Adjoint-Based Adaptation of Large-Eddy Simulations using Dynamic Closures

Krzysztof J. Fidkowski*
University of Michigan, Ann Arbor, MI, 48188

This paper presents a method for performing output-based mesh adaptation for large-eddy simulations of turbulence. Instead of an unsteady adjoint, which is expensive to compute and requires non-trivial regularization, the method is based on the field-inversion and machine-learning approach to data-driven turbulence modeling. The data here come not from experiments but from statistics computed from unsteady forward simulations. The resulting trained turbulence models yield steady-state solutions that represent the time-averaged unsteady flowfields. Adjoints computed from these steady-state models yield the sensitivity information required for an adjoint-weighted residual error estimate and adaptive indicator. Each adaptive iteration then only requires one unsteady primal solution with minimal storage: average statistics and fine-space residual information. Combined with unstructured mesh optimization, the method drives unsteady outputs to accurate values in only a few adaptive iterations. The performance is demonstrated on two-dimensional, low Reynolds number airfoil simulations, and comparisons are made to other techniques, including uniform and residual-based adaptation.

I. Introduction

In simulations of turbulent flow, both modeling and discretization errors affect flowfields and outputs of interest. Estimating these errors is important for robust and efficient calculations: incorrect models or coarse meshes lead to inaccurate solutions, whereas over-refined meshes unnecessarily increase computational time. Furthermore, without quantified discretization error estimates, simulations lack robustness or rely on expensive convergence studies. This work focuses on discretization errors, while at the same time tackling turbulence modeling errors by using data from higher-fidelity unsteady simulations.

Of particular interest here are output-based discretization error estimates, as these are useful for applications that demand accuracy in output scalars and efficient meshes for their prediction. Much previous work exists in this area, mostly for steady [1–4] and deterministic unsteady [5–10] simulations. However, these methods are not directly applicable to unsteady turbulent flow simulations, for which the requisite unsteady adjoint problem is unstable [11].

Strategies for controlling discretization errors in unsteady turbulent flows have included time windowing [12–14], least-squares shadowing [11, 15–17], model reduction [18, 19], and entropy-adjoint methods [20]. However, these strategies lack robustness, e.g. due to arbitrary time windows or large discretization errors in model reduction, add intractable expense for practical problems, e.g. in the adjoint solution for least-squares shadowing, or do not target specific outputs of interest, e.g. in the entropy-adjoint approach.

An alternative, cheaper strategy for controlling output errors in unsteady turbulent flows is to appeal to steady-state models, such as the Reynolds-averaged Navier-Stokes (RANS) equations. Output-based error estimation and adaptation has already been applied to such models in many previous works [21–24], with the lingering caveat that modeling errors can outweigh discretization errors, in particular for off-design conditions such as low Reynolds numbers or flows with separation. Adapting RANS meshes to strict discretization error tolerances in such cases is thus of questionable value.

RANS modeling error is the impetus for unsteady methods, such as large-eddy simulations. Once an investment is made in unsteady simulations, RANS ostensibly has little value. However, in this work, for the purpose of error estimation and adaptation, an effort is made to return to RANS via corrections that tune it to the unsteady data. An approach for doing this already exists in the field inversion and machine learning (FIML) framework [25–31], which has been applied more broadly to incorporate multiple sources of data, including experiments, into improved turbulence models. A unique and enabling 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

*Professor, Department of Aerospace Engineering, AIAA Associate Fellow.

retrained at each adaptive iteration. Once a corrected RANS model is created to reproduce the statistical outputs of interest, its adjoint is used as the basis for error estimates in the unsteady simulation.

The resulting unsteady adaptive method requires one unsteady primal simulation per adaptive iteration, without state storage nor checkpointing. Only the time-averaged state and residual need to be stored. Following the unsteady simulation, field inversion and machine-learning training yield an equation-level correction to a RANS model, which 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.

The adaptive method is presented in the context of a discontinuous Galerkin finite-element discretization, which is reviewed briefly in Section II. Section III presents the field-inversion and machine-learning algorithm, with a novel objective function geared for output error estimation. Sections IV and V 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 VI presents adaptive results, and Section VII concludes with a summary and a discussion of future directions.

II. Equations and Discretization

A. Governing Equations

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 [32, 33]. A detailed exposition of the equations and closure relations of the model as used presently can be found in our previous work [33, 34].

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

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

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. Other terms appearing 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.

B. The Discontinuous Galerkin Method

The discontinuous Galerkin (DG) method [35–37] 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 of which the state $\mathbf{u}$ is spatially approximated by $\mathbf{u}_h$, a linear combination of basis functions, here polynomials of order p_e, where e indexes the element, in reference space. No continuity constraints are imposed on the approximations on adjacent elements. Multiplying Eqn. 1 by test functions in the same space as the solution and integrating by parts to couple elements via fluxes, here the Roe [38] convective flux and the second form of Bassi and Rebay (BR2) [39] for the viscous flux, yields the weak form. Choosing a polynomial basis then yields a system of ordinary differential equations,

$$\mathbf{M} \frac{d\mathbf{U}}{dt} + \mathbf{R}(\mathbf{U}) = \mathbf{0}, \tag{3}$$

where $\mathbf{U} \in \mathbb{R}^N$ is the discrete state vector, N is the total number of unknowns, $\mathbf{R}(\cdot) \in \mathbb{R}^N$ 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 is not present. The solver consists of a Newton-Raphson method with the generalized minimum residual (GMRES) [40] linear solver, preconditioned by an element-line Jacobi smoother with a coarse-level ($p = 1$) correction [37, 41]. For unsteady

simulations, we use a third-order modified extended backward difference formula [42] applied to the the semi-discrete form.

C. The Output Adjoint

The adjoint solution for a scalar output is the sensitivity of the output to residual source perturbations [1, 4]. Only steady adjoint solutions are required in the present work, and these are much cheaper to compute than unsteady adjoints. Linearizing the residual and scalar output, $J(\mathbf{U})$, yields a linear system for the adjoint coefficients, $\boldsymbol{\Psi} \in \mathbb{R}^N$,

$$\left(\frac{\partial \mathbf{R}}{\partial \mathbf{U}}\right)^T \boldsymbol{\Psi} + \left(\frac{\partial J}{\partial \mathbf{U}}\right)^T = \mathbf{0}, \tag{4}$$

This equation is solved using a transposed version of the 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. The adjoint, as a residual sensitivity, weights the adjoint to produce the output error.

III. Field Inversion and Machine Learning

An unsteady simulation produces a time history of discrete states, $\mathbf{U}(t)$, form which only an average unsteady state can be computed,

$$\bar{\mathbf{U}} = \frac{1}{T_f - T_i} \int_{T_i}^{T_f} \mathbf{U}(t)\, dt, \tag{5}$$

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. For the error estimate, we also store an average residual, as described in Section IV. The goal in field inversion and machine learning (FIML) is to solve an inverse problem for a correction field that makes the corrected turbulence model solution close to $\bar{\mathbf{U}}$, and then 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.

A. Field Inversion

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

$$\begin{aligned} \min_{\boldsymbol{\beta}} \quad & J^{\text{inv}} \equiv \mathcal{F}(\mathbf{U}(\boldsymbol{\beta})) + \frac{\gamma}{2}(\boldsymbol{\beta} - 1)^T \mathbf{M}(\boldsymbol{\beta} - 1) \\ \text{s.t.} \quad & \mathbf{R}(\mathbf{U}, \boldsymbol{\beta}) = \mathbf{0} \end{aligned} \tag{6}$$

where $\mathcal{F}(\mathbf{U}, \bar{\mathbf{U}})$ measures the error in the solution relative to the unsteady average. The additional regularization term in Eqn. 6 penalizes, in a continuous spatial integral sense, large deviations of the correction field from the nominal value of one. For the problems considered here, γ is set to 10^{-7}, and this term is never active in the objective – it is included for completeness in exposition in cases of ill-conditioned choices of $\mathcal{F}$.

The baseline error measure is an integral of the surface stress distribution error,

$$\mathcal{F}^{\text{dist}}(\mathbf{U}, \bar{\mathbf{U}}) = \frac{1}{2} \int_{\text{airfoil}} \|\boldsymbol{\sigma}(\mathbf{U}) \cdot \vec{n} - \boldsymbol{\sigma}(\bar{\mathbf{U}}) \cdot \vec{n}\|^2 \, ds, \tag{7}$$

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

In output error estimation, the adjoint-weighted residual may be active not only on the airfoil surface, but also in the domain interior, where inaccuracies could propagate and affect the output of interest. Another field inversion error

measure is then a domain-interior integral of the inaccuracy in the state. To mimic the adjoint-weighted residual, the present measure is a weighted error norm, in which the weight is based on an adjoint solution,

$$\mathcal{F}^{\mathrm{AWR}}(\mathbf{U},\bar{\mathbf{U}}) = \sum_e \frac{1}{2}\left(\mathbf{W}_e^T\left(\mathbf{U}_e(\beta)-\bar{\mathbf{U}}_e\right)\right)^2 \qquad \mathbf{W}_e^T \equiv \boldsymbol{\Psi}_e^T \frac{\partial \mathbf{R}_e}{\partial \mathbf{U}_e}. \tag{8}$$

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 adapted output and is computed about the converged state $\mathbf{U}(\beta)$. The elemental Jacobian matrix converts state errors to residuals, which are then weighted by the adjoint. In cases of a mismatch in the state rank between $\mathbf{U}$ an $\bar{\mathbf{U}}$, e.g. due to the use of an augmented system in the turbulence modeling, only the common state components are compared, so that adjoint components corresponding to any new equations are not used.

