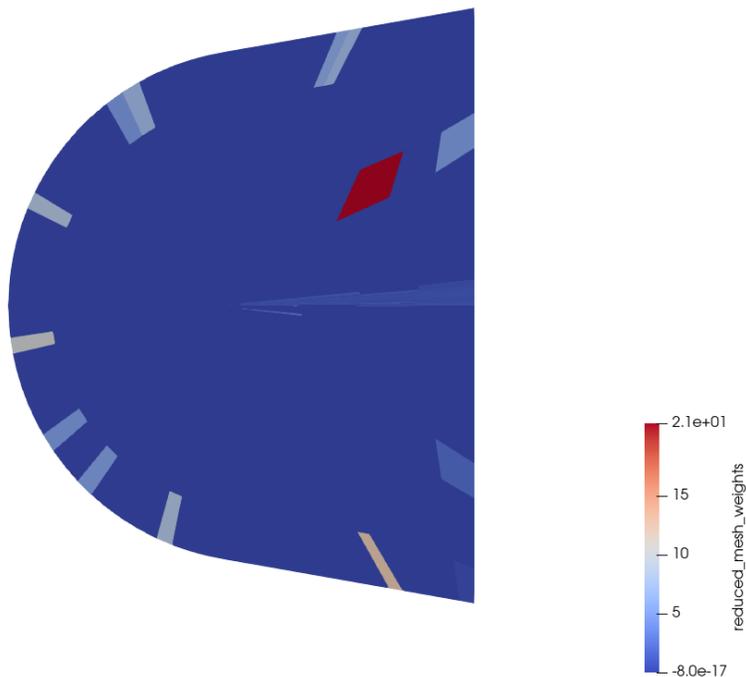


Figure 6.20: ECSW Reduced Mesh set for the HROM w/ Hyper-DWR

6.4 Two Parameter NACA0012 Airfoil in Inviscid Subsonic Flow on a Fine Mesh

The final test case will be for a two-design parameter NACA0012 airfoil in inviscid subsonic flow. The parameters are the Mach number in the range $[0.3, 0.55]$ and the angle of attack in the range $[0, 3]^\circ$. The finer mesh used in section 6.3 will be used here, which has 2240 cells and 8960 DOFs. The functional of interest will still be the lift coefficient, for which the estimated error tolerance is set to $1E - 4$.

Figures 6.21 and 6.22 show the final parameter space configurations and error distributions for the ROM and HROM. It can be seen that both tend to place more snapshots at the highest Mach number, this is likely because more variation is seen in the lift coefficient and flow solutions at higher Mach numbers.

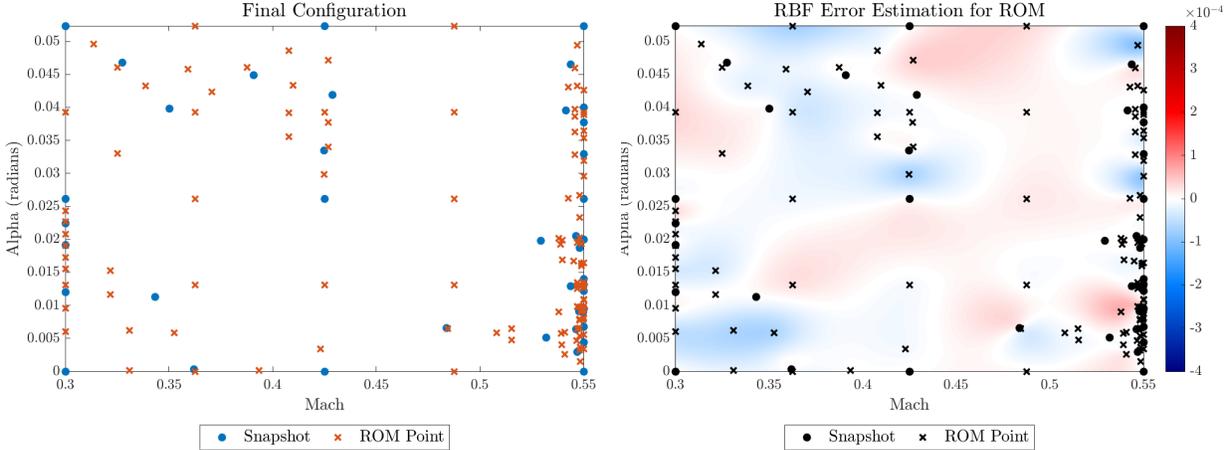


Figure 6.21: Final Configuration of the Parameter Space for the ROM and the Interpolated Estimated Error Distribution from the ROM Points

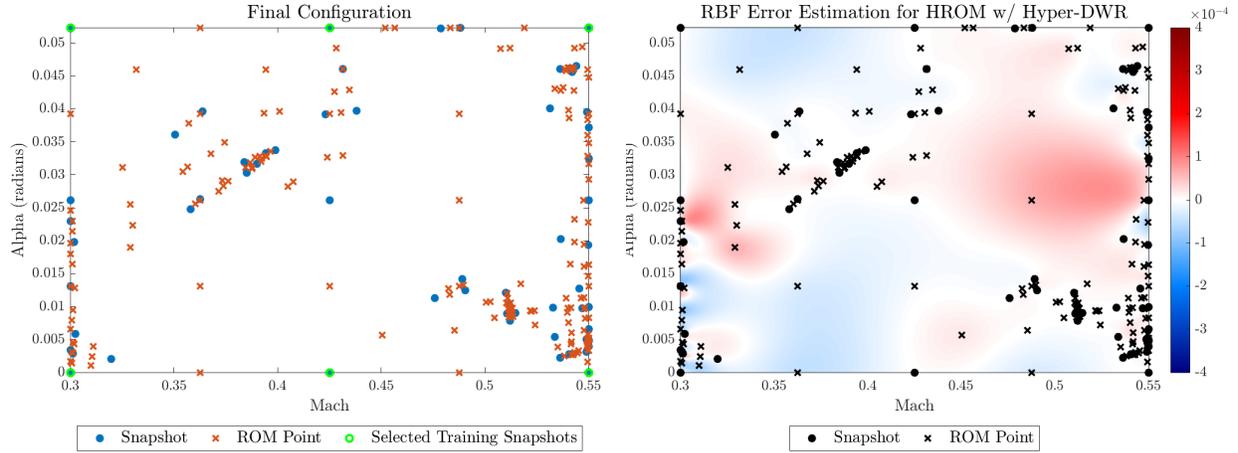


Figure 6.22: Final Configuration of the Parameter Space for the HROM w/ Hyper-DWR and the Interpolated Estimated Error Distribution from the ROM Points

