



Unsteady Output-Based Adaptation Using Continuous-in-Time Adjoints

Krzysztof J. Fidkowski*

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

We present a method for estimating spatial and temporal numerical errors in scalar outputs of unsteady fluid dynamics simulations using a continuous-in-time adjoint solution and general time-integration methods. A continuous formulation decouples the primal and adjoint temporal discretizations and allows the use of standard time-integration schemes for the adjoint. For non-variational methods, a scheme-agnostic temporal reconstruction of the primal and adjoint solutions replaces the functional representation in between time nodes. Finally, projection of the adjoint to a semi-refined space yields separate measures of the errors in space and time, which then drive adaptive refinement of the spatial and temporal discretizations. Results for the Euler equations and the compressible Navier-Stokes equations demonstrate accuracy of the error estimates and effectiveness of the adaptation.

I. Introduction

Solution-adaptive methods can dramatically improve the robustness and efficiency of computational fluid dynamics simulations through error estimates and optimized meshes. In an output-based setting, mesh resolution is automatically dictated by an error estimate for a scalar output of interest, such as a force component or solution integral. Much work has been done in this area for steady problems, typically with finite-volume and finite-element methods.¹⁻⁸ Unsteady problems pose additional implementation challenges and computational costs, namely in the solution of the unsteady adjoint, which drives the adaptation. However, output-based adaptive methods have been extended to such problems, with various adaptation mechanics, including static-mesh, dynamic-mesh, space-only, and combined space-time⁹⁻¹⁷ refinements.

A variational unsteady discretization, such as a finite-element method in space and time, puts output error estimation on a sound theoretical foundation.⁷ In this case, the adjoint solution, which is at the core of output-based methods, can be solved using a *discrete* approach in which the adjoint equations are obtained systematically from the primal discrete system by transposing the operator. This is the approach taken in many previous works.^{9,10,15-17} While the same approach can be applied to non-variational discretizations, it has pitfalls as the relationship between the resulting discrete adjoint coefficients and the underlying continuous adjoint solution may not be clear, making difficult the computation of the adjoint-weighted residual error estimate and the application of approximate adjoint solvers that rely on smoothness of the adjoint solution.

On the other hand, non-variational methods are ubiquitous for time integration of unsteady problems. Multi-step and multi-stage methods dominate such simulations, due to their simplicity

*Associate Professor, AIAA Senior Member

and generally lower computational cost compared to variational methods. Extension of output-based methods to such discretizations would therefore increase their utility, and this is the motivation for the present paper. Some work in this area has already been done,^{12,14} with scheme-specific algorithms for computing or approximating the fine-space discrete adjoint and linear or spline-based interpolation of solutions between time nodes. In this work we take a step back and derive a general approach for error estimation, through a *continuous*-in-time adjoint solution that accommodates nearly arbitrary time integration methods. We also show how to separate the error into contributions from the spatial and temporal discretizations.

In the remainder of this paper we first introduce our discretization (Section II), which is variational in space but non necessarily in time. This section also shows the derivation of the continuous-in-time adjoint equation, which is used in the output-error estimates in Section III. Section IV outlines the error localization and space-time mesh optimization procedure. Finally, Section V presents results that demonstrate the adaptive method for the compressible Navier-Stokes equations.

II. Governing Equations and Discretization

We simulate the compressible Navier-Stokes equations,

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

where $\mathbf{u} \in \mathbb{R}^s = [\rho, \rho \vec{v}, \rho E]$ is the conservative state vector of rank s , and $\vec{\mathbf{F}}(\mathbf{u}, \nabla \mathbf{u}) = \vec{\mathbf{F}}^{\text{conv}}(\mathbf{u}) + \vec{\mathbf{F}}^{\text{visc}}(\mathbf{u}, \nabla \mathbf{u})$ is the total flux, consisting of the convective and viscous components. We discretize this equation in space and time separately, using a semi-discrete approach, as outlined in this section.

II.A. Spatial Discretization

We employ a discontinuous Galerkin (DG) finite-element method in space. Denote by T_h the set of N_{elem} elements in a non-overlapping tessellation of the domain Ω . In DG, the state is approximated by polynomials of order p_e on each element, with no continuity constraints imposed on the approximations on adjacent elements. Formally, $\mathbf{u}_h \in \mathcal{V}_h = [\mathcal{V}_h]^s$, where $\mathcal{V}_h = \{u \in L_2(\Omega) : u|_{\Omega_e} \in \mathcal{P}^{p_e} \forall \Omega_e \in T_h\}$, and \mathcal{P}^{p_e} denotes polynomials of order p_e on an element Ω_e . The weak form of Eqn. 1 follows from multiplying the equation by test functions in the same approximation space, integrating by parts, and coupling elements via unique fluxes. We use the Roe approximate Riemann solver¹⁸ for the convective flux, and the second form of Bassi and Rebay (BR2)¹⁹ for the viscous flux. Choosing a basis for the test and trial spaces yields a system of ordinary differential equations,

$$\bar{\mathbf{R}}(\mathbf{U}) \equiv \mathbf{M} \frac{d\mathbf{U}}{dt} + \mathbf{R}(\mathbf{U}) = \mathbf{0}, \quad (2)$$

where \mathbf{M} is the mass matrix, $\mathbf{U} \in \mathbb{R}^N$ is the discrete state vector of basis function coefficients, \mathbf{R} is the discrete spatial residual vector, and $\bar{\mathbf{R}}$ is the strong-form unsteady residual.

II.B. Temporal Discretization

We consider general marching schemes for advancing the system of ODEs in Eqn. 2 in time from $t = 0$ to $t = T$. We do not assume a specific form for the time discretization and only require that the method advances the state one time step, Δt : from time node n to $n+1$, i.e. $\mathbf{U}^n \rightarrow \mathbf{U}^{n+1}$. This encompasses both variational time integrators, based on a weak-form in time, and non-variational

integrators such as traditional multi-step and multi-stage methods. In this work we show results for a backwards-difference multi-step method (BDF2), implicit Runge-Kutta methods (DIRK3, DIRK4, ESDIRK5), and a split Adams-Moulton method (SAMF3). Appendix A presents the details of these schemes.

II.C. Continuous-in-Time Adjoint

Consider an unsteady output of the form

$$\bar{J} \equiv \int_0^T J(\mathbf{U}(t), t) dt + J_T(\mathbf{U}(T)), \quad (3)$$

where J and J_T are functions of the spatial distribution of the state via the discrete coefficients \mathbf{U} . Note that J_T is a function of only the final-time state, $\mathbf{U}(T)$. The continuous-in-time adjoint, $\Psi(t)$, is the sensitivity of \bar{J} to source perturbations in the unsteady residual, \mathbf{R} , in Eqn. 2. To derive the equation for $\Psi(t)$, we define a Lagrangian

$$\mathcal{L} \equiv \bar{J} + \int_0^T \Psi^T \bar{\mathbf{R}} dt = \bar{J} + \int_0^T \Psi^T \left(\mathbf{M} \frac{d\mathbf{U}}{dt} + \mathbf{R}(\mathbf{U}) \right) dt. \quad (4)$$

