# Unsteady Adjoint Analysis for Output Sensitivity and Mesh Adaptation

by

Joshua Ambre Krakos

B.S., Iowa State University (2001)
M.S., Iowa State University (2003)

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

Doctor of Philosophy

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2012

Author . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
Department of Aeronautics and Astronautics
June 18, 2012

Certified by . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
David L. Darmofal
Professor of Aeronautics and Astronautics
Thesis Supervisor

Certified by . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
Mark Drela
Professor of Aeronautics and Astronautics
Thesis Committee

Certified by . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
Qiqi Wang
Assistant Professor of Aeronautics and Astronautics
Thesis Committee

Accepted by . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
Eytan Modiano
Professor of Aeronautics and Astronautics
Chairman, Committee on Graduate Students

**Unsteady Adjoint Analysis for Output Sensitivity and Mesh Adaptation**
by
Joshua Ambre Krakos

Submitted to the Department of Aeronautics and Astronautics
on June 18, 2012, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

## Abstract

Adjoint analysis in computational fluid dynamics (CFD) has been applied to design optimization and mesh adaptation, but due to the relative expense of unsteady analysis these applications have predominantly been for steady problems. As the use of adjoint methods continues to becomes more prevalent, more problems are encountered for which steady analysis may not be appropriate. This thesis examines three aspects of unsteady adjoint analysis.

First, this work investigates problems exhibiting small-scale output unsteadiness when solved with time-inaccurate iterative solvers. It is demonstrated that unconverged steady flow calculations, even with small output unsteadiness, can lead to significant variability in the estimated output sensitivity due to the arbitrary choice of unconverged state upon which the linearization is performed. Further, time-inaccurate "unsteady" iterative solutions depend on the iterative method used and may exhibit different output and output sensitivity compared to the steady flow or time-accurate unsteady flow.

With the motivation for unsteady simulation established, output and output parameter sensitivities of periodic unsteady problems are sought using finite-time averaging. Periodic outputs computed over a finite time span are found to converge slowly and output sensitivities may be nonconvergent when the period of oscillation is a function of the parameter of interest. A theoretical basis for this lack of convergence is identified and output windowing is proposed to alleviate its effect. Output windowing is shown to enable the accurate computation of periodic output sensitivities and to decrease simulation time to compute periodic outputs and sensitivities.

Finally, a spatial mesh adaptation approach is developed for unsteady wake problems and other problems with smooth and persistent regions of unsteadiness. For this class of problems, a higher-order discretization coupled with a single spatial mesh approach is appropriate to capture both steady and unsteady regions. The method proposed herein extends the anisotropic, output-based Mesh Optimization via Error Sampling and Synthesis (MOESS) algorithm of Yano and Darmofal to optimize the spatial mesh driven by an unsteady flow field.

Thesis Supervisor: David L. Darmofal
Title: Professor of Aeronautics and Astronautics

## Acknowledgments

I would like to express my gratitude to the many people who have made this thesis possible. First, I would like to thank my advisor, Professor David Darmofal, for giving me the opportunity to work with him. I am grateful for his guidance, inspiration and encouragement; and his willingness to let me do most of this doctorate from 1200 miles away. In addition, I would like to thank my committee members, Professors Mark Drela and Qiqi Wang, for their criticism and feedback, which led to many improvements in my research and this thesis. I am appreciative of my readers Dr. Mori Mani and Professor Steven Hall for providing comments and suggestions on this thesis. This work would not have been possible without the efforts of the entire ProjectX team past and present with whom I have worked (Julie Andren, Garrett Barter, Laslo Diosady, Krzysztof Fidkowski, Bob Haimes, Eric Liu, JM Modisette, Todd Oliver, Mike Park, Huafei Sun, David Walfisch, & Masa Yano). I never would have survived the qualifying exams without the friendship of Jeff Chambers & Alejandra Uranga.

When I said I wanted to go back to school, it was Mori Mani at The Boeing Company who insisted that I should apply to MIT, and supported me when I was accepted and wanted to take a two year leave of absence. That support has not flagged in the years that I have continued this work while also employed full-time. I would be remiss to not thank my manager, Bill Bower, for his support and my coworkers and friends in the Boeing BR&T CFD group in Saint Louis: Deric Babcock, Geoff Behrens, Andrew Cary, Andy Dorgan, Mark Fisher, Katie Fox, John Ladd, Matthew Lakebrink, Brian Lambert, Mori Mani, Todd Michal, and Chad Winckler. I consider myself lucky to work with such a professional and capable group of people.

I would like to thank my parents, Garry and Kay, for their love and support. They instilled in me the importance of education at an early age, an idea I have yet to shake. Thanks to my brothers, Andrew and Daniel, for their constant encouragement. My wife, Kyra, supported me through every day of my doctorate, and now gets to hold me to all the times I said "when I'm done with school, we'll..." When I began this work, I could not have imagined trying to finish school with children of my own. Now I cannot imagine life without them: Jack, the son that I got to choose; Ilse, our ray of sunshine; and Kate, who is inherently funny.

Finally, I would like to acknowledge the financial support I have received throughout my doctoral program from The Boeing Company Learning Together Program.

# Contents

# List of Figures

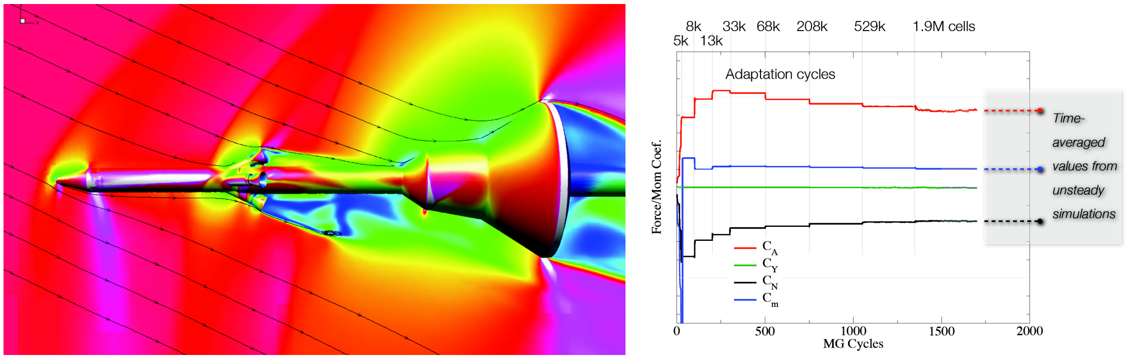
# Chapter 1

# Introduction

## 1.1 Motivation

The accurate calculation of an output of engineering interest may depend on spatial regions of seeming uninterest, especially in problems with significant disturbance propagation. The difficulty of determining *a priori* the region of sensitivity for a given output, or in fact by *a posteriori* inspection, motivates a method of automatically determining appropriate spatial mesh refinement for a given error tolerance in the output. Adjoint analysis addresses the problem of output-based error estimation in an automated and rigorous fashion. Furthermore, in aerodynamic design optimization the sensitivity of an output with respect to design inputs is a key component in improving toward an optimal solution. In the case where the number of outputs being optimized is small compared to the number of control input parameters, e.g. shape optimization, the adjoint provides a computationally superior means of computing the required output parameter sensitivities.

In the presence of small-scale unsteadiness, that is, outputs of interest exhibit unsteadiness that is small in magnitude compared to the mean output, it may be valid to assume that because the unsteadiness of the flow solution has little apparent effect on the output of interest, that the steady, or single-point, adjoint will provide sufficiently accurate sensitivity information. However, the adjoint for such cases may be difficult to compute. Campobasso and Giles[17] found that for unsteadiness based in the linear instability of the flow an iterative calculation of the steady adjoint equation may not converge. Nemec et al.[76] note this difficulty when their steady algorithm is applied to the inviscid simulation of a complex launch abort vehicle, Figure (1-1a). For particular conditions, the solution progresses from an initial mesh through several adaptation cycles until for a mesh with 1.9 million elements the mesh is finely resolved enough to capture some physical unsteadiness in the iterative solver, see Figure (1-1b). While this small-scale unsteadiness may have only a minor effect on the outputs of interest, the primal solution of the discrete equations does not converge to a steady solution. As with the case studied by Campobasso and Giles, in the primal these instabilities remain bounded due to the nonlinear stability of the flow, but when the flow is linearized about a single fixed point the adjoint equation exhibits a linear instability in the iterative scheme. The authors are only able to partially converge an adjoint solution in these cases, but the error estimate calculated from this solution varies significantly from previous mesh error estimates and is typically not usable for adapting to a new mesh. Recent appearances of small-scale unsteadiness in the literature include Park et al.[84] and Nielsen et al.[78].



(a) Launch Abort Vehicle geometry

(b) Convergence history of aerodynamic coefficients $C_A$, $C_Y$, $C_N$, and $C_m$ through several adaptation cycles

Figure 1-1: Nemec et al., LAV test case

Many unsteady problems equilibrate to periodic behavior. For these problems the sensitivity of periodic outputs to system parameters are often desired[45, 74, 99], and must be estimated from a finite time span or frequency domain calculation. Recently, Srinath and Mittal[108] performed adjoint-based shape optimization of a free-periodic unsteady NACA 0012 airfoil at $Re = 500$ using time-averaged outputs. They observed dependence of the robustness and accuracy of their method on the extent of the integration span, $M$. Wilkins *et al.*[123] investigated the calculation of periodic sensitivities from a finite time simulation of dynamical systems. They note the presence of unbounded terms in the calculated value of the sensitivity, $\mathcal{J}_s'$, as $M$ goes to infinity, a problem previously investigated by

Tomovic[112], among others. Wilkins also investigated the error induced by fixed boundary conditions of the unsteady problem (both initial and terminal).

Wilkins proposed a method to compute various parameter sensitivities by isolating the sensitivities of the period, the phase, and amplitude, then applied it to a mammalian circadian clock model with 73 states and 231 parameters, a small system when compared to the typical computational fluid dynamics (CFD) analysis. This motivates a method of computing periodic output sensitivities that is appropriate for unsteady CFD simulation and alleviates or eliminates the convergence difficulties when solving over a finite time span.

Over the last several decades, increased available computational power combined with improvements in numerical methods have made CFD an indispensable tool in aerospace design and analysis. Regardless, the ability of CFD solvers to reliably predict desired outputs in a robust and automated fashion is still limited. In general, mesh generation is a user-intensive process with the quality of the resulting mesh (and solutions computed thereon) heavily dependent on the previous experience of the user and their ability to predict mesh regions of importance. A work that investigates the inadequacy of this process was presented by Mavriplis[64], following the third AIAA Drag Prediction Workshop (DPW3)[1, 115]. Using a transonic wing-only geometry, Mavriplis demonstrated the dependence on the initial mesh by comparing families of refined meshes based on independent expert-generated initial meshes. The results, in Figure (1-2), show a difference between the two families of approximately four drag counts on the finest meshes. Using the approximations of Vassberg[114], a single drag count could account for as many as eight passengers on a long-range flight, so a four count discrepancy is significant.

Unsteady simulations represent a significant cost increase over steady simulations on the same geometry, possibly orders of magnitude increase in time and computational effort. For an unsteady simulation, the problem of manual mesh generation is compounded: spatial regions of importance can vary with time, and different time spans can have differing importance with regard to the output of interest. Any mesh adaptation for an unsteady problem must take this variation into account to build an appropriate mesh. The application of higher-order spatial discretizations and automated output-based mesh adaptation to unsteady problems gives the opportunity for large potential improvement in solution accuracy and turnaround.

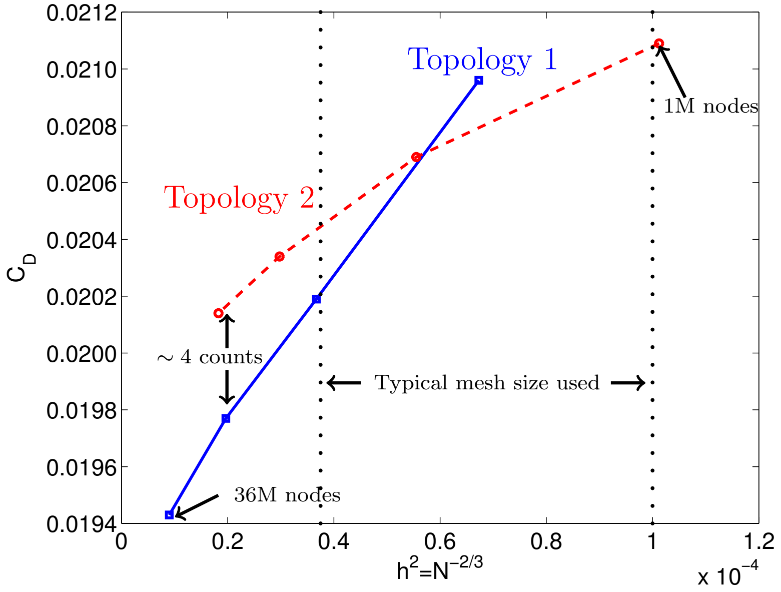


Figure 1-2: Computed drag convergence for a wing-only configuration at $M_\infty = 0.76$, $\alpha = 0.5°$, and $Re = 5 \times 10^6$ with global mesh refinement. Taken from Mavriplis[64]. Convergence of drag is plotted for the refinement of two mesh families on the same geometry.

## 1.2 Objectives

The objectives of this work are threefold. First, to investigate problems exhibiting small-scale unsteadiness when solved with time-inaccurate iterative solvers and characterize the effect of this unsteadiness on the accuracy of the flow solution and adjoint-based parameter sensitivities. Second, to investigate the calculation of time-average outputs and output sensitivities for periodic problems. Outputs computed using finite-time averaging are slow to converge and output sensitivities may be nonconvergent when the period of oscillation is sensitive to the parameter of interest. This work identifies the root cause of this slowed convergence and presents a method to alleviate its effect. Finally, unsteady CFD simulation is resource-intensive, and generation of an efficient and appropriate mesh is even more problematic than for steady simulations. The third objective of this work is to develop a spatial mesh adaptation approach for unsteady wake problems and other problems with smooth and persistent regions of unsteadiness. For this class of problems, a higher-order discretization coupled with a single spatial mesh approach is appropriate to capture both the steady and unsteady regions of the flow. The method proposed herein extends the anisotropic, output-based adaptive algorithm of Yano[125] to optimize the spatial mesh driven by an unsteady flow field.

## 1.3 Background

With its roots in the calculus of variations, adjoint analysis has long been used in optimal control theory, as a reference see Bryson and Ho[16], and weather modeling[20, 24, 41, 110]. Its use within the context of aerodynamics design is more recent and has been applied to design optimization[39, 48, 77] and to error estimation[8, 13, 25, 37, 44, 88, 116] enabling automated error control through mesh adaptation. In aerodynamic design optimization, the sensitivity of an objective function with respect to system parameters is an essential component in improving toward an optimal solution. In the case where the number of inputs is large compared to the number of outputs, the adjoint can be computationally superior to other methods, providing a rigorous and efficient means of computing the required output parameter sensitivities. Furthermore, adjoint-based error estimation addresses the difficulty of determining, either by *a priori* or *a posteriori* means, regions of mesh sensitivity for a given output and allows for automated and rigorous mesh refinement algorithms. Due to the overall cost of unsteady analysis of both the primal and the adjoint, adjoint analysis in aerodynamics has been predominantly for steady problems but has begun to make inroads into unsteady applications[62, 65, 71, 78, 97].

### 1.3.1 Small-Scale Unsteadiness

As the use of adjoint methods continues to increase within aerodynamics design, more problems are encountered for which a steady solution may not be appropriate or may not even exist, e.g., problems with regions of separation. For complex geometries and equation sets, the necessity of a steady state solution places what may be an unrealistic robustness requirement on the primal flow solver[80].

Campobasso and Giles[17] investigated the effects of flow instabilities in the linear analysis of turbomachinery aeroelasticity. The authors found that the fixed point iteration they used to solve the linearized Navier-Stokes equations was not stable for cases where the nonlinear background flow calculation did not converge to a steady solution but instead reached a small-amplitude limit cycle. While this unsteadiness in the primal nonlinear problem was small, the authors showed that the linearized system about a typical iterate of the unsteady solution had unstable eigenvalues. As a result, the iterative algorithm diverged when applied to the primal problem linearized about this state. These unstable modes are salient to the adjoint because the eigenvalues of the adjoint problem are the same as those of the linearized primal problem for the same iterative solver[38].

Nemec, Aftosmis, and Wintzer[76] note a similar difficulty in an adjoint-based adaptive, inviscid simulation of the flow about a complex launch abort vehicle. In their case, unsteadiness is encountered as the mesh is adapted and wakes are refined. While this unsteadiness has only a minor effect on the outputs of interest, the primal equations do not converge to a steady solution and the authors are unable to converge the adjoint about an iterate of the unsteady primal solution. The adjoint-based error estimation cannot be carried out and the adaptive process breaks down.

In the event of this small-scale unsteadiness, Campobasso and Giles proposed stabilizing the iterative scheme for the linearized equations utilizing GMRES[17, 100] or the Recursive Projection Method (RPM)[18, 104]. Both methods strengthen the linear solver, effectively making the stability of the linear solver less dependent on outlier eigenvalues of the linearized system. With these approaches, the linearized equations can be solved and the sensitivity analysis can be performed about an unconverged iterate from the steady nonlinear primal solver. However, parameter sensitivities and error estimates based on the adjoint require that the primal problem be satisfied, and applying steady adjoint methods to unconverged solution iterates introduces an error contribution from that nonconvergence, even when the output is relatively unaffected. As a result, the parameter sensitivity or error estimate computed from the adjoint at an arbitrary state in an unconverged iterative solution method will be dependent on the particular state at which the adjoint is computed.

Even if the nonlinear primal solver can be strengthened such that steady solutions can be achieved, a steady solution still does not capture or reflect any physical unsteadiness. That is, the steady flow solution may be quite different from the time average or snapshots of an unsteady solution. Although a steady sensitivity analysis may be possible, as the physical unsteadiness in the flow grows in magnitude relative to the mean or stationary point flowfields the resulting sensitivity may not be physically relevant (assuming that the unsteady solution is more likely to occur in practice), leading to erroneous or nonconvergent optimization.

Rather than calculate a steady primal and adjoint solution, a time-dependent flow and adjoint analysis can be performed. The unsteady adjoint has the benefit that the result is applicable for both small- and large-scale unsteady behavior. However, a straightforward implementation of the unsteady adjoint requires the storage of the entire primal solution for the time period of interest. This storage requirement has led to checkpointing methods[40, 121], trading some of the solution storage overhead for additional recomputation but without fundamentally changing the underlying algorithm. Despite its cost, the unsteady adjoint has seen increasing interest in CFD in the last decade. Nadarajah and Jameson[71] focus on aerodynamics shape optimization in the context of finite volume formulations. Using both the continuous and discrete unsteady adjoint equations, the authors derive the sensitivity to a shape parameter, then use a steepest descent method to iteratively improve the output based on that shape parameter. Each iteration of the optimization process requires a complete forward time solution of the flow, as well as a complete backward time solution of the adjoint. Mani and Mavriplis[62] and Mavriplis[65] investigated an unsteady discrete adjoint formulation for output matching and minimization applied to two and three-dimensional flow problems. Their method proved effective when applied to pressure distribution, load matching, and drag minimization on a sinusoidally pitching airfoil. Additionally, Rumpfkeil and Zingg[97] applied the discrete adjoint to aeroacoustic shape design.

Using the viscous, subsonic flow about a NACA 0012 airfoil at moderate angle of attack as a model problem, this work investigates the variability in output sensitivities due to linearization about unconverged primal iterates. A time-accurate unsteady adjoint is then implemented, and the differences between the converged steady solution, unconverged primal iterates, and time-accurate unsteady behavior are demonstrated.

### 1.3.2 Sensitivity Analysis of Limit Cycle Oscillations

For unsteady problems that exhibit periodic behavior, or asymptote to a limit cycle oscillation, a common objective function to optimize is the time-average of some periodic output[45, 74, 99]. Let $g(t, \beta)$ be such a periodic function of time with dependence on a parameter $\beta$. We define the objective function, $\mathcal{J}$, as the time-average of $g$ over a period, $T(\beta)$:

$$\mathcal{J}(\beta) \equiv \frac{1}{T(\beta)} \int_0^{T(\beta)} g(t, \beta) dt. \tag{1.1}$$

We are interested in calculating $\mathcal{J}'(\beta)$, the derivative of $\mathcal{J}$ with respect to $\beta$. A reasonable approach to compute $\mathcal{J}'$ is to approximate the periodic output over a finite time span,

$$\mathcal{J}_s(\beta, M) \equiv \frac{1}{M} \int_0^M g(t, \beta) dt, \tag{1.2}$$

and compute the sensitivity of this approximate function. For problems where the period of oscillation does not depend on the parameter of interest, especially for forced-period problems, this approach can be effective for both time domain methods[62, 72] and frequency domain methods[23, 73, 111]. However, for problems whose period of oscillation depends on the parameter of interest, i.e. $T = T(\beta)$, the number of complete and fractional periods contained in $M$ varies with $\beta$. For a given $M$, this dependence results in variation of $\mathcal{J}_s$ with $\beta$ which the periodic output does not exhibit. This additional variation can lead to local extrema of $\mathcal{J}_s$. When $\mathcal{J}_s$ and $\mathcal{J}_s'$ are computed as part of an optimization process, these extrema can prevent convergence of the optimization algorithm. This discrepancy cannot be corrected by increasing the integration span: whereas $\lim_{M\to\infty} \mathcal{J}_s = \mathcal{J}$ (converging at $\mathcal{O}(M^{-1})$), in general $\lim_{M\to\infty} \mathcal{J}_s' \neq \mathcal{J}'$. Use of a finite time average in conjunction with an adjoint or tangent method (continuous or discrete) can accurately compute the sensitivity of $\mathcal{J}_s$, but not the sensitivity of $\mathcal{J}$. Additional consideration must be taken to ensure accurate outputs and sensitivities.

To enable the calculation of accurate periodic sensitivities, we propose a method that is a simple modification of an existing unsteady tangent or adjoint sensitivity implementation. Namely, in the computation of the time-average, we apply a time-dependent weighting window to the instantaneous output:

$$\mathcal{J}_w(\beta, M) \equiv \frac{1}{M} \int_0^M g(t, \beta)\, w(t/M)\, dt, \tag{1.3}$$

where $w(\tau)$ is a time-dependent weighting function with $w(\tau) = 0\, \forall \tau \notin (0, 1)$. Similar approaches have been used in the past to improve the output behavior[7, 54, 98] by alleviating phase dependence caused by the sharp endpoints of a straightforward time-average with $w = w_s = 1$, also known as a square or rectangular window. This work extends that approach to output windows specifically designed to improve the behavior of output sensitivity calculations.

### 1.3.3 Spatial Adaptation for Unsteady Wake Problems

As computational power increases, CFD is applied to increasingly complicated geometries resulting in more complex flowfields, and more problems are encountered that are inherently unsteady. Due to the complexities of unsteady flow both in space and time, automated output-based mesh adaptation is necessary for these problems to be efficiently addressed, and gives the opportunity for large potential improvement in solution accuracy for a given computational cost. Additionally, it has been found that mesh adaptation is critical to realizing the advantages of the higher-order discontinuous Galerkin (DG) spatial discretization, even for steady problems[128].

The spectrum of possible approaches for solving unsteady problems range from a single, all-time spatial mesh to a fully-coupled space-time discretization, with several variations in between. Work on space-time discretizations has been published since at least the late 1980s with Bar-Yoseph[4] applying the method to several hyperbolic model problems. When paired with unstructured meshing, space-time discretizations allow for anisotropic refinement in both space and time, allowing for mesh elements to track the evolution of the flow. For problems to which it can be applied, unstructured space-time discretizations can give spectacular results for problems with space-time anisotropy. Unfortunately, the approach has significant drawbacks in application that have yet to be overcome, chiefly that for an implicit solver the entire space-time domain becomes a coupled system. The coupled space-time system makes it difficult to take advantage of the hyperbolic nature of time, such as is done naturally in traditional time-stepping approaches. This results in having to solve the entire spatial and temporal domain simultaneously, with the associated computer hardware requirements. Additionally, application to three dimensional spatial problems yields a four dimensional coupled problem, for which a simplex (hyper-tetrahedral) mesh is difficult to generate, let alone to adapt. As of the publication of Rendall[92], there were no four-dimensional unstructured meshers available, though there is some recent preliminary work in that direction[70]. As such, recent work has been limited in demonstration to one or two spatial dimensions[92, 124, 125].

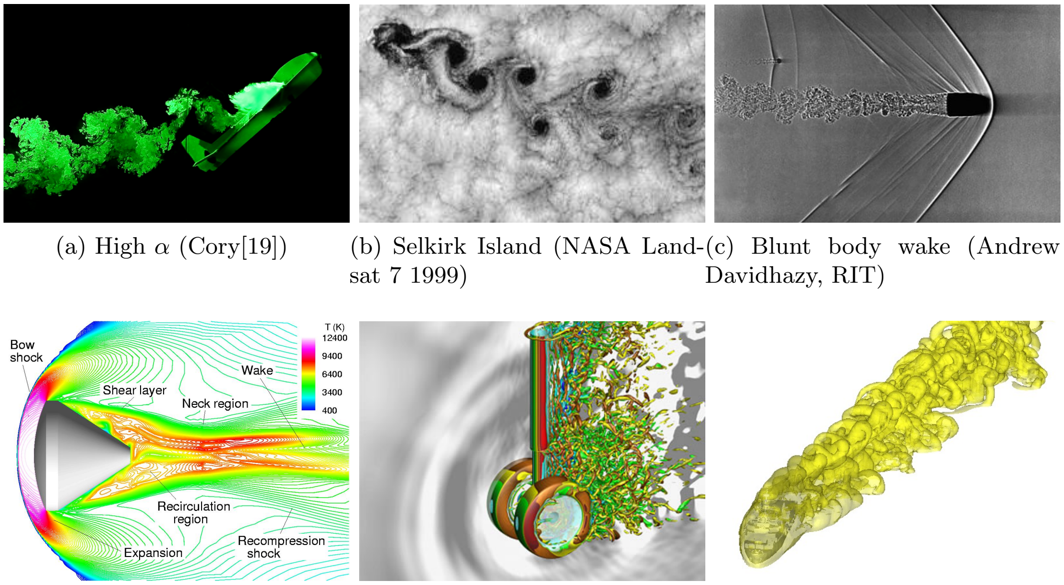
Because of the difficulties with the large, coupled space-time system and four-dimensional mesh generation, a popular alternative is the timeslab, or Roche, approach whereby the temporal domain is divided into subdomains. A different spatial mesh is associated with each timeslab, with the temporal dimension an extrusion of the spatial mesh. This allows the adaptation of the spatial meshes to target error sources only within their timespan of interest, improving efficiency over a single-mesh approach for problems with high degrees of spatial anisotropy in the unsteady region, such as acoustic waves[2, 3] or large regions that only see transient unsteadiness, e.g., wind gusts. This approach has been used with finite volume spatial discretizations[87, 91, 107], but recent work is predominantly with finite element (FE) discretizations, e.g. DG, in space[2, 3, 50, 113] as well as time[30, 58, 66, 67, 93, 102, 103]. The variational context of FE allows for a more straightforward error estimation formulation, enabling output-based adaptation. A notable exception of the finite element trend is the work of Mavriplis and collaborators[32, 61, 63], which applies the adjoint in a finite volume context to drive spatial and temporal adaptation, including nonuniform temporal refinement (timestep varies by spatial element) within a timeslab. This allows for significant gains in efficiency by controlling the spatial location of temporal refinement.

For problems with smooth and persistent regions of unsteadiness, the additional control afforded by coupled space-time and timeslab approaches may be unnecessary. For these problems, an unsteady, spatial adaptation approach using a single all-time mesh is applicable. This includes many unsteady wake problems, such as the examples in Figure (1-3). Anisotropic adaptation is necessary in the "steady" portions of the domain to capture nearly static features like boundary layers and shocks. However, assuming a sufficiently smooth flow, a higher order discretization is adequate in the wake.

An approach that targets only the spatial discretization error has none of the additional requirements on the temporal discretization that would arise from a temporal error estimation method. This allows an approach that is agnostic to, and independent of, the temporal discretization. The spatial error estimate can then be developed for a temporally continuous problem and applied to any temporal discretization. The only requirement on the temporal discretization scheme is that it can be used to integrate the spatial error estimate over the time span of interest.

### Anisotropic Adaptation

Of particular interest in the development of an unsteady spatial adaptation algorithm is the mechanism by which anisotropy is introduced into the mesh. A common tactic employed



(a) High $\alpha$ (Cory[19])

(b) Selkirk Island (NASA Landsat 7 1999)

(c) Blunt body wake (Andrew Davidhazy, RIT)

(d) Orion Reentry (Sinha[105])

(e) Landing gear (Sanders[101])

(f) LPD-4 (Whitehouse[122])

Figure 1-3: Unsteady wake examples

in steady-state adaptation is to size elements based on an output error estimate, then compute a desired anisotropy based on a solution Hessian[117], a process that was generalized to higher-order by Fidkowski & Darmofal[27, 29]. Leicht & Hartmann[52] introduced a variant based on the inter-element discontinuities of a DG formulation. An alternative approach, introduced by Formaggia et al.[33, 34] and refined by Loseille et al.[3, 57], still includes solution Hessians, but incorporates the anisotropy decision directly into the error minimization process. These methods presuppose that there is a scalar field whose behavior captures the desired anisotropy of the mesh, a supposition that is not met in general. Additionally, for an unsteady problem one is left with the question of how to combine the anisotropies prompted by the solution at each timestep. The anisotropy request for a timestep only relates the relative magnitudes of principal directions at that time, so metric averaging and intersection approaches cannot be applied. Combining Hessians by scaling them to a common scale arbitrarily equalizes the effect and importance of anisotropy at each timestep in the anisotropy decision process.

Competitive subdivision introduces anisotropy into the mesh using the error estimation framework without solution Hessians. First introduced by Houston et al.[47] for quadrilateral/hexahedral meshes, the method marks a fixed fraction of elements for refinement via an error estimate. Each marked element is the subdivided in each direction independently and the primal equations are resolved locally (with frozen neighbors). The algorithm then selects the anisotropic refinement or refinements in each element that provide the largest error decrease[47] or the largest error decrease per degree of freedom invested[35]. A similar approach which relies on splitting the error estimate into contributions in different coordinate directions was introduced by Richter[94], and extended to DG by Leicht and Hartmann[53]. Their approach has the benefit of not requiring local solves on anisotropically split meshes, but is limited to quad/hex meshes. Extensions of the approach to simplex meshes have met with limited success[83, 109].

### Mesh Optimization via Error Sampling and Synthesis (MOESS)

In this work, we present an unsteady extension to the Mesh Optimization via Error Sampling and Synthesis (MOESS) spatial adaptation algorithm developed by Yano & Darmofal[125–127]. This approach offers an application-nonspecific adaptation algorithm, within which mesh metric field (encoding both element size and anisotropy) changes are driven to minimize output error. MOESS has several desirable properties that together differentiate it from other available methods. The method

1. handles any discretization order, including mixed orders;
2. allows for arbitrarily-oriented anisotropic elements via simplex remeshing;
3. does not make any *a priori* assumption about the convergence behavior of the error, improving robustness when the mesh is under-resolved;
4. scales well in parallel, specifically, the error sampling is local to each element and scales perfectly in parallel;
5. drives both the sizing and the anisotropy decisions directly by the *a posteriori* error estimates, automatically capturing the behaviors of both the primal and dual solutions;
6. inherits the versatility of the adjoint-based error estimate, which exclusively governs adaptation decisions, allowing straightforward extension to different governing equations (e.g. Navier-Stokes, structural elasticity, Maxwell's).

Additionally, the error sampling process is extensible in a straightforward fashion to sampling over each timestep of an unsteady problem. Once the sampling is completed, the error model synthesis and optimization components of the algorithm are unaffected by this change.

## 1.4 Contributions

The primary contributions of this work are the following:

1. Demonstration that unconverged steady flow calculations, even with small output unsteadiness, can lead to significant variability in the estimated output sensitivity due to the arbitrary choice of unconverged state upon which the linearization is performed. Further, time-inaccurate "unsteady" iterative solutions depend on the iterative method used and may exhibit different output and output sensitivity compared to the steady flow or time-accurate unsteady flow.

2. Demonstration that finite time averaging causes an error in output sensitivity. An expression for this error was derived showing that the error scales with the dependence of the period on a parameter and does not decay with increasing range of integration. Additionally, due this nonconvergent error, finite time simulation introduces noise into the design space which can trap optimization in local extrema.

3. Development of two classes of windows to compute periodic outputs and output parameters using finite time simulations, analysis of their convergence properties, and demonstration of their application to output parameter sensitivity calculation.

4. Extension of the spatial mesh optimization framework of Yano & Darmofal to develop an anisotropic, output-based error estimation and spatial mesh adaptation approach for unsteady wake problems and other problems with smooth and persistent regions of unsteadiness.

This thesis is organized as follows: The adjoint and adjoint-based output parameter sensitivities are reviewed in Chapter 2, along with the DG spatial discretization. Chapter 3 investigates small-scale unsteadiness, including contribution 1, using a laminar NACA 0012 at moderate angle of attack as a model problem. Motivated by the results of Chapter 3, Chapter 4 discusses the need for and development of output windowing for problems with periodic behavior, including contributions 2 and 3. Chapter 5 discusses contribution 4, a new spatial mesh adaptation method for unsteady problems specifically targeted at wakes and other problems with smooth, persistent regions of unsteadiness. Finally, conclusions and suggestions for future work are given in Chapter 6.

# Chapter 2

# Primal Discretization & Adjoint Formulation

This chapter begins with the derivation of adjoint and adjoint-based output sensitivities for steady and unsteady nonlinear systems in Section 2.1. Section 2.2 introduces the spatial discretization of a partial differential equation (PDE), including finite dimensional adjoint and parameter sensitivities. The chapter concludes in Section 2.3 with a discussion of the general solution approach for steady and unsteady simulations.

## 2.1 Adjoint Formulation

### 2.1.1 Steady Adjoint

To develop the steady formulation, first introduce the steady nonlinear system for the primal problem

$$\boldsymbol{f}(\boldsymbol{u}_{ss}; \beta) = \boldsymbol{0},$$

where $\boldsymbol{f}(\cdot;\cdot) : \mathbb{R}^m \times \mathbb{R} \to \mathbb{R}^m$ is a nonlinear residual statement, $\boldsymbol{u}_{ss} \in \mathbb{R}^m$ is the primal state vector with $m$ components, and $\beta \in \mathbb{R}$ is an input parameter. Let $\mathcal{J}(\beta) = g(\boldsymbol{u}_{ss}; \beta)$ be a general nonlinear functional output of interest where $g(\cdot;\cdot) : \mathbb{R}^m \times \mathbb{R} \to \mathbb{R}$. The Lagrangian is then defined as, $\mathcal{L}(\cdot, \cdot; \cdot) : \mathbb{R}^m \times \mathbb{R}^m \times \mathbb{R} \to \mathbb{R}$

$$\mathcal{L}(\boldsymbol{v}, \boldsymbol{w}; \beta) \equiv g(\boldsymbol{v}; \beta) - \boldsymbol{w}^T \boldsymbol{f}(\boldsymbol{v}; \beta), \tag{2.1}$$

i.e., the nonlinear system is adjoined to the output functional via an adjoint, or dual, state. Taking the first variation of $\mathcal{L}$ with respect to $\boldsymbol{v}$, $\boldsymbol{w}$, and $\beta$:

$$\delta \mathcal{L} = \delta \boldsymbol{v}^T \left[ g_{,\boldsymbol{v}}^T - \boldsymbol{w}^T \boldsymbol{f}_{,\boldsymbol{v}} \right] - \delta \boldsymbol{w}^T \left[ \boldsymbol{f} \right] + \delta \beta \left[ g_{,\beta} - \boldsymbol{w}^T \boldsymbol{f}_{,\beta} \right] \tag{2.2}$$

where the $\delta$ implies the variation of a variable. Requiring Eq. (2.1) to be stationary with respect to $\boldsymbol{w}$ recovers the primal problem from the second boxed term of Eq. (2.2) with solution $\boldsymbol{v} = \boldsymbol{u}_{ss}$. Simultaneously requiring stationarity with respect to $\boldsymbol{v}$, the equation for the steady adjoint is obtained from the first boxed term of Eq. (2.2) with $\boldsymbol{w} = \boldsymbol{\psi}_{ss}$:

$$\boldsymbol{f}^{\psi}(\boldsymbol{u}_{ss}, \boldsymbol{\psi}_{ss}; \beta) \equiv \boldsymbol{f}_{,\boldsymbol{u}}(\boldsymbol{u}_{ss}; \beta)^T \boldsymbol{\psi}_{ss} - g_{,\boldsymbol{u}}(\boldsymbol{u}_{ss}; \beta)^T = \boldsymbol{0}, \tag{2.3}$$

where $\boldsymbol{\psi}_{ss}$ is the steady adjoint. In the limit of small parameter variation, along with $\boldsymbol{v} = \boldsymbol{u}_{ss}$, $\boldsymbol{w} = \boldsymbol{\psi}_{ss}$, the parameter sensitivity of the output to $\beta$ is given by

$$\frac{d\mathcal{J}}{d\beta} = \lim_{\delta\beta \to 0} \frac{\delta \mathcal{L}}{\delta \beta} = g_{,\beta} - \boldsymbol{\psi}_{ss}^T \boldsymbol{f}_{,\beta}.$$

The steady primal equation is commonly solved using an iterative solver[36, 38, 75, 79] of the form

$$\boldsymbol{u}_{n+1} = \boldsymbol{u}_n - M_P(\boldsymbol{u}_n; \beta)^{-1} \boldsymbol{f}(\boldsymbol{u}_n; \beta), \tag{2.4}$$

where $\boldsymbol{u}_n$ is the approximation to $\boldsymbol{u}_{ss}$ at iteration $n$ and $M_P(\boldsymbol{u}_n; \beta)$ is a preconditioning matrix that approximates the Jacobian, $\boldsymbol{f}_{,\boldsymbol{u}}$, evaluated at $\boldsymbol{u}_n$. At each iteration, $\boldsymbol{f}(\boldsymbol{u}_n)$ and $M_P(\boldsymbol{u}_n)$ are updated. Once Eq. (2.4) is acceptably converged, Eq. (2.3) can be solved via the fixed point iteration:

$$\boldsymbol{\psi}_{m+1} = \boldsymbol{\psi}_m + M_P(\boldsymbol{u}_{ss}; \beta)^{-T} \left[ g_{,\boldsymbol{u}}(\boldsymbol{u}_{ss}; \beta)^T - \boldsymbol{f}_{,\boldsymbol{u}}(\boldsymbol{u}_{ss}; \beta)^T \boldsymbol{\psi}_m \right]. \tag{2.5}$$

The convergence rate of this fixed-point iteration for the adjoint is controlled by the eigenvalues of the matrix $M_P(\boldsymbol{u}_{ss};\beta)^{-T}\boldsymbol{f}_{,\boldsymbol{u}}(\boldsymbol{u}_{ss};\beta)^T$. Note that as $\boldsymbol{u}_n \to \boldsymbol{u}_{ss}$, the convergence rate of Eq. (2.4) will be controlled by eigenvalues of $M_P(\boldsymbol{u}_{ss};\beta)^{-1}\boldsymbol{f}_{,\boldsymbol{u}}(\boldsymbol{u}_{ss};\beta)$. Since $\lambda(A^{-T}B^T) = \lambda(A^{-1}B)$, unstable eigenvalues present in the primal iterative method will also be present in the adjoint iterative method.

### 2.1.2 Unsteady Adjoint

Building on the steady adjoint method of sensitivity calculation, it is straightforward to derive an unsteady parameter sensitivity. First, the unsteady primal equations are

$$\frac{d\boldsymbol{u}(t)}{dt} + \boldsymbol{f}(\boldsymbol{u}(t);\beta) = \boldsymbol{0}, \qquad \forall t \in I = (t_0, t_f) \tag{2.6}$$

$$\boldsymbol{a}(\boldsymbol{u}(t_0);\beta) = \boldsymbol{0} \tag{2.7}$$

where $\boldsymbol{u}(\cdot) : I \to \mathbb{R}^m$ is the primal state vector that now varies with time. The function $\boldsymbol{a}(\cdot;\cdot) : \mathbb{R}^m \times \mathbb{R} \to \mathbb{R}^m$ defines the initial condition. Determining the initial state with a general function, rather than specifying an initial state, $\boldsymbol{u}(t_0) = \boldsymbol{u}^0$, is not necessary to compute the primal. The general function only becomes necessary for computing output sensitivities when the initial condition is dependent on $\beta$, and for the spatial error estimate introduced in Chapter 5 when the initial condition changes with the solution space. Two initial conditions used in this work are:

$$\boldsymbol{a}(\boldsymbol{u}(t_0);\beta) \equiv \boldsymbol{u}(t_0) - \tilde{\boldsymbol{u}} = \boldsymbol{0}, \tag{2.8}$$

where $\tilde{\boldsymbol{u}}$ is a desired initial state and there is no $\beta$ dependence; and a modified steady-state initial condition,

$$\boldsymbol{a}(\boldsymbol{u}(t_0);\beta) \equiv \boldsymbol{f}(\boldsymbol{u}(t_0) - \tilde{\boldsymbol{u}};\beta) = \boldsymbol{0}. \tag{2.9}$$

In the latter case, $\tilde{\boldsymbol{u}}$ is a perturbation to the steady state solution, e.g., a wind gust profile. $\tilde{\boldsymbol{u}}$ in both cases is assumed to be fixed and incorporated into the particular $\boldsymbol{a}$, and so is not included as an argument.

While in principle a more general output could be used, here we use a combination of instantaneous terminal and time-integrated outputs:

$$\mathcal{J}^E(\beta) = h(\boldsymbol{u}(t_f);\beta) + \int_I g\left(\boldsymbol{u};\beta\right)\, dt. \tag{2.10}$$

The superscript $E$ denotes a time-integrated function. The Lagrangian for the unsteady problem can then be defined by adjoining the primal residual and initial condition as

$$\mathcal{L}^E(\boldsymbol{v},\boldsymbol{w},\tilde{\boldsymbol{w}};\beta) \equiv \mathcal{J}^E(\beta) - \tilde{\boldsymbol{w}}^T\boldsymbol{a}\left(\boldsymbol{v}(t_0);\beta\right) - \int_I \boldsymbol{w}^T\left(\frac{d\boldsymbol{v}}{dt} + \boldsymbol{f}(\boldsymbol{v};\beta)\right)\, dt, \tag{2.11}$$

where $\boldsymbol{v}(t), \boldsymbol{w}(t) : t \in I \rightarrow \mathbb{R}^m$ and $\tilde{\boldsymbol{w}} \in \mathbb{R}^m$. As in the steady case, take the first variation with respect to all of the arguments, and integrate appropriately by parts to get

$$\begin{aligned}
\delta\mathcal{L}^E = &\left[h_{,\beta} - \tilde{\boldsymbol{w}}^T\boldsymbol{a}_{,\beta} + \int_I \{g_{,\beta} - \boldsymbol{w}^T\boldsymbol{f}_{,\beta}\}\, dt\right]\delta\beta \\
&+ \delta\boldsymbol{v}(t_f)^T\left(h^T_{,\boldsymbol{v}} - \boldsymbol{w}\right)\big|_{t_f} - \delta\boldsymbol{v}(t_0)^T\left(\boldsymbol{a}^T_{,\boldsymbol{v}}\tilde{\boldsymbol{w}} - \boldsymbol{w}\right)\big|_{t_0} \\
&- \delta\tilde{\boldsymbol{w}}^T\boldsymbol{a} - \int_I \delta\boldsymbol{w}^T\left(\frac{d\boldsymbol{v}}{dt} + \boldsymbol{f}\right)\, dt \\
&- \int_I \delta\boldsymbol{v}^T\left(-\frac{d\boldsymbol{w}}{dt} + \boldsymbol{f}^T_{,\boldsymbol{v}}\boldsymbol{w} - g^T_{,\boldsymbol{v}}\right)\, dt
\end{aligned} \tag{2.12}$$

Requiring stationarity with respect to $\boldsymbol{w}$ gives the primal equation, Eq. (2.6), which can be solved for the primal state $\boldsymbol{v} = \boldsymbol{u}$. Requiring stationarity with respect to $\tilde{\boldsymbol{w}}$ to recovers the primal initial condition, Eq. (2.7), and the initial state, $\boldsymbol{u}(t_0)$. Similarly, the adjoint equation for $\boldsymbol{w} = \boldsymbol{\psi}$,

$$-\frac{d\boldsymbol{\psi}(t)}{dt} + \boldsymbol{f}^T_{,\boldsymbol{u}}\boldsymbol{\psi}(t) - g^T_{,\boldsymbol{u}} = \boldsymbol{0}, \quad \forall t \in I. \tag{2.13}$$

is found by requiring stationarity with $\boldsymbol{v}$, and has the terminal condition,

$$\boldsymbol{\psi}(t_f) = h^T_{,\boldsymbol{u}}. \tag{2.14}$$

found by enforcing stationarity with respect to $\boldsymbol{v}(t_f)$. The adjoint equation must be solved in reverse time, which is clear from the terminal equation, Eq. (2.14), and the sign of the time derivative, $d\boldsymbol{\psi}/dt$, term compared to the linearized system term in Eq. (2.13). The dependence of the second and third terms of Eq. (2.13) on the primal solution requires the full storage or recalculation via checkpointing of the primal through the timespan of interest. The initial condition on the adjoint can be removed by finding $\tilde{\boldsymbol{w}} = \tilde{\boldsymbol{\psi}}$ that satisfies

$$-\boldsymbol{\psi}(t_0) + \boldsymbol{a}_{,\boldsymbol{u}}(\boldsymbol{u}(t_0);\beta)^T \tilde{\boldsymbol{\psi}} = \boldsymbol{0}. \tag{2.15}$$

For the specified initial condition as in Eq. (2.8) this happens automatically by letting $\tilde{\boldsymbol{\psi}} = \boldsymbol{\psi}(t_0)$.