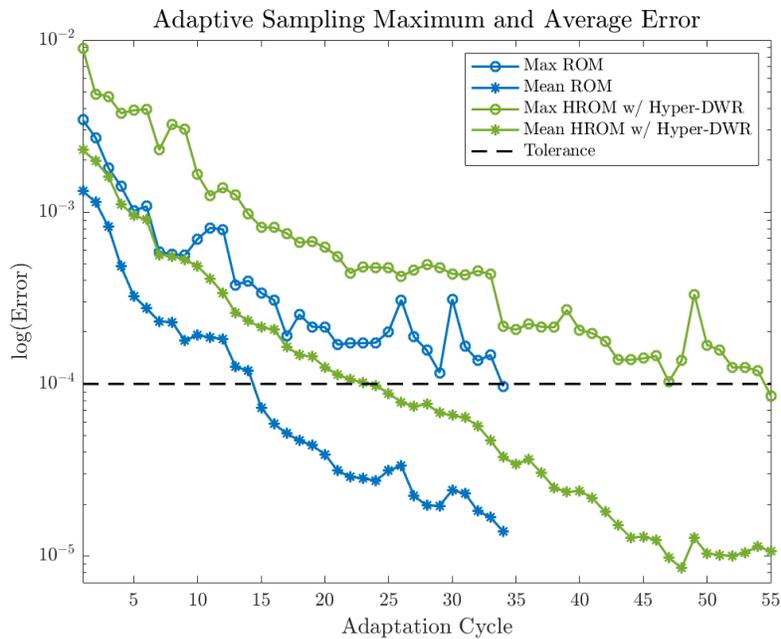


Figure 6.23: Maximum and Mean Error Estimate at the ROM Points during each Sampling Iteration for the ROM and the HROM with the Hyperreduced Error

Figure 6.23 shows the average and maximum ROM point error from the two models over the sampling cycles. We can see that the HROM takes 21 more iterations to achieve the same tolerance. It is interesting to note however that the average error dips below the

tolerance quite quickly but the maximum ROM point error takes longer to drop and even plateaus at some of the later iterations. This may be due to the training data used to find the ECSW reduced mesh set and weights. The error introduced by hyperreduction cannot be overcome by adding more snapshots into the POD, as the same six points highlighted in Figure 6.22 are always used for training. Reviewing the snapshot distribution, we can also see the HROM ends up placing more snapshots in the center of the parameter space than the ROM. This may be because the procedure is trying to compensate for the lack of points in this region used for hyperreduction. Future work could look into how this training data may be intelligently selected as well to result in the most accurate reduced mesh set.

Table 6.4: Two Parameter NACA0012 Airfoil in Subsonic Flow ROM and HROM Important Dimensions, Average ROM Point Error and Average Online Error

Model	POD Dim. n	$\ \xi\ _0$	Avg. ROM Error	Avg. Online Error
ROM	42	-	$1.3841E - 5$	$2.4963E - 5$
HROM w/ Hyper-DWR	63	221	$1.0600E - 5$	$2.0884E - 5$

Table 6.4 summarizes the key dimensions as well as the average ROM point and online error for the two models. The key result to note is that in this case, which is more complex than the one-parameter subsonic case in the previous section, there still is a significant reduction in the number of elements required to accurately evaluate the residual and Jacobian. The FOM mesh contains 2240 elements, while the HROM only requires 221 to solve for these quantities in the Newton iteration in Equation 3.9 and in the DWR error indicator in Equation 5.9. This is a 90.1% reduction in the number of mesh elements. We can note that the savings in the subsonic case are even greater than the two-parameter transonic case. This may be because of the mesh size, having more cells resulting in an even greater reduction in the mesh, or the complexity of the problem. Figure 6.24 shows the reduced mesh set for the HROM.

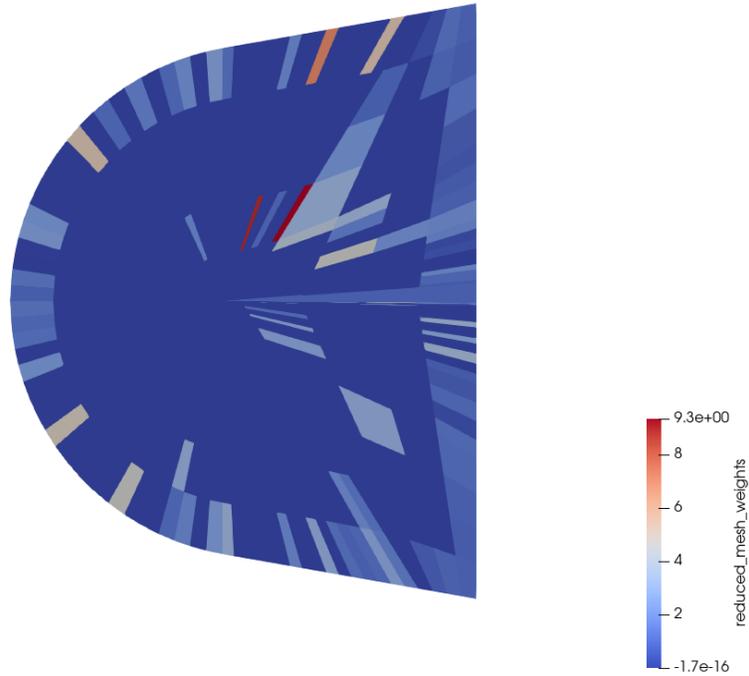


Figure 6.24: ECSW Reduced Mesh set for the HROM w/ Hyper-DWR

Figures 6.25 and 6.26 show the true error distributions from the ROM and HROM, respectively. We can see both models perform quite well online and only have small regions where the tolerance bound is not met.

