Output-Based Error Estimation and Mesh Adaptation for Unsteady Turbulent Flow Simulations

Krzysztof J. Fidkowski*

Department of Aerospace Engineering, University of Michigan, Ann Arbor, MI 48109

Abstract

This paper presents a method for estimating output errors and adapting computational meshes in simulations of unsteady turbulent flows. The chaotic nature of such problems prevents a stable unsteady adjoint solution, and existing regularization techniques are costly for large simulations. The method presented foregoes the unsteady adjoint and instead relies on a field-inversion machine-learning (FIML) framework, which only requires unsteady primal solutions without full-state storage or checkpointing. The FIML model yields an adjoint for the averaged solution, which is combined with an averaged unsteady residual to obtain an output error estimate and adaptive indicator. This error estimate is shown to be accurate when the FIML model augments the original unsteady equations with corrections that are not excessively large. The unsteady residual comes from sampling fine-space residual evaluations during the unsteady simulation. A novel objective function based on an adjoint-weighted residual is presented for the field inversion to improve the ability of the FIML model to predict output errors and the domain-interior state. The localized output error drives adaptation of the mesh size and approximation order. Results for three aerodynamic problems ranging in Reynolds number demonstrate accuracy of the error estimates and efficiency of the computational meshes when compared to other adaptive strategies, including uniform and residual-based refinement.

Keywords: Output Error Estimation, Mesh Adaptation, Turbulent Flows, Field Inversion, Machine Learning, Adjoint

1. Introduction

Errors due to finite spatial and temporal discretizations affect all numerical simulations. Quantifying such errors is important due to the trade-off between simulation cost and accuracy: over-refined discretizations lead to wasted computational resources, or, more typically, under-refined discretizations lead to inaccurate solutions. The growth of computational power and demands for higher-fidelity analyses in design have pushed the limits of complexity in physics, resolved scales, and problem size. However, numerical error quantification has lagged behind, even though effects of numerical errors on the solution can rival those of modeling errors or parameter uncertainties, particularly in turbulent flows [1, 2].

The present work considers a posteriori output-based error estimates, as these are the most useful for engineering applications that demand accuracy in output scalars and efficient meshes for their prediction. Much previous work exists in this area, mostly for steady [3, 4, 5, 6, 7, 8, 9, 10] and deterministic unsteady [11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21] simulations. However, current high-fidelity analyses are often unsteady, e.g. large-eddy and other scale-resolving simulations of turbulent flow [22, 2]. These are important for analyzing off-design conditions of aerospace vehicles [23], and also for on-design conditions of bluff bodies, such as road vehicles [24]. Unfortunately, the existing output-based error estimation approaches based on the adjoint solution do not easily extend to such flows, for two reasons. First, long simulation times required for obtaining adequate statistics make the additional storage and computational time of an unsteady adjoint impractical for engineering design. Second, unsteady adjoints are

*Corresponding author
Email address: kfid@umich.edu (Krzysztof J. Fidkowski)

unstable for chaotic or nonlinearly-stable systems, preventing their directed use in long-time sensitivity analyses or error estimates.

Strategies for controlling discretization errors in unsteady turbulent flows have included time windowing, least-squares shadowing, model reduction, and entropy-adjoint methods. In time windowing, the long-time horizon is broken up into smaller time windows, and adjoint-based error estimation is applied on each window separately [25, 26, 27]. However, this approach misses sensitivities arising from features that propagate at time scales longer than the time window, the choice of which is limited by stability of the adjoint. In addition, the cost of the adjoint calculation on each window is not trivial, requiring storage or checkpointing of the primal and backwards time marching [28, 29]. In least-squares shadowing, the long-time unsteady adjoint is regularized by reformulating the initial-condition evolution problem into an optimization problem that minimizes deviation of a perturbed unsteady state from a reference trajectory [30, 31, 32, 33]. However, the solution of the optimization problem is much more expensive than that of the original initial condition problem, even with non-intrusive advances [34, 35]. This cost can be addressed with reduced-order models [36, 37], but these may add additional discretization errors that are difficult to quantify. Finally, the entropy-adjoint approach takes advantage of the fact that entropy variables, which are obtained by a direct variable transformation of the conservative states, are adjoint solutions for an entropy-balance output in which the error arises from spurious entropy production [38, 39]. While the resulting adaptive indicator produces all-around good solutions even for turbulent flows [40], it can lead to over-refinement of areas not important for engineering outputs, and, except in the special case of steady drag [41], the corresponding error estimate is not tied to a specific engineering output.

In this work, we forego the unsteady adjoint or other sensitivity analyses of the unsteady system by restricting our attention to statistically-steady turbulent flows and outputs. Examples of such flows include resolved and under-resolved simulations of turbulence, low-Reynolds number separated flows, buffeting, limit cycle oscillations, etc. In contrast to short dynamic events, for which an unsteady adjoint would provide the requisite sensitivities, statistically-steady turbulent flows can be theoretically simpler, as only derivatives of time-averaged quantities are needed. Reynolds-averaged Navier-Stokes (RANS) models already provide a framework for modeling such flows, with the caveat that multiple models exist, each with advantages in particular flow regimes but lacking in generalization to be sufficiently accurate across regimes, notably in off-design conditions. Indeed, the relatively high degree of modeling error in RANS is the impetus for unsteady turbulent simulations for such cases. In this work, we employ RANS, but to address the accuracy issue, we develop corrected RANS models to match the unsteady results using field inversion and machine learning (FIML) [42, 43, 44, 45, 46, 47, 48]. One of the assumptions in FIML is that the RANS model *can* be corrected to match the unsteady data, which depends on the type of RANS model used and the regime of turbulence studied. We do not set out to characterize or prove the applicability of FIML in a general sense, but we have observed very good performance of the corrected model for the off-design, separated, flows of interest in this work. In addition, a key aspect of FIML as used in this work is that the corrected RANS model need not be general: accuracy is only required in the vicinity, in terms of resolution, of a particular flowfield, as the model is retrained at each adaptive iteration.

The unsteady adaptive method introduced in this paper requires one unsteady primal simulation per adaptive iteration. Neither state storage nor checkpointing [28, 29] are required, and only the time-averaged state and residual need to be stored. The latter is required by one of the two presented error estimates and can be obtained by sampling the fine-space residual during the unsteady run. Following the unsteady simulation, field inversion and machine-learning training yield an equation-level correction to a RANS model. The corrected RANS model is used to obtain a fine-space adjoint that weights the averaged unsteady residual to provide an output error estimate. An indicator from a localized version of this estimate then drives mesh refinement and optimization. The present method focuses only on spatial discretization errors, which are often the most important and difficult to quantify in complex flowfields.

We present the developed error estimation and adaptation method in the context of a discontinuous Galerkin finite-element discretization, which is reviewed briefly in Section 2. Section 3 presents the field-inversion and machine-learning algorithm, with a novel objective function geared for output error estimation. Sections 4 and 5 discuss the error estimation, based on the adjoint-weighted residual, and the adaptation methods, which include both h and p refinement and optimization. Finally, Section 6 presents adaptive results, and Section 7 concludes with a summary and a discussion of future directions.

2. Discretization

2.1. Governing Equations

We consider a system of unsteady partial differential equations in conservative form,

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot \vec{\mathbf{F}}(\mathbf{u}, \nabla \mathbf{u}) + \mathbf{S}(\mathbf{u}, \nabla \mathbf{u}) \quad = \quad \mathbf{0}, \tag{1}$$

where $\mathbf{u} \in \mathbb{R}^s$ is the s-component state vector, $\vec{\mathbf{F}} \in \mathbb{R}^{\dim \times s}$ is the flux vector, dim is the spatial dimension, and $\mathbf{S}$ is the source term arising from the turbulence model, in this work the Reynolds-averaged Navier-Stokes (RANS) equations with the Spalart-Allmaras (SA) closure [49, 50, 51]. A detailed exposition of the equations and closure relations of the model as used presently can be found in our previous work [51, 52].

The FIML approach relies on correcting the turbulence model at the differential-equation level, through a spatially-varying multiplicative factor, β, on the turbulent production term in the eddy-viscosity equation,

$$\frac{\partial(\rho\tilde{\nu})}{\partial t} + \nabla \cdot (\rho\vec{\mathrm{v}}\tilde{\nu}) - \frac{1}{\sigma}\nabla \cdot [\rho(\nu + \tilde{\nu}f_n)\nabla\tilde{\nu}] = -\frac{1}{\sigma}(\nu + \tilde{\nu}f_n)\nabla\rho \cdot \nabla\tilde{\nu} + \frac{c_{b2}}{\sigma}\rho\nabla\tilde{\nu} \cdot \nabla\tilde{\nu} + \boxed{\beta}P - D. \tag{2}$$

In this equation, P is the turbulence production function, and scaling P by β affects the entire solution as the eddy viscosity equation is coupled to the conservation equations. The other terms in the equation are the density, ρ, the velocity, $\vec{\mathrm{v}}$, the turbulence working variable, $\tilde{\nu}$, the turbulent destruction function, D, modeling constants c_{b2}, σ, and a function for dealing with negative viscosity, f_n.

To model high-Reynolds number unsteady flow, we use a variation of the detached-eddy simulation (DES) model [53], which adjusts the wall distance function d based on the mesh size. Since this work concerns mesh adaptation, the mesh size distribution changes across adaptive iterations. To eliminate the additional modeling variability caused by changing meshes, which is a topic of interest in follow-up work, we instead use a fixed attenuation function of the wall distance,

$$d = d_{\max} \tanh(d/d_{\max}), \tag{3}$$

where $d_{\max}$ is a prescribed distance, set to 2% of the chord length for airfoil simulations. Such a fixed function eliminates additional sources of error in the adaptive simulations, allowing us to focus on the impact of chaotic flow and limit-cycle oscillations on the computability of error estimates. Applying error estimation and adaptation to more realistic DES models is a natural extension of the present work.

2.2. The Discontinuous Galerkin Method

The discontinuous Galerkin (DG) method [54, 55, 56] is a finite-element discretization that allows for a comprehensive study of both h (mesh size) and p (approximation order) refinement. In DG, the domain Ω is divided into N_e non-overlapping elements, Ω_e. On each element, the state $\mathbf{u}$ is spatially approximated by $\mathbf{u}_h$, a linear combination of basis functions, here polynomials of order p_e. No continuity constraints are imposed on the approximations on adjacent elements. Formally, $\mathbf{u}_h \in \mathcal{V}_h = [\mathcal{V}_h]^s$, where $\mathcal{V}_h = \{u \in L_2(\Omega) : u|_{\Omega_e} \in \mathcal{P}^{p_e} \ \forall \Omega_e\}$, and $\mathcal{P}^{p_e}$ denotes polynomials of order p_e on element Ω_e.

We obtain the weak form of (1) by multiplying the equation by test functions $\mathbf{v}_h \in \mathcal{V}_h$ and integrating by parts to couple elements via fluxes, here the Roe [57] convective flux and the second form of Bassi and Rebay (BR2) [58] for the viscous flux. The final semilinear weak form is

$$\mathcal{R}_h(\mathbf{u}_h, \mathbf{v}_h) = \sum_{e=1}^{N_e} \mathcal{R}_h(\mathbf{u}_h, \mathbf{v}_h|_{\Omega_e}) = 0, \quad \forall \mathbf{v}_h \in \mathcal{V}_h, \tag{4}$$

where the elemental contribution to the weak form reads (summation on repeated indices i, j is implied)

$$\mathcal{R}_h(\mathbf{u}_h, \mathbf{v}_h|_{\Omega_e}) = \int_{\Omega_e} \mathbf{v}_h^T \partial_i \mathbf{u}_h \, d\Omega - \int_{\Omega_e} \partial_i \mathbf{v}_h^T \mathbf{F}_i \, d\Omega + \int_{\partial\Omega_e} \mathbf{v}_h^{+T} \widehat{\mathbf{F}} \, ds - \int_{\partial\Omega_e} \partial_i \mathbf{v}_h^{+T} \mathbf{K}_{ij}^{+} \left(\mathbf{u}_h^{+} - \widehat{\mathbf{u}}_h\right) n_j \, ds + \int_{\Omega_e} \mathbf{v}_h^T \mathbf{S} \, d\Omega \tag{5}$$

On the element boundary $\partial\Omega_e$, the notations $(\cdot)^+, (\cdot)^-, (\cdot)^b$ respectively denote quantities taken from the element interior, neighbor element, and domain boundary. The unique state on an interior face is $\widehat{\mathbf{u}}_h = (\mathbf{u}_h^+ + \mathbf{u}_h^-)/2$. $\widehat{\mathbf{F}}$ denotes an interface flux, and $\mathbf{K}_{ij}$ is the diffusivity tensor. Details on the definitions of these terms, and on the boundary conditions, can be found in [59].

Choosing a basis for the space $\mathcal{V}_h$, (4) becomes a system of ordinary differential equations,

$$\mathbf{R}^u(\mathbf{U}) \equiv \mathbf{M}\frac{d\mathbf{U}}{dt} + \mathbf{R}(\mathbf{U}) = \mathbf{0}, \tag{6}$$

where $\mathbf{U} \in \mathbb{R}^N$ is the discrete state vector, N is the total number of unknowns, $\mathbf{R}(\cdot) \in \mathbb{R}^N \to \mathbb{R}^N$ is the nonlinear spatial residual, and $\mathbf{M} \in \mathbb{R}^{N\times N}$ is the block-element sparse mass matrix. For steady simulations, the time derivative term drops out, although pseudo-time continuation remains in the solver to drive the steady residual to zero [60]. The solver consists of a Newton-Raphson method with the generalized minimum residual (GMRES) [61] linear solver, preconditioned by an element-line Jacobi smoother with a coarse-level ($p = 1$) correction [56, 62]. For unsteady simulations, we use a third-order modified extended backward difference formula [63] applied to the semi-discrete form.

At this point, we clarify notation used to distinguish between two systems of equations studied in this work. The first system admits an unsteady solution, e.g. Navier-Stokes without a turbulence model, and we denote its spatial residual by $\mathbf{R}(\mathbf{U}) = \mathbf{0}$. The second system augments the first and yields a steady approximation of the time-averaged solution, e.g. via a RANS turbulence model. We denote the discrete state and residual of this augmented system by $\widetilde{\mathbf{R}}(\widetilde{\mathbf{U}}) = \mathbf{0}$, where $\widetilde{\mathbf{U}} \in \mathbb{R}^{\widetilde{N}}$ is the augmented state vector, with $\widetilde{N} \geq N$. To map between the original and augmented states and residuals, we use the restriction, $\mathbf{I}^r \in \mathbb{R}^{N\times\widetilde{N}}$, and prolongation, $\mathbf{I}^p \in \mathbb{R}^{\widetilde{N}\times N}$, operators, which are sparse matrices consisting of single 1's in select rows/columns, and which are never constructed explicitly in practice. The augmented state approximates the time-averaged solution of the original unsteady problem, i.e. $\bar{\mathbf{U}} \approx \mathbf{I}^r\widetilde{\mathbf{U}}$. Also, for the RANS problems of interest here, the augmented residual can be written as

$$\widetilde{\mathbf{R}}(\widetilde{\mathbf{U}}) = \begin{bmatrix} \mathbf{R}(\mathbf{I}^r\widetilde{\mathbf{U}}) + \mathbf{R}^{\text{aug}}(\widetilde{\mathbf{U}}) \\ \widetilde{\mathbf{R}}^{\text{aug}}(\widetilde{\mathbf{U}}) \end{bmatrix}, \tag{7}$$

where $\mathbf{R}^{\text{aug}}$ is an additive change to the original residuals, and $\widetilde{\mathbf{R}}^{\text{aug}}$ is the set of new residuals associated with the eddy-viscosity equation, (2).

2.3. The Output Adjoint

The adjoint solution for a scalar output is the sensitivity of the output to residual source perturbations [4, 8]. We only require steady adjoints in the present work, which are significantly less expensive than their unsteady counterparts. Given a finite-dimensional solution $\mathbf{u}_h \in \mathcal{V}_h$ satisfying (4), and a scalar output functional of the solution, $\mathcal{J}_h(\mathbf{u}_h)$, the adjoint solution $\boldsymbol{\psi}_h \in \mathcal{V}_h$ satisfies the bilinear form

$$\mathcal{R}_h'[\mathbf{u}_h](\mathbf{w}_h, \boldsymbol{\psi}_h) + \mathcal{J}_h'[\mathbf{u}_h](\mathbf{w}_h) = \mathbf{0} \quad \forall \mathbf{w}_h \in \mathcal{V}_h, \tag{8}$$

where the primes denote Fréchet linearization about the argument in square brackets. Given a basis for $\mathcal{V}_h$, (8) becomes a linear system of equations for the adjoint coefficients, $\boldsymbol{\Psi} \in \mathbb{R}^N$,

$$\left(\frac{\partial \mathbf{R}}{\partial \mathbf{U}}\right)^T \boldsymbol{\Psi} + \left(\frac{\partial J}{\partial \mathbf{U}}\right)^T = \mathbf{0}, \tag{9}$$

where $J(\mathbf{U})$ is the discrete form of the output $\mathcal{J}_h(\mathbf{u}_h)$. This equation is solved using the same preconditioned-GMRES method used in the Newton-Raphson primal solver. The adjoint solution plays an important role in output-based error estimation and mesh adaptation because discretization errors originate as residual sources: a finite-dimensional state $\mathbf{u}_h$ will generally not satisfy the PDE everywhere. In a variational formulation, this residual source can be measured in a finer approximation space by a relatively inexpensive, compared to solving the fine-space primal, solution injection and residual evaluation. The adjoint, as a residual sensitivity, then weights the adjoint to produce the output error.