As in the steady case, the parameter sensitivity for unsteady problems, $d\mathcal{J}^E/d\beta$, is determined from the variation of the Lagrangian in the limit of small parameter variation

$$\frac{d\mathcal{J}^E}{d\beta} = \lim_{\delta\beta\to 0} \frac{\delta\mathcal{L}^E}{\delta\beta} = h_{,\beta} - \tilde{\boldsymbol{\psi}}^T \boldsymbol{a}_{,\beta} + \int_I \left\{ g_{,\beta} - \boldsymbol{\psi}^T \boldsymbol{f}_{,\beta} \right\} \, dt.$$

### 2.1.3 Temporally Discrete Unsteady Adjoint

In this work, several timestepping schemes are used for the computational examples, including a diagonally implicit Runge Kutta (DIRK) scheme. In order to simplify the analysis first-order backward differencing is used for comparison against the steady adjoint iteration. A more general temporal discretization of the unsteady adjoint is introduced in Appendix A, allowing application to multi-step and multi-stage temporal schemes.

The temporally-discrete primal problem is given by:

$$\begin{aligned} \frac{\boldsymbol{u}_i - \boldsymbol{u}_{i-1}}{\Delta t_i} + \boldsymbol{f}(\boldsymbol{u}_i;\beta) &= \boldsymbol{0}, \qquad \forall i \in [1, N],\ \beta \in \mathbb{R} \\ \boldsymbol{a}(\boldsymbol{u}_0;\beta) &= \boldsymbol{0} \end{aligned}$$

where $N$ is the number of time steps spanning $I$ with a variable time step $\Delta t_i = t_i - t_{i-1} > 0$, $t_0 = t_0$, and $t_N = t_f$. Deriving the time integration of the output from the time stepping scheme of the primal problem gives:

$$J^E(\beta) = h(\boldsymbol{u}_N;\beta) + \sum_{i=1}^{N} g(\boldsymbol{u}_i;\beta)\, \Delta t_i$$

The discrete Lagrangian can then be defined by:

$$L^E(\boldsymbol{v}, \boldsymbol{w}, \tilde{\boldsymbol{w}};\beta) = J^E(\beta) - \tilde{\boldsymbol{w}}^T \boldsymbol{a}\,(\boldsymbol{v}_0;\beta) - \sum_{i=1}^{N} \Delta t_i \boldsymbol{w}_i^T \left( \frac{\boldsymbol{v}_i - \boldsymbol{v}_{i-1}}{\Delta t_i} + \boldsymbol{f}(\boldsymbol{v}_i;\beta) \right)$$

Enforcing stationarity of the discrete Lagrangian with respect to $\boldsymbol{w}_i\, \forall i \in [1, N+1], \boldsymbol{v}_i\, \forall i \in$

$[0, N]$ again recovers the discrete primal equation for $\boldsymbol{v}_i = \boldsymbol{u}_i$ and gives the discrete adjoint equation for $\boldsymbol{w}_i = \boldsymbol{\psi}_i$ and $\tilde{\boldsymbol{w}} = \tilde{\boldsymbol{\psi}}$:

$$\begin{aligned} \frac{\boldsymbol{\psi}_i - \boldsymbol{\psi}_{i+1}}{\Delta t_i} + \boldsymbol{f}_{,\boldsymbol{u}}(\boldsymbol{u}_i;\beta)^T \boldsymbol{\psi}_i - g^T_{,\boldsymbol{u}}(\boldsymbol{u}_i;\beta) = \boldsymbol{0} \ \forall i \in [1, N] \\ \boldsymbol{\psi}_{N+1} = h^T_{,\boldsymbol{u}}(\boldsymbol{u}_N;\beta) \\ -\boldsymbol{\psi}_1 + \boldsymbol{a}_{,\boldsymbol{u}}(\boldsymbol{u}_0;\beta)^T \tilde{\boldsymbol{\psi}} = \boldsymbol{0} \end{aligned} \tag{2.16}$$

In contrast to Eq. (2.5), the linearization and output derivative in the unsteady adjoint equation are taken about the primal state at successive time instants, rather than at a fixed point. This stabilizes the calculation, as stability no longer depends on the eigenvalues of $\Delta t_i \boldsymbol{f}_{,\boldsymbol{u}}(\boldsymbol{u}_i;\beta)^T$ being within the unit circle at a particular timestep, $i$. Eq. (2.16) can now be used for deterministic problems to compute the adjoint even about unstable stationary points.

The parameter sensitivity for the first order backward difference is derived as in the continuous case:

$$\frac{dJ^E}{d\beta} = h_{,\beta}(\boldsymbol{u}_N;\beta) - \tilde{\boldsymbol{\psi}}^T \boldsymbol{a}_\beta(\boldsymbol{v}_0;\beta) + \sum_{i=1}^{N} \Delta t_i \left\{ g_{,\beta}(\boldsymbol{u}_i;\beta) - \boldsymbol{\psi}_i^T \boldsymbol{f}_{,\beta}(\boldsymbol{u}_i;\beta) \right\}$$

## 2.2 Discontinuous Galerkin Formulation

### Navier-Stokes Equations

The time dependent, compressible Navier-Stokes equations can be expressed as a general conservation law form as

$$\frac{\partial \boldsymbol{u}}{\partial t} + \nabla \cdot \mathcal{F}_i(\boldsymbol{u}) - \nabla \cdot \mathcal{F}_v(\boldsymbol{u}, \nabla \boldsymbol{u}) = \boldsymbol{S}(\boldsymbol{u}, \nabla \boldsymbol{u}) \quad \in \Omega \tag{2.17}$$

where $\Omega$ is the domain. In two dimensions, the state vector is given by $\boldsymbol{u} = [\rho, \rho u, \rho v, \rho e]^T$. $\mathcal{F}_i(\boldsymbol{u})$ and $\mathcal{F}_v(\boldsymbol{u}, \nabla \boldsymbol{u})$ are the inviscid and viscous flux vectors, respectively, and $\boldsymbol{S}$ is a source term. Discontinuous Galerkin finite element (DG) discretizations have become increasingly popular for the higher-order accurate solution of conservation laws, especially the Euler and Navier-Stokes equations[9, 10]. The DG method takes the strong form of the conservation law in Eq. (2.17) and derives a weak form. In this work, the numerical flux is computed using the Roe flux[96]. Boundary conditions are implemented as described in Oliver[81] and the second method of Bassi and Rebay[11, 12] (BR2) is utilized for the viscous flux. For a more detailed description of the DG implementation employed in this work, see Fidkowski et al.[31], Diosady and Darmofal[22], Barter and Darmofal[6], and Yano et al.[128].

### 2.2.1 Steady DG Formulation

The domain, $\Omega$, is represented by $\mathcal{T}_h$, a tessellation of the domain into non-overlapping elements $\kappa$ where $\overline{\Omega} = \cup\overline{\kappa}$ and $\kappa_i \cap \kappa_j = \emptyset, i \neq j$. Consider a discontinuous finite-dimensional space, $\mathcal{V}_h^p$:

$$\mathcal{V}_h^p = \left\{ \boldsymbol{v} \in L^2(\Omega) : \boldsymbol{v}|_\kappa \in P^p(\kappa) \forall \kappa \in \mathcal{T}_h \right\}.$$

In words, the function space of element discontinuous, piecewise polynomials of degree $p$. The finite-dimensional problem is then given by the semi-linear form:

$$\mathcal{R}_{hp}(\boldsymbol{u}_{hp}, \boldsymbol{v}_{hp}; \beta) = 0, \quad \forall \boldsymbol{v}_{hp} \in \mathcal{V}_h^p. \tag{2.18}$$

where $\mathcal{R}_{hp}(\cdot,\cdot;\cdot) : \mathcal{V}_h^p \times \mathcal{V}_h^p \times \mathbb{R} \to \mathbb{R}$ is a Frechét differentiable semilinear form with derivative $\mathcal{R}'[\cdot](\cdot,\cdot;\cdot)$. The term in brackets, $[\cdot]$, denotes the state about which the derivative is taken.

The output functional, $\mathcal{J}(\boldsymbol{u};\beta)$, has a finite-dimensional counterpart, $\mathcal{J}_{hp}(\boldsymbol{u}_{hp};\beta)$, but it is assumed hereafter that they are identical in form and are used interchangeably. Due to the different function spaces considered in the derivation of the error estimate, the state dependence will be included in the output notation for the DG context. Next, the finite-dimensional Lagrangian is given by, $\mathcal{L}_{hp}(\cdot,\cdot;\cdot) : \mathcal{V}_h^p \times \mathcal{V}_h^p \times \mathbb{R} \to \mathbb{R}$,

$$\mathcal{L}_{hp}(\boldsymbol{v}_{hp}, \boldsymbol{w}_{hp}; \beta) = \mathcal{J}_{hp}(\boldsymbol{v}_{hp}; \beta) - \mathcal{R}_{hp}(\boldsymbol{v}_{hp}, \boldsymbol{w}_{hp}; \beta).$$

Again requiring stationarity of the first variation of $\mathcal{L}$ with respect to permissible variations in $\boldsymbol{v}_{hp}$ and $\boldsymbol{w}_{hp}$, the equation for the steady finite-dimensional adjoint is obtained

$$\mathcal{R}'_{hp}\left[\boldsymbol{u}_{hp}\right](\boldsymbol{v}_{hp}, \boldsymbol{\psi}_{hp}; \beta) = \mathcal{J}'_{hp}\left[\boldsymbol{u}_{hp}\right](\boldsymbol{v}_{hp}; \beta), \quad \forall \boldsymbol{v}_{hp} \in \mathcal{V}_h^p.$$

The discrete form of the equations is obtained by defining a basis, $\boldsymbol{\phi}_{hp}$, that spans the space $\mathcal{V}_h^p$. The solution vector, $\boldsymbol{u}_{hp}$, may then be expressed as a linear combination of basis functions, using as coefficients the discrete solution vector, $\boldsymbol{U}_{hp}$, such that

$$\boldsymbol{u}_{hp}(\boldsymbol{x}) = \sum_i \boldsymbol{U}_{hpi} \boldsymbol{\phi}_{hpi}(\boldsymbol{x}).$$

Except when explicitly limited to a single element, $\boldsymbol{u}$ and other variables in $\mathcal{V}_h^p$ describe vectors spanning the entire tessellated domain, $\mathcal{T}_h$, so $\boldsymbol{x}$ is removed from the notation. By defining the discrete residual,

$$\boldsymbol{R}_{hp}(\boldsymbol{U}_{hp}; \beta)_i = \mathcal{R}(\boldsymbol{u}_{hp}, \boldsymbol{\phi}_{hpi}; \beta) = 0,$$

and the discrete output functional,

$$J_{hp}(\boldsymbol{U}_{hp}; \beta) = \mathcal{J}_{hp}(\boldsymbol{u}_{hp}; \beta),$$

the resulting linear system is then

$$\frac{\partial \boldsymbol{R}_{hp}(\boldsymbol{U}_{hp}; \beta)}{\partial \boldsymbol{U}_{hp}}^T \boldsymbol{\Psi}_{hp} = \frac{\partial J_{hp}(\boldsymbol{U}_{hp}; \beta)}{\partial \boldsymbol{U}_{hp}}^T, \tag{2.19}$$

with $\boldsymbol{\Psi}_{hp}$ the discrete adjoint solution and $\boldsymbol{U}_{hp}$ such that

$$\boldsymbol{R}_{hp}(\boldsymbol{U}_{hp}; \beta) = 0. \tag{2.20}$$

A similar exercise for the sensitivity as for the continuous equation gives

$$\frac{dJ_{hp}}{d\beta} = \frac{\partial J_{hp}}{\partial \beta} - \boldsymbol{\Psi}_{hp}^T \frac{\partial \boldsymbol{R}_{hp}(\boldsymbol{U}_{hp}; \beta)}{\partial \beta}$$

with $\boldsymbol{U}_{hp}$ and $\boldsymbol{\Psi}_{hp}$ constrained by Eq. (2.20) and Eq. (2.19), respectively.

The solution to Eq. (2.20) is found by nonlinear iteration similar to Eq. (2.4),

$$\boldsymbol{U}_{hp}^{n+1} = \boldsymbol{U}_{hp}^n - M_P^{-1} \boldsymbol{R}_{hp}(\boldsymbol{U}_{hp}^n; \beta) \tag{2.21}$$

with $M_P$ the preconditioning matrix that now approximates the Jacobian, $\partial \boldsymbol{R}_{hp} / \partial \boldsymbol{U}_{hp}$.

The solution of Eq. (2.19) can then be found via an iterative linear solver as in Eq. (2.5),

$$\boldsymbol{\Psi}_{hp}^{n+1} = \boldsymbol{\Psi}_{hp}^{n} + M_P^{-T}\left(\frac{\partial J_{hp}(\boldsymbol{U}_{hp};\beta)}{\partial \boldsymbol{U}_{hp}}^T - \frac{\partial \boldsymbol{R}_{hp}(\boldsymbol{U}_{hp};\beta)}{\partial \boldsymbol{U}_{hp}}^T \boldsymbol{\Psi}_{hp}^{n}\right)$$

with the final solution of Eq. (2.21) used to define the preconditioning and Jacobian matrices.

### 2.2.2 Unsteady DG Formulation

The weak form of the finite-dimensional temporal term is given by the inner product of the time derivative of the state with a test function:

$$\left(\boldsymbol{v}_{hp}, \frac{d\boldsymbol{u}_{hp}(t)}{dt}\right),$$

which is added to the spatial residual to form the unsteady primal residual:

$$\left(\boldsymbol{v}_{hp}, \frac{d\boldsymbol{u}_{hp}(t)}{dt}\right) + \mathcal{R}_{hp}(\boldsymbol{u}_{hp}(t), \boldsymbol{v}_{hp};\beta) \;=\; 0, \quad \forall \boldsymbol{v}_{hp} \in \mathcal{V}_h^p,\ \forall t \in I. \tag{2.22}$$

The initial condition, $\boldsymbol{u}_{hp}(t_0)$, satisfies

$$\mathcal{A}_{hp}(\boldsymbol{u}_{hp}(t_0), \boldsymbol{v}_{hp};\beta) = 0, \quad \forall \boldsymbol{v}_{hp} \in \mathcal{V}_h^p.$$

The finite-dimensional unsteady output is given by:

$$\mathcal{J}_{hp}^E(\boldsymbol{u}_{hp};\beta) \equiv \mathcal{H}_{hp}(\boldsymbol{u}_{hp}(t_f);\beta) + \int_I \mathcal{G}_{hp}(\boldsymbol{u}_{hp};\beta)\,dt \tag{2.23}$$

where $\mathcal{G}_{hp}(\cdot;\cdot)$, $\mathcal{H}_{hp}(\cdot;\cdot) : \mathcal{V}_h^p \times \mathbb{R} \to \mathbb{R}$ are Frechét differentiable. As in the steady case, the finite-dimensional output, $\mathcal{J}_{hp}^E(\cdot;\cdot)$, and the unsteady output, $\mathcal{J}^E(\cdot;\cdot)$, are assumed to be equivalent. The associated adjoint equation is given by

$$-\left(\frac{d\boldsymbol{\psi}_{hp}(t)}{dt}, \boldsymbol{v}_{hp}\right) + \mathcal{R}'_{hp}[\boldsymbol{u}_{hp}(t)](\boldsymbol{v}_{hp}, \boldsymbol{\psi}_{hp}(t);\beta) - \mathcal{G}'_{hp}[\boldsymbol{u}_{hp}(t)](\boldsymbol{v}_{hp};\beta) = 0,$$
$$\forall \boldsymbol{v}_{hp} \in \mathcal{V}_h^p,\ \forall t \in I,$$

with terminal condition

$$\boldsymbol{\psi}_{hp}(t_f) = \mathcal{H}(\boldsymbol{u}_{hp}(t_f);\beta).$$

The initial condition adjoint equation is

$$-\big(\boldsymbol{\psi}_{hp}(t_0), \boldsymbol{v}_{hp}\big) + \mathcal{A}'_{hp}[\boldsymbol{u}_{hp}(t_0)](\boldsymbol{v}_{hp}, \tilde{\boldsymbol{\psi}}_{hp};\beta) = 0, \quad \forall \boldsymbol{v}_{hp} \in \mathcal{V}_h^p.$$

Finally, the unsteady parameter sensitivity to $\beta$ is then

$$\frac{d\mathcal{J}_{hp}^E}{d\beta} = \mathcal{J}_{hp,\beta}^E(\boldsymbol{u}_{hp};\beta) - \mathcal{A}_{hp,\beta}(\boldsymbol{u}_{hp}(t_0), \tilde{\boldsymbol{\psi}}_{hp};\beta) - \int_I \mathcal{R}_{hp,\beta}(\boldsymbol{u}_{hp}, \boldsymbol{\psi}_{hp};\beta)\, dt.$$

For the discrete problem, the temporal term becomes the primal state coefficients premultipled by a mass matrix, $\mathcal{M}$, where

$$\mathcal{M}_{ij} = \int_\kappa \boldsymbol{\phi}_{hp_i} \boldsymbol{\phi}_{hp_j}\, d\Omega_\kappa$$

for each element. For the entire domain, the mass matrix is element-block diagonal. The spatially discrete primal, adjoint, and initial adjoint equations are then:

$$0 = \mathcal{M}\frac{d\boldsymbol{U}_{hp}(t)}{dt} + \boldsymbol{R}_{hp}(\boldsymbol{U}_{hp}(t);\beta), \quad \forall t \in I \tag{2.24}$$

$$0 = -\mathcal{M}\frac{d\boldsymbol{\Psi}_{hp}(t)}{dt} + \left[\frac{\partial \boldsymbol{R}_{hp}(\boldsymbol{U}_{hp}(t);\beta)}{\partial \boldsymbol{U}_{hp}(t)}\right]^T \boldsymbol{\Psi}_{hp}(t) - \frac{\partial G_{hp}(\boldsymbol{U}_{hp}(t);\beta)}{\partial \boldsymbol{U}_{hp}(t)}^T, \quad \forall t \in I \tag{2.25}$$

$$0 = -\mathcal{M}\boldsymbol{\Psi}_{hp}(t_0) + \left[\frac{\partial \boldsymbol{A}_{hp}(\boldsymbol{U}_{hp}(t_0);\beta)}{\partial \boldsymbol{U}_{hp}(t_0)}\right]^T \tilde{\boldsymbol{\Psi}}_{hp} \tag{2.26}$$

with parameter sensitivity given by

$$\frac{dJ_{hp}^E}{d\beta} = \frac{\partial J^E}{\partial \beta} - \tilde{\boldsymbol{\Psi}}^T \frac{\partial \boldsymbol{A}_{hp}}{\partial \beta} - \int_I \boldsymbol{\Psi}_{hp}^T \frac{\partial \boldsymbol{R}_{hp}^E}{\partial \beta}\, dt. \tag{2.27}$$

## 2.3 Solution Approach

For steady simulations, Newton's method is used with backward Euler pseudotime stepping to increase robustness during transients. For uniform flow initial conditions the simulation is started with $CFL = 1$ and allowed to increase (by factors of 2) as long as the flow solution maintains positive density and pressure, up to a maximum $CFL_{max} = 10^{30}$. All time-inaccurate results are reported after the $CFL$ has reached $CFL_{max}$.

For time-accurate simulations, the temporal derivative term is discretized either with a fourth order explicit singly diagonal implicit Runge-Kutta (ESDIRK)[118] or fourth order diagonal implicit RK (DIRK) scheme[90]. At each timestep and stage of the (ES)DIRK scheme, the same Newton's method as for steady problems is used to converge the unsteady primal equation Eq. (2.24).

The linear systems arising in both the time-inaccurate and time-accurate simulations can be solved with stationary iterative methods (block-Jacobi and line-Jacobi) either as the solver or as a preconditioner for GMRES[100]. At each iteration, the block-Jacobi method solves for the degrees of freedom within each element as a block. The line-Jacobi stationary iterative method expands the size of the block, solving within a line of elements. As a result, the stationary iterative methods are most effective at reducing errors within blocks (i.e. local errors). The GMRES solver uses a Krylov-subspace which is computed over the entire flow domain, incorporating more than nearest neighbor dependence and making it more effective at reducing global errors. Unlike the stationary iterative methods, GMRES is guaranteed to converge monotonically, regardless of the presence of outlier (unstable) modes, for any nonsingular matrix. Because the GMRES linear solver does not have the same susceptibility as the stationary iterative methods to unstable modes, it is capable of computing a stationary point solution if one exists, whether or not the stationary point solution is stable. For additional convergence improvement, an incomplete LU factorization[22] with MDF reordering[86]. can replace the stationary iterative method as the GMRES preconditioner. Because of the improved efficiency and robustness, GMRES with ILU preconditioning is used predominantly in this work. However, the stronger ILU preconditioner completely masks the time-inaccurate, unsteady behavior under investigation in Chapter 3, so the stationary iterative methods are used instead.

For time-inaccurate problems, a simulation is considered converged if the $L_2$ norm of the spatial residual is decreased by 10 orders of magnitude from its initial value. For time-accurate problems, the same convergence criterion is used for the unsteady residual including the additional temporal terms of each timestep and timestep stage.

# Chapter 3

# Effect of Small-Scale Output Unsteadiness on Adjoint-Based Sensitivity

## 3.1 Iterative Methods for Small-Scale Unsteadiness

Campobasso & Giles[17] investigated the effects of small-scale flow instabilities in the linear analysis of turbomachinery aeroelasticity. In their work, they conduct a linear unsteady flow analysis in which they compute the nonlinear steady solution of the Navier-Stokes equations as a background flow, then calculate the impact of small disturbances using the linearized Navier-Stokes equations.

To solve the perturbed linear system, the authors first linearize the unsteady flow equations Eq. (2.6) about the steady background solution to get

$$\frac{d\tilde{\boldsymbol{u}}}{dt} + L\tilde{\boldsymbol{u}} = \tilde{\boldsymbol{f}} \tag{3.1}$$

where $L$ is $\boldsymbol{f}_{,\boldsymbol{u}}$ evaluated at the steady-state or background solution calculated by Eq. (2.4) and $\tilde{\boldsymbol{f}}$ is a forcing function resulting from the perturbation of the boundary conditions and/or deformation of the mesh. Because the problem is time-periodic and linear, the problem is decomposed into a sum of complex harmonics and the problem for each mode becomes

$$(j\omega I + L)\,\hat{\boldsymbol{u}} = \hat{\boldsymbol{f}} \tag{3.2}$$

where $\hat{\boldsymbol{u}}$ and $\hat{\boldsymbol{f}}$ are the Fourier transforms of $\tilde{\boldsymbol{u}}$ and $\tilde{\boldsymbol{f}}$, respectively. By setting $A = (j\omega I + L)$ and $\boldsymbol{b} = \tilde{\boldsymbol{f}}$, each mode can then be viewed as a linear system of the form

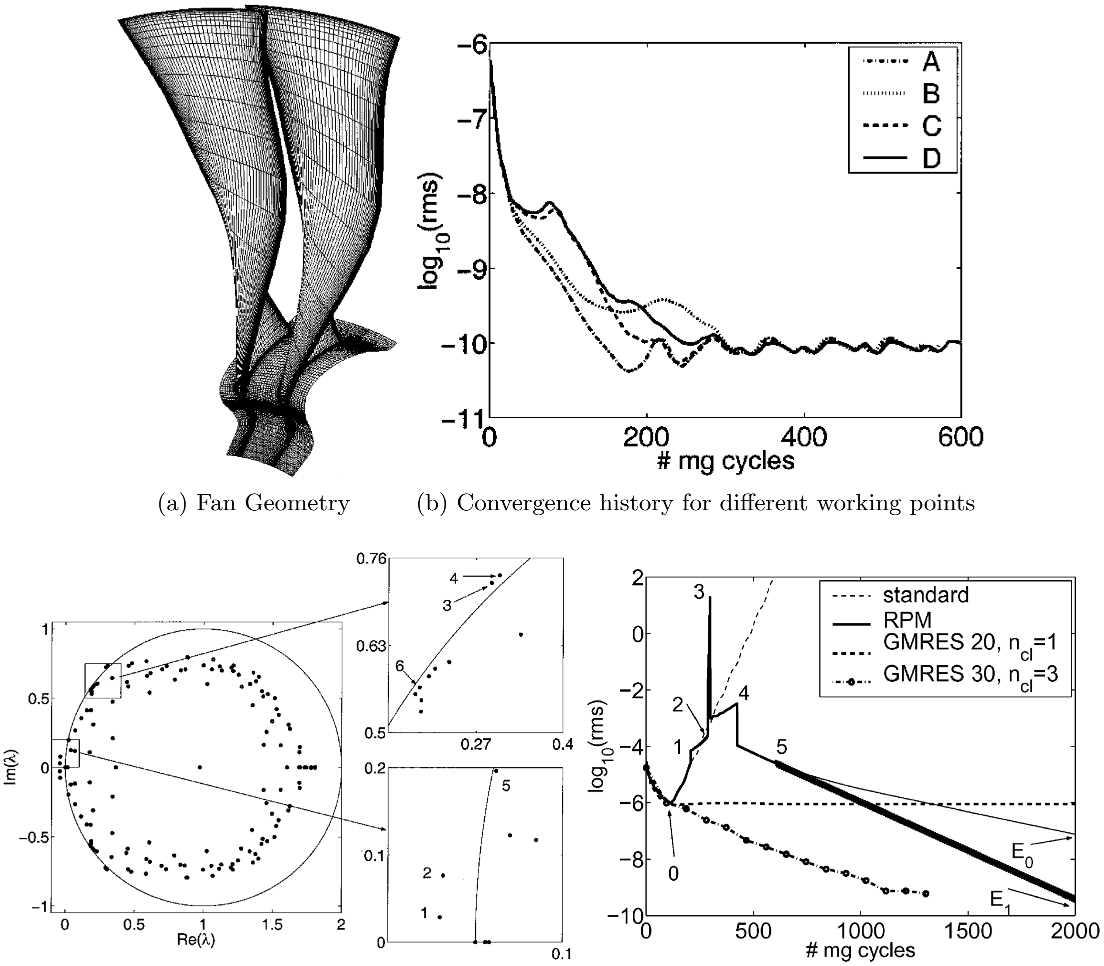
$$A\hat{\boldsymbol{u}} = \boldsymbol{b}, \tag{3.3}$$

where $A$ is evaluated at the approximate background solution, $\boldsymbol{u}_{ss}$, from the solution of Eq. (2.4). The solution can be sought iteratively using the same preconditioning algorithm as for the background solution, and can be written in standard form as

$$\hat{\boldsymbol{u}}_{n+1} = \hat{\boldsymbol{u}}_n + M_P^{-1}(\boldsymbol{b} - A\hat{\boldsymbol{u}}_n) \tag{3.4}$$

where $M_P$ is again the preconditioning matrix approximation to $A$. The necessary condition for convergence of the method is that the eigenvalues of $(I - M_P^{-1}A)$ must lie within the unit circle centered at the origin. The authors found that this condition is not met for cases where the background flow calculation does not fully converge, but instead reaches a small-amplitude limit cycle. They attribute this instability to separation bubbles, corner stalls, vortex shedding, and other unsteady physical phenomena. While these instabilities do not prevent the primal nonlinear problem from converging to an acceptable level, they still result in a small number of eigenvalues outside the convergence region in the linear problem. Figure (3-1b) shows the convergence history of the fan blade at several working points. Figure (3-1c) shows the eigenvalues $M_P^{-1}A$ for one of the working points with a small number outside the stability region. This is salient to the adjoint problem because, as discussed by Giles, the eigenvalues of the adjoint problem are the same as those of the linearized NS equations when the same iterative solver is used.

For problems with such an instability, Campobasso & Giles proposed stabilizing the iterative scheme utilizing GMRES[17] or the Recursive Projection Method (RPM)[18]. The two approaches are discussed further in Appendix C. When the number of unstable modes is small these methods can efficiently stabilize the iteration. For more strongly unsteady problems, these methods will both become prohibitively expensive to stabilize the problem.



(a) Fan Geometry

(b) Convergence history for different working points

(c) Linearized system eigenvalues

(d) Linearized system convergence history for different iterative solvers

Figure 3-1: Campobasso & Giles, 3D fan test case[17]

A time-inaccurate approach will also become more inaccurate as the difference between the parameter sensitivity and error estimates calculated at an arbitrary point solution and those averaged over time becomes large, as will be demonstrated by an airfoil example in Section 3.3.

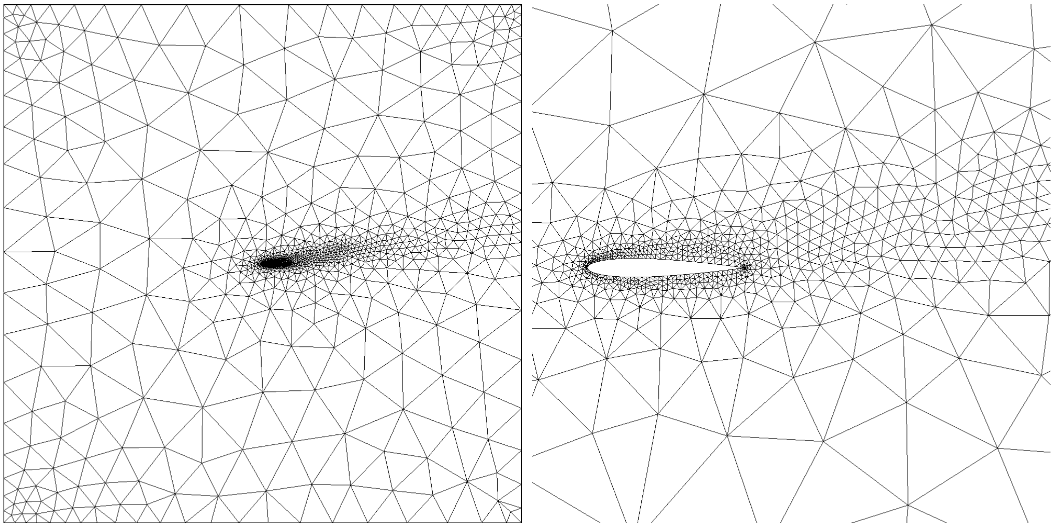
## 3.2 Laminar Airfoil Model Problem

Subsonic, laminar flow about an airfoil is used to demonstrate the impact of small-scale output unsteadiness on sensitivity analysis. Small-scale unsteadiness in an output is characterized by unsteadiness that is small in magnitude compared its mean. Specifically, we consider the $Ma = 0.5$, $Re = 1500$ flow about a two-dimensional NACA 0012 airfoil over a range of angles of attack. This problem was chosen because the flowfield exhibits a stable stationary point at lower angles of attack, but for higher angles of attack has an unstable stationary point leading to unsteady behavior. In the computational mesh, the airfoil leading edge is located at $(0, 0)$ with a unit chord. The domain is square and extends $[-10, 10] \times [-10, 10]$. The mesh refinement was determined using adjoint-based adaptation, driving the error in lift coefficient, $C_L$, to less than 0.0005 for a $p = 3$ stationary point solution at integer angles of attack from $\alpha = 0°$ to $\alpha = 13°$. For the problems in this work, a mesh was generated with 2158 elements, seen in Figure (3-2). The resulting airfoil surface is defined by 101 nodes, connected by cubic polynomial edges ($q = 3$), and with the endpoint and edge nodes projected to the analytic geometry. For this problem, the flow and adjoint states are represented in each element by a polynomial of degree 3 ($p = 3$). For time-accurate unsteady problems, a timestep of 0.1 is used with the ESDIRK4 temporal discretization where a time of 1 corresponds to convection time to traverse the airfoil chord. A mesh and timestep dependence study was done and is discussed in Appendix D.

## 3.3 Time-Inaccurate Simulations

### 3.3.1 $9°$ Angle of Attack

For freestream conditions $Ma = 0.5$ and $Re = 1500$, a stationary point solution of a NACA 0012 airfoil at $9°$ angle of attack can be found using a line-Jacobi preconditioned GMRES linear solver. This stationary point solution exhibits a large laminar separation region,

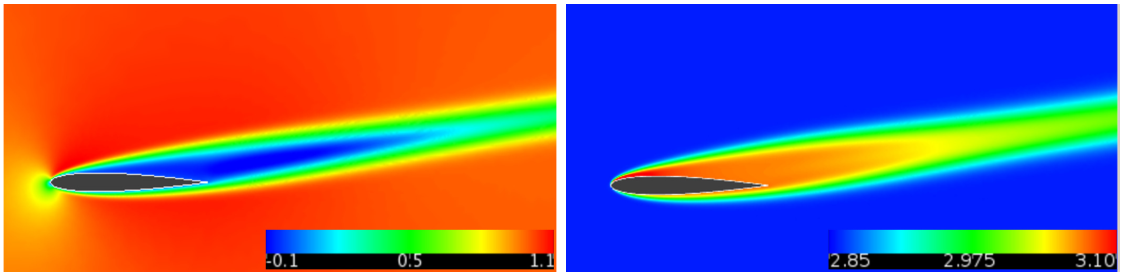


(a) Computational domain

(b) Airfoil mesh

Figure 3-2: NACA 0012 mesh

shown in Figure (3-3). For weaker linear solvers, such as line-Jacobi without GMRES,



(a) X-momentum

(b) Entropy

Figure 3-3: NACA 0012, GMRES stationary point solution ($Ma = 0.5$, $Re = 1500$, $\alpha = 9°$)

the eigenvalues of the linear system lie outside the stability region which is a unit circle centered at the origin. Specifically, the eigenvalues for the linear system at the stationary point of the line-Jacobi preconditioned system ($I - M_P^{-1}\boldsymbol{R_U}$) are shown in Figure (3-4). These eigenvalues were calculated using a Krylov subspace of 1000 vectors, giving 1000 eigenvalues. There are two eigenvalues located just outside the unit circle (denoted by the red squares), and the line-Jacobi iterative method will be unstable for the linearized primal as well as the adjoint problem. Further, this instability will also prevent the nonlinear primal iteration from converging to the stationary point. As seen in Figure (3-5), when the line-Jacobi iterative solver is started from the prolongated $p = 2$ stationary point solution, the nonlinear residual is initially $\sim 10^{-3}$, but immediately begins rising until it reaches

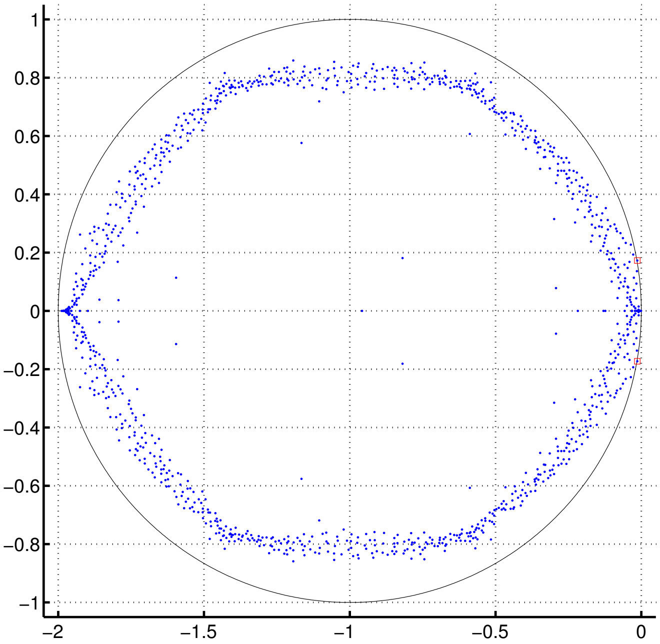
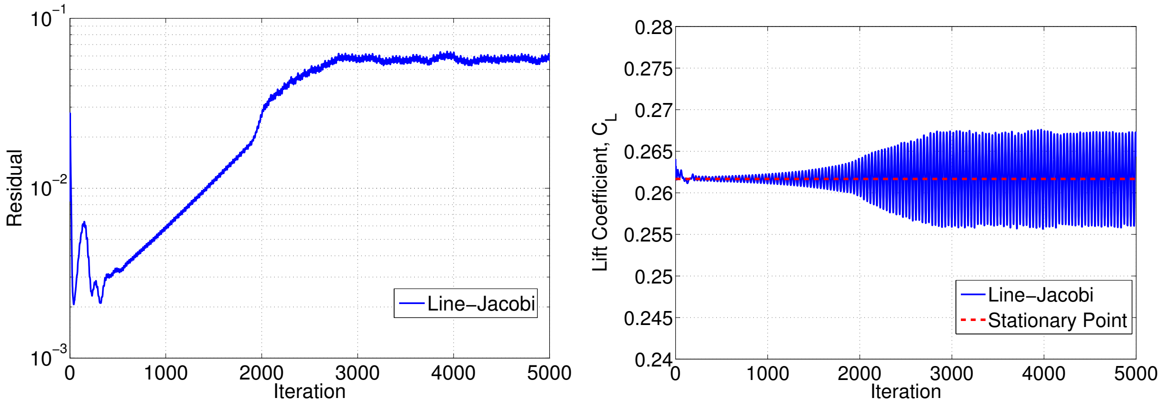


Figure 3-4: Eigenvalues of line-preconditioned iterative system $(I - M_P^{-1}\boldsymbol{R_U})$ at stationary point ($Ma = 0.5$, $Re = 1500$, $\alpha = 9°$, 4 processors). Unstable eigenvalues plotted as red squares.

an equilibrium at $\sim 0.06$. Likewise, the lift does not converge to a single value but has a small amplitude time-inaccurate unsteadiness with a mean near the value of the stationary point. After the transients have diminished, the standard deviation of $C_L$ from the mean is approximately 1.5% of the stationary point value.

To examine the time-inaccurate unsteadiness in the line-Jacobi solution further, the transient was allowed to die out, and the mean and the standard deviation of the time-inaccurate solution were calculated over 50000 iterations. The standard deviation of each component of the state was computed by using a Lagrange basis and calculating the value at each Lagrange point. At a Lagrange point $i$, the mean is computed as $\overline{\boldsymbol{u}^i} = \frac{1}{N}\sum_{n=1}^{N}\boldsymbol{u}_n^i$ and the standard deviation is computed as $\sigma^i(\boldsymbol{u}) = \left[\frac{1}{N}\sum_{n=1}^{N}\left(\boldsymbol{u}_n^i - \overline{\boldsymbol{u}^i}\right)^2\right]^{1/2}$. The mean and standard deviation of the x-momentum are plotted in Figure (3-6). The standard deviation in the x-momentum is representative of the variation of all of the states, with a maximum magnitude of approximately 6% of the mean flow on the laminar separation bubble boundary.

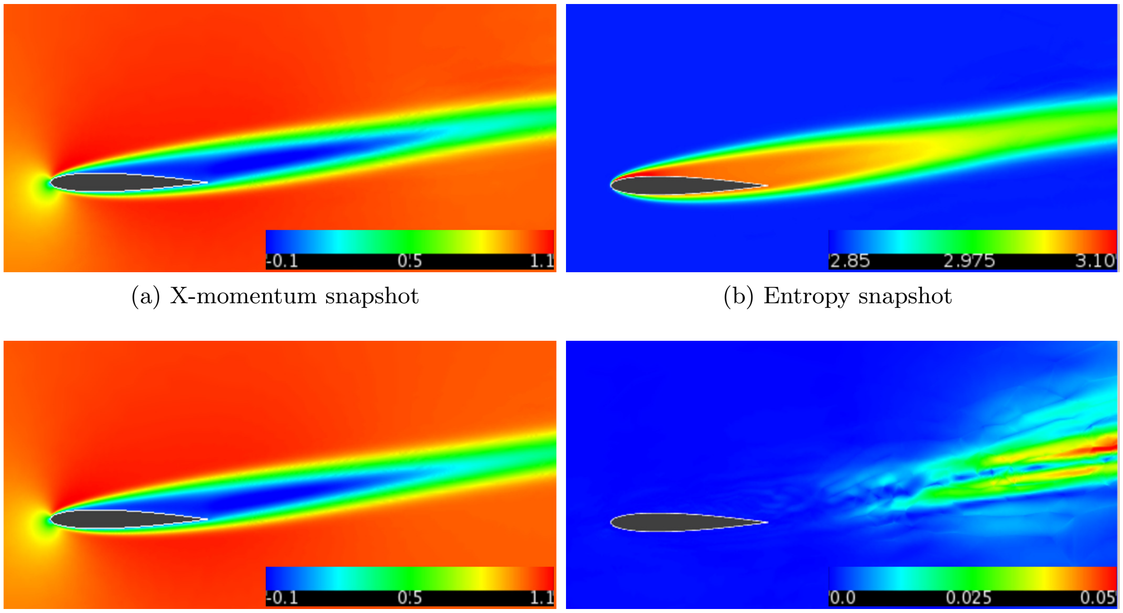
The time-inaccurate unsteady behavior in this case is highly dependent on the strength of the iterative solver. For stationary iterative solvers, time-inaccurate unsteadiness begins to



(a) Residual History

(b) Lift Coefficient History

Figure 3-5: Convergence history for line-Jacobi at $Ma = 0.5$, $Re = 1500$, $\alpha = 9°$, 4 processors



(a) X-momentum snapshot

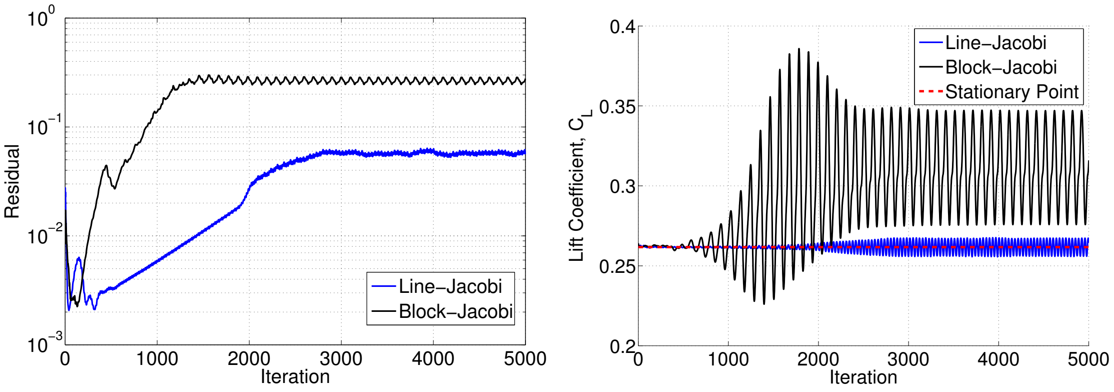
(b) Entropy snapshot

(c) X-momentum mean

(d) X-momentum standard deviation

Figure 3-6: NACA 0012, line-Jacobi iterative solver ($Ma = 0.5$, $Re = 1500$, $\alpha = 9°$)

appear in the wake region, 1-2 chords downstream, as can be in Figure (3-6d). Regardless, the mean flow near the airfoil appears almost identical to the stationary point, as in a comparison of Figure (3-3a) and Figure (3-6c). Replacing the line-Jacobi iterative solver with a block-Jacobi solver decreases the strength of the preconditioning and removes fewer unstable modes. The convergence history for the same conditions, but with the weaker solver, is shown in Figure (3-7), exhibiting larger time-inaccurate unsteadiness in the output as well as a different mean value. The mean and standard deviation of the x-momentum are plotted in Figure (3-8), showing an increase in the magnitude of the standard deviation in the wake. The x-momentum and entropy snapshots show large unsteadiness in the wake, with an indication of vortex shedding into the wake (though we emphasize this is not a time-accurate simulation).