Using either of the two error measures, Eqn. 7 or Eqn. 8, the inverse problem is solved using a limited-memory Broyden–Fletcher–Goldfarb–Shanno (L-BFGS) algorithm [43], with a history of 10 updates for approximating the inverse Hessian. The required gradient of J^{inv} with respect to $\boldsymbol{\beta}$ is evaluated using an adjoint. The residual linearization with respect to $\boldsymbol{\beta}$ is calculated using finite differences, 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.

B. Machine Learning

While field inversion produces a single correction field for a particular mesh and solution approximation order, error estimation involves a comparison of solutions or outputs between different approximation spaces. This requires a model for how the correction field changes with the solution. The model is assumed to be local, which simplifies the calculations and does not preclude capturing convective propagation effects, as the correction enters at the residuals level.

In FIML, an artificial neural network is used as the model that maps local states and geometry information to the correction field [25]. Previous works have shown that FIML yields corrected turbulence models that can better match the high-fidelity target data [26, 28, 29, 31], although the generality of this approach is still in debate [44]. Fortunately, in the present work a general-purpose turbulence mode is not sought, as adaptation only requires a model that is accurate close to a given training solution.

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

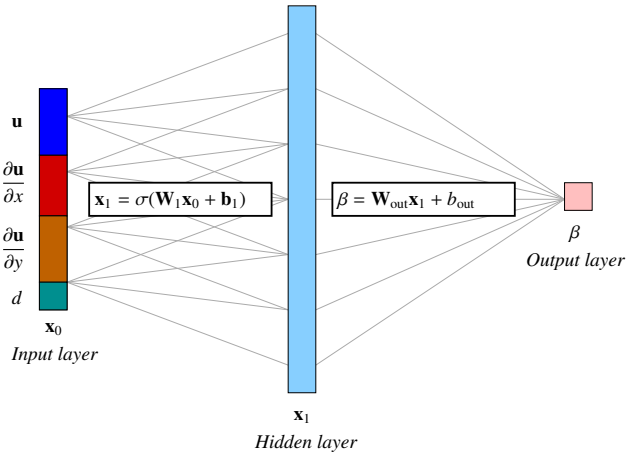


Fig. 1 Structure of the artificial neural networks used to predict the correction field.

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, its 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 [45], 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.

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 integration requirements when using the turbulence correction.

IV. Output-Based Error Estimation

A. Unsteady Output Perturbations

For a time-averaged output computed from the unsteady discrete state,

$$\bar{J} \equiv \frac{1}{T_f - T_i} \int_{T_i}^{T_f} J(\mathbf{U}(t))\, dt, \tag{9}$$

the unsteady adjoint, $\boldsymbol{\Psi}(t)$, arises from the Lagrangian

$$\mathcal{L} \equiv \bar{J} + \frac{1}{T_f - T_i} \int_{T_i}^{T_f} \boldsymbol{\Psi}(t)^{\mathrm{T}} \mathbf{R}^u(\mathbf{U}(t))\, dt, \tag{10}$$

where $\mathbf{R}^u$ is the unsteady residual. The unsteady adjoint-weighted residual [46–48] yields an estimate of the impact of a perturbation to the unsteady residual, $\delta\mathbf{R}^u(t)$, on the time-averaged output,

$$\delta\bar{J} \approx \frac{1}{T_f - T_i} \int_{T_i}^{T_f} \boldsymbol{\Psi}(t)^{\mathrm{T}} \delta\mathbf{R}^u(t)\, dt. \tag{11}$$

Assuming that the spatial error dominates the temporal error, the u superscript on $\mathbf{R}$ can be dropped. Decomposing the adjoint and residual perturbation into time-averaged and time-varying components,

$$\boldsymbol{\Psi}(t) = \bar{\boldsymbol{\Psi}} + \boldsymbol{\Psi}'(t), \quad \delta\mathbf{R}(t) = \delta\bar{\mathbf{R}} + \delta\mathbf{R}'(t), \tag{12}$$

Eqn. 11 becomes

$$\delta\bar{J} \approx \bar{\boldsymbol{\Psi}}^{\mathrm{T}} \delta\bar{\mathbf{R}} + \frac{1}{T_f - T_i} \int_{T_i}^{T_f} \boldsymbol{\Psi}'(t)^{\mathrm{T}} \delta\mathbf{R}'(t)\, dt. \tag{13}$$

When the time-varying components of the adjoint and residual perturbation are not strongly correlated, a reasonable assumption for the chaotic problems of interest in this work, the first term dominates, and this is the steady-state expression written using time-averaged quantities.

B. Augmented Systems

Suppose that a steady-state system of equations exists that extends the original unsteady states and equations with the purpose of modeling the time-averaged solution directly. The relevant case in this work is a (FIML-corrected) RANS model approximating the time-averaged state of a turbulent Navier-Stokes simulation. Denote the discretized form of this steady problem by

$$\widetilde{\mathbf{R}}(\widetilde{\mathbf{U}}) = \mathbf{0}, \tag{14}$$

where $\widetilde{\mathbf{U}} \in \mathbb{R}^{\widetilde{N}}$ is the augmented state vector, with $\widetilde{N} \geq N$. Restriction, $\mathbf{I}^r \in \mathbb{R}^{N \times \widetilde{N}}$, and prolongation, $\mathbf{I}^p \in \mathbb{R}^{\widetilde{N} \times N}$, operators map augmented states/residuals to original ones and vice versa. These are sparse matrices consisting of single 1's in select rows/columns and are never constructed explicitly in practice. The augmented state approximates the time-averaged solution of the original unsteady problem, i.e. $\bar{\mathbf{U}} \approx \mathbf{I}^r \widetilde{\mathbf{U}}$.

For the RANS problems of interest here, the augmented residual can be written as

$$\widetilde{\mathbf{R}}(\widetilde{\mathbf{U}}) = \begin{bmatrix} \mathbf{R}(\mathbf{I}^r \widetilde{\mathbf{U}}) + \mathbf{R}^{\text{aug}}(\widetilde{\mathbf{U}}) \\ \widetilde{\mathbf{R}}^{\text{aug}}(\widetilde{\mathbf{U}}), \end{bmatrix}, \tag{15}$$

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 (e.g. associated with the eddy-viscosity equation). Lastly, define the output of the augmented steady-state problem, and the associated adjoint, as

$$\widetilde{J}(\widetilde{\mathbf{U}}) \equiv J(\mathbf{I}^r \widetilde{\mathbf{U}}), \qquad \left(\frac{\partial \widetilde{\mathbf{R}}}{\partial \widetilde{\mathbf{U}}}\right)^T \widetilde{\mathbf{\Psi}} + \left(\frac{\partial \widetilde{J}}{\partial \widetilde{\mathbf{U}}}\right)^T = \mathbf{0}, \tag{16}$$

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

C. Error Estimate

In numerical error estimation, residual perturbations arise from differences in the resolution of the discretizations. A coarse solution, subscript H, will in general not satisfy the residual equations on a finer space, subscript h, where the finer space in this work is obtained by order enrichment. 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 the fine space. Evaluating the fine-space residual using the prolongated solution gives a residual perturbation,

$$\delta \mathbf{R}_h(t) = -\mathbf{R}_h(\mathbf{U}_h^H(t)) + \mathbf{M}_h \mathbf{I}_h^H \mathbf{M}_H^{-1} \mathbf{R}_H(\mathbf{U}_H(t)). \tag{17}$$

The negative sign on the first term is present because residual perturbations are added to the left-hand side of the discrete equations. The second term is necessary when only the spatial residual is used from Eqn. 3 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.

From Eqn. 13, the output error can be estimated by time-averaging the unsteady reasidual perturbation and weighting it by the time-averaged adjoint,

$$\delta \bar{J}_h \approx \delta \bar{\mathbf{\Psi}}_h^{\mathrm{T}} \delta \bar{\mathbf{R}}_h, \qquad \delta \bar{\mathbf{R}}_h \equiv \frac{1}{T_f - T_i} \int_{T_i}^{T_f} \delta \mathbf{R}_h(t)\, dt. \tag{18}$$

This expression uses the adjoint perturbation instead of the fine-space adjoint itself, in order to reduce errors arising from lack of strict Galerkin orthogonality in the residuals [10]. These perturbations are obtained by subtracting from the fine-space adjoint its projection onto the coarse space.