Integrating the first term in the integral by parts, substituting Eqn. 3, and requiring stationarity of the Lagrangian with respect to permissible state variations, $\delta\mathbf{U}$, gives

$$\frac{dJ_T}{d\mathbf{U}} \delta\mathbf{U} \Big|_{t=T} + \Psi^T \mathbf{M} \delta\mathbf{U} \Big|_{t=T} - \Psi^T \mathbf{M} \delta\mathbf{U} \Big|_{t=0} + \int_0^T \left[\frac{\partial J}{\partial \mathbf{U}} - \frac{d\Psi^T}{dt} \mathbf{M} + \Psi^T \frac{\partial \mathbf{R}}{\partial \mathbf{U}} \right] \delta\mathbf{U} dt = \mathbf{0}. \quad (5)$$

The middle term at $t = 0$ drops out since the initial condition on the primal fully constrains the state there, so $\delta\mathbf{U} = 0$ at $t = 0$. The remaining terms yield the adjoint differential equation (from the time integrand, transposed),

$$-\mathbf{M} \frac{d\Psi}{dt} + \frac{\partial \mathbf{R}^T}{\partial \mathbf{U}} \Psi + \frac{\partial J^T}{\partial \mathbf{U}} = \mathbf{0}, \quad (6)$$

and the terminal condition (from the $t = T$ terms),

$$\Psi(T) = -\mathbf{M}^{-1} \frac{dJ_T^T}{d\mathbf{U}}. \quad (7)$$

Due to the terminal condition, the adjoint is solved backwards in time, from $t = T$ to $t = 0$, using a time-integration method of choice, which need not be the same as that for the primal. We note that when marching backwards in time, it is useful to define $\tau = T - t$ and to rewrite Eqn. 6 in a form similar to the primal,

$$\mathbf{M} \frac{d\Psi}{d\tau} + \underbrace{\frac{\partial \mathbf{R}^T}{\partial \mathbf{U}} \Psi + \frac{\partial J^T}{\partial \mathbf{U}}}_{\mathbf{R}^\Psi(\mathbf{U}, \Psi)} = \mathbf{0}, \quad (8)$$

where \mathbf{R}^Ψ is the adjoint residual. In terms of τ , the unsteady adjoint equation takes the same form as the primal equation, Eqn. 2, and this simplifies the application of general time-marching schemes.

II.D. Temporal Reconstruction

For nonlinear problems, the adjoint residual in Eqn. 8 depends on the primal state \mathbf{U} . In multi-stage time integration, the residual must be evaluated at times in between the time nodes, but \mathbf{U} is not directly available at these time locations; at least not when using general non-variational time integrators, where only the nodal states are saved. The same situation occurs in error estimation, where both the primal and adjoint are required in between time nodes for integrating the adjoint-weighted residual. In this work we evaluate the primal/adjoint in between time nodes using a temporal reconstruction with a prescribed order of accuracy.

Without loss of generality, we present the reconstruction for the primal solution. Re-arranging the semi-discrete form in Eqn. 2 (Eqn. 8 for the adjoint), we can evaluate the state slope,

$$\frac{d\mathbf{U}}{dt} = -\mathbf{M}^{-1}\mathbf{R}(\mathbf{U}). \quad (9)$$

Hence, the slope in time requires a residual evaluation at a known state. Suppose that we have the states at the endpoints of a time interval, t^n and t^{n+1} , from a general time-marching scheme. A simple reconstruction is a linear connection of the states, though this only gives second-order accuracy. We can improve the accuracy by applying Eqn. 9 to evaluate the slopes at the endpoints, and using these four pieces of information reconstruct a cubic solution in time, as shown in Figure 1.

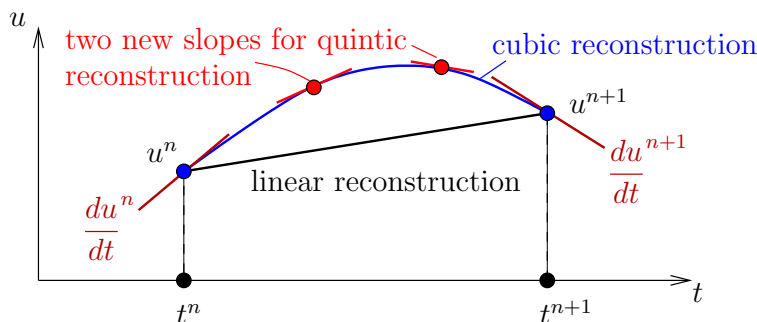


Figure 1. Illustration of solution reconstruction on a time interval. High-order approximations can be constructed using slope information readily available from residual evaluations.

Furthermore, we can gain additional accuracy by applying Eqn. 9 yet again, in between the time nodes, starting with the reconstructed cubic solution. The locations of slope calculations are not fixed, and we use quadrature points to allow for reuse of residual evaluations when integrating, e.g. during error estimation. For two quadrature points, we evaluate the cubic at the two points and apply Eqn. 9 to obtain the slopes there. With these additional two pieces of information, we can reconstruct a quintic solution in time. However, as the original state came from a cubic reconstruction, the order of accuracy of this quintic will only be one higher than a cubic, i.e. fifth order. To obtain sixth-order accuracy, we simply iterate the process, using the current approximation to re-evaluate the slopes at the two middle points, which then yields a new approximation. Just one such iteration will yield the expected sixth-order convergence rates, but an additional iteration will further lower the errors.

Table 1 shows results from a numerical test of the reconstruction for a scalar governed by the ordinary differential equation $\frac{du}{dt} = u^2$. Reconstruction is performed over one interval, $[0, \Delta t]$, and the error is defined as $E^2 = \frac{1}{\Delta t} \int_0^{\Delta t} (u - u_{\text{exact}})^2 dt$. The results show the optimal fourth-order convergence for the cubic reconstruction, and sixth-order convergence for the quintic reconstruction with at least one iteration.

Δt	Cubic		Quintic-0		Quintic-1		Quintic-2	
	Error	Rate	Error	Rate	Error	Rate	Error	Rate
1/4	8.87e-05	–	6.02e-06	–	8.54e-07	–	1.56e-07	–
1/8	7.24e-06	3.62	2.58e-07	4.55	1.93e-08	5.47	3.55e-09	5.46
1/16	5.22e-07	3.79	9.55e-09	4.75	3.68e-10	5.71	6.79e-11	5.71
1/32	3.52e-08	3.89	3.26e-10	4.87	6.38e-12	5.85	1.18e-12	5.85
1/64	2.28e-09	3.94	1.07e-11	4.93	1.03e-13	5.95	1.75e-14	6.07

Table 1. Errors and orders of accuracy of time-step reconstruction of a sample scalar problem, using endpoint values and slopes only (cubic) and additional interior slopes and n iterations (quintic- n).