3. Field Inversion and Machine Learning

An unsteady simulation produces a time history of discrete states, $\mathbf{U}(t)$. Depending on the time discretization, such a history may be represented by nodal state values, coefficients of temporal basis functions on time intervals, or intermediate stage values within time intervals. We do not restrict the temporal discretization to any specific type, nor do we require storage of the unsteady time history of the state. Instead, we only require the calculation of an average unsteady state,

$$\bar{\mathbf{U}} = \frac{1}{T_f - T_i} \int_{T_i}^{T_f} \mathbf{U}(t)\, dt, \tag{10}$$

where, in a turbulent simulation, the averaging start time T_i is assumed to be after any initial-condition transients, while the final time T_f is assumed to be sufficiently long after T_i to ensure adequate statistics. Note, for one of the error estimates, we also store an average residual, as described in Section 4.

The goal in field inversion and machine learning (FIML) is two-fold: first, to solve an inverse problem for a correction field that makes the corrected turbulence model solution close to $\bar{\mathbf{U}}$; and second, to train a machine-learning model, here a neural network, to output the correction field as a function of local state information in the flow-field.

3.1. Field Inversion

The correction, $\beta(\vec{x})$, is a scalar field that multiplies the production term in the eddy-viscosity transport equation of the SA turbulence model. The nominal value of this correction field is $\beta = 1$. In this work, $\beta(\vec{x})$ is approximated by a $p = 1$ DG Lagrange basis on each element, represented by the discrete vector $\boldsymbol{\beta}$. Field inversion then reduces to the solution of a discrete optimization problem,

$$\begin{aligned} \min_{\boldsymbol{\beta}} \quad & J^{\text{inv}} \equiv \mathcal{F}(\widetilde{\mathbf{U}}(\boldsymbol{\beta}), \bar{\mathbf{U}}) + \frac{\gamma}{2}(\boldsymbol{\beta} - 1)^T \mathbf{M}(\boldsymbol{\beta} - 1) \\ \text{s.t.} \quad & \widetilde{\mathbf{R}}(\widetilde{\mathbf{U}}, \boldsymbol{\beta}) = \mathbf{0} \end{aligned} \tag{11}$$

where $\mathcal{F}(\widetilde{\mathbf{U}}, \bar{\mathbf{U}})$ measures the error in the solution relative to the unsteady average, and the 1 appearing in $\boldsymbol{\beta} - 1$ corresponds to a vector of all ones. The additional term in (11) is a continuous Tikhonov regularization that penalizes large deviations of the correction field from the nominal value of one. For the problems considered here, a broad range of values of γ was found to be acceptable, and a default value of $\gamma = 10^{-7}$ was used.

In this work, for a baseline error measure, we use the integral of the surface stress distribution error,

$$\mathcal{F}^{\text{dist}}(\widetilde{\mathbf{U}}, \bar{\mathbf{U}}) = \frac{1}{2} \int_{\text{airfoil}} \|\boldsymbol{\sigma}(\widetilde{\mathbf{U}}) \cdot \vec{n} - \boldsymbol{\sigma}(\bar{\mathbf{U}}) \cdot \vec{n}\|^2 \, ds, \tag{12}$$

where $\boldsymbol{\sigma}$ is the stress tensor and $\vec{n}$ is the unit normal vector on the airfoil surface. This error measure drives $\widetilde{\mathbf{U}}(\boldsymbol{\beta})$ to a state that closely reproduces the stress distribution, and hence forces/moments, obtained from the averaged unsteady flowfield.

For numerical error estimation, of interest is not only the surface stress, but also the accuracy of the flowfield in areas of the domain where inaccuracies could propagate and affect the output of interest. We therefore also consider a domain-interior based measure of inaccuracy in the state. Taking the time-average solution as the target truth, a simple measure of flow inaccuracy is the norm of the difference between $\widetilde{\mathbf{U}}$ and $\bar{\mathbf{U}}$. However, such a norm is not invariant with respect to scaling of the state components, and it does not distinguish areas of the domain where errors could affect the output from ones where they could not. To address both of these shortcomings, we use a weighted error norm, in which the weight is based on an adjoint solution,

$$\mathcal{F}^{\text{AWR}}(\widetilde{\mathbf{U}}, \bar{\mathbf{U}}) = \sum_e \frac{1}{2} \left(\mathbf{W}_e^T (\mathbf{I}^r \widetilde{\mathbf{U}}_e(\beta) - \bar{\mathbf{U}}_e) \right)^2 \qquad \mathbf{W}_e^T \equiv \boldsymbol{\Psi}_e^T \frac{\partial \mathbf{R}_e}{\partial \mathbf{U}_e}. \tag{13}$$

The superscript AWR stands for adjoint-weighted residual, which refers to the use of a combination of the adjoint and residual Jacobian matrix for the weight. The adjoint corresponds to the output chosen for adaptation and contains the

residual sensitivity information of the output with respect to domain-interior errors. The elemental Jacobian matrix converts, in a linear manner, state errors to residuals, which are then weighted by the adjoint. When there is a mismatch in the state rank between $\widetilde{\mathbf{U}}$ and $\bar{\mathbf{U}}$, the restriction operator in (13) ensures that only the common state components are compared.

Using either of the two error measures, (12) or (13), the field inversion problem is solved using a limited-memory Broyden–Fletcher–Goldfarb–Shanno (L-BFGS) algorithm [64], with a history of 10 updates for approximating the inverse Hessian. Each iteration amounts to a steady RANS solution, and 20 iterations are taken, by which point the inversion objective J^{inv} has been observed to drop 2-3 orders of magnitude and stagnate. These are not expensive as the solution from each iteration provides the initial guess for the state at the next iteration, so that an aggressive pseudo-time continuation solution strategy can be used. Compared to the unsteady simulation, the computational cost of the inversion is in all cases at least an order of magnitude lower.

The gradient of J^{inv} with respect to $\boldsymbol{\beta}$ required for the inversion is evaluated using an adjoint. The residual linearization with respect to $\boldsymbol{\beta}$ is calculated using finite differences, and this is an inexpensive calculation because the correction affects a residual source term, so that no inter-element effects need to be considered. The linearization for all elements is computed non-intrusively with a few entire-domain residual evaluations.

3.2. Machine Learning

Field inversion yields a single correction field for a particular mesh and solution approximation order. For error estimation, we would like to compare solutions or outputs computed on different approximation spaces, and hence we need a model for how the correction field changes with the solution. More subtly, the dependence of the correction field on the state affects the adjoint used in output error estimation, through the residual Jacobian linearization, and proper accounting for this effect requires a non-static correction field.

For tractability of computation, we assume a functional dependence of the correction field on the local state, its gradient, and the geometry, specifically the wall distance. Such a local dependence can still capture propagation effects inherent to convective systems, as the correction enters at the residuals level. The functional dependence can be an analytical model [65] or an artificial neural network [66, 67], with the latter forming the basis of the field-inversion machine-learning (FIML) approach to corrected turbulence modeling [42].

Previous works have shown that FIML yields corrected turbulence models that better match the high-fidelity target data [43, 45, 46, 48], although the generality of this approach is still in debate [68]. Presently, we do not seek a general-purpose turbulence model, only one that is accurate close to a given training solution. Such a solution-specific model also simplifies the network inputs, which can be dimensional states and gradients instead of general, non-dimensional features.

Figure 1 shows the structure of the network used in this work, which is a single-hidden-layer perceptron that maps the feature vector, $\mathbf{x}_0 \in \mathbb{R}^{n_0}$, to the correction scalar, β, through the hidden layer, $\mathbf{x}_1 \in \mathbb{R}^{n_1}$. The parameters associated

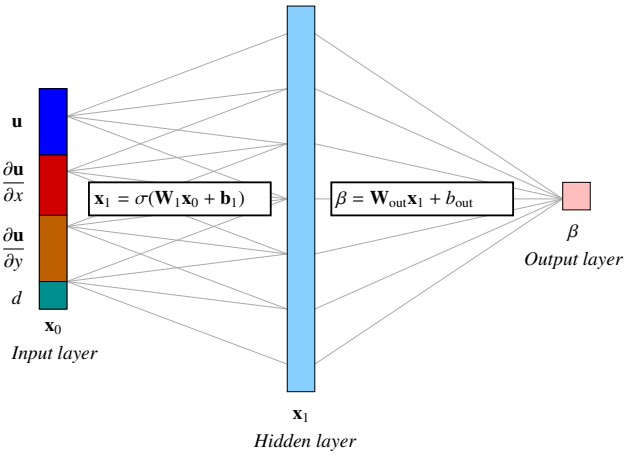


Figure 1: Structure of the artificial neural networks used to predict the correction field.

with the network consist of the weights and biases, $\mathbf{W}_i \in \mathbb{R}^{n_i \times n_{i-1}}$, $\mathbf{b}_i \in \mathbb{R}^{n_i}$, where n_i is the number of neurons in layer i. An entry-wise sigmoid activation function, $\sigma(x) = 1/(1 + e^{-x})$, is applied to the map from the input layer to the hidden layer. The size of the input layer is $n_0 = (1 + \dim)s + 1$ neurons (state, gradient, wall distance), and the size of the hidden layer is set to $n_1 = 30$ for the two-dimensional problems considered in this work.

The network parameters are optimized using an adaptive moment (Adam) estimation algorithm in TensorFlow [69], with a mean squared error loss between the predicted and actual output-layer values. Training values of the correction field come from the field inversion result for one particular unsteady simulation, measured at quadrature points of each element. Thus, each inversion result yields an amount of training data that depends on the mesh size and quadrature order.

The training data are randomized and split into mini-batches of size 1000 for the optimizer, and the learning rate is set to .001. Prior to training, the weights and biases are initialized randomly from a unit normal distribution. 500,000 optimization iterations are taken in each training session for all of the presented results, although the mean-squared error typically stabilizes well before this number. Two to three orders of magnitude drop in the loss are usually observed. The computational time of the network training is negligible, at least two orders of magnitude lower, compared to the unsteady flow solution.

Once trained, the network is implemented as a physics model in the turbulence source calculation of the flow solver. This model affects the residual, which changes the primal and adjoint solutions. The network is linearized in the calculation of the residual Jacobian matrix for the Newton and discrete-adjoint solvers. Due to the nonlinear nature of the network, an extra polynomial order of accuracy is added to the quadrature requirements when using the turbulence correction.

4. Output-Based Error Estimation

We are interested in estimating errors in a time-averaged output, which is a function of the unsteady discrete solution, $\mathbf{U}(t)$,

$$\bar{J} \equiv \frac{1}{T_f - T_i} \int_{T_i}^{T_f} J(\mathbf{U}(t))\, dt. \tag{14}$$

Suppose that we have an unsteady adjoint, $\mathbf{\Psi}(t)$, arising from the Lagrangian

$$\mathcal{L} \equiv \bar{J} + \frac{1}{T_f - T_i} \int_{T_i}^{T_f} \mathbf{\Psi}(t)^{\mathrm{T}} \mathbf{R}^u(\mathbf{U}(t))\, dt, \tag{15}$$

where $\mathbf{R}^u$ is the unsteady residual defined in (6). The division by $T_f - T_i$ is introduced to make the unsteady adjoint of $\bar{J}$ reduce to the steady adjoint of J for steady-state problems. For chaotic problems, we assume that the unsteady adjoint arises from a regularized solution procedure, such as least-squares shadowing [31], although we stress that our approach will not actually compute or use the unsteady adjoint. The unsteady adjoint-weighted residual [70, 19, 71] then yields an estimate of the impact of a perturbation to the unsteady residual, $\delta\mathbf{R}^u(t)$, on the time-averaged output,

$$\delta\bar{J} \approx \frac{1}{T_f - T_i} \int_{T_i}^{T_f} \mathbf{\Psi}(t)^{\mathrm{T}} \delta\mathbf{R}^u(t)\, dt. \tag{16}$$

Going forward, we assume that unsteady residual perturbations arise from the steady residual component in (6), i.e. that the spatial error dominates the temporal error in the PDE discretization, and we drop the u superscript on $\mathbf{R}$. Temporal errors can be estimated through projections onto semi-refined spaces [70], but we reserve this additional complication for future study.

To relate the unsteady error estimation formula in (16) to its steady counterpart, we decompose the unsteady adjoint and residual perturbation into time-averaged and time-varying components,

$$\mathbf{\Psi}(t) = \bar{\mathbf{\Psi}} + \mathbf{\Psi}'(t), \quad \delta\mathbf{R}(t) = \delta\bar{\mathbf{R}} + \delta\mathbf{R}'(t), \tag{17}$$

so that (16) becomes

$$\delta\bar{J} \approx \bar{\mathbf{\Psi}}^{\mathrm{T}} \delta\bar{\mathbf{R}} + \frac{1}{T_f - T_i} \int_{T_i}^{T_f} \mathbf{\Psi}'(t)^{\mathrm{T}} \delta\mathbf{R}'(t)\, dt. \tag{18}$$

When the time-varying components of the adjoint and residual perturbation are not strongly correlated, the output perturbation is dominated by the first term, which is the steady-state expression written using time-averaged quantities. For the chaotic problems of interest in this work, we do not expect a high degree of correlation between local residual perturbations, which depend on the primal solution, and adjoint deviations from the mean, where the unsteady adjoint is obtained in a regularized manner and remains stable, due to the tendency of both the primal and adjoint to decorrelate from their respective initial and terminal conditions for long time integrations. We therefore assume that, for statistically steady-state problems, the first term in (18) dominates.

Now suppose that instead of the original unsteady problem, we solve an alternative steady-state problem to approximate the time-averaged solution directly. The relevant example in this work is a (FIML-corrected) RANS model approximating the time-averaged state of an unsteady simulation of turbulent flow, such as a large/detached eddy simulation based on the Navier-Stokes equations. We assume that the steady problem solves an augmented system of equations for an augmented state, as given in (7). Define the output of the augmented steady-state problem, and the associated adjoint, as

$$\widetilde{J}(\widetilde{\mathbf{U}}) \equiv J(\mathbf{I}^r\widetilde{\mathbf{U}}), \qquad \left(\frac{\partial\widetilde{\mathbf{R}}}{\partial\widetilde{\mathbf{U}}}\right)^T\widetilde{\mathbf{\Psi}} + \left(\frac{\partial\widetilde{J}}{\partial\widetilde{\mathbf{U}}}\right)^T = \mathbf{0} \tag{19}$$

so that the augmented-system output perturbation is $\delta\widetilde{J} \approx \widetilde{\mathbf{\Psi}}^T\delta\widetilde{\mathbf{R}}$.

In numerical error estimation, residual perturbations arise from differences in the resolution of the discretizations. A coarse solution in the approximation space $\mathcal{V}_H$ will in general not satisfy the residual equations on a finer space, $\mathcal{V}_h$, where the finer space in this work is obtained by unit order enrichment. We denote by $\mathbf{U}_h^H(t) = \mathbf{I}_h^H\mathbf{U}_H(t)$ the prolongation of the discrete coarse unsteady state $\mathbf{U}_H(t)$ into its discrete representation in the fine space, $\mathcal{V}_h$. Similarly, $\widetilde{\mathbf{U}}_h^H$ is the prolongation of $\widetilde{\mathbf{U}}_H$ into the fine space. Evaluating the respective fine-space residuals using these prolongated solutions gives the following residual perturbations,

$$\delta\mathbf{R}_h(t) = -\mathbf{R}_h(\mathbf{U}_h^H(t)) + \mathbf{M}_h\mathbf{I}_h^H\mathbf{M}_H^{-1}\mathbf{R}_H(\mathbf{U}_H(t)), \tag{20}$$

$$\delta\widetilde{\mathbf{R}}_h = -\widetilde{\mathbf{R}}_h(\widetilde{\mathbf{U}}_h^H). \tag{21}$$

The negative signs are present because the residual perturbations are added to the left-hand side of the discrete equations. The second term in the expression for $\delta\mathbf{R}_h(t)$ is necessary when only the spatial residual is used from (6) instead of the full unsteady residual, as the coarse spatial residual is not zero in an unsteady problem. The residual prolongation operator, $\mathbf{M}_h\mathbf{I}_h^H\mathbf{M}_H^{-1} \in \mathbb{R}^{N_h\times N_H}$, enables the calculation of the desired residual difference arising from a change in the spatial approximation space.