From the augmented residual decomposition in 15, the restricted augmented adjoint, $\mathbf{I}^r \delta \widetilde{\mathbf{\Psi}}_h$, weights perturbations in the original residual. This means that $\mathbf{I}^r \delta \widetilde{\mathbf{\Psi}}_h$ is the sensitivity of the output to perturbations in $\mathbf{R}_h$. It is therefore used as a surrogate for the time-averaged adjoint in Eqn. 18, $\delta \bar{\mathbf{\Psi}}_h \approx \mathbf{I}^r \delta \widetilde{\mathbf{\Psi}}_h$.

While 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 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 based on the new model that would not be appropriate for weighting the unsteady residuals of the original model.

The error estimate requires 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. An unsteady adjoint is not required for the error estimate.

V. Mesh Adaptation and Implementation

The error estimate in Eqn. 18 is calculated after an unsteady simulation, which provides the time-averaged residual and FIML training data. In order to adapt, the error estimate is localized to elements, as described below.

A. 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 DG, the localization to elements, e, yields an elemental error indicator

$$\mathcal{E}_e \equiv \left|\delta\boldsymbol{\Psi}_{he}^{\mathrm{T}}\delta\mathbf{R}_{he}\right|, \qquad \delta J^{\mathrm{cons}} \equiv \sum_e \mathcal{E}_e, \tag{19}$$

where the subscript e denotes degrees of freedom associated with element e. The above equation also defines 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. 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 [4].

B. Adaptive Strategies

Three adaptive strategies are considered 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 $f^{\mathrm{adapt}} = 0.2$ of elements with the highest error indicators is chosen for p refinement. In the output-based case, the error indicator comes directly from Eqn. 19. For comparison, an unweighted residual indicator is obtained by omitting the adjoint from Eqn. 19: $\mathcal{E}_e^{\mathrm{res}} \equiv \left|\delta\mathbf{R}_e\right|$.

The third adaptive strategy is mesh optimization through error sampling and synthesis, MOESS [49–51]. 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.

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 becomes expensive. However, even by taking just one MOESS adaptation iteration per unsteady run, rapid convergence to the optimal mesh has been observed, so that only 3-4 unsteady simulations are typically required. The number of adaptation iterations of course 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 [51] 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.

C. 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 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.

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. In addition, the length of each unsteady

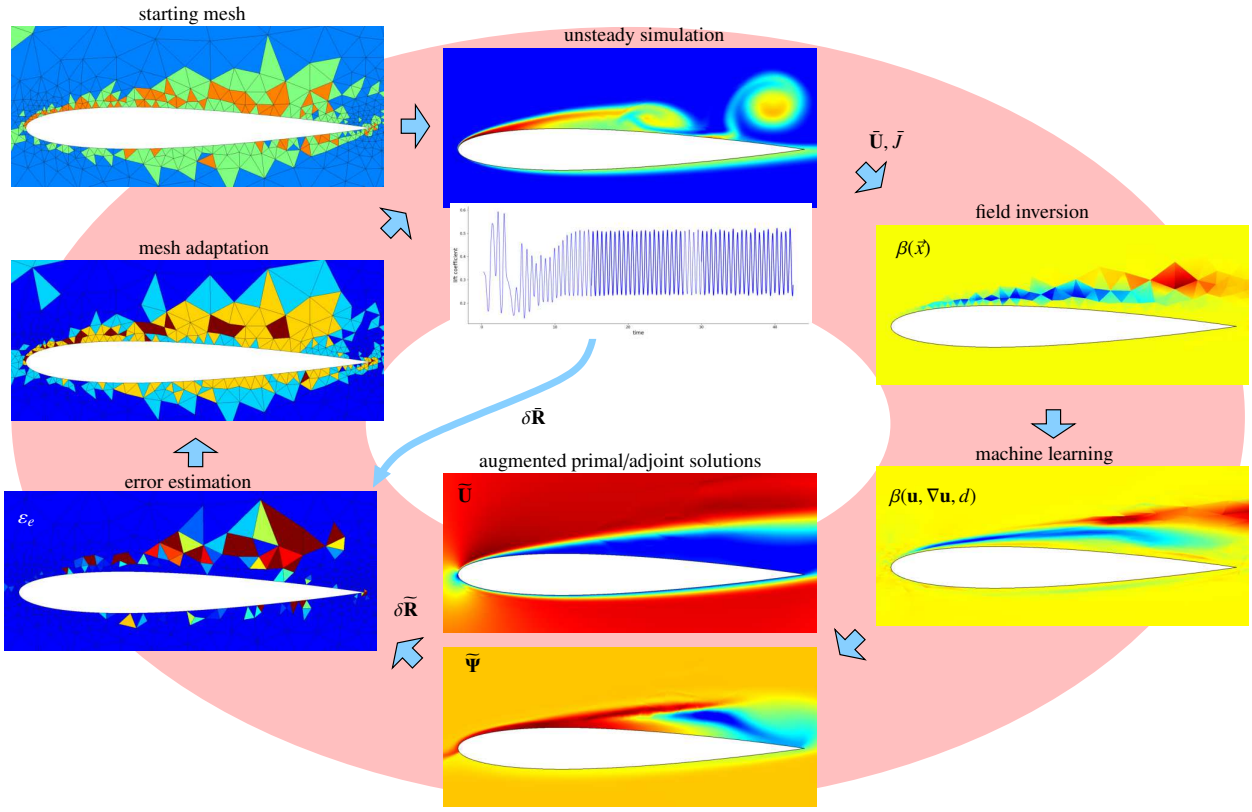


Fig. 2 Flowchart of the proposed unsteady mesh-adaptation iterations.

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 are topics for future work.

VI. Results

This section presents results for two unsteady aerodynamics problems. All solutions are obtained using $p = 2$ approximation order, except for the p refinement study. 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 a given problem, the temporal discretization remains fixed at a high resolution, determined empirically such that temporal errors do not pollute the results relative to spatial errors. For clarity, the adopted naming convention for the adaptive strategies is as follows.

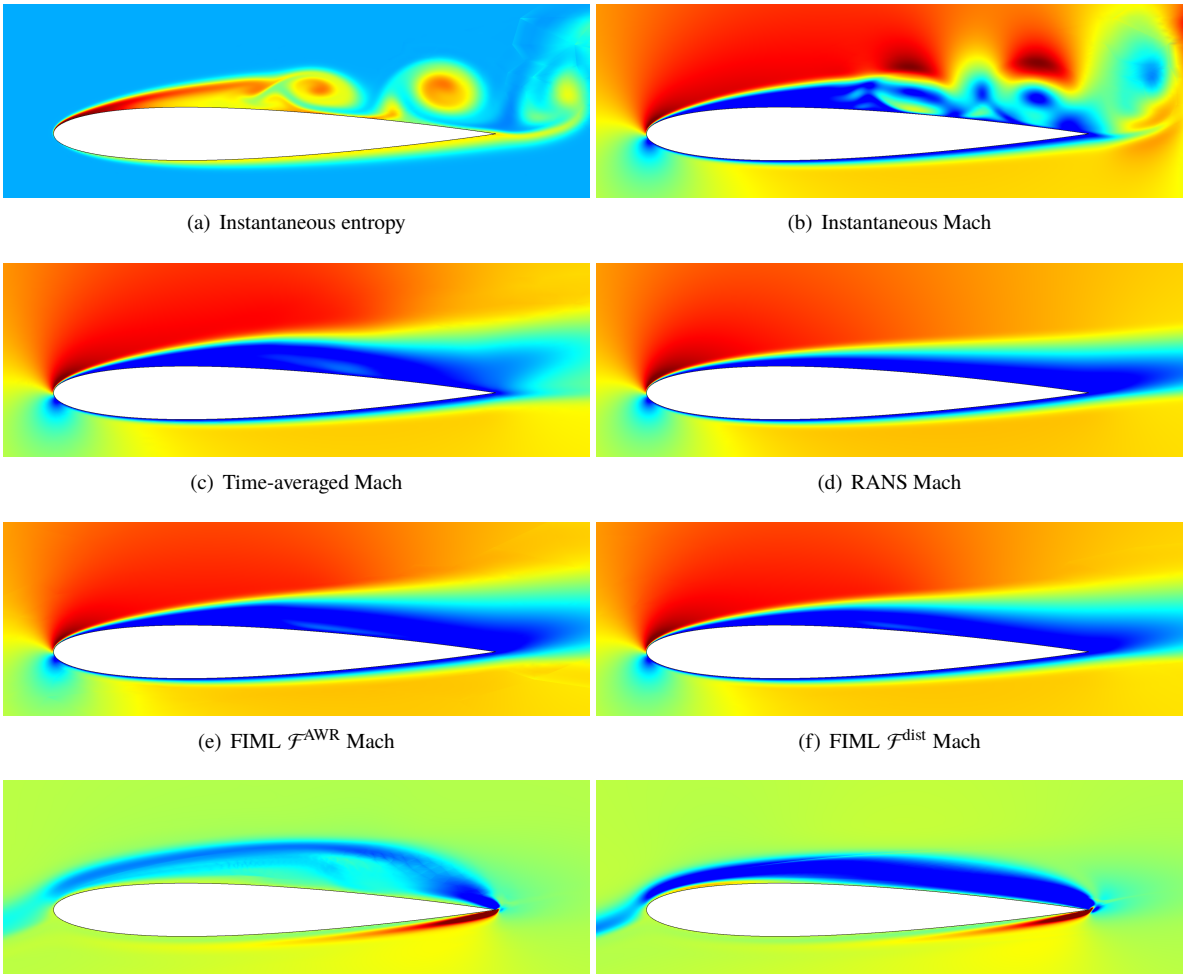
- **FIML MOESS [AWR]**: Mesh optimization at a given target `dof` using the FIML-based approach for the adjoint and field-inversion objective $\mathcal{F}^{\text{dist}}$ (default) or $\mathcal{F}^{\text{AWR}}$ (AWR).
- **RANS MOESS**: Mesh optimization at a given target `dof` using the baseline, uncorrected 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.