III. Output Error Estimation

An adjoint solution can be used to estimate the numerical error in the corresponding output of interest, \bar{J} , through the adjoint-weighted residual.^{2,7} Consider first only temporal error, which is the numerical error arising from the discretization in time. Denote by $\mathbf{U}_H(t)$ the approximate primal solution obtained from a chosen time integration method and time step size. If we had the exact unsteady adjoint solution, $\Psi(t)$, we could use Eqn. 4 to estimate the error in \bar{J} ,

$$\delta\bar{J} \equiv \bar{J}(\mathbf{U}_H) - \bar{J}(\mathbf{U}) \approx \frac{\partial\bar{J}}{\partial\mathbf{U}}\delta\mathbf{U} \approx - \int_0^T \Psi^T \bar{\mathbf{R}}(\mathbf{U}_H) dt, \quad (10)$$

where $\delta\mathbf{U} \equiv \mathbf{U}_H - \mathbf{U}$ is the state error, and $\bar{\mathbf{R}}(\mathbf{U}_H) \approx \frac{\partial\bar{\mathbf{R}}}{\partial\mathbf{U}}(\delta\mathbf{U})$ is the generally nonzero unsteady residual obtained from the approximate primal. In practice, the exact adjoint is not available and must be approximated in a *fine space*, denoted by subscript h , which in our work is a higher-order time integration method. Numerical error due to the spatial discretization can be measured by also refining the spatial discretization. In the present work this is accomplished by increasing the spatial order of the DG discretization by one. The final form of the error estimate is

$$\delta\bar{J} \approx - \int_0^T \Psi_h^T \bar{\mathbf{R}}_h(\mathbf{U}_h^H) dt, \quad (11)$$

where \mathbf{U}_h^H is the injection of the primal from space H to space h . In the spatial domain, this is a pure injection to higher order: $p \rightarrow p + 1$. In the temporal domain, this involves a sufficiently-accurate reconstruction over the time interval, e.g. cubic for DIRK3 or DIRK4. The integral in Eqn. 11 is a summation of integrals over time intervals, each performed with quadrature. Temporal reconstruction of the fine-space adjoint yields the adjoint at the quadrature points.

The output error estimate in Eqn. 11 can be separated into spatial and temporal components by selectively refining the fine space only in space or only in time. Specifically, we obtain the temporal error, $\delta\bar{J}^{\text{time}}$, by projecting the fine-space adjoint, Ψ_h , spatially back down to order p and recalculating the error via Eqn. 11 with the projected adjoint. As non-variational time integrators do not support a straightforward projection, we do not use a similar approach for the spatial error. Instead, we define $\delta\bar{J}^{\text{space}} \equiv \delta\bar{J} - \delta\bar{J}^{\text{time}}$.

IV. Adaptation

To improve accuracy of the output prediction, we adapt the spatial and temporal discretizations. In this section, we discuss how the error is localized to provide an indicator, and how the

discretization refinement (or coarsening) is applied.

IV.A. Error Localization

The error estimate in Eqn. 11 is a time integral of an inner product between vectors that contain spatially-local data specific to each element. We can rewrite the output error as

$$\delta\bar{J} \approx - \int_0^T \Psi_h^T \bar{\mathbf{R}}_h(\mathbf{U}_h^H) dt, = \sum_{n=1}^{N_t} \sum_{e=1}^{N_e} \underbrace{\int_{t^{n-1}}^{t^n} -\Psi_{h,e}^T \bar{\mathbf{R}}_{h,e}(\mathbf{U}_h^H) dt}_{\varepsilon_e^n}, \quad (12)$$

where N_t is the number of time steps, N_e is the number of elements, and the subscript e on the adjoint and residual denotes restriction to element e . The underbrace defines ε_e^n , the error contribution of element e during time step n . The temporal error contribution is estimated by using a spatially down-projected adjoint,

$$\varepsilon_e^{n,\text{time}} = - \int_{t^{n-1}}^{t^n} \left(\mathbf{I}_H^h \Psi_{h,e} \right)^T \bar{\mathbf{R}}_{H,e}(\mathbf{U}_H) dt, \quad (13)$$

where \mathbf{I}_H^h is a least-squares spatial projection operator, specifically from order $p_e + 1$ to order p_e on element e . Note that the residual is evaluated on the spatially-coarse approximation space, using the coarse primal solution \mathbf{U}_H .

The spatial error for element e at time step n is then defined as $\varepsilon_e^{n,\text{space}} \equiv \varepsilon_e^n - \varepsilon_e^{n,\text{time}}$. For static spatial mesh adaptation, we use the aggregate spatial error, which for element e is

$$\varepsilon_e^{\text{space}} \equiv \sum_{n=1}^{N_t} \varepsilon_e^{n,\text{space}}. \quad (14)$$

IV.B. Adaptive Mechanics and Optimization

The localized error estimates guide refinement or coarsening of the spatial and temporal discretizations. In this work we consider a uniform temporal discretization, where all time steps are of the same size, $\Delta t = T/N_t$. In space, we use order adaptation or hanging-node mesh refinement.

The adaptation relies on models of how the spatial and temporal errors behave with refinement, and of the cost of these refinements. For the cost, we use the total number of space-time degrees of freedom,

$$C \equiv C^{\text{space}} C^{\text{time}}, \quad C^{\text{space}} \equiv \sum_{e=1}^{N_e} n(p_e), \quad C^{\text{time}} \equiv N_t n_r, \quad (15)$$

where $n(p_e)$ is the number of spatial degrees for an element of order p_e , and n_r is the number of temporal degrees of freedom, i.e. system solves, per time step. For example, BDF2 has $n_r = 1$, while DIRK schemes have $n_r = n_{\text{stage}}$, the number of stages.

For the error, we use a simple a priori model based on the degrees of freedom,

$$|\delta\bar{J}| = |\delta\bar{J}^{\text{space}}| + |\delta\bar{J}^{\text{time}}|, \quad |\delta\bar{J}^{\text{space}}| \propto (C^{\text{space}})^{-(p+1)/d}, \quad |\delta\bar{J}^{\text{time}}| \propto (C^{\text{time}})^{-(r+1)}, \quad (16)$$

where d is the spatial dimension, p is the average spatial order in the mesh, and r is the formal order of accuracy of the time integration scheme.

We consider a space-time distribution of degrees of freedom optimal if the marginal error reduction per cost increase is the same between the spatial and temporal discretizations. If this were not the case, we could further reduce the error with no cost increase by reallocating degrees of freedom between the spatial and temporal discretizations. For one adaptive iteration, define growth factors f^{space} and f^{time} in the spatial and temporal degrees of freedom, respectively, so that the changes in the costs are

$$C^{\text{space}} = C_0^{\text{space}} f^{\text{space}}, \quad C^{\text{time}} = C_0^{\text{time}} f^{\text{time}},$$

where the subscript 0 indicates values in the unrefined mesh. Similarly, the changes in the errors are, from Eqn. 16,

$$|\delta \bar{J}^{\text{space}}| = |\delta \bar{J}_0^{\text{space}}| (f^{\text{space}})^{-(p+1)/d}, \quad |\delta \bar{J}^{\text{time}}| = |\delta \bar{J}_0^{\text{time}}| (f^{\text{time}})^{-(r+1)}. \quad (17)$$

To determine f^{space} and f^{time} , we require equal marginal error to cost ratios, $\lambda^{\text{space}} = \lambda^{\text{time}}$, where

$$\lambda^{\text{space}} \equiv \frac{\partial(\delta \bar{J})}{\partial f^{\text{space}}} \left[\frac{\partial C}{\partial f^{\text{space}}} \right]^{-1}, \quad \lambda^{\text{time}} \equiv \frac{\partial(\delta \bar{J})}{\partial f^{\text{time}}} \left[\frac{\partial C}{\partial f^{\text{time}}} \right]^{-1}. \quad (18)$$

This is one equation for two variables, and the second equation is a constraint on the growth of the total degrees of freedom: $f^{\text{space}} f^{\text{time}} = f^{\text{tot}}$, where f^{tot} is a growth factor prescribed by the user. Substituting the error and cost models into the expressions for λ^{space} and λ^{time} , we obtain the following solution:

$$f^{\text{time}} = \left[\frac{d(r+1)}{p+1} \frac{|\delta \bar{J}^{\text{time}}|}{|\delta \bar{J}^{\text{space}}|} (f^{\text{tot}})^{1+(p+1)/d} \right]^{\frac{1}{r+3+(p+1)/d}}, \quad f^{\text{space}} = \frac{f^{\text{tot}}}{f^{\text{time}}}.$$

This equation appropriately adjusts the spatial and temporal refinement fractions to make equal the marginal error to cost ratio of the two refinement options. Depending on the ratio of the spatial and temporal errors, either the spatial or the temporal mesh will be refined more, and in some cases one of the meshes could be coarsened.