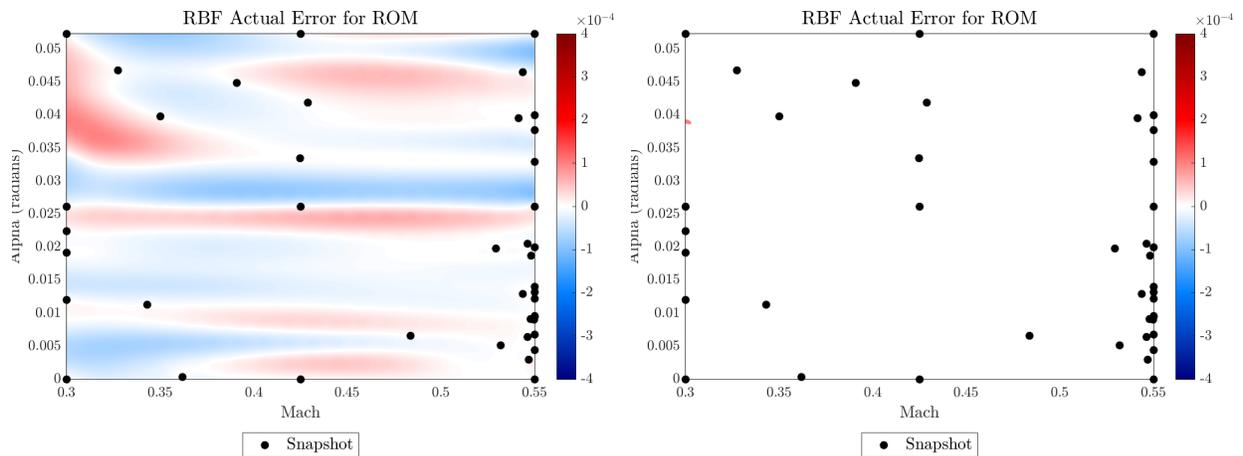


Figure 6.25: True Error for the ROM and Zones of Tolerance Violation

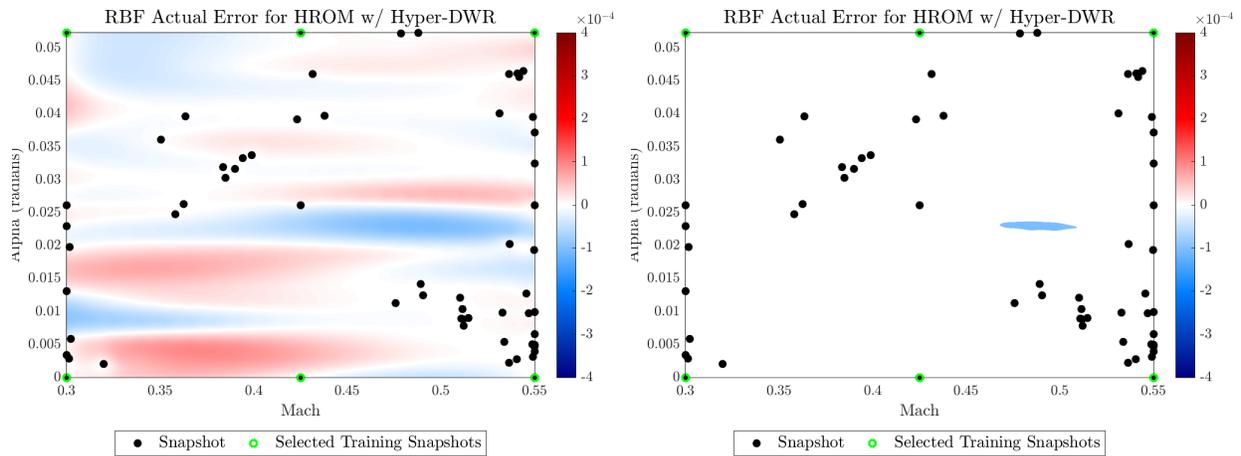


Figure 6.26: True Error for the HROM w/ Hyper-DWR and Zones of Tolerance Violation

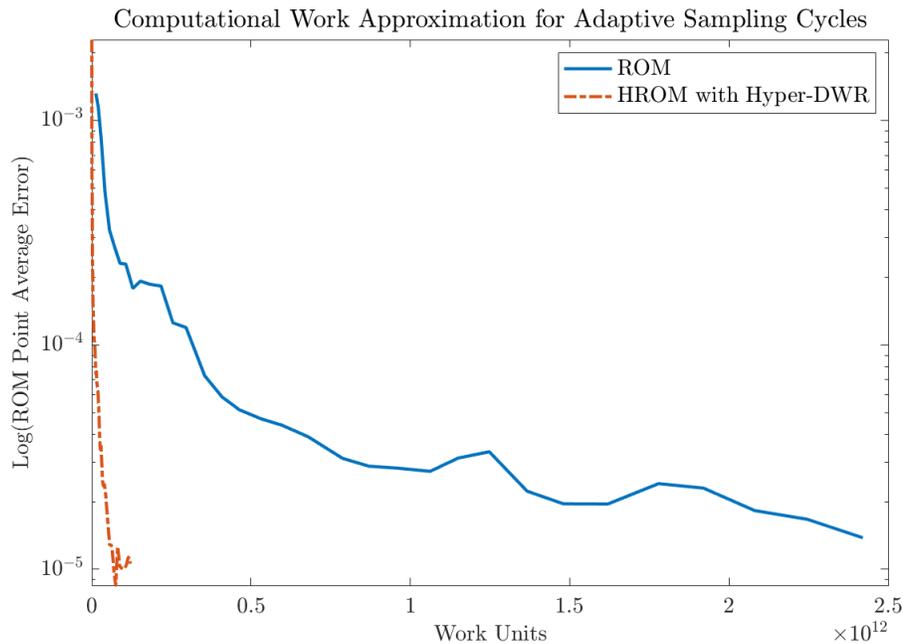


Figure 6.27: Work Units for the Adaptive Sampling Procedure to Build the Three Models

Finally, the work units for each model are plotted in Figure 6.27. We can see the HROM with the hyperreduced DWR error indicator costs much less to achieve the same average ROM point error. This means the HROM is both less expensive to assemble offline and also more computationally efficient online through the approximation of the residual and Jacobian.

Chapter 7

Conclusion

This thesis explores the use of hyperreduction as a way to address the computational bottlenecks caused by the re-evaluation of high-dimensional quantities in a projection-based reduced-order modelling framework, specifically for parametric, highly nonlinear computational models. The energy-conserving sampling and weighting technique was selected out of the many available techniques for hyperreduction. It had been previously used in least-squares Petrov-Galerkin projection models and has been shown to be robust, accurate, and stable [15, 19, 23, 45].

First, it was verified that the ECSW method successfully identified a reduced mesh set in both one-dimensional and two-dimensional test cases with one or two parameters. As expected, an additional source of error is introduced once this set is used to approximate the residual and Jacobian in a ROM. The accuracy of the hyperreduction is impacted both by the source of the training data and the NNLS tolerance used. The Jacobian-based training data proved to be more robust and reliable, confirming the results found in [40]. The NNLS tolerance serves as a measure of the tradeoff between accuracy and computational savings. The lower the tolerance, the more elements are included in the reduced mesh resulting in a less

efficient HROM but likely a more accurate approximation of the projected high-dimensional quantities.