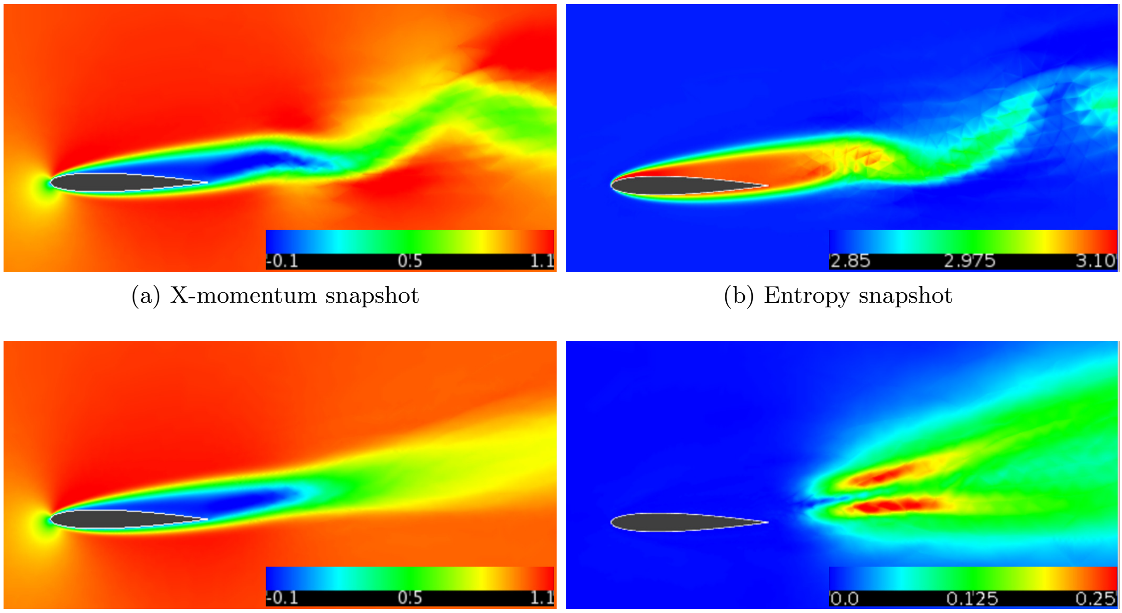


(a) Residual History

(b) Lift Coefficient History

Figure 3-7: Convergence history for block-Jacobi at $Ma = 0.5$, $Re = 1500$, $\alpha = 9°$

Returning to the line-Jacobi iterative results, one could reasonably presume from the low variation in the lift coefficient that linearizing about an iterate of the unconverged primal is sufficient to provide accurate lift sensitivity estimates. However, the derivation of the adjoint-based sensitivity depends on the stationarity of the Lagrangian with respect to variation of the adjoint. When the spatial residual is not zero for the primal state about which the adjoint analysis is performed, the output sensitivity has an additional term. Examination of Eq. (2.2) shows that the output sensitivity then becomes dependent on the inner product of the residual and the derivative of the adjoint with respect to the parameter. To demonstrate the impact that using an unconverged solution can have on the sensitivity analysis, we use a stronger linear solver (GMRES with ILU preconditioning) to perform adjoint analysis by linearizing about iterates, $\boldsymbol{u}_n$, for a series of iterations, $n$. The insensitivity



(a) X-momentum snapshot

(b) Entropy snapshot

(c) X-momentum mean

(d) X-momentum standard deviation

Figure 3-8: NACA 0012, block-Jacobi iterative solver ($Ma = 0.5$, $Re = 1500$, $\alpha = 9°$)

to unstable modes of the stronger linear solver enables convergence of the adjoint residual linearized about the unconverged iterate solutions. As seen in Figure (3-9), this estimate of $dC_L/d\alpha$ (*i.e.*, $g = C_L$ and $\beta = \alpha$) varies over a range from 0.519 to $0.639\, rad^{-1}$. Depending on the iterate used in the linearization, the sensitivity from an unconverged state could be significantly different from the sensitivity computed about the stationary point ($0.558\, rad^{-1}$). Taking the average of these iterate sensitivities gives $0.571\, rad^{-1}$, which is within $\sim 2.3\%$ of the stationary point sensitivity. When the GMRES linear solver is used to calculate the output sensitivity of the iteration average of the line-Jacobi solutions, the sensitivity is $0.571\, rad^{-1}$, matching the average of the iterate sensitivities. The usefulness of these sensitivities is limited, as they all required the stronger GMRES solver to compute the adjoint, which could then have been available to solve for the stationary flow solution. Finally, perturbing the angle of attack by ±0.01° and calculating a central difference of the averages of the lift over the range of iterations plotted in Figure (3-9) gives a sensitivity of $0.648\, rad^{-1}$, approximately 16% greater than the stationary point sensitivity.

In order to determine the sources of error in the sensitivity when the iterative solvers are used, the difference between the iterate and stationary point sensitivities is decomposed

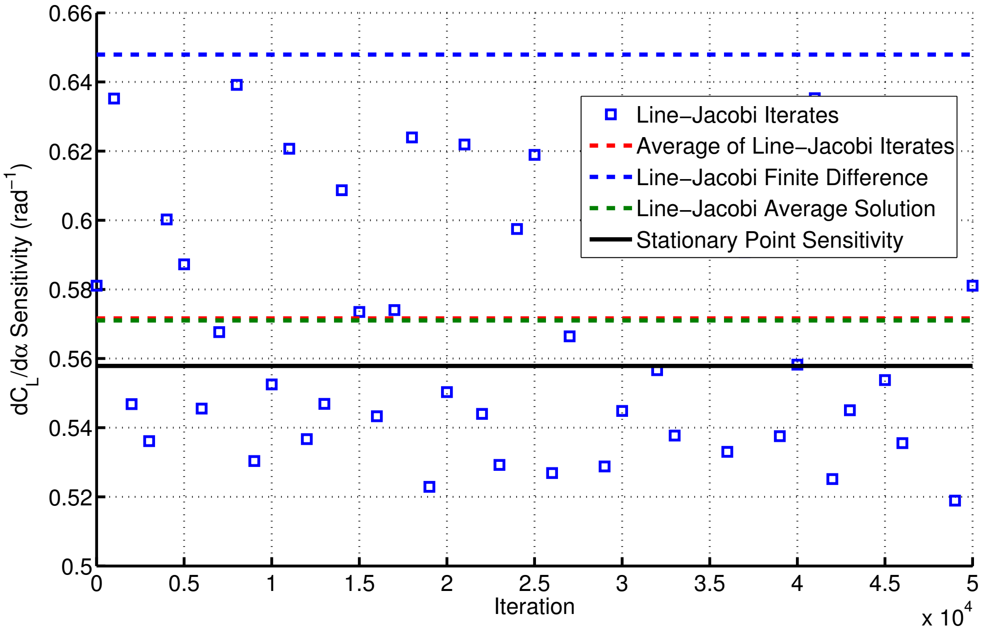


Figure 3-9: $dC_L/d\alpha$ comparison, $\alpha = 9°$

into four terms:

$$\begin{aligned}\frac{d\mathcal{J}(\boldsymbol{u}_n;\beta)}{d\beta} - \frac{d\mathcal{J}(\boldsymbol{u}_{ss};\beta)}{d\beta} = &\left(g_\beta(\boldsymbol{u}_n;\beta) - g_\beta(\boldsymbol{u}_{ss};\beta)\right) \\ &- \boldsymbol{\psi}_{ss}^T\left(\boldsymbol{f}_\beta(\boldsymbol{u}_n;\beta) - \boldsymbol{f}_\beta(\boldsymbol{u}_{ss};\beta)\right) \\ &- \left(\boldsymbol{\psi}_{\boldsymbol{u}_n} - \boldsymbol{\psi}_{ss}\right)^T \boldsymbol{f}_\beta(\boldsymbol{u}_{ss};\beta) \\ &- \left(\boldsymbol{\psi}_{\boldsymbol{u}_n} - \boldsymbol{\psi}_{ss}\right)^T \left(\boldsymbol{f}_\beta(\boldsymbol{u}_n;\beta) - \boldsymbol{f}_\beta(\boldsymbol{u}_{ss};\beta)\right)\end{aligned} \tag{3.5}$$

Figure (3-10) shows the four contributing error terms for the same line-Jacobi iterate snapshots shown in Figure (3-9). Each term is normalized by the stationary point sensitivity. The adjoint error term, $\left(\boldsymbol{\psi}_{\boldsymbol{u}_n} - \boldsymbol{\psi}_{ss}\right)^T \boldsymbol{f}_\beta(\boldsymbol{u}_{ss};\beta)$, causes almost the entire error, with contributions from the other terms negligible in comparison. This suggests that a correction method that depends on the contribution of the adjoint variation being small will not provide an accurate correction, and that computing an accurate adjoint is paramount.

### 3.3.2 10° Angle of Attack

For the same freestream conditions, $Ma = 0.5$ and $Re = 1500$, a stationary point solution of the NACA 0012 airfoil can be computed at 10° angle of attack. This stationary point solution is qualitatively identical to the stationary point at $\alpha = 9°$ shown in Figure (3-3).

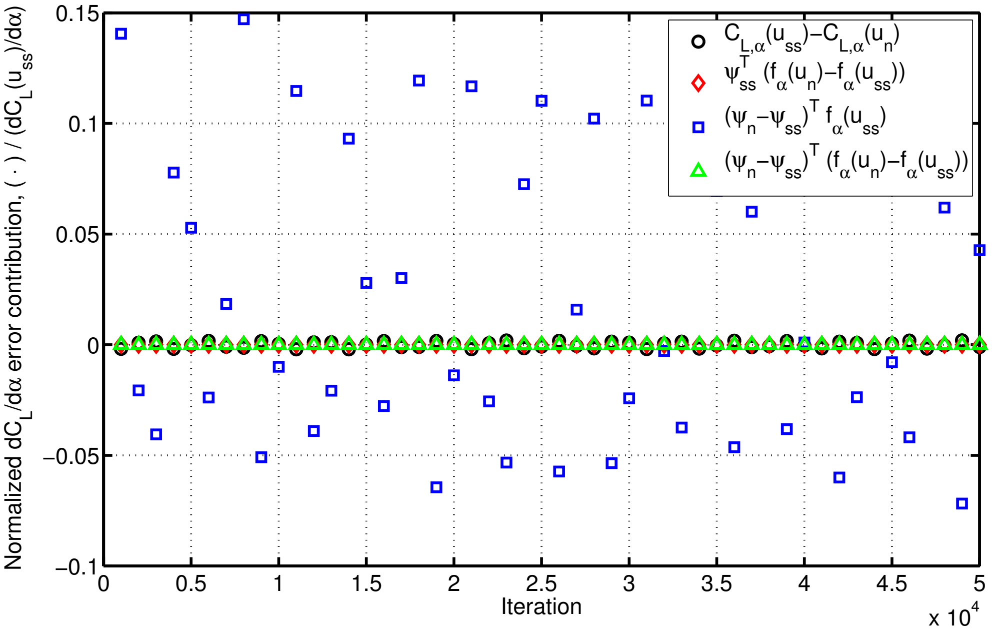


Figure 3-10: Time-inaccurate sensitivity error terms, normalized by stationary point sensitivity, $\alpha = 9^\circ$

The eigenvalues for the line-preconditioned linear system computed at the stationary point are shown in Figure (3-11). In comparison with the eigenvalues at $\alpha = 9^\circ$, the number of unstable eigenvalues (denoted by the squares) has increased, indicating that the instability has increased with angle of attack. The increase in time-inaccurate unsteadiness is verified in Figure (3-12): the momentum and entropy snapshots show a visible disturbance in the wake and the largest magnitude of the momentum standard deviation is three times the $\alpha = 9^\circ$ value. When the weaker block-Jacobi solver is used, the increase in time-inaccurate unsteadiness is even more pronounced. Figure (3-13d) shows a large area of unsteadiness approximately one chord downstream. Even though this is not a time-accurate simulation, the x-momentum and entropy snapshots now clearly indicate the shedding of vortices into the wake.

The residual and $C_L$ iteration histories in Figure (3-14) also show the increase in time-inaccurate unsteady behavior. The residual equilibrates at a higher value for either the block- or line-Jacobi iterative solver. For the line-Jacobi solver, the time-inaccurate standard deviation of $C_L$ is approximately 3.0% of the stationary point $C_L$, twice the standard deviation with the same solver at $\alpha = 9^\circ$.

At 9° angle of attack, the sensitivities computed at individual iterate solutions vary by as much as 15% from the stationary point sensitivity, but the averaging methods provide

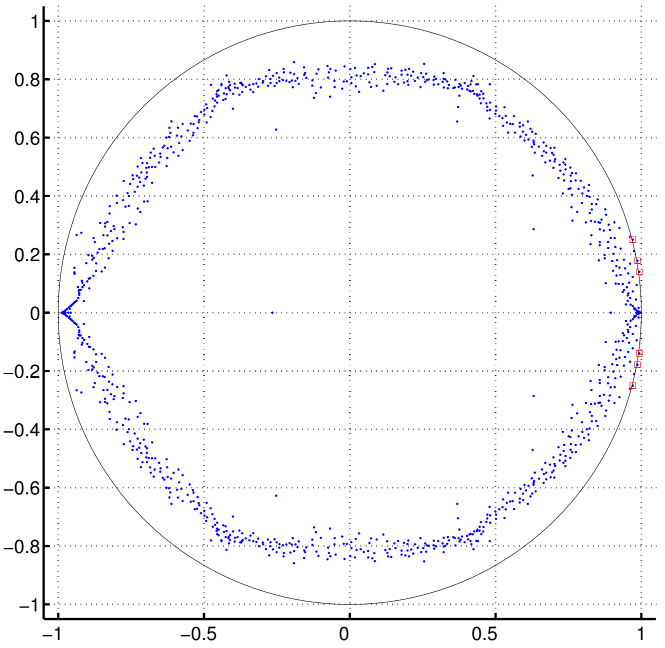
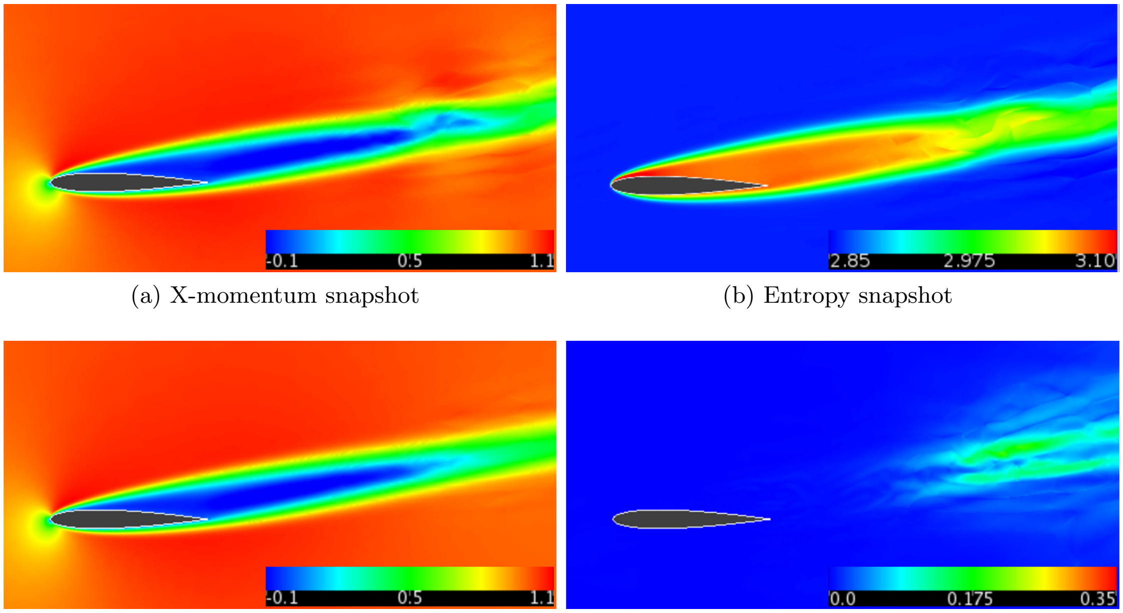


Figure 3-11: Eigenvalues of line-preconditioned iterative system $(I - M_P^{-1}\boldsymbol{R_U})$ at stationary point ($Ma = 0.5$, $Re = 1500$, $\alpha = 10°$, 4 processors)

sensitivities within 2.3%. One might expect that at $\alpha = 10°$ by doubling the variation in $C_L$ that the unsteadiness in the sensitivities would increase by a similar amount. This assumption is investigated in Figure (3-15). At $\alpha = 10°$ the $dC_L(\boldsymbol{u}_n)/d\alpha$ varies from 0.175 to 1.318 $rad^{-1}$, a significantly larger range than computed at $\alpha = 9°$. The average of these iterate sensitivities gives 0.760 $rad^{-1}$, again significantly different than the stationary point sensitivity of 0.499 $rad^{-1}$. When the GMRES linear solver is used to calculate the output sensitivity of the iteration average of the line-Jacobi solutions, the sensitivity is 0.707 $rad^{-1}$. Finally, a central difference of $C_L(\boldsymbol{u}_n)$ with ±0.01° angle of attack perturbations gives a sensitivity of 1.015 $rad^{-1}$, more than twice the stationary point sensitivity.

## 3.4 Time-Accurate Simulations

The flowfield about the airfoil when calculated using a time-accurate fourth-order ESDIRK scheme shows a markedly different character than the stationary point solution from the time-inaccurate iterative approach. Figure (3-16) shows the unsteady behavior of the output for a case started near the stationary point solution, the same initial condition as was used for the line-Jacobi iterative method in the time-inaccurate analysis. A snapshot of that



(a) X-momentum snapshot

(b) Entropy snapshot

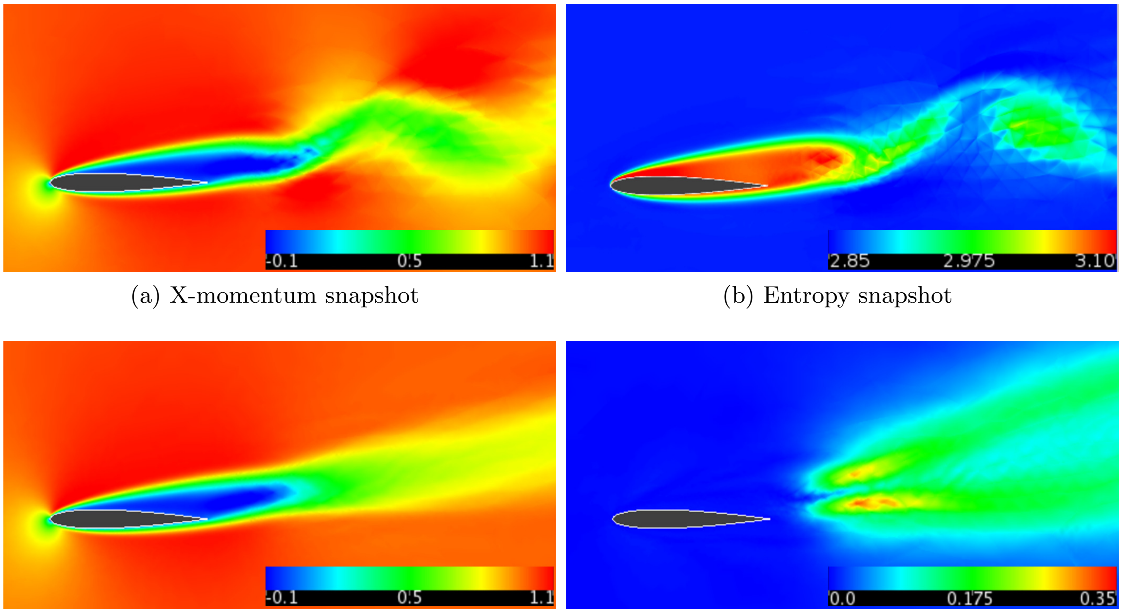
(c) X-momentum mean

(d) X-momentum standard deviation

Figure 3-12: NACA 0012, line-Jacobi iterative solver ($Ma = 0.5$, $Re = 1500$, $\alpha = 10°$)

flow, as well as the mean X-momentum and the standard deviation of the X-momentum state are shown in Figure (3-17). The output suggests, and the snapshots clearly show, that the time-accurate flow has a series of alternating vortices shedding into the wake. This difference is reflected in the lift, almost doubling it compared to the stationary point. Further comparison of the time-accurate behavior against stationary point solutions over an angle of attack sweep shows that the time-accurate unsteady behavior is not only dominant at $\alpha = 10°$, but that unsteadiness begins somewhere between $\alpha = 4°$ and $\alpha = 5°$, shown in Figure (3-18). A time-accurate simulation is considered steady when the $L_2$ norm of the spatial residual (or equivalently, the temporal term) reaches and remains below the residual convergence criteria of Section 2.3. To compute the stationary point solutions, the ILU preconditioned GMRES solver was used to compute all angles of attack in the sweep. Alternately, when the element block-Jacobi iterative solver was employed it converged below $\alpha = 8°$, while line-Jacobi across 4 processors converged below $\alpha = 9°$. This implies that the ability to achieve a stationary point solution does not guarantee that no physical unsteady behavior exists or, if it does exist, that it is even small.

Given the significant difference between the stationary point and time-accurate solutions, reliable sensitivity analysis for flows that could exhibit unsteadiness must clearly be based on a time-accurate approach. To demonstrate this time-accurate adjoint analysis, the unsteady



(a) X-momentum snapshot

(b) Entropy snapshot

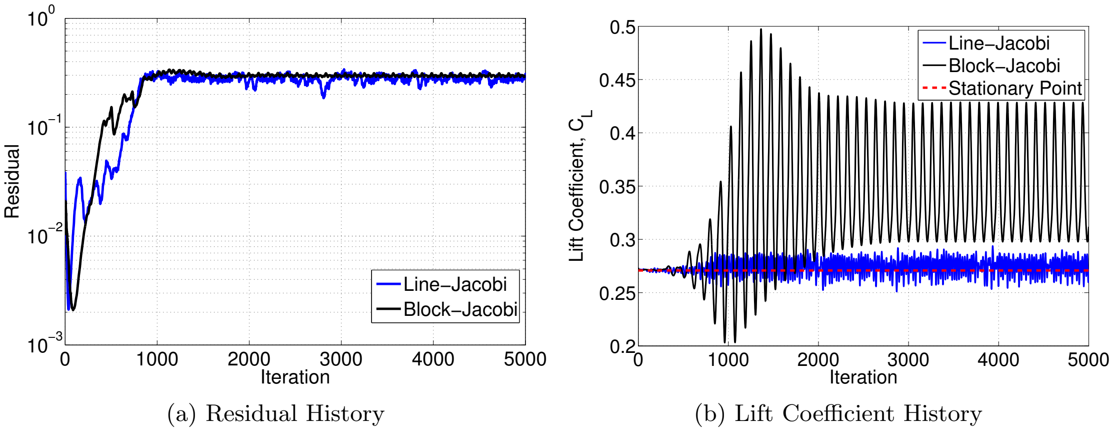
(c) X-momentum mean

(d) X-momentum standard deviation

Figure 3-13: NACA 0012, block-Jacobi iterative solver ($Ma = 0.5$, $Re = 1500$, $\alpha = 10°$)

adjoint equations were solved at 10° angle of attack using a fourth-order ESDIRK time integration scheme. As opposed to using the initial condition near the stationary point, the solution at $t = 100$ was chosen (eliminating the initial transient in the primal, leaving an equilibrium oscillation). Then the flow was integrated over approximately 94 cycles starting from this condition to $t = 200$. Finally, the adjoint was integrated backward in time. The convergence history of the time-accurate lift and sensitivity integrand are shown in Figure (3-19), along with the running sensitivity integral, $\frac{1}{t_f - t}\int_t^{t_f} S(t)\,dt$, where $S(t)$ is the integrand of Eq. (2.27) at time $t$, and the finite difference results from the flow solution. The stationary point results are also shown for comparison. It was found that the adjoint-based sensitivity is $3.3737\,rad^{-1}$ and is plotted as the slope at 10° in Figure (3-18). To verify the sensitivity results, a solution was started with the same initial condition at a perturbed angles of attack, ±0.01°, and the results were finite differenced using central differencing to give an approximate sensitivity of $3.3742\,rad^{-1}$, within 0.02% of the sensitivity calculated using the adjoint. Note the transient unsteady behavior in the adjoint near $t = 200$ due to the terminal condition ($t = 200$) of the adjoint variable due to the fixed initial condition ($t = 100$) of the primal.

It must be noted that the time-average computed here is not a periodic time-average. Specifically, the problem as stated is the sensitivity of the time-average lift coefficient to



(a) Residual History

(b) Lift Coefficient History

Figure 3-14: Convergence history for block- and line-Jacobi at $Ma = 0.5$, $Re = 1500$, $\alpha = 10^\circ$

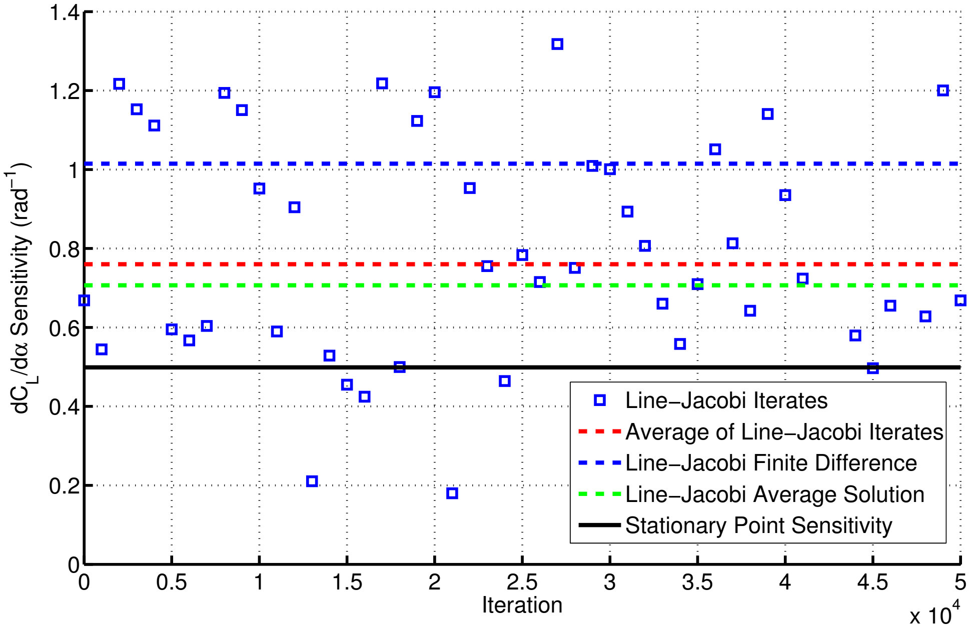
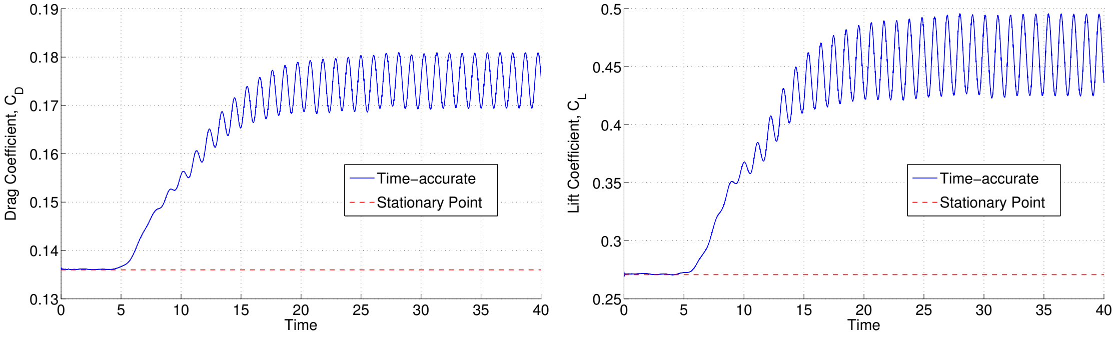
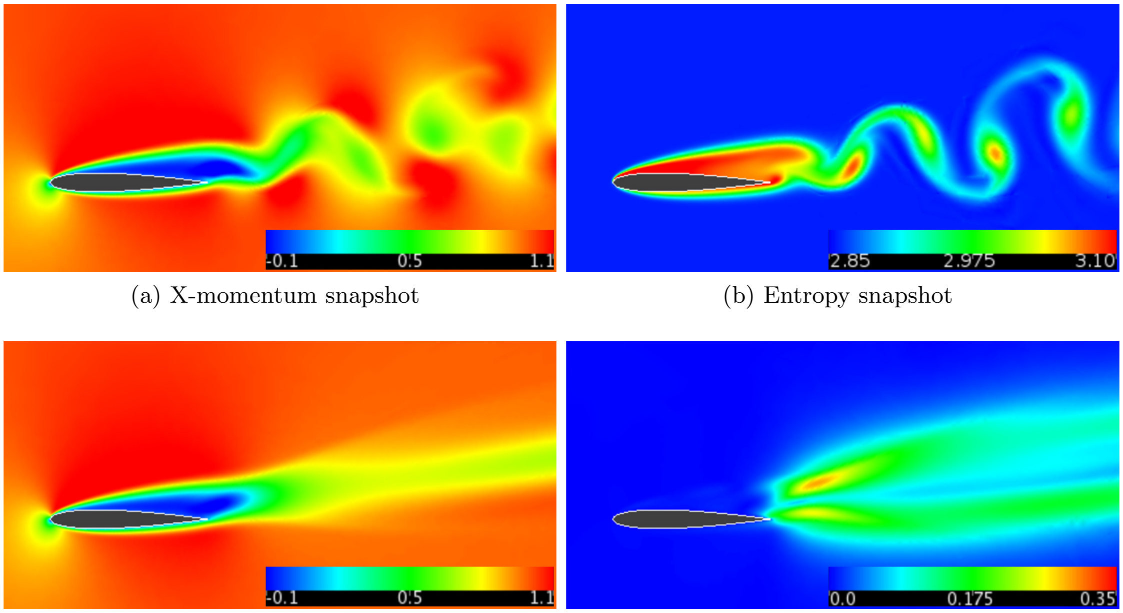


Figure 3-15: $dC_L/d\alpha$ comparison, $\alpha = 10^\circ$



(a) Drag Coefficient History (b) Lift Coefficient History

Figure 3-16: Force output for unsteady NACA 0012 at $Ma = 0.5$, $Re = 1500$, $\alpha = 10°$



(a) X-momentum snapshot (b) Entropy snapshot

(c) X-momentum mean (t=100. . . 200) (d) X-momentum standard deviation (t=100. . . 200)

Figure 3-17: NACA 0012, time-accurate solution ($Ma = 0.5$, $Re = 1500$, $\alpha = 10°$)

angle of attack for a problem with a particular initial condition (an unsteady snapshot at the baseline angle of attack). The sensitivity will then include some component of dependence on this initial condition, and as discussed in the next chapter, on the overlap of the periodic output with the integration span used to compute the average.

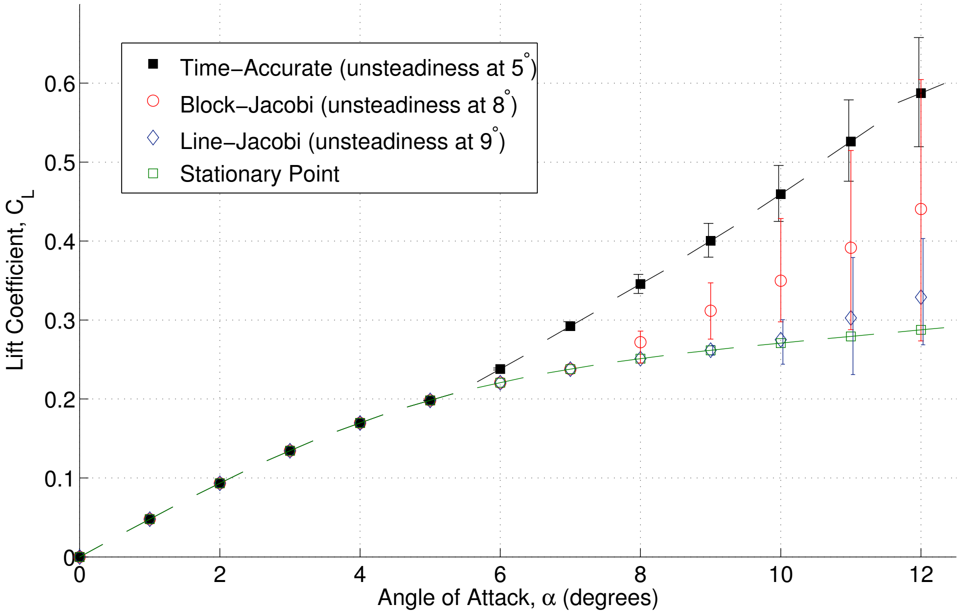


Figure 3-18: NACA 0012 steady and time-accurate unsteady lift slope curve (dashes show $dC_L/d\alpha$; error bars show min and max)

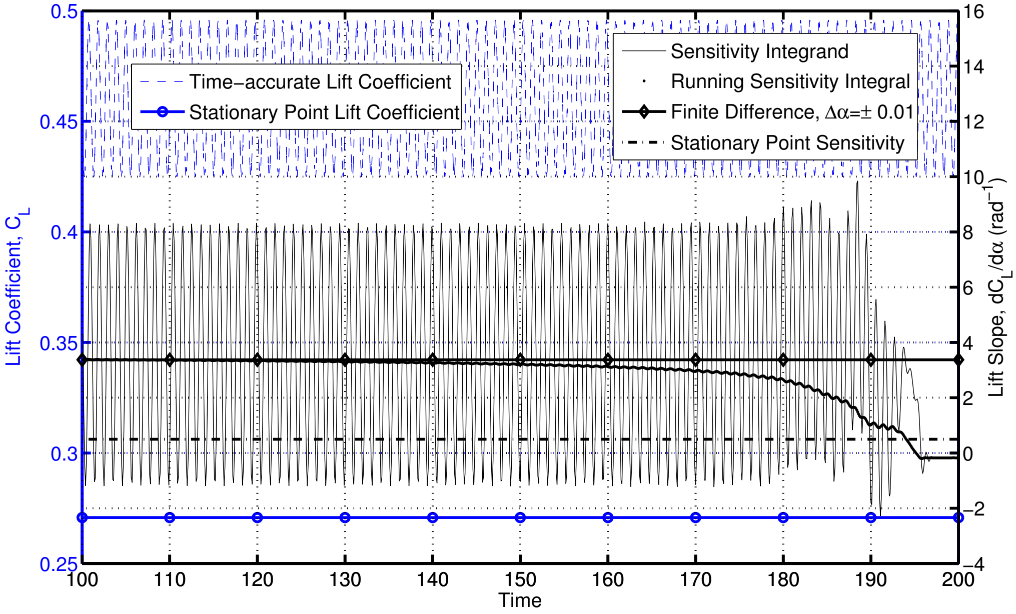


Figure 3-19: $dC_L/d\alpha$ unsteady sensitivity convergence history ($Ma = 0.5$, $Re = 1500$, $\alpha = 10°$)

# Chapter 4

# Sensitivity Analysis of Limit Cycle Oscillations

## 4.1 Mathematical Formulation

To investigate the behavior of the time-average output and sensitivity, it is useful to define a function $h$ such that

$$h(\tau, \beta) = g(\tau\, T(\beta), \beta), \quad g(t, \beta) = h(t/T(\beta), \beta),$$

then $h(\tau, \beta)$ has a period of 1, independent of $\beta$, at the expense of $\beta$ dependence in both arguments when integrating over $t$. Because $h$ is periodic with period 1,

$$\int_0^1 h_{,\tau}(\tau, \beta)\, d\tau = 0. \tag{4.1}$$

With $h$, the objective function can be represented as

$$\mathcal{J}(\beta) = \int_0^1 h(\tau, \beta)\, d\tau$$

and its derivative as

$$\mathcal{J}'(\beta) = \int_0^1 h_{,\beta}(\tau, \beta)\, d\tau$$

### 4.1.1 Breakdown of Square Windowing

The square window time-average, $\mathcal{J}_s(\beta, M)$, in Eq. (1.2) can be written as

$$\mathcal{J}_s(\beta, M) = \frac{1}{M}\int_0^{NT} g(t,\beta)\,dt - \frac{1}{M}\int_M^{NT} g(t,\beta)\,dt$$

where hereafter $N \equiv \lceil M/T \rceil$. The error in $\mathcal{J}_s$ is then

$$|\mathcal{J}_s(\beta, M) - \mathcal{J}(\beta)| \leq \frac{T}{M}\max|g|$$

which converges at $\mathcal{O}(M^{-1})$ as $M$ increases. If $M$ is an integer multiple of the $T$, then $\mathcal{J}_s(\beta, M) = \mathcal{J}(\beta)$ for any $N$. The sensitivity of $\mathcal{J}_s$ is

$$\mathcal{J}_s'(\beta, M) = \frac{1}{M}\int_0^{M} g_{,\beta}(t,\beta)dt.$$

Since

$$g_{,\beta}(t,\beta) = \frac{d}{d\beta}h(t/T(\beta),\beta) = -\frac{t\,T'}{T^2}h_{,\tau}(t/T,\beta) + h_{,\beta}(t/T,\beta),$$

integration by parts on the first term yields

$$\mathcal{J}_s'(\beta, M) = \frac{1}{M}\int_0^{M} h_{,\beta}(t/T,\beta)\,dt + \frac{T'}{T}\left[\mathcal{J}_s(\beta, M) - h(M/T,\beta)\right]. \tag{4.2}$$

In the limit as $M \to \infty$, the error in $\mathcal{J}_s'$ is given by

$$\lim_{M\to\infty}\left(\mathcal{J}_s'(\beta, M) - \mathcal{J}'(\beta)\right) = \frac{T'}{T}\left[\mathcal{J}(\beta) - h(M/T,\beta)\right]. \tag{4.3}$$

If $T$ is independent of $\beta$, the approximation to the sensitivity converges to the periodic value, and is exact for $M$ an integer multiple of $T$. In general, the deviation of the instantaneous output from its time-average is nonzero and for problems with nonzero dependence of the period on the parameter, $T'(\beta) \neq 0$, the error is nonzero. Moreover, because the error term in Eq. (4.3) is dependent on $\beta$, $\mathcal{J}_s'$ will vary relative to $\mathcal{J}'$ as $\beta$ is changed. This can introduce additional zeros of $\mathcal{J}_s'$, reflecting local extrema of $\mathcal{J}_s$ not present in $\mathcal{J}$, which hinder the optimization of $\mathcal{J}$ with respect to $\beta$.

Transformed into the frequency domain, a periodic function can be expressed as a combination of Dirac delta functions at $\omega_0 = 0$ (the time average) and at harmonics of the period at $\omega_i = 2\pi i/T\,\forall i \in \mathbb{N}$ (unsteady terms). An unweighted time average over a fixed span is equivalent to convolving the output, $g$, with a square window and dividing by the window span, M. To do this, the Fourier transform of the output is multiplied by the Fourier transform of a square function, i.e., a *sinc* function:

$$\widehat{w}_s(t/M) = \frac{M}{\sqrt{2\pi}} sinc\left(\frac{M\omega}{2\pi}\right) = \begin{cases} \frac{1}{\omega}\sqrt{\frac{2}{\pi}} sin\left(\frac{M\omega}{2}\right) & \omega \neq 0 \\ M/\sqrt{2\pi} & \omega = 0 \end{cases} \tag{4.4}$$

where the hat, $\widehat{(\cdot)}$, signifies the Fourier transform. When the square window spans integer periods, the delta functions of the periodic output signal correspond to zeros of the *sinc* function and only the time-average contribution remains. However, when a parameter of the dynamical system changes the period of oscillation, the location of the harmonics "slide" outwards (inwards) as the period is decreased (increased), see Figure (4-1). The output sensitivity, $\mathcal{J}'$, is given by the inverse Fourier transform of the derivative of the $\omega_0$ component of the periodic output with respect to $\beta$. Convolution of the output with the square window, $w_s$, includes contributions from the harmonics, $\omega_i$. At the harmonics of the output, the value of $w_s$ is zero, but the derivative of the *sinc* function in Eq. (4.4) with respect to $\omega$ is nonzero. This causes the derivative of the square-window convolved output, $d(\widehat{w}_s(\omega_i)\widehat{g}(\omega_i))/d\omega$, to also be nonzero. Through $dT/d\beta \neq 0 \rightarrow d\omega_i/d\beta \neq 0$ and the chain rule, this results in a nonzero and nondiminishing erroneous contribution to $\mathcal{J}_s'(\beta)$.

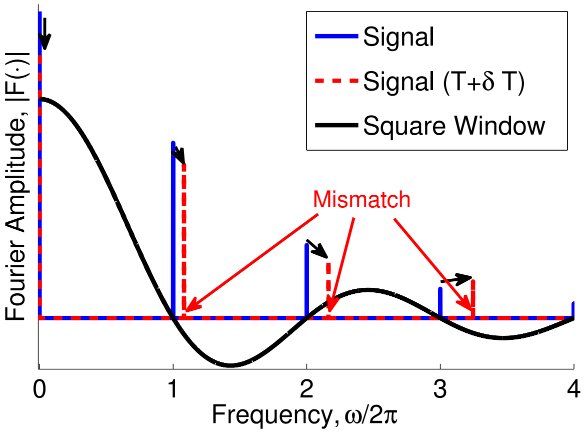


Figure 4-1: Magnitude of Fourier transform of square window and a periodic signal ($T = 1$)

### 4.1.2 Windowing

The nonconvergence of the time-average sensitivity can be addressed by applying a time-varying weighting to the instantaneous objective function. This weighting is termed a *window* or *apodization* function and is a commonly used tool in signal processing. The windowing function satisfies

$$w(\tau) = 0 \text{ for } \tau \notin (0,1), \qquad \int_0^1 w(\tau)\, d\tau = 1. \tag{4.5}$$

The definition of the window function outside $(0, 1)$ is necessary when $M$ is a noninteger multiple of $T$. By decomposing the integral in Eq. (1.3) into a summation over multiple periods, the error in $\mathcal{J}_w$ can be expressed as

$$\mathcal{J}_w(\beta, M) - \mathcal{J}(\beta) = \frac{1}{T}\int_0^T g(\theta,\beta)\mathcal{A}(\theta, M) d\theta \tag{4.6}$$

where

$$\mathcal{A}(\theta, M) = \frac{T}{M}\sum_{i=0}^{N-1} w\left(\frac{\theta + iT}{M}\right) - 1. \tag{4.7}$$

with $\theta \in [0, 1]$. Recall $N \equiv \lceil M/T \rceil$ which allows $t = \theta + (N-1)T > M$. Because $w(\tau) = 0$ for $\tau \notin (0, 1)$, the integral is unchanged. Slightly modifying Eq. (4.2), the sensitivity can be approximated with

$$\begin{aligned} \mathcal{J}'_w(\beta, M) &= \frac{1}{M}\int_0^M g_\beta(t,\beta)\, w(t/M)\, dt \\ &= \frac{1}{M}\int_0^M \left\{ -\frac{t\,T'}{T^2} h_{,\tau}(t/T,\beta)\, w(t/M) + h_{,\beta}(t/T,\beta)\, w(t/M) \right\} dt. \end{aligned} \tag{4.8}$$

Introducing

$$\mathcal{B}(\theta, M) = \sum_{i=0}^{N-1} \frac{\theta + iT}{M} w\left(\frac{\theta + iT}{M}\right) - C(N) \tag{4.9}$$

with $C(N) \in \mathbb{R}$ arbitrary due to Eq. (4.1), the error in the sensitivity can now be expressed as

$$\mathcal{J}'_w(\beta, M) - \mathcal{J}'(\beta) = \frac{1}{T}\int_0^T h_{,\beta}(\theta/T,\beta)\mathcal{A}(\theta,M)d\theta - \frac{T'}{T^2}\int_0^T h_{,\tau}(\theta/T,\beta)\mathcal{B}(\theta,M)d\theta. \quad (4.10)$$

**Theorem 1.**

$$|\mathcal{J}_w(\beta, M) - \mathcal{J}(\beta)| \leq \max|h| \max|\mathcal{A}| \quad (4.11)$$

$$\left|\mathcal{J}'_w(\beta, M) - \mathcal{J}'(\beta)\right| \leq \frac{T'}{T}\max|h_{,\tau}|\max|\mathcal{B}| + \max|h_{,\beta}|\max|\mathcal{A}| \quad (4.12)$$

*Proof.* The proof is trivial by bounding the integrals of Eq. (4.6) and Eq. (4.10) with the maxima of their integrands.

□

The benefit of non-square windows can be further clarified by examining their effect in the frequency domain. Using windowing functions other than a square window allows for two improvements:

1. increased rate at which the Fourier transform of the window converges to zero with increasing frequency

2. increased smoothness near the harmonic frequencies of the output, specifically, increasing the order of lowest order nonzero derivative at the harmonics.

The Fourier transform of the square window converges to zero with $\mathcal{O}(1/\omega)$ as $\omega$ increases and has nonzero derivatives at its zeros. Long-time windows specifically address improvement (1) to speed convergence as the number of periods spanned increases, with no attempt to match the period of the output. Short-time windows do attempt to match the period for a small number of periods and are designed to address improvement (2) when the period used to design the window is approximate.

### 4.1.3 Long-Time Windowing

We propose *long-time* windows where $M$ is allowed to become large relative to $T$. A benefit of taking a long-time window is that there is no need to accurately determine the period.

The convergence of long-time windows with the span of the window is governed by the convergence of $\mathcal{A}$ and $\mathcal{B}$ to the integrals they approximate, which in turn is driven by the smoothness of the function $w(\tau)$ in the interior, $\tau \in (0, 1)$, and at the boundaries $\tau = 0$ and $\tau = 1$. The convergence analysis using Fourier transforms is more straightforward over an integer, $N$, number of periods, so a *periodic extension* of $w$ is defined to fill in the remainder of the period outside of the span, i.e., for $M < t \leq NT$.

**Theorem 2.** *For a periodic function $v$ of class $C^\ell$ on the interval $[0, 1]$, the trapezoidal rule with $N$ intervals, $I_N(v)$, will converge at*

$$I_N(v) = \int_0^1 v(\tau)\, d\tau + \mathcal{O}(N^{-p}) \tag{4.13}$$

*where*

$$p = \begin{cases} 1 & \ell = -1 \\ \ell + 1 & \ell \geq 0, \ell \text{ even} \\ \ell + 2 & \ell > 0, \ell \text{ odd} \end{cases} \tag{4.14}$$

*Proof.* The convergence rate of trapezoidal rule quadrature of a periodic function can be determined using the convergence of the Fourier coefficients for a periodic function, see Boyd[15], chapter 2. Note that the convergence rate for an infinitely smooth (analytic or not) periodic function is exponential. □

**Theorem 3.** *For a window $w(\tau)$ of class $C^\ell$ and with $N = \lceil M/T \rceil$*

$$|\mathcal{J}_w(\beta, M) - \mathcal{J}(\beta)| \leq \max |h| \mathcal{O}(N^{-p}) \tag{4.15}$$

*with $p$ defined by Eq. (4.14)*

*Proof.* The sum in $\mathcal{A}(\theta, M)$ is a trapezoidal rule quadrature of $w(\tau)$ for $\tau \in (0, 1)$. In order to integrate over noninteger numbers of periods with $M \neq NT$, recall from the requirements of a generic window that $w(\tau) = 0 \, \forall \tau \notin (0, 1)$. If we treat the extended $w(\tau)$ as periodic with period $NT$, the rate of convergence of the trapezoidal rule numerical quadrature is given by Theorem (2) and depends on the smoothness of the extended $w(\tau)$. In the asymptotic limit of large $M$ with $\theta \in (0, T)$:

$$\mathcal{A}(\theta, M) = \frac{T}{M} \sum_{i=0}^{N-1} w\left(\frac{\theta + iT}{M}\right) - 1 = \int_0^1 w(\tau)\, d\tau + \mathcal{O}(N^{-p}) - 1 = \mathcal{O}(N^{-p}) \tag{4.16}$$

Combining this result with Eq. (4.11), proves the theorem.

