Output-Based Space-Time Mesh Optimization for Unsteady Flows Using Continuous-in-Time Adjoints

Krzysztof J. Fidkowski*

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

Abstract

We present a method for estimating spatial and temporal numerical errors in scalar outputs of unsteady fluid dynamics simulations using continuous-in-time adjoint solutions and general time-integration methods. A continuous formulation decouples the primal and adjoint temporal discretizations and allows for 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. The output error is still estimated through an adjoint-weighted residual, which takes the form of a space-time integral. Separate temporal and spatial error estimates arise from projection of the adjoint to semi-refined spaces. These estimates drive adaptive refinement, which first requires a calculation of the appropriate cost distribution between the spatial and temporal discretizations. The adaptive mechanics consist of uniform temporal refinement/coarsening, and several localized spatial refinements: order adaptation, hanging-node refinement, and unstructured mesh optimization. Results for scalar advection-diffusion and for the compressible Navier-Stokes equations demonstrate the effectivity of the error estimates, the efficiency of the adaptive refinement, independence of optimized meshes to the starting mesh, and the importance of high-order spatial and temporal approximations.

Keywords: Discontinuous Galerkin, Output Error Estimation, Continuous Adjoint, Unsteady Adaptation, Mesh Optimization

1. 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. Often not all features of the flow need to be well-resolved to accurately predict the scalar output, though sometimes the regions requiring resolution cannot be intuitively predicted. Robust identification of the important regions of the computational domain is possible through the solution of a linear adjoint system, which yields the sensitivity of the output of interest to residual source perturbations. Adaptive refinement schemes can then be designed to target regions where both residual perturbations and the output sensitivity are large, through an adjoint-weighted residual. Much work has been done in this area for steady problems, typically with finite-volume and finite-element methods [1, 2, 3, 4, 5, 6, 7, 8].

Unsteady problems pose additional implementation challenges and computational costs for output-based methods, namely in the solution of the unsteady adjoint equation, which, as a sensitivity problem, must be marched backwards in time. For nonlinear problems, the requisite linearizations also require the state at each time, and this is typically saved to disk at a cost of storage and computational time. In spite of these costs, output-based adaptive methods have been extended to unsteady problems, with various adaptation mechanics, including static-mesh, dynamic-mesh, space-only, and combined space-time refinement [9, 10, 11, 12, 13, 14, 15, 16, 17, 18].

*Corresponding author

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

A variational 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, 16, 13, 17, 18]. 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 and generally lower computational cost compared to variational time-marching 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, 15], 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, and how to optimize multi-dimensional spatial meshes to efficiently reduce the spatial output error. The time history of the error indicator provides sufficient information to drive dynamic mesh optimization, but presently we only consider the error in marginalized form, for driving static spatial mesh adaptation. Dynamic mesh and order refinement mechanics are left for future work.

In the remainder of this paper we first introduce our discretization (Section 2), 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 3. Section 4 outlines the error localization and space-time mesh optimization procedure. Finally, Section 5 presents results that demonstrate the adaptive method for the compressible Navier-Stokes equations.

2. Governing Equations and Discretization

We simulate systems governed by transport equations of the form

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

which encompasses scalar advection-diffusion and the compressible Navier-Stokes equations. For the latter, the conservative state vector is $\mathbf{u} \in \mathbb{R}^s = [\rho, \rho\vec{v}, \rho E]$, where s is the state rank, equal to four in two dimensions, ρ is the density, $\vec{v}$ is the velocity, and ρE is the total energy [19]. The total flux can be decomposed into convective and viscous components, via $\vec{\mathbf{F}}(\mathbf{u}, \nabla\mathbf{u}) = \vec{\mathbf{F}}^{\text{conv}}(\mathbf{u}) + \vec{\mathbf{F}}^{\text{visc}}(\mathbf{u}, \nabla\mathbf{u})$. We discretize this equation in space and time separately, using a semi-discrete approach, as outlined in this section.

2.1. Spatial Discretization

We employ a discontinuous Galerkin (DG) finite-element method in space. Denote by T_h the set of N_e 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 between 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 (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 solver [20] for the convective flux, and the second form of Bassi and Rebay (BR2) [21] for the viscous flux. Choosing a basis for the test and trial spaces yields a system of ordinary differential equations,

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

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.

2.2. Temporal Discretization

We consider general marching schemes for advancing the system of ODEs in (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) and implicit Runge-Kutta methods (DIRK3, DIRK4, ESDIRK5). Appendix A presents the details of these schemes.

2.3. Continuous-in-Time Adjoint

Consider an unsteady output of the form

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

where J and J_T are functionals 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, $\boldsymbol{\Psi}(t)$, is the sensitivity of $\bar{J}$ to source perturbations in the unsteady residual, $\bar{\mathbf{R}}$, in (2). To derive the equation for $\boldsymbol{\Psi}(t)$, we define a Lagrangian

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

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

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

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),

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

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

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

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 (6) in a form similar to the primal,

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

where $\mathbf{R}^{\boldsymbol{\Psi}}$ is the adjoint residual.

2.4. Temporal Reconstruction

For nonlinear problems, the adjoint residual in (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, for which 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 (2), (8) for the adjoint, we can evaluate the state slope,

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

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 (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.

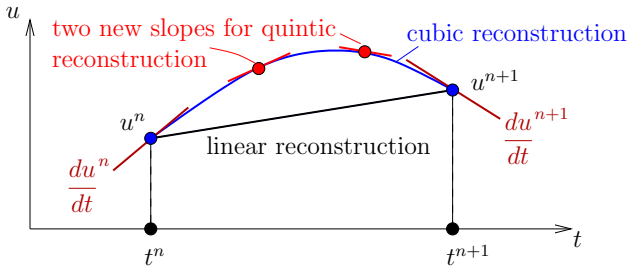


Figure 1: Illustration of solution reconstruction on a time interval. High-order approximations can be constructed using slope information readily available from residual evaluations.

Furthermore, we can gain additional accuracy by applying (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 (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 E is defined via $E^2 = \frac{1}{\Delta t}\int_0^{\Delta t}(u - u_{\text{exact}})^2\, dt$. The results show optimal fourth-order convergence for the cubic reconstruction, and sixth-order convergence for the quintic reconstruction with at least one iteration.

	Cubic		Quintic-0		Quintic-1		Quintic-2	
Δt	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).

3. 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, $\mathbf{\Psi}(t)$, we could use (4) to estimate the error in $\bar{J}$,

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

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 approximation order of the DG discretization by one. The final form of the error estimate is

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

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: $p \to p+1$. In the temporal domain, this involves a sufficiently-accurate reconstruction over the time interval, e.g. cubic for DIRK3 or DIRK4. The integral in (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.

4. 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 and coarsening are applied.

4.1. Error Localization

The error estimate in (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

$$\delta\bar{J} \approx -\int_0^T \mathbf{\Psi}_h^{\mathrm{T}}\bar{\mathbf{R}}_h(\mathbf{U}_h^H)\,dt, = \sum_{n=1}^{N_t}\sum_{e=1}^{N_e} \underbrace{\int_{t^{n-1}}^{t^n} -\mathbf{\Psi}_{h,e}^{\mathrm{T}}\bar{\mathbf{R}}_{h,e}(\mathbf{U}_h^H)\,dt}_{\mathcal{E}_e^n}, \tag{12}$$

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 $\mathcal{E}_e^n$, the error contribution of element e during time step n. The temporal error contribution is estimated by using a spatially down-projected adjoint [22],

$$\mathcal{E}_e^{n,\mathrm{time}} = -\int_{t^{n-1}}^{t^n} \left(\mathbf{I}_H^h\mathbf{\Psi}_{h,e}\right)^{\mathrm{T}}\bar{\mathbf{R}}_{H,e}(\mathbf{U}_H)\,dt, \tag{13}$$

where $\mathbf{I}_H^h$ is a least-squares spatial projection operator 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 defined as $\mathcal{E}_e^{n,\mathrm{space}} \equiv \mathcal{E}_e^n - \mathcal{E}_e^{n,\mathrm{time}}$. The aggregated temporal and spatial errors are then given by the sum over elements and time slabs of the respective contributions,

$$\delta\bar{J}^{\mathrm{time}} = \sum_{n=1}^{N_t}\sum_{e=1}^{N_e}\mathcal{E}_e^{n,\mathrm{time}}, \quad \delta\bar{J}^{\mathrm{space}} = \sum_{n=1}^{N_t}\sum_{e=1}^{N_e}\mathcal{E}_e^{n,\mathrm{space}} = \delta\bar{J} - \delta\bar{J}^{\mathrm{time}}. \tag{14}$$

For adaptation, we use conservative estimates of the localized errors, in order to avoid noise caused by coincidental cancellations of errors. These are obtained by expressions similar to those for $\delta\bar{J}^{\text{time}}$ and $\delta\bar{J}^{\text{space}}$, but with absolute values:

$$\epsilon^{\text{time}} \equiv \sum_{n=1}^{N_t} \left| \sum_{e=1}^{N_e} \mathcal{E}_e^{n,\text{time}} \right|, \quad \epsilon^{\text{space}} \equiv \sum_{e=1}^{N_e} \epsilon_e^{\text{space}}, \quad \epsilon_e^{\text{space}} = \left| \sum_{n=1}^{N_t} \mathcal{E}_e^{n,\text{space}} \right|. \tag{15}$$

4.2. Space-Time Cost Allocation

The localized error estimates guide refinement or coarsening of the spatial and temporal discretizations. As a first step, we must decide how many degrees of freedom should be allocated to the temporal versus the spatial discretization. This is a high-level split of the error, following which the error can be further localized over the time slabs, e.g. for dynamic temporal refinement, and over the elements, e.g. for localized spatial refinement.