Hyperreduction is not only important for achieving computational efficiency in online numerical predictions but also for accelerating the offline construction of a ROM through a greedy or adaptive sampling procedure. While the ECSW hyperreduction technique has been used in other greedy sampling frameworks [40], these use residual or solution-based error indicators. In this work, we incorporate hyperreduction into a goal-oriented adaptive sampling method which uses dual-weighted residual errors that tailor a reduced-order model to a specific output of interest [16].

One of the key contributions of [16] is the addition of a second dual-weighted residual error between a coarse and a fine reduced-order model, which makes it possible to recycle an existing reduced-order model solution computation. In this work, the error indicator is updated to measure the error between two hyperreduced reduced-order models. This allows the sampling procedure to capture the additional error introduced by the hyperreduction, and consider this when selecting new snapshot locations.

The updated sampling framework was tested on two-dimensional test cases, specifically the NACA0012 with various design parameter combinations and flow conditions. It was shown that the ECSW hyperreduction method can significantly reduce the number of elements re-evaluated in the high-dimensional quantities and that with the updated DWR errors the framework can still construct an ROM which provides a certain level of confidence in the expected error in the output functional. In all test cases, the difference between the prescribed tolerance in the sampling procedure and the true error online was small. Furthermore, the work units required to assemble the model are significantly reduced through hyperreduction in the reduced-order solutions and in the error indicator. This suggests savings both in the offline and online stages of the reduced-order modelling procedure.

7.1 Future Work

There are multiple avenues for future work in this research. The impact of problem complexity and mesh size on the effectiveness of the hyperreduction can be further studied. The scope of the research can also be extended to unsteady CFD problems, 3D problems, and problems with more complex parameter spaces. In addition, there are several components of the sampling procedure that could be examined further. These include the way in which ROM probing points are placed and the interpolation method used to model the error distribution, specifically in regards to the new hyperreduced DWR error indicator which appears to result in more snapshot clustering when compared to the original sampling distributions. When considering the hyperreduction approach, the training procedure can be studied in more detail. This could include research into how the training data is selected, including both how many snapshots are included and how their placement impacts both the computational cost of finding and the accuracy of the resulting reduced mesh set. The cost and necessity of re-training the hyperreduction at every sampling cycle could also be researched; in particular, possible additional error measures, like those in [40], could be used to determine whether the hyperreduction must be recomputed at a particular iteration. Alternatives to the non-negative least squares problem for finding the reduced set of important mesh elements could also be considered. Finally, full implementation of the hyperreduction approach which only computes the residual and Jacobian on the elements in the reduced mesh should be tested in order to study the true computational cost and storage savings.

Appendix A

Work Units Derivations

A.1 ROM Solution Non-linear Iteration Work Units

Equation 5.10 is an evaluation of the work units required to solve:

$$\left[\mathbf{V}^T \frac{\partial \mathbf{R}^{(k)T}}{\partial \mathbf{w}} \quad \frac{\partial \mathbf{R}^{(k)}}{\partial \mathbf{w}} \quad \mathbf{V} \right] \mathbf{p}^{(k)} = -\mathbf{V}^T \frac{\partial \mathbf{R}^{(k)T}}{\partial \mathbf{w}} \mathbf{R}^{(k)}. \quad (\text{A.1})$$

We assume it requires one work unit to evaluate every entry in the residual and Jacobian. Since $\mathbf{R}^{(k)} \in \mathbb{R}^N$ and $\frac{\partial \mathbf{R}^{(k)}}{\partial \mathbf{w}} \in \mathbb{R}^{N \times N}$, they require N and N^2 work units, respectively. To evaluate the LHS of the above equation, three matrix multiplications must be performed. By simply counting the number of operations required, the FLOPs necessary to perform matrix multiplication between a matrix of dimension $n \times p$ and another with dimension $p \times m$ is $nm(2p - 1)$.

Therefore, the total number of FLOPs to assemble the LHS (which are assumed to be one work unit) is:

$$\begin{aligned}
W_{\text{LHS}} &= (\text{FLOPs to multiply } \mathbf{V} \text{ and } \frac{\partial \mathbf{R}^{(k)}}{\partial \mathbf{w}}) + \\
&\quad (\text{FLOPs to multiply the matrix resulting from the previous step and } \frac{\partial \mathbf{R}^{(k)T}}{\partial \mathbf{w}}) + \\
&\quad (\text{FLOPs to multiply the matrix resulting from the previous step and } \mathbf{V}^T) \\
&= (\text{FLOPs to multiply matrices of dimensions } (N \times N) \text{ and } (N \times n_i)) + \\
&\quad (\text{FLOPs to multiply matrices of dimensions } (N \times N) \text{ and } (N \times n_i)) + \\
&\quad (\text{FLOPs to multiply matrices of dimensions } (n_i \times N) \text{ and } (N \times n_i)) \\
&= [Nn_i(2N - 1)] + [Nn_i(2N - 1)] + [n_i^2(2N - 1)] \\
&= (2Nn_i + n_i^2)(2N - 1).
\end{aligned} \tag{A.2}$$

Similarly, to assemble the RHS:

$$\begin{aligned}
W_{\text{RHS}} &= (\text{FLOPs to multiply } \mathbf{R}^{(k)} \text{ and } \frac{\partial \mathbf{R}^{(k)}}{\partial \mathbf{w}}) + \\
&\quad (\text{FLOPs to multiply the matrix resulting from the previous step and } \mathbf{V}^T) \\
&= (\text{FLOPs to multiply matrices of dimensions } (N \times N) \text{ and } (N \times 1)) + \\
&\quad (\text{FLOPs to multiply matrices of dimensions } (n_i \times N) \text{ and } (N \times 1)) \\
&= [N(2N - 1)] + [n_i(2N - 1)] \\
&= (N + n_i)(2N - 1).
\end{aligned} \tag{A.3}$$

Therefore, the total cost to assemble the linear system is:

$$\begin{aligned} W_{\text{assemble}} &= (2Nn_i + n_i^2)(2N - 1) + (N + n_i)(2N - 1) \\ &= (2Nn_i + n_i^2 + N + n_i)(2N - 1). \end{aligned} \tag{A.4}$$