V. Results

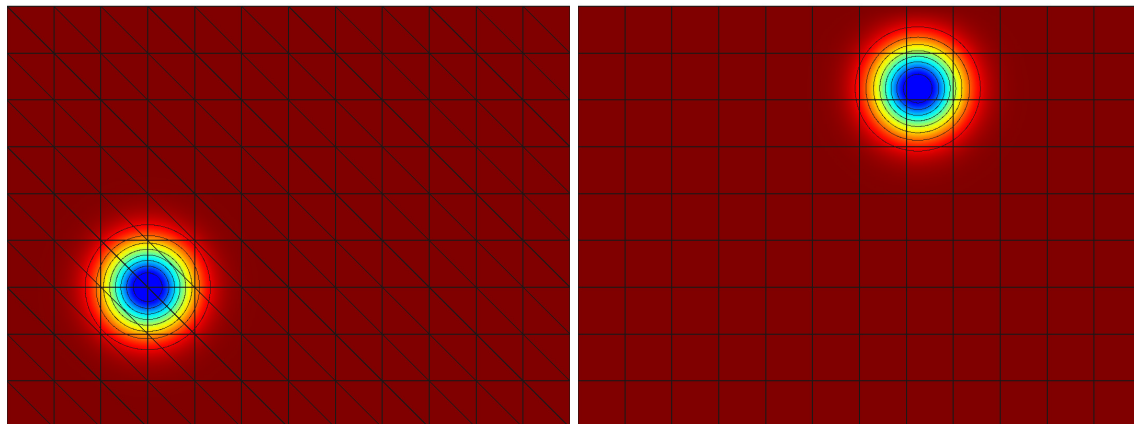
In this section, we demonstrate the effectivity of the error estimate and the efficiency of the adapted space-time meshes on two test cases.

V.A. Euler Vortex Propagation

First, we consider an analytical vortex solution to the Euler equations.^{20,21} The domain consists of a rectangular box, $[0, 20] \times [0, 15]$, and the vortex begins at $(x, y) = (5, 5)$ and moves up and to the right with unit velocity. With the exception of the velocity direction, all other properties of the vortex solution are the same as in our previous work.²¹ The final time is $T = 10$, and no mesh motion is imposed in this case.

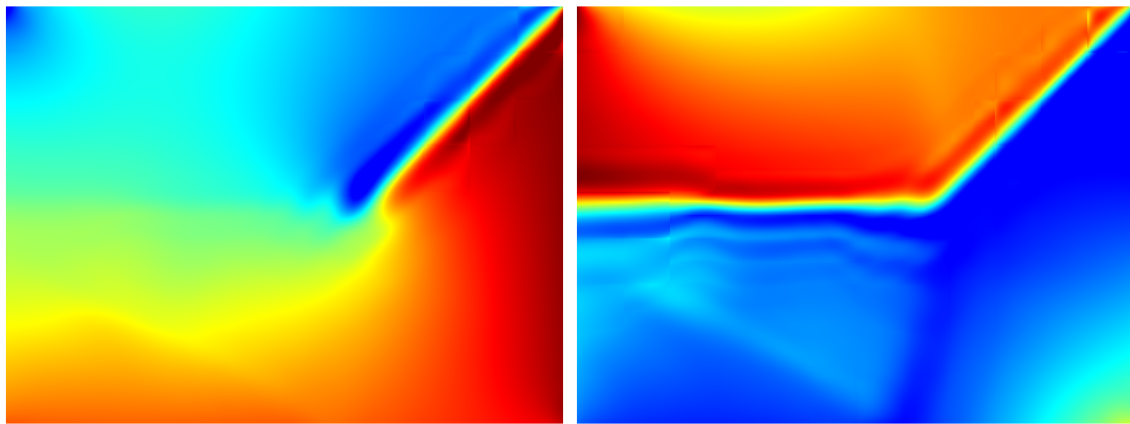
Figure 2 shows the initial computational meshes and vortex solutions at the initial and final times. Both triangular and quadrilateral meshes were considered for this case. The output of interest in this study is a time integrated y -momentum flux integral over the top boundary of the domain,

$$\bar{J} = \int_0^T \int_{\text{top}} \rho v \, dx \, dt \quad (19)$$



(a) Initial pressure, 216 triangles

(b) Final pressure, 108 quadrilaterals



(c) $t = 0$ adjoint, conservation of y momentum equation

(d) $t = 0$ adjoint, conservation of energy equation

Figure 2. Euler vortex: pressure contours at the initial and final times on the two starting meshes used for this study. Also shown are two components of the adjoint at $t = 0$.

Figure 2 also shows two components of the adjoint for this output, at $t = 0$. The feature in the upper-right corners of both plots is caused by the advection mode of the Euler equations, for the 45° flow direction in this case.

Figure 3 shows the results of the present adjoint-based adaptive approach applied to various starting meshes and adaptive mechanics. “tri-p” refers to order refinement on the 216-element triangular mesh, starting with $p = 1$, while “quad-h” refers to hanging-node refinement of quadrilaterals, starting with the 108-element mesh. The degrees of freedom were measured as the total number of space-time unknowns, and growth factors of $f^{\text{tot}} = 1.5, 2.0$ were used for the triangular and quadrilateral refinements, respectively. In the legend labels, N_t refers to the number of time steps on the initial mesh.

For error calculation, the exact/truth solution was computed on a $p = 6$ quadrilateral mesh with $N_t = 80$ time steps of ESDIRK5. From the figure, we see that although the adaptive refinements beat the corresponding uniform refinements, the differences are not overly large. This is in part because the flow in this case is smooth, with no sharp features requiring extra resolution, and because the vortex passes through a large portion of the domain, which ends up targeted for refinement with our static approach. However, we can use these results for verification, since also shown in the figure are the output values corrected with the error estimates available at every adaptive iteration. These corrected outputs are shown as dashed lines, and they converge faster than the original outputs, which indicates that the error estimation approach is effective at identifying the space-time error. Even for this smooth verification case, the corrected values show a significant improvement over uniform refinement in terms of the degrees of freedom required to achieve a given error tolerance.