Splitting the error into spatial and temporal components relies on two models:

1. A cost model that ties spatial and temporal refinement to changes in the expense of the simulation.
2. An error model that dictates how the spatial and temporal errors behave with refinement.

Several cost models are available, including ones based on degrees of freedom, operation counts, or the system size measured by the number of nonzeros in the residual Jacobian matrix. For the present work, we use a simple measure that consists of the total number of space-time degrees of freedom. This measure treats equally the choices of increasing spatial mesh resolution and of adding more time steps, and it is not specific to the type of spatial or temporal discretization used. Consideration of more sophisticated, discretization-specific, cost models is left for future work.

For static spatial meshes, the total number of space-time degrees of freedom, C, is given by the product of the spatial degrees of freedom, C^{space}, and the temporal degrees of freedom, C^{time},

$$C \equiv C^{\text{space}} C^{\text{time}}, \quad C^{\text{space}} \equiv \sum_{e=1}^{N_e} n(p_e), \quad C^{\text{time}} \equiv N_t n_r, \tag{16}$$

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 per time step. In this equation, the number of spatial degrees of freedom is given by the sum of the degrees of freedom in each element. The number of temporal degrees of freedom is the number of time steps times n_r. For non-variational time-integrators, the value of n_r is set to the number of system solves: 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 an additive model of the spatial and temporal errors,

$$|\delta\bar{J}| = |\delta\bar{J}^{\text{space}}| + |\delta\bar{J}^{\text{time}}|. \tag{17}$$

This choice was made for simplicity, and more sophisticated models could be considered that account for coupling of spatial and temporal errors. Further, we assume an a priori relationship between the spatial and temporal errors and their respective degrees of freedom,

$$|\delta\bar{J}^{\text{space}}| \propto (C^{\text{space}})^{-(p+1)/d}, \quad |\delta\bar{J}^{\text{time}}| \propto (C^{\text{time}})^{-(r+1)}, \tag{18}$$

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. The conservative choice of $p + 1$ [23] for the spatial convergence rate, as opposed to the asymptotically ideal order doubling for many outputs in adjoint-consistent discretizations, yields more aggressive spatial refinement that accelerates adaptive convergence for under-resolved meshes. We note that these a priori rate estimates need not be perfect, as areas of high error will be targeted over multiple adaptive iterations. In addition, spatial singularities are not expected to significantly deteriorate the rates due to localized spatial refinement [24].

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 unadapted mesh. The corresponding changes in the errors are, from (18),

$$|\delta \bar{J}^{\text{space}}| = |\delta \bar{J}_0^{\text{space}}| (f^{\text{space}})^{-(p+1)/d}, \quad |\delta \bar{J}^{\text{time}}| = |\delta \bar{J}_0^{\text{time}}| (f^{\text{time}})^{-(r+1)}. \tag{19}$$

To relate f^{space} and f^{time}, we require equal marginal error to cost ratios [25], $\lambda^{\text{space}} = \lambda^{\text{time}}$, where

$$\lambda^{\text{space}} \equiv \frac{\partial(\delta \bar{J})}{\partial f^{\text{space}}} \left[\frac{\partial C}{\partial f^{\text{space}}} \right]^{-1}, \quad \lambda^{\text{time}} \equiv \frac{\partial(\delta \bar{J})}{\partial f^{\text{time}}} \left[\frac{\partial C}{\partial f^{\text{time}}} \right]^{-1}. \tag{20}$$

The equality of these ratios provides one equation for two variables. To enable a solution for f^{space} and f^{time}, we impose 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.

4.3. Mesh Optimization

Given the cost allocations for temporal and spatial discretizations from the previous sections, we can individually adapt these discretizations. In time, which is one dimensional, we employ uniform refinement, where the factor f^{time} changes N_t and all time steps are of the same size, $\Delta t = T/N_t$. In space, however, which can be multidimensional, we employ localized mesh adaptation that depends on the chosen mechanics.

For order adaptation and hanging node-refinement, we must specify which elements need to be refined. This is done in a greedy fashion by sorting elements according to the spatial error indicator, $\epsilon_e^{\text{space}}$ in (15), and choosing elements with the highest error for refinement until the desired growth fraction, f^{space}, is met [16]. When $f^{\text{space}} < 1$, elements with the lowest error are targeted for coarsening until the desired number of degrees of freedom is reached. Coarsening is only employed for order adaptation, as it is not supported in our hanging-node mesh refinement implementation.

In this work we also consider a more sophisticated spatial adaptation method, unstructured mesh optimization through error sampling and synthesis (MOESS) [25]. This method has been extended to spatial-only adaptation for unsteady flows [26, 27], and in the present work we apply it to the spatial portion of our combined space-time adaptation scheme. In addition, we extend a simple error sampling technique, which we have previously introduced for steady-state calculations [28], to unsteady simulations. We now briefly review this method and discuss its modifications.

In MOESS, the optimal spatial mesh is sought using a metric-based approach, in which both the size and shape of the elements are modified in order to equidistribute the marginal error-to-cost ratio of refinement. This process requires models for how the cost and error change with refinement of the metric field. A degrees of freedom count provides an adequate cost model, and at a fixed approximation order, the cost can be directly related to element size. On the other hand, the error model is more difficult to determine, as it can vary among elements due to pre-asymptotic behavior or the presence of singularities. The error model is therefore obtained empirically by assuming a general form and then fitting free parameters to error samples obtained by selected refinements of the element. Specifically, the general form of the spatial error for element e is

$$\epsilon_e^{\text{space}} = \epsilon_{e0}^{\text{space}} \exp\left[\text{tr}(\mathcal{R}_e \mathcal{S}_e)\right], \tag{21}$$

where $\epsilon_{e0}^{\text{space}}$ is the original spatial error on the element, $\mathcal{S}_e$ is a proposed metric step matrix that modifies the original metric $\mathcal{M}_0$ to yield a new metric $\mathcal{M} = \mathcal{M}_0^{\frac{1}{2}} \exp(\mathcal{S}) \mathcal{M}_0^{\frac{1}{2}}$ [29], and $\mathcal{R}_e$ is a symmetric *rate tensor*. This rate tensor is determined separately for each element through an error sampling procedure in which the element is refined in different configurations and the resulting changes to the error are estimated.