The work units, as shown in Chapter 5, for one non-linear iteration for the ROM solution is:

$$\begin{aligned} W_{\text{non lin}} &= (\text{units to evaluate the residual}) + \\ &\quad (\text{units to evaluate the Jacobian}) + \\ &\quad W_{\text{assemble}} + \\ &\quad (\text{units to solve for solution update}) \\ &= (N) + (N^2) + [(2Nn_i + n_i^2 + N + n_i)(2N - 1)] + (n_i^3). \end{aligned} \tag{A.5}$$

A.2 HRDM Solution Non-linear Iteration Work Units

Once the ECSW hyperreduction is introduced, the cost to assemble and solve the Newton iterations changes. The following equation is the one to be evaluated:

$$\left[\tilde{\mathbf{W}}^{(k)T} \tilde{\mathbf{W}}^{(k)} \right] \mathbf{p}^{(k)} = -\tilde{\mathbf{R}}^{(k)}, \tag{A.6}$$

where $\tilde{\mathbf{W}}^{(k)}$ is the hyperreduced test basis and $\tilde{\mathbf{R}}^{(k)}$ is the hyperreduced residual. Recall, the hyperreduced test basis can be written as:

$$\tilde{\mathbf{W}}^{(k)} = \bar{\mathbf{J}}^{(k)} \mathbf{V} = \left(\sum_{e \in \tilde{\mathcal{E}}} \xi_e \mathbf{L}_e^T \mathbf{J}_e^{(k)} \mathbf{L}_{e^+} \right) \mathbf{V}. \tag{A.7}$$

It can be shown that for a sparse matrix, the number of FLOPs for matrix multiplication with a dense matrix (i.e. $S * D$ where S is sparse) is two times the number of non-zeros in S , times the number of columns in D [59]. Again, assuming the cost of evaluating one entry in the matrix is one work unit, the cost to assemble the elemental Jacobian contribution $\mathbf{J}_e^{(k)}$ is $d_e d_{e+}$. This must be then multiplied by the number of elements in the reduced mesh n_{e_i} . Assuming we can find the elemental Jacobian contribution in the global dimension (i.e. ignoring the matrix multiplication with \mathbf{L}_e and \mathbf{L}_{e+}), the cost of evaluating the hyperreduced test basis is:

$$\begin{aligned}
W_{\text{test}} &= (\text{units to evaluate the } n_{e_i} \text{ Jacobian contributions from each element in Equation 3.28}) + \\
&\quad (\text{units to sum each elemental contribution}) + \\
&\quad (\text{FLOPs to multiply the matrix resulting from the previous step and } \mathbf{V}) \\
&= (n_{e_i} d_e d_{e+}) + (n_{e_i} d_e d_{e+}) + \\
&\quad (\text{FLOPs to multiply matrices of dimensions } (N \times N) \text{ and } (N \times n_i) \text{ where the first is sparse}) \\
&= (2n_{e_i} d_e d_{e+}) + (2n_{e_i} d_e d_{e+} n_i).
\end{aligned} \tag{A.8}$$

where n_{e_i} is the number of elements in the reduced mesh at a particular sampling cycle i .

The hyperreduced residual can be written as:

$$\hat{\mathbf{R}}^{(k)} \approx \tilde{\mathbf{R}}^{(k)} = \sum_{e \in \tilde{\mathcal{E}}} \xi_e \tilde{\mathbf{W}}^{(k)T} \mathbf{L}_e^T \mathbf{R}_e^{(k)}. \tag{A.9}$$

Again neglecting the projection of the elemental residual into the FOM dimension, the work units to find the hyperreduced is:

$$\begin{aligned}
W_{\text{residual}} &= (\text{units to evaluate the } n_e \text{ residual contributions from each element in Equation 3.15}) + \\
&\quad (\text{FLOPs to multiply the matrix resulting from the previous step and } \tilde{\mathbf{W}}^{(k)}) + \\
&\quad (\text{units to sum each elemental contribution}) \\
&= (n_{e_i} d_e) + (2n_{e_i} n_i d_e) + (n_{e_i} n_i) \\
&= n_{e_i} (d_e + 2n_i d_e + n_i).
\end{aligned} \tag{A.10}$$

The cost to assemble the LHS and RHS of Equation A.6 is then:

$$\begin{aligned}
W_{\text{assemble}} &= (\text{FLOPs to multiply } \tilde{\mathbf{W}}^{(k)} \text{ and } \tilde{\mathbf{W}}^{(k)}) \\
&= [n_i^2 (2N - 1)].
\end{aligned} \tag{A.11}$$

The work units, as shown in Chapter 5, for one non-linear iteration for the HROM solution is:

$$\begin{aligned}
W_{\text{non lin}} &= W_{\text{residual}} + W_{\text{test}} + W_{\text{assemble}} + (\text{units to solve for solution update}) \\
&= [n_{e_i} (d_e + 2n_i d_e + n_i)] + [2n_{e_i} d_e d_e^+ + 2d_e d_e^+ n_{e_i} n_i] + [n_i^2 (2N - 1)] + (n_i^3).
\end{aligned} \tag{A.12}$$

A.3 DWR Error ϵ_r Work Units

Prior to incorporating hyperreduction, the adjoint problem for the second DWR error indicator between the coarse and fine ROM is:

$$\left[\mathbf{V}_h^T \frac{\partial \mathbf{R}}{\partial \tilde{\mathbf{w}}_h} \Big|_{\tilde{\mathbf{w}}_H} \frac{\partial \mathbf{R}}{\partial \tilde{\mathbf{w}}_h} \Big|_{\tilde{\mathbf{w}}_H} \mathbf{V}_h \right]^T \hat{\psi}_h = - \left[\frac{\partial \mathcal{J}}{\partial \tilde{\mathbf{w}}_h} \Big|_{\tilde{\mathbf{w}}_H} \mathbf{V}_h \right]^T. \tag{A.13}$$

The cost to assemble the LHS of the adjoint problem is then:

$$\begin{aligned}
W_{\text{LHS adj}} &= (\text{FLOPs to multiply } \mathbf{V}_h \text{ and } \left. \frac{\partial \mathbf{R}}{\partial \tilde{\mathbf{w}}_h} \right|_{\tilde{\mathbf{w}}_H}) + \\
&\quad (\text{FLOPs to multiply the matrix resulting from the previous step and } \left. \frac{\partial \mathbf{R}}{\partial \tilde{\mathbf{w}}_h} \right|_{\tilde{\mathbf{w}}_H}^T + \\
&\quad (\text{FLOPs to multiply the matrix resulting from the previous step and } \mathbf{V}_h^T) \\
&= (\text{FLOPs to multiply matrices of dimensions } (N \times N) \text{ and } (N \times n_i)) + \\
&\quad (\text{FLOPs to multiply matrices of dimensions } (N \times N) \text{ and } (N \times n_i)) + \\
&\quad (\text{FLOPs to multiply matrices of dimensions } (n_i \times N) \text{ and } (N \times n_i)) \\
&= [Nn_i(2N - 1)] + [Nn_i(2N - 1)] + [n_i^2(2N - 1)] \\
&= (2Nn_i + n_i^2)(2N - 1).
\end{aligned} \tag{A.14}$$