□

**Theorem 4.**

$$\left|\mathcal{J}'_w(\beta, M) - \mathcal{J}'(\beta)\right| \leq \left|\frac{T'}{T}\right| \max|h_{,\tau}|\, \mathcal{O}(N^{-(p-1)}) + \max|h_{,\beta}|\, \mathcal{O}(N^{-p}) \tag{4.17}$$

*Proof.* As with $\mathcal{A}(\theta, M)$, the sum in $\mathcal{B}(\theta, M)$ is a periodic trapezoidal quadrature, this time of $\tau\, w(\tau)$ for $\tau \in (0, 1)$. For large $M$ (and hence large $N$) with $\theta \in (0, T)$:

$$\mathcal{B}(\theta, M) = \sum_{i=0}^{N-1} \frac{\theta + iT}{M} w\left(\frac{\theta + iT}{M}\right) - C(N) \tag{4.18}$$

$$= N\left[\int_0^1 \tau\, w(\tau)\, d\tau + \mathcal{O}(N^{-p})\right] - C(N) \tag{4.19}$$

The arbitrary $C$ can be selected to cancel the integral, giving

$$\mathcal{B}(\theta, M) = \mathcal{O}(N^{-(p-1)}).$$

Combining the result with Eq. (4.12) gives the first expression on the right-hand side of Eq. (4.17). Combining the derivative error Eq. (4.10) with theorem (3) gives the second term on the right-hand side of Eq. (4.18), and the theorem is proven.

□

Possible choices of $w(\tau)$ conforming to Eq. (4.5) for use with the windowed objective function Eq. (1.3) include the Hann window ($C^1 \rightarrow p = 3$)

$$w(\tau) = 1 - \cos 2\pi\tau, \qquad \tau \in (0, 1)$$

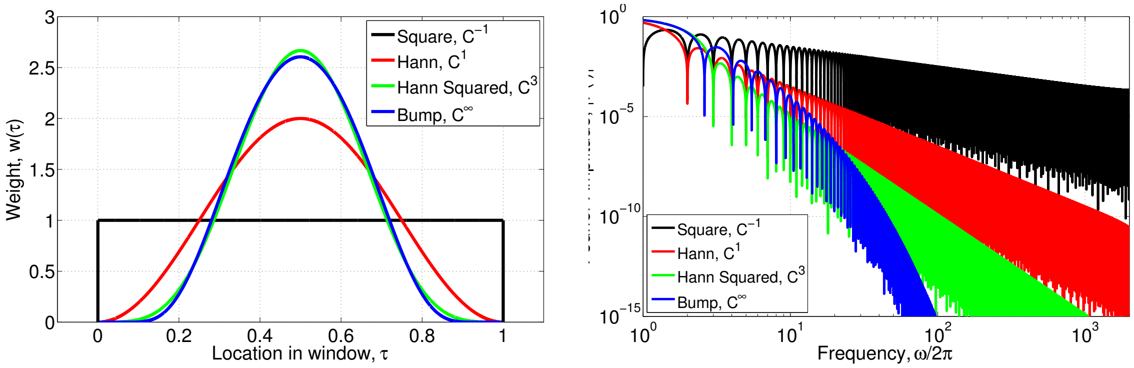
the Hann-squared window ($C^3 \rightarrow p = 5$)

$$w(\tau) = \frac{2}{3}(1 - \cos 2\pi\tau)^2, \qquad \tau \in (0, 1)$$

and the bump window ($C^\infty \to p = \infty$)

$$w(\tau) = \frac{1}{A} e^{-1/(\tau-\tau^2)}, \qquad \tau \in (0,1)$$

where $A$ is the appropriate area under the window and infinite $p$ gives exponential convergence. These windows were selected as they demonstrate differing convergence rates, all have compact support, and are zero at their endpoints. This facilitates the use of the periodic extension for $M \neq NT$, otherwise there is a $C^0$ discontinuity at the endpoints as part of the extension. Examination of long-time windows using the frequency domain in Figure (4-2b) shows the magnitude of $\widehat{w}(\omega)$ decreasing at the expected rates with increasing $\omega$. As the window span is increased, the frequencies that correspond to the harmonics of the output are moved proportionally to the right, are attenuated more, and the errors in the windowed time-average output and sensitivities are decreased accordingly.



(a) Time domain

(b) Frequency domain ($|F(\cdot)| = \widehat{w}(\omega)$)

Figure 4-2: Long-time windows

The output window of Barth[7] is a modified square window, with one-period sized $C^\infty$ caps on both ends. The span of the window is modified by increasing the span of the enclosed square window. This window is $C^\infty$ smooth, but the convergence rates previously derived are not applicable because the Barth window is not self-similar with respect to $\tau$. Thus, the limits employed in Eq. (4.16) and Eq. (4.18) behave differently and the convergence rate reduces to that of the square window (the limit behavior most similar to the Barth window), albeit with a lower constant factor than the square window.

### 4.1.4 Short-Time Windowing

The second type of output window we classify as a *short-time* window, that is applied over a small, known number of periods. Short-time windows offer the possibility of decreasing the span of integration, as the output weighting is nonzero for only a small number of periods. These windows are designed such that $\mathcal{A} = \mathcal{B} = 0$. For $M = T$, $\mathcal{A} = 0$ requires $w = 1$, the square window. However, $w = 1 \Rightarrow \mathcal{B} \neq 0$, and the error in $\mathcal{J}'_w$ is nonzero. For $M = \{nT \,:\, n \in \mathbb{N}\}$, we choose symmetric window functions from period-wise polynomials of degree $n-1$. Enforcing the conditions $\mathcal{A} = \mathcal{B} = 0$, $C^{n-2}$ continuity between polynomials, and symmetry about $\tau = 1/2$, gives a linear system of coefficients which can be solved for the corresponding window. The resulting short-time windows up through 5 periods are shown in Figure (4-3a).

We note that the conditions to define the piecewise polynomials are equivalent to repeated convolution with the single-period square window, which has the Fourier transform given in Eq. (4.4) with $M = T$. For an $n$ period window, the resultant Fourier transform is

$$\widehat{w_n}(\tau) \;=\; \frac{1}{A(n)}\left(\widehat{w}_s(\tau)\right)^n \;=\; \frac{1}{A(n)}\left(\frac{T}{\sqrt{2\pi}}\right)^n sinc\left(\frac{\omega T}{2\pi}\right)^n \;=\; sinc\left(\frac{\omega T}{2\pi}\right)^n \tag{4.20}$$

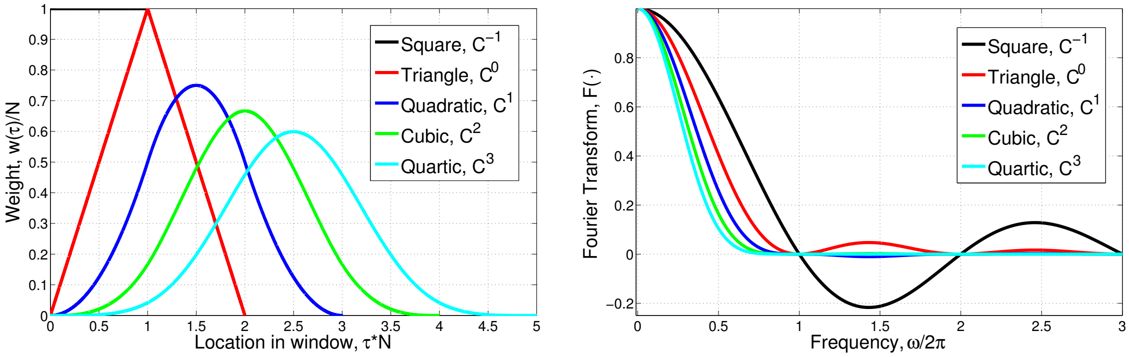
where $A(n)$ is the renormalization factor necessary to maintain unit area under the window. Demonstrated in the frequency domain in Figure (4-3b), short-time windows address improvement (2) by increasing the order of the lowest order nonzero derivative of the window with respect to frequency at the harmonics of the periodic signal. In the limit of $n \to \infty$, repeated convolution gives a Gaussian distribution, suggesting a possible long-time window. Because the Gaussian distribution is not compactly supported, we have chosen the similarly exponential 'bump' function for testing as a long-time window.

Figure (4-4) shows the contributions of each period-long segment of $\mathcal{A}$ and $\mathcal{B}$.

**Triangular Window** ($M = 2T$)

For $n = 2$, the resulting piecewise linear function is a triangle window. The objective can be expressed as

$$\mathcal{J}_2(\beta, 2T) = \frac{1}{M}\int_0^M g(t,\beta)\, w_2(t/M)\, dt$$



(a) Time domain

(b) Frequency domain ($|F(\cdot)| = \widehat{w}(\omega)$)

Figure 4-3: Short-time windows

where the triangular windowing function is

$$w_2(\tau) = \begin{cases} 4\tau & \tau \in (0, 1/2) \\ 4(1-\tau) & \tau \in (1/2, 1) \end{cases}$$

Consistent with the conditions of short-time windows, substituting the triangle window into Eq. (4.9), along with $C = 1$, gives $\mathcal{A} = \mathcal{B} = 0$ and $\mathcal{J}_2 - \mathcal{J} = \mathcal{J}_2' - \mathcal{J}' = 0$.

### Piecewise Quadratic Window ($M = 3T$)

For $n = 3$, the window is piecewise quadratic, with the windowing function given by

$$w_3(\tau) = \frac{9}{2}\begin{cases} 3\tau^2 & \tau \in (0, 1/3) \\ 6\tau(1-\tau) - 1 & \tau \in (1/3, 2/3) \\ 3(1-\tau)^2 & \tau \in (2/3, 1) \end{cases}$$

It can be shown that the piecewise quadratic window meets the condition for zero error, $\mathcal{A} = \mathcal{B} = 0$. Note that to minimize $|\mathcal{B}|$ in Eq. (4.9) let $C = 3/2$.

### Cubic Polynomial Windows and Higher

In addition to the first two short-time windows presented, larger span windows can be devised in similar fashion. The increased time span allows for increased smoothness of the window, and decreases dependence on an accurate calculation of the period. To examine

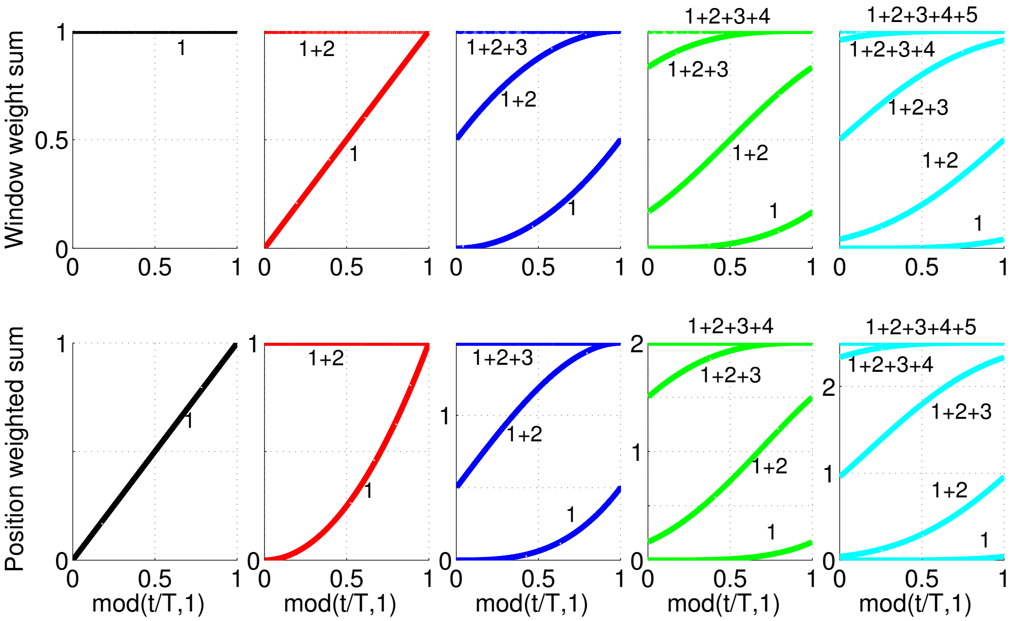


Figure 4-4: Contributions from each period to $\mathcal{A}$ and $\mathcal{B}$ for short-time windows

this further, demonstrations and comparisons of piecewise cubic ($n = 4$) and quartic ($n = 5$) windows are included.

### Convergence with Error in Period Estimate

The period may not be exactly known for short-time windows and an approximate period $T_{est} = T + \delta T$ may be used to set the window span.

**Theorem 5.** *For an approximate period $T_{est}$ with period error $\delta T = T_{est} - T$ and integration over $n$ periods, the errors in the objective function and sensitivity satisfy:*

$$|\mathcal{J}_n(\beta, n(T \pm \delta T)) - \mathcal{J}(\beta)| \leq \mathcal{O}(\delta T^n) \tag{4.21}$$

$$\left|\mathcal{J}_n'(\beta, n(T \pm \delta T)) - \mathcal{J}'(\beta)\right| \leq \mathcal{O}(\delta T^{n-1}), \tag{4.22}$$

*respectively.*

*Proof.* At the estimated harmonics of the output, computed from $T_{est}$, the multiply-convolved window Eq. (4.20) has zero derivatives with respect to the frequency, up to and including the $(n-1)^{th}$ derivative, with nonzero $n^{th}$ and higher derivatives. To determine the error in the window weight near a harmonic frequency, $\omega_i$, take a Taylor series expansion about the estimated harmonic frequency, $\omega_{i,est} \equiv 2\pi i/T_{est}$, with $\omega_i = \omega_{i,est} + \delta\omega_i$, which reduces to:

$$\begin{aligned} \widehat{w}_n(\omega_i) - \widehat{w}_n(\omega_{i,est}) &= \frac{\widehat{w}_n^{(n)}(\omega_{i,est})}{n!}(\delta\omega_i)^n + \mathcal{O}((\delta\omega_i)^{n+1}) + \dots \\ &= \mathcal{O}((\delta\omega_i)^n) \end{aligned} \tag{4.23}$$

In the limit $\delta T \to 0$, $\delta T \propto \delta\omega$. From Eq. (4.23), each harmonic's component is $\mathcal{O}((\delta T)^n)$, and the total contribution to the output error over all harmonics is of the same order, directly giving Eq. (4.21). As previously described, the error in the output sensitivity is proportional to the error introduced into the derivative of the convolved output with respect to the frequency over all $\omega > 0$ harmonics. A Taylor series for the derivative of $d\widehat{w}(\omega_i)/d\omega$ about $\omega_{i,est}$ results in:

$$\begin{aligned} \frac{d\widehat{w}_n(\omega_i)}{d\omega} - \frac{d\widehat{w}_n(\omega_{i,est})}{d\omega} &= \frac{\widehat{w}_n^{(n)}(\omega_{i,est})}{(n-1)!}(\delta\omega_i)^{n-1} + \mathcal{O}((\delta\omega_i)^n) + \dots \\ &= \mathcal{O}((\delta\omega_i)^{n-1}) \end{aligned} \tag{4.24}$$

which, similar to the output, translates directly to Eq. (4.22).

□

## 4.2 Periodic Sensitivity Calculation

The periodic sensitivity may be computed using either a tangent or adjoint differentiation of the primal problem, Eq. (2.6). The choice between the tangent and adjoint is problem dependent, typically determined by the number of inputs relative to the number of outputs: a tangent state must be computed for each input parameter, an adjoint state for each output. Additionally, the adjoint requires the storage or recalculation of the primal solution for use in the adjoint backward-time sweep[40, 121], a requirement avoided by the tangent. The tangent sensitivity was used for the test cases in this work, as it allowed the evaluation of multiple windows for a single primal and tangent forward time solution.

The tangent problem corresponding to integrating the windowed output over a span of length $M$ is:

$$\frac{d\boldsymbol{v}}{dt} + \boldsymbol{f}_{,\beta}(\boldsymbol{u};\beta) + \boldsymbol{f}_{,\boldsymbol{u}}(\boldsymbol{u};\beta)\boldsymbol{v} = 0 \qquad \forall t \in [t_{eq}, t_r + M] \tag{4.25}$$
$$\boldsymbol{v}(t_{eq}) = 0$$

where $\boldsymbol{v} \equiv \boldsymbol{u}_{,\beta}$ is the tangent variable. For problems with multiple input parameters, a tangent variable associated with each parameter must be computed. The primal state is required in Eq. (4.25), and is typically solved for simultaneously with the tangent, using $\boldsymbol{u}_{eq}$ as a fixed initial condition for the primal problem. $t_r$ is a relaxation time defined such that $t \in (t_{eq}, t_r)$ is sufficient time that periodicity can be recovered given a small perturbation to the equilibrium solution at time $t_{eq}$. The relaxation time allows the tangent variable to equilibrate, and is required to remove the effects of the fixed initial condition, $\boldsymbol{v}(t_{eq}) = 0$. Rather than beginning calculation of the tangent state at time $t_{eq}$, the primal and tangent states may also be equilibrated simultaneously: the strict requirement is that both the primal state, $\boldsymbol{u}$, and the tangent state, $\boldsymbol{v}$, have reached equilibrium at time $t_r$. Once the primal and tangent states are computed to time $t_r+M$, the output sensitivity to a parameter is given by:

$$\frac{d\mathcal{J}_w}{d\beta} = \int_{t_r}^{t_r+M} w((t-t_r)/M)\left\{g_{,\beta} + g_{,\boldsymbol{u}}\boldsymbol{v}\right\}\, dt \tag{4.26}$$

where $g(\boldsymbol{u};\beta)$ is the instantaneous objective function with the time dependence coming through $\boldsymbol{u} \equiv \boldsymbol{u}(t,\beta)$. To maintain consistency with the temporal discretization of the primal and tangent problems, the time integration of the output sensitivity should be carried out using the same timestepping scheme. If the unwindowed integrands of a temporal discretization of Eq. (4.26) are stored for each timestep, the window weighting may be applied as a post-processing step, allowing computation with several different windows using the same primal and tangent solutions.

In summary, the process to apply a window to a tangent solution, Figure (4-5), is:

1. Solve primal problem forward in time to equilibrium periodic oscillation at time $t_{eq}$ for $\boldsymbol{u}_{eq} \equiv \boldsymbol{u}(t_{eq}, \beta)$;

2. Solve primal and tangent problems forward in time to $t_r$ to equilibrate the tangent solution;

3. Continue primal and tangent problems forward in time over the span of the window to $t_r + M$;

    - Simultaneously integrate the sensitivity, or
    - Store the unweighted sensitivity integrand of Eq. (4.26) for later sensitivity calculations.

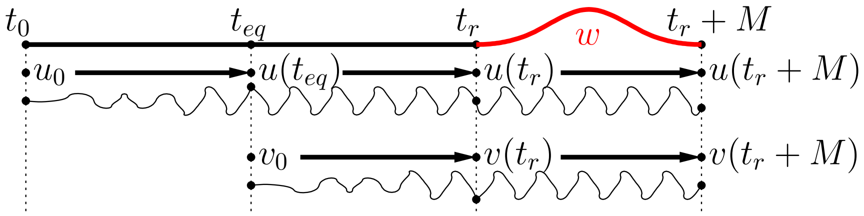


Figure 4-5: Process to compute windowed tangent sensitivity

The adjoint problem to solve in backward time is:

$$\boldsymbol{\psi}(t_r + M) = 0 \tag{4.27}$$

$$-\frac{d\boldsymbol{\psi}}{dt} + \boldsymbol{f}_{,\boldsymbol{u}}^T \boldsymbol{\psi} = w((t - t_r)/M) g_{,\boldsymbol{u}} \qquad \forall t \in [t_r, t_r + M] \tag{4.28}$$

$$-\frac{d\boldsymbol{\psi}}{dt} + \boldsymbol{f}_{,\boldsymbol{u}}^T \boldsymbol{\psi} = 0 \qquad \forall t \in [t_{eq}, t_r] \tag{4.29}$$

With a zero terminal condition, the adjoint is excited by the windowed output derivative in a backward-time solution of Equation (4.28). Equation (4.29) is continued backward in time from $t_r$ with no forcing to allow the adjoint state to equilibrate. This is analogous to the equilibration of the tangent state before the nonzero window can be applied. The output sensitivity to a parameter is:

$$\frac{d\mathcal{J}_w}{d\beta} = \int_{t_r}^{t_r+M} w((t - t_r)/M) g_{,\beta} \, dt - \int_{t_0}^{t_r+M} \boldsymbol{\psi} \boldsymbol{f}_{,\beta} \, dt$$

Unlike the tangent state, the calculation of the adjoint state is not separable from the window weighting. While the cost of the adjoint is insensitive to the number of input parameters, each adjoint state is specific to a particular output and window. The process to apply a window to an adjoint solution, Figure (4-6), is:

1. Solve primal problem forward in time to equilibrium periodic oscillation for $\boldsymbol{u}_{eq}$;

2. Continue primal problem forward in time to $t_r + M$, storing the solution;
3. Solve adjoint backward in time from $t_r + M$ to $t_r$ with nonzero forcing term from the windowed output, simultaneously integrating the sensitivity;
4. Continue adjoint backwards solve with no forcing term to allow it to equilibrate at $t_{eq}$, continuing the sensitivity calculation.

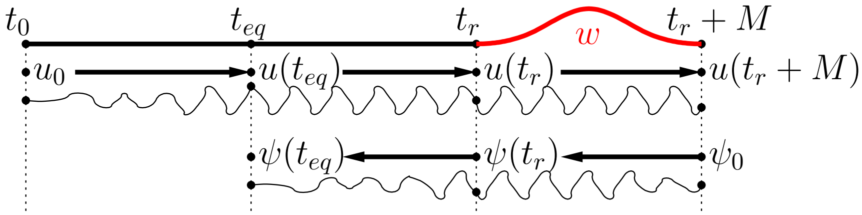


Figure 4-6: Process to compute windowed adjoint sensitivity

## 4.3 van der Pol Oscillator

### 4.3.1 Model Problem

The van der Pol oscillator is a nonlinear modification of the linear oscillator:

$$u_{tt} + \beta(u^2 - 1)u_t + u = 0 \tag{4.30}$$

which reduced to a first order system becomes:

$$\begin{bmatrix} u_1 \\ u_2 \end{bmatrix}_t + \begin{bmatrix} -u_2 \\ \beta(u_1^2 - 1)u_2 + u_1 \end{bmatrix} = 0 \tag{4.31}$$

with two states, $u_1 \equiv u$ and $u_2 \equiv u_t$. For a positive nonlinear parameter, $\beta > 0$, the oscillator exhibits a limit cycle oscillation. Moreover, the period of the limit cycle depends on the nonlinear parameter, leading to failure of square windowing calculations. The instantaneous output is given by the square of the first state, $g(t) = u(t)^2$, for which the time-average output and sensitivity with respect to $\beta$ are of interested. The resulting problem is solved using a fourth order ESDIRK timestepping scheme. The sensitivity is computed with the tangent derivative method, using the tangent of the discrete primal system. The primal solution for the first 10 periods is shown in Figure (4-7). The primal demonstrates that the

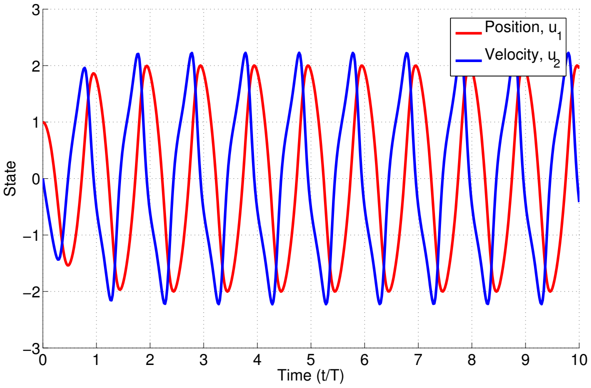


Figure 4-7: van der Pol oscillator primal solution ($\beta = 0.5$)

van der Pol oscillator equilibrates to the limit cycle extremely quickly, to machine precision within five periods.

### 4.3.2 Adjoint and Sensitivity Behavior

In order to examine the behavior of the adjoint state, the vdP was investigated for windows extending across $t \in (0, 10T)$, with a lead-in from $t = -5T$. The first adjoint state for the four long-time windows is shown in Figure (4-8). The magnitude of the oscillation of the adjoints closely follow the shape of the windows, indicating that the adjoint is highly dependent on the local excitation from the windowed output derivative (and that the primal state is insensitive to perturbations). The running sensitivities are shown in Figure (4-9). The square window has completely different behavior from the smoother windows, so much that the magnitude of the sensitivity is similar, but with opposite sign. Not shown, but increasing or decreasing the lead-in by $T/4$ flips the sign of the sensitivity, exemplifying the dependence on the overlap of the square window and the oscillation.

### 4.3.3 Results

For simulations presented here, the initial condition is $\boldsymbol{u} = (1, 0)$, with $t_r = 50T$ to equilibrate the tangent derivative. For a finite-span square window, the resulting time-average slowly converges, as seen by the output error in Figure (4-10a). The rate of convergence of the output is shown in Figure (4-10b) for several long-time windows to demonstrate the improved rate of convergence of the output versus window span as the smoothness of the output window is increased. All the computed time-averaged outputs are plotted as

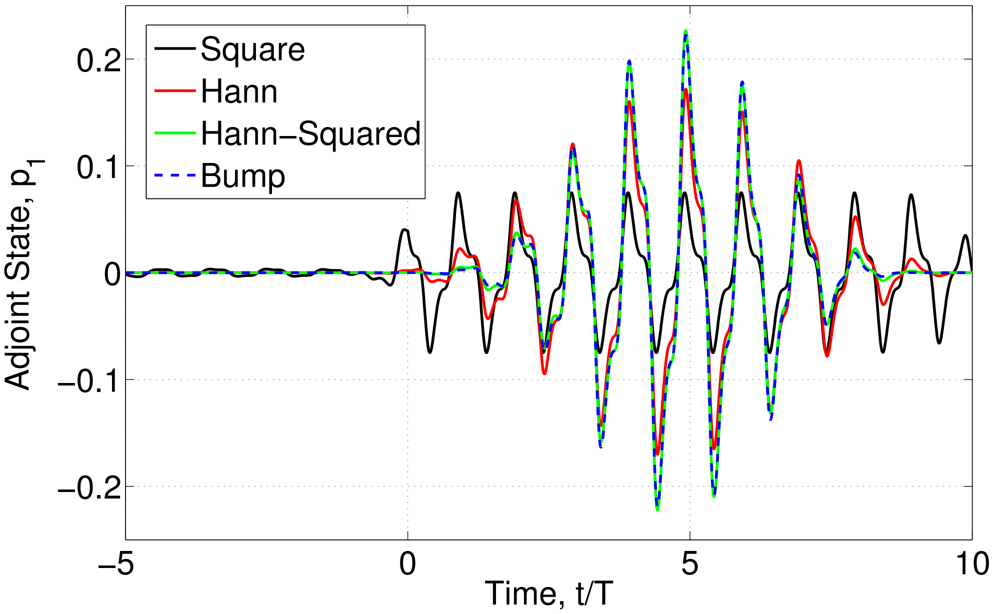
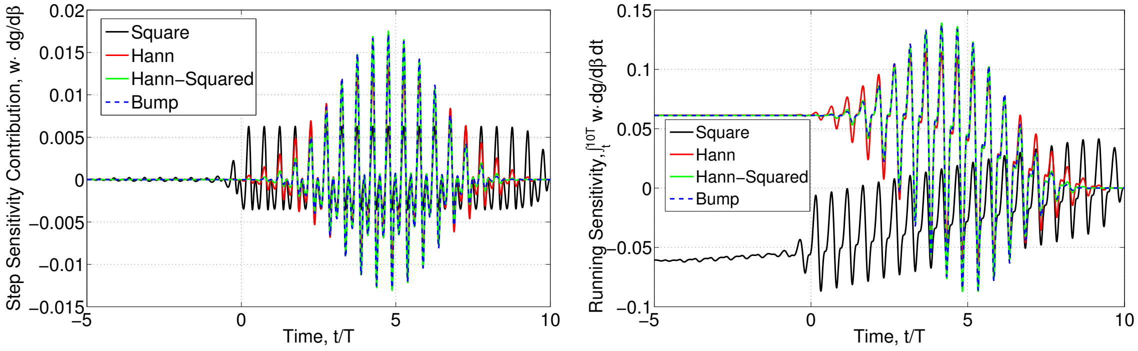


Figure 4-8: Adjoint comparison for short span vdP model problem

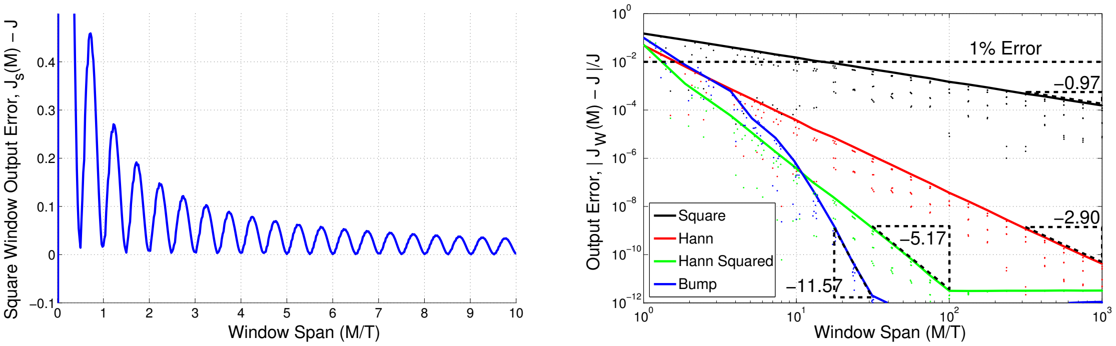


(a) Step Contributions

(b) Running Sensitivities

Figure 4-9: Sensitivity integrand and running sensitivity comparison

disconnected points in the figure. Because the error in the output is highly dependent on the phase of the window, i.e. where the window overlaps with the oscillation, there can be large variations in the error with small changes in the window span. In order to show error bounds, each point used to define the solid curves in Figure (4-10b) is the maximum over a one period range of window sizes. Because the exact output is not known, the error is computed relative to the output with the bump window at $M = 800T$. Consistent with theory,



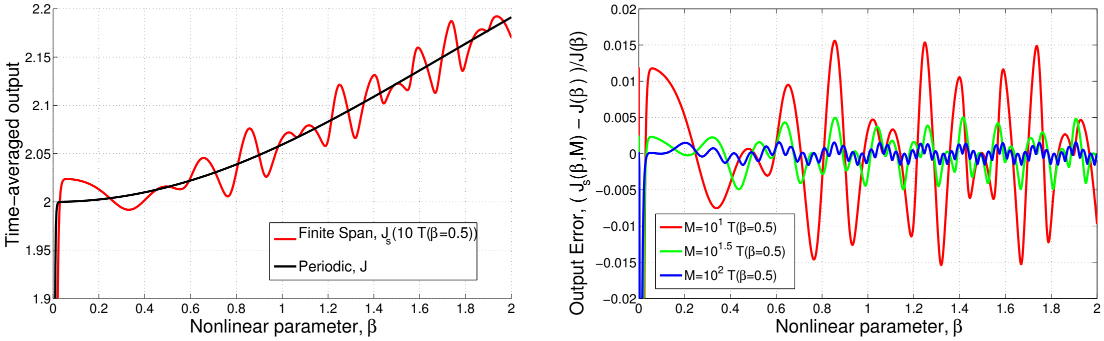
(a) Output error vs. window size (square window)

(b) Output error vs. window size

Figure 4-10: Output behavior vs. $M$ for $\beta = 0.5$

the output error is convergent with increasing window span for all windows. The square window converges at $\mathcal{O}(M^{-1})$, and the Hann and Hann-squared at $\mathcal{O}(M^{-3})$ and $\mathcal{O}(M^{-5})$, respectively. Finally, the bump appears to exhibit the expected exponential convergence.

To gain insight into the behavior of the output sensitivity, we first look at the variation of $\mathcal{J}$ and $\mathcal{J}_s$ for $0 < \beta \leq 2$ in Figure (4-11a). The time-average smoothly increases, but the square window with $M = 10T_{\beta=0.5}$ oscillates with approximately 1% error. Plotting the output error for three different window spans in Figure (4-11b) shows not only a decrease in the error magnitude, but a coincident increase in the frequency of the output error oscillation. This is caused by the sensitivity of the period to changes in $\beta$. A change in $\beta$ then causes the phase of the oscillation to shift at the window span boundaries. As the window span increases, the span contains a larger number of periods of output oscillation and changes in $\beta$ cause larger shifts in phase at the window endpoints. As previously demonstrated, increasing $M$ causes the square windowed output to converge at $\mathcal{O}(M^{-1})$, but the phase shift with changing $\beta$ simultaneously increases at $\mathcal{O}(M)$. The sensitivity of the square windowed output combines these behaviors, resulting in a nondiminishing output sensitivity error. If the square windowed output and its sensitivity to $\beta$ were used for optimization with respect to $\beta$, the oscillation of $\mathcal{J}_s$ observed in Eq. (4-11) could stall the optimization process in a local extremum.



(a) Output vs. $\beta$

(b) Finite span output error vs. $\beta$

Figure 4-11: Output behavior vs. nonlinear parameter $\beta$

The output sensitivity with respect to $\beta$ was investigated in Figure (4-12) with the same long-time windows as for the output. For shorter span windows, the convergence of the square and Hann windows match the convergence rate of the output, i.e., the error in the sensitivity is dominated by the $\mathcal{A}$ term. For larger span (smaller error) square and Hann windows, the convergence rate decreases by one order as predicted by Theorem (4). The Hann-squared window matches the output convergence rate, but levels off due to numerical precision with the expectation that it would have otherwise had similar behavior. The convergence of the bump window for the sensitivity integration retains the same exponential shape as for the output before leveling off due to precision.

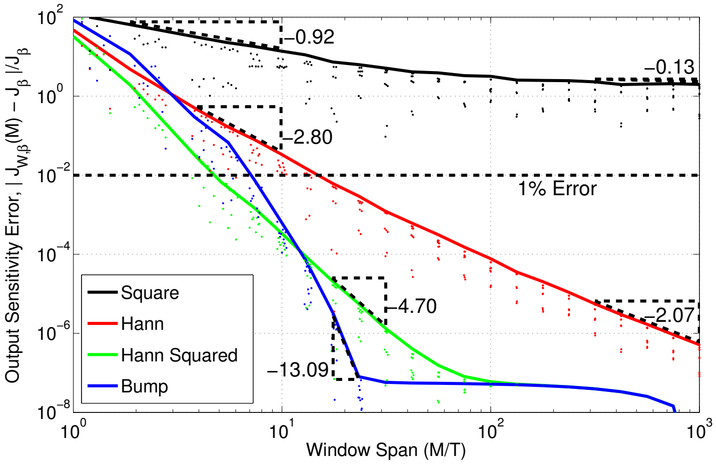
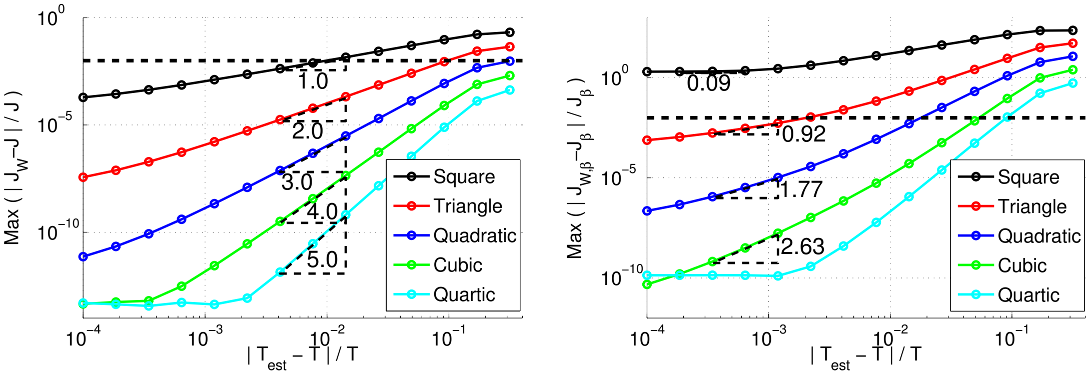


Figure 4-12: Output sensitivity error vs. window size

Short-time windows were investigated with the square (1 period), triangle, piecewise quadratic, cubic, and quartic windows for both positive and negative period error. Figure (4-13a) confirms that for an approximate period $T_{est}$ with period error $\delta T = T_{est} - T$, the error in the output converges at the expected rate. Figure (4-13b) shows that the output sensitivity follows the same convergence rate as the output for $\delta T/T > 2e - 3$, indicating that in this range the error is bounded by the error associated with $\mathcal{A}$. For smaller errors, the error is dominated by the $\mathcal{B}$ term and converges at the expected slower rate.



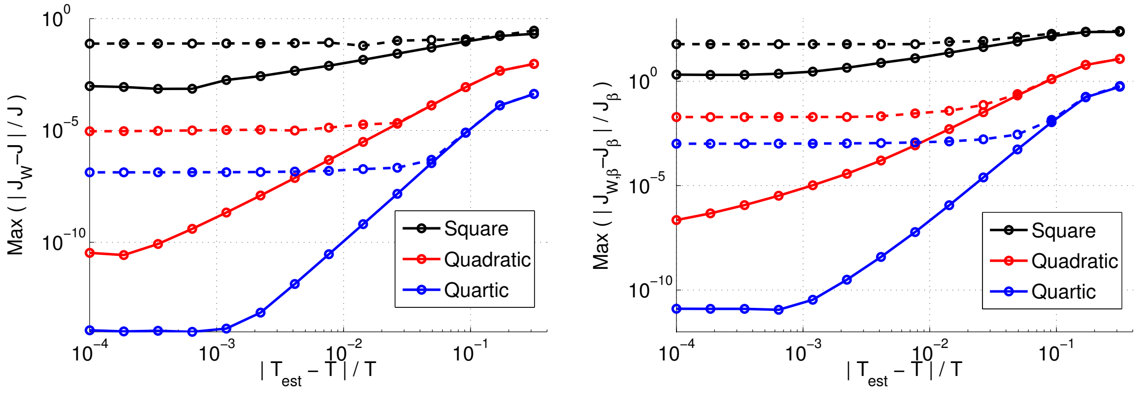
(a) Maximum output error

(b) Maximum sensitivity error

Figure 4-13: Short-time window time-average error vs. $|\delta T|/T$ for $\beta = 0.5$ (dash line at 1% error)

These convergence rates are highly dependent on the overlap of the timesteps with the window. In the case where the timestep does not line up with the window discontinuities, an error (dependent on the smoothness of the window) can be introduced which decreases the convergence rate. This is demonstrated in Figure (4-14) for timesteps corresponding to 10 and 1000 steps per period and the error defined relative to the piecewise quartic window at the best calculated period for that timestep. Whereas the convergence rates with respect to window span for long time windows are relatively insensitive to the timestep, short time windows have proven to be extremely sensitive.

For many engineering applications, 1% error is an acceptable error threshold. To achieve this error level in the output requires ≈ 10 periods with square window and 2 or fewer for any of the smoother windows. For the sensitivity the square window levels off before achieving this level accuracy. The Hann window requires almost 15 periods; the Hann-squared, 5 periods; and the bump, 7. It is only for lower error that the bump window requires a shorter window span than the Hann-squared window. Any of the non-square short-time windows,



(a) Maximum output error

(b) Maximum sensitivity error

Figure 4-14: Timestep study of short-time window time-average error vs. $|\delta T|/T$ for $\beta = 0.5$ [$log_{10}(T/\Delta t)$: solid = 3; dash = 1]

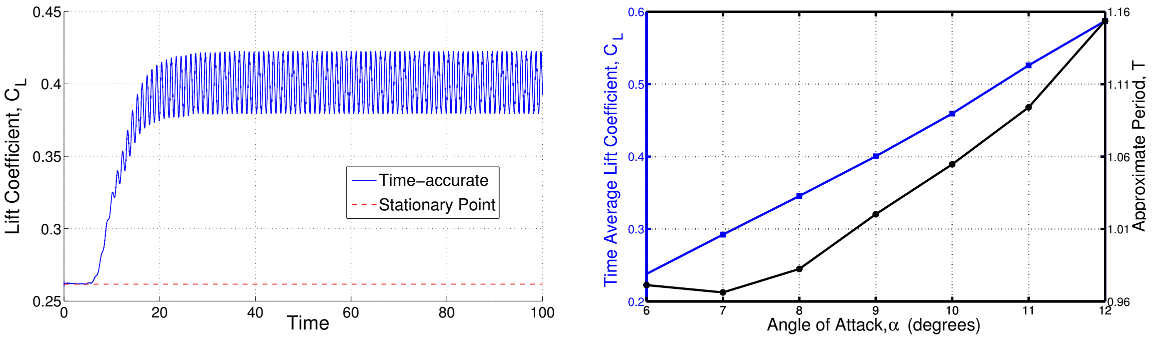
achieve 1% error in the output with error in the period as high as 10%, and the quartic window achieves 1% error in the sensitivity with period approximation 9%. Compared to long-time windows, short-time windows offer a modest decrease in the number of periods that must be simulated to compute the output sensitivity, even using relatively inaccurate period approximations. Relative to the size of the lead-in, this savings may not justify the need to approximate the period and modify the output window accordingly.

## 4.4 Airfoil at High Angle of Attack

### 4.4.1 Results

The subsonic, laminar flow NACA0012 case described in Section 3.2 at $\alpha = 9°$ is used to demonstrate the impact of output windowing on sensitivity analysis of an aerodynamics application. The NACA 0012 problem at $\alpha = 9°$ was examined using several windows, with the goal of recreating the results of the van der Pol oscillator for the more complex system of equations, in particular Figure (4-12) and Figure (4-13). The output of interest is the time-average lift coefficient with the angle of attack as the system parameter to which we wish to compute sensitivity. Shown in Figure (4-15a), the problem is initialized to a small perturbation of the stationary point solution and allowed to run to time $t = 200$, such that the initial transient has for the most part died out leaving an equilibrium oscillation. The windowed time-average lift, $\mathcal{J}_w$, was computed for a range of window spans from one period up to 1000 periods using a lead-in to equilibrate the tangent state of 175 periods.

The period is approximated by interrogating the trough-to-trough time over approximately 90 periods. At $\alpha = 9°$, the period is approximately $T = 1.016123$. The NACA 0012 at these conditions has a non-negligible dependence of the oscillation period and the time-average lift coefficient on the angle of attack, shown in Figure (4-15b) using a bump window with $M = 1000T$.



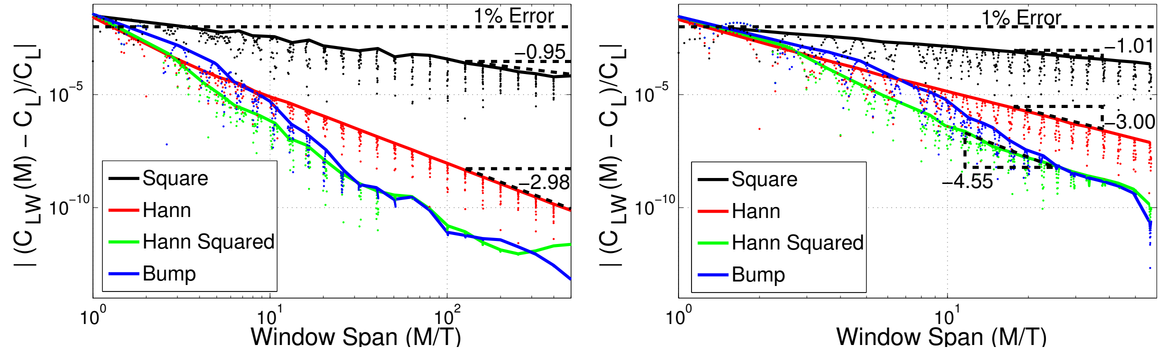
(a) Problem initialization: $C_L$ vs. time

(b) $\overline{C_L}$ & $T$ vs. angle of attack

Figure 4-15: NACA 0012 - unsteady solution

Figure (4-16a) show the output error for several windows computed relative to the output with the bump window and $M/T = 1000$ giving $C_L = 0.4004230632468$. The output error of the square window is bounded by $\mathcal{O}(M^{-1})$. As with the van der Pol oscillator, the output weighted by the Hann window converges at $\mathcal{O}(M^{-3})$. The Hann-squared window converges at approximately $\mathcal{O}(M^{-5})$, but the convergence rate is not as uniform as the Hann window due to numerical precision. The bump window error appears to follow an exponential trajectory through 30 periods, but levels off to converge at approximately the same rate as the Hann-squared windowed output. When the timestep is decreased to $\Delta t = 0.002$, the error for the Hann-squared and bump windows exhibit less noise and adhere more closely to the expected rates.