Based on the above two residual perturbations, we define and study two error estimates in this work:

E1. Time-averaged unsteady residual weighted by a time-averaged adjoint:

$$\delta\bar{J}_h \approx \delta\bar{\mathbf{\Psi}}_h^{\mathrm{T}}\delta\bar{\mathbf{R}}_h \approx \delta\widetilde{\mathbf{\Psi}}_h^{\mathrm{T}}\delta\bar{\mathbf{R}}_h, \qquad \delta\bar{\mathbf{R}}_h \equiv \frac{1}{T_f - T_i}\int_{T_i}^{T_f}\delta\mathbf{R}_h(t)\,dt. \tag{22}$$

E2. Augmented-system residual weighted by the augmented-system adjoint:

$$\delta\widetilde{J}_h \approx \delta\widetilde{\mathbf{\Psi}}_h^{\mathrm{T}}\delta\widetilde{\mathbf{R}}_h = (\mathbf{I}^r\delta\widetilde{\mathbf{\Psi}}_h)^{\mathrm{T}}\delta\mathbf{R}_h(\mathbf{I}^r\widetilde{\mathbf{U}}_h^H) + (\mathbf{I}^r\delta\widetilde{\mathbf{\Psi}}_h)^{\mathrm{T}}\delta\mathbf{R}_h^{\mathrm{aug}}(\widetilde{\mathbf{U}}_h^H) + \delta\widetilde{\mathbf{\Psi}}_h^{\mathrm{T}}\delta\widetilde{\mathbf{R}}_h^{\mathrm{aug}}(\widetilde{\mathbf{U}}_h^H). \tag{23}$$

Note that E2 estimates the spatial discretization error in the FIML-corrected RANS model, which may not provide an appropriate indicator for adapting the unsteady simulations, due to potentially different mesh requirements for reducing spatial residuals of the time-averaged solution versus those of the transient solutions. In contrast, E1 brings information from the unsteady simulation through the averaged unsteady residual, $\delta\bar{\mathbf{R}}_h$. We investigate both for comparison, as E2 has the advantage of not requiring the calculation of $\delta\bar{\mathbf{R}}_h$.

In both error expressions, we use adjoint perturbations instead of the fine-space adjoints themselves, in order to reduce errors arising from lack of strict Galerkin orthogonality in the residuals [21]. These perturbations are obtained by subtracting from the fine-space adjoint its projection onto the coarse space, $\delta\mathbf{\Psi}_h \equiv \mathbf{\Psi}_h - \mathbf{I}_h^H\mathbf{I}_H^h\mathbf{\Psi}_h$. In addition, (23) follows from (7) and (21), and the perturbations in the residuals come from writing $\delta\widetilde{\mathbf{R}}_h = \widetilde{\mathbf{R}}_h(\widetilde{\mathbf{U}}_h^H) - \widetilde{\mathbf{R}}_h(\widetilde{\mathbf{U}}_h)$, as $\widetilde{\mathbf{R}}_h(\widetilde{\mathbf{U}}_h) = \mathbf{0}$.

From the output error expression in (23), we see that the restricted augmented adjoint, $\mathbf{I}^r\delta\widetilde{\boldsymbol{\Psi}}_h$, weights perturbations in the original residual. This means that $\mathbf{I}^r\delta\widetilde{\boldsymbol{\Psi}}_h$ is the sensitivity of the output to perturbations in $\mathbf{R}_h$. We therefore use it as a surrogate for the time-averaged adjoint in (22), $\delta\bar{\boldsymbol{\Psi}}_h \approx \mathbf{I}^r\delta\widetilde{\boldsymbol{\Psi}}_h$. We note that this would not be possible if the system $\widetilde{\mathbf{R}}(\widetilde{\mathbf{U}})$ were not obtained by augmenting the original residuals, e.g. by instead defining completely new equations. In such a case, the property that the restricted augmented adjoint is the sensitivity of the output to perturbations in the original residuals would be lost. In the present work, FIML augments the original Navier-Stokes equations via a corrected RANS model.

We note that many models could be used to augment a given unsteady system. The degree to which the proposed method applies to different steady-state models depends on the size of the correction, measured ultimately by the magnitude of the augmented residuals relative to the original unsteady ones. The quality of the error estimates depends on the corrections being small. Large changes, e.g. by removing most of an existing model and replacing it with another, would yield an incorrect adjoint field (and residuals for E2) based on the new model that would not be appropriate for weighting the unsteady residuals of the original model. Mathematically, this means that we assume that the first term in (23) is not small relative to the other two terms.

Both error estimates, E1 and E2, require the solution of the augmented system, here obtained from the FIML approach. The field inversion and machine-learning rely on an unsteady simulation for the training data, which is the expensive part of the process, but it is only a primal solution. Neither error estimate relies on any unsteady adjoint calculations, and in particular, for E1, we use the restricted augmented-system adjoint as a surrogate for the time-averaged adjoint. Due to the approximations made and the lack of perfect inversion and modeling in FIML, the error estimates obtained will not necessarily strictly converge to the actual error in the unsteady solution as the mesh or order is refined. However, the numerical experiments demonstrate that the error predictions are reasonably accurate, sufficient for adaptation and order of magnitude estimates. In addition, because both error estimates are based on residuals, they will both converge to zero asymptotically with refinement.

5. Mesh Adaptation and Implementation

The error estimates in (22) and (23) are performed after an unsteady simulation, which provides the time-averaged residual (for E1) and FIML training data for both error estimates. Both E1 and E2 take the same form of a steady adjoint-weighted residual on the fine-space discretization, and hence in the following exposition we drop the bar and tilde notation, with the understanding that each error estimate uses its own adjoint/residual. We also drop the h subscript, as all calculations are performed on the fine space.

5.1. Error Localization

When the spatial discretization admits a separation of degrees of freedom into disjoint groups associated with the mesh tessellation, such as elements in DG or cells in the finite-volume method, an inner product between two vectors can be localized to those groups. In our case of DG, we localize an error estimate of the form $\delta J = \delta\boldsymbol{\Psi}^{\mathrm{T}}\delta\mathbf{R}$ to elements, e, to obtain an elemental error indicator

$$\mathcal{E}_e \equiv \left|\delta\boldsymbol{\Psi}_e^{\mathrm{T}}\delta\mathbf{R}_e\right|, \qquad \delta J^{\mathrm{cons}} \equiv \sum_e \mathcal{E}_e, \tag{24}$$

where the subscript e denotes degrees of freedom associated with element e. We have also defined a *conservative* error estimate, δJ^{cons}, as the sum of the error indicators. This error estimate is more robust for chaotic unsteady simulations, as it avoids cancellation of errors between elements – residual perturbations could occur at different times on different elements, leading to unpredictable cancellation that cannot be recovered from time-averaged residuals. We note, however, that, as the problems of interest are nonlinear, even the conservative error estimate is not a bound, and underestimation of the error is possible, especially on under-resolved meshes [8].

5.2. Adaptive Strategies

We consider three adaptive strategies in this work: output-based p-adaptation, residual-based p-adaptation, and output-based mesh optimization. The first two rely solely on an elemental error indicator and follow a fixed-fraction strategy. The elements are sorted according to their error indicators, and a fraction f^{adapt} of elements with the highest

error indicators are chosen for p refinement. A fixed fraction of $f^{\text{adapt}} = 0.2$ is used based on experimentation with the airfoil problems studied in this work. In the output-based case, the error indicator comes directly from (24). For comparison, we also consider refinement based on an unweighted-residual indicator, obtained by omitting the adjoint from (24): $\mathcal{E}_e^{\text{res}} \equiv \left|\delta \mathbf{R}_e\right|$.

The third adaptive strategy is mesh optimization through error sampling and synthesis, MOESS [72, 73, 74]. This method optimizes the mesh using anisotropic metric-based adaptation, by minimizing the output error at a prescribed computational cost. It relies on a metric field for describing the size, anisotropy, and orientation of mesh elements, and changes to the metric field are applied in optimization iterations that equidistribute the marginal error to cost ratio. In the following we highlight the key elements of this method.

MOESS uses a model for how the error on an element changes as the metric changes. Denoting a current elemental error by $\mathcal{E}_{e0}$, the error after applying a step matrix $\mathcal{S}_e \in \mathbb{R}^{\dim \times \dim}$ to the metric on the element is $\mathcal{E}_e = \mathcal{E}_{e0} \exp\left[\text{tr}(\mathcal{R}_e \mathcal{S}_e)\right]$, where $\mathcal{R}_e$ is a symmetric rate tensor that extends the concept of a convergence rate to ansiotropic sizing. This tensor is determined separately on each element through a local sampling procedure that involves projections of the adjoint [74]. Error calculations on the refinement samples provide anisotropy information beyond the scalar error indicator from the inner product in (24). By virtue of the projection approach to sampling, no additional adjoint or residual evaluations are necessary for computing the error samples. This is important for error estimate E1, which uses a single average residual perturbation from the unsteady simulation on one mesh. The total error over the mesh is the sum of the elemental errors, $\mathcal{E} = \sum_{e=1}^{N_e} \mathcal{E}_e$.

MOESS also requires a measure of cost, for which we use degrees of freedom, `dof`. Denoting by $C_{e0} = \texttt{dof}_{e0}$ the current number of `dof` on element e, the new number of `dof` after refinement is $C_e = C_{e0} \exp\left[\frac{1}{2}\text{tr}(\mathcal{S}_e)\right]$. The trace of the step matrix dictates the change in the volume of the element, and decreases in the volume lead to more elements and increased cost. The total cost over the mesh is the sum of the elemental costs, $C = \sum_{e=1}^{N_e} C_e$.

Given a mesh with its mesh-implied metric, and error and cost models that dictate how both of these quantities change with the metric, MOESS iteratively determines a step matrix field, $\mathcal{S}(\vec{x})$, represented by a continuous nodal approximation, that minimizes the error at a fixed cost. This is done through a prescribed number of sweeps through the mesh, each time updating the step matrix nodal values to equidistribute the marginal error to cost ratio. These sweeps are inexpensive as they do not require remeshing or any additional solutions. The result is a new metric field, represented on the current mesh, which is provided to a metric-based mesher to obtain the next mesh. For the two-dimensional remeshing in this work, we use the Bi-dimensional Anisotropic Mesh Generator (BAMG) [75].

As the MOESS a posteriori error estimates and rate calculations depend on the state, achieving an optimal mesh generally requires several mesh optimization iterations, each with separate primal and adjoint solutions and residual evaluations. For unsteady problems, the cost of multiple simulations, even just primal, could become prohibitive. However, for the present case of unsteady simulations modeled and trained by a steady-state augmented system, we have additional flexibility in how to adapt. Error estimate E2 uses the augmented system adjoint and residual and hence does not have an explicit link to the unsteady simulation residuals. Multiple inexpensive MOESS adaptive iterations can therefore be applied on the augmented system without re-running the unsteady simulation, which only serves to provide training data for the model. After converging to a mesh, the unsteady simulation can be re-run to re-train the model. In practice, only 2-3 unsteady simulations have been found to be necessary, depending on the proximity of the starting mesh to optimality.

We cannot apply a similar steady-MOESS sub-cycling procedure for error estimate E1, which uses the average residual from the unsteady simulation, as the residual depends on the mesh. However, even by taking just one MOESS adaptation iteration per unsteady run, we have observed rapid convergence to the optimal mesh, requiring only 3-4 unsteady simulations. Of course, the number of adaptation iterations depends on the starting mesh, which for the present work is the result of MOESS applied to the uncorrected augmented system (baseline RANS). Finally, we note that the use of MOESS sampling via adjoint projections [74] obviates the need for extra residual evaluations on locally-refined elements, which would add to the implementation and run-time costs of the unsteady simulations.

5.3. Implementation

Figure 2 illustrates the unsteady adaptation loop presented in this work. The starting point is a mesh/order distribution, usually obtained from a steady adaptation/optimization using the augmented but uncorrected model. The unsteady simulation is then performed on the current mesh, with a sufficiently-long time horizon for accurate statistics,

and with the measurement start time after the initial transients. The result of the unsteady simulation is an averaged state and output, which provide the target data for field inversion. Machine learning follows field inversion, converting the correction field $\beta(\vec{x})$ into a neural network model for β in terms of the state, its gradient, and the wall distance. This model is used in the corrected-model simulation, which yields the augmented primal and adjoint fields. These are used together with the time-averaged residual from the unsteady simulation (for error estimate E1) to calculate the output-error estimate and adaptive indicator, which then drive mesh adaptation. The process repeats with an unsteady simulation on the adapted mesh.

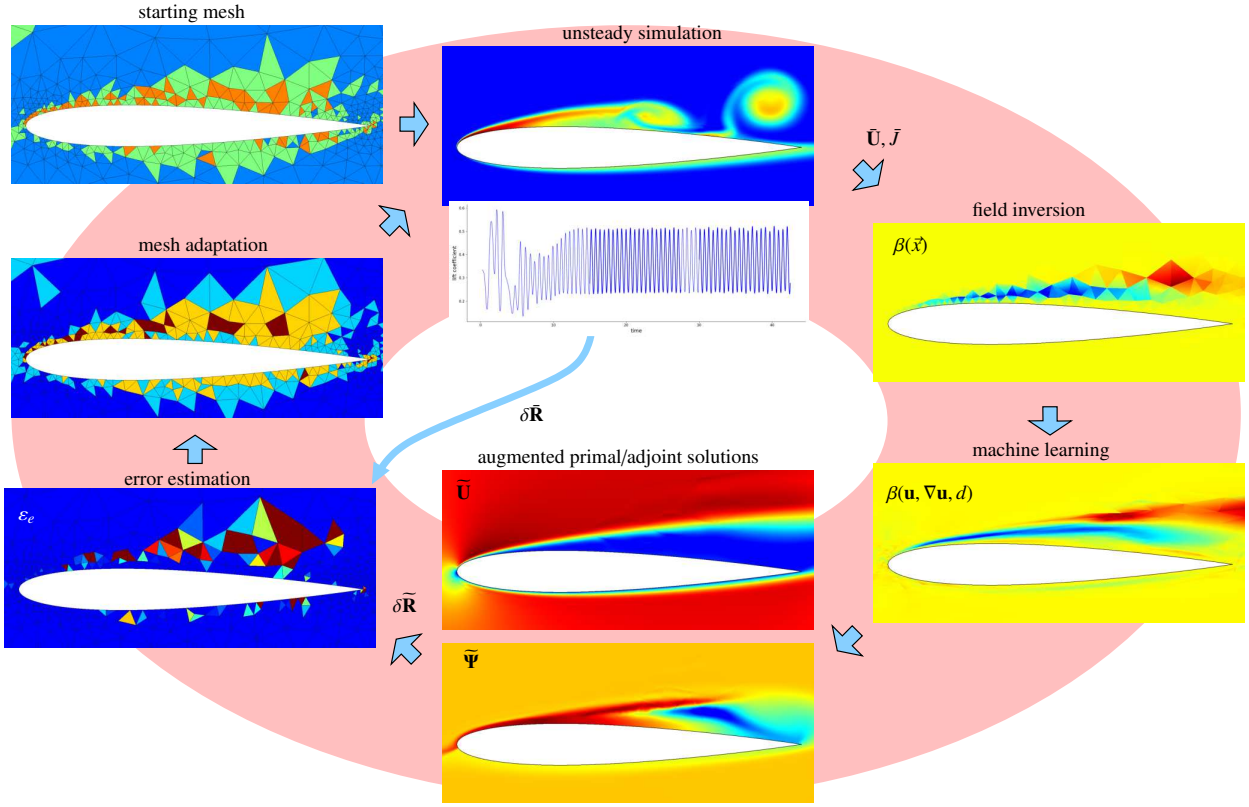


Figure 2: Flowchart of the proposed unsteady mesh-adaptation iterations.

The present adaptation only addresses the error arising from the spatial discretization, which generally dominates other sources of error. The temporal discretization is held fixed with a relatively small step size, determined a priori through temporal convergence studies, to ensure that the spatial error dominates by at least an order of magnitude. In addition, the length of each unsteady simulation is determined a priori to keep the statistical averaging error low. The statistical error is further monitored for each run, by breaking the averaging horizon into two segments and comparing the average on each to the total average. Estimation of temporal and statistical errors, and incorporating them into a unified adaptive process is a topic for future work.