The cost to assemble the RHS of the adjoint problem is:

$$\begin{aligned}
W_{\text{RHS adj}} &= (\text{FLOPs to multiply } \mathbf{V}_h \text{ and } \left. \frac{\partial \mathcal{J}}{\partial \tilde{\mathbf{w}}_h} \right|_{\tilde{\mathbf{w}}_H}) \\
&= (\text{FLOPs to multiply matrices of dimensions } (1 \times N) \text{ and } (N \times n_i)) \\
&= [n_i(2N - 1)].
\end{aligned} \tag{A.15}$$

The error estimate found using the adjoint from above is then:

$$\epsilon_r = \hat{\psi}_h^T \hat{\mathbf{R}}_h(\tilde{\mathbf{w}}_H) = \hat{\psi}_h^T \mathbf{V}_h^T \left. \frac{\partial \mathbf{R}}{\partial \tilde{\mathbf{w}}_h} \right|_{\tilde{\mathbf{w}}_H}^T \mathbf{R}_h(\tilde{\mathbf{w}}_H). \tag{A.16}$$

The work units required to solve for ϵ_r can be broken down into:

$$\begin{aligned}
W_{\epsilon_r} &= (\text{FLOPs to multiply } \mathbf{R}_h(\tilde{\mathbf{w}}_H) \text{ and } \left. \frac{\partial \mathbf{R}}{\partial \tilde{\mathbf{w}}_h} \right|_{\tilde{\mathbf{w}}_H}^T) + \\
&\quad (\text{FLOPs to multiply the matrix resulting from the previous step and } \mathbf{V}_h^T) + \\
&\quad (\text{FLOPs to multiply the matrix resulting from the previous step and } \hat{\psi}_h^T) \\
&= (\text{FLOPs to multiply matrices of dimensions } (N \times N) \text{ and } (N \times 1)) + \\
&\quad (\text{FLOPs to multiply matrices of dimensions } (n_i \times N) \text{ and } (N \times 1)) + \\
&\quad (\text{FLOPs to multiply matrices of dimensions } (1 \times n_i) \text{ and } (n_i \times 1)) \\
&= [N(2N - 1)] + [n_i(2N - 1)] + (2n_i - 1) \\
&= (N + n_i)(2N - 1) + (2n_i - 1).
\end{aligned} \tag{A.17}$$

Recalling the result in Chapter 5, the work units associated with solving the DWR at one ROM point is:

$$\begin{aligned}
W_{\text{DWR error}} &= (\text{units to evaluate the residual}) + \\
&\quad (\text{units to evaluate the Jacobian}) + \\
&\quad (\text{units to evaluate the derivative of the functional w.r.t. } \tilde{\mathbf{w}}) + \\
&\quad W_{\text{LHS adj}} + W_{\text{RHS adj}} + \\
&\quad (\text{units to solve for the adjoint}) + W_{\epsilon_r} \\
&= (N) + (N^2) + (N) + [(2Nn_i + n_i^2 + n_i)(2N - 1)] + \\
&\quad (n_i^3) + [(N + n_i)(2N - 1) + (2n_i - 1)].
\end{aligned} \tag{A.18}$$

A.4 Hyperreduced DWR Error ϵ_r Work Units

As discussed in Chapter 5, the updated error indicator with hyperreduction can be found using the following adjoint:

$$\left[\left\{ \sum_{e \in \tilde{\mathcal{E}}} \xi_e \mathbf{W}_h^T \mathbf{L}_e^T \frac{\partial \mathbf{R}_e}{\partial \tilde{\mathbf{w}}_h} \Big|_{\tilde{\mathbf{w}}_H} \right\} \mathbf{V}_h \right]^T \tilde{\psi}_h = - \left[\frac{\partial \mathcal{J}}{\partial \tilde{\mathbf{w}}_h} \Big|_{\tilde{\mathbf{w}}_H} \mathbf{V}_h \right]^T \quad (\text{A.19})$$

The cost to assemble the LHS of the adjoint problem is then:

$$\begin{aligned} W_{\text{LHS adj}} &= (\text{FLOPs to evaluate quantity in the curly brackets}) + \\ &\quad (\text{FLOPs to multiply the matrix resulting from the previous step and } \mathbf{V}_h) \\ &= (\text{FLOPs to evaluate quantity in the curly brackets}) + \\ &\quad (\text{FLOPs to multiply matrices of dimensions } (n_i \times N) \text{ and } (N \times n_i)) \\ &= [n_{e_i}(d_e n_i + n_i)] + [n_i^2(2N - 1)]. \end{aligned} \quad (\text{A.20})$$

The cost to assemble the RHS of the adjoint problem is:

$$\begin{aligned} W_{\text{RHS adj}} &= (\text{FLOPs to multiply } \mathbf{V}_h \text{ and } \frac{\partial \mathcal{J}}{\partial \tilde{\mathbf{w}}_h} \Big|_{\tilde{\mathbf{w}}_H}) \\ &= (\text{FLOPs to multiply matrices of dimensions } (1 \times N) \text{ and } (N \times n_i)) \\ &= [n_i(2N - 1)]. \end{aligned} \quad (\text{A.21})$$

The error indicator can then be found using:

$$\epsilon_r = -\tilde{\psi}_h^T \tilde{\mathbf{R}}_h(\tilde{\mathbf{w}}_H). \quad (\text{A.22})$$

The work units required to solve for ϵ_r can be broken down into:

$$\begin{aligned}
W_{\epsilon_r} &= (\text{FLOPs to multiply } \tilde{\mathbf{R}}_h(\tilde{\mathbf{w}}_H) \text{ and } \tilde{\psi}_h^T) \\
&= (\text{FLOPs to multiply matrices of dimensions } (1 \times n_i) \text{ and } (n_i \times 1)) \\
&= (2n_i - 1).
\end{aligned} \tag{A.23}$$