The error sampling procedure is similar to the steady-state case [28]. One element at a time, we consider several refinement options, indexed by i. Figure 2 shows the four options for a triangle. For each refinement option, we

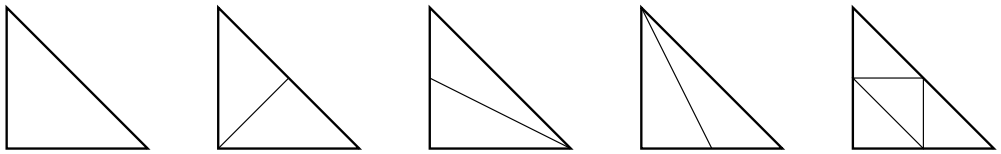


Figure 2: Four refinement options for a triangle. Each one is considered implicitly during error sampling, though the elements are never actually refined.

estimate $\Delta\epsilon_{ei}^{\text{space}}$, which is the spatial error uncovered by refining with option i. We do this by an unsteady adjoint-weighted residual,

$$\Delta\epsilon_{ei}^{\text{space}} \equiv \int_0^T \left| (\delta\boldsymbol{\psi}_{hi,e})^{\mathrm{T}} \bar{\mathbf{R}}_{h,e}(\mathbf{U}_h^H) \right| dt, \tag{22}$$

where $\delta\boldsymbol{\psi}_{hi,e}$ is the coarse-space (order p, assumed the same for all elements) adjoint on refinement option i projected into the fine space (order $p+1$) on the original unrefined element. The motivation for (22) is that using the adjoint on a refined element, but still at the same approximation order p, in an adjoint-weighted residual calculation will give the error estimate relative to refinement option i. We call this the error uncovered by refinement option i. The δ on the adjoint indicates that, in the case of unsteady calculations, we use the fine-space adjoint with the coarse-space (spatial) projection removed. This is because the coarse-space adjoint-weighted residual still yields nonzero error, but this error arises from the temporal discretization. We do not actually refine the element to compute the adjoint for a given option. Instead, we use a projection approach [28], under which

$$\delta\boldsymbol{\psi}_{hi,e} = \mathcal{T}_i \delta\boldsymbol{\psi}_{h,e}, \tag{23}$$

where $\delta\boldsymbol{\psi}_{h,e} \equiv \boldsymbol{\psi}_{h,e} - \mathbf{I}_h^H \mathbf{I}_H^h \boldsymbol{\psi}_{h,e}$ is the fine-space adjoint with its coarse-space projection removed ($\mathbf{I}_H^h$ is a least-squares projection from $p+1$ to p), and $\mathcal{T}_i$ is a projection operator that converts an order $p+1$ field into its approximation when restricted to order p on refinement option i. This operator is calculated through a sequence of projections, starting from order $p+1$ on the original element, to order $p+1$ on the refined element, down to order p on the refined element, and then up to order $p+1$ on the original element. The resulting expression is [28]

$$\mathcal{T}_i = \left[M_0(\phi_0^{p+1}, \phi_0^{p+1}) \right]^{-1} \sum_{k=1}^{n_i} \mathcal{T}_{ik}, \tag{24}$$

$$\mathcal{T}_{ik} = M_k(\phi_0^{p+1}, \phi_k^p) \left[M_k(\phi_k^p, \phi_k^p) \right]^{-1} M_k(\phi_k^p, \phi_k^{p+1}) \left[M_k(\phi_k^{p+1}, \phi_k^{p+1}) \right]^{-1} M_k(\phi_k^{p+1}, \phi_0^{p+1}). \tag{25}$$

In these equations, n_i is the number of sub-elements in refinement option i, k is an index over these sub-elements, ϕ_k^p, ϕ_k^{p+1} are order p and $p+1$ basis functions on sub-element k, ϕ_0^p, ϕ_0^{p+1} are order p and $p+1$ basis functions on the original element, and components of the mass-like matrices are defined as

$$M_k(\phi_l, \phi_m) = \int_{\Omega_k} \phi_l \phi_m d\Omega, \qquad M_0(\phi_l, \phi_m) = \int_{\Omega_0} \phi_l \phi_m d\Omega, \tag{26}$$

where Ω_k is sub-element k and Ω_0 is the original element. Note that $\mathcal{T}_i$ can be calculated for each refinement option i once in reference space and then used for all elements, so that the calculation of $\Delta\epsilon_{ei}^{\text{space}}$ consumes minimal additional cost – and most importantly, no solves or residual evaluations are needed on the refined element.

The estimate of the error uncovered by refinement option i, $\Delta\epsilon_{ei}^{\text{space}}$, is converted to an estimate of the remaining error by subtracting it from the error between order p and $p+1$ on the original element, i.e. $\epsilon_{e0}^{\text{space}}$. Thus, the remaining spatial error after refining element e with option i is estimated as

$$\epsilon_{ei}^{\text{space}} = \epsilon_{e0}^{\text{space}} - \Delta\epsilon_{ei}^{\text{space}}. \tag{27}$$

These error samples are used in a regression to determine a single rate tensor $\mathcal{R}_e$, in (21), for the convergence of the spatial error in element e for the entire unsteady simulation. The remainder of the MOESS method consists of iterations to equidistribute the marginal error to cost ratio, and these are the same as in the steady case.

5. Results

In this section, we demonstrate the effectivity of the error estimates and the efficiency of the adaptation for three test cases, culminating with a compressible Navier-Stokes simulation on a deforming domain.

5.1. Scalar Advection-Diffusion

We first consider a two-dimensional scalar advection-diffusion problem, governed by the following equation:

$$\frac{\partial u}{\partial t} + \nabla \cdot \left(\vec{V}u\right) - \nabla \cdot (\nu \nabla u) = 0,$$

where $\vec{V}$ is the advection velocity and ν is the diffusivity. The computational domain is a square, $(x, y) \in [0, 3]^2$, and the initial condition is a Gaussian concentration profile centered at $(x, y) = (2.1, 0.9)$ with unit amplitude and variance of 1/50, as shown in Figure 3. The velocity field is induced by an irrotational counterclockwise vortex flow centered at $(x, y) = (-1, -1)$ and with a vortex strength of $\Gamma = 3$; i.e. at a point a distance l away from the vortex center, the velocity magnitude is $|\vec{V}| = \Gamma/(2\pi l)$ and the direction is counterclockwise tangential relative to the vortex center. The diffusivity is $\nu = 0.01$, so that the Peclet number based on the vortex strength is $Pe \equiv \Gamma/\nu = 300$. The output is a space-time weighted integral of u,

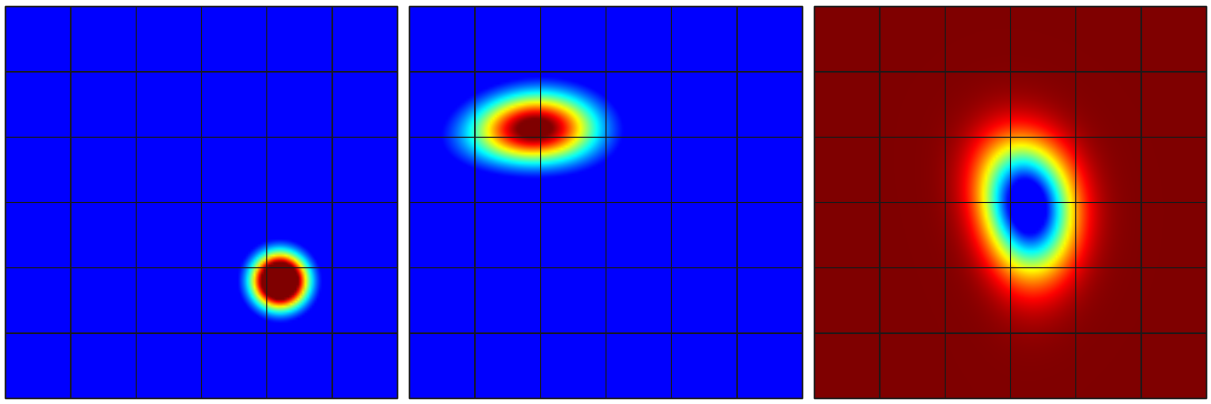
$$\bar{J} = \int_0^T \int_\Omega w_t(t) w_\Omega(\vec{x}) u(\vec{x}, t)\, d\Omega\, dt, \tag{28}$$

where $w_t(t) = \exp[-30(t-1)^2]$ and $w_\Omega = \exp[-10(x-1)^2 - 10(y-2)^2]$. These weights ensure that the adjoint, shown at $t = 0$ in Figure 3, is a smooth function in space and time.

We compare several adaptive strategies for this problem. These include order adaptation (Adapt-p), unstructured metric-based mesh optimization (MOESS), hanging-node adaptation (Hang-Node) and uniform order and element size refinement (Uniform-p, Uniform-h). In all cases, the temporal discretization is uniform: Δt is constant and set adaptively or by uniform subdivision. Time-step subdivision is employed for both Uniform-h and Uniform-p refinements. Two initial meshes are used: 6×6, denoted by m0, and 12×12, denoted by m1. Square elements are subdivided into two triangles for all methods except hanging-node adaptation. The initial number of time steps is denoted by N_t^0, and for order adaptation, the initial uniform order is $p = 1$.

The cost measure is total number of space-time degrees of freedom, as given in (16). For hanging-node refinement, which does not allow spatial coarsening in our implementation, we prescribe a constant growth factor of $f^{\text{tot}} = 1.5$ at each adaptive iteration. For order adaptation and mesh optimization, we vary the growth factor by setting it to 1 for five iterations, followed by one iteration of growth with $f^{\text{tot}} = 2.0$, and then another five iterations of $f^{\text{tot}} = 1.0$. In this manner, the space-time mesh is allowed to equilibrate at a given cost level, prior to an increase in the allowable cost. When presenting results from these runs, we generally show only the data for the last run at $f^{\text{tot}} = 1.0$ prior to a cost increase.