6. Results

In this section we present results for three unsteady aerodynamics problems, ranging from low to high Reynolds numbers. All solutions are obtained using $p = 2$ approximation order, unless otherwise noted, e.g. for p refinement. The unsteady simulations solve the compressible Navier-Stokes equations with or without the DES model given in Section 2. The results focus on efficacy of the error estimates and the convergence of the time-averaged outputs with respect to cost, as measured by spatial degrees of freedom. For clarity, the adopted naming convention for the adaptive strategies is given below.

- **FIML MOESS [E1] [E2] [AWR]**: Mesh optimization at a given target `dof` using the FIML-based approach for the adjoint, error estimates E1 (default) or E2, and field-inversion objective $\mathcal{F}^{\text{dist}}$ (default) or $\mathcal{F}^{\text{AWR}}$ (AWR).
- **RANS MOESS**: Mesh optimization at a given target `dof` using the baseline, uncorrected RANS model.
- **FIML p-adapt [output]**: p refinement at a fixed fraction f^{adapt} using the FIML adjoint and error estimate E1.
- **p-adapt residual**: p refinement using the unweighted unsteady residual.

6.1. Low Reynolds-Number Separated Flow

As a first example, we consider flow governed by the compressible Navier-Stokes equations over a NACA 0012 airfoil, at Mach number $M = 0.2$, angle of attack $\alpha = 7^\circ$, and a relatively low Reynolds number $Re = 10^4$. The computational domain extends approximately 100 chord lengths away from the airfoil and is meshed with unstructured triangles that are curved at the airfoil surface using a cubic mapping.

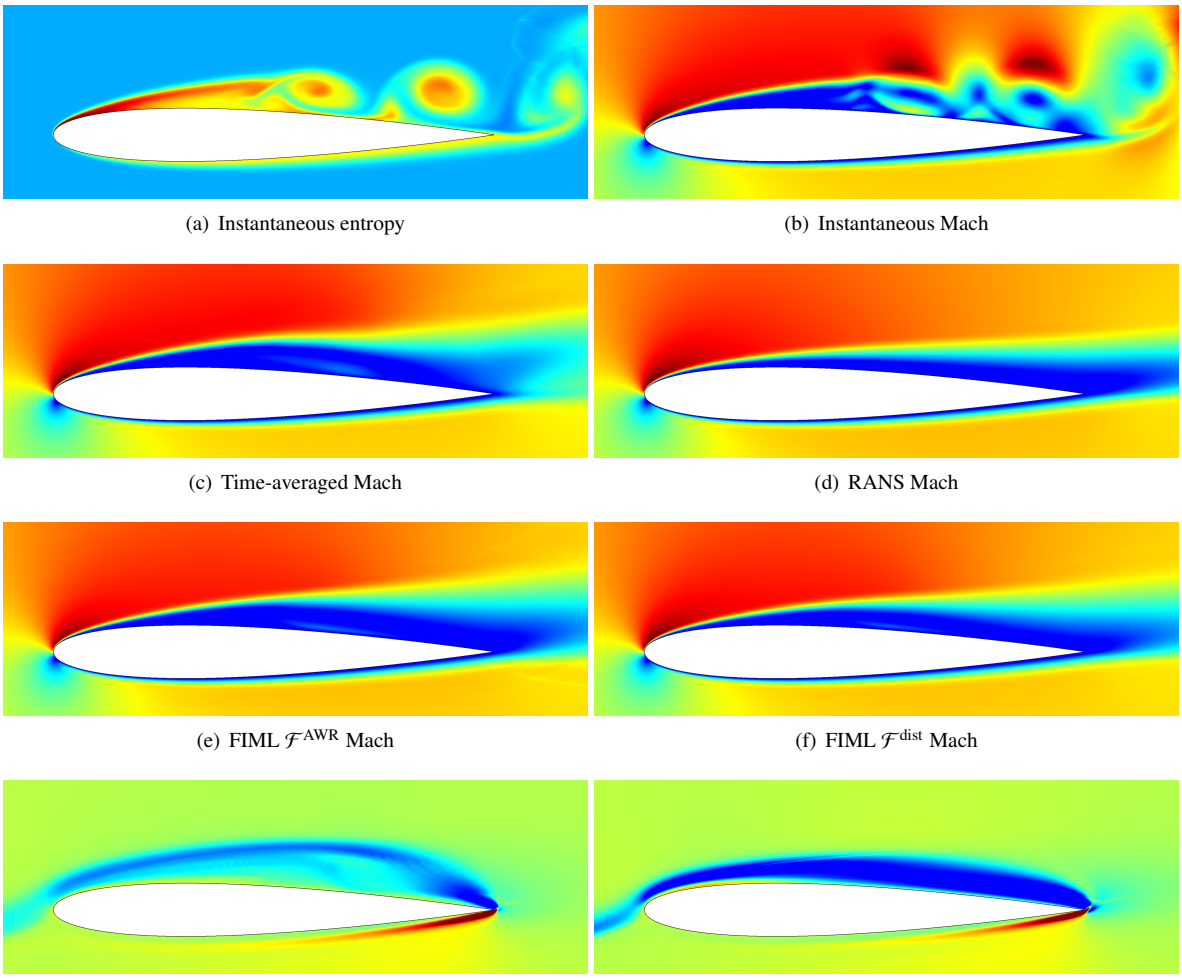
The flow for this problem consists of a thin laminar boundary layer on the lower surface and a thicker boundary layer on the upper surface that breaks down into unsteady vortices approximately half-way down the chord. The vortices are apparent in the instantaneous entropy and Mach number field plots shown in Figure 3. This figure also shows the time-averaged Mach contours, which exhibit a large separation bubble structure on the upper surface. The uncorrected RANS solution, also shown in the figure, fails to predict this structure and instead exhibits a thinner, mostly attached boundary layer. In contrast, the corrected RANS fields, obtained using FIML, more closely represent the time-averaged state. A difference also exists between the field inversion error measures: $\mathcal{F}^{\text{AWR}}$ yields a more accurate flowfield compared to $\mathcal{F}^{\text{dist}}$, due to the former's accounting of errors arising from the domain interior. The last plots in Figure 3 compare the lift adjoint fields, conservation of x-momentum component, for RANS versus FIML, plotted on the same color axis range. The FIML adjoint differs from the RANS one because it is based on a different baseline flow state, and because the residual Jacobian matrix includes a linearization of the ML model. Both adjoint fields recognize the high sensitivity of the lift to trailing-edge residuals, but the FIML adjoint has a more intricate sensitivity field on the upper surface, which reflects the transitional flow in that region. We note that a direct unsteady adjoint solution is unstable for this flow, as shown in Figure 4.

We now look at error estimates and the performance of various adaptive strategies for this case, including output and residual-based methods and uniform refinement in h and p. In the case of FIML MOESS, six unsteady iterations were run, and results are presented as averages over the last three. For RANS MOESS, ten steady iterations were performed at each `dof`, and the presented results are averages over the last five.

Figure 5 shows the results of the adaptations, plotted as the output of interest, the average lift coefficient in this case, versus spatial degrees of freedom. Since the temporal discretization remains fixed, the number of spatial `dof` measures the solution cost. The error estimates are obtained using the conservative sum of indicators, (24), using either E1 or E2. These are shown as shaded bands around the outputs, at $\pm\delta J^{\text{cons}}$. The exact solution is obtained from a $p = 3$ simulation on a uniformly-refined version of the finest lift-adapted mesh.

We make several observations from Figure 5. First, uncorrected RANS converges quickly but under-predicts the lift output. The lack of agreement between RANS and the time-averaged solution is not a surprise, given that the present RANS model is not designed for such low Reynolds numbers. On the other hand, the unsteady output-adaptive approaches perform much better. In particular, mesh optimization yields very good results, even with coarse meshes: e.g. at 3000`dof`, the meshes contain only 500 elements. The only other adaptive approach that comes close is lift-based p refinement, which eventually jumps close to the correct average value by the third adaptive iteration. For this problem, p refinement is efficient, as demonstrated by the relative advantage of uniform p compared to uniform h. However, local p refinement cannot be done haphazardly, as demonstrated by the poor performance of residual-based indicator, which does not get close to the correct output due to its preoccupation with large residuals in areas that minimally impact the lift output. In addition, p refinement is sensitive to the initial mesh, which in this case already has reasonable refinement in areas such as the leading and trailing edge, as shown in Figure 6. This mesh is also used as the initial mesh for uniform h and p refinement.

Figure 7 shows the output history for lift-based p-adaptation applied to this case. The multiple unsteady runs are shown sequentially, so that the horizontal axis indicates the total unsteady run time, as measured in simulation units, where one time unit is the airfoil chord divided by the freestream speed. Each unsteady run includes a "burn" time followed by a time-averaging window. At the initial $p = 1$ uniform order, the approximation space is too coarse to



(a) Instantaneous entropy

(b) Instantaneous Mach

(c) Time-averaged Mach

(d) RANS Mach

(e) FIML $\mathcal{F}^{\text{AWR}}$ Mach

(f) FIML $\mathcal{F}^{\text{dist}}$ Mach

(g) FIML x-momentum adjoint

(h) RANS x-momentum adjoint

Figure 3: Instantaneous and time-averaged entropy (qualitative range), Mach number (range 0 to 0.5), and adjoint (qualitative range) contours for flow over a NACA 0012 airfoil at $M = 0.2, Re = 1 \times 10^4, \alpha = 7°$.

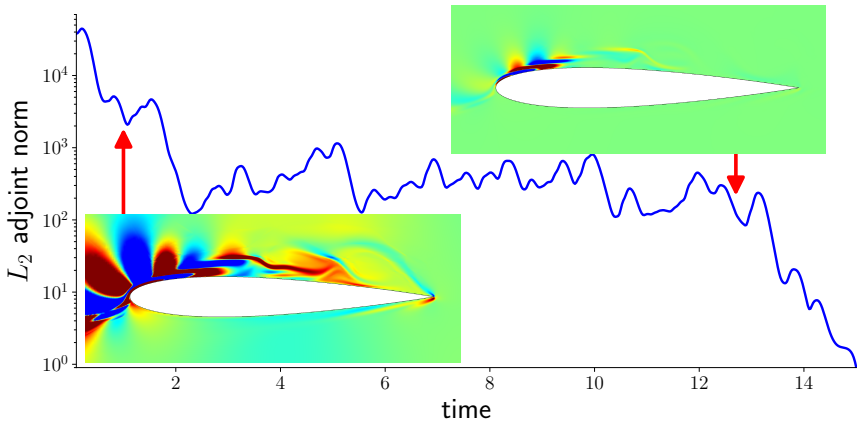


Figure 4: Instability of the unsteady adjoint of the time-averaged lift output, for flow over a NACA 0012 airfoil at $M = 0.2, Re = 1 \times 10^4, \alpha = 7°$. Contours show the x-momentum component of the adjoint.

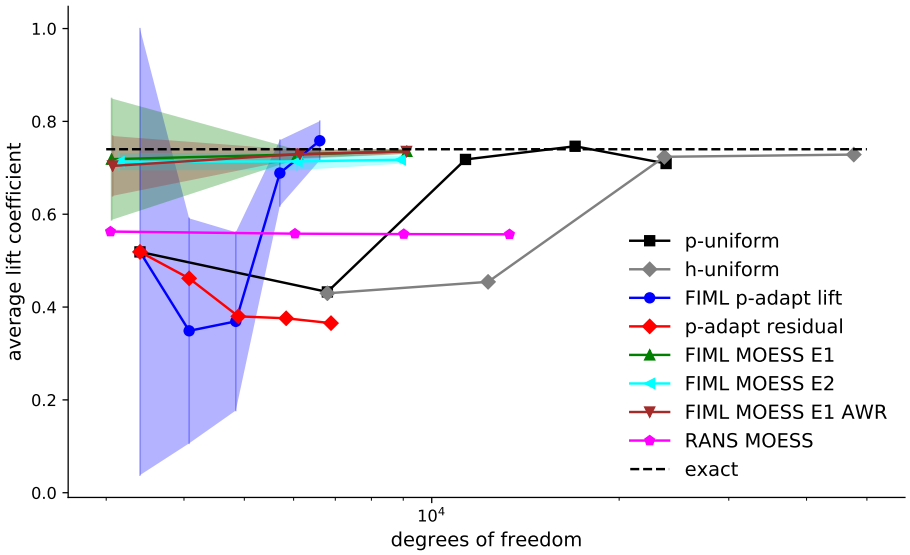
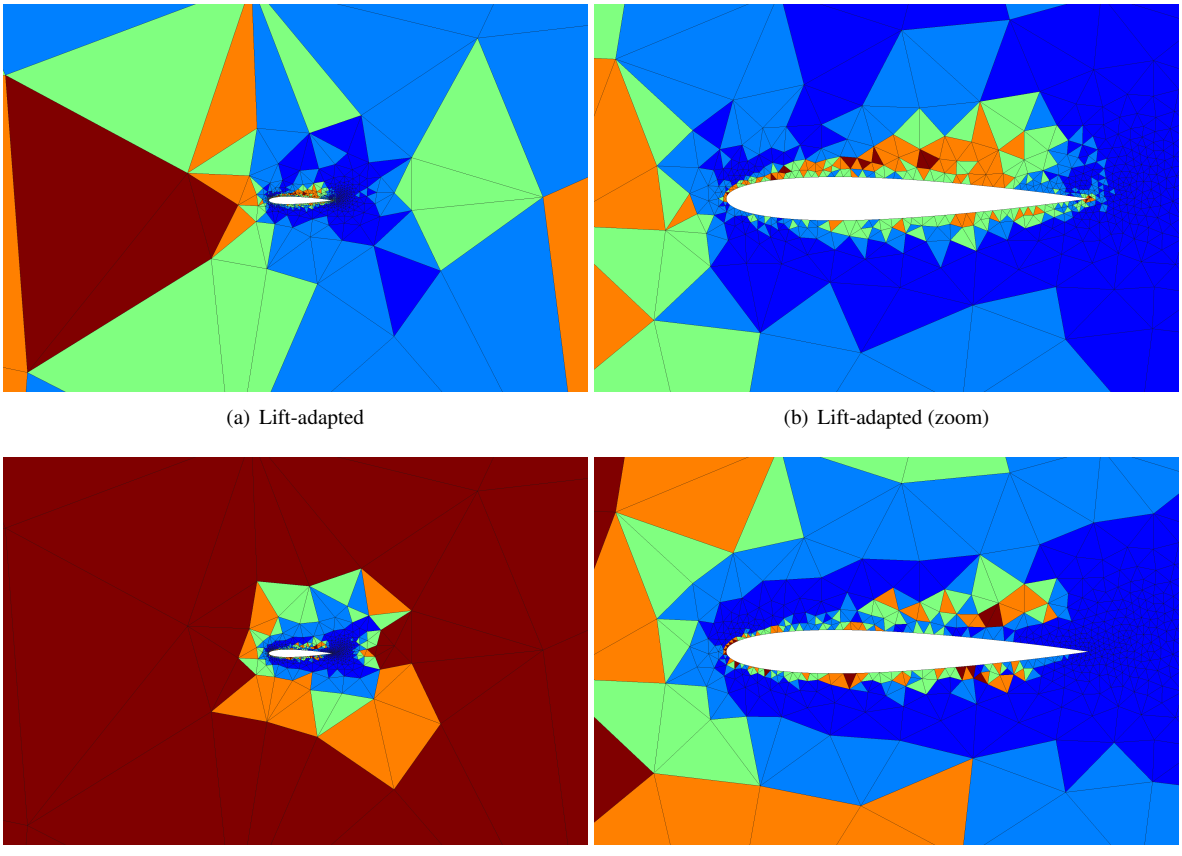


Figure 5: Lift coefficient convergence histories for various adaptive methods applied to the NACA 0012 airfoil at $M = 0.2, Re = 10^4, \alpha = 7°$.



(a) Lift-adapted

(b) Lift-adapted (zoom)

(c) Residual-adapted

(d) Residual-adapted (zoom)

Figure 6: Order fields, showing $p = 1$ (blue) to $p = 5$ (red) for lift and residual-based order-adapted fields for the NACA 0012 airfoil at $M = 0.2, Re = 1 \times 10^4, \alpha = 7°$.

resolve any unsteady behavior, and hence the initial flat output history. We note that the initial mesh has only 1133 elements. The green bars at the end of each unsteady run represent the $\pm\delta J^{\text{cons}}$ error estimates, obtained using the E1 adjoint-weighted residual. We see that estimates are reasonably accurate (within a factor of 2), and that adaptation improves the average output with only localized high-order refinement.