Recalling the result in Chapter 5, the work units associated with solving the hyperreduced DWR at one ROM point is:

$$\begin{aligned}
W_{\text{DWR error}} &= (\text{units to evaluate the hyperreduced residual}) + \\
&\quad (\text{units to evaluate the hyperreduced test basis}) + \\
&\quad (\text{units to evaluate the derivative of the functional w.r.t. } \tilde{\mathbf{w}}) + \\
&\quad W_{\text{LHS adj}} + W_{\text{RHS adj}} + \\
&\quad (\text{units to solve for the adjoint}) + W_{\epsilon_r} \\
&= [n_{e_i}(d_e + 2n_i d_e + n_i)] + [n_{e_i} d_e d_e^+ + 2d_e d_e^+ n_{e_i} n_i] + (N) \\
&\quad + [n_{e_i}(d_e n_i + n_i) + (n_i^2)(2N - 1) + n_i(2N - 1)] + (n_i^3) + (2n_i - 1).
\end{aligned} \tag{A.24}$$

Bibliography

- [1] R. Swischuk, L. Mainini, B. Peherstorfer, and K. Willcox, “Projection-based model reduction: Formulations for physics-based machine learning,” *Computers & Fluids*, vol. 179, pp. 704–717, 2019.
- [2] T. Bui-Thanh, M. Damodaran, and K. Willcox, “Proper orthogonal decomposition extensions for parametric applications in compressible aerodynamics,” *21st AIAA Applied Aerodynamics Conference*, p. 4213, 2003.
- [3] T. Franz, R. Zimmermann, S. Görtz, and N. Karcher, “Interpolation-based reduced-order modelling for steady transonic flows via manifold learning,” *International Journal of Computational Fluid Dynamics*, vol. 28, pp. 106–121, 2014.
- [4] T. Franz, “Reduced-order modeling for steady transonic flows via manifold learning,” Ph.D. dissertation, Deutsches Zentrum für Luft-und Raumfahrt, 2016.
- [5] J. D. Martin and T. W. Simpson, “Use of kriging models to approximate deterministic computer models,” *AIAA journal*, vol. 43, no. 4, pp. 853–863, 2005.
- [6] J. Laurenceau and P. Sagaut, “Building efficient response surfaces of aerodynamic functions with kriging and cokriging,” *AIAA journal*, vol. 46, no. 2, pp. 498–507, 2008.

- [7] S. A. Renganathan, R. Maulik, and V. Rao, “Machine learning for nonintrusive model order reduction of the parametric inviscid transonic flow past an airfoil,” *Physics of Fluids*, vol. 32, no. 4, p. 047110, 2020.
- [8] K. Lee and K. T. Carlberg, “Model reduction of dynamical systems on nonlinear manifolds using deep convolutional autoencoders,” *Journal of Computational Physics*, vol. 404, p. 108973, 2020.
- [9] R. Maulik, B. Lusch, and P. Balaprakash, “Reduced-order modeling of advection-dominated systems with recurrent neural networks and convolutional autoencoders,” *Physics of Fluids*, vol. 33, no. 3, p. 037106, 2021.
- [10] P. Holmes, J. L. Lumley, G. Berkooz, and C. W. Rowley, *Turbulence, coherent structures, dynamical systems and symmetry*. Cambridge university press, 2012.
- [11] L. Sirovich, “Turbulence and the dynamics of coherent structures,” *Quarterly of Applied mathematics*, vol. 45, no. 1-3, pp. 583–590, 1987.
- [12] C. W. Rowley, T. Colonius, and R. M. Murray, “Model reduction for compressible flows using pod and galerkin projection,” *Physica D: Nonlinear Phenomena*, vol. 189, no. 1-2, pp. 115–129, 2004.
- [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*, vol. 86, no. 2, pp. 155–181, 2011.
- [14] K. Carlberg, M. Barone, and H. Antil, “Galerkin v. least-squares petrov–galerkin projection in nonlinear model reduction,” *Journal of Computational Physics*, vol. 330, pp. 693–734, 2017.

- [15] S. Grimberg, C. Farhat, R. Tezaur, and C. Bou-Mosleh, “Mesh sampling and weighting for the hyperreduction of nonlinear petrov–galerkin reduced-order models with local reduced-order bases,” *International Journal for Numerical Methods in Engineering*, vol. 122, no. 7, pp. 1846–1874, 2021.
- [16] D. Blais, “Goal-oriented adaptive sampling for projection-based reduced-order models,” Ph.D. dissertation, McGill University, 2022.
- [17] S. Grimberg, C. Farhat, and N. Youkilis, “On the stability of projection-based model order reduction for convection-dominated laminar and turbulent flows,” *Journal of Computational Physics*, vol. 419, p. 109681, 2020.
- [18] C. Farhat, S. Grimberg, A. Manzoni, and A. Quarteroni, *5 Computational bottlenecks for PROMs: precomputation and hyperreduction*. Berlin, Boston: De Gruyter, 2021, pp. 181–244.
- [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*, vol. 102, no. 5, pp. 1077–1110, 2015.
- [20] 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 Mathématique*, vol. 339, no. 9, pp. 667–672, 2004.
- [21] Grepl, Martin A., Maday, Yvon, Nguyen, Ngoc C., and Patera, Anthony T., “Efficient reduced-basis treatment of nonaffine and nonlinear partial differential equations,” *ESAIM: M2AN*, vol. 41, no. 3, pp. 575–605, 2007.

- [22] S. Chaturantabut and D. Sorensen, “Nonlinear model reduction via discrete empirical interpolation,” *SIAM Journal on Scientific Computing*, vol. 32, pp. 2737–2764, 2010.
- [23] 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*, vol. 98, no. 9, pp. 625–662, 2014.
- [24] 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*. AIAA, 2003, p. 3847.
- [25] K. Veroy and A. T. Patera, “Certified real-time solution of the parametrized steady incompressible navier–stokes equations: rigorous reduced-basis a posteriori error bounds,” *International Journal for Numerical Methods in Fluids*, vol. 47, no. 8-9, pp. 773–788, 2005.
- [26] T. Bui Thanh, K. Willcox, and O. Ghattas, “Model reduction for large-scale systems with high-dimensional parametric input space,” *SIAM Journal on Scientific Computing*, vol. 30, no. 6, pp. 3270–3288, 2008.
- [27] A. Paul-Dubois-Taine and D. Amsallem, “An adaptive and efficient greedy procedure for the optimal training of parametric reduced-order models,” *International Journal for Numerical Methods in Engineering*, vol. 102, no. 5, pp. 1262–1292, 2015.
- [28] P. J. Blonigan, F. Rizzi, M. Howard, J. A. Fike, and K. T. Carlberg, “Model reduction for steady hypersonic aerodynamics via conservative manifold least-squares petrov–galerkin projection,” *AIAA Journal*, vol. 59, no. 4, pp. 1296–1312, 2021.