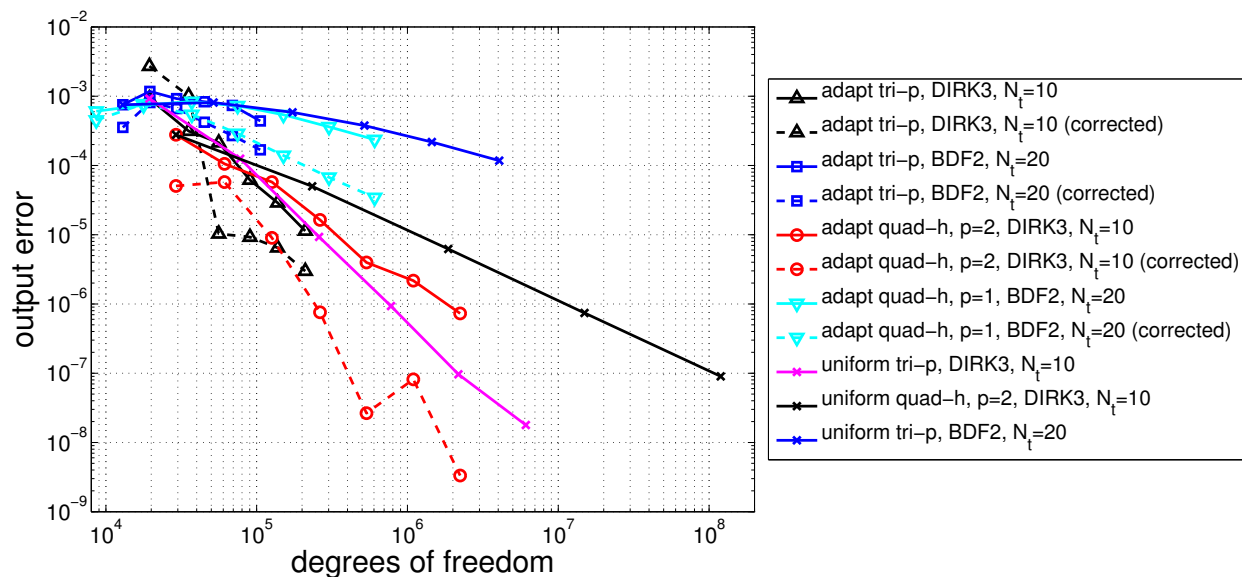
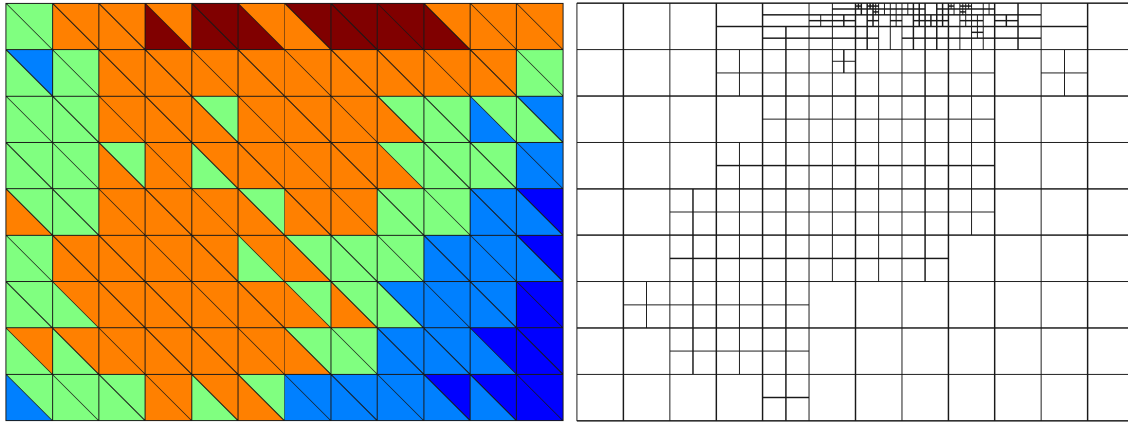


Figure 3. Euler vortex: convergence of a top-boundary y -momentum integral output using various adaptive and uniform refinement strategies.

Finally, Figure 4 shows the order distribution for the order-adaptive triangular case, and the mesh for the hanging node adaptive quadrilateral case. Both are shown at the fourth adaptive iteration. The regions targeted for refinement are similar: the path of the vortex and the top boundary, on which the output is measured.



(a) Adapt tri- p , DIRK3, $N_t = 10$

(b) Adapt quad- h , $p = 2$, DIRK3, $N_t = 10$

Figure 4. Euler vortex: element polynomial orders (0 to 5) and adapted meshe at the fourth adaptive iteration for order-refinement on triangles and hanging-node refinement on quadrilaterals.

V.B. Heaving Airfoil

This case consists of a modified NACA 0012 airfoil undergoing pure heaving (plunge) motion in viscous flow governed by the compressible Navier-Stokes equations ($\gamma = 1.4, Pr = 0.72$, constant viscosity). The modified NACA 0012 geometry has a closed trailing edge,

$$y(x) = \pm 0.6(0.2969\sqrt{x} - 0.1260x - 0.3516x^2 + 0.2843x^3 - 0.1036x^4), \quad x \in [0, 1].$$

The initial condition is a steady state solve at free-stream Mach number $M_\infty = 0.2$ and Reynolds number $Re = 1000$. We use convenient units in which the airfoil chord is $c = 1$ and the free-stream density and speed are unity, so that the free-stream conservative state vector is

$$[\rho, \rho u, \rho v, \rho E] = [1, 1, 0, 0.5 + 1/[M_\infty^2 \gamma (\gamma - 1)]] .$$

Full-state boundary conditions are imposed on the farfield boundary, which is 100 chord-lengths away. The mesh, shown in Figure 5a, consists of 664 triangles, curved to quartic geometry representation on the boundary.

Following a steady-state solve for initialization, the unsteady simulation begins with a plunge motion in which the vertical displacement is given by $h(t) = t^2(3-t)/4$, $t \in [0, T]$, and $T = 2$. The output of interest is a weighted time-integral of the lift force on the airfoil,

$$\bar{J} = \int_0^T w(t)L(t) dl, \quad (20)$$

where $w(t) = \exp(-30(t-1)^2)$ is a temporal weight function whose role is to make the output smooth in time, and $L(t)$ is the instantaneous lift (vertical) force on the airfoil.

Figure 5 shows Mach number contours in the flow at various times during the unsteady simulation. The combined motion of the airfoil disturbs both the boundary layer, wake, and a sizable portion of the flow below the airfoil. In addition, adjoint contours show, at two different times, the sensitivities of the output to source perturbations in the conservation of energy equation.

Figure 6 shows the convergence of the error in the weighted lift integral output, using the current adjoint-based approach compared to uniform refinement. At every stage of uniform refinement, the order of each element is increased by 1, and the number of time steps is doubled. The initial