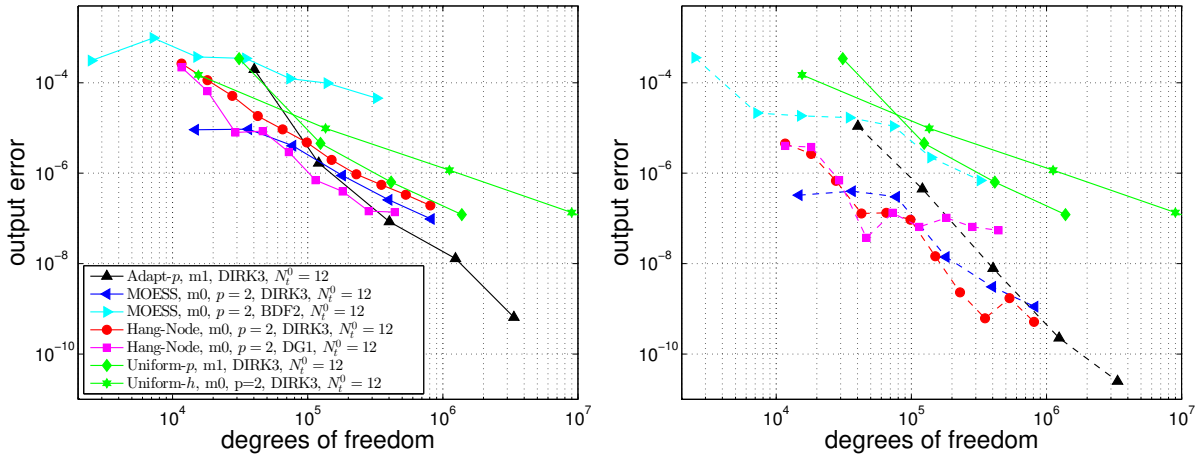
Figure 4 shows the convergence of the output error with total degrees of freedom for various adaptive runs and uniform refinement. The error is measured relative to a truth solution computed on a 48×48 mesh, using $p = 4$ and 100 time steps of ESDIRK5. As this test case exhibits smooth behavior in space and time, high order is advantageous over low order, and hence even uniform order refinement performs well. It is competitive with the (uncorrected) MOESS and hanging-node refinements at $p = 2$. In addition, out of the adaptive strategies, the most efficient one is



(a) Primal at $t = 0$ (b) Primal at $t = 2$ (c) Adjoint at $t = 0$

Figure 3: Scalar advection-diffusion: primal and adjoint solutions on the coarse (m0) quadrilateral mesh. Triangular meshes were obtained by diagonally splitting each quadrilateral.

order adaptation, Adapt-p, with DIRK3 time stepping. MOESS at $p = 2$ and DIRK3 time stepping is not far behind, but we note the importance of time stepping accuracy: using BDF2 yields much slower convergence, as many more degrees of freedom have to be allocated to time stepping to reduce the temporal error. Hanging-node mesh refinement is comparable, though slightly less efficient, compared to the unstructured remeshing of MOESS. In all cases, the fine temporal space consists of higher-order time discretization, using the sequence: BDF2, DIRK3, DIRK4, ESDIRK5.



(a) Error in raw output (b) Error in corrected output

Figure 4: Scalar advection-diffusion: output convergence versus degrees of freedom for different adaptation schemes.

For runs with available error estimates, the corrected output is also shown as a dashed line in Figure 4(b). We see generally very good effectivity of the error estimates, with as much as two orders of magnitude error reduction in some cases. Table 2 presents the numerical values for the effectivity index, defined as the ratio of the estimated error to the actual error, for the first five adaptive iterations of schemes employing the adjoint correction. With the exception of the under-resolved first adaptive iteration when using BDF2, we observe values close to unity for all of the methods. Figure 4 also includes one result from a variational, discrete-adjoint approach, using hanging-node refinement at $p = 2$ and discontinuous Galerkin time stepping with linear temporal approximation (DG1, fine space

Table 2: Scalar advection-diffusion: effectivity indices of the output error estimates for the first five adaptive iterations using different schemes.

	Effectivity index				
Scheme	1	2	3	4	5
Adapt-p, m1, DIRK3, $N_t^0 = 12$	1.06	1.27	1.09	1.02	1.04
MOESS, m0, $p = 2$, DIRK3, $N_t^0 = 12$	1.04	1.04	1.07	1.02	1.01
MOESS, m0, $p = 2$, BDF2, $N_t^0 = 12$	2.16	1.02	1.05	0.95	0.91
Hang-Node, m0, $p = 2$, DIRK3, $N_t^0 = 12$	0.98	0.98	0.99	0.99	0.99
Hang-Node, m0, $p = 2$, DG1, $N_t^0 = 12$	0.98	0.94	1.09	1.00	1.05

DG2) [16]. We see that this strategy performs comparably to its MOESS counterpart, slightly better in the actual output but not as well in the correction. In addition, the degrees of freedom cost measure in the case of DG1 hides the coupling of intra-slab degrees of freedom, which increases the cost of the solver relative to the multi-stage techniques.

Figure 5 shows the breakdown of the cost into spatial and temporal degrees of freedom (`dof`). Here, uniform-h refinement appears as a straight line, as the temporal and spatial `dof` increase by the same factor at every adaptive iteration. At the two extremes of the plot are simulations with mismatched spatial/temporal orders of accuracy: $p = 2$ with BDF2, and $p = 1$ with DIRK4. The former allocates mostly temporal degrees of freedom, while the latter increases spatial degrees of freedom. The adaptive refinements with $p = 2$ and DIRK3 all show similar `dof` allocation breakdowns.

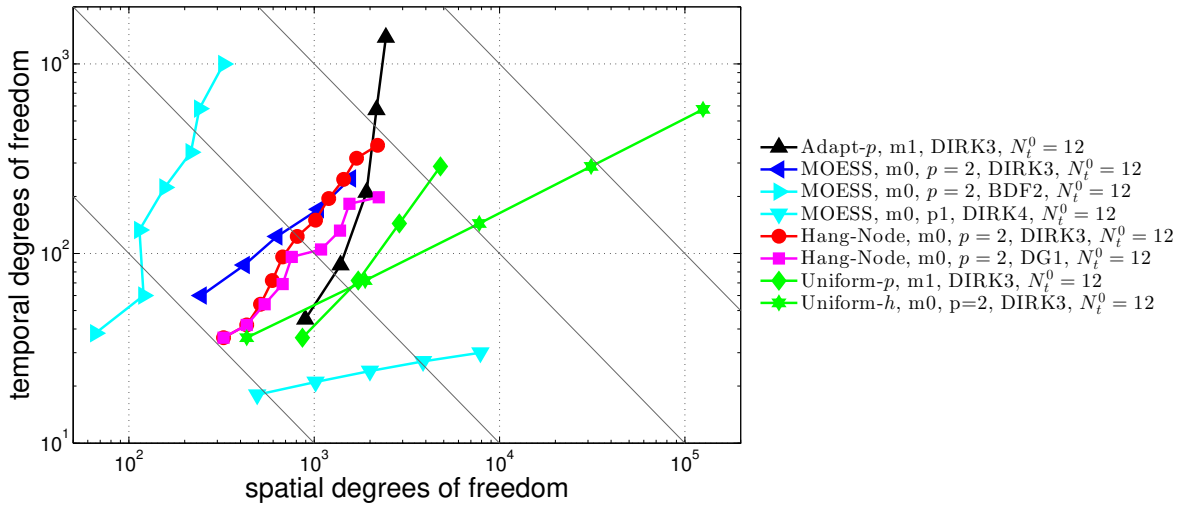


Figure 5: Scalar advection-diffusion: spatial and temporal degree of freedom trajectories for different adaptation schemes.

To test the effectivity of the spatial and temporal error estimates in (14), we plot the spatial and temporal error estimates for several of the adaptive runs in Figure 6. In addition to the estimates, the figure also shows the actual spatial and temporal errors, computed by re-running the primal simulation with finer (higher-order) spatial and temporal discretizations, respectively. We observe good agreement between the estimated and actual errors, and, in this case, similar magnitude temporal and spatial errors on the optimized meshes.