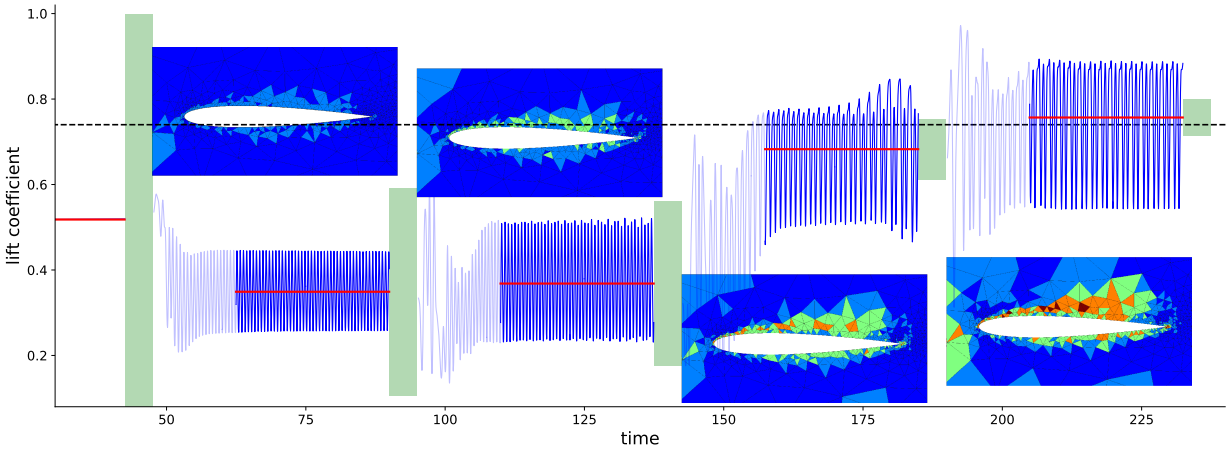


Figure 7: p-adaptation history for the NACA 0012 airfoil at $M = 0.2, Re = 10^4, \alpha = 7°$. The order field range is $p = 1$ (blue) to $p = 5$ (red). Red lines denote time-average outputs, and the green bars are error estimates at $\pm\delta J^{\text{cons}}$ with E1.

In comparison to lift-based p-adaptation, residual-based p-adaptation does not perform very well, as observed in the results shown in Figure 5. The reason for the poor performance of the unweighted residual is that residuals remain large away from the airfoil, particularly on the larger elements. Without an adjoint field to provide a sensitivity, the residual error indicator cannot discriminate between areas of the domain that affect the lift versus those that do not. As a result, the method unnecessarily refines the order on large elements away from the airfoil, as shown in Figure 6. In contrast, the lift-based adaptation focuses more refinement close to the airfoil, in particular at the trailing edge and the region of unsteadiness above the airfoil. Refinement in these areas appears to be necessary for correct average lift prediction in this case.

Turning to FIML MOESS, Figure 8 shows the output history for lift adaptation at 6000`dof`, obtained using error estimate E1. The starting mesh is the RANS mesh, also at 6000`dof`. The unsteady output is severely under-predicted by an unsteady simulation on the initial RANS mesh, even though the resolution does not appear egregiously incorrect in any area. In just one unsteady adaptation iteration, however, a redistribution of the mesh resolution dramatically improves the unsteady output, and additional iterations bring about smaller changes. The error estimate under-predicts the actual error on the coarse mesh, which may be of insufficient resolution to obtain accurate errors. On subsequent iterations, the error estimates become accurate.

Table 1 presents the outputs, error estimates, and actual errors for various FIML MOESS adaptive simulations. The outputs and error estimates were obtained by averaging the results of the latter half of the MOESS iterations. We see a general trend of decreasing actual errors and improved error estimate efficacy with increasing degrees of freedom. Adaptation driven by the E2 error estimate exhibits the slowest error convergence, and by 9000`dof` it still has an error comparable to the 3000`dof` meshes. Its error estimate also does not reflect the larger error. A reason for this reduced performance could be a relative lack of trailing-edge refinement, as shown in Figure 9 possibly due to differences in the fine-space time-averaged residual versus the FIML residual there. This observation suggests that trailing edge resolution is more important for the unsteady simulation than for a steady RANS simulation. Comparing the two FIML inversion error measures, we see that using $\mathcal{F}^{\text{AWR}}$ leads to tighter error estimates, although the actual errors do not change much from the baseline $\mathcal{F}^{\text{dist}}$ case. The tighter error estimates could be due to improved FIML domain-interior solutions with $\mathcal{F}^{\text{AWR}}$ producing a more accurate adjoint field.

Finally, Figure 9 compares four adapted meshes, each at the final MOESS iteration. The distribution of mesh resolution and anisotropy reflects some of the solution features shown in Figure 3. First, the RANS-adapted mesh contains elements of relatively high anisotropy along the upper surface, mostly close to the surface and at the edge

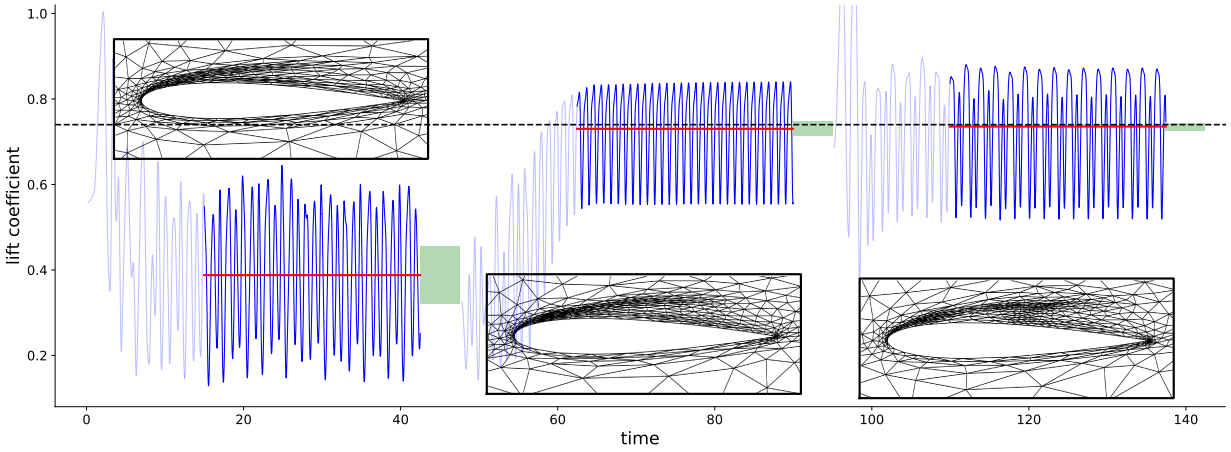
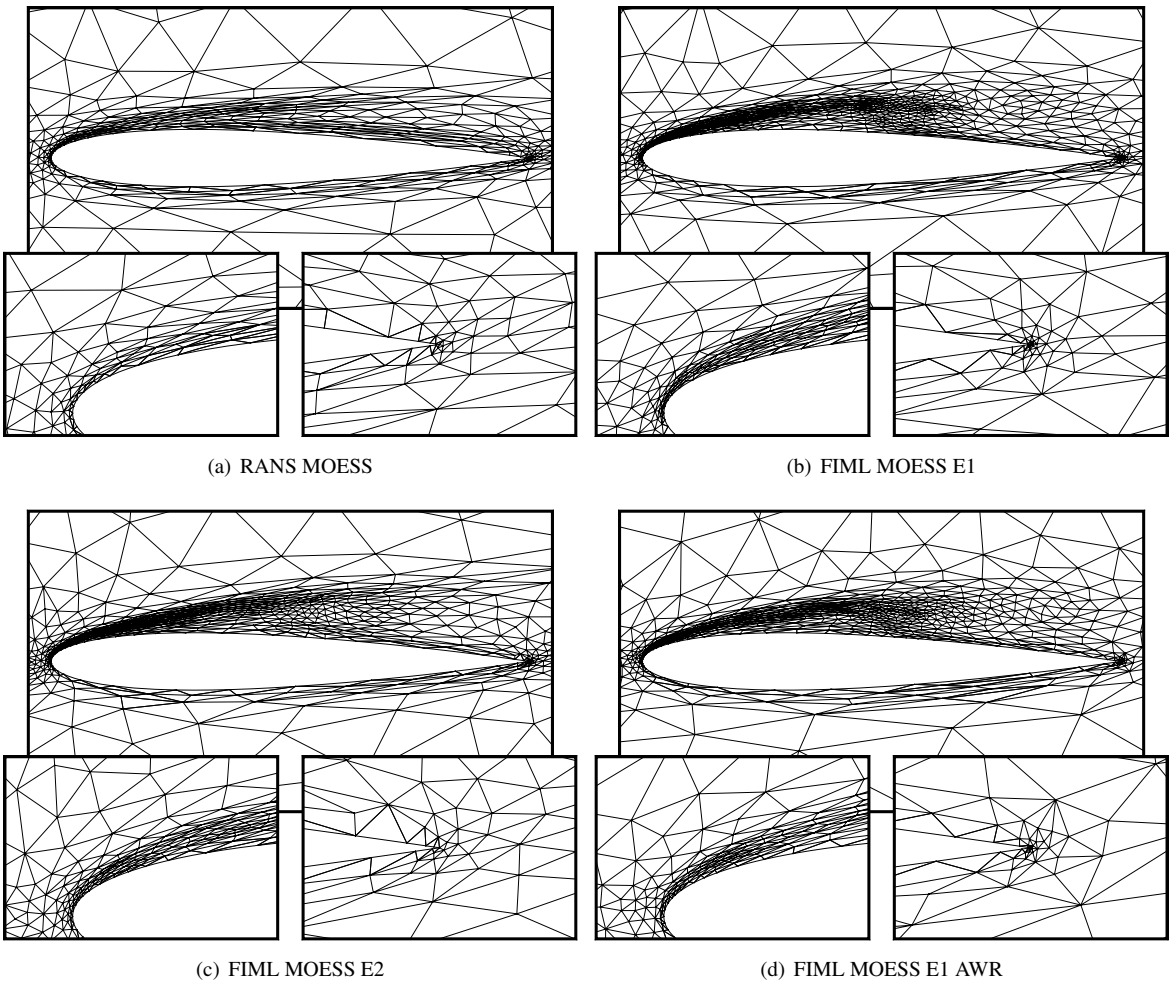


Figure 8: FIML MOESS adaptation history using 6000`dof` for the NACA 0012 airfoil at $M = 0.2, Re = 10^4, \alpha = 7°$. Red lines denote time-averaged outputs, and the green bars are error estimates at $\pm\delta J^{\text{cons}}$ with E1.

Table 1: Averaged lift coefficient and error estimates for FIML MOESS adaptive runs for the NACA 0012 airfoil at $M = 0.2, Re = 10^4, \alpha = 7°$.

Target `dof`	Error Estimate	FIML objective	Output	Error Estimate	Actual Error
3000	E1	$\mathcal{F}^{\text{dist}}$	0.7186	0.1309	0.0214
3000	E1	$\mathcal{F}^{\text{AWR}}$	0.7040	0.0645	0.0360
3000	E2	$\mathcal{F}^{\text{dist}}$	0.7131	0.0184	0.0269
6000	E1	$\mathcal{F}^{\text{dist}}$	0.7288	0.0099	0.0112
6000	E1	$\mathcal{F}^{\text{AWR}}$	0.7294	0.0087	0.0106
6000	E2	$\mathcal{F}^{\text{dist}}$	0.7138	0.0189	0.0262
9000	E1	$\mathcal{F}^{\text{dist}}$	0.7355	0.0057	0.0045
9000	E1	$\mathcal{F}^{\text{AWR}}$	0.7353	0.0051	0.0047
9000	E2	$\mathcal{F}^{\text{dist}}$	0.7175	0.0087	0.0225

of the boundary layer. In contrast, meshes obtained using MOESS driven by the presented unsteady error estimates exhibit thicker boundary-layer adaptation, with more isotropic elements and localized refinement approximately halfway down the chord on the upper surface. In this region, the laminar flow off the leading edge begins to break down into unsteady vortices. The output-adaptive approaches tag this region as important for an accurate average force prediction on the airfoil. In addition, the output-adapted meshes contain more refinement close to the upper aft surface of the airfoil, likely due to the importance of capturing the interaction of the vortices and the boundary layer. The output adaptation also heavily targets the edge of the laminar region coming off the front of the airfoil, as the location of this feature dictates the outer flow over the airfoil, which strongly affects the lift. Lastly, the two meshes obtained using the E1 error estimate are similar, while the one obtained using the E2 error estimate has slight differences: less resolution in the mid-upper surface region, close to the airfoil on the aft upper surface, and near the trailing edge. The reduced refinement in these regions is likely due to differences in the time-averaged residual, which drives the E1 estimate, compared to the steady-state residual of the augmented system.



(a) RANS MOESS

(b) FIML MOESS E1

(c) FIML MOESS E2

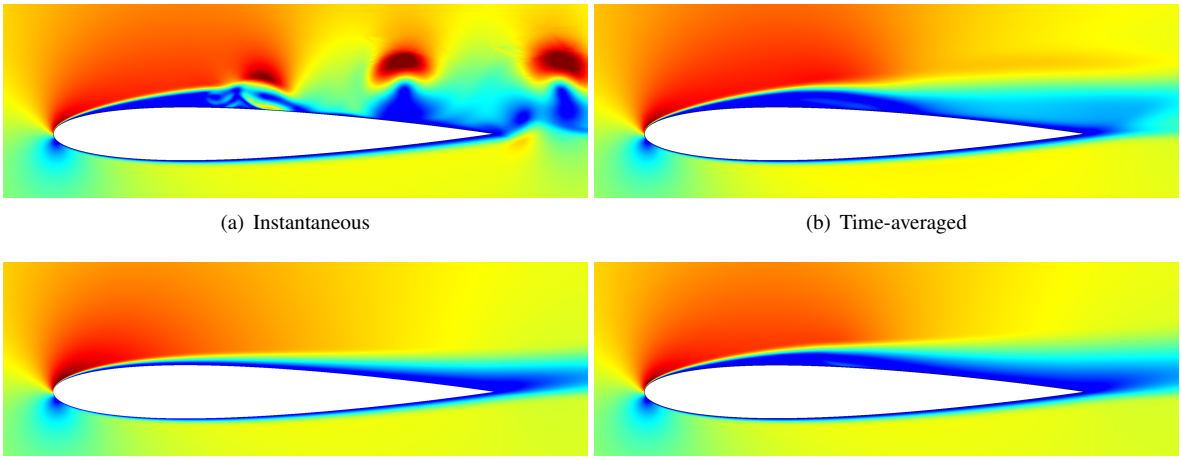
(d) FIML MOESS E1 AWR

Figure 9: Various adapted meshes at 9000`dof` for the NACA 0012 airfoil at $M = 0.2, Re = 10^4, \alpha = 7°$.

6.2. Intermediate Reynolds-Number Transitional Flow

For a second test case, we consider the same flow as in the previous example, but at a higher Reynolds number of $Re = 2{\times}10^4$. The higher Reynolds number reduces the thickness of the boundary layer and increases the complexity of the vortex structure over the upper surface. As shown in the instantaneous flow snapshot in Figure 10, the breakdown

of the laminar flow over the upper surface now occurs earlier, at about 40% of the chord. The time-averaged Mach number field exhibits a separation bubble, with re-attachment at about 60% of the chord. In contrast, the uncorrected RANS model does not produce such a bubble and predicts a mostly attached, thinner boundary layer on the upper surface. With the FIML correction, the bubble structure is better represented, particularly in the boundary-layer growth over the first half of the airfoil. Similarly to the previous example, the unsteady adjoint for a time-averaged lift output is unstable for this case, as shown in Figure 11.



(a) Instantaneous (b) Time-averaged

(c) RANS (d) FIML $\mathcal{F}^{\text{dist}}$

Figure 10: Instantaneous and time-averaged Mach number contours (range 0 to 0.5) for flow over a NACA 0012 airfoil at $M = 0.2, Re = 2 \times 10^4, \alpha = 7°$.

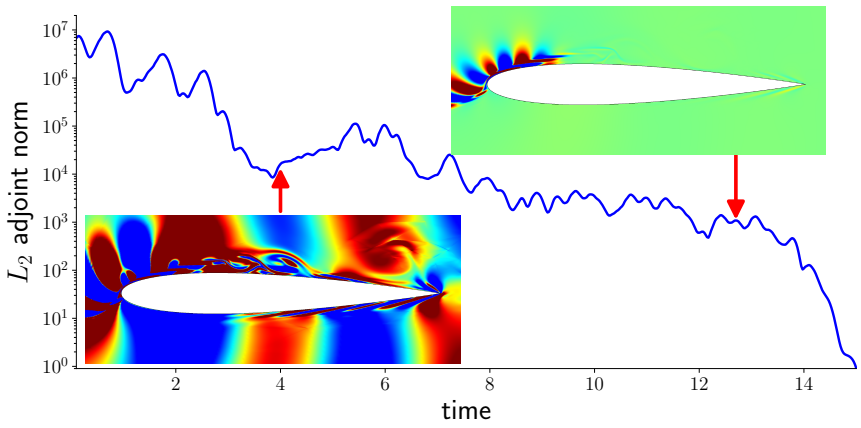


Figure 11: Instability of the unsteady adjoint of the time-averaged lift output, for flow over a NACA 0012 airfoil at $M = 0.2, Re = 2 \times 10^4, \alpha = 7°$. Contours show the x-momentum component of the adjoint.