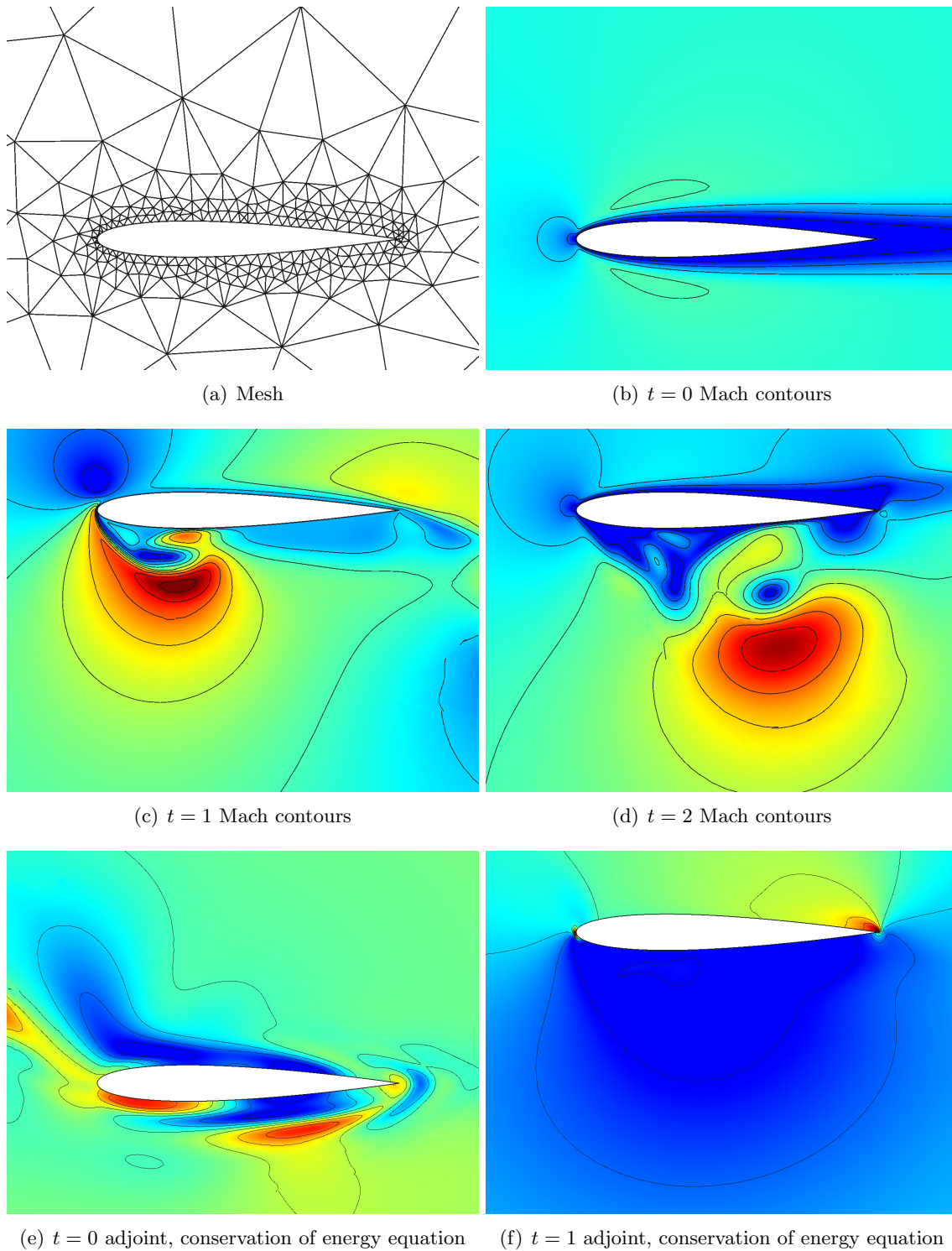


Figure 5. Compressible Navier-Stokes flow over a heaving NACA 0012 airfoil: mesh and Mach contours (0 to 0.5) at various times in the simulation. Also shown are the conservation of energy equation components of the adjoint at two times.

solution is on a $p = 1$ mesh with $N_t = 12$ time steps. Two sets of adaptive refinements are shown, one using DIRK3, and one using a split Adams-Moulton method, SAMF3, for the primal solve. In both cases, the continuous fine-space adjoint is solved using ESDIRK5 on an order-refined mesh in which the spatial order is incremented by 1 on each element. The growth factor for the total number of space-time degrees of freedom per adaptive iteration is set to $f^{\text{tot}} = 1.5$.

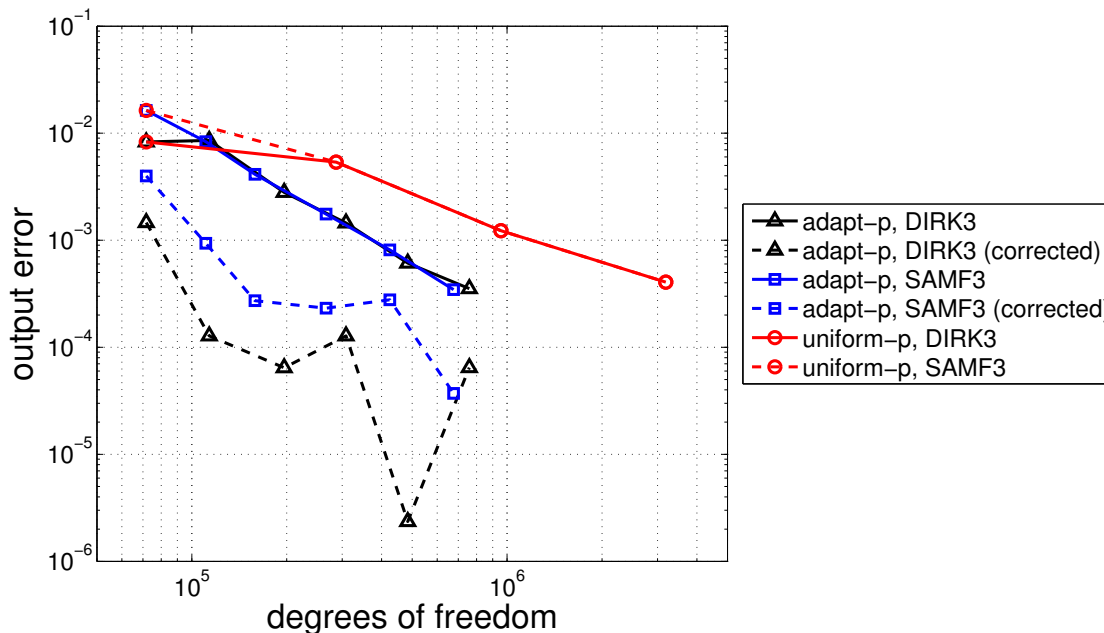


Figure 6. Compressible Navier-Stokes flow over a heaving NACA 0012 airfoil: convergence of a weighted lift integral output using adaptation and uniform refinement.

The exact solution was computed on a $p = 6$ mesh with $N_t = 80$ time steps of ESDIRK5. As shown in Figure 6, the output-based adaptive simulations beat uniform order refinement in error convergence with degrees of freedom. Furthermore, the availability of the space-time error estimate allows for a correction to be applied to the output values at each adaptive iteration. The resulting corrected outputs, shown as dashed lines, converge faster than the outputs, indicating effectivity of the error estimates, and provide significant savings versus uniform refinement in the degrees of freedom required to achieve a low error tolerance.

Finally, Figure 7 shows the spatial distributions of element orders for several adaptive iterations of the DIRK3 run. As expected, we see order refinement near the airfoil, generally for the larger elements. However, not all regions are targeted equally, as evidenced by the presence of low-order elements around the airfoil, especially over the upper surface.