An aspect important to the robustness of space-time mesh optimization is a lack in sensitivity of the optimized mesh to the starting mesh. Figure 7 shows a result of such a test, using MOESS with $p = 2$ and DIRK3. Three different spatial meshes (m0, m1, m2) are used, obtained by successive uniform refinement of the starting 72-element triangular mesh. The initial number of time steps is $N_t^0 = 12$ for all cases. From Figure 7, we see that after a short transient, the three simulations coalesce to the same error versus `dof` curve and the same relative distribution of space-time `dof`. That is, when starting from a very fine spatial mesh, the optimization correctly identifies the leading error

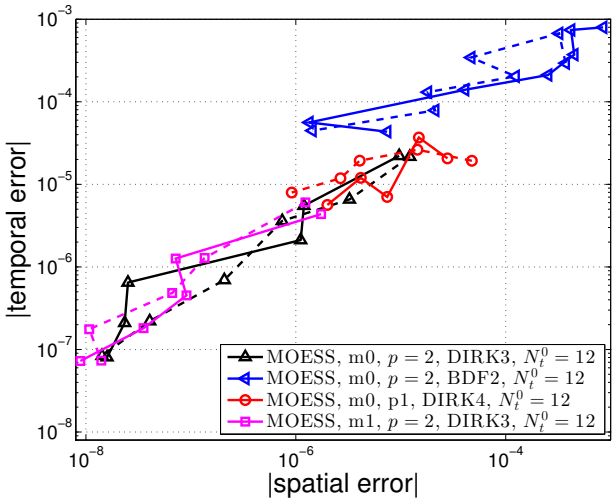
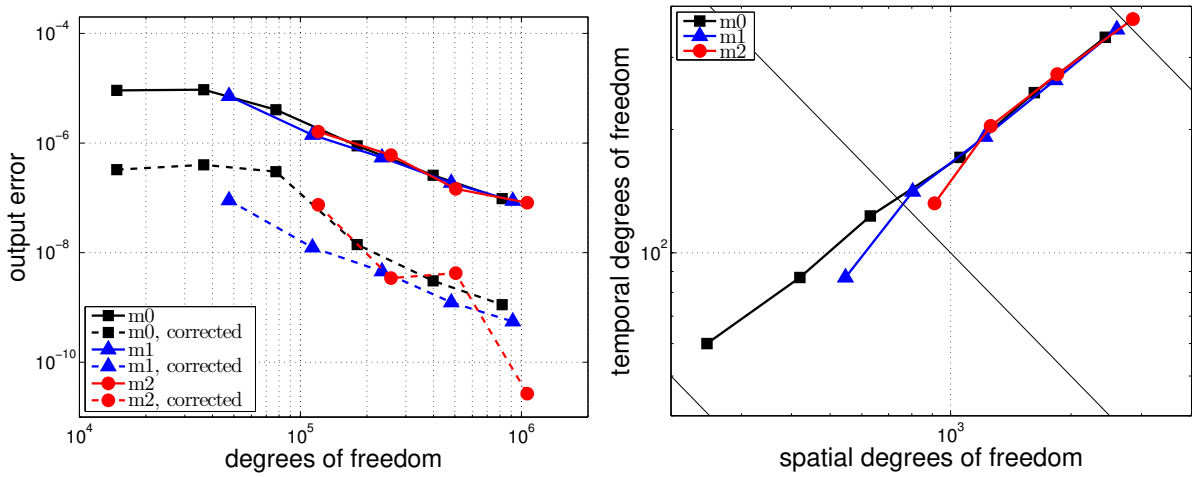


Figure 6: Scalar advection-diffusion: spatial and temporal error trajectories for several adaptation schemes. The dashed lines denote error estimates, while the solid lines are actual errors.

source as temporal and shifts degrees of freedom from the spatial to the temporal discretization.



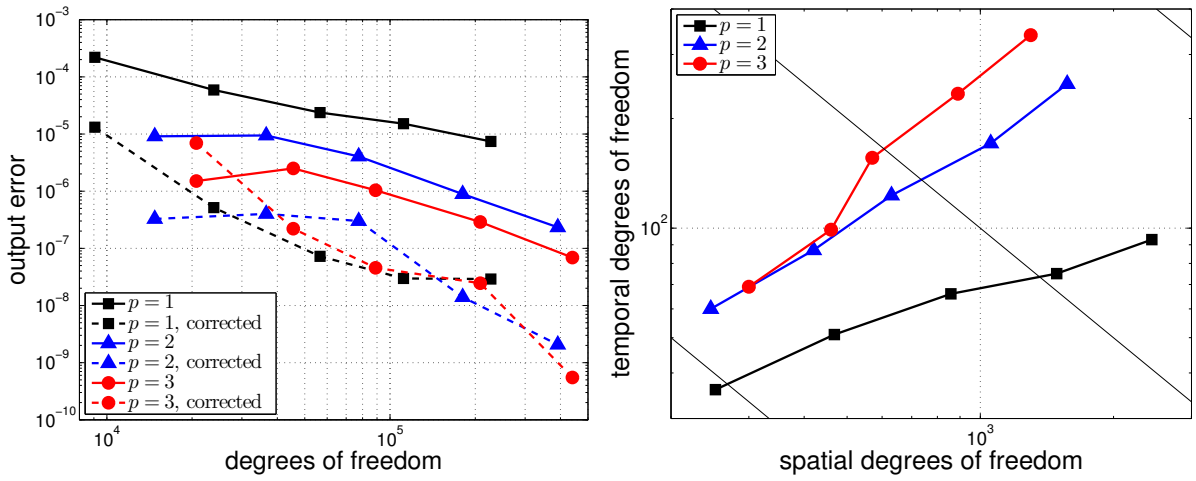
(a) Output convergence

(b) Spatial and temporal dof trajectories

Figure 7: Scalar advection-diffusion: mesh-independence study, using MOESS, $p = 2$, DIRK3, and $N_t^0 = 12$.

We now consider the effect of spatial approximation order, p. Figure 8 shows the error convergence and spatial/temporal `dof` breakdown using $p = 1, 2, 3$ with MOESS and DIRK3 time stepping, starting with $N_t^0 = 12$ time steps. We see that higher order in space yields lower error at a given `dof` and lower allocation of `dof` to the spatial versus temporal discretizations.

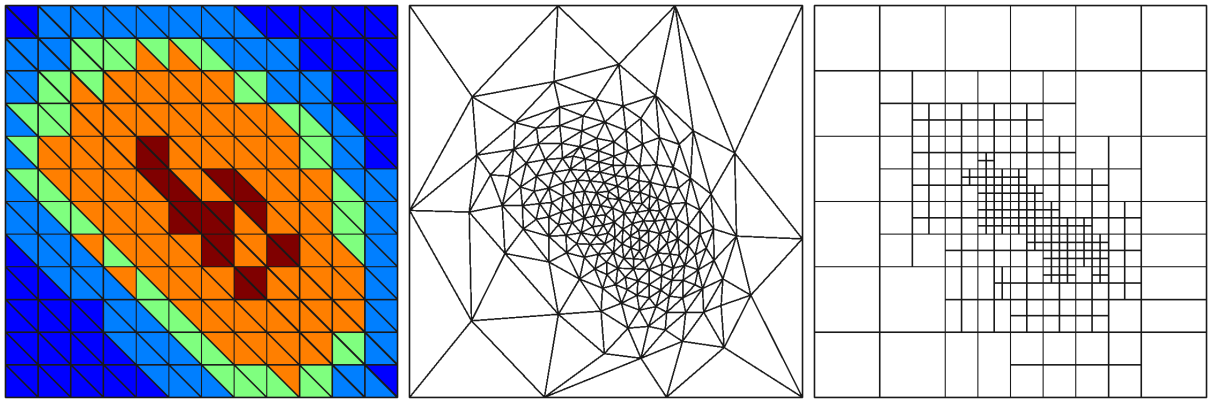
Finally, Figure 9 shows optimized meshes at similar total degrees of freedom for three refinement strategies. We see similar refinement patterns in the spatial meshes: the propagation path of the nonzero scalar distribution in the vortical velocity field from $t = 0$ to $t = 2$. As a fairly large portion of the spatial domain requires resolution, uniform refinement does not fare significantly worse compared to the adaptive refinements.



(a) Output convergence

(b) Spatial and temporal dof trajectories

Figure 8: Scalar advection-diffusion: effect of approximation order p on the performance of MOESS, using DIRK3 and $N_t^0 = 12$.