As in the previous case, we compare various adaptive methods to uniform h and p refinement. The output of interest remains the lift coefficient, and the cost is measured by the spatial degrees of freedom. The exact value of the lift is approximated by a value obtained on a refined mesh at $p = 3$. Figure 12 shows the convergence results. In this case, the RANS prediction of the lift is closer to the exact value, likely due to the re-attachment exhibited by the time-averaged unsteady solution. However, the RANS solution is still inaccurate, and the error does not change with mesh refinement, indicating a persistent modeling error instead of a numerical error. The RANS error estimates reflect this as well, as they are very low, imperceptible on the plot.

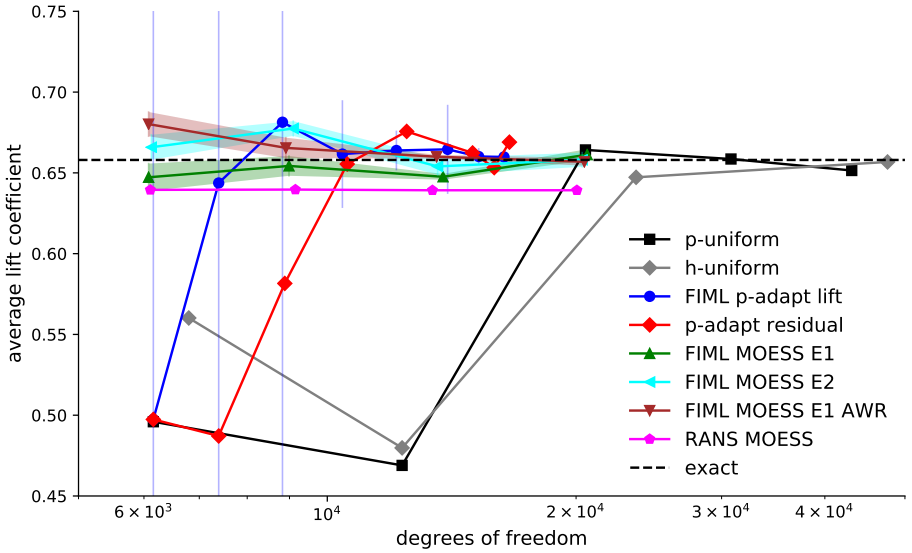


Figure 12: Lift coefficient convergence histories for various adaptive methods applied to the NACA 0012 airfoil at $M = 0.2, Re = 2 \times 10^4, \alpha = 7°$.

The FIML MOESS results are better, as the outputs are closer to the exact value compared to any other method at low `dof`, and the error estimates are representative of the actual error. The error bands decrease with increasing mesh refinement in all of the three results, and the error estimate E1 with $\mathcal{F}^{\text{AWR}}$ yields the best performance. The output-based p-refinement result begins with an inaccurate result on the first iteration, as expected since we start at $p = 1$, and the initial mesh has only 2053 elements. Residual-based p refinement takes longer to get close to the output, and furthermore, by the last two adaptive iterations, it appears to be oscillating away from the exact value instead of converging to it. We note that the code used supports a maximum approximation order of $p = 5$, and elements already at this order that are flagged for refinement remain at $p = 5$. Uniform h and p refinement converge to the exact value, but with more degrees of freedom compared to the adaptive methods.

Figure 13 shows the time history of the lift and the error estimates for the output-based p-refinement simulation. Over the course of five unsteady adaptive iterations, the average lift coefficient generally improves, and the error estimate decreases. Refinement occurs in the expected critical areas of the vortex breakdown region above the airfoil, the trailing edge, and the lower-surface boundary layer.

Figure 14 shows the time history of the adaptation using FIML MOESS and the E1 unsteady error estimate, at 6000`dof`. In just one adaptive iteration starting with the RANS mesh, the time-averaged lift coefficient is already quite close to the exact value. Subsequent variations in the output and error estimate are likely due to a combination of low `dof` and imperfections in the metric-based meshing.

Table 2 presents the average outputs, error estimates, and actual errors for the three FIML MOESS adaptation strategies at various target `dof`. We see a general trend of a reduction in both the estimated and actual error with increasing `dof`. With the exception of the coarsest mesh, which is arguably under-resolved for this problem, the use of $\mathcal{F}^{\text{AWR}}$ yields the most reliable error estimates.

Finally, Figure 15 shows the adapted meshes at 9000`dof` for this simulation. As in the previous example, the refinement patterns correlate to some of the features observed in the primal solutions in Figure 10. The RANS mesh places anisotropic elements inside and on the edge of the boundary layers, which is expected due to the slow variation of the RANS solution along the chord. In contrast, the FIML-based methods place more isotropic elements above the airfoil to capture the vortex breakdown. They also add more refinement in the thicker boundary-layer region towards the front of the airfoil, as flow development in this region affects behavior of the flow over the rest of the airfoil. We note that out of the three FIML-based methods, the one with the E2 error estimate, which does not use the averaged unsteady residual, is closest to the RANS result. Nevertheless, differences are present, in the boundary-layer thickness and the refinement in the transition region.

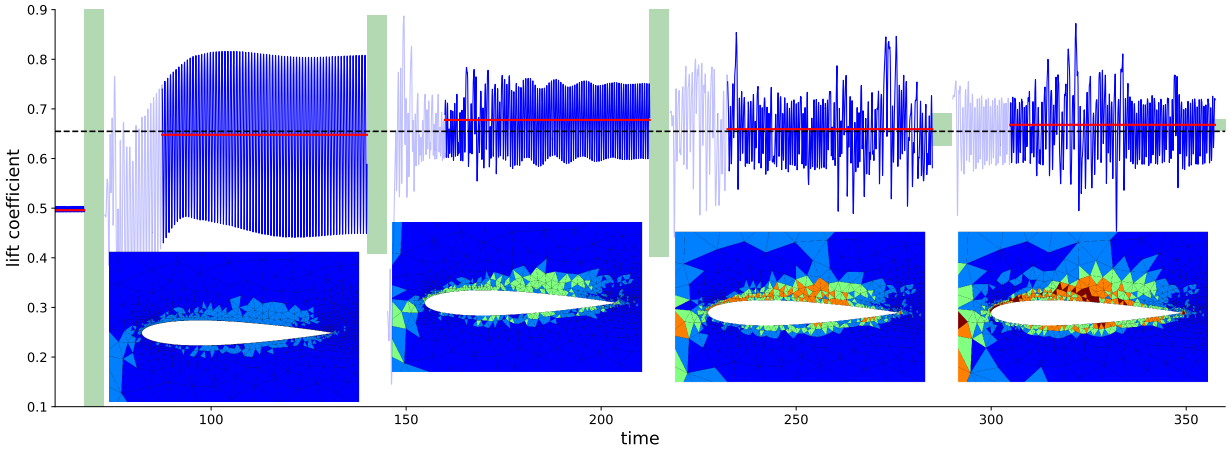


Figure 13: p-adaptation history for the NACA 0012 airfoil at $M = 0.2, Re = 2 \times 10^4, \alpha = 7°$. The order field range is $p = 1$ (blue) to $p = 5$ (red). Red lines denote time-averaged outputs, and the green bars are error estimates at $\pm\delta J^{\text{cons}}$ with E1.

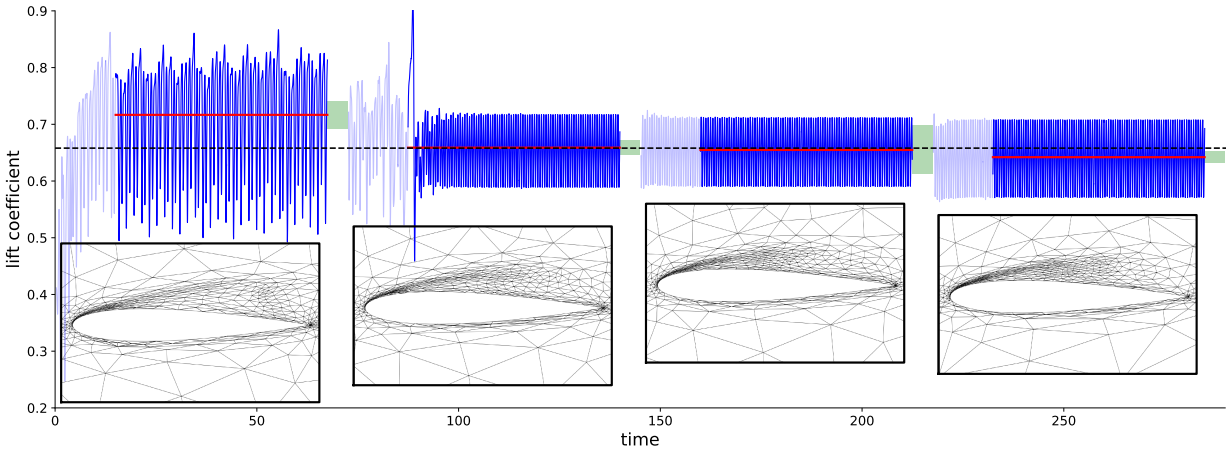
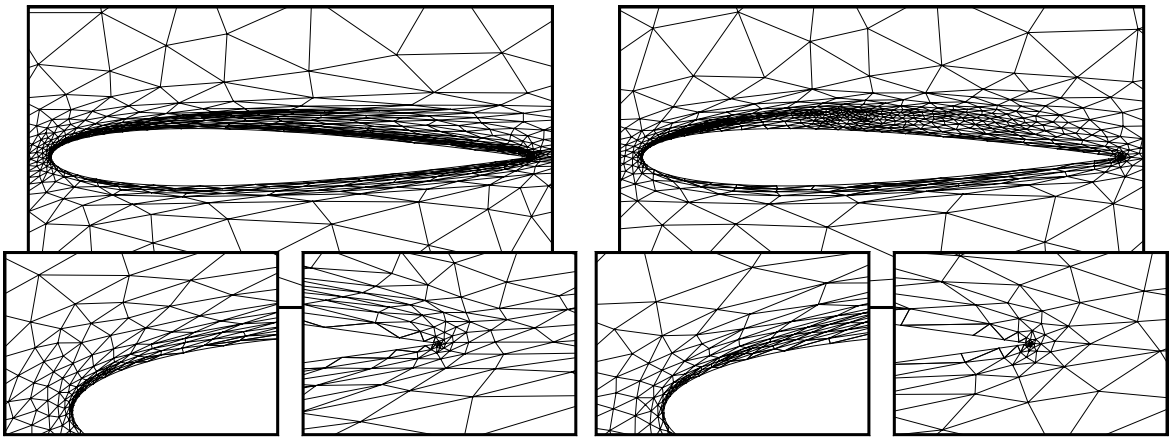


Figure 14: FIML MOESS adaptation history using 6000`dof` for the NACA 0012 airfoil at $M = 0.2, Re = 2 \times 10^4, \alpha = 7°$. Red lines denote time-averaged outputs, and the green bars are error estimates at $\pm\delta J^{\text{cons}}$ with E1.

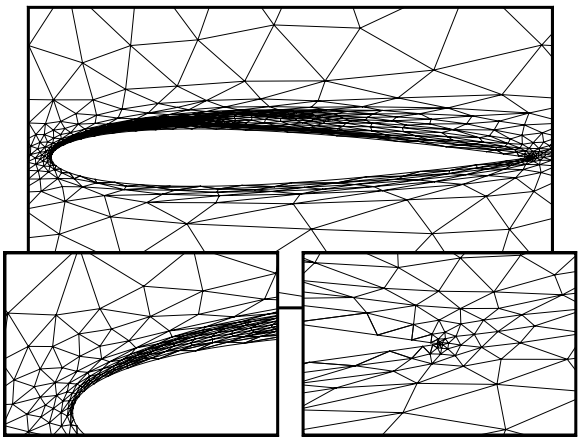
Table 2: Averaged lift coefficient and error estimates for FIML MOESS adaptive runs for the NACA 0012 airfoil at $M = 0.2, Re = 2 \times 10^4, \alpha = 7°$.

Target `dof`	Error Estimate	FIML objective	Output	Error Estimate	$\lvert$Actual Error$\rvert$
6000	E1	$\mathcal{F}^{\text{dist}}$	0.6473	0.0085	0.0107
6000	E1	$\mathcal{F}^{\text{AWR}}$	0.6802	0.0075	0.0222
6000	E2	$\mathcal{F}^{\text{dist}}$	0.6659	0.0077	0.0079
9000	E1	$\mathcal{F}^{\text{dist}}$	0.6543	0.0061	0.0037
9000	E1	$\mathcal{F}^{\text{AWR}}$	0.6655	0.0063	0.0075
9000	E2	$\mathcal{F}^{\text{dist}}$	0.6775	0.0046	0.0195
13500	E1	$\mathcal{F}^{\text{dist}}$	0.6476	0.0018	0.0104
13500	E1	$\mathcal{F}^{\text{AWR}}$	0.6600	0.0012	0.0020
13500	E2	$\mathcal{F}^{\text{dist}}$	0.6540	0.0052	0.0040
20250	E1	$\mathcal{F}^{\text{dist}}$	0.6613	0.0034	0.0033
20250	E1	$\mathcal{F}^{\text{AWR}}$	0.6569	0.0024	0.0011
20250	E2	$\mathcal{F}^{\text{dist}}$	0.6581	0.0044	0.0001

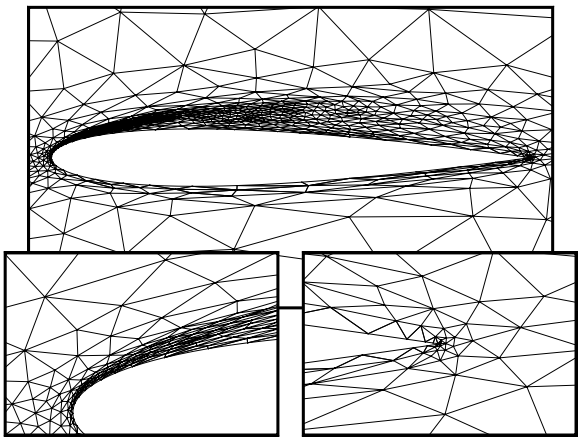


(a) RANS MOESS

(b) FIML MOESS E1



(c) FIML MOESS E2



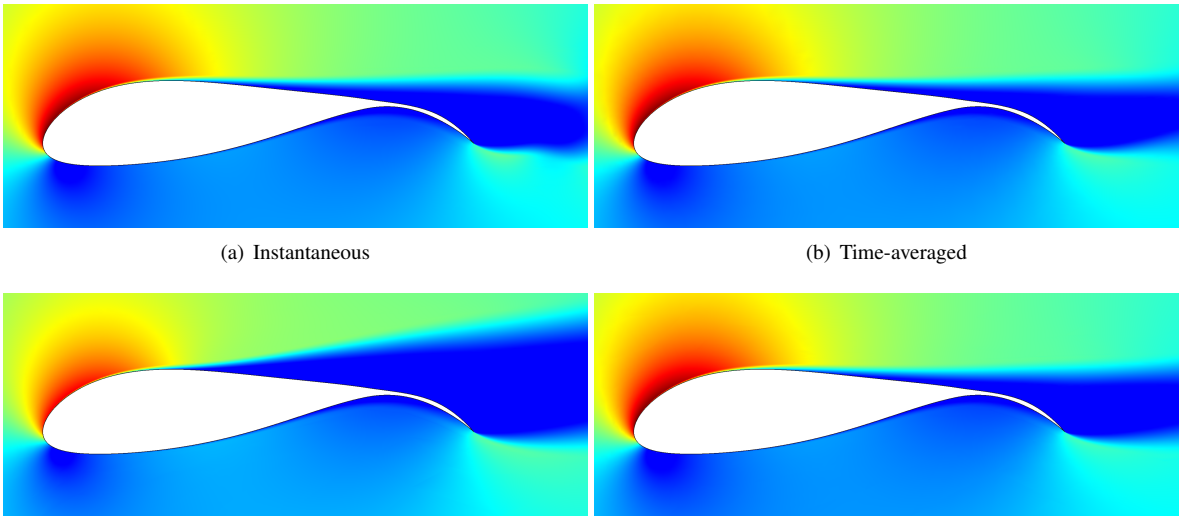
(d) FIML MOESS E1 AWR

Figure 15: Various adapted meshes at 9000`dof` for the NACA 0012 airfoil at $M = 0.2, Re = 2 \times 10^4, \alpha = 7°$.