VI. Conclusions

We present an approach for estimating numerical errors in outputs of unsteady fluid dynamics simulations using a continuous-in-time adjoint solution and general time-integration methods. The decoupling of the primal and adjoint temporal discretizations removes constraints on the choice of the adjoint time integration scheme and allows for standard schemes to be used for the adjoint. The error estimation uses a scheme-agnostic temporal reconstruction of the primal and adjoint to obtain consistent and high-order representations of the primal and adjoint solutions within each time step. We separate spatial and temporal error contributions by projecting the fine-space adjoint down to a coarse spatial discretization and recalculating the error estimate. The resulting spatial and temporal error estimates drive an adaptive procedure in which the spatial mesh and time-step size

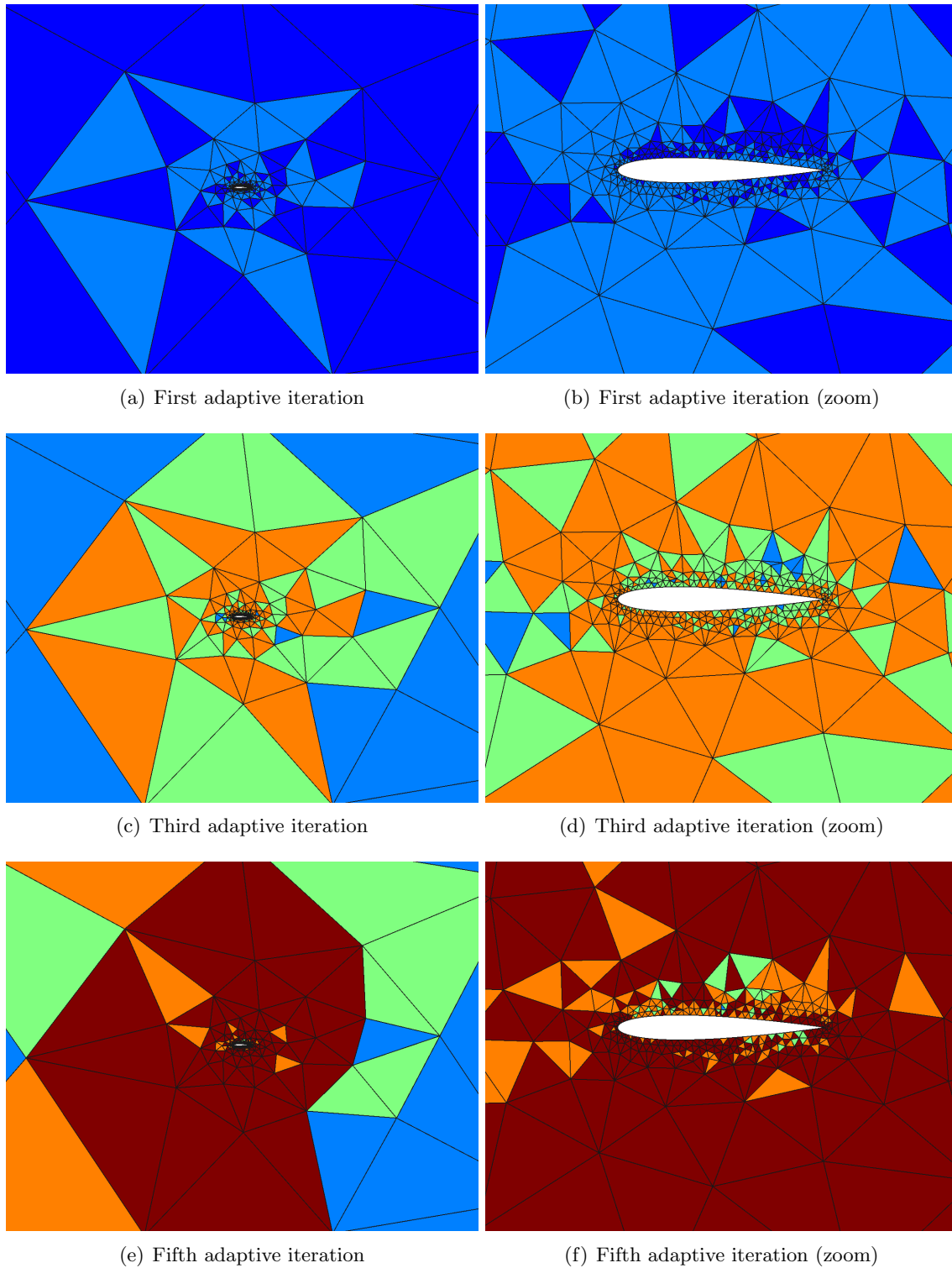


Figure 7. Compressible Navier-Stokes flow over a heaving NACA 0012 airfoil: element polynomial orders (0 to 5) in the adapted meshes at three of the adaptive iterations.

are refined or coarsened to produce an optimal space-time mesh. Results for the Euler equations and the compressible Navier-Stokes equations demonstrate accuracy of the error estimates and effectiveness of the adaptation.

Acknowledgments

The author acknowledges support from the Department of Energy under grant DE-FG02-13ER26146/DE-SC0010341, and from The Boeing Company, with technical monitor Dr. Mori Mani.

A. Time Schemes

An n_{step} backwards differentiation (BDF) formula takes the form

$$\frac{\mathbf{M}}{\Delta t} \sum_{i=0}^{n_{\text{step}}} a_i \mathbf{U}^{n+1-i} + \mathbf{R}(\mathbf{U}^{n+1}, t^{n+1}) = \mathbf{0},$$

where for BDF1, $n_{\text{step}} = 1$, $a_i = [1, -1]$, and for BDF2, $n_{\text{step}} = 2$, $a_i = [\frac{3}{2}, -2, \frac{1}{2}]$.

An n_{stage} diagonally-implicit Runge-Kutta (DIRK) method for advancing the state from \mathbf{U}^n to \mathbf{U}^{n+1} over a time step of size Δt takes the form

$$\begin{aligned} &\text{for } i = 1 : n_{\text{stage}} \\ &\quad \mathbf{S}_i = -\frac{\mathbf{M}}{\Delta t} \mathbf{W}^0 + \sum_{j=1}^{i-1} a_{ij} \mathbf{R}(\mathbf{W}^j, t_j) \\ &\quad \text{solve: } \frac{\mathbf{M}}{\Delta t} \mathbf{W}^i + a_{ii} \mathbf{R}(\mathbf{W}^i, t_i) + \mathbf{S}_i = \mathbf{0} \\ &\text{end} \end{aligned}$$

where $\mathbf{W}^0 = \mathbf{U}^n$ is the state at the start of the time interval, $\mathbf{U}^{n+1} = \mathbf{W}^{n_{\text{stage}}}$ is the desired result, and $t_i = t^n + b_i \Delta t$ are the stage times. The coefficients a_{ij} and b_i define the method. For the third-order accurate DIRK3 scheme, $n_{\text{stage}} = 3$, and the coefficients are²²

$$a_{ij} = \begin{bmatrix} \alpha & 0 & 0 \\ \tau - \alpha & \alpha & 0 \\ \beta_1 & \beta_2 & \alpha \end{bmatrix}, \quad b_i = \begin{bmatrix} \alpha \\ \tau \\ 1 \end{bmatrix}.$$

where α is the root of $x^3 - 3x^2 + \frac{3}{2}x - \frac{1}{6} = 0$ lying in $(\frac{1}{6}, \frac{1}{2})$, $\tau = (1 + \alpha)/2$, $\beta_1 = -(6\alpha^2 - 16\alpha + 1)/4$, and $\beta_2 = (6\alpha^2 - 20\alpha + 5)/4$. A fourth-order accurate DIRK4 scheme with $n_{\text{stage}} = 5$ is²³

$$a_{ij} = \begin{bmatrix} \frac{1}{4} & 0 & 0 & 0 & 0 \\ \frac{1}{2} & \frac{1}{4} & 0 & 0 & 0 \\ \frac{17}{50} & -\frac{1}{25} & \frac{1}{4} & 0 & 0 \\ \frac{371}{1360} & -\frac{137}{2720} & \frac{15}{544} & \frac{1}{4} & 0 \\ \frac{25}{24} & -\frac{49}{48} & \frac{125}{16} & -\frac{85}{12} & \frac{1}{4} \end{bmatrix}, \quad b_i = \begin{bmatrix} \frac{1}{4} \\ \frac{3}{4} \\ \frac{11}{20} \\ \frac{1}{2} \\ 1 \end{bmatrix}.$$