(a) Adapt-p, $N_t = 96$, dof $= 841k$

(b) MOESS, $p = 2$, $N_t = 112$, dof $= 818k$

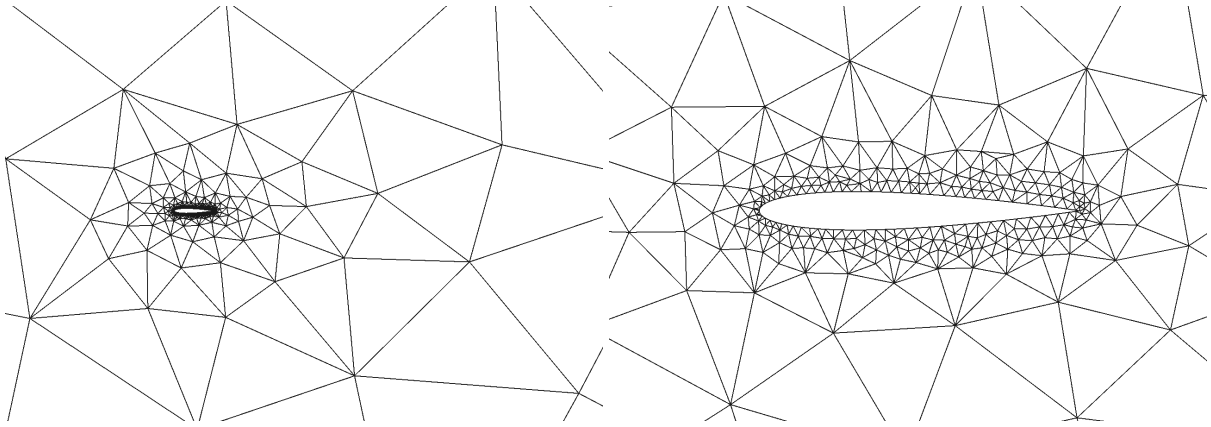
(c) Hang-Node, $p = 2$, $N_t = 123$, dof $= 807k$

Figure 9: Scalar advection-diffusion: adapted meshes at similar degrees of freedom.

5.2. Airfoil Gust Encounter

This test case consists of an encounter between a NACA 0012 airfoil and a vertical gust. The computational domain is a square that extends 100 chord lengths away from the airfoil. Figure 10 shows a coarse, m0, mesh used for the simulations. The initial condition is obtained by first converging a steady-state flow at $Re = 1000$, $M_\infty = 0.4$, $\alpha = 2°$, and with adiabatic wall boundaries. Constant viscosity is assumed, with $Pr = 0.72$. A vertical gust is then superimposed on this solution by perturbing the vertical (perpendicular to the freestream) velocity ahead of the airfoil with a Gaussian function of amplitude $0.3U_\infty$, standard deviation of one chord length, centered five chords upstream of the leading edge, and offset to reach zero perturbation one chord length upstream of the leading edge. Figure 11(a) shows the resulting y-momentum contours. The simulation time runs from $t = 0$ to $t = 15$, where 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

$$[\rho, \rho u, \rho v, \rho E] = \left[1,\ 1,\ 0,\ 0.5+1/[M_\infty^2 \gamma(\gamma-1)]\right]. \tag{29}$$



(a) Initial mesh, 664 elements (b) Initial mesh (zoom)

Figure 10: Gust encounter simulation: initial coarse mesh (m0). Elements adjacent to the geometry are curved using a quartic mapping.

Hence, one time unit is the chord divided by the freestream speed. Figure 11(b) shows the Mach number contours half-way through the simulation. We see that the gust encounter induces flow separation and vortex shedding.

The output of interest is a weighted time-integral of the lift force,

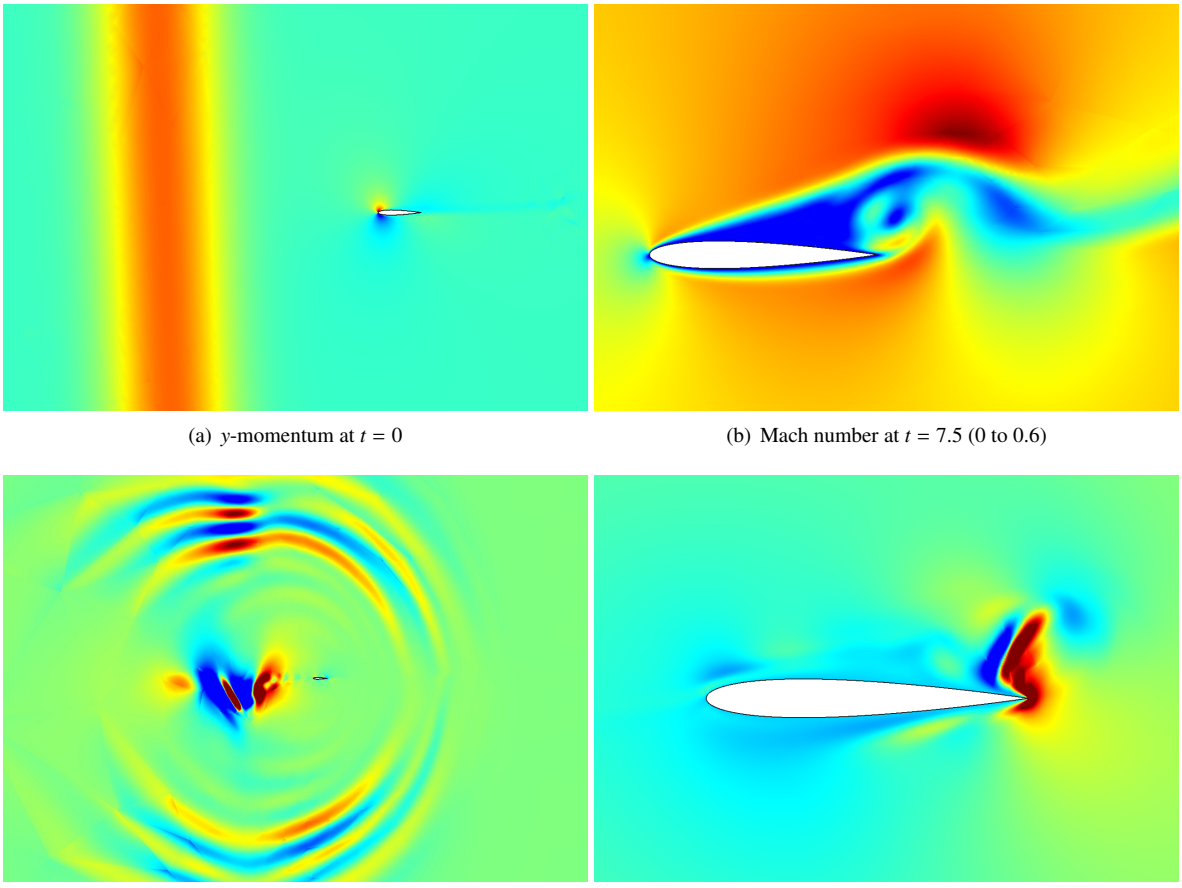
$$\bar{J} = \int_0^T w(t)L(t)\,dt, \tag{30}$$

where $w(t) = \exp(-0.5(t-7.5)^2)$ is a temporal weight function whose role is to make the output smooth in time, and $L(t)$ is the instantaneous lift force on the airfoil; these are both plotted in Figure 12. The truth output for error calculations is computed on a 43,410-element mesh, generated by three quasi-uniform refinements of the initial mesh, using $p = 4$ and 200 time steps of ESDIRK5.

Figure 13 shows the convergence of the output error with `dof` for several adaptive strategies introduced in the previous example. Again, errors after correcting the outputs with the adjoint-weighted residual error estimate are shown as dashed lines. In this case, unstructured mesh optimization via MOESS with $p = 2$ and DIRK3 shows the best performance, both in the raw output and the corrected value. The next best strategy out of the ones tried is order adaptation, Adapt-p, though its errors are about an order of magnitude higher compared to the MOESS, $p = 2$, DIRK3 result. Note that order refinement was started from a finer mesh, m1, obtained by quasi-uniform refinement of m0, in order to improve its performance through a better initial spatial resolution. Note again the importance of accurate time stepping: BDF2 performs markedly worse than DIRK3 and does not exhibit asymptotic behavior, as evidenced by the wrong sign of the error estimate.

To test for sensitivity with respect to the starting mesh, we run the MOESS space-time adaptive strategy using two different initial meshes, m0 and m1. In addition, for the m1 mesh we try two different initial time steps, $N_t^0 = 15$ and $N_t^0 = 4$. Figure 14 shows the results of these runs. We observe very good coalescence of the output convergence plots and the space/time `dof` trajectories, indicating similarity of the optimized space-time meshes.