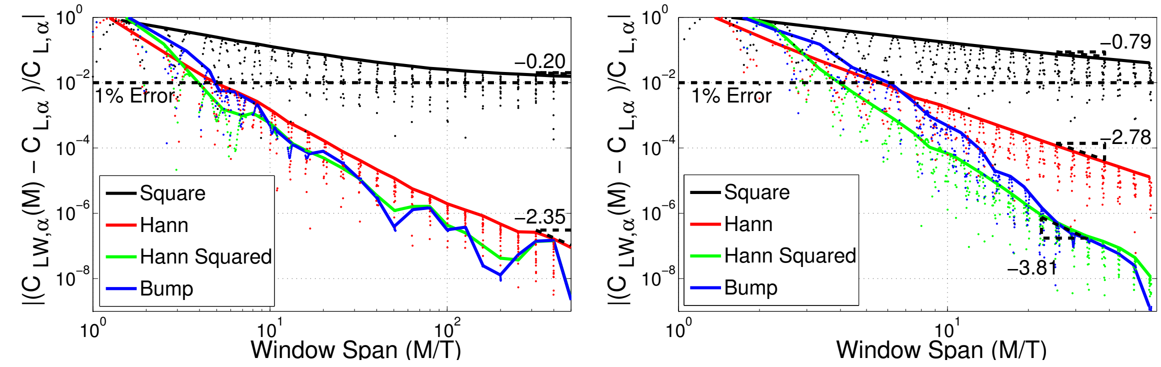
The sensitivity of the time-average $C_L$ to angle of attack was computed using the tangent equations, and is investigated in Figure (4-17a). The output sensitivity error is computed relative to the sensitivity using the bump window and $M/T = 10^{2.5}$ giving $C_{L,\alpha} = 3.2801494$. For smaller window spans (from 1 to 10 periods), the square and Hann windows converge the output sensitivity at the same rate as for the output, but as the window spans continue to increase through 100, they both begin level off. This leveling off results in the nonconvergence of the square window and the error of the Hann window



(a) $\Delta t = 0.1$ (b) $\Delta t = 0.002$

Figure 4-16: NACA 0012 - Output error vs. $M/T$ (smaller max $M$ for $\Delta t = 0.002$ due to cost)

converges at $\mathcal{O}(M^{-2})$. The bump and Hann-squared windows exhibit lower error than the Hann window, but at approximately the same rate for this case. Again, when the timestep is decreased to $\Delta t = 0.002$, the error curves have smoother convergence rates with decreased noise and exhibit improved convergence rates for increased span windows compared to the larger timestep. For this problem, for both the output and the sensitivity to achieve a 1% relative error, any of the suggested non-square windows will suffice with a small number of periods ($M < 10T$). The smoother windows are only necessary for tighter accuracy requirements, as the effect of increased convergence is not found until larger $M$.



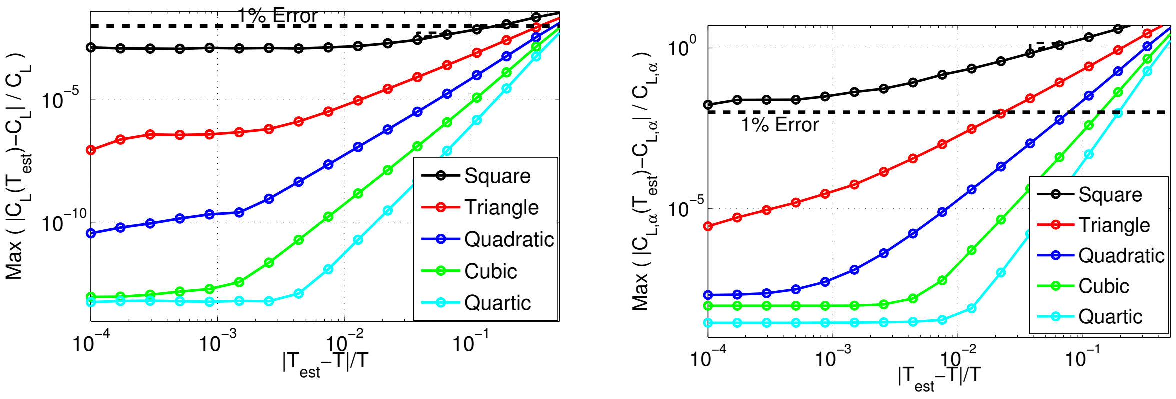
(a) $\Delta t = 0.1$ (b) $\Delta t = 0.002$

Figure 4-17: NACA 0012 - Output sensitivity error vs. $M/T$ (smaller max $M$ for $\Delta t = 0.002$ due to cost)

For short-time windows, the output and output sensitivity of the NACA 0012 were calculated using the square, triangle, and piecewise quadratic, piecewise cubic, and piecewise quartic windows. For timesteps comparable to that used to compute the long-time windows

($\Delta t = 0.1$), error due to the misalignment of the timestep with the window endpoints is dominant and does not diminish with an improved period estimate. A smaller timestep is required to decrease the influence of this misalignment, and so $\Delta t = 0.002$ is used. The smaller timestep gives a more accurate solution than the large timestep used for the long-time windows, and so the $\Delta t = 0.1$ results are incompatible for determining convergence with regards to the period approximation. To correct this, the error is calculated relative to the bump window using $\Delta t = 0.002$ at $\sim 60$ periods, and the results are shown in Figure (4-18a) and Figure (4-18b). In the output, all five windows converge with $\mathcal{O}(\delta T^n)$ before leveling off, verifying the improved convergence rate of smoother windows. Testing with various timesteps suggests that for the square, triangle, and quadratic windows the leveling off is caused by the timestep misalignment. As with the outputs, the windowed sensitivities converge at $\mathcal{O}(\delta T^n)$ or better before leveling off, indicating that for larger period error and $\Delta t = 0.002$, the error is dominated by the $\mathcal{A}$ term. For smaller error, only the square and triangle window show the decreased convergence rate related to the $\mathcal{B}$ term. For both the output and sensitivity, the smoother windows level off at approximately the same error, but with a distinct decrease in error with increased window smoothness. Examination of the root mean square with the phase of the windows of both the output and output sensitivity show that the sensitivity to the phase levels off as well, revealing the limit of the accuracy of the simulation on the limit cycle: for the output, the flow solution; and for the sensitivity, the tangent state solution. Regardless of this limitation, all $n > 1$ short-time windows outperform the square window.

At 1% error in the period approximation, the triangle is 2 orders of magnitude more accurate than the square window, the quadratic another 2, and the cubic and quartic windows more than 6 orders of magnitude more accurate. To achieve 1% error relative error in the output, any of the $n > 1$ short-time windows will suffice with even a 10% error in period approximation. To achieve the same relative error in the output sensitivity, any of the $n > 1$ windows will suffice with 1% error in the period approximation. The quartic window gives less than 1% sensitivity relative error for 10% error in the period approximation, demonstrating the insensitivity to inaccuracy in the period approximation for the smoother, larger $n$ windows. For this calculation, the lead-in to equilibrate the derivative variable (tangent or adjoint) is much larger than the window span ($t_r = 175 \gg 5$), thus the marginal increase in total calculation is small between a triangle or quadratic window and the longer cubic or quartic windows.



(a) Output error

(b) Sensitivity error

Figure 4-18: Effect of period estimate error on periodic time-average error

To achieve a 1% error in the time-average lift, the choice of long-time windows make little difference: all, including the square window, require less than 2 periods. For $dC_L/d\alpha$, however, the square window asymptotes to error greater than 1%. The non-square windows require between 4 and 6 periods, with the Hann-squared requiring the shortest window. Short-time windows, as with the van der Pol oscillator, require only a crude approximation of the period to reach the 1% error tolerance. Again, as with the van der Pol oscillator, the lead-in time combined with the need to compute the period of the oscillation, however approximate, may make long-time windows the desirable option.

# Chapter 5

# Spatial Adaptation for Unsteady Wake Problems

In this chapter, a spatial mesh adaptation method appropriate for unsteady wake problems is proposed. A review of adjoint-based output error estimation is carried out in Section 5.1, including steady error estimation and the unsteady extension used in this work. Section 5.2 introduces the Mesh Optimization via Error Sampling and Synthesis (MOESS) algorithm of Yano & Darmofal[126, 127], and Section 5.3 develops the proposed modification to the algorithm for unsteady problems with an all-time spatial mesh. Section 5.4 demonstrates the proposed method for several inviscid and laminar Navier-Stokes problems, respectively.

## 5.1 Adjoint-Based Error Estimation

The spatial error estimation strategy used as a basis for the unsteady spatial error estimate is based on the method of dual-weighted residuals (DWR) of Becker and Rannacher[13]. Output-based error estimation and adaption have been studied extensively in the literature[8, 27, 37, 42, 43, 59, 60, 88, 116]. Recent reviews of the DWR method in the context of aerospace CFD applications are provided by Hartmann and Houston[44] and Fidkowski and Darmofal[26]. A posteriori output error estimates enable automated mesh adaptation by localizing error to specific elements, so that the spatial refinement can specifically target larger-error elements. The error estimate summed over all elements gives some measure of global output error to evaluate the quality of the solution and to determine when sufficient accuracy is achieved.

### 5.1.1 Steady Error Estimate

The Lagrangian approach, as was used to derive the adjoint and parameter sensitivities, is commonly used to derive an output error estimate within an optimal control framework. The error estimation analysis presented here closely follows the presentation of Modisette[69]. First, assume consistency in the finite-dimensional residual statement of Eq. (2.18), i.e., the exact solution, $\boldsymbol{u} \in \mathcal{V}$, satisfies

$$\mathcal{R}_{hp}(\boldsymbol{u}, \boldsymbol{v}_{hp}) = 0, \qquad \forall \boldsymbol{v}_{hp} \in \mathcal{V}_h^p$$

Note that the explicit $\beta$ parameter dependency is dropped from the notation in this chapter. Define the mean value linearizations (MVL) of the finite-dimensional spatial residual and output as

$$\begin{aligned} \bar{\mathcal{R}}_{hp}(\boldsymbol{u}, \boldsymbol{u}_{hp}; \boldsymbol{v}, \boldsymbol{w}) &\equiv \int_0^1 \mathcal{R}'_{hp}\left[\theta \boldsymbol{u} + (1-\theta)\boldsymbol{u}_{hp}\right](\boldsymbol{v}, \boldsymbol{w})\, d\theta \\ \bar{\mathcal{J}}_{hp}(\boldsymbol{u}, \boldsymbol{u}_{hp}; \boldsymbol{v}) &\equiv \int_0^1 \mathcal{J}'_{hp}\left[\theta \boldsymbol{u} + (1-\theta)\boldsymbol{u}_{hp}\right](\boldsymbol{v})\, d\theta \end{aligned}$$

with $\boldsymbol{v}, \boldsymbol{w} \in \mathcal{W}_h^p = \mathcal{V} + \mathcal{V}_h^p$. The primes denote Frećhet derivatives taken about the bracketed terms. The residual and output MVL are then combined to define the Lagrangian MVL,

$$\bar{\mathcal{L}}_{hp}(\boldsymbol{u}, \boldsymbol{u}_{hp}; \boldsymbol{v}, \boldsymbol{w}) \equiv \bar{\mathcal{J}}_{hp}(\boldsymbol{u}, \boldsymbol{u}_{hp}; \boldsymbol{v}) - \bar{\mathcal{R}}_{hp}(\boldsymbol{u}, \boldsymbol{u}_{hp}; \boldsymbol{v}, \boldsymbol{w}).$$

As in Section 2.2.1 enforcing stationarity of the MVL Lagrangian with respect to variation in $\boldsymbol{v}$ defines the dual, $\boldsymbol{\psi} \in \mathcal{W}_h^p$, such that

$$\bar{\mathcal{R}}_{hp}(\boldsymbol{u}, \boldsymbol{u}_{hp}; \delta\boldsymbol{v}, \boldsymbol{\psi}) - \bar{\mathcal{J}}_{hp}(\boldsymbol{u}, \boldsymbol{u}_{hp}; \delta\boldsymbol{v}) = 0, \quad \forall \delta\boldsymbol{v} \in \mathcal{W}_h^p. \tag{5.1}$$

For the case where $\delta\boldsymbol{v} = \boldsymbol{u}_{hp} - \boldsymbol{u}$, Eq. (5.1) can be rearranged to express the output error:

$$\begin{aligned} \mathcal{E} = \mathcal{J}_{hp}(\boldsymbol{u}_{hp}) - \mathcal{J}(\boldsymbol{u}) &= \bar{\mathcal{J}}_{hp}(\boldsymbol{u}, \boldsymbol{u}_{hp}; \boldsymbol{u}_{hp} - \boldsymbol{u}) \\ &= \bar{\mathcal{R}}_{hp}(\boldsymbol{u}, \boldsymbol{u}_{hp}; \boldsymbol{u}_{hp} - \boldsymbol{u}, \boldsymbol{\psi}) \\ &= \mathcal{R}_{hp}(\boldsymbol{u}_{hp}, \boldsymbol{\psi}) - \cancelto{0}{\mathcal{R}_{hp}(\boldsymbol{u}, \boldsymbol{\psi})} \\ &= \mathcal{R}_{hp}(\boldsymbol{u}_{hp}, \boldsymbol{\psi}) \end{aligned} \tag{5.2}$$

Therefore, if the dual, $\boldsymbol{\psi}$, is known, the output error can be computed exactly by using it to weight the finite dimensional primal residual.

In general $\boldsymbol{u}$ and $\boldsymbol{\psi}$ are not known and approximations must be used to enable the use of the output error estimate. The first approximation replaces the mean value linearizations with linearizations about the finite-dimensional primal state, $\boldsymbol{u}_{hp}$. The second replaces the exact adjoint in the residual weighting with a *truth surrogate* state. Rannacher[89], Barth and Larson[8], Lu[59], and Fidkowski and Darmofal[28] employ $H^1$ patch reconstruction of $\boldsymbol{\psi}_{hp} \in \mathcal{V}_h^p$ into $\mathcal{V}_h^{p+1}$. Barter[5] and Oliver[82], noting smearing introduced by patch reconstruction in problems with discontinuities and near curved boundaries, instead project $\boldsymbol{\psi}_{hp}$ into a finer space, $\mathcal{V}_h^{p'}$ with $p' > p$, and carry out several element-Jacobi smoothing iterations. Other authors have striven for more accurate surrogates, including solving the adjoint completely at higher polynomial order (Hartmann[43]) and a combination of increased polynomial order and refined triangulation (Solin & Demkowicz[106]). The adjoint truth surrogate used herein follows Hartmann[43] and seeks the adjoint solution in the enriched space $\mathcal{V}_h^{p'}$ for $p' = p + 1$. Using these approximations, the dual truth surrogate is computed by

$$\mathcal{R}'_{hp}[\boldsymbol{u}_{hp'}](\boldsymbol{v}_{hp'}, \boldsymbol{\psi}_{hp'}) - \mathcal{J}'_{hp'}[\boldsymbol{u}_{hp}](\boldsymbol{v}_{hp'}) = 0, \quad \forall \boldsymbol{v}_{hp'} \in \mathcal{V}_h^{p'},$$

in the enriched polynomial space exactly.

By restricting the adjoint state to an element $\kappa$, the error estimate may be localized to an elemental error estimate:

$$\eta_\kappa = \mathcal{R}_{hp}(\boldsymbol{u}_{hp}, \boldsymbol{\psi}_{hp'}|_\kappa). \tag{5.3}$$

This localized error estimate is the key to the adaptive process, and enables the coarsening and refinement for each element appropriate to its current error contribution.

### 5.1.2 Unsteady Spatial Error Estimate

Building on the steady adjoint error estimation, an estimate of the spatial error for an unsteady problem is derived. The error estimate is modified by including the unsteady term and the initial condition of the primal problem such that the unsteady MVL Lagrangian is

$$\bar{\mathcal{L}}^E_{hp}(\boldsymbol{u}, \boldsymbol{u}_{hp}; \boldsymbol{v}, \boldsymbol{w}, \tilde{\boldsymbol{w}}) \equiv \bar{\mathcal{J}}^E_{hp}(\boldsymbol{u}, \boldsymbol{u}_{hp}; \boldsymbol{v}) - \bar{\mathcal{R}}^E_{hp}(\boldsymbol{u}, \boldsymbol{u}_{hp}; \boldsymbol{v}, \boldsymbol{w}) - \bar{\mathcal{A}}_{hp}(\boldsymbol{u}(t_0), \boldsymbol{u}_{hp}(t_0); \boldsymbol{v}(t_0), \tilde{\boldsymbol{w}})$$

where $\bar{\mathcal{R}}^E_{hp}$ and $\bar{\mathcal{A}}_{hp}$ are the MVL of the integrated residual and initial condition, respectively:

$$\begin{aligned} \bar{\mathcal{R}}^E_{hp}(\boldsymbol{u}, \boldsymbol{u}_{hp}; \boldsymbol{v}, \boldsymbol{w}) &\equiv \int_I \left\{ \left( \frac{d\boldsymbol{v}}{dt}, \boldsymbol{w} \right) + \bar{\mathcal{R}}_{hp}\left(\boldsymbol{u}, \boldsymbol{u}_{hp}; \boldsymbol{v}, \boldsymbol{w}\right) \right\} dt \\ \bar{\mathcal{A}}_{hp}(\boldsymbol{u}(t_0), \boldsymbol{u}_{hp}(t_0); \boldsymbol{v}(t_0), \tilde{\boldsymbol{w}}) &\equiv \int_0^1 \mathcal{A}'_{hp}\left[\theta \boldsymbol{u}(t_0) + (1-\theta)\boldsymbol{u}_{hp}(t_0)\right] \left(\boldsymbol{v}(t_0), \tilde{\boldsymbol{w}}\right) d\theta. \end{aligned}$$

$\bar{\mathcal{J}}^E_{hp}$ is the MVL of Eq. (2.23), the integrated unsteady output,

$$\bar{\mathcal{J}}^E_{hp}(\boldsymbol{u}, \boldsymbol{u}_{hp}; \boldsymbol{v}) \equiv \bar{\mathcal{H}}_{hp}(\boldsymbol{u}(t_f), \boldsymbol{u}_{hp}(t_f); \boldsymbol{v}(t_f)) + \int_I \bar{\mathcal{G}}_{hp}(\boldsymbol{u}, \boldsymbol{u}_{hp}; \boldsymbol{v})\, dt,$$

where $\bar{\mathcal{H}}_{hp}$ and $\bar{\mathcal{G}}_{hp}$ are defined by,

$$\begin{aligned} \bar{\mathcal{H}}_{hp}(\boldsymbol{u}(t_f), \boldsymbol{u}_{hp}(t_f); \boldsymbol{v}(t_f)) &\equiv \int_0^1 \mathcal{H}'_{hp}\left[\theta \boldsymbol{u}(t_f) + (1-\theta)\boldsymbol{u}_{hp}(t_f)\right] \left(\boldsymbol{v}(t_f)\right) d\theta \\ \bar{\mathcal{G}}_{hp}(\boldsymbol{u}(t), \boldsymbol{u}_{hp}(t); \boldsymbol{v}(t)) &\equiv \int_0^1 \mathcal{G}'_{hp}\left[\theta \boldsymbol{u}(t) + (1-\theta)\boldsymbol{u}_{hp}(t)\right] \left(\boldsymbol{v}(t)\right) d\theta. \end{aligned}$$

The first variation of the MVL Lagrangian with variations in $\boldsymbol{w}$, $\tilde{\boldsymbol{w}}$, and $\boldsymbol{v}$ can be expressed as

$$\begin{aligned} \delta \bar{\mathcal{L}}^E_{hp} = &- \bar{\mathcal{R}}^E_{hp}(\boldsymbol{u}, \boldsymbol{u}_{hp}; \boldsymbol{v}, \delta\boldsymbol{w}) - \bar{\mathcal{A}}(\boldsymbol{u}, \boldsymbol{u}_{hp}; \boldsymbol{v}(t_0), \delta\tilde{\boldsymbol{w}}) \\ &+ \bar{\mathcal{J}}^E_{hp}(\boldsymbol{u}, \boldsymbol{u}_{hp}; \delta\boldsymbol{v}) - \bar{\mathcal{R}}^E_{hp}(\boldsymbol{u}, \boldsymbol{u}_{hp}; \delta\boldsymbol{v}, \boldsymbol{w}) - \bar{\mathcal{A}}_{hp}(\boldsymbol{u}(t_0), \boldsymbol{u}_{hp}(t_0); \delta\boldsymbol{v}(t_0), \tilde{\boldsymbol{w}}). \end{aligned} \tag{5.4}$$

By maintaining stationarity, the first line recovers the MVL primal residual (stationarity wrt $\delta\boldsymbol{w}$) and initial condition (stationarity wrt $\delta\tilde{\boldsymbol{w}}$). The remainder constrains the MVL adjoint by maintaining stationarity with respect to $\delta\boldsymbol{v}$. To develop the error estimate, seek adjoints, $\boldsymbol{w} = \boldsymbol{\psi} \in \mathcal{W}^p_h \times I$ and $\tilde{\boldsymbol{w}} = \tilde{\boldsymbol{\psi}} \in \mathcal{W}^p_h$, that satisfy:

$$-\bar{\mathcal{J}}^E_{hp}(\boldsymbol{u}, \boldsymbol{u}_{hp}; \boldsymbol{v}) + \bar{\mathcal{R}}^E_{hp}(\boldsymbol{u}, \boldsymbol{u}_{hp}; \boldsymbol{v}, \boldsymbol{\psi}) + \bar{\mathcal{A}}_{hp}(\boldsymbol{u}(t_0), \boldsymbol{u}_{hp}(t_0); \boldsymbol{v}(t_0), \tilde{\boldsymbol{\psi}}) = 0, \;\; \forall \boldsymbol{v} \in \mathcal{W}^p_h \times I. \tag{5.5}$$

Employing integration by parts and expressing $\bar{\mathcal{J}}^E$ using $\bar{\mathcal{G}}$ and $\bar{\mathcal{H}}$, this can be separated into the familiar adjoint equation:

$$0 = -\left(\boldsymbol{v}, \frac{d\boldsymbol{\psi}(t)}{dt}\right) + \bar{\mathcal{R}}_{hp}(\boldsymbol{u}(t), \boldsymbol{u}_{hp}(t); \boldsymbol{v}, \boldsymbol{\psi}(t)) - \bar{\mathcal{G}}_{hp}(\boldsymbol{u}(t), \boldsymbol{u}_{hp}(t); \boldsymbol{v}), \;\; \forall \boldsymbol{v} \in \mathcal{W}_h^p, \; \forall t \in I, \tag{5.6}$$

adjoint terminal condition:

$$0 = \bar{\mathcal{H}}_{hp}(\boldsymbol{u}(t_f), \boldsymbol{u}_{hp}(t_f); \boldsymbol{v}) - \left(\boldsymbol{\psi}(t_f), \boldsymbol{v}\right), \quad \forall \boldsymbol{v} \in \mathcal{W}_h^p, \tag{5.7}$$

and initial condition adjoint equation:

$$0 = -\left(\boldsymbol{v}, \boldsymbol{\psi}(t_0)\right) + \bar{\mathcal{A}}_{hp}(\boldsymbol{u}(t_0), \boldsymbol{u}_{hp}(t_0); \boldsymbol{v}, \tilde{\boldsymbol{\psi}}), \quad \forall \boldsymbol{v} \in \mathcal{W}_h^p. \tag{5.8}$$

For the case with $\boldsymbol{v} = \boldsymbol{u}_{hp} - \boldsymbol{u}$, Eq. (5.5) can be manipulated to give:

$$\begin{aligned}
\mathcal{E}^E &= \mathcal{J}^E_{hp}(\boldsymbol{u}_{hp}) - \mathcal{J}^E_{hp}(\boldsymbol{u}) \\
&= \bar{\mathcal{J}}^E_{hp}(\boldsymbol{u}, \boldsymbol{u}_{hp}; \boldsymbol{u}_{hp} - \boldsymbol{u}) \\
&= \bar{\mathcal{R}}^E_{hp}(\boldsymbol{u}, \boldsymbol{u}_{hp}; \boldsymbol{u}_{hp} - \boldsymbol{u}, \boldsymbol{\psi}) + \bar{\mathcal{A}}_{hp}(\boldsymbol{u}(t_0), \boldsymbol{u}_{hp}(t_0); \boldsymbol{u}_{hp}(t_0) - \boldsymbol{u}(t_0), \tilde{\boldsymbol{\psi}}) \\
&= \mathcal{R}^E_{hp}(\boldsymbol{u}_{hp}, \boldsymbol{\psi}) - \cancelto{0}{\mathcal{R}^E_{hp}(\boldsymbol{u}, \boldsymbol{\psi})} + \mathcal{A}_{hp}(\boldsymbol{u}_{hp}(t_0), \boldsymbol{\psi}(t_0), \tilde{\boldsymbol{\psi}}) - \cancelto{0}{\mathcal{A}_{hp}(\boldsymbol{u}(t_0), \boldsymbol{\psi}(t_0), \tilde{\boldsymbol{\psi}})} \\
&= \mathcal{R}^E_{hp}(\boldsymbol{u}_{hp}, \boldsymbol{\psi}) + \mathcal{A}_{hp}(\boldsymbol{u}_{hp}(t_0), \boldsymbol{\psi}(t_0), \tilde{\boldsymbol{\psi}})
\end{aligned}$$

With $\boldsymbol{u} \in \mathcal{V} \times I$ and $\boldsymbol{u}_{hp} \in \mathcal{V}_h^p \times I$, $\mathcal{E}^E$ is the error induced by the spatial discretization in $\mathcal{V}_h^p$ for a temporally continuous problem. In general, $\boldsymbol{u}$, $\boldsymbol{\psi}$, and $\tilde{\boldsymbol{\psi}}$ are unavailable and must be approximated. As in the steady case, the mean value linearizations in Eq. (5.6), (5.7), and (5.8) are replaced by linearizations about $\boldsymbol{u}_{hp}$. The adjoint states, $\boldsymbol{\psi}$ and $\tilde{\boldsymbol{\psi}}$, are then computed in the enriched polynomial space, $\mathcal{V}_h^{p'}$, using the approximate linearizations. By defining the instantaneous unsteady primal residual, $\mathcal{R}^e_{hp}(\boldsymbol{u}_{hp}, \boldsymbol{v}_{hp}; t)$, as the integrand of $\mathcal{R}^E_{hp}$ at time $t$ such that

$$\mathcal{R}^E_{hp} = \int_I \mathcal{R}^e_{hp} \, dt,$$

the error at a given time can be approximated by

$$\mathcal{E}^e(t) \approx \mathcal{R}^e_{hp}(\boldsymbol{u}_{hp}, \boldsymbol{\psi}_{hp'}; t), \qquad \forall t \in I \tag{5.9}$$

and for the entire time domain as

$$\begin{aligned}\mathcal{E}^E &= \mathcal{J}^E_{hp}(\boldsymbol{u}_{hp}) - \mathcal{J}^E(\boldsymbol{u}) \\ &\approx \mathcal{R}^E_{hp}(\boldsymbol{u}_{hp}, \boldsymbol{\psi}_{hp'}) + \mathcal{A}_{hp}(\boldsymbol{u}_{hp}(t_0), \boldsymbol{\psi}_{hp'}(t_0), \tilde{\boldsymbol{\psi}}_{hp'})\end{aligned} \tag{5.10}$$

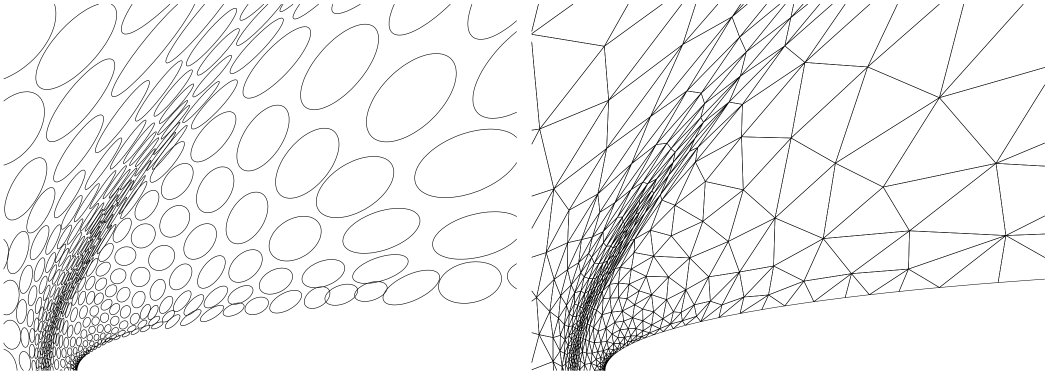
The term including the initial condition allows for spatial error due to an initial condition specified in a different space than $\mathcal{V}^p_h$, such as an analytic function or an 'exact' solution in $\mathcal{V}$. The resulting error estimate is a measure of the spatial error only, and will be unable to measure or localize temporal discretization error. Though it will not be utilized as such here, Eq. (5.9) does localize spatial error in time, and so could be used toward developing temporally evolving spatial meshes. As with the steady case, the error estimate can be localized to an elemental error estimate by restricting the adjoint to a single element:

$$\eta^E_\kappa = \int_I \eta^e_\kappa(t)\, dt \approx \int_I \mathcal{R}^e_{hp}(\boldsymbol{u}_{hp}, \boldsymbol{\psi}_{hp'}\big|_\kappa, t)\, dt + \mathcal{A}_{hp}(\boldsymbol{u}_{hp}(t_0), \boldsymbol{\psi}_{hp'}(t_0), \tilde{\boldsymbol{\psi}}_{hp'}\big|_\kappa). \tag{5.11}$$

The above terms are not specific to a particular temporal discretization. As the goal is only to estimate spatial error, the adjoint truth surrogate, $\boldsymbol{\psi}_{hp'}$, is discretized temporally using the dual of the temporally discrete primal, as described in Appendix (A). It is not necessary, as it is for temporal error estimation, to compute the adjoint truth surrogate using a more accurate temporal discretization.

## 5.2 Steady Adaptation Algorithm

The spatial adaptation method for unsteady flows developed here utilizes the Mesh Optimization via Error Sampling and Synthesis (MOESS) algorithm developed by Yano & Darmofal[126, 127]. In this section, MOESS is reviewed in preparation for its extension to an unsteady problem.



(a) Metric field, $\mathcal{M}$

(b) Simplex mesh, $\mathcal{T}_h$

Figure 5-1: Metric-mesh pair example (Modisette[69])

### 5.2.1 Metric-Based Anisotropic Mesh Generation

This section reviews the concept of a metric tensor and the metric manipulation framework employed by Yano & Darmofal[127], drawing on the work of Loseille and Alauzet[55].

**Definition 1.** *A Riemannian metric field $\{\mathcal{M}(x)\}_{x\in\Omega}$ is a smoothly varying field of symmetric positive definite (SPD) matrices on $\Omega \subset \mathbb{R}^d$. The length of a segment $\overrightarrow{ab}$ from $a \in \Omega$ to $b \in \Omega$ in the metric space is given by*

$$\ell_{\mathcal{M}}(\overrightarrow{ab}) = \int_0^1 \sqrt{\overrightarrow{ab}^T \mathcal{M}(a + \overrightarrow{ab}s)\overrightarrow{ab}}\; ds.$$

**Definition 2.** *A metric-conforming tessellation is one such that all edges in the tessellation are approximately unit length in the metric space as computed consistent with Definition 1. That is,*

$$\ell_{\mathcal{M}}(e) \approx 1 \quad \forall e \in \mathrm{Edges}(\mathcal{T}_h).$$

A metric tensor, $\mathcal{M}_\kappa \in Sym_d^+$, is a unique symmetric positive definite (SPD) matrix which can encodes an element's anisotropic element size and shape information, thereby also encoding the ability of the element $\kappa \in \mathcal{T}_h$ to approximate a function. While the metric for each element computed from a given tessellation is unique, a nonunique family of meshes can be constructed to conform to a given metric field. An example of a complementary metric-mesh pair is shown in Figure (5-1). A requested change in the size or shape of a tessellation, as part of a mesh adaptation algorithm, can be expressed as a desired change in the mesh metric field.

It is found that the manipulation of the components of the metric matrices in the Euclidean sense, i.e., $\mathcal{M} = \mathcal{M}_0 + \delta\mathcal{M}$, presents some difficulties, chiefly that the $\delta\mathcal{M}$ must be carefully selected to retain positive definiteness of $\mathcal{M}$, and the means to realize specific changes in size/shape of $\mathcal{M}$ via $\delta\mathcal{M}$ is not readily apparent. A solution is found with the affine-invariant Riemannian framework introduced by Pennec et al.[85]. In their framework, the change to a metric $\mathcal{M}_0$ is induced by the exponential map of a tangent metric, $S \in Sym_d$:

$$\mathcal{M}(S) \equiv \mathcal{M}_0^{1/2} \exp(S) \mathcal{M}_0^{1/2}. \tag{5.12}$$

The metric field defined on elements is discontinuous in $\mathcal{T}_h$, so it is advantageous to construct a continuous metric field. First, metrics are constructed at the vertices, $\nu$, of the tessellation from a mean of the element metrics touching that vertex. Using the affine invariant framework, the mean can be computed by finding the metric that minimizes the distance to the elemental metrics (in the affine invariant sense, i.e., the magnitude of $S$):

$$\mathcal{M}_\nu = \text{mean}^{\text{affinv}}(\{\mathcal{M}_\kappa\}_{\kappa \in \omega(\nu)}) = \arg\min_{\mathcal{M}} \sum_{\kappa \in \omega(\nu)} \| \log\left(\mathcal{M}_\kappa^{-1/2} \mathcal{M} \mathcal{M}_\kappa^{-1/2}\right) \|_F^2.$$

where $\omega(\nu)$ is the set of elements touching node $\nu$ and $\| \cdot \|_F$ is the Frobenius norm of the metric tensor. A continuous metric can then be constructed by interpolating the vertex metrics:

$$\mathcal{M}(\boldsymbol{x}) \equiv \arg\min_{\mathcal{M}} \sum_{\nu \in \mathcal{V}(\kappa)} w_\nu(\boldsymbol{x}) \| \log\left(\mathcal{M}_\nu^{-1/2} \mathcal{M} \mathcal{M}_\nu^{-1/2}\right) \|_F^2, \boldsymbol{x} \in \kappa,$$

where $\mathcal{V}(\kappa)$ is the set of nodes in element $\kappa$ and $w_\nu(\boldsymbol{x})$ is the barycentric coordinate corresponding to vertex $\nu$.

### 5.2.2 Steady Output Error Minimization Definition and Relaxation

Given that the accuracy and computational expense of the discrete problem is dependent on the tessellation, $\mathcal{T}_h$, the adaptation problem can be cast as the constrained optimization problem to find the optimal tessellation, $\mathcal{T}_h^*$:

$$\mathcal{T}_h^* = \arg\inf_{\mathcal{T}_h} \mathcal{E}(\mathcal{T}_h) \quad \text{s.t.} \quad \mathcal{C}(\mathcal{T}_h) \leq \text{dof}_{\text{target}} \tag{5.13}$$

where $\mathcal{C}(\mathcal{T}_h)$ is the cost, in terms of degrees of freedom, of representing the solution in the space $\mathcal{V}_h^p$. The tessellation, $\mathcal{T}_h$, is defined by the nodal connectivity and locations, as such Eq. (5.13) is a generally-intractable discrete-continuous optimization problem. Following the approach of Loseille and Alauzet[56] this problem can be made tractable by relaxing the control of the tessellation to the control of a Riemannian metric field, $\{\mathcal{M}(\boldsymbol{x})\}_{\boldsymbol{x}\in\Omega}$:

$$\begin{aligned} \mathcal{M}^* &= \arg\inf_{\mathcal{M}} \mathcal{E}(\mathcal{T}_h(\mathcal{M})) \quad \text{s.t.} \quad \mathcal{C}(\mathcal{T}_h(\mathcal{M})) \leq \text{dof}_{\text{target}} \\ &= \arg\inf_{\mathcal{M}} \mathcal{E}(\mathcal{M}) \quad \text{s.t.} \quad \mathcal{C}(\mathcal{M}) \leq \text{dof}_{\text{target}} \end{aligned} \tag{5.14}$$

The cost can be expressed in terms of the metric field by

$$\begin{aligned} \mathcal{C}(\mathcal{M}) &= \int_\Omega c(\mathcal{M}(\boldsymbol{x}), \boldsymbol{x})\, dx \\ &= \int_\Omega c_p \sqrt{\det(\mathcal{M}(\boldsymbol{x}))}, \end{aligned} \tag{5.15}$$

with $c_p$ a constant for each element dependent on the polynomial order of the solution. To estimate the behavior of $\mathcal{E}(\mathcal{M})$, the error functional, it is necessary to assume that the error is the sum of local contributions, i.e.,

$$\mathcal{E}(\mathcal{M}) = \int_\Omega e(\mathcal{M}(\boldsymbol{x}), \boldsymbol{x})\, d\boldsymbol{x} \tag{5.16}$$

where $e(\cdot,\cdot) : Sym_d^+ \times \mathbb{R}^d \to \mathbb{R}^+$. While this assumption is formally only applicable to local errors such as projection errors, it has been found to work well for the output-based error estimates used herein. The elemental error can then be expressed as

$$\eta_\kappa = \int_\kappa e(\mathcal{M}(\boldsymbol{x}), \boldsymbol{x})\, d\boldsymbol{x} \approx \int_\kappa e(\mathcal{M}_\kappa, \boldsymbol{x})\, d\boldsymbol{x} \Rightarrow \eta_\kappa = \eta_\kappa(\mathcal{M}_\kappa)$$

### 5.2.3 Local Error Sampling

In the previous section, the error minimization problem is posed in terms of local error Eq. (5.16) and cost Eq. (5.15) contributions. Whereas the cost is easily derived from the

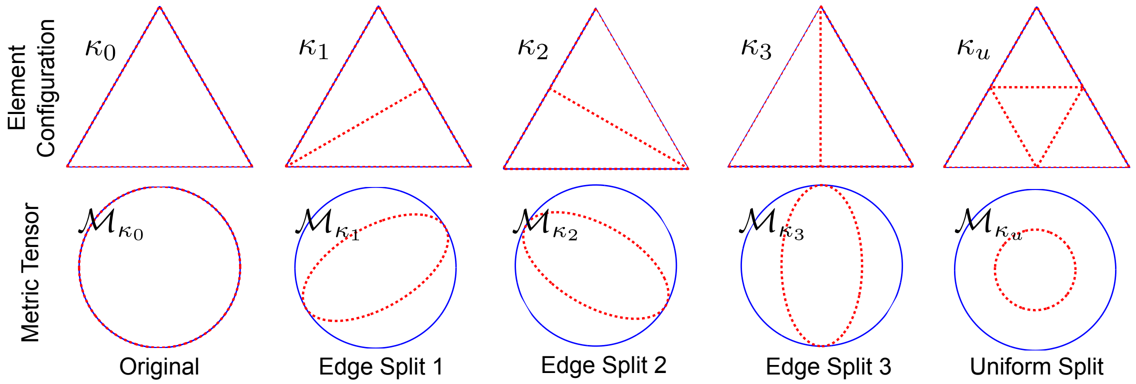


Figure 5-2: Original, edge split, and uniform split configurations and corresponding metric modifications (Yano[127])

solution order and metric in each cell, an estimate of the error behavior, $\eta_\kappa(\mathcal{M}_\kappa)$ is still needed. For quadrilateral elements, Houston et al.[47] propose directly monitoring the change in error with change in element shape. A similar approach is taken by Richter[95]. For an unstructured mesh this monitoring is accomplished by local sampling, whereby each element undergoes a set of metric modifications. For a simplex element with original configuration $\kappa_0$, the metric is modified by splitting each of the $d \times (d+1)/2$ edges in turn, or by a uniform refinement. This is shown in two dimensions in Figure (5-2) which shows edge splits and the corresponding metric changes for a reference element. The element error estimate Eq. (5.3) can then be re-evaluated as the sum over all the sub-elements $j$ in each configuration,

$$\eta_{\kappa_i} \equiv \sum_j \left| \mathcal{R}_{hp}(\boldsymbol{u}_{hp}^{\kappa_i^j}, \boldsymbol{\psi}_{h\,p+1}|_{\kappa_0}) \right|, \quad i = 0, \dots, n_{config}, \tag{5.17}$$

to construct metric-error pairs, $\{\mathcal{M}_{\kappa_i}, \eta_{\kappa_i}\}_{i=1}^{n_{config}}$. Note that due to Galerkin orthogonality it is not necessary to compute a local update to the adjoint state.

### 5.2.4 Local Error Model

Using the metric-error pairs from the local sampling of the error, a continuous local error model $\eta_\kappa(\cdot) : Sym_d^+ \to \mathbb{R}^+$ can be constructed. First, the metrics in the metric-error pairs are expressed as the logarithmic map of the metric relative to $\mathcal{M}_{\kappa_0}$, the inverse of Eq. (5.12),

$$S_{\kappa_i} = \log\left(\mathcal{M}_0^{-1/2} \mathcal{M}_{\kappa_i} \mathcal{M}_0^{-1/2}\right), \quad i = 0, \dots, n_{config}.$$

Similarly, the errors are measured logarithmically relative to the original configuration $\kappa_0$:

$$f_{\kappa_i} = \log\left(\eta_{\kappa_i}/\eta_{\kappa_0}\right), \quad i = 0, \ldots, n_{config}.$$

To relate $f_\kappa$ to $S_\kappa$, a linear function in the entries of $S_\kappa$ is selected

$$f_{\kappa_i}(S_\kappa) = \mathrm{tr}(R_\kappa S_\kappa) \tag{5.18}$$

where the rate tensor, $R_\kappa$, satisfies a least-squares fit of the sampled data. The least-squares reduces to an interpolation when $n_{config}$ matches the degrees of freedom of the symmetric tensor, e.g., when the isotropic refinement is omitted from the sampling. This error model is a generalization of an isotropic power law to the anisotropic tensor:

$$\eta_\kappa^{iso}(h) = \eta_{\kappa_0}\left(h/h_0\right)^{r_\kappa^{iso}}$$

where $r_\kappa^{iso}$ is the isotropic convergence rate, hence the name "rate tensor" for $R_\kappa$.

The vertex-based metric, $\mathcal{M}_\nu$, by combining with metric interpolation, can be used to reconstruct a continuous representation of the metric field. The vertex metric can generally be used directly by the mesh generator, so it is advantageous to recast the error model in terms of a metric field defined at vertices. Assign the change in configuration metric, $S_\kappa$, as the arithmetic mean of the change at its vertices,

$$S_\kappa = \overline{\{S_\nu\}_{\nu\in\mathcal{V}(\kappa)}} \equiv \frac{1}{|\mathcal{V}(\kappa)|}\sum_{\nu\in\mathcal{V}(\kappa)} S_\nu.$$

The error can then be expressed as

$$\mathcal{E}(\{S_\nu\}_{\nu\in\mathcal{V}(\kappa)}) = \sum_{\kappa\in\mathcal{T}_h} \eta_\kappa\left(\overline{\{S_\nu\}_{\nu\in\mathcal{V}(\kappa)}}\right),$$

### 5.2.5 Local Cost Model

Using the same method as for the local error model, the cost function can be expressed using $S_\kappa$

$$\rho_\kappa(S_\kappa) = \int_\kappa c_p\sqrt{\det\mathcal{M}(\boldsymbol{x})}\,d\boldsymbol{x} = \int_\kappa c_p\sqrt{\det(\mathcal{M}_{\kappa_0}^{1/2}\exp\left(S_\kappa\right)\mathcal{M}_{\kappa_0}^{1/2})}\,d\boldsymbol{x} = \rho_{\kappa_0}\exp\left(\frac{\mathrm{tr}(S_\kappa)}{2}\right).$$

In terms of $S_\nu$, the cost is

$$\mathcal{C}(\{S_\nu\}_{\nu\in\mathcal{V}(\kappa)}) = \sum_{\kappa\in\mathcal{T}_h} \rho_\kappa\left(\overline{\{S_\nu\}_{\nu\in\mathcal{V}(\kappa)}}\right).$$

### 5.2.6 Model Optimization

With expressions for the error and cost, the error minimization can now be formulated:

$$\begin{aligned}\{S_\nu^*\}_{\nu\in\mathcal{V}} = \underset{\{S_\nu\}_{\nu\in\mathcal{V}}}{\arg\inf}\, &\mathcal{E}\left(\{S_\nu\}_{\nu\in\mathcal{V}}\right)\\ \text{s.t.}\quad &\mathcal{C}\left(\{S_\nu\}_{\nu\in\mathcal{V}}\right) = \text{dof}_{\text{target}}\\ &|(S_\nu)_{ij}| \le \alpha, \quad i,j = 1,\dots,d, \quad \forall\nu\in\mathcal{V}.\end{aligned} \tag{5.19}$$

where the last term is a limit on the change allowed in the vertex metric for a given optimization step. To carry out the approximate optimization, gradients are computed for the error and cost, specifically with respect to a trace and trace-free decomposition of $S_\nu = s_\nu I + \tilde{S}_\nu$. The first order optimality condition is then:

$$\frac{\partial\mathcal{E}}{\partial s_\nu} + \lambda\frac{\partial\mathcal{C}}{\partial s_\nu} = 0, \tag{5.20}$$

$$\frac{\partial\mathcal{E}}{\partial\tilde{S}_\nu} = 0, \quad \forall\nu\in\mathcal{V}, \tag{5.21}$$

with a Lagrange multiplier $\lambda$. Defining a local Lagrange multiplier, $\lambda_\nu \equiv \frac{\partial\mathcal{E}}{\partial s_\nu}\Big/\frac{\partial\mathcal{C}}{\partial s_\nu}$, as the marginal change in $\mathcal{E}$ for a given change in $\mathcal{C}$, at optimality, $\lambda_\nu = \lambda, \forall\nu\in\mathcal{V}$.

To solve the optimization problem, Yano proposes the following gradient-based algorithm:

-1. Evaluate $\rho_{\kappa_0}$, $\eta_{\kappa_0}$, and $R_\kappa$ that define local cost and error models.

0. Set $\delta s = \alpha/n_{\text{step}}$, which limits the maximum change in the metric. Initialize $S_\nu = 0$, $\forall\nu\in\mathcal{V}$ and set $n = 0$.

1. Compute vertex derivatives and the local Lagrange multiplier ($\partial\mathcal{E}/\partial s_\nu$, $\partial\mathcal{E}/\partial\tilde{S}_\nu$, $\partial\mathcal{C}/\partial s_\nu$, and $\lambda_\nu$) about the current $\{S_\nu\}_{\nu\in\mathcal{V}}$.