ESDIRK5 is a seven-stage, fifth-order scheme that takes the same form. Coefficients for this scheme can be found in.²⁴

Finally, we also employ schemes of the split Adams-Moulton family (SAMF).²⁵ These methods consists of two stages, the first of which is a predictor,

$$\frac{\mathbf{M}}{\Delta t} \frac{1}{b_0 + c_0 \theta} \bar{\mathbf{U}} + \mathbf{R}(\bar{\mathbf{U}}, t^{n+1}) + \mathbf{S}_0 = \mathbf{0},$$

where

$$\mathbf{S}_0 = \frac{\mathbf{M}}{\Delta t} \left[-(1 + a_{\text{sum}}\theta)\mathbf{U}^n + \sum_{i=2}^{n_{\text{stage}}} a_i\theta\mathbf{U}^{n+1-i} \right] + \sum_{i=1}^{n_{\text{stage}}} (b_i + c_i\theta)\mathbf{R}(\mathbf{U}^{n+1-i}, t^{n+1-i}),$$

and $a_{\text{sum}} = \sum_{i=2}^{n_{\text{stage}}} a_i$. The second stage is a corrector,

$$\frac{\mathbf{M}}{\Delta t}\mathbf{U}^{n+1} + (b_0 + c_0\theta)\mathbf{R}(\mathbf{U}^{n+1}, t^{n+1}) + \mathbf{S}_1 = \mathbf{0},$$

where

$$\mathbf{S}_1 = -\frac{\mathbf{M}}{\Delta t}\mathbf{U}^n + \sum_{i=1}^{n_{\text{stage}}} b_i\mathbf{R}(\mathbf{U}^{n+1-i}, t^{n+1-i}) + \theta\mathbf{R}(\bar{\mathbf{U}}, t^{n+1}).$$

Specifically, SAMF3 is a three-step, two-stage, fourth order, method with $A(89.999^\circ)$ stability (almost A -stable), with coefficients (starting at index 0),

$$\begin{aligned} \mathbf{a} &= [1, 0, -72/15, 16/15], \\ \mathbf{b} &= [9/24, 19/24, -5/24, 1/24], \\ \mathbf{c} &= [-1, 209/45, -11/9, 11/45], \\ \theta &= -15/88 - 3\sqrt{5}/22. \end{aligned}$$

For the initial time steps of this multi-step scheme, we use DIRK3.

References

- ¹Pierce, N. A. and Giles, M. B., “Adjoint recovery of superconvergent functionals from PDE approximations,” *SIAM Review*, Vol. 42, No. 2, 2000, pp. 247–264.
- ²Becker, 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.
- ³Hartmann, R. and Houston, P., “Adaptive discontinuous Galerkin finite element methods for the compressible Euler equations,” *Journal of Computational Physics*, Vol. 183, No. 2, 2002, pp. 508–532.
- ⁴Venditti, 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.
- ⁵Sen, S., Veroy, K., Huynh, D., Deparis, S., Nguyen, N., and Patera, A., ““Natural norm” a posteriori error estimators for reduced basis approximations,” *Journal of Computational Physics*, Vol. 217, 2006, pp. 37–62.
- ⁶Nemec, M. and Aftosmis, M. J., “Error Estimation and Adaptive Refinement for Embedded-Boundary Cartesian Meshes,” AIAA Paper 2007-4187, 2007.
- ⁷Fidkowski, K. J. and Darmofal, D. L., “Review of Output-Based Error Estimation and Mesh Adaptation in Computational Fluid Dynamics,” *American Institute of Aeronautics and Astronautics Journal*, Vol. 49, No. 4, 2011, pp. 673–694.
- ⁸Woopan, M., Balan, A., May, G., and Schütz, J., “A comparison of hybridized and standard DG methods for target-based hp-adaptive simulation of compressible flow,” *Computers & Fluids*, Vol. 98, 2014, pp. 3–16.
- ⁹Schmich, M. and Vexler, B., “Adaptivity with Dynamic Meshes for Space-Time Finite Element Discretizations of Parabolic Equations,” *SIAM Journal on Scientific Computing*, Vol. 30, No. 1, 2008, pp. 369–393.
- ¹⁰Barth, 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.
- ¹¹Besier, M. and Rannacher, R., “Goal-oriented space-time adaptivity in the finite element Galerkin method for the computation of nonstationary incompressible flow,” *International Journal for Numerical Methods in Fluids*, Vol. 70, 2012, pp. 1139–1166.
- ¹²Mani, 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.

¹³Belme, A., Dervieux, A., and Alauzet, F., “Error Estimation and Adaptation for Functional Outputs in Time-Dependent Flow Problems,” *Journal of Computational Physics*, Vol. 231, 2012, pp. 6323–6348.

¹⁴Flynt, B. T. and Mavriplis, D. J., “Discrete Adjoint Based Adaptive Error Control in Unsteady Flow Problems,” AIAA Paper 2012-0078, 2012.

¹⁵Fidkowski, 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.

¹⁶Fidkowski, K. J., “An Output-Based Dynamic Order Refinement Strategy for Unsteady Aerodynamics,” AIAA Paper 2012-77, 2012.

¹⁷Kast, 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.

¹⁸Roe, P., “Approximate Riemann solvers, parameter vectors, and difference schemes,” *Journal of Computational Physics*, Vol. 43, 1981, pp. 357–372.

¹⁹Bassi, F. and Rebay, S., “GMRES discontinuous Galerkin solution of the compressible Navier-Stokes equations,” *Discontinuous Galerkin Methods: Theory, Computation and Applications*, edited by K. Cockburn and Shu, Springer, Berlin, 2000, pp. 197–208.

²⁰Persson, P.-O., Bonet, J., and Peraire, J., “Discontinuous Galerkin Solution of the Navier-Stokes, Equations on Deformable Domains,” *Computer Methods in Applied Mechanics and Engineering*, Vol. 198, 2009, pp. 1585–1595.

²¹Fidkowski, K. J., “A Hybridized Discontinuous Galerkin Method on Mapped Deforming Domains,” *Computers and Fluids*, Vol. In Press, 2016.

²²Alexander, R., “Diagonally implicit Runge-Kutta methods for stiff ODE’s,” *SIAM Journal on Numerical Analysis*, Vol. 14, No. 6, 1977, pp. 1006–1021.

²³Rattenbury, N., *Almost Runge-Kutta Methods for Stiff and Non-Stiff Problems*, Ph.D. thesis, The University of Auckland, 2005.

²⁴Kennedy, C. A. and Carpenter, M. H., “Additive Runge-Kutta schemes for convection-diffusion-reaction equations,” *Applied Numerical Mathematics*, Vol. 44, 2003, pp. 139–181.

²⁵Voss, D. A. and Casper, M. J., “Efficient Split Linear Multistep Methods for Stiff Ordinary Differential Equations,” *SIAM Journal on Scientific and Statistical Computing*, Vol. 10, No. 5, 1989, pp. 990–999.