A. NACA 0012 Airfoil at $\alpha = 7°$, $Re = 10{,}000$

The first example considers flow governed by the compressible Navier-Stokes equations over a NACA 0012 airfoil, at Mach number $M = 0.2$, angle of attack $\alpha = 7°$, 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 fowfield 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.



(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

Fig. 3 Instantaneous and time-averaged entropy, Mach number, and adjoint contours for flow over a NACA 0012 airfoil at $M = 0.2, Re = 1 \times 10^4, \alpha = 7°$.

The error estimates and adaptive strategies, including output and residual-based methods and uniform refinement in h and p, are now compared. 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 4 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, Eqn. 19. 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.

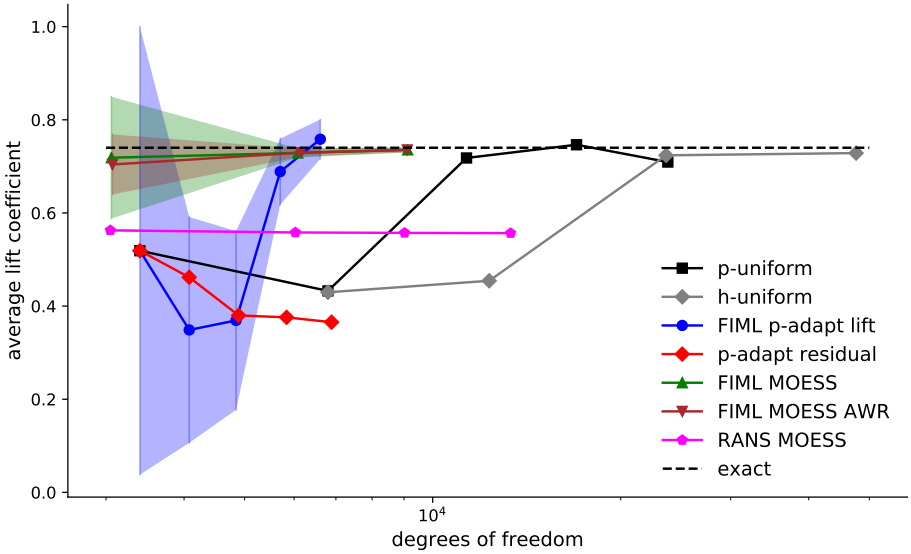


Fig. 4 Lift coefficient convergence histories for various adaptive methods applied to the NACA 0012 airfoil at $M = 0.2, Re = 10^4, \alpha = 7°$.

Several observations can be made from Figure 4. 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 the 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 5 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 resolve any unsteady behavior, and hence the initial flat output history. Note that the initial mesh only has 1133 elements. The green bars at the end of each unsteady run represent the $\pm\delta J^{\text{cons}}$ error estimates, obtained using the adjoint-weighted residual. The estimates are reasonably accurate, and adaptation improves the average output with only localized high-order refinement.

In comparison to lift-based p-adaptation, residual-based p-adaptation does not perform very well, as observed in the results shown in Figure 4. 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

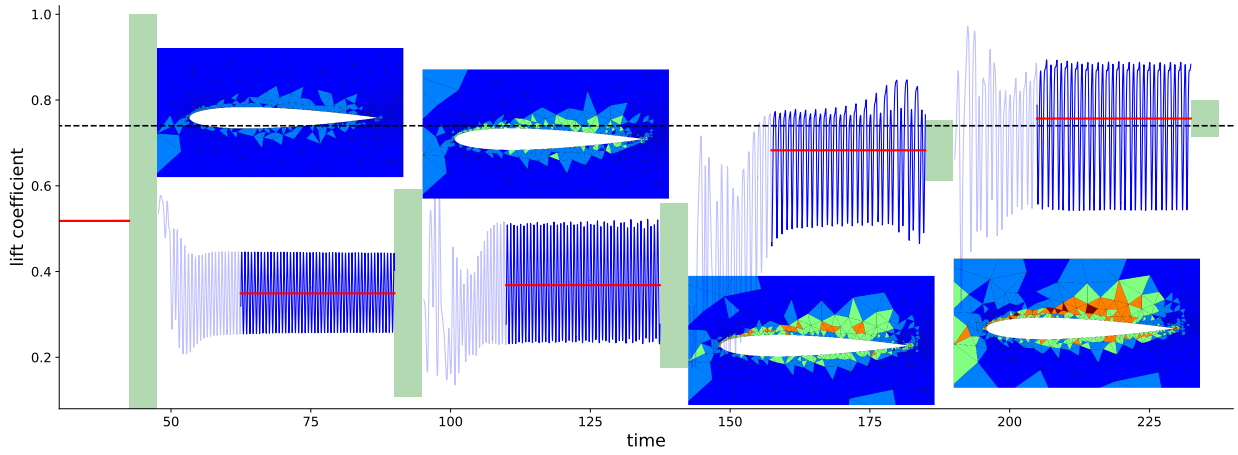
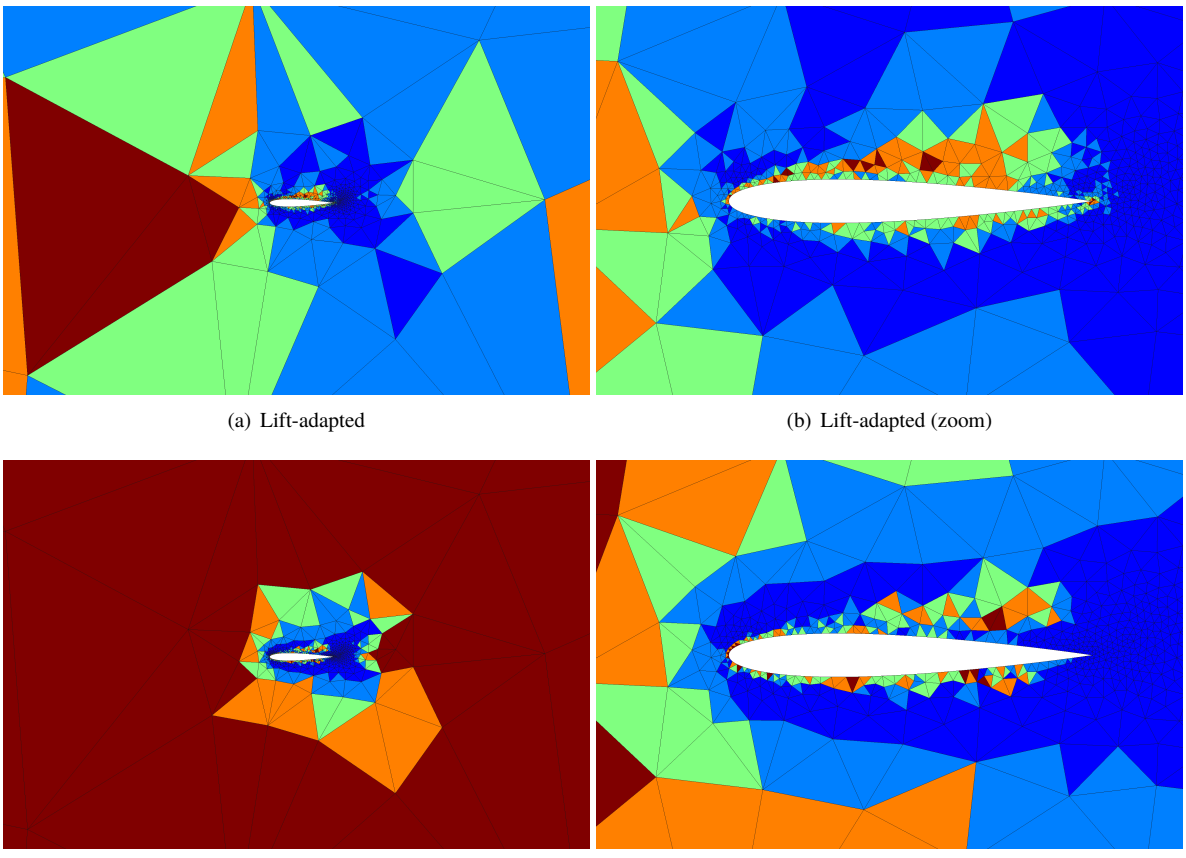


Fig. 5 p**-adaptation history for the NACA 0012 airfoil at** $M = 0.2, Re = 10^4, \alpha = 7°$**. The order field range is** $p = 1$ **(blue) to** $p = 5$ **(red). Red lines denote time-average outputs, and the green bars are error estimates at** $\pm\delta J^{\text{cons}}$**.**

this case.