6.3. High Reynolds-Number Detached-Eddy Simulation

The last case we consider is a detached-eddy simulation (DES) of a high-lift airfoil at $M = 0.2$, $\alpha = 17°$, and $Re = 10^6$. The DES model is the simple prescribed wall-distance modification presented in Section 2.1. The high-lift airfoil geometry is obtained from previous in-house optimization simulations, and it is not a standard airfoil. The domain extends approximately 100 chord lengths away from the airfoil and cubic curved elements provide the geometric fidelity.

Figure 16 shows the Mach number fields for the unsteady, time-averaged, RANS, and FIML solutions. In this case, the unsteady variations are of small scale and characterized by oscillations in the wake. Differences between the instantaneous and time-averaged solutions are therefore small. However, a large difference appears when the standard RANS model is used: the separation region is much larger, leading to a lower lift prediction. The FIML correction brings the flowfield much closer to the time-averaged result. The unsteady adjoint for this case is unstable, as shown by the time-averaged lift output adjoint history in Figure 17, precluding the use of a direct unsteady adjoint approach to adaptation.



(a) Instantaneous (b) Time-averaged

(c) RANS (d) FIML $\mathcal{F}^{\text{dist}}$

Figure 16: Instantaneous and time-averaged Mach number contours (range 0 to 0.5) for flow over a high-lift airfoil at $M = 0.2, Re = 10^6, \alpha = 17°$.

Due to the high Reynolds number of this case, anisotropic h refinement is critical for efficient resolution of the boundary layer. We therefore compare only the MOESS adaptive results, as p-adaptive and uniform h or p refinement results would be highly sensitive to the initial mesh choice. Figure 18 shows the convergence of the output of interest, the average lift coefficient, with increasing `dof`. As in the previous cases, the MOESS outputs and error estimates are computed by averaging the latter half of the adaptive iterations, which number six and ten for the FIML and RANS adaptations, respectively. We see that RANS severely under-predicts the result, and based on its low error estimates and lack of change in the output, the error is due to modeling instead of discretization. The FIML-based methods instead yield results much closer to the exact output, approximated as the result of a fine-mesh, high-order simulation. Their error estimates are also representative of the actual error and decrease with refinement.

Table 3 presents the output error results of the FIML-based methods at four target `dof`. In this case, the error estimates are generally similar. On the coarser meshes, the use of $\mathcal{F}^{\text{AWR}}$ yields a slightly better combination of low actual error and tighter error estimate compared to the other two strategies. By the finest meshes, the error estimates over-predict the output error by about a factor of two, which is reasonable due to the lack of error cancellation in the conservative sum of error indicators.

Figure 19 shows an adaptive convergence history of the lift coefficient versus simulation time for MOESS with the E1 error estimate. In contrast to the previous results, for which the RANS mesh served as the starting mesh for the FIML-based adaptation, this figure demonstrates the ability of the unsteady adaptation to converge to the optimal

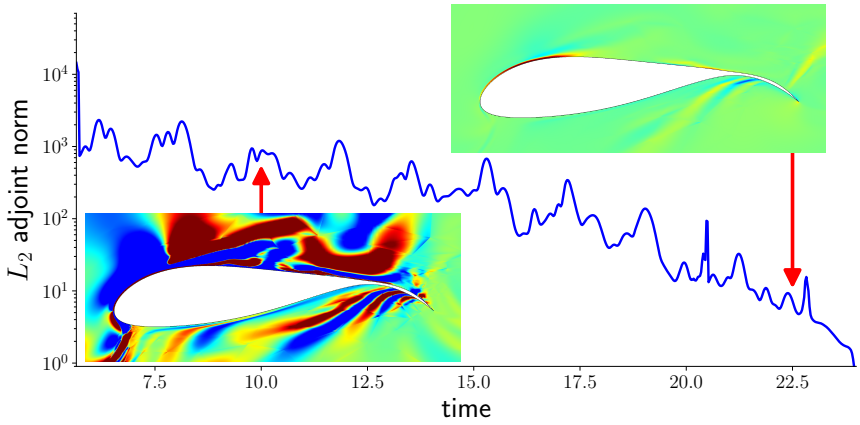


Figure 17: Instability of the unsteady adjoint of the time-averaged lift output, for flow over a high-lift airfoil at $M = 0.2, Re = 10^6, \alpha = 17°$. Contours show the x-momentum component of the adjoint.

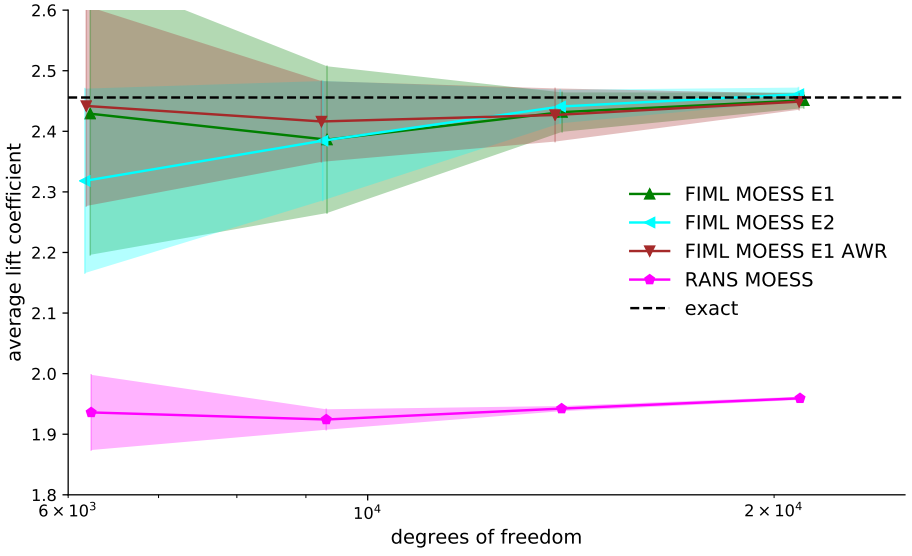


Figure 18: Lift coefficient convergence histories for various adaptive methods applied to a high-lift airfoil at $M = 0.2, Re = 10^6, \alpha = 17°$.

Table 3: Averaged lift coefficient and error estimates for a high-lift airfoil at $M = 0.2, Re = 10^6, \alpha = 17°$.

Target `dof`	Error Estimate	FIML objective	Output	Error Estimate	Actual Error
6000	E1	$\mathcal{F}^{\text{dist}}$	2.4291	0.2329	0.0267
6000	E1	$\mathcal{F}^{\text{AWR}}$	2.4418	0.1651	0.0140
6000	E2	$\mathcal{F}^{\text{dist}}$	2.3182	0.1522	0.1376
9000	E1	$\mathcal{F}^{\text{dist}}$	2.3862	0.1211	0.0696
9000	E1	$\mathcal{F}^{\text{AWR}}$	2.4162	0.0667	0.0396
9000	E2	$\mathcal{F}^{\text{dist}}$	2.3845	0.0986	0.0713
13500	E1	$\mathcal{F}^{\text{dist}}$	2.4317	0.0325	0.0241
13500	E1	$\mathcal{F}^{\text{AWR}}$	2.4269	0.0439	0.0289
13500	E2	$\mathcal{F}^{\text{dist}}$	2.4409	0.0277	0.0149
20250	E1	$\mathcal{F}^{\text{dist}}$	2.4511	0.0128	0.0047
20250	E1	$\mathcal{F}^{\text{AWR}}$	2.4490	0.0133	0.0068
20250	E2	$\mathcal{F}^{\text{dist}}$	2.4611	0.0110	0.0053

mesh starting from a mesh that is far from optimal. The initial mesh is isotropic, with 13500`dof`, and although the boundary layer has small elements, the lack of anisotropy there makes the boundary layer severely under-resolved. We see that on this mesh, the output is far from the exact value, and the large error estimate reflects this. In about three adaptive iterations at the same target `dof`, the result is much improved, and by the fourth adaptive iteration the error estimate begins to be more accurate as well. This result shows that, even for a poor-quality initial mesh, only a few unsteady primal simulations are required for the FIML-based MOESS adaptation to hone in on the optimal mesh.

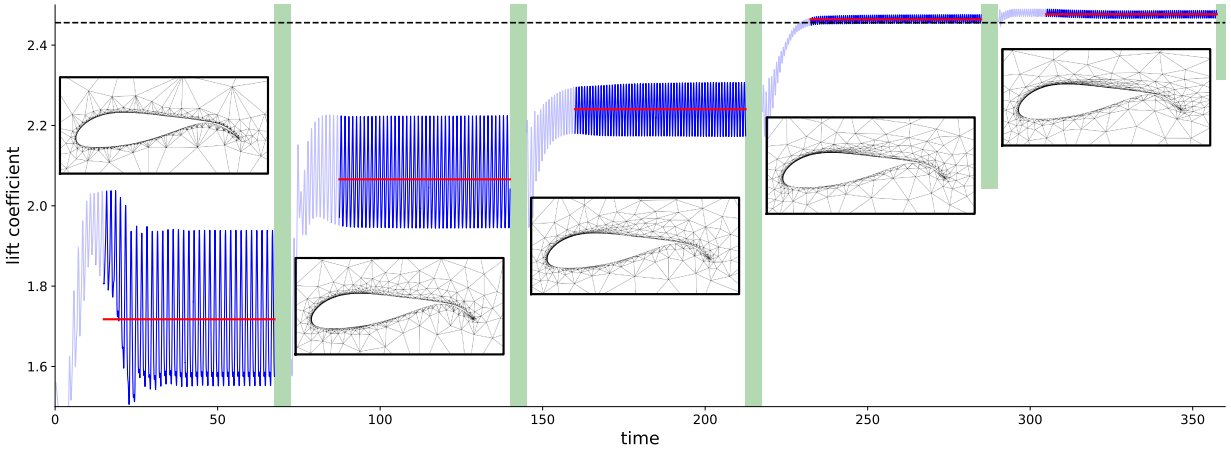
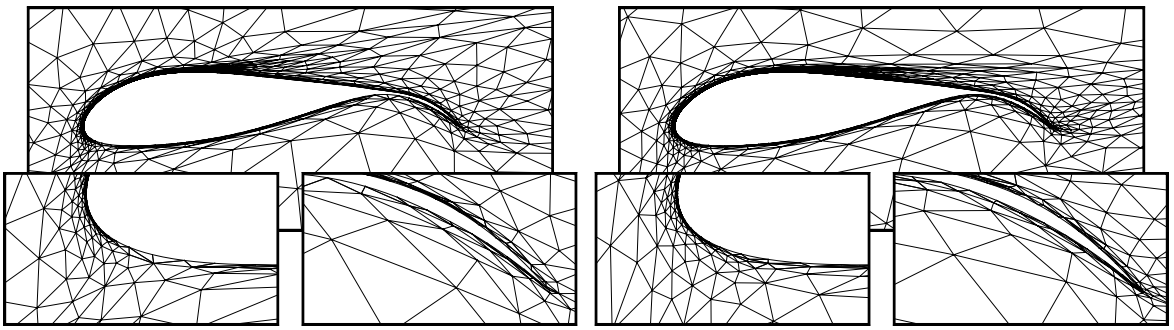


Figure 19: FIML MOESS adaptation history using 13500`dof` for a high-lift airfoil at $M = 0.2, Re = 10^6, \alpha = 17°$. Red lines denote time-averaged outputs, and the green bars are error estimates at $\pm\delta J^{\text{cons}}$ with E1.

Figure 20 shows the adapted meshes for the various adaptive strategies, at 13500`dof`. The meshes mostly follow the flow features evident in Figure 16 and exhibit high resolution with anisotropic elements in the boundary layer. The biggest difference occurs between the RANS and FIML-based methods, in the refinement of the separation region on the upper surface. The RANS adaptation tracks the earlier, wider separation region, whereas the FIML methods track the delayed, less pronounced separation characteristic of the time-averaged solution. In addition, the FIML-based methods exhibit more resolution near the trailing edge and shortly behind it in the wake, an area important for the development and early propagation of wake vorticity. Comparing the FIML-based methods to each other, the meshes are overall similar for this case.



(a) RANS MOESS

(b) FIML MOESS E1

(c) FIML MOESS E2

(d) FIML MOESS E1 AWR

Figure 20: Various adapted meshes at 13500`dof` for a high-lift airfoil at $M = 0.2, Re = 10^6, \alpha = 17^\circ$.

7. Conclusions

This paper presents a method for estimating output errors and adapting meshes in numerical simulations of unsteady turbulent flow. Applications of interest include large-eddy and detached-eddy simulations, LES and DES, with statistically-steady output scalars, such as time-averaged forces. For such simulations, standard output-based methods face the challenge of unstable unsteady adjoint solutions, which must be regularized using techniques that add significant computational cost relative to the already-expensive primal simulation. We address this challenge by using an augmented steady-state model, trained and corrected to match the unsteady data, to calculate a steady-state adjoint suitable for error estimation using the adjoint-weighted residual.

In the present work, Reynolds-averaged Navier-Stokes, with a correction on the turbulence production term, serves as the augmented model to the unsteady Navier-Stokes system. The correction field is determined through a field-inversion calculation that drives the RANS solution towards the time-averaged unsteady solution, using the surface stress distribution as the figure of merit. Following inversion, a neural-network model for the correction field is trained using pointwise samples of the corrected state at element quadrature points. The inputs to the network consist of readily-available local data: the state, its gradient, and the wall distance.

We use the field-inversion, machine-learning approach to derive a single-use rather than a general-purpose turbulence model. The model need not generalize broadly, as in error estimation we only rely on it being accurate locally, for solutions close to the training data. One unsteady primal solution provides the average state for training, and possibly an average residual for error estimation. The augmented system, once trained, then takes over for error estimation and adaptation.

Two error estimates are considered in this work, labeled as E1 and E2. Both measure the spatial error relative to an order-incremented, $p + 1$, fine approximation space, and both use an adjoint-weighted residual form. The difference between the two lies in the residual calculation. The error estimate E1 uses a time-averaged spatial residual perturbation, obtained during the unsteady simulation by sampling the difference in the spatial residual between the coarse and fine spaces, both evaluated using the same state. Keeping track of this residual is slightly intrusive into the unsteady simulation and adds some expense due to the $p + 1$ spatial residual evaluations, although typically the cost of an implicit time step solution still dominates. In contrast, the error estimate E2 uses the fine-space residual directly from the augmented system. While this avoids averaging of the unsteady residual, the extension of E2 to the estimation of temporal errors is not as clear. Finally, both error estimates use the augmented-system adjoint, solved exactly on the $p + 1$ space – a relatively inexpensive, steady calculation. Regarding the applicability of this adjoint to error estimate E1, we show that for augmented systems that retain the original equations, e.g. RANS modifying the Navier-Stokes equations, a restriction of the augmented-system adjoint serves as the sensitivity of the output to residual perturbations in the original equations.

Localized forms of the error estimates drive adaptation of the spatial mesh used in the unsteady calculation. The mesh remains fixed throughout each unsteady calculation. We consider several output-adaptive strategies, using fixed-fraction order adaptation and mesh optimization, and we compare these to uniform refinement, residual-based adaptation, and adaptation on the uncorrected augmented system, i.e. RANS alone. Results are shown on unsteady problems in two dimensions: low-Reynolds number large-eddy simulation, intermediate-Reynolds number transitional flow, and a simple detached-eddy simulation at a high Reynolds number. These results demonstrate: (1) reasonable accuracy of the error estimates, especially in the latter stages of adaptation; (2) slightly better performance of E1 compared to E2; (3) poor performance of the unweighted-residual adaptation, which is distracted by wake and farfield residuals that do not impact the output; and (4) incorrect results from uncorrected RANS adaptation, for which modeling errors outweigh the discretization errors. Regarding the fourth point, we note that the test cases were chosen to be unsteady and/or at Reynolds number outside of the usual domain of applicability of RANS, and hence the large modeling error of RANS is not representative of its applicability to other cases, including a wide class of attached high-Reynolds number flows.

The error estimation and adaptation strategy introduced here relies only on a small number of unsteady primal solutions, and no unsteady adjoints. Field-inversion, machine-learning, and steady-adjoint capabilities are required, but these could be implemented separately from the unsteady simulation program. The only constraints are that both the unsteady and steady programs operate on the same meshes, and that the steady-state equations augment the unsteady ones through a model that admits a correction field. We note that whereas the present results are in two dimensions, the developed methods extend directly to three-dimensional flows, which were not attempted in this

developmental study due to the high computational cost of three-dimensional unsteady simulations.

We have not considered temporal or statistical errors in the present work, but such an extension is possible in the current framework. For example, to estimate temporal errors, the unsteady residual could be decomposed into spatial and temporal contributions, and perturbations to each could be measured separately during the unsteady simulation. Statistical errors could also be accounted for by monitoring the time history of the output of interest. Designing an adaptive technique for balancing these errors to yield the most efficient unsteady simulations is a topic of ongoing work.