- [29] M. K. Sleeman and M. Yano, “Goal-oriented model reduction for parametrized time-dependent nonlinear partial differential equations,” *Computer Methods in Applied Mechanics and Engineering*, vol. 388, p. 114206, 2022.
- [30] M. A. Park, “Adjoint-based, three-dimensional error prediction and grid adaptation,” *AIAA Journal*, vol. 42, no. 9, pp. 1854–1862, 2004.
- [31] K. J. Fidkowski and D. L. Darmofal, “Review of output-based error estimation and mesh adaptation in computational fluid dynamics,” *AIAA Journal*, vol. 49, no. 4, p. 673 – 694, 2011.
- [32] M. Yano and D. L. Darmofal, “An optimization-based framework for anisotropic simplex mesh adaptation,” *Journal of Computational Physics*, vol. 231, no. 22, p. 7626 – 7649, 2012.
- [33] N. Ringue and S. Nadarajah, “An optimization-based framework for anisotropic hp-adaptation of high-order discretizations,” *Journal of Computational Physics*, vol. 375, pp. 589–618, 2018.
- [34] M. Meyer and H. G. Matthies, “Efficient model reduction in non-linear dynamics using the karhunen-loeve expansion and dual-weighted-residual methods,” *Computational Mechanics*, vol. 31, no. 1, pp. 179–191, 2003.
- [35] G. Collins, K. Fidkowski, and C. E. Cesnik, “Output error estimation for projection-based reduced models,” in *AIAA Aviation 2019 Forum*. AIAA, 2019, p. 3528.
- [36] D. Blais and S. Nadarajah, “Goal-oriented adaptive sampling procedure for projection-based reduced-order models for aerodynamic flows,” *AIAA SCITECH 2023 Forum*, 2023.

- [37] —, “Goal-oriented adaptive sampling for projection-based reduced-order models,” *AIAA Aviation 2022 Forum*, 2022.
- [38] D. Shi-Dong, S. Nadarajah, J. Brillon, D. Blais, K. MacLean, P. Thakur, C. Pethrick, A. Cicchino, M. Tatarelli, and C. Biondic, “dougshidong/philip: Philip version 2.0.0,” May 2022.
- [39] J. S. Hesthaven and T. Warburton, *Nodal discontinuous Galerkin methods: algorithms, analysis, and applications*. Springer Science & Business Media, 2007.
- [40] R. Tezaur, F. As’ad, and C. Farhat, “Robust and globally efficient reduction of parametric, highly nonlinear computational models and real time online performance,” *Computer Methods in Applied Mechanics and Engineering*, vol. 399, p. 115392, 2022.
- [41] M. J. Zahr, “Adaptive model reduction to accelerate optimization problems governed by partial differential equations,” Ph.D. dissertation, Stanford University, 2016.
- [42] P. Benner, W. Schilders, S. Grivet-Talocia, A. Quarteroni, G. Rozza, and L. Miguel Silveira, *Model order reduction: volume 3 applications*. De Gruyter, 2020.
- [43] K. M. Washabaugh, “Faster fidelity for better design: A scalable model order reduction framework for steady aerodynamic design applications,” Ph.D. dissertation, Stanford University, 2016.
- [44] K. M. Washabaugh, M. J. Zahr, and C. Farhat, “On the use of discrete nonlinear reduced-order models for the prediction of steady-state flows past parametrically deformed complex geometries,” *54th AIAA Aerospace Sciences Meeting*, p. 1814, 2016.
- [45] S. J. Grimberg and C. Farhat, “Hyperreduction of cfd models of turbulent flows using a machine learning approach,” *AIAA Scitech 2020 Forum*, 2020.

- [46] D. Ryckelynck, “A priori hyperreduction method: an adaptive approach,” *Journal of Computational Physics*, vol. 202, no. 1, pp. 346–366, 2005.
- [47] R. Everson and L. Sirovich, “Karhunen–loève procedure for gappy data,” *JOSA A*, vol. 12, 08 1995.
- [48] D. Wirtz, D. C. Sorensen, and B. Haasdonk, “A posteriori error estimation for deim reduced nonlinear dynamical systems,” *SIAM Journal on Scientific Computing*, vol. 36, no. 2, pp. A311–A338, 2014.
- [49] P. Astrid, S. Weiland, K. Willcox, and T. Backx, “Missing point estimation in models described by proper orthogonal decomposition,” *IEEE Transactions on Automatic Control*, vol. 53, no. 10, pp. 2237–2251, 2008.
- [50] 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*, vol. 242, pp. 623–647, 2013.
- [51] M. Yano, *6 Model reduction in computational aerodynamics*. Berlin, Boston: De Gruyter, 2021, pp. 201–236.
- [52] A. T. Patera and M. Yano, “An lp empirical quadrature procedure for parametrized functions,” *Comptes Rendus Mathématique*, vol. 355, no. 11, pp. 1161–1167, 2017.
- [53] 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*, vol. 344, pp. 1104–1123, 2019.

- [54] J. Hernández, M. Caicedo, and A. Ferrer, “Dimensional hyper-reduction of nonlinear finite element models via empirical cubature,” *Computer Methods in Applied Mechanics and Engineering*, vol. 313, pp. 687–722, 2017.
- [55] M. Yano, “Goal-oriented model reduction of parametrized nonlinear partial differential equations: Application to aerodynamics,” *International Journal for Numerical Methods in Engineering*, vol. 121, no. 23, pp. 5200–5226, 2020.
- [56] T. Chapman, P. Avery, P. Collins, and C. Farhat, “Accelerated mesh sampling for the hyper reduction of nonlinear computational models,” *International Journal for Numerical Methods in Engineering*, vol. 109, 07 2016.
- [57] M. Rewienski, “A trajectory piecewise-linear approach to model order reduction of nonlinear dynamical systems,” Ph.D. dissertation, Massachusetts Institute of Technology, 2003.
- [58] M. J. Zahr and C. Farhat, “Progressive construction of a parametric reduced-order model for pde-constrained optimization,” *International Journal for Numerical Methods in Engineering*, vol. 102, no. 5, pp. 1111–1135, 2015.
- [59] R. Tibshirani, “Lecture notes in convex optimization,” November 5 2015. [Online]. Available: https://www.stat.cmu.edu/~ryantibs/convexopt-F18/scribes/Lecture_19.pdf