2. Improve the equidistribution of the local Lagrange multiplier by updating the isotropic part of $S_\nu$ according to:

   - Refine a fixed fraction (here 30%) of the vertices with the largest $\lambda_\nu$, incrementing their $S_\nu$ by $\delta s I$
   - Coarsen a fixed fraction (here 30%) of the vertices with the smallest $\lambda_\nu$, decrementing their $S_\nu$ by $\delta s I$

3. Improve toward the local shape optimality condition, Eq. (5.21), modifying the anisotropic component of $S_\nu$ by decrementing it with $\delta s (\partial \mathcal{E} / \partial \tilde{S}_\nu)/(\partial \mathcal{E}/\partial s_\nu)$.

4. Rescale $S_\nu$ to obtain the desired degrees of freedom. That is, increment $S_\nu$ by $\beta I$, where $\beta$ is computed to satisfy the cost constraint in Eq. (5.19).

5. Set $n = n + 1$. If $n < n_{\text{step}}$ go back to 1.

Upon computing $S_\nu$, the metric field, $\mathcal{M}_\nu$, is updated and passed to a metric conforming mesh generator such as the INRIA-developed Bidimensional Anisotropic Mesh Generator (BAMG)[14, 46] in two dimensions or the Boeing Company-developed Edge Primitive Insertion and Collapse (EPIC)[68] in three dimensions.

## 5.3 Unsteady Algorithm Modification

For the unsteady case, the spatial metric-error sampling can be estimated locally in space (for a particular element) and time (for a particular timeslab). As the interest is only to control the spatial error, only spatial refinement is needed in the sampling. Figure (5-3) shows a 1d example spatial element, $k$, at timestep $n$ undergoing local sampling in the spatial dimension. The local sampling for element $\kappa$ integrated over timeslab $n$ with $I_n = (t_{n-1}, t_n)$ becomes:

$$\eta^{en}_{\kappa_i}(\mathcal{M}) = \sum_j \int_{I_n} \mathcal{R}^e_{hp}(\boldsymbol{u}^{\kappa_i^j}_{hp}, \boldsymbol{\psi}_{h\,p+1}\big|_{\kappa_0})\, dt, \quad \forall n \in [1, N], \;\; i = 0, \ldots, n_{config},$$

Combined with the contribution from the initial condition:

$$\eta^{e0}_{\kappa_i}(\mathcal{M}) = \mathcal{A}_{hp}(\boldsymbol{u}_{hp}(t_0), \boldsymbol{\psi}_{h\,p+1}(t_0)\big|_{\kappa_0}, \tilde{\boldsymbol{\psi}}_{h\,p+1}\big|_{\kappa_0}),$$

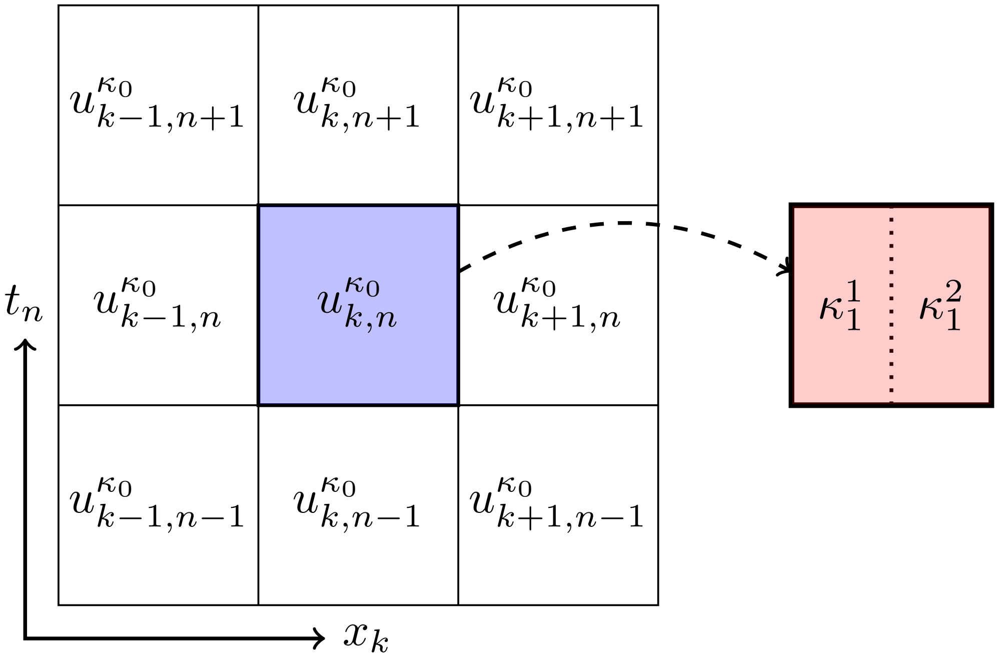


Figure 5-3: Timeslab spatial sampling – unsteady 1d example

the total spatial error sample for a given spatial element is

$$\eta^E_{\kappa_i}(\mathcal{M}) = \sum_{n=0}^{N} \eta^{en}_{\kappa_i}(\mathcal{M}).$$

The unsteady metric-error pairs $\{\mathcal{M}_{\kappa_i}, \eta^E_{\kappa_i}\}$ are then used to build up the local error model, Eq. (5.18), for each spatial element. Using the same timestepping scheme for each timeslab with a constant timestep, the temporal cost is the same for each timeslab and factors out of the optimization problem, Eq. (5.19), which can be solved as before.

The algorithm of Section 5.2.6 is modified by replacing Step -1 with the following:

-4. During forward time sweep, compute $\boldsymbol{u}^n_{hp}$

-3. During backward time sweep, compute $\boldsymbol{\psi}^n_{h\,p+1}$ and local sampling $\eta^{en}_{\kappa_i}$

-2. Evaluate $\eta^E_{\kappa_i}$ and $R_\kappa$ that define the local error models.

-1. Evaluate $\rho_{\kappa_0}$ that defines the local cost model.

From these modifications it is clear that once $\eta^E_{\kappa_i}$ are computed from each timeslab's contribution to the error model that the remainder of the algorithm is unchanged.

## 5.4 Adaptation Results

The effectiveness of unsteady MOESS adaptation is examined by applying it to a convecting, isentropic vortex. The algorithm is then applied to several laminar airfoils exhibiting unsteady wakes with different freestream and initial conditions. For each case, the algorithm is compared against the isotropic fixed fraction algorithm described by Modisette[69]. The fixed fraction algorithm is modified for unsteady problems by applying it to the time-integrated elemental error estimate of Eq. (5.11). The 20% of elements with the largest and smallest errors are marked for refinement or coarsening, respectively. Refined elements are decreased in area by a factor of 3, and coarsened elements have their requested area doubled.

### 5.4.1 Isentropic Vortex

The first unsteady problem considered is the convection of an isentropic vortex in uniform inviscid flow. This problem was selected because the resulting mesh is expected to be isotropic with a clearly defined region of mesh refinement. By removing mesh anisotropy, the isotropic fixed fraction and MOESS adaptation are expected to perform similarly and the isotropic performance of both methods can be evaluated.

The isentropic vortex as tested here is the same problem examined by Wang and Mavriplis[118]. The vortex is given by

$$\begin{aligned}
\rho &= T^{1/(\gamma-1)} = \left[1 - \frac{\alpha^2(\gamma-1)}{16\phi\gamma\pi^2} e^{2\phi(1-r^2)}\right]^{1/(\gamma-1)} \\
u &= u_\infty - \frac{\alpha}{2\pi}\left(y - y_o\right) e^{\phi(1-r^2)} \\
v &= \frac{\alpha}{2\pi}\left(x - x_o\right) e^{\phi(1-r^2)}
\end{aligned}$$

with $\alpha = 4$, $\phi = 1$, and freestream conditions:

$$(\rho_\infty, u_\infty, v_\infty, p_\infty, T_\infty) = (1, 1/2, 0, 1, 1).$$

These conditions correspond to $Ma_\infty = 0.423$. The exact solution is pure convection of the vortex with the freestream velocity. The vortex is initialized by an $L_2$ projection of the exact function into $\mathcal{V}_h^p$ at time $t_0 = 0$ with the vortex centered at $x = 0$. The vortex is

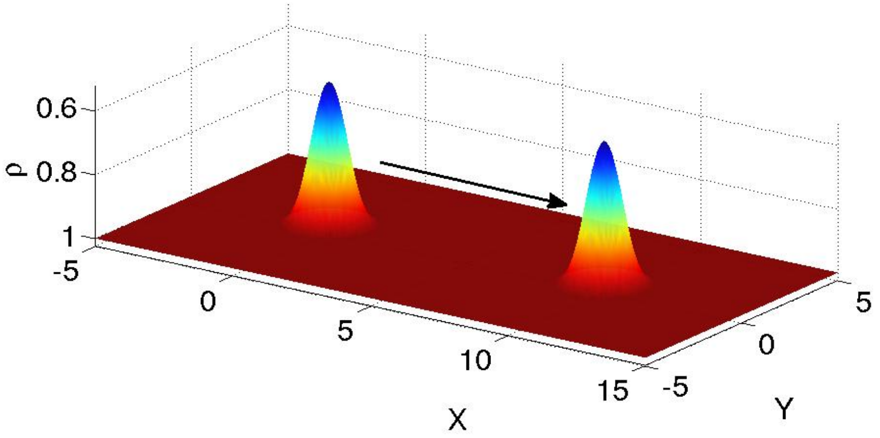


Figure 5-4: Initial & final density profile

solved forward in time to $t_f = 20$ with the bump centered at $x = 10$ using the $4^{th}$-order five-stage DIRK temporal scheme with a timestep of $\Delta t = 0.5$. A timestep independence study indicates that this timestep gives temporal error much smaller than the spatial error for the range of meshes evaluated in this study. The spatial domain extends $-10 \leq x \leq 20$ and $-10 \leq y \leq 10$ and is discretized with DG for $p = 1$, $p = 2$, and $p = 3$. The boundaries of the domain are a minimum 10 mesh units from the vortex center at any time so that the exact solution is within machine precision of the freestream condition at the boundaries.

The output of interest is the domain integral of momentum perturbation evaluated at the final time:

$$\mathcal{J} = \left[ \int_\Omega (\rho u - \rho_\infty u_\infty)^2 + (\rho v - \rho_\infty v_\infty)^2 \; dx|_{t=t_f} \right]^{1/2}$$

Because the exact solution is pure convection, any discretization error that is introduced into the solution will convect with the vortex. Therefore any spatial locations through which the same portion of the vortex travels should have the same spatial refinement.

Adaptation was initialized with a coarse uniform unstructured mesh. The mesh was then adapted 15 times at the desired number of degrees of freedom (DOF) to find each method's approximation of an optimal mesh at that DOF. The final mesh was used to initialize adaptation for the next higher DOF value. The output error was computed for

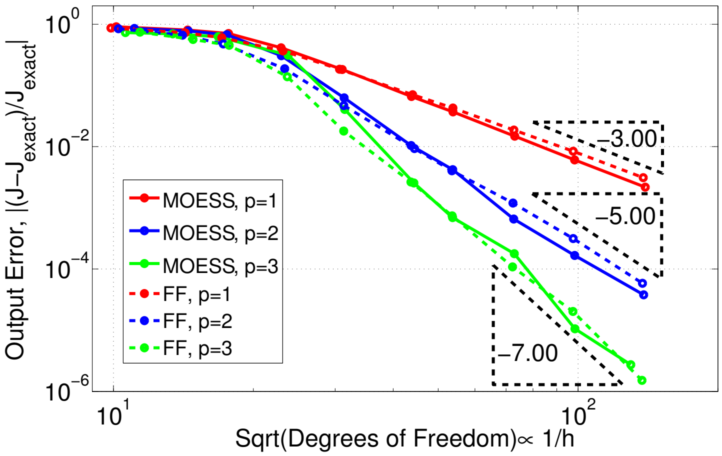
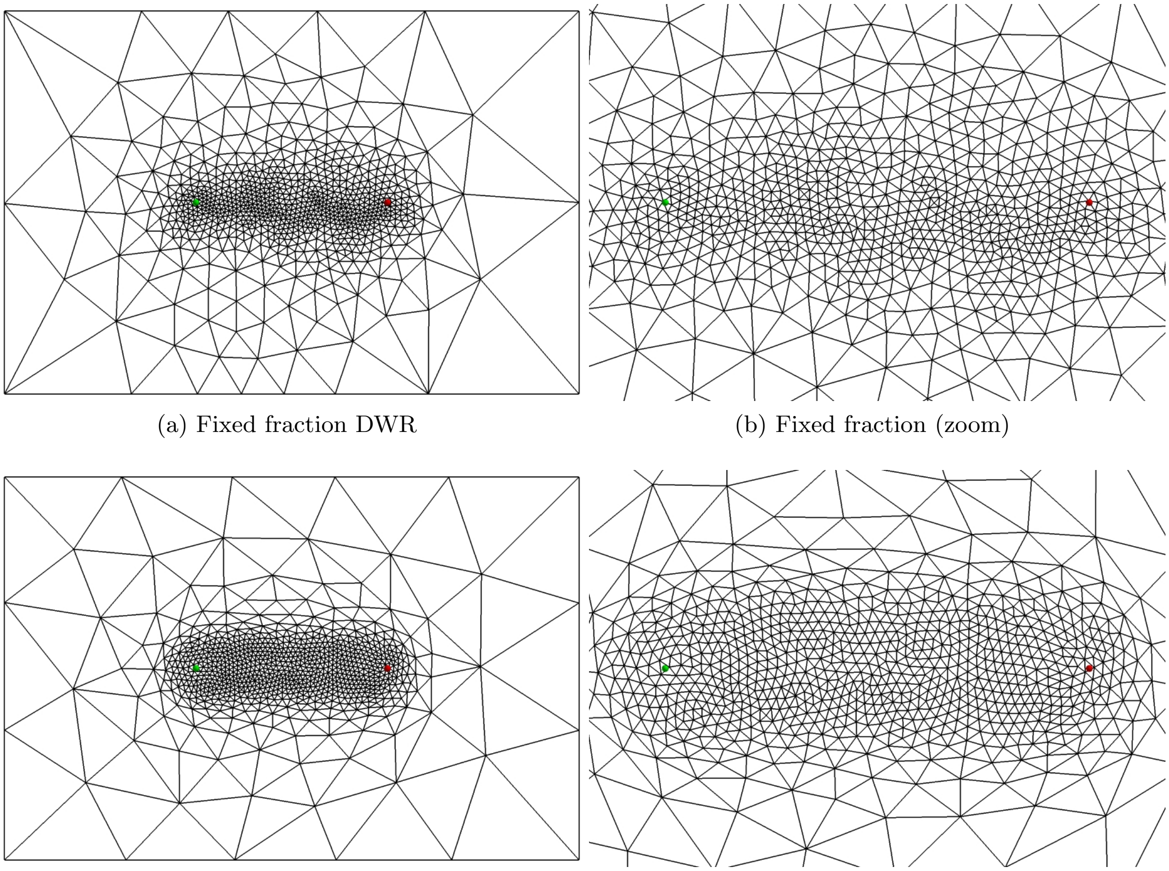


Figure 5-5: Convecting vortex - Fixed fraction DWR/MOESS comparison (error relative to continuous problem)

the last 5 iterations for each DOF, and the average plotted in Figure (5-5). The MOESS and fixed fraction approaches give nearly identical behavior. Both result in approximately $\mathcal{E} = \mathcal{O}(h^{2p+1})$ error convergence, a result that is consistent with the convective nature of the problem. The biggest difference between the meshes for the fixed fraction algorithm and the MOESS algorithm is the small amount of noise introduced into the mesh sizing for the fixed fraction algorithm shown in Figure (5-6). This is due to the fixed fraction of elements that are marked for refinement and coarsening not equilibrating to a smooth mesh metric field, but continually overshooting and correcting the refinement. One possible solution is to decrease the amount of change prescribed for the refined and coarsened elements, but the effect on the solution here appears to be minimal.

Whereas the primal solution is pure convection, the adjoint exhibits additional complexity. Figure (5-7) shows snapshots of the mass and x-momentum components of the adjoint at intervals of 5 time units. The largest magnitude of the adjoint tracks the primal vortex with weak disturbances that propagate outward from the vortex at the speed of sound. As it propagates backward in time, the large magnitude core takes on a spiral character not found in the primal.



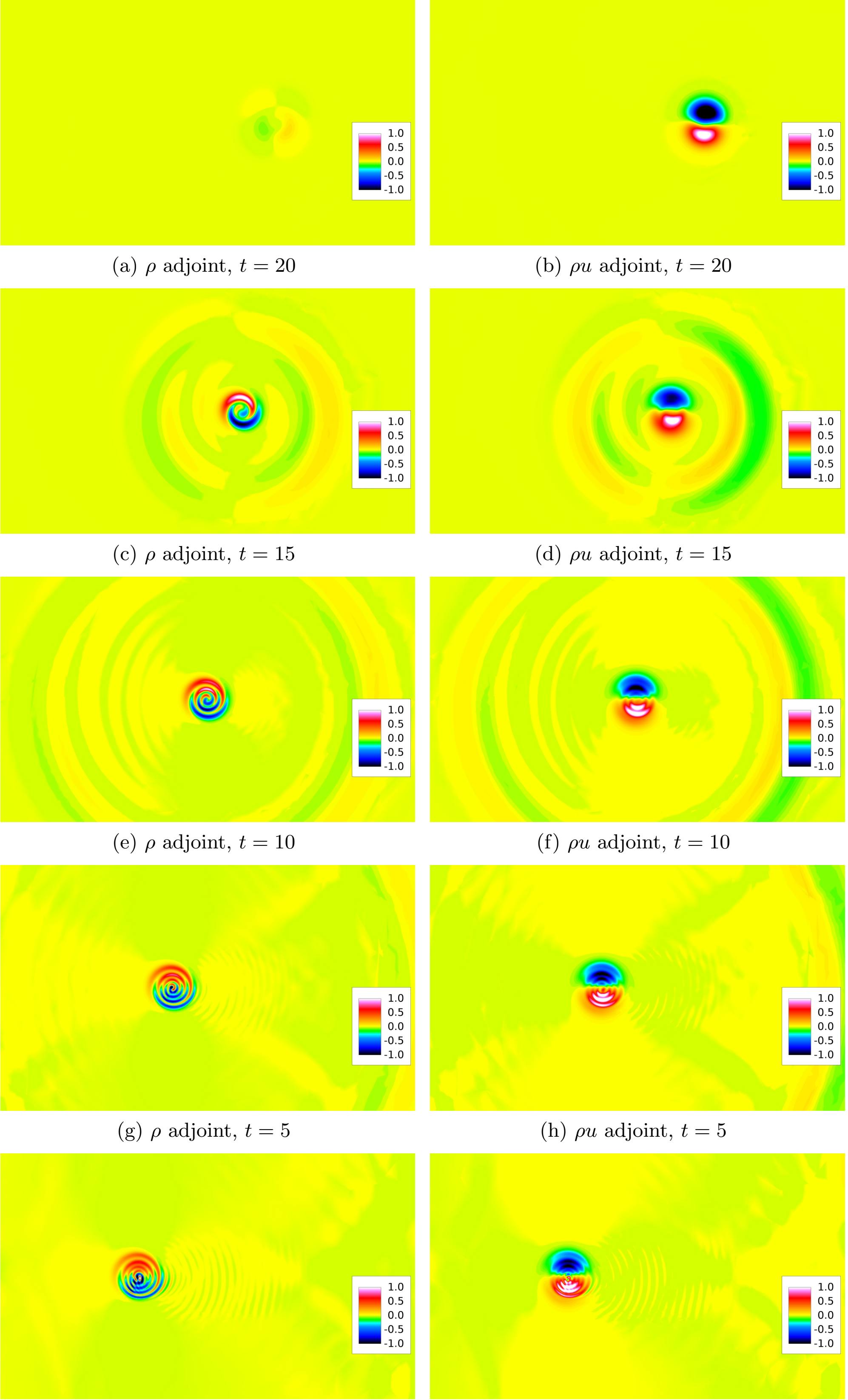
(a) Fixed fraction DWR

(b) Fixed fraction (zoom)

(c) MOESS approach

(d) MOESS approach (zoom)

Figure 5-6: Convecting vortex - Final mesh comparison



(a) $\rho$ adjoint, $t = 20$

(b) $\rho u$ adjoint, $t = 20$

(c) $\rho$ adjoint, $t = 15$

(d) $\rho u$ adjoint, $t = 15$

(e) $\rho$ adjoint, $t = 10$

(f) $\rho u$ adjoint, $t = 10$

(g) $\rho$ adjoint, $t = 5$

(h) $\rho u$ adjoint, $t = 5$

(i) $\rho$ adjoint, $t = 0$

(j) $\rho u$ adjoint, $t = 0$

Figure 5-7: Convecting vortex - Mass and x-momentum adjoint snapshots

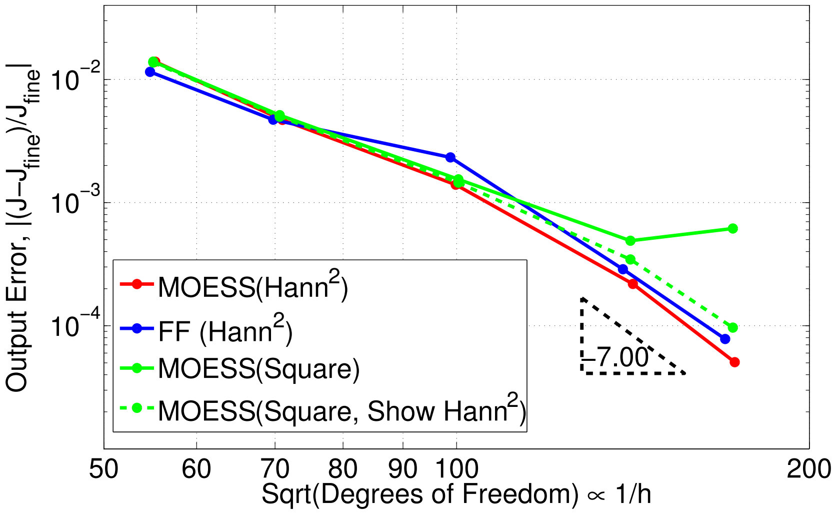


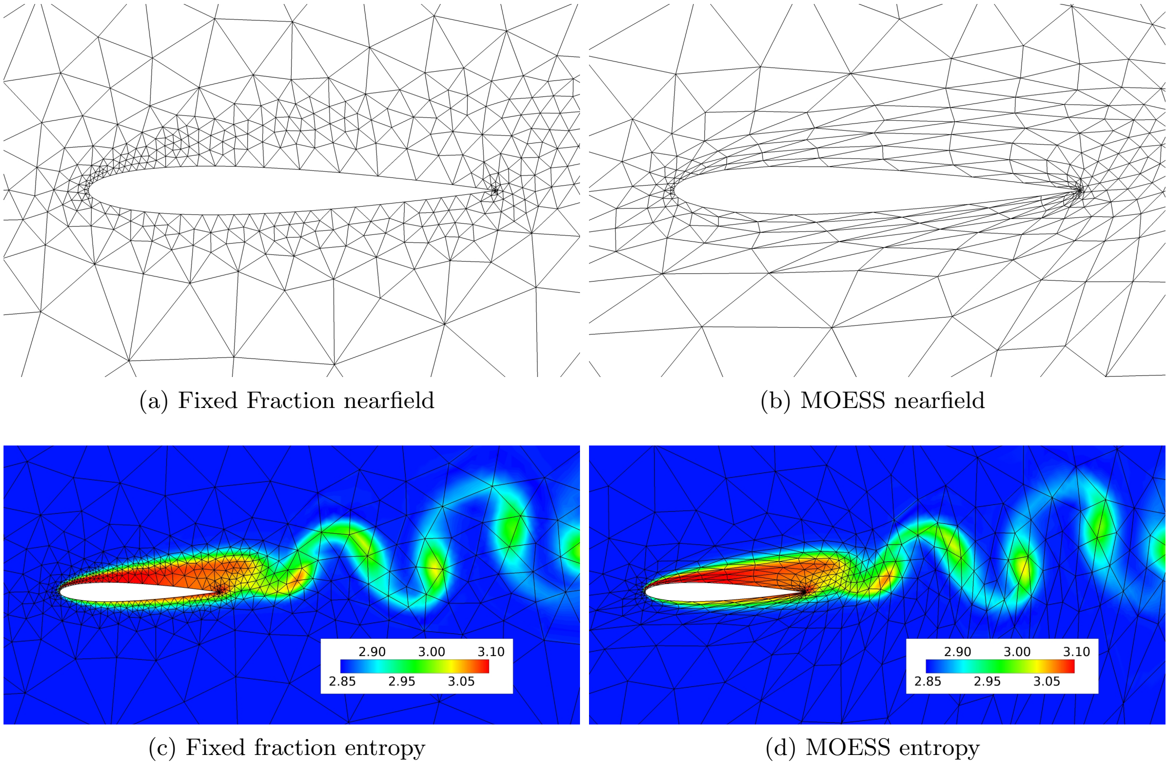
Figure 5-8: Periodic time-average lift error

### 5.4.2 NACA Airfoils

**NACA 0012 Long-Time Windowed Average,** $\alpha = 9°$**,** $Re = 1500$

For continuity with the previous sections, the NACA 0012 is first adapted for the same problem as used in Chapters (3) and (4). The same timestep of $\Delta t = 0.1$ is used here, now with the $4^{th}$-order, five-stage DIRK scheme. For adaptation the spatial domain is extended to $-50 \leq x, y \leq 50$. The time-average lift coefficient is computed using a $p = 3$ DG spatial discretization. A lead-in time of $t \in [-80, 0]$ is used to allow the primal to equilibrate in the forward time sweep and the adjoint in the backward sweep. The windowed time-average is then calculated over $t \in [0, 20]$ using both a Hann-squared and a square window. The same stationary adaptation procedure as for the isentropic vortex is employed here with 15 adaptation iterations at each DOF level.

Convergence of the output error with mesh size is plotted in Figure (5-8). At the highest DOFs tested (20k-30k), the convergence rate is approximately $\mathcal{E} = \mathcal{O}(h^{2p+1})$ for both the MOESS and fixed fraction algorithms using the Hann-squared window. There is no significant difference in error level between the adaptation methods for this problem. Unlike the Hann-squared windows, when the adaptation is driven by the output with the square window, it begins to level off just below $10^{-3}$ relative error. This is due to the limited accuracy of the square window in capturing a periodic output for the chosen window span, as expected from Figure (4-16). Also shown in Figure (4-16) is the convergence



(a) Fixed Fraction nearfield

(b) MOESS nearfield

(c) Fixed fraction entropy

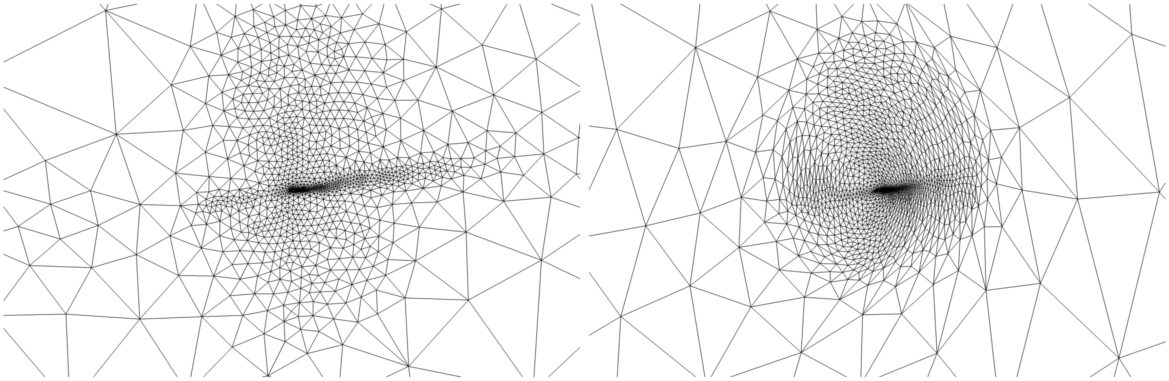
(d) MOESS entropy

Figure 5-9: NACA 0012 periodic average lift - nearfield mesh and entropy (30k DOF)

of the Hann-squared output computed from the MOESS mesh adapted using the square window output. This output converges without any degradation. The insensitivity of the adaptation's performance to the detail of the unsteady output (Hann-squared versus square windows) reinforces the applicability to wake problems of a single spatial mesh.

Because of the low Reynolds number and high $p$ order, Figure (5-9b) shows the MOESS approach only introduces a small amount of boundary layer anisotropic refinement. This allows some additional refinement in the wake, which does not significantly improve the overall error performance with respect to the fixed fraction DWR approach, but does result in a somewhat smoother convergence history. Figures (5-9c) and (5-9d) show entropy snapshots and mesh of the vortex street for both adaptation algorithms. Both methods result in very similar vortex streets even with the difference in nearfield anisotropy.

The farfield of Figure (5-10) shows refinement several body lengths away from the airfoil. The radius of the refined region is consistent with the maximum range for a disturbance originating at the surface within the time span of the windowed output. Snapshots of the Mach number and the adjoint of the x-momentum equation are shown in Figure (5-11).


(a) Fixed fraction farfield

(b) MOESS farfield

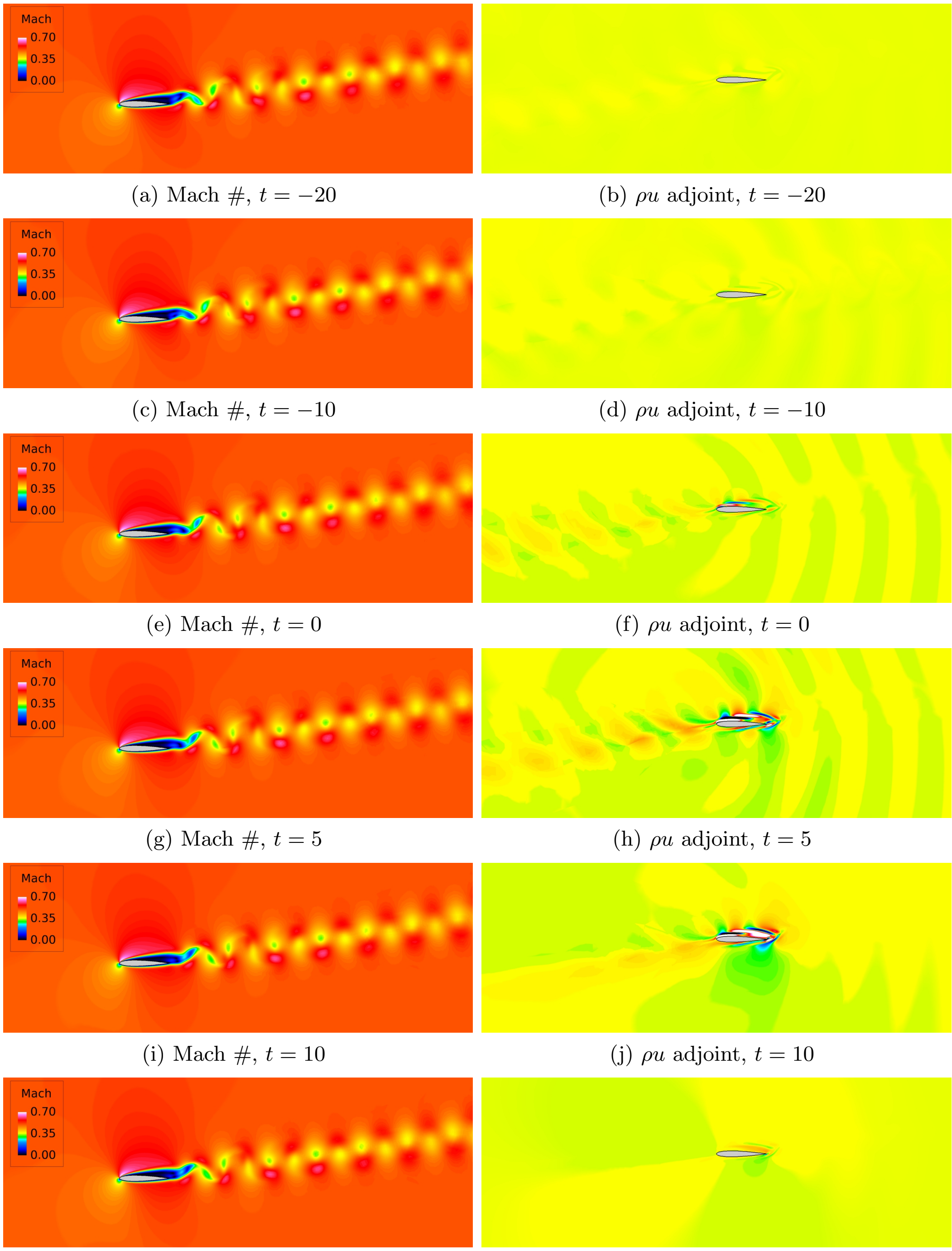
Figure 5-10: NACA 0012 periodic average lift - farfield (31k DOF)

Starting at the final snapshot and progressing backward in time, the magnitude of the adjoint increases as the Hann-squared window function increases. As the window tapers off ($t = 5$ and $t = 0$), the adjoint begins to develop an oscillatory character in the wake and along the stagnation line that matches the vortex shedding frequency. The magnitude of the adjoint does not diminish as quickly as the window function and is not symmetric over the window span. In the unforced region, $t < 0$, the adjoint slowly decreases magnitude until it is truncated at $t = -80$. The mach number shows the same equilibrium vortex street throughout.

### Impulsively Started NACA 0012

The second test with the NACA 0012 geometry in laminar viscous flow is an impulsive start, where for $t \geq 0$ the freestream conditions are $Ma = 0.25$, $\alpha = 8°$, $Re = 5000$. A similar problem was investigated by Fidkowski & Luo[30]. Here the entire flow field is initialized to uniform freestream, which introduces an inconsistency with the no-slip boundary condition at the airfoil surface. The simulation is run for $t \in [0, 10]$ using the fourth-order DIRK with $\Delta t = 0.05$. The output is the time-average over the last convective time unit: $\mathcal{J} = \int_9^{10} C_L \, dt$. DG $p = 1$ and $p = 2$ are used spatially with expected convergence rate $\mathcal{E} = \mathcal{O}(h^{(p+1)}) = \mathcal{O}(DOF^{-(p+1)/2})$. The stationary adaptation procedure was again employed with 15 adaptive iterations at each DOF level, averaging the output from the final 5 iterations.

For $p = 2$, MOESS gives consistently lower error at the same DOF as compared to the isotropic fixed fraction DWR. The lower order $p = 1$ adaptation shows a larger improvement



(a) Mach #, $t = -20$

(b) $\rho u$ adjoint, $t = -20$

(c) Mach #, $t = -10$

(d) $\rho u$ adjoint, $t = -10$

(e) Mach #, $t = 0$

(f) $\rho u$ adjoint, $t = 0$

(g) Mach #, $t = 5$

(h) $\rho u$ adjoint, $t = 5$

(i) Mach #, $t = 10$

(j) $\rho u$ adjoint, $t = 10$

(k) Mach #, $t = 15$

(l) $\rho u$ adjoint, $t = 15$

Figure 5-11: NACA 0012 periodic average - Mach # and x-momentum adjoint snapshots

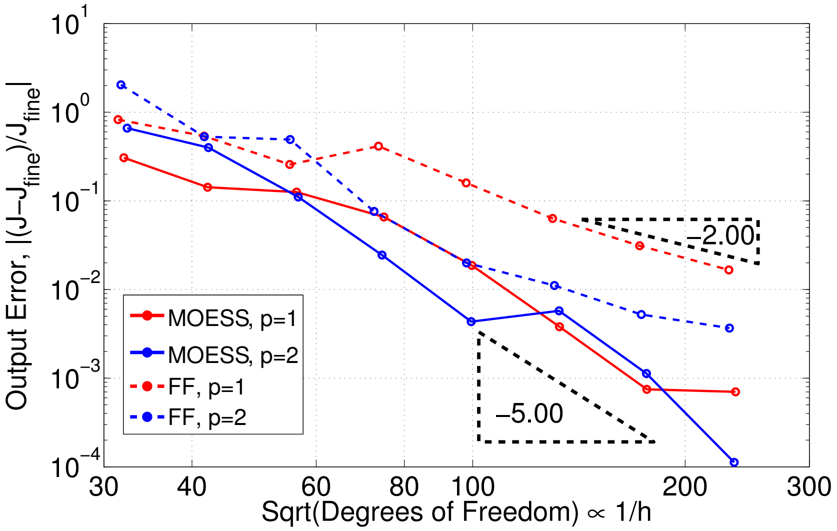
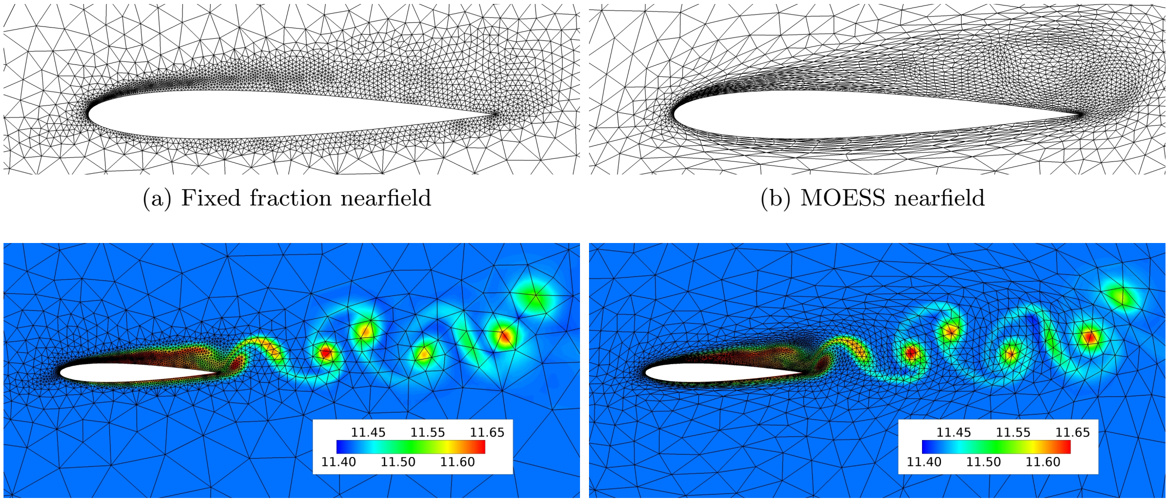


Figure 5-12: NACA 0012 impulse start average lift over $t \in [9, 10]$

by using MOESS because more anisotropy is introduced in the near wall mesh. The reason for this is that the relatively modest increase in Reynolds number from 1500 in Section 5.4.2 to 5000 requires more boundary layer refinement, as shown in Figure (5-13). The anisotropic capability of MOESS is more efficient in the viscous layer, allowing additional degrees of freedom to be used in the unsteady wake region (compared to the isotropic fixed fraction method). As with the long-time average, the farfield shows refinement several body lengths from the airfoil. This is shown in Figure (5-14). Here the domain of refinement is limited by the maximum range for signals that originate at $t = 0$ at the surface and impact the lift coefficient for $9 \leq t \leq 10$,the timespan of interest for the output. Figure (5-15) shows the lift coefficient time history at several DOF levels for both adaptation approaches and $p$ orders. After the initial transient, the lift coefficient appears to be moving toward a limit cycle oscillation. Snapshots of the Mach number and the adjoint of the x-momentum equation are shown in Figure (5-16). Concurrent with the starting vortex formation and convection, a large-magnitude, alternating-sign section of adjoint convects upstream (in backward time) along the stagnation line. Simultaneous with the first unsteady vortex separation between $t = 3.5$ and $t = 4.0$, there is an increase (in decreasing time) in the magnitude of the adjoint on the airfoil upper surface. This corresponds to the sensitivity of the output to the onset of vortex separation, and hence to the starting vortex and the stability of the incipient separation bubble. This results in a relatively small change in the solution during the problem startup having a large effect on $C_L$ in the output window,



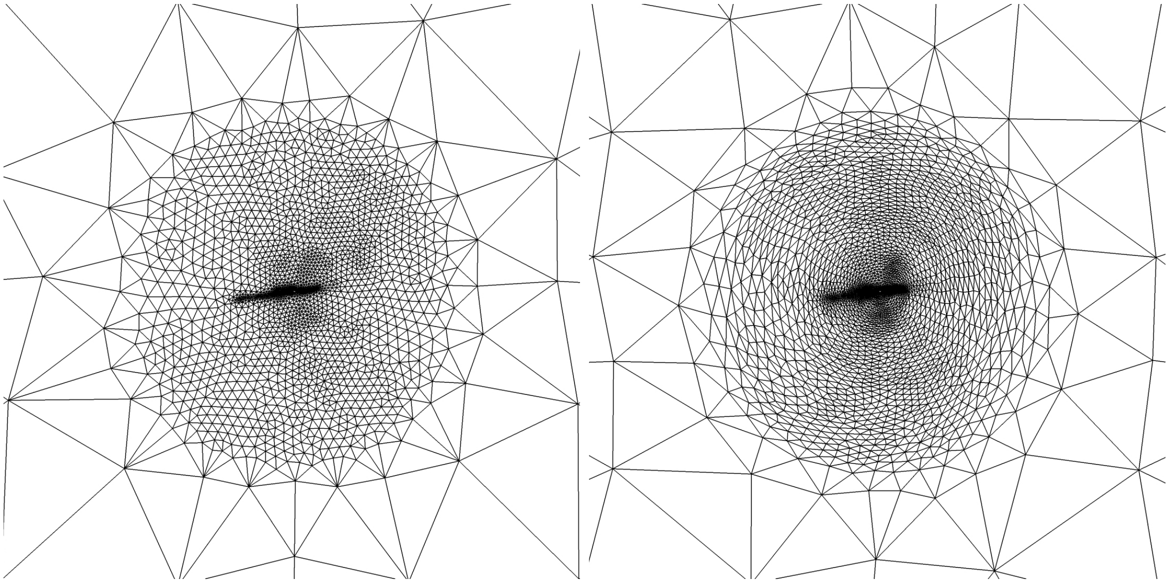
(a) Fixed fraction nearfield

(b) MOESS nearfield

(c) Fixed fraction entropy

(d) MOESS entropy

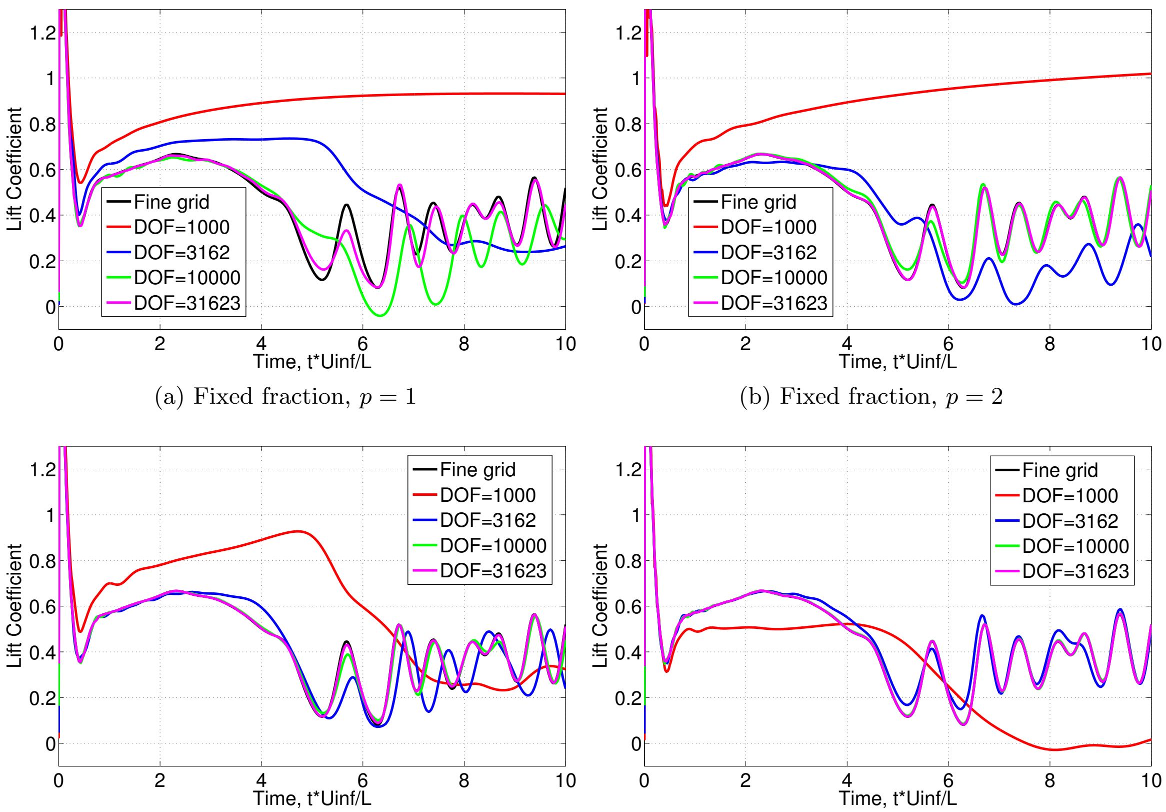
Figure 5-13: NACA 0012 impulse start - nearfield mesh and entropy, $p = 2$, $t = 10$



(a) Fixed fraction farfield

(b) MOESS farfield

Figure 5-14: NACA 0012 impulse start - farfield & nearfield, $p = 2$



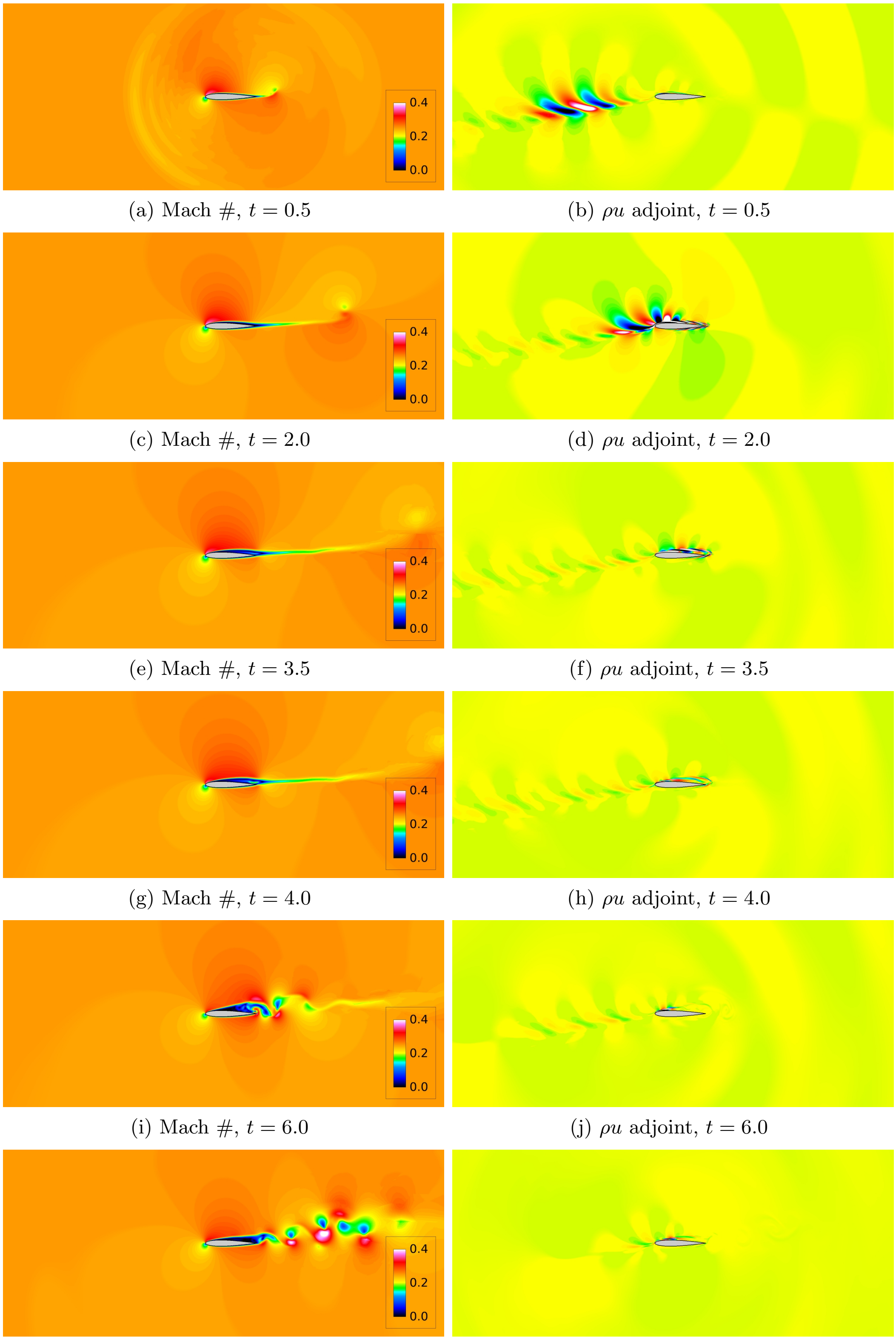
(a) Fixed fraction, $p = 1$

(b) Fixed fraction, $p = 2$

(c) MOESS, $p = 1$

(d) MOESS, $p = 2$

Figure 5-15: NACA 0012 impulse start - $C_L$ history



(a) Mach #, $t = 0.5$

(b) $\rho u$ adjoint, $t = 0.5$

(c) Mach #, $t = 2.0$

(d) $\rho u$ adjoint, $t = 2.0$

(e) Mach #, $t = 3.5$

(f) $\rho u$ adjoint, $t = 3.5$

(g) Mach #, $t = 4.0$

(h) $\rho u$ adjoint, $t = 4.0$

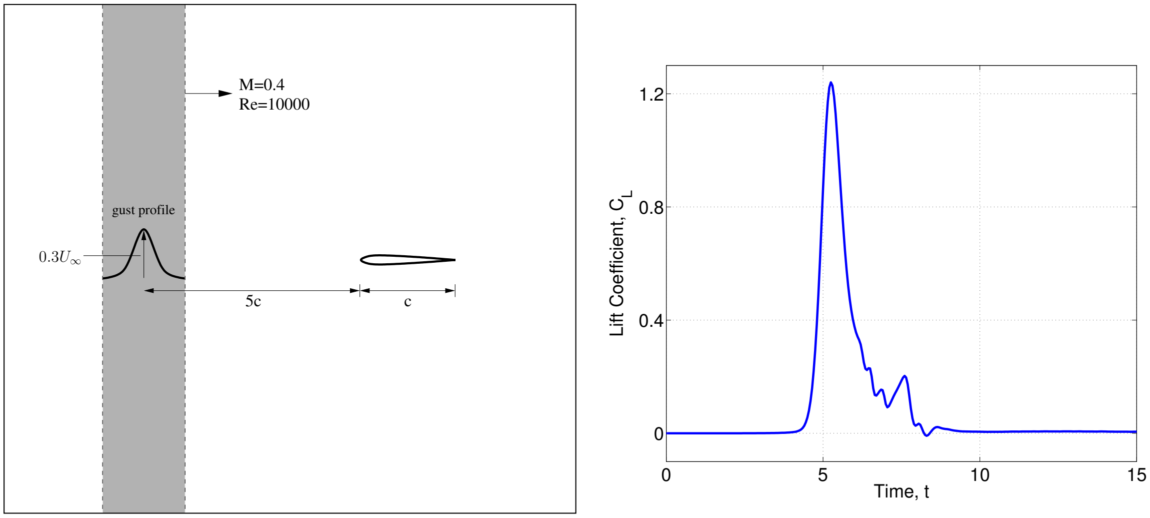
(i) Mach #, $t = 6.0$

(j) $\rho u$ adjoint, $t = 6.0$

(k) Mach #, $t = 8.0$

(l) $\rho u$ adjoint, $t = 8.0$

Figure 5-16: NACA 0012 impulse start - Mach # and x-momentum adjoint snapshots



(a) Problem setup

(b) Lift history

Figure 5-17: NACA 0006 gust encounter - problem setup and fine mesh lift history

$t \in [9, 10]$. Once the separation bubble becomes unstable, however, the flow is less sensitive to further perturbations.