(a) Lift-adapted

(b) Lift-adapted (zoom)

(c) Residual-adapted

(d) Residual-adapted (zoom)

Fig. 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** $M = 0.2, Re = 1 \times 10^4, \alpha = 7°$**.**

Turning to FIML MOESS, Figure 7 shows the output history for lift adaptation at 6000 `dof`. 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.

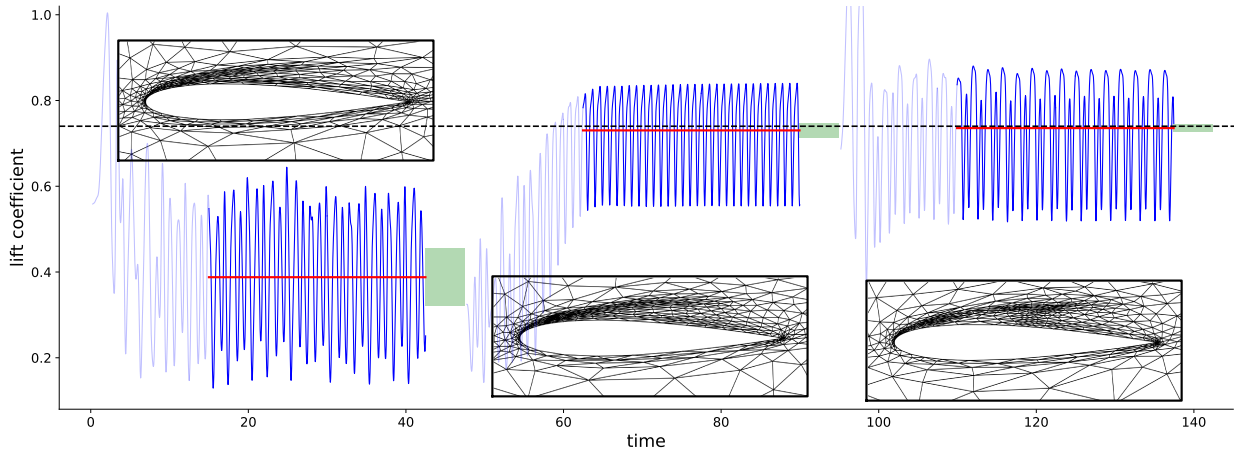


Fig. 7 FIML MOESS adaptation history using 6000 `dof` **for the NACA 0012 airfoil at** $M = 0.2, Re = 10^4, \alpha = 7°$**. Red lines denote time-averaged outputs, and the green bars are error estimates at** $\pm\delta J^{\text{cons}}$**.**

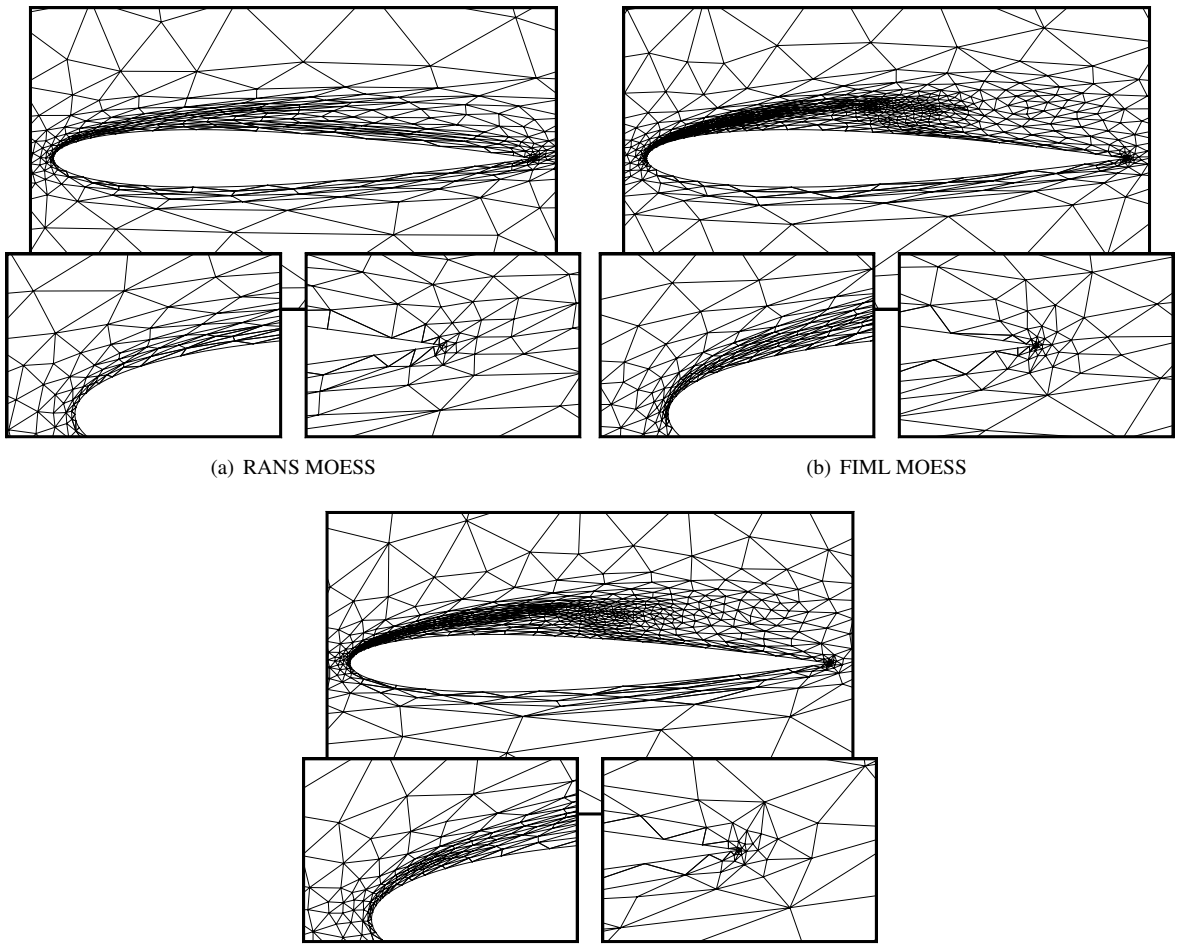
Table 1 presents the outputs, error estimates, and actual errors for various FIML MOESS adaptive simulations. The outputs and error estimates were obtained by averaging the results of the latter half of the MOESS iterations. We see a general trend of decreasing actual errors and improved error estimate efficacy with increasing degrees of freedom. Comparing the two FIML inversion error measures, 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.

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`	FIML objective	Output	Error Estimate	$\lvert$Actual Error$\rvert$
3000	$\mathcal{F}^{\text{dist}}$	0.7186	0.1309	0.0214
3000	$\mathcal{F}^{\text{AWR}}$	0.7040	0.0645	0.0360
6000	$\mathcal{F}^{\text{dist}}$	0.7288	0.0099	0.0112
6000	$\mathcal{F}^{\text{AWR}}$	0.7294	0.0087	0.0106
9000	$\mathcal{F}^{\text{dist}}$	0.7355	0.0057	0.0045
9000	$\mathcal{F}^{\text{AWR}}$	0.7353	0.0051	0.0047

Finally, Figure 8 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 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 half-way 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.