References

[1] S. B. Pope, Ten questions concerning the large-eddy simulation of turbulent flows, New Journal of Physics 6 (1) (2004) 35.

[2] J. Larsson, Q. Wang, The prospect of using large eddy and detached eddy simulations in engineering design, and the research required to get there, Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences 372 (2022) (2014) 20130329. doi:https://doi.org/10.1098/rsta.2013.0329.

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

[4] R. Becker, R. Rannacher, An optimal control approach to a posteriori error estimation in finite element methods, in: A. Iserles (Ed.), Acta Numerica, Cambridge University Press, 2001, pp. 1–102.

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

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

[7] M. Nemec, M. J. Aftosmis, Error estimation and adaptive refinement for embedded-boundary Cartesian meshes, AIAA Paper 2007-4187 (2007).

[8] K. J. Fidkowski, D. L. Darmofal, Review of output-based error estimation and mesh adaptation in computational fluid dynamics, AIAA Journal 49 (4) (2011) 673–694. doi:10.2514/1.J050073.

[9] B. R. Ahrabi, W. K. Anderson, J. C. Newman, An adjoint-based hp-adaptive stabilized finite-element method with shock capturing for turbulent flows, Computer Methods in Applied Mechanics and Engineering 318 (2017) 1030–1065. doi:https://doi.org/10.1016/j.cma.2017.02.001.
URL https://www.sciencedirect.com/science/article/pii/S0045782516304947

[10] A. Bazile, E. Hachem, J.-C. Larroya-Huguet, Y. Mesri, Variational multiscale error estimator for anisotropic adaptive fluid mechanic simulations: Application to convection–diffusion problems, Computer Methods in Applied Mechanics and Engineering 331 (2018) 94–115. doi:https://doi.org/10.1016/j.cma.2017.11.019.

[11] D. Meidner, B. Vexler, Adaptive space-time finite element methods for parabolic optimization problems, SIAM Journal on Control Optimization 46 (1) (2007) 116–142. doi:https://doi.org/10.1137/060648994.

[12] M. Schmich, B. Vexler, Adaptivity with dynamic meshes for space-time finite element discretizations of parabolic equations, SIAM Journal on Scientific Computing 30 (1) (2008) 369–393.

[13] T. J. Barth, Space-time error representation and estimation in Navier-Stokes calculations, in: S. C. Kassinos, C. A. Langer, G. Iaccarino, P. Moin (Eds.), Complex Effects in Large Eddy Simulations, Springer Berlin Heidelberg, Lecture Notes in Computational Science and Engineering Vol 26, 2007, pp. 29–48.

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

[15] K. Mani, D. J. Mavriplis, Error estimation and adaptation for functional outputs in time-dependent flow problems, Journal of Computational Physics 229 (2010) 415–440. doi:https://doi.org/10.1016/j.jcp.2009.09.034.

[16] A. Belme, A. Dervieux, F. Alauzet, Time accurate anisotropic goal-oriented mesh adaptation for unsteady flows, Journal of Computational Physics 231 (19) (2012) 6323–6348. doi:10.1016/j.jcp.2012.05.003.

[17] B. T. Flynt, D. J. Mavriplis, Discrete adjoint based adaptive error control in unsteady flow problems, AIAA Paper 2012-0078 (2012).

[18] K. J. Fidkowski, Y. Luo, Output-based space-time mesh adaptation for the compressible Navier-Stokes equations, Journal of Computational Physics 230 (2011) 5753–5773. doi:10.1016/j.jcp.2011.03.059.

[19] Y. Luo, K. J. Fidkowski, Output-based space time mesh adaptation for unsteady aerodynamics, AIAA Paper 2011-491 (2011). doi:10.2514/6.2011-491.

[20] K. J. Fidkowski, An output-based dynamic order refinement strategy for unsteady aerodynamics, AIAA Paper 2012-77 (2012). doi:10.2514/6.2012-77.

[21] S. M. Kast, K. J. Fidkowski, Output-based mesh adaptation for high order Navier-Stokes simulations on deformable domains, Journal of Computational Physics 252 (1) (2013) 468–494. doi:10.1016/j.jcp.2013.06.007.

[22] J. Hoffman, C. Johnson, A new approach to computational turbulence modeling, Computer Methods in Applied Mechanics and Engineering 195 (2006) 2865–2880.

[23] C. L. Rumsey, J. P. Slotnick, A. J. Sclafani, Overview and summary of the third AIAA high lift prediction workshop, Journal of Aircraft 56 (2) (2019) 621–644.

[24] E. M. Papoutsis-Kiachagias, V. G. Asouti, K. C. Giannakoglou, K. Gkagkas, S. Shimokawa, E. Itakura, Multi-point aerodynamic shape optimization of cars based on continuous adjoint, Structural and Multidisciplinary Optimization 59 (2019) 675–694. doi:https://doi.org/10.1007/s00158-018-2091-3.

[25] J. A. Krakos, Q. Wang, S. R. Hall, D. L. Darmofal, Sensitivity analysis of limit cycle oscillations, Journal of Computational Physics 231 (8) (2012) 3228–3245. doi:https://doi.org/10.1016/j.jcp.2012.01.001.

[26] Y. S. Shimizu, K. J. Fidkowski, Output error estimation for chaotic flows, AIAA Paper 2016-3806 (2016). doi:https://doi.org/10.2514/6.2016-3806.
[27] S. W. Chung, J. B. Freund, An optimization method for chaotic turbulent flow, Journal of Computational Physics 457 (2022) 111077. doi:https://doi.org/10.1016/j.jcp.2022.111077.
[28] A. Griewank, A. Walther, Revolve: An implementation of checkpointing for the reverse or adjoint mode of computational differentiation, ACM Transactions on Mathematical Software 26 (1) (2000) 19–45. doi:https://doi.org/10.1145/347837.347846.
[29] E. C. Cyr, J. Shadid, T. Wildey, Towards efficient backward-in-time adjoint computations using data compression techniques, Computer Methods in Applied Mechanics and Engineering 288 (2015) 24–44, error Estimation and Adaptivity for Nonlinear and Time-Dependent Problems. doi:https://doi.org/10.1016/j.cma.2014.12.001.
URL https://www.sciencedirect.com/science/article/pii/S0045782514004800
[30] Q. Wang, Forward and adjoint sensitivity computation of chaotic dynamical systems, Journal of Computational Physics 235 (2013) 1–13. doi:https://doi.org/10.1016/j.jcp.2012.09.007.
[31] Q. Wang, R. Hu, P. Blonigan, Least squares shadowing sensitivity analysis of chaotic limit cycle oscillations, Journal of Computational Physics 267 (2014) 210–224. doi:https://doi.org/10.1016/j.jcp.2014.03.002.
[32] Q. Wang, Convergence of the least squares shadowing method for computing derivative of ergodic averages, SIAM Journal on Numerical Analysis 52 (1) (2014) 156–170. doi:https://doi.org/10.1137/130917065.
[33] P. J. Blonigan, S. A. Gomez, Q. Wang, Least squares shadowing for sensitivity analysis of turbulent fluid flows, AIAA Paper 2014–1426 (2014). doi:https://doi.org/10.2514/6.2014-1426.
[34] A. Ni, Q. Wang, Sensitivity analysis on chaotic dynamical systems by non-intrusive least squares shadowing (NILSS), Journal of Computational Physics 347 (2017) 56–77. doi:10.1016/j.jcp.2017.06.033.
[35] A. Ni, Q. Wang, P. Fernández, C. Talnikar, Sensitivity analysis on chaotic dynamical systems by finite difference non-intrusive least squares shadowing (FD-NILSS), Journal of Computational Physics 394 (2019) 615–631. doi:doi.org/10.1016/j.jcp.2019.06.004.
[36] X. Li, S. Hulshoff, S. Hickel, Towards adjoint-based mesh refinement for large eddy simulation using reduced-order primal solutions: Preliminary 1d Burgers study, Computer Methods in Applied Mechanics and Engineering 379 (2021) 113733. doi:https://doi.org/10.1016/j.cma.2021.113733.
URL https://www.sciencedirect.com/science/article/pii/S0045782521000694
[37] Y. S. Shimizu, K. J. Fidkowski, Output-based error estimation for chaotic flows using reduced-order modeling, AIAA Paper 2018–0826 (2018). doi:10.2514/6.2018-0826.
[38] K. J. Fidkowski, P. L. Roe, An entropy adjoint approach to mesh refinement, SIAM Journal on Scientific Computing 32 (3) (2010) 1261–1287. doi:10.1137/090759057.
[39] S. M. Kast, K. J. Fidkowski, P. L. Roe, An unsteady entropy adjoint approach for adaptive solution of the shallow-water equations, AIAA Paper 2011-3694 (2011). doi:10.2514/6.2011-3694.
[40] F. Bassi, A. Colombo, A. Crivellini, K. J. Fidkowski, M. Franciolini, A. Ghidoni, G. Noventa, An entropy-adjoint p-adaptive discontinuous Galerkin method for the under-resolved simulation of turbulent flows, AIAA Journaldoi:10.2514/1.J058847.
[41] K. J. Fidkowski, M. A. Ceze, P. L. Roe, Entropy-based drag error estimation and mesh adaptation in two dimensions, AIAA Journal of Aircraft 49 (5) (2012) 1485–1496. doi:10.2514/1.C031795.
[42] E. J. Parish, K. Duraisamy, A paradigm for data-driven predictive modeling using field inversion and machine learning, Journal of Computational Physics 305 (2016) 758–774. doi:10.1016/j.jcp.2015.11.012.
URL https://www.sciencedirect.com/science/article/pii/S0021999115007524
[43] A. P. Singh, S. Medida, K. Duraisamy, Machine-learning-augmented predictive modeling of turbulent separated flows over airfoils, AIAA Journal 55 (7) (2017) 2215–2227. doi:10.2514/1.j055595.
[44] A. P. Singh, S. Pan, K. Duraisamy, Characterizing and improving predictive accuracy in shock-turbulent boundary layer interactions using data-driven models, AIAA Paper 2017–0314 (2017). doi:10.2514/6.2017-0314.
[45] J. R. Holland, J. D. Baeder, K. Duraisamy, Towards integrated field inversion and machine learning with embedded neural networks for RANS modeling, AIAA Paper 2019-1884 (2019). doi:10.2514/6.2019-1884.
[46] J. Ho, A. West, Field inversion and machine learning for turbulence modelling applied to three-dimensional separated flows, AIAA Paper 2021–2903 (2021). doi:https://doi.org/10.2514/6.2021-2903.
[47] C. Yan, H. Li, Y. Zhang, H. Chen, Data-driven turbulence modeling in separated flows considering physical mechanism analysis (2021). arXiv:2109.09095.
[48] F. Jäckel, A closed-form correction for the Spalart-Allmaras turbulence model for separated flow, AIAA Paper 2022–0462 (2022). doi:10.2514/6.2022-0462.
[49] P. R. Spalart, S. R. Allmaras, A one-equation turbulence model for aerodynamic flows, La Recherche Aérospatiale (1) (1994) 5–21.
[50] S. Allmaras, F. Johnson, P. Spalart, Modifications and clarifications for the implementation of the Spalart-Allmaras turbulence model, Seventh International Conference on Computational Fluid Dynamics (ICCFD7) 1902 (2012).
[51] M. A. Ceze, K. J. Fidkowski, High-order output-based adaptive simulations of turbulent flow in two dimensions, AIAA Paper 2015–1532 (2015). doi:10.2514/6.2015-1532.
[52] K. J. Fidkowski, Three-dimensional benchmark RANS computations using discontinuous finite elements on solution-adapted meshes, AIAA Paper 2018–1104 (2018). doi:10.2514/6.2018-1104.
[53] P. R. Spalart, Detached-eddy simulation, Annual Review of Fluid Mechanics 41 (2009) 181 – 202. doi:10.1146/annurev.fluid.010908.165130.
[54] W. Reed, T. Hill, Triangular mesh methods for the neutron transport equation, Los Alamos Scientific Laboratory Technical Report LA-UR-73-479 (1973).
[55] B. Cockburn, C.-W. Shu, Runge-Kutta discontinuous Galerkin methods for convection-dominated problems, Journal of Scientific Computing 16 (3) (2001) 173–261. doi:https://doi.org/10.1023/A:1012873910884.
[56] K. J. Fidkowski, T. A. Oliver, J. Lu, D. L. Darmofal, *p*-Multigrid solution of high–order discontinuous Galerkin discretizations of the com-

pressible Navier-Stokes equations, Journal of Computational Physics 207 (2005) 92–113. doi:10.1016/j.jcp.2005.01.005.

[57] P. Roe, Approximate Riemann solvers, parameter vectors, and difference schemes, Journal of Computational Physics 43 (1981) 357–372. doi:https://doi.org/10.1016/0021-9991(81)90128-5.

[58] F. Bassi, S. Rebay, Numerical evaluation of two discontinuous Galerkin methods for the compressible Navier-Stokes equations, International Journal for Numerical Methods in Fluids 40 (2002) 197–207. doi:https://doi.org/10.1002/fld.338.

[59] K. J. Fidkowski, High-order output-based adaptive methods for steady and unsteady aerodynamics, in: H. Deconinck, R. Abgrall (Eds.), 37$^{\text{th}}$ Advanced CFD Lectures series; Von Karman Institute for Fluid Dynamics (December 9–12 2013), von Karman Institute for Fluid Dynamics, 2013.

[60] M. A. Ceze, K. J. Fidkowski, Constrained pseudo-transient continuation, International Journal for Numerical Methods in Engineering 102 (2015) 1683–1703. doi:10.1002/nme.4858.

[61] Y. Saad, M. H. Schultz, GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems, SIAM Journal on Scientific Computing 7 (3) (1986) 856–869. doi:https://doi.org/10.1137/0907058.

[62] P.-O. Persson, J. Peraire, Newton-GMRES preconditioning for discontinuous Galerkin discretizations of the Navier-Stokes equations, SIAM Journal on Scientific Computing 30 (6) (2008) 2709–2733. doi:https://doi.org/10.1137/070692108.

[63] J. Cash, The integration of stiff initial value problems in ODEs using modified extended backward differentiation formulae, Computers & mathematics with applications 9 (5) (1983) 645–657. doi:https://doi.org/10.1016/0898-1221(83)90122-0.

[64] D. C. Liu, J. Nocedal, On the limited memory BFGS method for large scale optimization, Mathematical Programming 45 (1989) 503 – 528. doi:10.1007/BF01589116.

[65] P. Spalart, M. L. Shur, On the sensitization of turbulence models to rotation and curvature, Aerospace Science and Technology 1 (5) (1997) 297–302. doi:10.1016/S1270-9638(97)90051-1.

[66] P. Werbos, Beyond regression: New tools for prediction and analysis in the behavioral science, Ph.D. thesis, Harvard University (1974).

[67] A. R. Barron, Approximation and estimation bounds for artificial neural networks, Machine Learning 14 (1) (1994) 115–133. doi:https://doi.org/10.1007/BF00993164.

[68] C. Rumsey, G. Coleman, L.Wang, In search of data-driven improvements to RANS models applied to separated flows, AIAA Paper 2022–0937 (2022). doi:10.2514/6.2022-0937.

[69] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, X. Zheng, TensorFlow: Large-scale machine learning on heterogeneous systems, software available from tensorflow.org (2015).
URL http://tensorflow.org/

[70] K. J. Fidkowski, Output error estimation strategies for discontinuous Galerkin discretizations of unsteady convection-dominated flows, International Journal for Numerical Methods in Engineering 88 (12) (2011) 1297–1322. doi:10.1002/nme.3224.

[71] K. J. Fidkowski, Output-based space-time mesh optimization for unsteady flows using continuous-in-time adjoints, Journal of Computational Physics 341 (15) (2017) 258–277. doi:10.1016/j.jcp.2017.04.005.

[72] M. Yano, An optimization framework for adaptive higher-order discretizations of partial differential equations on anisotropic simplex meshes, Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts (2012).

[73] M. Yano, D. Darmofal, An optimization-based framework for anisotropic simplex mesh adaptation, Journal of Computational Physics 231 (22) (2012) 7626–7649. doi:10.1016/j.jcp.2012.06.040.

[74] K. J. Fidkowski, A local sampling approach to anisotropic metric-based mesh optimization, AIAA Paper 2016–0835 (2016). doi:10.2514/6.2016-0835.

[75] H. Borouchaki, P. George, F. Hecht, P. Laug, E. Saltel, Mailleur bidimensionnel de Delaunay gouverné par une carte de métriques. Partie I: Algorithmes, INRIA-Rocquencourt, France. Tech Report No. 2741 (1995).