Figure 15 shows adapted meshes from three strategies, at similar degrees of freedom. We see order adaptation favoring the separated flow region on the airfoil top surface, the wake, and the initial location of the gust, where resolution is required but the initial mesh contains large elements. Adapted meshes generated by MOESS exhibit similar trends, but with clearer targeting of the leading and trailing edges and a portion of the wake where errors could propagate upstream via acoustic modes to affect the unsteady output. In addition, the MOESS-generated meshes begin to show anisotropic refinement, which is to be expected due to the presence of boundary layers and relatively thin wakes for flow at $Re = 1000$.



(a) y-momentum at $t = 0$

(b) Mach number at $t = 7.5$ (0 to 0.6)

(c) y-momentum adjoint at $t = 0$

(d) y-momentum adjoint at $t = 7.5$

Figure 11: Gust encounter simulation: primal and adjoint solutions.

5.3. Pitching and Plunging Airfoil

The final example consists of a NACA 0012 airfoil undergoing combined pitching and plunging motion in viscous flow governed by the compressible Navier-Stokes equations ($\gamma = 1.4, Pr = 0.72$, constant viscosity). The initial condition is obtained from a steady-state solve at free-stream Mach number $M_\infty = 0.2$ and Reynolds number $Re = 1000$. As in the previous example, we use convenient units in which the airfoil chord, c, and the free-stream density and speed are unity, so that the free-stream conservative state vector is given by (29). Full-state boundary conditions are imposed on the farfield boundary, which is 100 chord-lengths away. The coarsest mesh, m0, is the same as in the previous example.

Following a steady-state solve for initialization, the unsteady simulation begins with a pitch and plunge motion in which the vertical displacement is given by $h(t) = t^2(3 - t)/4$, the pitch angle about the $1/3$ chord location is given by $\theta(t) = (\pi/3)t^2(t^2 - 4t + 4)$, and $t \in [0, T], T = 2$. The motion is simulated in an arbitrary Lagrangian-Eulerian formulation [30, 18], using a quintic blending function that smoothly attenuates the motion from rigid body at one chord from the airfoil to no motion fifty chords away.

The output of interest is a weighted time-integral of the lift force on the airfoil, of the form in (30), with temporal weight $w(t) = \exp(-30(t - 1)^2)$. Figure 16 shows the instantaneous lift and weight function for this case. The truth output for error calculations is computed on a 43,410-element mesh, generated by three quasi-uniform refinements of the initial mesh, using $p = 4$ and 200 time steps of ESDIRK5.

Figure 17 shows the primal and adjoint solutions at various times in the simulation. The $t = 1$ plots show the high

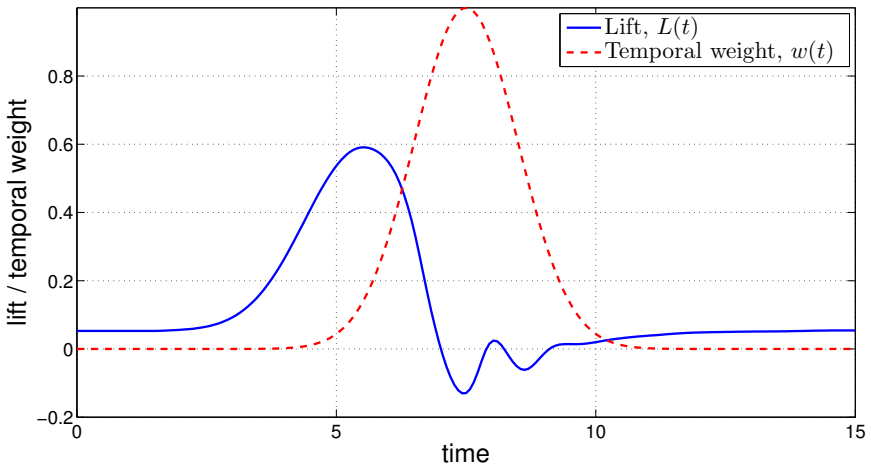


Figure 12: Gust encounter simulation: instantaneous lift and temporal weight function for the time-integral lift output definition.

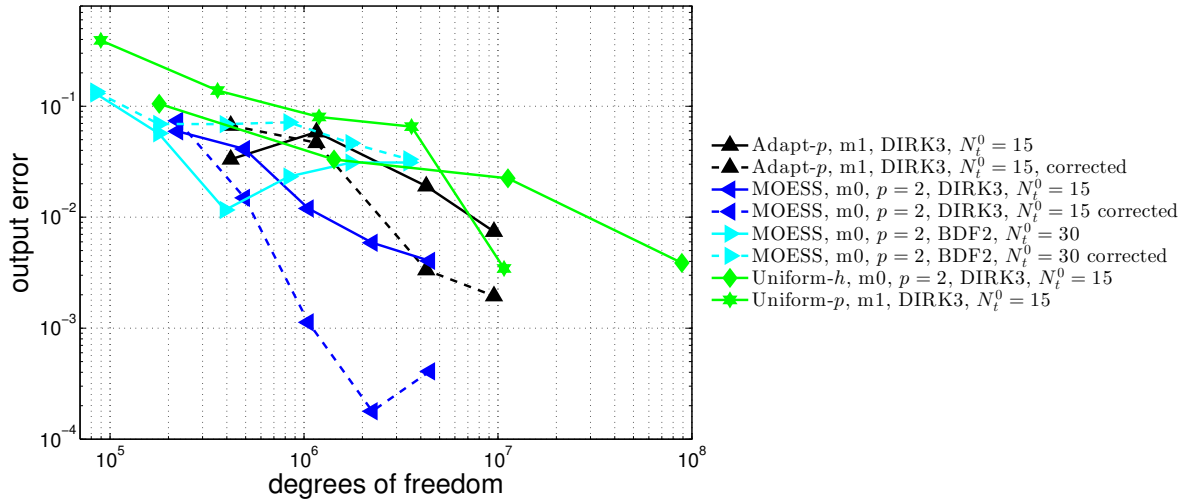
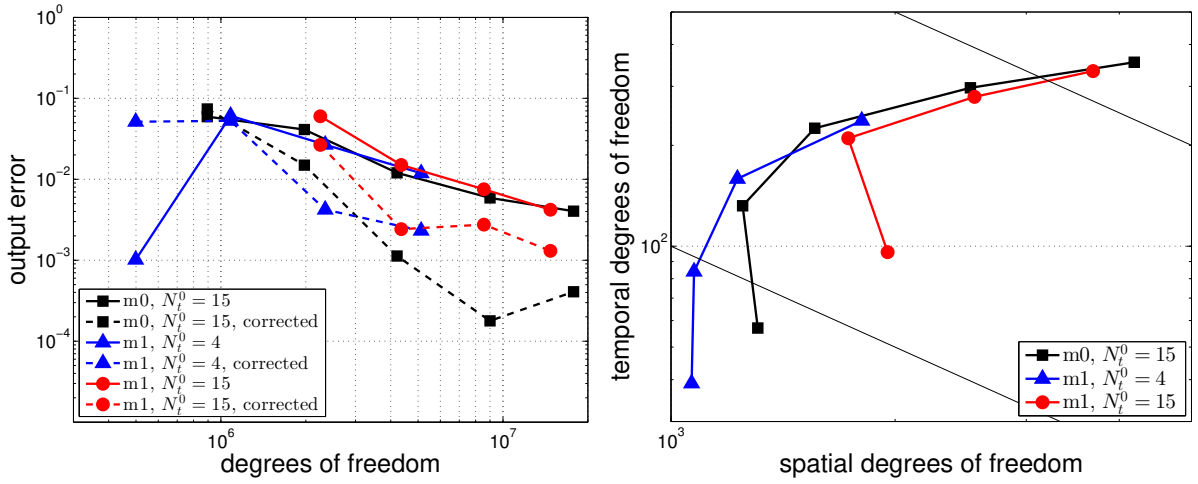


Figure 13: Gust encounter simulation: output convergence versus degrees of freedom for different adaptation schemes.

amplitude of the pitch motion, which leads to a rich solution at $t = 2$. The adjoint fields indicate regions of the flow to which the output is sensitive, which include the vicinity of the airfoil, especially on the aft portion of the upper surface at the initial time.

Figure 18 shows the convergence of the raw and corrected outputs using several adaptation schemes and uniform refinement. Out of the cases tried, the best performance comes from MOESS, with $p = 2$ and DIRK3. Next is order adaptation, followed by uniform refinement, also with DIRK3. Switching the time stepping to BDF2, as shown for MOESS and $p = 2$, significantly deteriorates the accuracy of the output calculation, as more `dof` must be allocated to reduce the temporal error. Figure 18 also shows the outputs corrected with the error estimate, which is generally sufficiently effective to reduce the error by a factor of 2 to 10. The correction in the BDF2 run is quite accurate in this case, reducing the error by over an order of magnitude in some iterations.