(a) RANS MOESS

(b) FIML MOESS

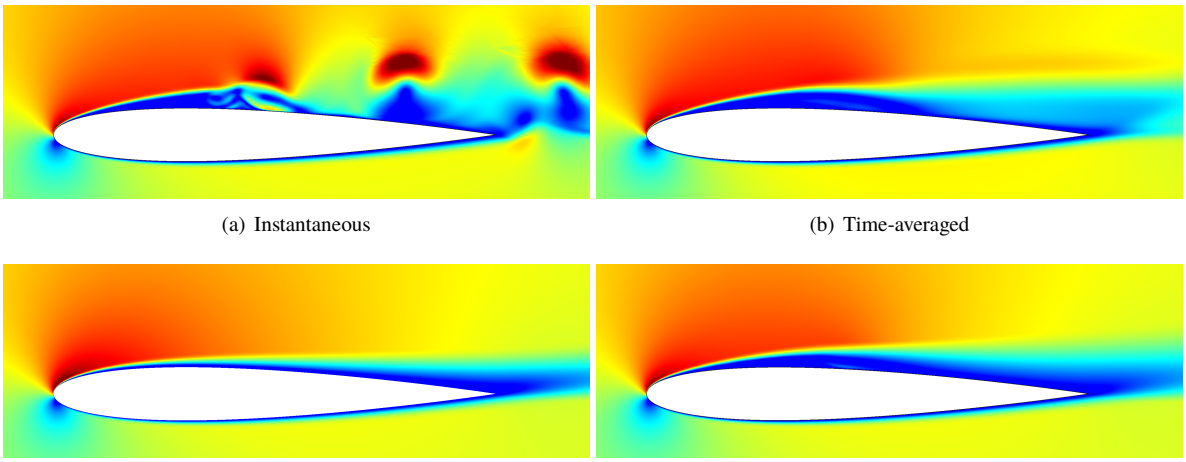
(c) FIML MOESS AWR

Fig. 8 Various adapted meshes at 9000 `dof` **for the NACA 0012 airfoil at** $M = 0.2, Re = 10^4, \alpha = 7°$**.**

B. NACA 0012 Airfoil at $\alpha = 7°$**,** $Re = 20{,}000$

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 9, 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.

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 10 shows the convergence results. In this case, the RANS prediction of the lift is closer to the exact value 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,



(a) Instantaneous (b) Time-averaged

(c) RANS (d) FIML

Fig. 9 Instantaneous and time-averaged Mach number contours for flow over a NACA 0012 airfoil at $M = 0.2, Re = 2 \times 10^4, \alpha = 7°$.

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.

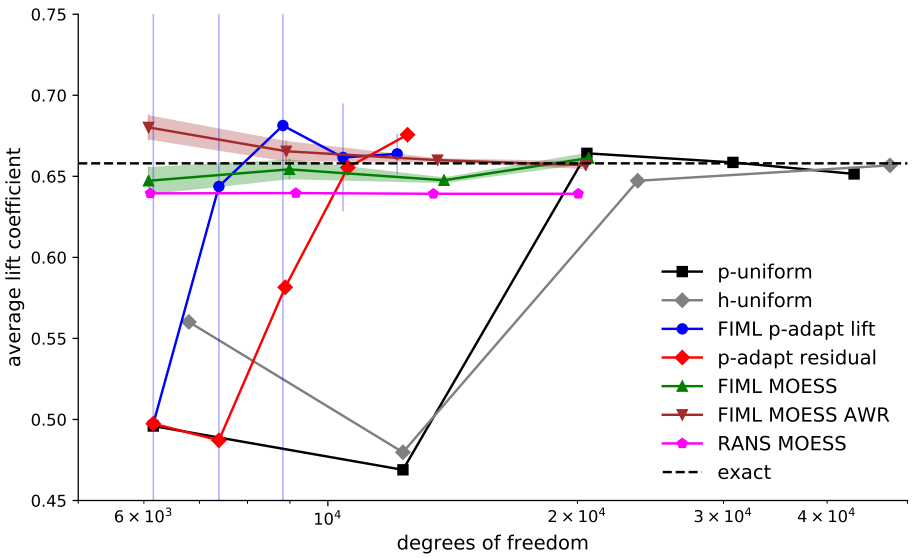


Fig. 10 Lift coefficient convergence histories for various adaptive methods applied to the NACA 0012 airfoil at $M = 0.2, Re = 2 \times 10^4, \alpha = 7°$.

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 results. 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 adaptive iteration it appears to be overshooting the exact value instead of converging to it. Uniform h and p refinement converge to the exact value, but with more degrees of freedom compared to the adaptive methods.

Figure 11 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.

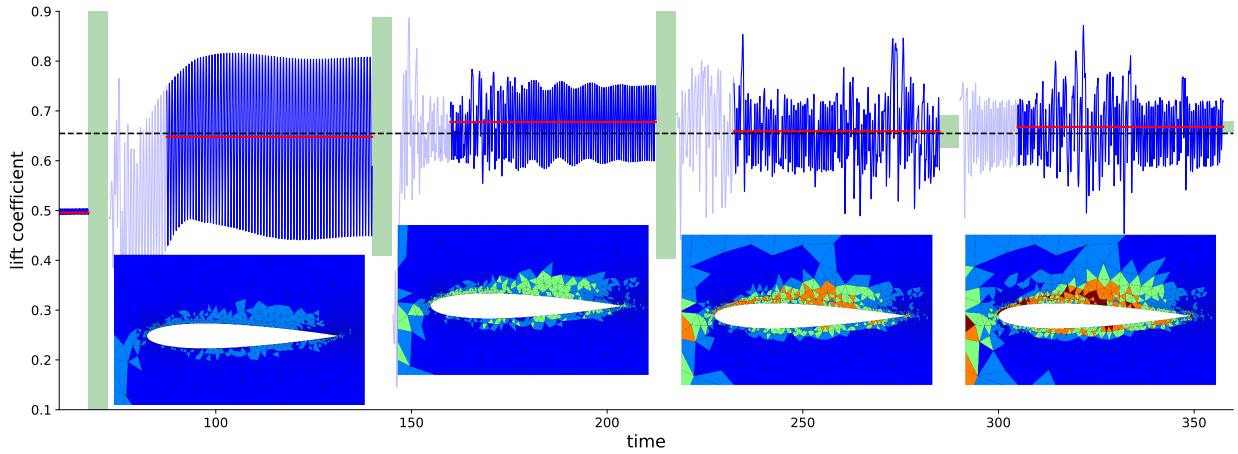


Fig. 11 p-adaptation history for the NACA 0012 airfoil at $M = 0.2, Re = 2 \times 10^4, \alpha = 7°$. The order field range is $p = 1$ (blue) to $p = 5$ (red). Red lines denote time-averaged outputs, and the green bars are error estimates at $\pm\delta J^{\text{cons}}$.

Figure 12 shows the time history of the adaptation using FIML MOESS 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.

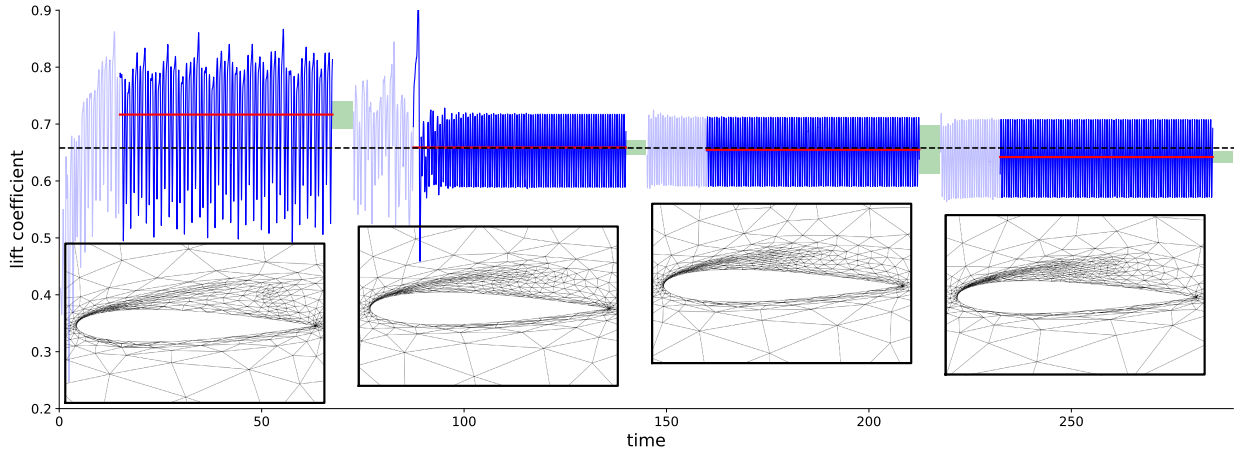


Fig. 12 FIML MOESS adaptation history using 6000 `dof` for the NACA 0012 airfoil at $M = 0.2, Re = 2 \times 10^4, \alpha = 7°$. Red lines denote time-averaged outputs, and the green bars are error estimates at $\pm\delta J^{\text{cons}}$.