### NACA 0006 Subsonic Gust, $\alpha = 0°$, $Re = 10000$

The final laminar flow test replaces the NACA 0012 airfoil with a thinner NACA 0006. Freestream conditions for this test are $Ma = 0.4$, $\alpha = 0°$, and $Re = 10000$. The initial condition is the steady solution superimposed with a vertical gust profile centered 5 chord lengths ahead of the leading edge ($x_c = -5c$). The gust, shown in Figure (5-17a) is described by a Gaussian profile with a peak magnitude of $0.3U_\infty$ and a standard deviation of one-quarter chord. The gust profile is clipped and its magnitude is shifted such that the perturbation is nonzero only for $|x - x_c| \leq 4$ and is $C^0$ continuous at $|x - x_c| = 4$: Additionally, the gust profile is smoothed to zero near the upper and lower boundaries of the domain to maintain consistency with the freestream boundary condition. The gust profile, $\boldsymbol{u}_{gust}$, is introduced into the initial condition by $\boldsymbol{A} = \boldsymbol{R}(\boldsymbol{u}(0) - \boldsymbol{u}_{gust}) = 0$. Rather than interpolating the gust into the finite-dimensional solution space, $\mathcal{V}_h^p$, the exact gust is evaluated and subtracted from the state (and its gradient from the state gradient) at each quadrature point in the residual evaluation. The simulation is run for $t \in [0, 10]$ using the fourth-order DIRK with $\Delta t = 0.05$. Solution orders $p = 1$ and $p = 2$ are used spatially. The adaptation procedure was again employed with 15 adaptive iterations at each DOF level,

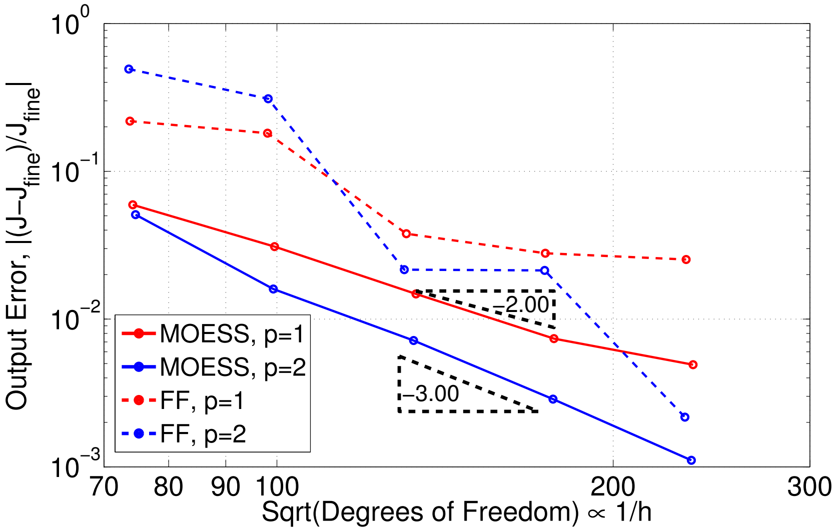
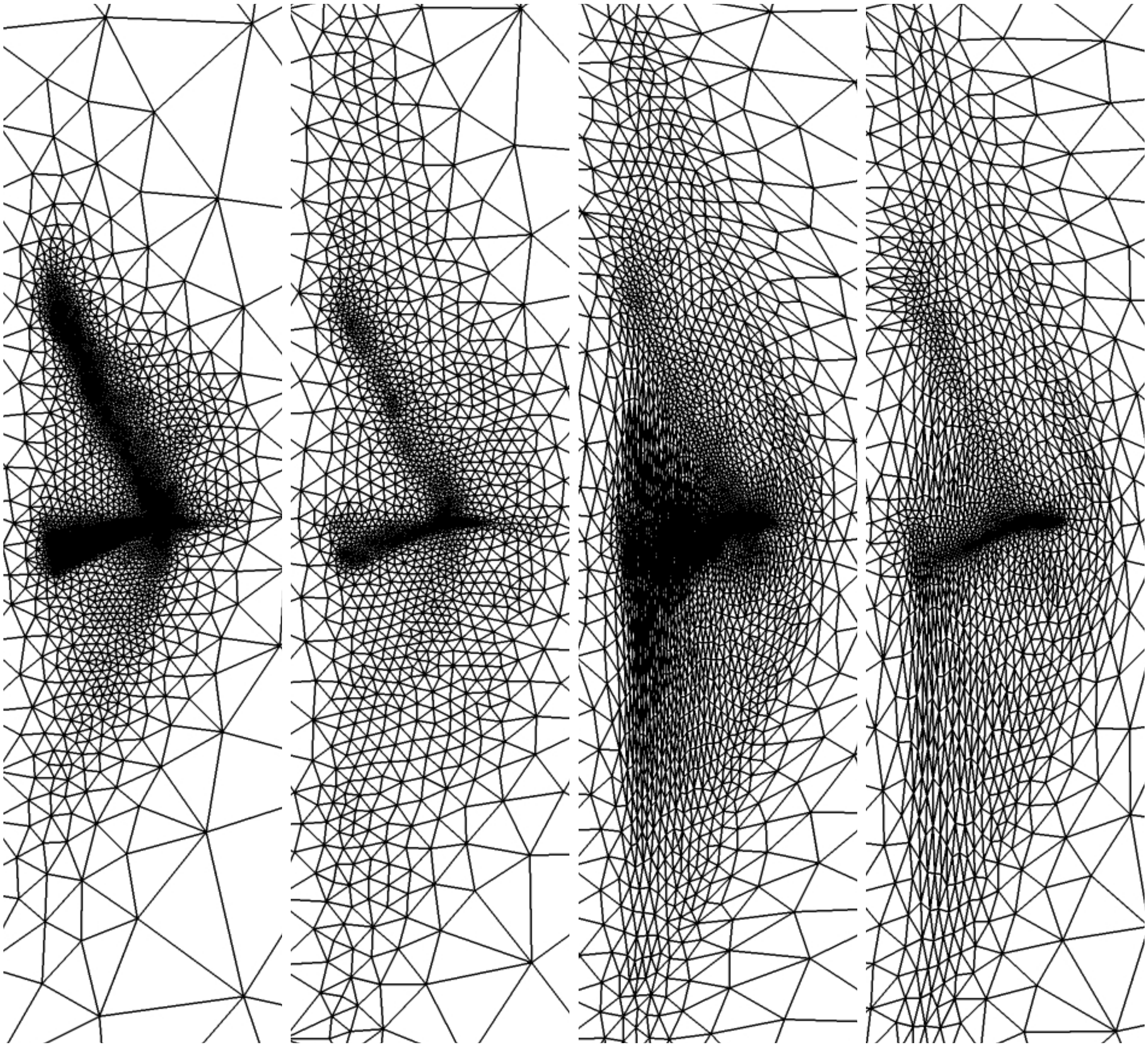


Figure 5-18: NACA 0006 gust encounter - output error

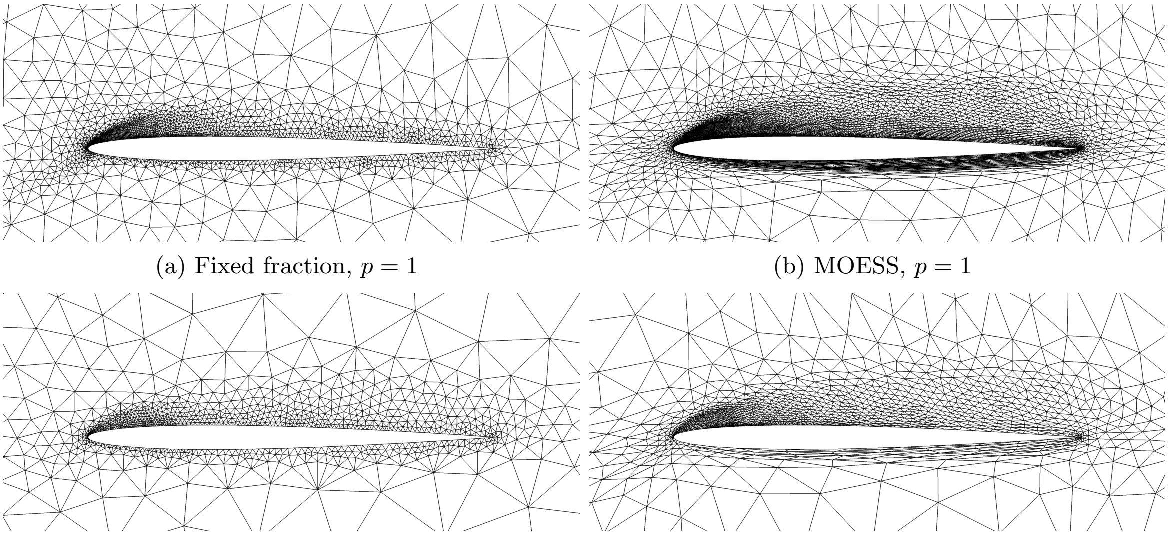
averaging the outputs from the final 5 iterations. The output is the time-average lift coefficient over the entire timespan. The fine mesh output history is shown in Figure (5-17b). The lift is initially zero and remains so until the gust reaches the airfoil. Because the airfoil is at zero angle of attack, the lift spikes and oscillates for a short time before settling back toward zero.

As with the impulsively started NACA 0012, the increased Reynolds number leads to a relative improvement of MOESS (which allows anisotropy) over the fixed fraction isotropic approach. Both $p = 1$ and $p = 2$ MOESS converge at $\mathcal{E} = \mathcal{O}(h^{p+1})$, though the slope of the $p = 1$ line does lessen for the last point. The fixed fraction isotropic error convergence is erratic, between a factor of 2 and an order of magnitude larger than the same $p$ MOESS results. The overall convergence rate is slower than for the previous airfoil cases. This is likely due to the non-smooth introduction of the gust, decreasing the regularity of the solution. The farfield meshes for these simulations, shown in Figure (5-19), give some further insight into the slower convergence. The mesh refinement tracking the gust extends up to 25 chord lengths above and below the airfoil, and the mesh is exceptionally fine for $p = 1$. This leaves fewer degrees of freedom available for refinement near the body. The fixed fraction and MOESS nearfield meshes at $31k$ DOF are shown in Figure (5-20), demonstrating the anisotropy introduced by the MOESS algorithm. A snapshot of the entropy at $t = 7.5$ in Figure (5-22) shows the vortices induced by the passing gust, which is faintly visible a body length and a half downstream of the airfoil. The vortices are more clearly defined in



(a) Fixed fraction, $p = 1$ (b) Fixed fraction, $p = 2$ (c) MOESS, $p = 1$ (d) MOESS, $p = 2$

Figure 5-19: NACA 0006 gust encounter - farfield (31k DOF)



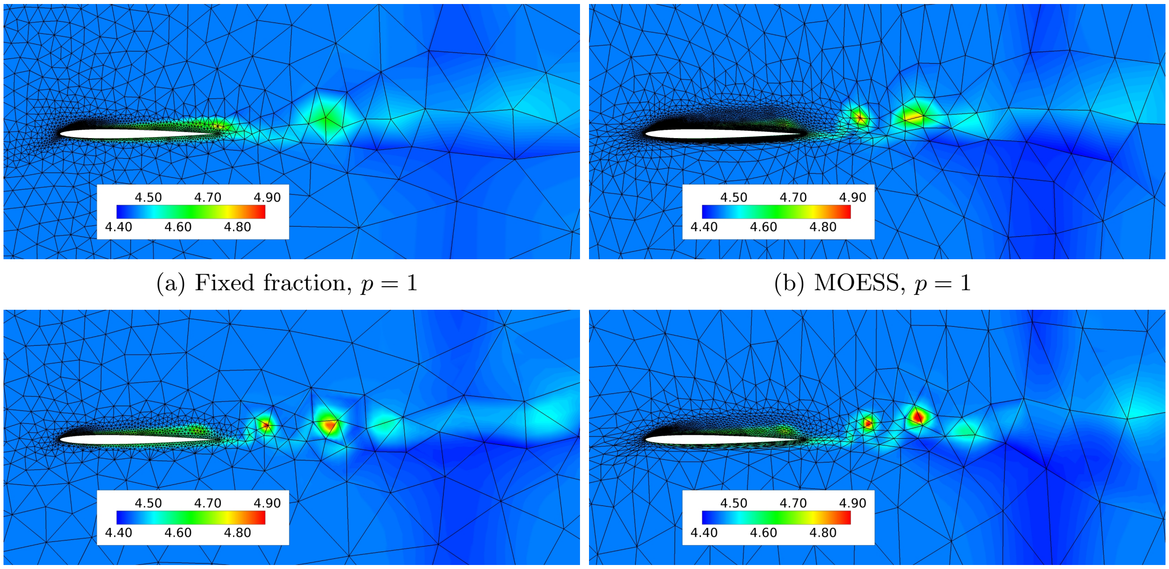
(a) Fixed fraction, $p = 1$

(b) MOESS, $p = 1$

(c) Fixed fraction, $p = 2$

(d) MOESS, $p = 2$

Figure 5-20: NACA 0006 gust encounter - nearfield mesh (31k DOF)



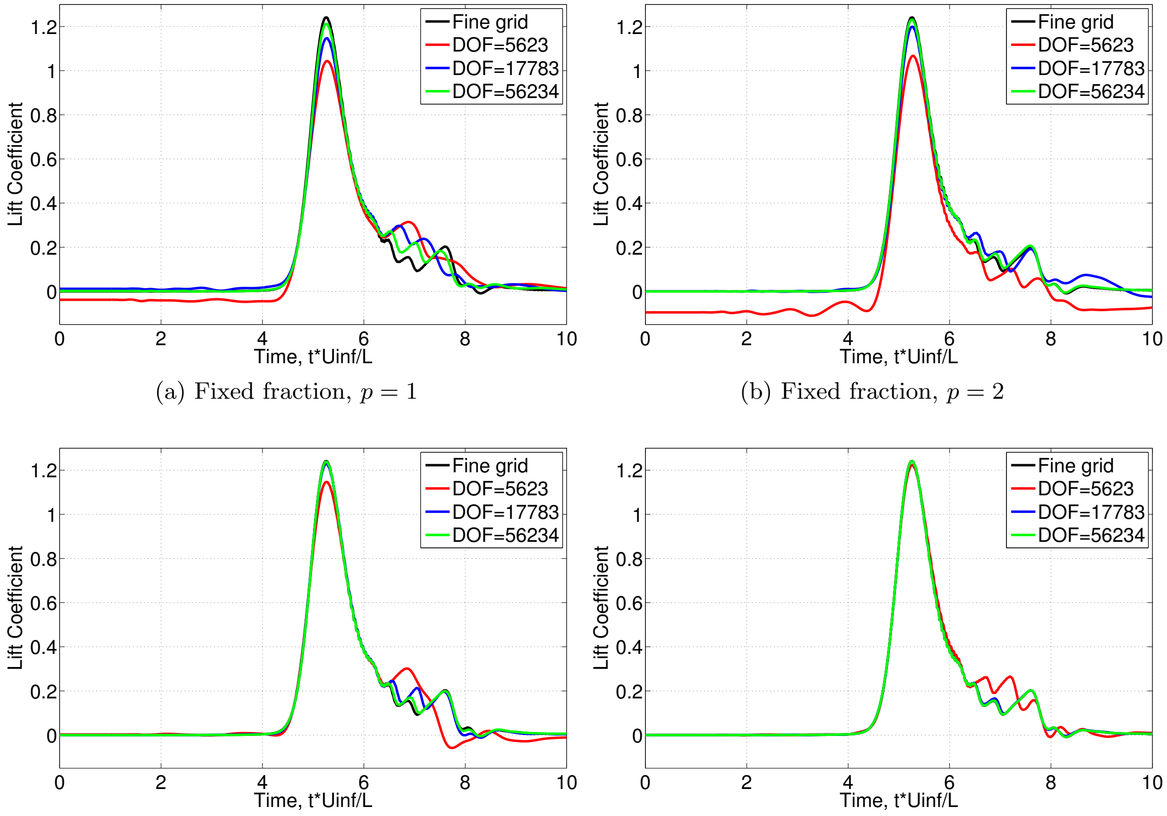
(a) Fixed fraction, $p = 1$

(b) MOESS, $p = 1$

(c) Fixed fraction, $p = 2$

(d) MOESS, $p = 2$

Figure 5-21: NACA 0006 gust encounter - entropy (t=7.5) (31k DOF)



(a) Fixed fraction, $p = 1$

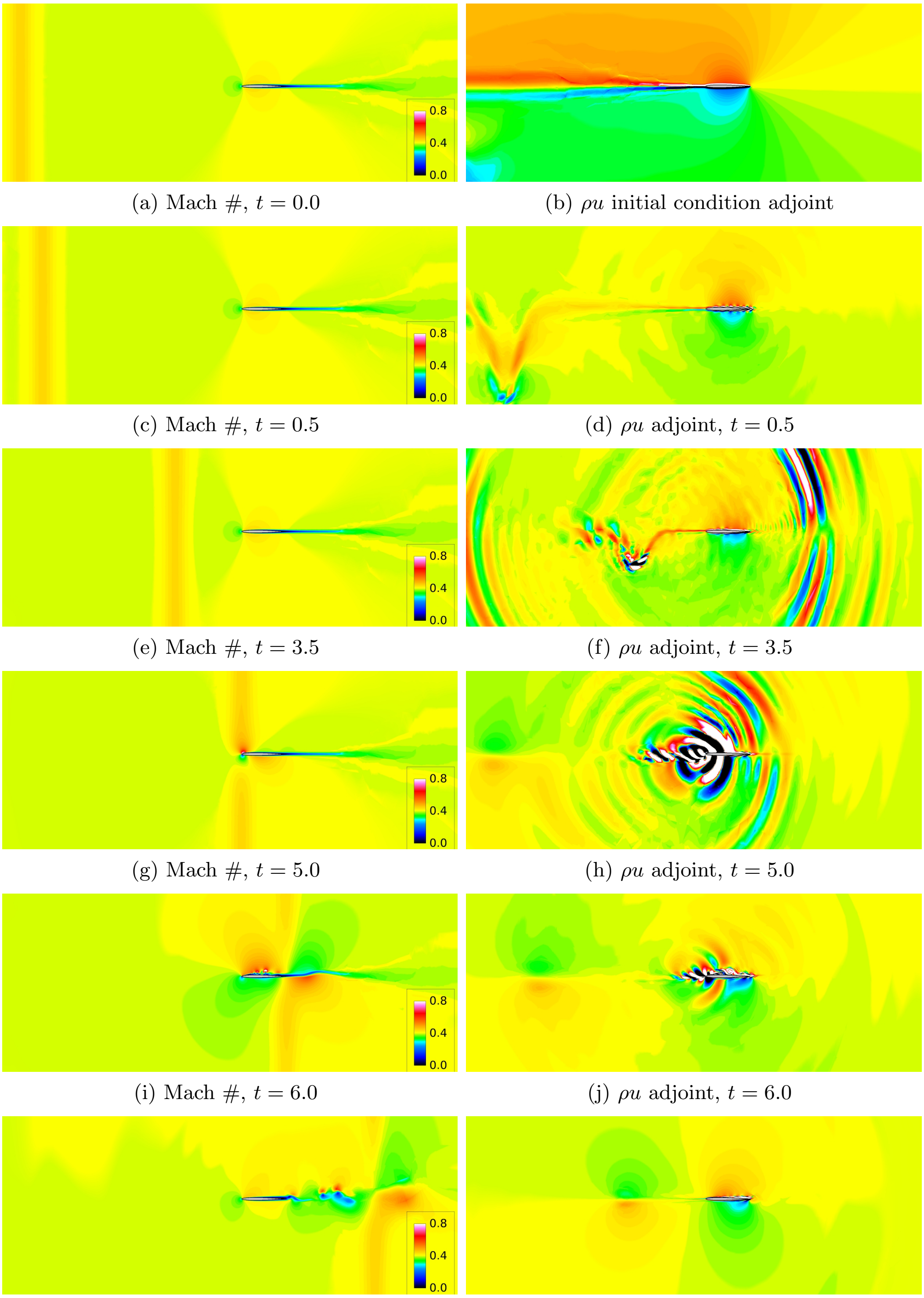
(b) Fixed fraction, $p = 2$

(c) MOESS, $p = 1$

(d) MOESS, $p = 2$

Figure 5-22: NACA 0006 gust encounter - $C_L$ history

the MOESS mesh, reflecting the increased availability of degrees of freedom. Figure (5-22) shows the lift coefficient time history at several DOF levels for both adaptation approaches and $p$ orders. After the initial transient, the lift coefficient quickly returns to the steady state value. Finally, snapshots of the Mach number and the adjoint of the x-momentum equation are shown in Figure (5-23). The most prominent features in the adjoint are the large magnitude region that tracks the gust and the acoustic-like wave excited by the passage of the gust past the airfoil. The initial condition adjoint, $\tilde{\boldsymbol{\psi}}$ is significantly different than $\boldsymbol{\psi}$, resembling the adjoint for the steady-state problem.



(a) Mach #, $t = 0.0$

(b) $\rho u$ initial condition adjoint

(c) Mach #, $t = 0.5$

(d) $\rho u$ adjoint, $t = 0.5$

(e) Mach #, $t = 3.5$

(f) $\rho u$ adjoint, $t = 3.5$

(g) Mach #, $t = 5.0$

(h) $\rho u$ adjoint, $t = 5.0$

(i) Mach #, $t = 6.0$

(j) $\rho u$ adjoint, $t = 6.0$

(k) Mach #, $t = 8.0$

(l) $\rho u$ adjoint, $t = 8.0$

Figure 5-23: NACA 0006 gust encounter - Mach # and x-momentum adjoint snapshots

# Chapter 6

# Conclusions

## 6.1 Summary & Contributions

This thesis examines three aspects of unsteady adjoint analysis in computational fluid dynamics. First, the need for unsteady analysis is motivated, even for problems that can be solved for a steady-state solution or that possess small-scale output unsteadiness. Next, the nonconvergence of finite-time averages to approximate periodic outputs and output sensitivities is investigated. The theoretical basis for this lack of convergence is identified and output windowing is proposed to alleviate it. Finally, unsteady simulation introduces additional complexity into error estimation and mesh adaptation. To address this, an unsteady error estimate and anisotropic, spatial adaptation method for unsteady wake problems is developed.

**Contribution 1:** *Demonstration that unconverged steady flow calculations, even with small output unsteadiness, can lead to significant variability in the estimated output sensitivity due to the arbitrary choice of unconverged state upon which the linearization is performed. Further, time-inaccurate "unsteady" iterative solutions depend on the iterative method used and may exhibit different output and output sensitivity compared to the steady flow or time-accurate unsteady flow.*

The effects of small-scale unsteadiness on adjoint-based output sensitivities were investigated through the use of a viscous, subsonic airfoil model problem. This investigation demonstrated the potential for large variability in the estimated output sensitivity using a steady sensitivity analysis when the nonlinear flow fails to converge to a steady state.

Results also showed that iterative methods commonly employed are sufficiently strong, i.e., insensitive to unstable modes, to converge to a stationary point solution, even when there is strong unsteady behavior in the time-accurate solution. This unstable stationary point flow solution satisfies the steady-state equations but is not physical and is of questionable design value. An implication of this study is the need to consider unsteady (time-accurate) flow and adjoint analysis even when steady-state solutions can be achieved using strong solvers. Finally, we show that the unsteady adjoint can provide accurate sensitivities for the time-averaged lift.

**Contribution 2:** *Demonstration that finite time averaging causes an error in output sensitivity. An expression for this error was derived showing that the error scales with the dependence of the period on a parameter and does not decay with increasing range of integration. Additionally, due this nonconvergent error, finite time simulation introduces noise into the design space which can trap optimization in local extrema.*

**Contribution 3:** *Development of two classes of windows to compute periodic outputs and output parameters using finite time simulations, analysis of their convergence properties, and demonstration of their application to output parameter sensitivity calculation.*

In this work, we investigated the convergence of the outputs and output parameter sensitivities of periodic problems using finite time simulation. We showed that outputs and sensitivities computed in this manner can take excessive time to converge, and that parameter sensitivities can fail altogether to converge to the periodic value. We developed a theoretical basis for this lack of convergence, showing that it results from the dependency of the period on the parameter of interest, and demonstrated it using ODE and PDE examples. To mitigate this lack of convergence, we investigated long-time and short-time windows applied to the instantaneous output: testing examples from both classes, demonstrating their respective convergence rates, and identifying their advantages and disadvantages. These windows enabled the accurate computation of periodic output sensitivities and decreased simulation time to compute time-averaged outputs.

Simulations using both long- and short-time windows require a relaxation time to equilibrate the solution and adjoint or tangent variables. For engineering problems, this lead-in time may be much longer than the required window span. Short-time windows demonstrate that the output of interest can be windowed over a very small number of periods, but additionally require knowledge of the period of oscillation. Additionally, with short-time windows the temporal discretization must accurately capture the shape of the window at any discontinuities, and failure to do so can introduce error into the output or sensitivity. Long-time windows still require a relaxation time to equilibrate adjoint or tangent states, but can compute accurate outputs and sensitivities without the need to compute the period of oscillation. In our experiments, long-time windows have been less sensitive to interactions between the temporal discretization and the window shape with only a small increase in the span of the window to achieve a given error tolerance.

The primary benefit of output windows is limited to non-chaotic/stable periodic problems. Other work[51] and preliminary testing with a chaotic Lorenz oscillator confirm this, with non-square windows performing no better than the square window for either output or parameter sensitivity calculations. This is due to the exponential growth of perturbations exhibited by chaotic systems. Despite this, recent work on sensitivities for chaotic problems indicates that output windowing yields improved results when combined with other techniques[119, 120].

**Contribution 4:** *Extension of the spatial mesh optimization framework of Yano & Darmofal to develop an anisotropic, output-based error estimation and spatial mesh adaptation approach for unsteady wake problems and other problems with smooth and persistent regions of unsteadiness.*

The unsteady spatial adaptation method developed in this work is an unsteady extension of the MOESS algorithm of Yano & Darmofal. As in the steady case, the output error due to spatial discretization is estimated and localized to each element. The MOESS algorithm is then applied to control the spatial discretization error on the output of the unsteady flow.

The algorithm was demonstrated in two dimensions on an inviscid convecting vortex and three laminar Navier-Stokes airfoil cases. The isentropic vortex demonstrated the ability of the unsteady error estimate to drive a decrease in spatial error while achieving expected convergence rates with respect to spatial degrees of freedom for both the fixed fraction isotropic and the optimization-based anisotropic MOESS approach. The meshes generated by MOESS are in general smoother than those produced by the fixed fraction algorithm, due to the continual mesh refinement and coarsening of the latter method.

The adaptation algorithm was then applied to a vortex-shedding laminar NACA 0012 airfoil to compute the periodic time-average of lift coefficient. The applicability of the single mesh approach to wake problems was demonstrated by the relative insensitivity of the adaptive performance to the windowing filter applied to the unsteady objective function. However, the resulting mesh exhibits low levels of anisotropy and little benefit over the isotropic fixed fraction approach. The NACA 0012 was then tested with an impulsive start initial condition and somewhat higher Reynolds number. The increased Reynolds anisotropy in the boundary layer leads to an order of magnitude improvement in relative error over the isotropic approach for both $p = 1$ and $p = 2$. Finally, for a NACA 0006 encounter with a vertical wind gust at $Re = 10000$, the MOESS approach results in as much as an order of magnitude decrease in error relative to the isotropic approach.

## 6.2 Recommendations and Future Work

### Unsteadiness detection

The work of Chapter 3 demonstrates the need for unsteady simulations and the danger of ignoring unsteady effects for outputs and especially for output sensitivities. Unfortunately, we do not propose any method of detection of potential unsteadiness in a steady simulation beyond

- intentionally using a linear solver that is weak enough to be sensitive to any unstable modes,
- computing the eigenvalues of a converged solution, or
- attempting to evolve the solution in a time-accurate fashion.

All of these options are potentially costly and none of them are conclusive as to the degree and impact of any underlying unsteady behavior. An efficient method of detecting physical unsteadiness from a steady solution, or from the process of computing a steady solution, and providing some measure of its impact on outputs of interest is of high value in CFD application. Such a method would provide assurance that a steady state solution is appropriate for a given geometry and flow conditions, and remove the potential cost of ‘experimental‘ unsteady simulations.

### Time-averaged outputs and sensitivities for chaotic problems

While output windows can provide an effective means of speeding convergence of periodic outputs and output sensitivities, windowing alone is insufficient when applied to chaotic problems. For such problems any disturbance grows exponentially, causing any method that depends on a linearization of the state equations, i.e., tangent or adjoint methods, to diverge exponentially as well. Recent work by Wang[119, 120] is able to limit this exponential growth and is a promising avenue toward computing output sensitivities for chaotic problems at a reasonable cost.

### Temporal adaptation & timeslabs

While the current implementation is restricted to single-mesh spatial adaptation and a prescribed timestep, the algorithm is agnostic to the temporal discretization. This allows a straightforward application of much of the technology developed for timeslab approaches. For example, coupling the MOESS approach with a DG temporal discretization enables a relatively straightforward extension to temporal error estimation and timestep control by including error sampling in the temporal dimension. The flexibility of the sizing control of MOESS will allow it to be applied to individual timeslabs, affording additional control over the spatial mesh. Inclusion of temporal adaptation and temporally varying spatial refinement will enable efficient application of this method to more general unsteady problems. However, additional algorithmic development work is necessary to balance error and cost across multiple timeslabs. If a point-based timestepping scheme is to be retained, temporal adaptation may still be developed by incorporating methods similar to Mani & Mavriplis[63]. With the spatial and temporal error estimates computed independently in this fashion, a suitable means of balancing spatial and temporal error must be developed.

The MOESS algorithm has been successfully demonstrated in three dimensions to steady compressible flow problems[125]. While the unsteady extension in this thesis is fully compatible with three spatial dimensions with no further modification, availability of computational resources defer demonstration to future work.

# Bibliography

[1] "3rd AIAA CFD Drag Prediction Workshop." San Francisco, CA. http://aaac.larc.nasa.gov/tsab/cfdlarc/aiaa-dpw/Workshop3.

[2] Alauzet, F., Frey, P. J., George, P. L., and Mohammadi, B. "3D transient fixed point mesh adaptation for time-dependent problems: Application to CFD simulations." *Journal of Computational Physics*, 222(2):592–623, 2007.

[3] Alauzet, F., Belme, A., and Dervieux, A. "Anisotropic Goal-Oriented Mesh Adaptation for Time Dependent Problems." In Quadros, W. R., editor, *Proceedings of the 20th International Meshing Roundtable*, pages 99–121. Springer Berlin Heidelberg, 2012.

[4] Bar-Yoseph, P. "Space-time discontinuous finite element approximations for multidimensional nonlinear hyperbolic systems." *Computational Mechanics*, 5:145–160, 1989.

[5] Barter, G. E. *Shock Capturing with PDE-Based Artificial Viscosity for an Adaptive, Higher-Order, Discontinuous Galerkin Finite Element Method.* PhD thesis, Massachusetts Institute of Technology, Department of Aeronautics and Astronautics, June 2008.

[6] Barter, G. E. and Darmofal, D. L. "Shock capturing with PDE-based artificial viscosity for DGFEM: Part I, Formulation." *Journal of Computational Physics*, 229(5): 1810–1827, 2010.

[7] Barth, T. "Space-Time Error Representation and Estimation in Navier-Stokes Calculations." In Barth, T. J., Griebel, M., Keyes, D. E., Nieminen, R. M., Roose, D., Schlick, T., Kassinos, S. C., Langer, C. A., Iaccarino, G., and Moin, P., editors,

*Complex Effects in Large Eddy Simulations*, volume 56 of *Lecture Notes in Computational Science and Engineering*, pages 29–48. Springer Berlin Heidelberg, 2007. ISBN 978-3-540-34234-2.

[8] Barth, T. J. and Larson, M. G. "A posteriori error estimates for higher order Godunov finite volume methods on unstructured meshes." In Herban, R. and Kröner, D., editors, *Finite Volumes for Complex Applications III*, London, 2002. Hermes Penton.

[9] Bassi, F. and Rebay, S. "High-order accurate discontinuous finite element solution of the 2D Euler equations." *Journal of Computational Physics*, 138(2):251–285, 1997.

[10] Bassi, F. and Rebay, S. "A high-order discontinuous finite element method for the numerical solution of the compressible Navier-Stokes equations." *Journal of Computational Physics*, 131:267–279, 1997.

[11] Bassi, F. and Rebay, S. "GMRES discontinuous Galerkin solution of the compressible Navier-Stokes equations." In Cockburn, K. and Shu, editors, *Discontinuous Galerkin Methods: Theory, Computation and Applications*, pages 197–208. Springer, Berlin, 2000.

[12] Bassi, F. and Rebay, S. "Numerical evaluation of two discontinuous Galerkin methods for the compressible Navier-Stokes equations." *International Journal for Numerical Methods in Fluids*, 40:197–207, 2002.

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

[14] Borouchaki, H., George, P. L., Hecht, F., Laug, P., and Saltel, E. "Delaunay mesh generation governed by metric specifications. Part I algorithms." *Finite Elements in Analysis and Design*, 25(1-2):61–83, 1997.

[15] Boyd, J. P. *Chebyshev and Fourier Spectral Methods.* Dover, Mineola, NY, 2001. http://www-personal.engin.umich.edu/jpboyd/.

[16] Bryson, A. E. and Ho, Y. C. *Applied Optimal Control.* Blaisdell, London, 1969.

[17] Campobasso, M. S. and Giles, M. B. "Effects of Flow Instabilities on the Linear Analysis of Turbomachinery Aeroelasticity." *AIAA Journal of Propulsion and Power*, 19(2):250–259, 2003.

[18] Campobasso, M. S. and Giles, M. B. "Stabilization of a Linear Flow Solver for Turbomachinery Aeroelasticity Using Recursive Projection Method." *AIAA Journal*, 42(9):323–332, 2004.

[19] Cory, R. E. *Supermaneuverable Perching.* PhD thesis, Massachusetts Institute of Technology, 2010.

[20] Derber, J. C. "A Variational Continuous Assimilation Technique." *Monthly Weather Review*, 117(11):2436–2446, 1989.

[21] Diosady, L. T. "A linear multigrid preconditioner for the solution of the Navier-Stokes equations using a discontinuous Galerkin discretization." Master's thesis, Mass. Inst. of Tech., CDO, May 2007.

[22] Diosady, L. T. and Darmofal, D. L. "Preconditioning methods for discontinuous Galerkin solutions of the Navier-Stokes equations." *Journal of Computational Physics*, 228:3917–3935, 2009.

[23] Duta, M. C., Giles, M., and Campobasso, M. S. "The Harmonic Adjoint Approach to Unsteady Turbomachinery Design." *International Journal for Numerical Methods in Fluids*, 40(3–4):323–332, 2002.

[24] Errico, R. M. and Vukicevic, T. "Sensitivity Analysis Using an Adjoint of the PSU-NCAR Mesoscale Model." *Monthly Weather Review*, 120(8):1644–1660, 1991.

[25] Fidkowski, K. and Darmofal, D. "Output error estimation and adaptation in computational fluid dynamics: Overview and recent results." AIAA 2009-1303, 2009.

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

[27] Fidkowski, K. J. *A Simplex Cut-Cell Adaptive Method for High-Order Discretizations of the Compressible Navier-Stokes Equations.* PhD thesis, Massachusetts Institute of Technology, Department of Aeronautics and Astronautics, June 2007.

[28] Fidkowski, K. J. and Darmofal, D. L. "An adaptive simplex cut-cell method for discontinuous Galerkin discretizations of the Navier-Stokes equations." AIAA 2007-3941, 2007.

[29] Fidkowski, K. J. and Darmofal, D. L. "A triangular cut-cell adaptive method for higher-order discretizations of the compressible Navier-Stokes equations." *Journal of Computational Physics*, 225:1653–1672, 2007.

[30] Fidkowski, K. J. and Luo, Y. "Output-based spacetime mesh adaptation for the compressible NavierStokes equations." *Journal of Computational Physics*, 230(14): 5753 – 5773, 2011.

[31] Fidkowski, K. J., Oliver, T. A., Lu, J., and Darmofal, D. L. "*p*-Multigrid solution of high-order discontiunous Galerkin discretizations of the compressible Navier-Stokes equations." *Journal of Computational Physics*, 207(1):92–113, 2005.

[32] Flynt, B. T. and Mavriplis, D. J. "Discrete adjoint based adaptive error control in unsteady flow problems." AIAA 2012-0078, January 2012.

[33] Formaggia, L., Micheletti, S., and Perotto, S. "Anisotropic mesh adaptation with applications to CFD problems." In Mang, H. A., Rammerstorfer, F. G., and Eberhardsteiner, J., editors, *Fifth World Congress on Computational Mechanics*, Vienna, Austria, July 7-12 2002.

[34] Formaggia, L., Perotto, S., and Zunino, P. "An anisotropic a-posteriori error estimate for a convection-diffusion problem." *Computing and Visualization in Science*, 4:99–104, 2001.

[35] Georgoulis, E. H., Hall, E., and Houston, P. "Discontinuous Galerkin methods on *hp*-anisotropic meshes II: A posteriori error analysis and adaptivity." *Applied Numerical Mathematics*, 59:2179–2194, 2009.

[36] Giles, M. "On the use of Runge-Kutta time-marching and multigrid for the solution of steady adjoint equations." Report NA00/10, Oxford University Computing Laboratory, 2000.

[37] Giles, M. B. and Süli, E. "Adjoint methods for PDEs: a posteriori error analysis and postprocessing by duality." In *Acta Numerica*, volume 11, pages 145–236, 2002.

[38] Giles, M. B. "On the Iterative Solution of Adjoint Equations." In Corliss, G., Faure, C., Griewank, A., Hascoet, L., and Naumann, U., editors, *Automatic Differentiation: From Simulation to Optimization*. Springer, 2001.

[39] Giles, M. B. and Pierce, N. A. "An Introduction to the Adjoint Approach to Design." *Flow, Turbulence and Combustion*, 65:393–415, 2000.

[40] Griewank, A. and Walther, A. "Revolve: An Implementation of Checkpointing for the Reverse or Adjoint Mode of Computational Differentiation." *ACM Transactions on Mathematical Software*, 26(1):19–45, 2000.

[41] Hall, M. C. G. "Application of Adjoint Sensitivity to an Atmospheric General Circulation Model." *Journal of the Atmospheric Sciences*, 43(22):2644–2652, 1986.

[42] Hartmann, R. "Adaptive discontinuous Galerkin methods with shock-capturing for the compressible Navier-Stokes equations." *International Journal for Numerical Methods in Fluids*, 51:1131–1156, 2006.

[43] Hartmann, R. and Houston, P. "Adaptive discontinuous Galerkin finite element methods for the compressible Euler equations." *Journal of Computational Physics*, 183(2): 508–532, 2002.

[44] Hartmann, R. and Houston, P. "Error estimation and adaptive mesh refinement for aerodynamic flows." In Deconinck, H., editor, *VKI LS 2010-01: 36$^{th}$ CFD/ADIGMA course on hp-adaptive and hp-multigrid methods, Oct. 26-30, 2009*. Von Karman Institute for Fluid Dynamics, Rhode Saint Genèse, Belgium, 2009.

[45] He, J.-W., Glowinski, R., Metcalfe, R., and Periaux, J. "Active Control and Drag Optimization for Flow Past a Circular Cylinder." *Journal of Computational Physics*, 163:83–117, 2000.

[46] Hecht, F. "BAMG: Bidimensional Anisotropic Mesh Generator." 1998. `http://www-rocq1.inria.fr/gamma/cdrom/www/bamg/eng.htm`.

[47] Houston, P., Georgoulis, E. H., and Hall, E. "Adaptivity and a posteriori error estimation for DG methods on anisotropic meshes." *International Conference on Boundary and Interior Layers*, 2006.

[48] Jameson, A. “Aerodynamic design via control theory.” *Journal of Science and Computation*, 3:233–260, 1988.

[49] Jothiprasad, G., Mavriplis, D. J., and Caughey, D. A. “Higher-order time integration schemes for the unsteady Navier-Stokes equations on unstructured meshes.” Report, 2003.

[50] Klaij, C. M., van der Vegt, J. J. W., and van der Ven, H. “Space-Time Discontinuous Galerkin Method for the Compressible Navier-Stokes Equations.” *Journal of Computational Physics*, 217:589–611, 2006.

[51] Lea, D. J., Allen, M. R., and Haine, T. W. N. “Sensitivity analysis of the climate of a chaotic system.” *Tellus A*, 52(5):523–532, 2000. ISSN 1600-0870.

[52] Leicht, T. and Hartmann, R. “Anisotropic mesh refinement for discontinuous Galerkin methods in two-dimensional aerodynamic flow simulations.” *International Journal for Numerical Methods in Fluids*, 56:2111–2138, 2008.

[53] Leicht, T. and Hartmann, R. “Error estimation and anisotropic mesh refinement for 3d laminar aerodynamic flow simulations.” *Journal of Computational Physics*, 229: 7344–7360, 2010.

[54] Lockard, D. P. “An efficient, two-dimensional implementation of the Ffowcs Williams and Hawkings equation.” *Journal of Sound and Vibration*, 229(4):897–911, 1999.

[55] Loseille, A. and Alauzet, F. “Continuous mesh model and well-posed continuous interpolation error estimation.” INRIA RR-6846, 2009.

[56] Loseille, A. and Alauzet, F. “Optimal 3D highly anisotropic mesh adaptation based on the continuous mesh framework.” In *Proceedings of the 18th International Meshing Roundtable*, pages 575–594. Springer Berlin Heidelberg, 2009.

[57] Loseille, A., Dervieux, A., and Alauzet, F. “On 3-d goal-oriented anisotropic mesh adaptation applied to inviscid flows in aeronautics.” AIAA 2010-1067, 2010.

[58] Lowrie, R. B., Roe, P. L., and v.Leer, B. “Properties of space-time discontinuous Galerkin.” LAUR LA-UR–96-3705, 1998.

[59] Lu, J. *An a Posteriori Error Control Framework for Adaptive Precision Optimization Using Discontinuous Galerkin Finite Element Method.* PhD thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts, 2005.

[60] Machiels, L., Peraire, J., and Patera, A. "A Posteriori Finite-Element Output Bounds for the Incompressible Navier-Stokes Equations: Application to a Natural Convection Problem." *Journal of Computational Physics*, 172:401–425, 2001.

[61] Mani, K. *Application of the Discrete Adjoint Method to Coupled Multidisciplinary Unsteady Flow Problems for Error Estimation and Optimization.* PhD dissertation, University of Wyoming, Department of Mechanical Engineering, March 2009.

[62] Mani, K. and Mavriplis, D. J. "An Unsteady Discrete Adjoint Formulation for Two-Dimensional Flow Problems with Deforming Meshes." AIAA 2007-60, 2007.

[63] Mani, K. and Mavriplis, D. J. "Efficient Solutions of the Euler Equations in a Time-Adaptive Space-Time Framework." AIAA 2011-774, 2011.

[64] Mavriplis, D. J. "Results from the 3rd Drag Prediction Workshop using the NSU3D unstructured mesh solver." AIAA 2007-256, 2007.

[65] Mavriplis, D. J. "Solution of the unsteady discrete adjoint for three-dimensional problems on dynamically deforming unstructured meshes." AIAA 2008-727, 2008.

[66] Meidner, D. and Vexler, B. "Adaptive space-time finite element methods for parabolic optimization problems." *SIAM Journal on Control and Optimization*, 46(1):116–142, 2007.

[67] Meidner, D. and Vexler, B. "Adaptive space-time finite element methods for parabolic optimization problems." *SIAM Journal on Control and Optimization*, 46(1):116–142, 2008.

[68] Michal, T. and Krakos, J. "Anisotropic mesh adaptation through edge primitive operations." AIAA 2012-159, 2012.

[69] Modisette, J. M. *An Automated Reliable Method for Two-Dimensional Reynolds-averaged Navier-Stokes Simulations.* PhD thesis, Massachusetts Institute of Technology, Department of Aeronautics and Astronautics, September 2011.

[70] Mont, A. D. “Adaptive unstructured spacetime meshing for four-dimensional spacetime discontinuous Galerkin finite element methods.” Master’s thesis, University of Illinois at Urbana-Champaign, Department of Computer Science, December 2011.

[71] Nadarajah, S. K. and Jameson, A. “Optimal Control of Unsteady Flows Using a Time Accurate Method.” AIAA 2002-5436, 2002.

[72] Nadarajah, S. K. and Jameson, A. “Optimum Shape Design for Unsteady Flows With Time-Accurate Continuous and Discrete Adjoint Methods.” *AIAA Journal*, 45 (7):1478–1491, 2007.

[73] Nadarajah, S. K. and Jameson, A. “Optimum Shape Design for Unsteady Three-Dimensional Viscous Flows Using a Nonlinear Frequency-Domain Method.” *AIAA Journal of Aircraft*, 44(5):1513–1527, 2007.

[74] Nadarajah, S. K., McMullen, M. S., and Jameson, A. “Optimum Shape Design for Unsteady Flows Using Time Accurate and Non-Linear Frequency Methods.” AIAA 2003-3875, 2003.

[75] Nemec, M., Aftosmis, M., Murman, S., and Pulliam, T. “Adjoint formulation for an embedded-boundary Cartesian method.” AIAA 2005-0877, 2005.

[76] Nemec, M., Aftosmis, M. J., and Wintzer, M. “Adjoint-based adaptive mesh refinement for complex geometries.” AIAA 2008-725, 2008.

[77] Nielsen, E. J. and Anderson, W. K. “Recent improvements in aerodynamic design optimization on unstructured meshes.” *AIAA Journal*, 40(6):1155–1163, 2002. doi: 10.2514/2.1765.

[78] Nielsen, E. J., Diskin, B., and Yamaleev, N. K. “Discrete Adjoint-Based Design Optimization of Unsteady Turbulent Flows on Dynamic Unstructured Grids.” *AIAA Journal*, 48(6):1195–1206, Jun 2010.

[79] Nielsen, E. J., Lu, J., Park, M. A., and Darmofal, D. L. “An implicit, exact dual adjoint solution method for turbulent flows on unstructured grids.” *Computers & Fluids*, 33:1131–1155, 2004.

[80] Oliver, T. and Darmofal, D. "An unsteady adaptation algorithm for discontinuous Galerkin discretizations of the RANS equations." AIAA 2007-3940, 2007.

[81] Oliver, T. A. "Multigrid solution for high-order discontinuous Galerkin discretizations of the compressible Navier-Stokes equations." Master's thesis, Massachusetts Institute of Technology, Department of Aeronautics and Astronautics, June 2004.

[82] Oliver, T. A. *A Higher-Order, Adaptive, Discontinuous Galerkin Finite Element Method for the Reynolds-averaged Navier-Stokes Equations.* PhD thesis, Massachusetts Institute of Technology, Department of Aeronautics and Astronautics, June 2008.

[83] Park, M. A. *Anisotropic Output-Based Adaptation with Tetrahedral Cut Cells for Compressible Flows.* PhD thesis, Massachusetts Institute of Technology, Department of Aeronautics and Astronautics, 2008.

[84] Park, M. A., Lee-Rausch, E. M., and Rumsey, C. L. "FUN3D and CFL3D Computations for the First High Lift Prediction Workshop." AIAA 2011-936, 2011.

[85] Pennec, X., Fillard, P., and Ayache, N. "A Riemannian framework for tensor computing." *International Journal of Computer Vision*, 66(1):41–66, 2006.

[86] Persson, P.-O. "Scalable Parallel Newton-Krylov Solvers for Discontinuous Galerkin discretizations." AIAA 2009-606, 2009.

[87] Pervaiz, M. M. and Baron, J. R. "Spatiotemporal Adaptation Algorithm for Two-Dimensional Reacting Flows." *AIAA Journal*, 27(10):1368–1376, 1989.

[88] Pierce, N. A. and Giles, M. B. "Adjoint recovery of superconvergent functionals from PDE approximations." *SIAM Review*, 42(2):247–264, 2000.

[89] Rannacher, R. "Adaptive Galerkin finite element methods for partial differential equations." *Journal of Computational and Applied Mathematics*, 128:205–233, 2001.

[90] Rattenbury, N. *Almost Runge-Kutta Methods for Stiff and Non-Stiff Problems.* PhD thesis, University of Auckland, 2005.

[91] Rausch, R. D., Batina, J. T., and Yang, H. T. Y. "Spatial Adaptation of Unstructured Meshes for Unsteady Aerodynamic Flow Computations." *AIAA Journal*, 30(5):1243–1251, 1992.

[92] Rendall, T. C. S. and Allen, C. B. "Conservative Unsteady Simulation of Arbitrary Boundary Deformation using Spacetime Meshes." AIAA 2010-509, 2010.

[93] Richter, T. "Discontinuous Galerkin as Time-Stepping Scheme for the Navier-Stokes Equations." In proceedings of the 4th International Conference on High Performance Scientific Computing, Hanoi (submitted), 2009.

[94] Richter, T. "A posteriori error estimation and anisotropy detection with the dual-weighted residual method." *International Journal for Numerical Methods in Fluids*, 62(1):90–118, 2010. ISSN 1097-0363.

[95] Richter, T. "*A posteriori* error estimation and anisotropy detection with the dual-weighted residual method." *International Journal for Numerical Methods in Fluids*, 62:90–118, 2010.

[96] Roe, P. L. "Approximate Riemann solvers, parameter vectors, and difference schemes." *Journal of Computational Physics*, 43(2):357–372, 1981.

[97] Rumpfkeil, M. P. and Zingg, D. W. "Unsteady Optimization Using a Discrete Adjoint Approach Applied to Aeroacoustic Shape Design." AIAA 2008-18, 2008.

[98] Rumpfkeil, M. P. and Zingg, D. W. "A hybrid algorithm for far-field noise minimization." *Computers & Fluids*, 39:1516–1528, 2010.

[99] Rumpfkeil, M. P. and Zingg, D. W. "The optimal control of unsteady flows with a discrete adjoint method." *Optimization and Engineering*, 11:5–22, 2010. ISSN 1389-4420.

[100] Saad, Y. and Schultz, M. H. "GMRES: A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems." *SIAM Journal on Scientific and Statistical Computing*, 7(3):856–869, 1986.

[101] Sanders, L., Manoha, E., Khelil, B., and Franąois, C. "LAGOON : CFD/CAA Coupling

for Landing Gear Noise and Comparison with Experimental Database." AIAA 2011-2822, 2011.

[102] Schmich, M. and Vexler, B. "Adaptivity with Dynamic Meshes for Space-Time Finite Element Discretizations of Parabolic Equations." *SIAM Journal for Scientific Computing*, 30(1):369–393, 2008.

[103] Shakib, F. and Hughes, T. J. R. "A new finite element formulation for computational fluid dynamics: IX. Fourier analysis of space-time Galerkin/least-squares algorithms." *Computer Methods in Applied Mechanics and Engineering*, 87(1):35–58, 1991.

[104] Shroff, G. M. and Keller, H. B. "Stabilization of Unstable Procedures: The Recursive Projection Method." *SIAM Journal on Numerical Analysis*, 30(4):1099–1120, 1993.

[105] Sinha, K. and Candle, G. V. "Grid Sensitivity of Detached Eddy Simulation of a Mach 16 Re-entry Configuration." AIAA 2007-1115, 2007.

[106] Solín, P. and Demkowicz, L. "Goal-oriented hp-adaptivity for elliptic problems." *Computer Methods in Applied Mechanics and Engineering*, 193:449–468, 2004.

[107] Speares, W. and Berzins, M. "A 3D Unstructured Mesh Adaptation Algorithm for Time-Dependent Shock-Dominated Problems." *International Journal for Numerical Methods in Fluids*, 25(1):81–104, 1997.

[108] Srinath, D. N. and Mittal, S. "An Adjoint Method for Shape Optimization in Unsteady Viscous Flows." *Journal of Computational Physics*, 229:1994–2008, 2010.

[109] Sun, H. "Impact of triangle shapes using high-order discretizations and direct mesh adaptation for output error." Master's thesis, Massachusetts Institute of Technology, Computation for Design and Optimization Program, 2009.

[110] Talagrand, O. and Courtier, P. "Variational Assimilation of Meteorological Observations with the Adjoint Vorticity Equation. I: Theory." *Quarterly Journal of the Royal Meteorological Society*, 113:1311–1328, 1987.

[111] Thomas, J. P., Hall, K. C., and Dowell, E. H. "Discrete Adjoint Approach for Modeling Unsteady Aerodynamic Design Sensitivities." *AIAA Journal*, 43(9):1931–1936, 2005.

[112] Tomovic, R. and Vulkobratovic, M. *General Sensitivity Theory.* American Elsevier, New York, NY, 1972.

[113] van der Vegt, J. J. W. and van der Ven, H. "Space-time discontinuous Galerkin finite element method with dynamic grid motion for inviscid compressible flows." *Journal of Computational Physics*, 182:546–585, 2002.

[114] Vassberg, J. C., DeHaan, M. A., and Sclafani, T. J. "Grid generation requirements for accurate drag predictions based on OVERFLOW calculations." AIAA 2003-4124, 2003.

[115] Vassberg, J. C., Tinoco, E. N., Mani, M., Brodersen, O. P., Eisfeld, B., Wahls, R. A., Morrison, J. H., Zickuhr, T., k.R. Laflin, and Mavriplis, D. J. "Summary of the Third AIAA CFD Drag Prediction Workshop." AIAA 2007-260, 2007.

[116] Venditti, D. A. and Darmofal, D. L. "Grid adaptation for functional outputs: application to two-dimensional inviscid flows." *Journal of Computational Physics*, 176(1): 40–69, 2002.

[117] Venditti, D. A. and Darmofal, D. L. "Anisotropic grid adaptation for functional outputs: Application to two-dimensional viscous flows." *Journal of Computational Physics*, 187(1):22–46, 2003.

[118] Wang, L. and Mavriplis, D. J. "Implicit solution of the unsteady Euler equations for high-order accurate discontinuous Galerkin discretizations." AIAA 2006-0109, 2006.

[119] Wang, Q. "Forward and Adjoint Sensitivity Computation of Chaotic Dynamical Systems." *ArXiv e-prints*, February 2012.

[120] Wang, Q. and Hu, R. "Sensitivity computation of periodic and chaotic limit cycle oscillations." *ArXiv e-prints*, April 2012.

[121] Wang, Q., Moin, P., and Iaccarino, G. "Minimal Repetition Dynamic Checkpointing Algorithm for Unsteady Adjoint Calculation." *SIAM Journal on Scientific Computing*, 31(4):2549–2567, 2009.

[122] Whitehouse, G. R. and Tadghighi, H. "Investigation of Hybrid Grid-Based CFD Methods for Rotorcraft Flow Analysis." American helicopter society, 2010.

[123] Wilkins, A. K., Tidor, B., White, J., and Barton, P. I. "Sensitivity Analysis for Oscillating Dynamical Systems." *SIAM Journal on Scientific Computing*, 31(4):2706–2732, 2009.

[124] Yang, Z. and Mavriplis, D. J. "Development of An Adaptive Space-Time Method for High-Order Resolution of Discontinuities." AIAA 2008-758, 2008.

[125] Yano, M. *An Optimization Framework for Adaptive Higher-Order Discretizations of Partial Differential Equations on Anisotropic Simplex Meshes.* PhD thesis, Massachusetts Institute of Technology, Department of Aeronautics and Astronautics, June 2012.

[126] Yano, M. and Darmofal, D. "An Optimization Framework for Anisotopic Simplex Mesh Adaptation." *Journal of Computational Physics*, (submitted), 2011.

[127] Yano, M. and Darmofal, D. "An optimization framework for anisotropic simplex mesh adaptation: application to aerodynamic flows." AIAA 2012–0079, January 2012.

[128] Yano, M., Modisette, J. M., and Darmofal, D. "The importance of mesh adaptation for higher-order discretizations of aerodynamic flows." AIAA 2011–3852, June 2011.

# Appendix A

# Adjoint Temporal Discretization

Because the time integration is discretized to evolve the problem forward in time, care must be taken to discretize the unsteady adjoint in a manner that maintains consistency with the forward problem. For a steady adjoint, this consistent adjoint discretization can be derived by applying the exact dual of Giles[38], and later investigated by Nielsen, *et al.*[79]. For the discretization of the unsteady adjoint, the adjoint is taken of the temporally discrete unsteady primal. This gives an adjoint that is consistent with the discrete temporal forward problem, including with forward (tangent) sensitivities.

For a general unsteady scheme, including multistep backward difference schemes and multistage diagonally implicit Runge-Kutta (DIRK) schemes with $S_i$ stages at each timestep $i$, the unsteady residual statement at a timestep, Eq. (2.6), can be expressed as: *For s=1,...,*$S_i$

$$0 = \mathcal{M}\left(a_{i,0}\boldsymbol{u}_i^{(s)} + \sum_{j=1}^{P_i} a_{i,j}\boldsymbol{u}_{i-j}\right) + d_i^s \boldsymbol{f}(\boldsymbol{u}_{i-1};\beta) + \sum_{j=1}^{s} b_i^{sj}\boldsymbol{f}(\boldsymbol{u}_i^{(j)};\beta)$$

with $\boldsymbol{u}_i \equiv \boldsymbol{u}_i^{(S_i)}$ and the initial condition satisfying

$$\boldsymbol{a}(\boldsymbol{u}_0;\beta) = 0.$$

At timestep $i$, $a_{i,j}$ is the coefficient of the state at the $j^{th}$ previous step, where $j = 0 \ldots P_i$. The coefficients $a_{i,j}$ include the timestep, $\Delta t$, or combinations of timesteps at multiple steps as appropriate. $b_i^{jk}$ is the Butcher coefficient at stage $j$ weighting previous stage residual

$\boldsymbol{f}(\boldsymbol{u}_i^{(k)}, t; \beta)$. Coefficients for several common schemes are given in Appendix B. The mass matrix, $\mathcal{M}$, is included here to maintain consistency with a spatial discretization of $\boldsymbol{f}$.

For temporal discretizations with an explicit first stage, such as the fourth order ESDIRK scheme investigated by[118], the first explicit stage simply assigns the previous time step state to the first stage of the new step. If an adjoint state is maintained for each stage, including the explicit stage, this adds redundant storage and an additional "placeholder" adjoint that is proportional to the timestep. The Butcher coefficients related to the explicit first stage are used to multiply the spatial residual of the previous time step, $\boldsymbol{f}(\boldsymbol{u}_{i-1}; \beta)$, eliminating the unnecessary state and adjoint vectors. Denoting the original Butcher coefficients by $\hat{b}^{ij}$ with $\hat{S} = S + 1$ stages, for all $\hat{b}^{1j} = 0$ the modified implementation is:

$$\begin{bmatrix} \begin{bmatrix} 0 \end{bmatrix} & \begin{bmatrix} 0 & \cdots & 0 \end{bmatrix} \\ \begin{bmatrix} d^1 \\ \vdots \\ d^S \end{bmatrix} & \begin{bmatrix} b^{11} & \cdots & b^{1S} \\ \vdots & \ddots & \vdots \\ b^{S1} & \cdots & b^{SS} \end{bmatrix} \end{bmatrix} = \begin{bmatrix} \hat{b}^{11} & \hat{b}^{12} & \cdots & \hat{b}^{1\hat{S}} \\ \hat{b}^{21} & \hat{b}^{22} & \cdots & \hat{b}^{2\hat{S}} \\ \vdots & \vdots & \ddots & \vdots \\ \hat{b}^{\hat{S}1} & \hat{b}^{\hat{S}2} & \cdots & \hat{b}^{\hat{S}\hat{S}} \end{bmatrix}$$

For an unsteady output, $\mathcal{J}^E$, given by Eq. (2.10) and including a terminal output and time integrated output, the integration of the output can be carried out by differentiating the integral expression,

$$\frac{d\mathcal{J}^E}{dt} = g(\boldsymbol{u}; \beta).$$

which can then be integrated forward in time using the same time integration scheme as the primal problem:

$$\begin{aligned} \sum_{j=0}^{P_i} a_{i,j} J_{i-j} &= d_i^s g(\boldsymbol{u}_{i-1}; \beta) + \sum_{s=1}^{S_i} b_i^{S_i s} g\left(\boldsymbol{u}_i^{(s)}; \beta\right), \qquad \text{for } i = 1, N \\ J_0 &= h(\boldsymbol{u}_N; \beta) \end{aligned}$$

where the approximate unsteady output is given by $\mathcal{J}^E \approx J_N$. Using the same integration for the output as for the primal problem ensures that the same order of accuracy is maintained in the output integration. Only the last stage is necessary, as the left-hand side is an output of the right-hand side and does not feed back into it, hence there is no dependence of the right-hand side on the intermediate stages. To remove the summation from the left-hand side, call the right-hand side $G_i$ at timestep $i = [1, N]$. The problem can be written as:

$$AJ = G$$

where $A_{i,i-j} = a_{i,j}$. Let $C = A^{-1}$, and $\boldsymbol{c}$ the last row of $C$, giving

$$J^E = J_N = \boldsymbol{c}G + J_0$$

For the Lagrangian in Eq. (2.11) we can apply the same procedure:

$$\begin{aligned}\mathcal{L}^E &\equiv h(\boldsymbol{u}(t_f);\beta) - \tilde{\boldsymbol{\psi}}^T \boldsymbol{a}(\boldsymbol{u}(0);\beta) \\ &\quad + \int_0^{t_f} \left\{g(\boldsymbol{u}(t);\beta) - \boldsymbol{\psi}(t)^T \boldsymbol{f}^e(\boldsymbol{u}(t), t;\beta)\right\} dt \\ \frac{d\mathcal{L}^E}{dt} &= g(\boldsymbol{u}(t);\beta) - \boldsymbol{\psi}(t)^T \boldsymbol{f}^e(\boldsymbol{u}(t), t;\beta).\end{aligned}$$

The Lagrangian doesn't appear on the right-hand side so, as with the output, only the final stage of each step is necessary. Define $L_i$ for $i \in [0, N]$ as the incremental contribution to the $L^E$, i.e.,

$$\begin{aligned}\frac{1}{c_i} L_i &= \frac{\delta_{iN}}{c_i} h(\boldsymbol{u}_N;\beta) + d_i^s g(\boldsymbol{u}_{i-1};\beta) + \sum_{s=1}^{S_i} \left\{ b_i^{S_i s} g\left(\boldsymbol{u}_i^{(s)};\beta\right) - \left(\boldsymbol{\psi}_i^{(s)}\right)^T \boldsymbol{f}_{\boldsymbol{e}}(\boldsymbol{u}_i^{(s)}, i;\beta) \right\} \\ L_0 &= -\tilde{\boldsymbol{\psi}}^T \boldsymbol{a}(\boldsymbol{u}_0;\beta) \\ L^E &= \sum_{i=0}^{N} L_i.\end{aligned}$$

Note that the $b_i^{S_i s}$ factor is only applied to the output portion of the right-hand side and not including the adjoined portion. This is done because in the event of $b_i^{S_i s} = 0$ for some $s$, the $\boldsymbol{\psi}_i^{(s)}$ becomes unavailable to remove all the terms of $\frac{\partial \mathcal{L}^E}{\partial \boldsymbol{u}_i^{(s)}}$ and stationarity wrt $\boldsymbol{u}_i^{(s)}$ cannot be maintained.

To derive the discrete adjoint equations, first differentiate each component, $L_i$, of the Lagrangian with respect to:

1. the final timestep state, $\boldsymbol{u}_i$:

$$\frac{dL_i}{d\boldsymbol{u}_i} = \delta_{iN} h_{,\boldsymbol{u}}(\boldsymbol{u}_N;\beta) + c_i \left[ b_i^{S_iS_i} g_{,\boldsymbol{u}}(\boldsymbol{u}_i;\beta) - \boldsymbol{\psi}_i^T \left\{ a_{i,0}\mathcal{M} + b_i^{S_iS_i} \boldsymbol{f}_{,\boldsymbol{u}}(\boldsymbol{u}_i;\beta) \right\} \right],$$

2. intermediate stage states, $\boldsymbol{u}_i^{(s)}$ for $s = 1 \ldots S_i - 1$:

$$\frac{dL_i}{d\boldsymbol{u}_i^{(s)}} = c_i \left[ b_i^{S_is} g_{,\boldsymbol{u}}(\boldsymbol{u}_i^{(s)};\beta) - a_{i,0}\boldsymbol{\psi}_i^{(s)T}\mathcal{M} - \sum_{j=s}^{S_i} b_i^{js}\boldsymbol{\psi}_i^{(j)T}\boldsymbol{f}_{,\boldsymbol{u}}(\boldsymbol{u}_i^{(s)};\beta) \right],$$

3. previous timestep states for $j = 1 \ldots P_i$:

$$\frac{dL_i}{d\boldsymbol{u}_{i-j}} = c_i\delta_{j1} d_i^{S_i} g_{,\boldsymbol{u}}(\boldsymbol{u}_{i-1};\beta) - c_i \sum_{s=1}^{S_i} \left[ a_{i,j}\boldsymbol{\psi}_i^{(s)T}\mathcal{M} + \delta_{j1} d_i^s \boldsymbol{\psi}_i^{(s)T}\boldsymbol{f}_{,\boldsymbol{u}}(\boldsymbol{u}_{i-j};\beta) \right],$$

and

4. the initial condition constraint:

$$\frac{dL_0}{d\boldsymbol{u}_0} = -\tilde{\boldsymbol{\psi}}^T \boldsymbol{a}_{,\boldsymbol{u}}(\boldsymbol{u}_0;\beta).$$

The derivatives then can be regrouped to maintain stationarity with respect to each $\boldsymbol{u}_i^s$, each of which gives the equation to solve for the corresponding $\boldsymbol{\psi}_i^s$. To solve the adjoint for a particular timestep, first solve for $\boldsymbol{\psi}_i = \boldsymbol{\psi}_i^{(S_i)}$ via:

$$\begin{aligned} 0 = {} & \left[ a_{i,0}\mathcal{M} + b_i^{S_iS_i} \boldsymbol{f}_{,\boldsymbol{u}}(\boldsymbol{u}_i;\beta) \right]^T \boldsymbol{\psi}_i - \frac{\delta_{iN}}{c_N} h_{,\boldsymbol{u}}^T(\boldsymbol{u}_N;\beta) - \left[ b_i^{S_iS_i} + \frac{c_{i+1}}{c_i} d_{i+1}^{S_{i+1}} \right] g_{,\boldsymbol{u}}^T(\boldsymbol{u}_i;\beta) \\ & + \sum_{\substack{j=1 \\ i+j \le P_{i+j}}}^{N-i} \frac{c_{i+j}}{c_i} \sum_{s=1}^{S_{i+j}} \left[ \left\{ a_{i+j,j}\mathcal{M} + \delta_{j1} d_{i+j}^s \boldsymbol{f}_{,\boldsymbol{u}}(\boldsymbol{u}_i;\beta) \right\}^T \boldsymbol{\psi}_{i+j}^{(s)} \right] \end{aligned}$$

which for $i = N$ simplifies to

$$0 = \left[ a_{N,0}M + b_N^{S_NS_N} \boldsymbol{f}_{,\boldsymbol{u}}(\boldsymbol{u}_N;\beta) \right]^T \boldsymbol{\psi}_N - \frac{1}{c_N} h_{,\boldsymbol{u}}^T(\boldsymbol{u}_N;\beta) - b_N^{S_NS_N} g_{,\boldsymbol{u}}^T(\boldsymbol{u}_N;\beta).$$

Once the final stage adjoint is determined for a timestep, one can then solve for the stage adjoints, $\boldsymbol{\psi}_i^{(s)}$, starting with $s = S_i - 1$ down to $s = 1$:

$$\left[a_{i,0}\mathcal{M} + b_i^{ss}\boldsymbol{f}_{,\boldsymbol{u}}(\boldsymbol{u}_i^{(s)})\right]^T \boldsymbol{\psi}_i^{(s)} - b_i^{S_i s}\, g_{,\boldsymbol{u}}(\boldsymbol{u}_i^{(s)})^T + \boldsymbol{f}_{,\boldsymbol{u}}^T(\boldsymbol{u}_i^{(s)}) \sum_{j=s+1}^{S_i} b_i^{js}\boldsymbol{\psi}_i^{(j)} = 0.$$

For the case where $b_i^{ss} = 0$, *e.g.*, an explicit step in the primal problem, the adjoint only requires a premultiplication of a vector with the mass matrix inverse rather than a full linear solve with the Jacobian. Note that the equations for the intermediate stage adjoints are only dependent on the later stage solutions within that timestep, and all influence from subsequent (later time) timesteps is through the final stage adjoint. To solve for the initial condition adjoint, $\tilde{\boldsymbol{\psi}}$, at $i = 0$,

$$0 = \; - c_1 d_1^{S_1} g_{,\boldsymbol{u}}^T(\boldsymbol{u}_0;\beta) + \sum_{\substack{j=1 \\ j \le P_j}}^{N} c_j \sum_{s=1}^{S_j} \left[\left\{a_{j,j}\mathcal{M} + \delta_{j1} d_1^s \boldsymbol{f}_{,\boldsymbol{u}}(\boldsymbol{u}_0;\beta)\right\}^T \boldsymbol{\psi}_j^{(s)}\right] + \boldsymbol{a}_{,\boldsymbol{u}}(\boldsymbol{u}_0;\beta)^T \tilde{\boldsymbol{\psi}}.$$

The sensitivity to a parameter is then given by:

$$\begin{aligned}
\frac{dJ^E}{d\beta} &= \frac{dL^E}{d\beta} \\
&= h_{,\beta}\left(\boldsymbol{u}_N^{(S_N)}\right) - \tilde{\boldsymbol{\psi}}^T \boldsymbol{a}_{,\beta}\left(\boldsymbol{u}_0;\beta\right) + \sum_{i=1}^{N} c_i \left(d_i^{S_i} g_{,\beta}(\boldsymbol{u}_{i-1}) + \sum_{s=1}^{S_i}\left\{b_i^{S_i s}\, g_{,\beta}\left(\boldsymbol{u}_i^{(s)}\right)\right\}\right) \\
&\quad - \sum_{i=1}^{N} c_i \sum_{s=1}^{S_i}\left\{\boldsymbol{\psi}_i^{(s)T}\left(d_i^s \boldsymbol{f}_{,\beta}(\boldsymbol{u}_{i-1}) + \sum_{j=1}^{(s)} b_i^{sj}\boldsymbol{f}_{,\beta}\left(\boldsymbol{u}_i^{(j)}\right)\right)\right\}.
\end{aligned}$$

# Appendix B

# Temporal Coefficients

The following coefficients are for use in the equation:

*For s=1,...,S*

$$\boldsymbol{f_e}(\boldsymbol{u}_i^{(s)}, t; \beta) = \mathcal{M}\left(a_{i,0}\boldsymbol{u}_i^{(s)} + \sum_{j=1}^{P} a_{i,j}\boldsymbol{u}_{i-j}\right) + d_i^s \boldsymbol{f}(\boldsymbol{u}_{i-1}; \beta) + \sum_{j=1}^{s} b_i^{sj} \boldsymbol{f}(\boldsymbol{u}_i^{(j)}; \beta)$$

$$u(0) = u_0$$

a general temporal discretization of Eq. (2.6).

**First-order backward difference**

$P = 1$, $S = 1$

$$a = 1/\Delta t \left[\begin{array}{cc} 1 & -1 \end{array}\right], \qquad b = \left[\begin{array}{c} 1 \end{array}\right], \qquad d = \left[\begin{array}{c} 0 \end{array}\right]$$

**Second-order backward difference**

$P = 2$, $S = 1$

$$a = 1/\Delta t \left[\begin{array}{ccc} 3/2 & -2 & 1/2 \end{array}\right], \qquad b = \left[\begin{array}{c} 1 \end{array}\right], \qquad d = \left[\begin{array}{c} 0 \end{array}\right]$$

**Second-order trapezoidal rule**

$P = 1$, $S = 1$

$$a = 1/\Delta t \begin{bmatrix} 1 & -1 \end{bmatrix}, \qquad b = \begin{bmatrix} 1/2 \end{bmatrix}, \qquad d = \begin{bmatrix} 1/2 \end{bmatrix}$$

**Second-order explicit singly-diagonal implicit Runge-Kutta (ESDIRK)**

$P = 1$, $S = 3$

$$a = 1/\Delta t \begin{bmatrix} 1 & -1 \end{bmatrix}, \qquad b = \begin{bmatrix} 1/2 & & \\ -3/8 & 1/2 & \\ -1/3 & -2/9 & 1/2 \end{bmatrix}, \qquad d = \begin{bmatrix} 1/2 \\ 5/8 \\ 7/18 \end{bmatrix}$$

**Fourth-order explicit singly-diagonal implicit Runge-Kutta (ESDIRK)**

Reference Jothiprasad[49]. $P = 1$, $S = 5$

$$a = 1/\Delta t \begin{bmatrix} 1 & -1 \end{bmatrix}$$

$$b = \begin{bmatrix} 1/4 & & & & \\ -\frac{1743}{31250} & \frac{1}{4} & & & \\ -\frac{654441}{2922500} & \frac{174375}{388108} & \frac{1}{4} & & \\ -\frac{71443401}{120774400} & \frac{730878875}{902184768} & \frac{2285395}{8070912} & \frac{1}{4} & \\ 0 & \frac{15625}{83664} & \frac{69875}{102672} & -\frac{2260}{8211} & \frac{1}{4} \end{bmatrix}, \qquad d = \begin{bmatrix} \frac{1}{4} \\ \frac{8611}{62500} \\ \frac{5012029}{34652500} \\ \frac{15267082809}{155376265600} \\ \frac{82889}{524892} \end{bmatrix}$$

**Third-order diagonal implicit Runge-Kutta (DIRK)**

Reference Rattenbury[90]. $P = 1$, $S = 3$

$$a = 1/\Delta t \begin{bmatrix} 1 & -1 \end{bmatrix}$$

$$b = \begin{bmatrix} 0.4358665215 & & \\ 0.28206673925 & 0.4358665215 & \\ 1.208496649153235 & -0.64436317065323 & 0.4358665215 \end{bmatrix}, \qquad d = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

**Fourth-order diagonal implicit Runge-Kutta (DIRK)**

Reference Rattenbury[90]. $P = 1$, $S = 5$

$$a = 1/\Delta t \begin{bmatrix} 1 & -1 \end{bmatrix}$$

$$b = \begin{bmatrix} 1/4 & & & & \\ 1/2 & 1/4 & & & \\ 17/50 & -1/25 & 1/4 & & \\ 371/1360 & -137/2720 & 15/544 & 1/4 & \\ 25/24 & -49/48 & 125/16 & -85/12 & 1/4 \end{bmatrix}, \qquad d = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

# Appendix C

# Strong Linear Solvers

In this work, the strength of a linear preconditioner or solver refers to its stability and convergence in the presence of unstable modes. The algorithms discussed below are stronger alternatives to the 'standard' iteration of Eq. (2.4).

## C.1 GMRES

In order to stabilize their iterative method, Campobasso & Giles[17] proposed the use of the GMRES method[100]. GMRES builds an approximate solution in the Krylov subspace

$$\mathcal{K}_m = \langle b, M_P^{-1}Ab, \ldots, \left(M_P^{-1}A\right)^{m-1} b\rangle \tag{C.1}$$

of the preconditioned system. At each iteration, a new vector is generated in the Krylov subspace and added to the basis of the solution. Unlike the time-marching iteration method, GMRES is guaranteed to converge monotonically in a maximum of $n$, the size of the system, iterations regardless of the presence of outlier eigenvalues as long as the matrix $A$ is nonsingular. The full GMRES routine can become quite memory intensive, as each Krylov vector must be stored, and so the restarted GMRES method, called GMRES(m), is often used in practice[21]. In the restarted GMRES method, a maximum number of Krylov vectors is specified, after which the method is restarted with the current approximate solution as the initial guess. The restarted GMRES method does not guarantee convergence, but maintains monotonicity in convergence. The three-dimensional turbine problem of Campobasso & Giles, Figure (3-1d) is an example where too few GMRES vectors does not give convergence. At 20 vectors, the solution stagnates at a residual of $\sim 10^{-6}$, but at 30 vectors converges several orders of magnitude further.

The GMRES method has the property that it solves outlier modes first, and for a small number of outlier eigenvalues the restarted method allows for convergence with relatively few Krylov vectors. As the number of outlier eigenvalues grows, however, so does the minimum number of Krylov vectors needed to solve the problem and the attendant memory requirement.

## C.2 Recursive Projection Method (RPM)

Another option to stabilize the iterative method is the Recursive Projection Method (RPM)[104]. RPM projects the iteration on the unstable subspace $\mathcal{P}$ and the complementary subspace $\mathcal{Q}$. On the $\mathcal{P}$ subspace a Newton-Raphson iteration is performed, while the standard fixed point iteration is projected on $\mathcal{Q}$.

The orthogonal projectors $P$ and $Q$ of the subspaces $\mathcal{P}$ and $\mathcal{Q}$ are defined by the outlier basis, $Z$, respectively, as

$$P = ZZ^T \qquad\qquad Q = I - P. \tag{C.2}$$

The fixed point iteration (3.4), is then projected into two separate functions:

$$\boldsymbol{f} = PF = P\left[(I - M_P^{-1}A)\boldsymbol{x}_n + M_P^{-1}\boldsymbol{b}\right] \tag{C.3}$$

$$\boldsymbol{g} = QF = Q\left[(I - M_P^{-1}A)\boldsymbol{x}_n + M_P^{-1}\boldsymbol{b}\right]. \tag{C.4}$$

The stabilized iterations can then be written as

$$\boldsymbol{p}_{n+1} = \boldsymbol{p}_n + (I - \boldsymbol{f}_p)^{-1}\left[\boldsymbol{f}(\boldsymbol{p}_n, \boldsymbol{q}_n) - \boldsymbol{p}_n\right] \tag{C.5}$$

$$\boldsymbol{q}_{n+1} = \boldsymbol{g}(\boldsymbol{p}_n, \boldsymbol{q}_n) \tag{C.6}$$

where

$$\boldsymbol{p} = P\boldsymbol{x}, \qquad \boldsymbol{q} = Q\boldsymbol{x}, \qquad \boldsymbol{f}_p = P(I - M_P^{-1}A)P.$$

The additional effort of the Newton-Raphson iteration in the unstable subspace is generally small, as it only involves the inversion of a matrix sized by the number of outlier modes:

$$(I - \boldsymbol{f}_p)^{-1} = Z \left[ I - Z^T \left( I - M_P^{-1} A \right) Z \right]^{-1} Z^T. \tag{C.7}$$

When the iteration is detected to be diverging, an approximation of the dominant eigenmodes is found by carrying out a $QR$ factorization for two or more previous iterates of $\boldsymbol{q}$, that is, $\Delta \boldsymbol{q}_n = \boldsymbol{q}_{n+1} - \boldsymbol{q}_n$, and the relative magnitude of the diagonal terms of the resulting upper triangular matrix are compared to determine which columns, if more than one, must be added to $Z$. Both Shroffand Campobasso & Gilesonly attempted to add one real or one complex pair to $Z$ at a time. While both noted that it would be possible, and potentially beneficial, to detect more than two modes at a time, neither did so in their own implementations.

The additional storage cost of the RPM algorithm is directly proportional to the number of unstable outlier modes detected. In their work with turbomachinery problems, Campobasso & Giles[18] generally encountered less than 10 unstable modes, for which GMRES required 30 Krylov vectors to converge.

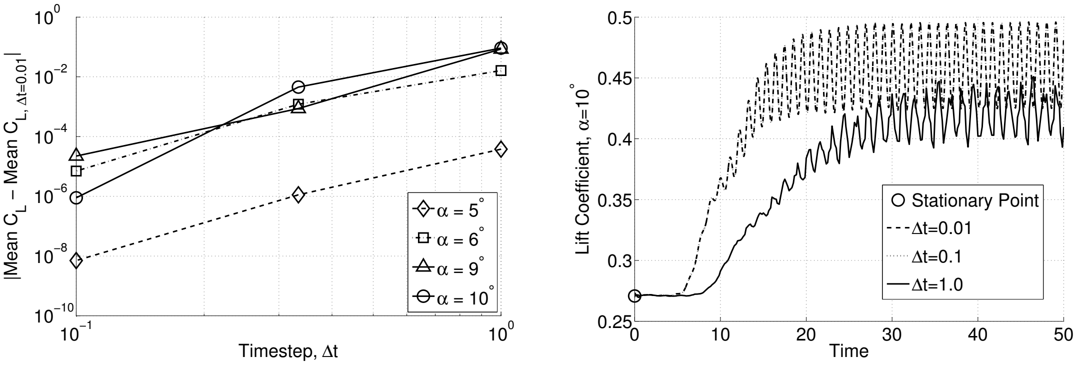
Using the same example case as used for GMRES, the convergence history for RPM in Figure (3-1d) shows the favorable convergence rate once all of the unstable modes are stabilized. The figure also highlights the delay in achieving that convergence rate while RPM searches for the approximate unstable modes.

## Appendix D

# NACA 0012 Mesh and Timestep Refinement Study

In order to verify the accuracy of the time-accurate calculations in Chapters 3 and 4, the impact of mesh and timestep refinement was studied. The dependence of the solution on the timestep was considered at 5, 6, 9, and 10 degrees angle of attack using the baseline mesh with $\Delta t = 0.01$, 0.1, $0.\bar{3}$, and 1.0. Figure (D-1a) plots the difference between the mean lift coefficient of the three coarser timesteps and the mean lift coefficient at the finest timestep. The solutions with $\Delta t = 0.01$ are sufficiently more accurate than the other timesteps such that the difference is a good approximation of the temporal error. For the four angles of attack, the difference decreases by approximately 4 orders of magnitude for a timestep change from 1.0 to 0.1, implying that the time stepping is in the asymptotic, fourth-order accurate, range. The temporal error in the mean lift coefficient with $\Delta t = 0.1$ is less than $10^{-4}$ for all of the angles of attack and less than $10^{-6}$ for the $\alpha = 10^\circ$ case. Figure (D-1b) shows the time-accurate evolution of the solution at 10° for several timesteps. The coarse timestep varies significantly from the fine timestep solution, while the medium timestep shows good agreement. The medium timestep ($\Delta t = 0.1$) was used elsewhere in this study.

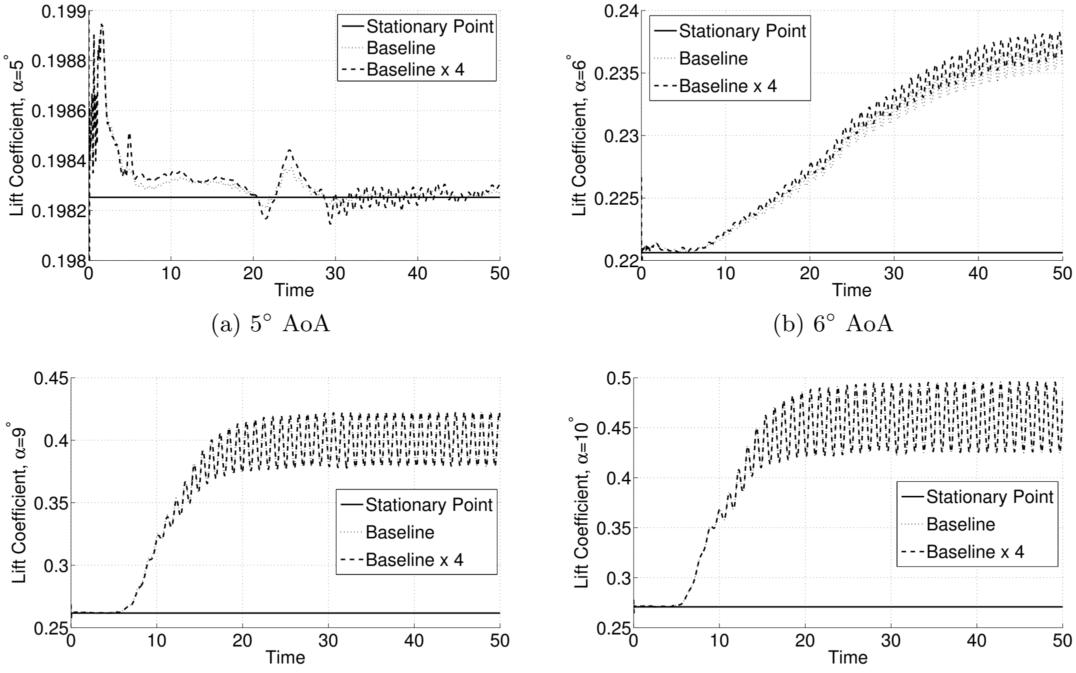
A single uniform mesh refinement is shown for $\alpha = 5^\circ$, $6^\circ$, $9^\circ$, and $10^\circ$in Figure (D-2). For 5°, there is a difference in the unsteadiness between the baseline and refined meshes but remains relatively small. At 6°, the degree of unsteadiness is similar, with an apparent shift in time. At larger angles of attack, however, the differences diminish. While further mesh refinement studies could be conducted, we believe the flow does become unstable somewhere below 6°.



(a) Mean lift difference vs. timestep

(b) $\alpha = 10^\circ$ time history

Figure D-1: Time step study for unsteady NACA 0012 at $Ma = 0.5$, $Re = 1500$



(a) 5° AoA

(b) 6° AoA

(c) 9° AoA

(d) 10° AoA

Figure D-2: Mesh study for unsteady NACA 0012 at $Ma = 0.5$, $Re = 1500$, $\Delta t = 0.01$, ESDIRK4