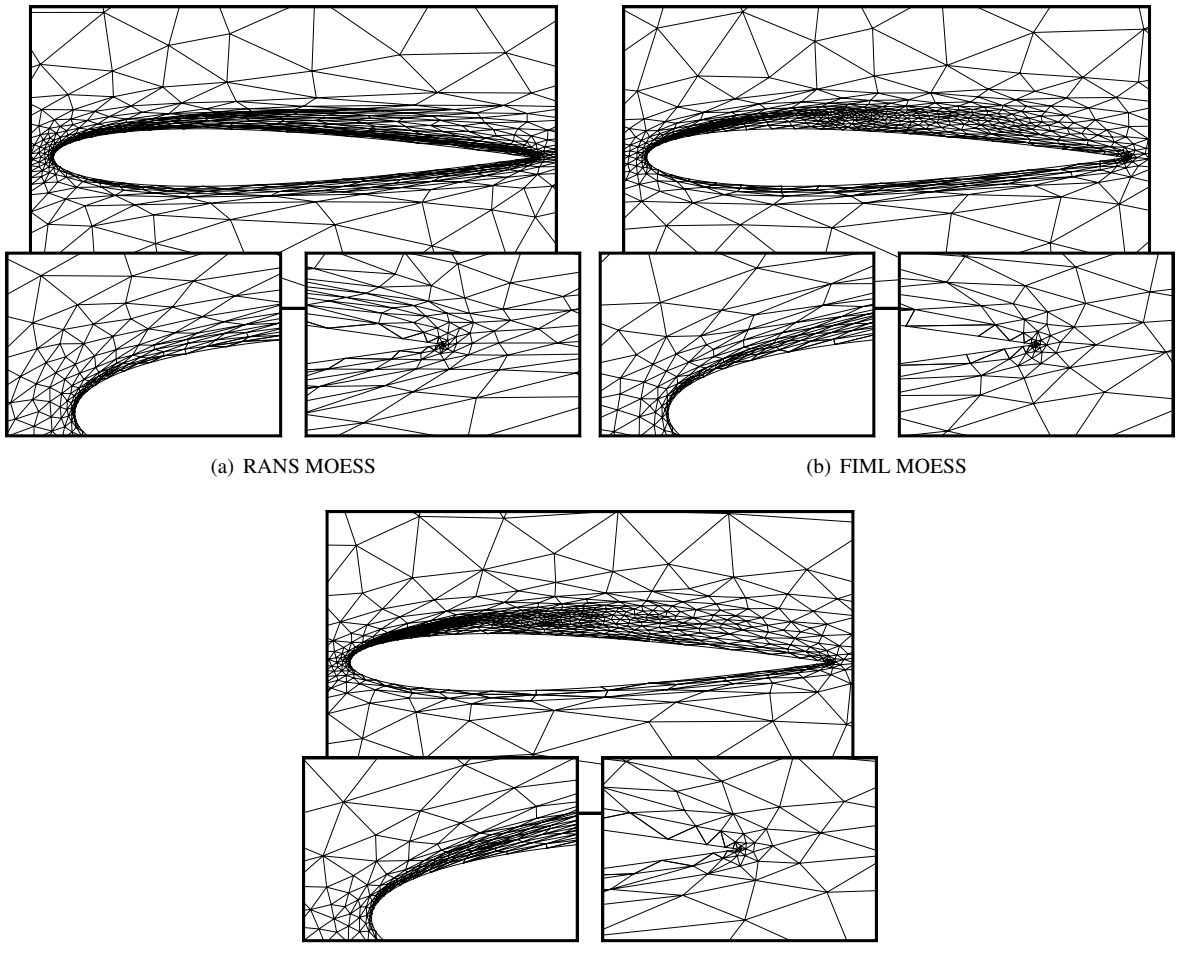
Table 2 presents the average outputs, error estimates, and actual errors for the FIML MOESS adaptation strategies at various target `dof`. We see a general trend of a reduction in both the estimated and actual error with increasing `dof`. With the exception of the coarsest mesh, which is arguably under-resolved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

Finally, Figure 13 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 9. The RANS mesh places anisotropic elements inside and on the edge of the boundary layers, which is expected due to the slow variation of

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`	FIML objective	Output	Error Estimate	$\lvert$Actual Error$\rvert$
6000	$\mathcal{F}^{\text{dist}}$	0.6473	0.0085	0.0107
6000	$\mathcal{F}^{\text{AWR}}$	0.6802	0.0075	0.0222
9000	$\mathcal{F}^{\text{dist}}$	0.6543	0.0061	0.0037
9000	$\mathcal{F}^{\text{AWR}}$	0.6655	0.0063	0.0075
13500	$\mathcal{F}^{\text{dist}}$	0.6476	0.0018	0.0104
13500	$\mathcal{F}^{\text{AWR}}$	0.6600	0.0012	0.0020
20250	$\mathcal{F}^{\text{dist}}$	0.6613	0.0034	0.0033
20250	$\mathcal{F}^{\text{AWR}}$	0.6569	0.0024	0.0011

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 regions affects behavior of the flow over the rest of the airfoil.



(a) RANS MOESS

(b) FIML MOESS

(c) FIML MOESS AWR

Fig. 13 Various adapted meshes at 9000 `dof` **for the NACA 0012 airfoil at $M = 0.2, Re = 2 \times 10^4, \alpha = 7°$.**

VII. 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 form 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] Becker, R., and Rannacher, R., "An optimal control approach to a posteriori error estimation in finite element methods," *Acta Numerica*, edited by A. Iserles, Cambridge University Press, 2001, pp. 1–102.

[2] Venditti, D. A., and Darmofal, D. L., "Anisotropic grid adaptation for functional outputs: application to two-dimensional viscous flows," *Journal of Computational Physics*, Vol. 187, No. 1, 2003, pp. 22–46.

[3] Nemec, M., and Aftosmis, M. J., "Error Estimation and Adaptive Refinement for Embedded-Boundary Cartesian Meshes," AIAA Paper 2007-4187, 2007.

[4] Fidkowski, K. J., and Darmofal, D. L., "Review of Output-Based Error Estimation and Mesh Adaptation in Computational Fluid Dynamics," *AIAA Journal*, Vol. 49, No. 4, 2011, pp. 673–694. https://doi.org/10.2514/1.J050073.

[5] Meidner, D., and Vexler, B., "Adaptive Space-Time Finite Element Methods for Parabolic Optimization Problems," *SIAM Journal on Control Optimization*, Vol. 46, No. 1, 2007, pp. 116–142. https://doi.org/https://doi.org/10.1137/060648994.

[6] Barth, T. J., "Space-Time Error Representation and Estimation in Navier-Stokes Calculations," *Complex Effects in Large Eddy Simulations*, edited by S. C. Kassinos, C. A. Langer, G. Iaccarino, and P. Moin, Springer Berlin Heidelberg, Lecture Notes in Computational Science and Engineering Vol 26, 2007, pp. 29–48.

[7] Mani, K., and Mavriplis, D. J., "Error Estimation and Adaptation for Functional Outputs in Time-Dependent Flow Problems," *Journal of Computational Physics*, Vol. 229, 2010, pp. 415–440. https://doi.org/https://doi.org/10.1016/j.jcp.2009.09.034.

[8] Belme, A., Dervieux, A., and Alauzet, F., "Time accurate anisotropic goal-oriented mesh adaptation for unsteady flows," *Journal of Computational Physics*, Vol. 231, No. 19, 2012, pp. 6323–6348. https://doi.org/10.1016/j.jcp.2012.05.003.

[9] Fidkowski, K. J., and Luo, Y., "Output-based Space-Time Mesh Adaptation for the Compressible Navier-Stokes Equations," *Journal of Computational Physics*, Vol. 230, 2011, pp. 5753–5773. https://doi.org/10.1016/j.jcp.2011.03.059.

[10] Kast, S. M., and Fidkowski, K. J., "Output-based Mesh Adaptation for High Order Navier-Stokes Simulations on Deformable Domains," *Journal of Computational Physics*, Vol. 252, No. 1, 2013, pp. 468–494. https://doi.org/10.1016/j.jcp.2013.06.007.

[11] Wang, Q., "Forward and adjoint sensitivity computation of chaotic dynamical systems," *Journal of Computational Physics*, Vol. 235, 2013, pp. 1–13. https://doi.org/https://doi.org/10.1016/j.jcp.2012.09.007.

[12] Krakos, J. A., Wang, Q., Hall, S. R., and Darmofal, D. L., "Sensitivity analysis of limit cycle oscillations," *Journal of Computational Physics*, Vol. 231, No. 8, 2012, pp. 3228–3245. https://doi.org/https://doi.org/10.1016/j.jcp.2012.01.001.

[13] Shimizu, Y. S., and Fidkowski, K. J., "Output Error Estimation for Chaotic Flows," AIAA Paper 2016-3806, 2016. https://doi.org/https://doi.org/10.2514/6.2016-3806.

[14] Chung, S. W., and Freund, J. B., "An optimization method for chaotic turbulent flow," *Journal of Computational Physics*, Vol. 457, 2022, p. 111077. https://doi.org/https://doi.org/10.1016/j.jcp.2022.111077.

[15] Wang, Q., Hu, R., and Blonigan, P., "Least Squares Shadowing sensitivity analysis of chaotic limit cycle oscillations," *Journal of Computational Physics*, Vol. 267, 2014, pp. 210–224. https://doi.org/https://doi.org/10.1016/j.jcp.2014.03.002.

[16] Wang, Q., "Convergence of the least squares shadowing method for computing derivative of ergodic averages," *SIAM Journal on Numerical Analysis*, Vol. 52, No. 1, 2014, pp. 156–170. https://doi.org/https://doi.org/10.1137/130917065.

[17] Blonigan, P. J., Gomez, S. A., and Wang, Q., "Least Squares Shadowing for Sensitivity Analysis of Turbulent Fluid Flows," AIAA Paper 2014–1426, 2014. https://doi.org/https://doi.org/10.2514/6.2014-1426.

[18] Li, X., Hulshoff, S., and Hickel, S., "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*, Vol. 379, 2021, p. 113733. https://doi.org/https://doi.org/10.1016/j.cma.2021.113733, URL https://www.sciencedirect.com/science/article/pii/S0045782521000694.

[19] Shimizu, Y. S., and Fidkowski, K. J., "Output-Based Error Estimation for Chaotic Flows Using Reduced-Order Modeling," AIAA Paper 2018–0826, 2018. https://doi.org/10.2514/6.2018-0826.

[20] Bassi, F., Colombo, A., Crivellini, A., Fidkowski, K. J., Franciolini, M., Ghidoni, A., and Noventa, G., "An entropy-adjoint p-adaptive discontinuous Galerkin method for the under-resolved simulation of turbulent flows," *AIAA Journal*, 2020. https://doi.org/10.2514/1.J058847.

[21] Oliver, T. A., "A High–order, Adaptive, Discontinuous Galerkin, Finite Elemenet Method for the Reynolds-Averaged Navier-Stokes Equations," Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts, 2008.

[22] Hartmann, R., Held, J., and Leicht, T., "Adjoint-based error estimation and adaptive mesh refinement for the RANS and $k - \omega$ turbulence model equations," *Journal of Computational Physics*, Vol. 230, No. 11, 2011, pp. 4268 – 4284. https://doi.org/10.1016/j.jcp.2010.10.026.

[23] Yano, M., Modisette, J., and Darmofal, D., "The Importance of mesh adaptation for higher-order discretizations of aerodynamics flows," AIAA Paper 2011-3852, 2011.

[24] Ceze, M. A., and Fidkowski, K. J., "High-order output-based adaptive simulations of turbulent flow in two dimensions," *AIAA Journal*, Vol. 54, No. 9, 2016. https://doi.org/10.2514/1.J054517.

[25] Parish, E. J., and Duraisamy, K., "A paradigm for data-driven predictive modeling using field inversion and machine learning," *Journal of Computational Physics*, Vol. 305, 2016, pp. 758–774. https://doi.org/10.1016/j.jcp.2015.11.012, URL https://www.sciencedirect.com/science/article/pii/S0021999115007524.

[26] Singh, A. P., Medida, S., and Duraisamy, K., "Machine-learning-augmented predictive modeling of turbulent separated flows over airfoils," *AIAA Journal*, Vol. 55, No. 7, 2017, pp. 2215–2227. https://doi.org/10.2514/1.j055595.

[27] Singh, A. P., Pan, S., and Duraisamy, K., "Characterizing and Improving Predictive Accuracy in Shock-Turbulent Boundary Layer Interactions Using Data-driven Models," AIAA Paper 2017–0314, 2017. https://doi.org/10.2514/6.2017-0314.

[28] Holland, J. R., Baeder, J. D., and Duraisamy, K., "Towards Integrated Field Inversion and Machine Learning With Embedded Neural Networks for RANS Modeling," AIAA Paper 2019-1884, 2019. https://doi.org/10.2514/6.2019-1884.

[29] Ho, J., and West, A., "Field Inversion and Machine Learning for turbulence modelling applied to three-dimensional separated flows," AIAA Paper 2021–2903, 2021. https://doi.org/https://doi.org/10.2514/6.2021-2903.

[30] Yan, C., Li, H., Zhang, Y., and Chen, H., "Data-driven turbulence modeling in separated flows considering physical mechanism analysis," , 2021.

[31] Jäckel, F., "A Closed-form Correction for the Spalart-Allmaras Turbulence Model for Separated Flow," AIAA Paper 2022–0462, 2022. https://doi.org/10.2514/6.2022-0462.

[32] Allmaras, S., Johnson, F., and Spalart, P., "Modifications and Clarifications for the Implementation of the Spalart-Allmaras Turbulence Model," Seventh International Conference on Computational Fluid Dynamics (ICCFD7) 1902, 2012.

[33] Ceze, M. A., and Fidkowski, K. J., "High-Order Output-Based Adaptive Simulations of Turbulent Flow in Two Dimensions," AIAA Paper 2015–1532, 2015. https://doi.org/10.2514/6.2015-1532.

[34] Fidkowski, K. J., "Three-Dimensional Benchmark RANS Computations Using Discontinuous Finite Elements on Solution-Adapted Meshes," AIAA Paper 2018–1104, 2018. https://doi.org/10.2514/6.2018-1104.

[35] Reed, W., and Hill, T., "Triangular Mesh Methods for the Neutron Transport Equation," Los Alamos Scientific Laboratory Technical Report LA-UR-73-479, 1973.

[36] Cockburn, B., and Shu, C.-W., "Runge-Kutta discontinuous Galerkin methods for convection-dominated problems," *Journal of Scientific Computing*, Vol. 16, No. 3, 2001, pp. 173–261. https://doi.org/https://doi.org/10.1023/A:1012873910884.

[37] Fidkowski, K. J., Oliver, T. A., Lu, J., and Darmofal, D. L., "*p*-Multigrid solution of high–order discontinuous Galerkin discretizations of the compressible Navier-Stokes equations," *Journal of Computational Physics*, Vol. 207, 2005, pp. 92–113. https://doi.org/10.1016/j.jcp.2005.01.005.

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

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

[40] Saad, Y., and Schultz, M. H., "GMRES: A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems," *SIAM Journal on Scientific Computing*, Vol. 7, No. 3, 1986, pp. 856–869. https://doi.org/https://doi.org/10.1137/0907058.

[41] Persson, P.-O., and Peraire, J., "Newton-GMRES Preconditioning for Discontinuous Galerkin Discretizations of the Navier-Stokes Equations," *SIAM Journal on Scientific Computing*, Vol. 30, No. 6, 2008, pp. 2709–2733. https://doi.org/https://doi.org/10.1137/070692108.

[42] Cash, J., "The integration of stiff initial value problems in ODEs using modified extended backward differentiation formulae," *Computers & mathematics with applications*, Vol. 9, No. 5, 1983, pp. 645–657. https://doi.org/https://doi.org/10.1016/0898-1221(83)90122-0.

[43] Liu, D. C., and Nocedal, J., "On the Limited Memory BFGS Method for Large Scale Optimization," *Mathematical Programming*, Vol. 45, 1989, pp. 503 – 528. https://doi.org/10.1007/BF01589116.

[44] Rumsey, C., Coleman, G., and L.Wang, "In Search of Data-Driven Improvements to RANS Models Applied to Separated Flows," AIAA Paper 2022–0937, 2022. https://doi.org/10.2514/6.2022-0937.

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

[46] Fidkowski, K. J., "Output error estimation strategies for discontinuous Galerkin discretizations of unsteady convection-dominated flows," *International Journal for Numerical Methods in Engineering*, Vol. 88, No. 12, 2011, pp. 1297–1322. https://doi.org/10.1002/nme.3224.

[47] Luo, Y., and Fidkowski, K. J., "Output-based space time mesh adaptation for unsteady aerodynamics," AIAA Paper 2011-491, 2011. https://doi.org/10.2514/6.2011-491.

[48] Fidkowski, K. J., "Output-Based Space-Time Mesh Optimization for Unsteady Flows Using Continuous-in-Time Adjoints," *Journal of Computational Physics*, Vol. 341, No. 15, 2017, pp. 258–277. https://doi.org/10.1016/j.jcp.2017.04.005.

[49] Yano, M., "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.

[50] Yano, M., and Darmofal, D., "An optimization-based framework for anisotropic simplex mesh adaptation," *Journal of Computational Physics*, Vol. 231, No. 22, 2012, p. 7626–7649. https://doi.org/10.1016/j.jcp.2012.06.040.

[51] Fidkowski, K. J., "A Local Sampling Approach to Anisotropic Metric-Based Mesh Optimization," AIAA Paper 2016–0835, 2016. https://doi.org/10.2514/6.2016-0835.