Figure 19 shows the adapted meshes at similar degrees of freedom. From the zoomed-out views, we see that the adaptation focuses on a disk around the airfoil, from within which disturbances can propagate to the airfoil and affect the lift integral output. Within this disk, the region aft of the airfoil, which contains the wake, is targeted most for



(a) Output convergence

(b) Spatial and temporal dof trajectories

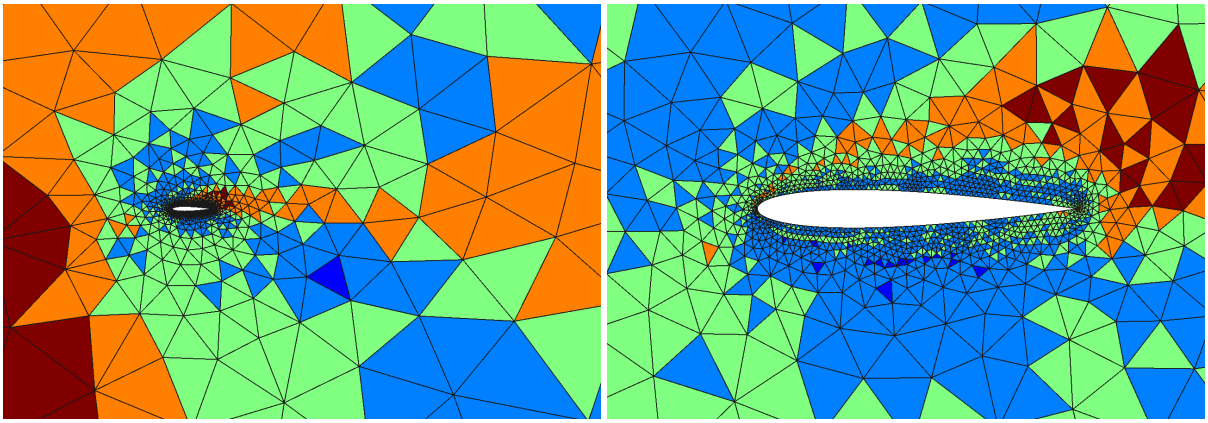
Figure 14: Gust encounter simulation: mesh-independence study, using MOESS and $p = 2$.

refinement. In particular, the intersection of the wake and the disk at approximately five chord lengths away from the airfoil is refined, as this is the region where initial disturbances can propagate and hit the airfoil precisely at the time when the output temporal weight is largest. We also observe refinement emanating from the trailing edge of the airfoil and moving upward, as this is the path through with the trailing edge will move during the simulation. From the near-field plots, we see that, as expected, much of the refinement focuses on the leading edge and especially the trailing edge. Regions above the airfoil are refined more than those below, and the refinement does not extend significantly beyond the proximity of the airfoil. The BDF2 meshes are the coarsest, as many degrees of freedom are allocated to the temporal discretization: $N_t = 589$ time steps versus $N_t = 83$ time steps when using DIRK3. In the order-refined mesh, the mix of orders in the vicinity of the airfoil indicates that the initial mesh is not well-optimized for this problem: the blue elements are too small whereas the orange and red elements are too large, and these occur in close proximity. Ideally, mesh optimization should be combined with order adaptation, and this is a topic of ongoing work.

6. Conclusions

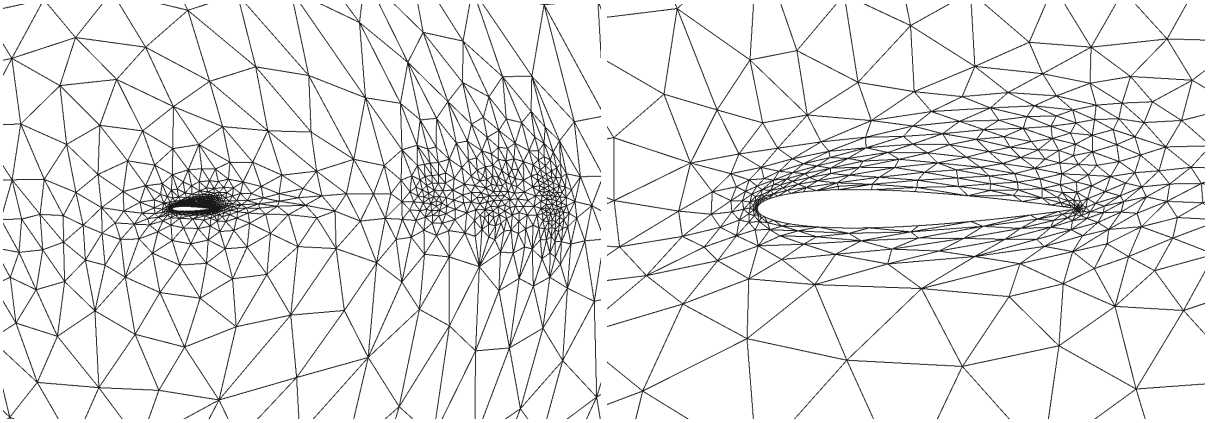
We have presented an approach for estimating numerical errors in outputs of unsteady fluid dynamics simulations using a continuous-in-time adjoint solution and general multistep and multistage time-integration methods. The decoupling of the primal and adjoint temporal discretizations removes constraints from the choice of the adjoint time integration scheme and allows for various standard schemes to be used for the adjoint solve. The adjoint solver, output calculation, and error estimates all employ a scheme-agnostic temporal reconstruction of the primal and adjoint to obtain high-order representations of the primal and adjoint solutions that are consistent with the order of accuracy of the temporal discretization within each time step.

The output error is calculated by an adjoint-weighted residual over the course of the simulation, using an adjoint solution that is finer in space and time. 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 are refined or coarsened to produce an optimal space-time mesh. The allocation of degrees of freedom to space versus time is made by equating marginal error to cost ratios of spatial and temporal refinement, assuming an a priori error convergence model based on the orders in space and time. The temporal discretization is refined uniformly to meet the desired degrees of freedom, whereas the spatial discretization is refined locally, based on the magnitude of the error estimate, using various techniques: order refinement, hanging-node subdivision, and unstructured mesh optimization.



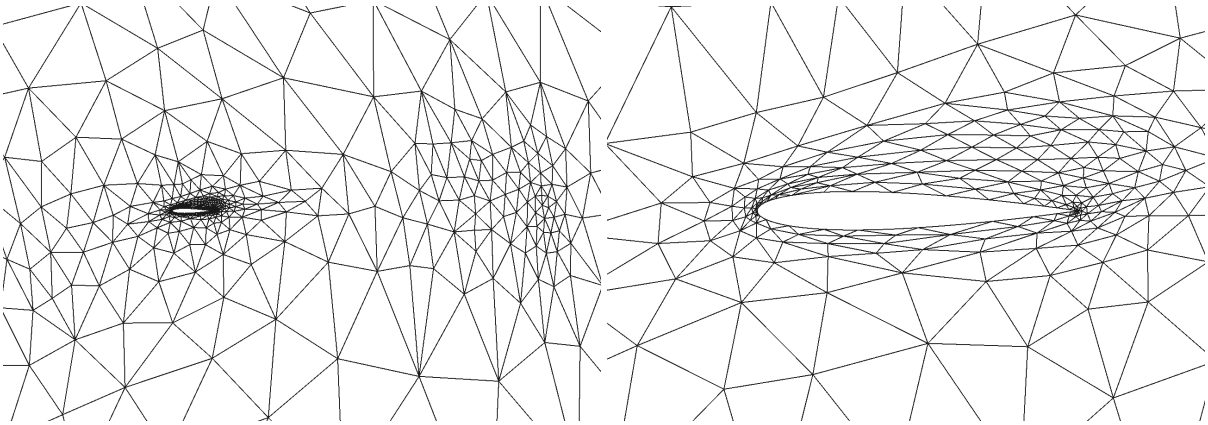
(a) Adapt-p, m1, DIRK3, orders (1-6): $N_t = 71$, $\texttt{dof} = 4.27M$

(b) Adapt-p, m1, DIRK3, orders (zoom)



(c) MOESS, m0, $p = 2$, DIRK3, $N_t^0 = 15$: $N_t = 119$, $\texttt{dof} = 4.45M$

(d) MOESS, m0, $p = 2$, DIRK3, $N_t^0 = 15$ (zoom)



(e) MOESS, m0, $p = 2$, BDF2, $N_t^0 = 30$: $N_t = 495$, $\texttt{dof} = 3.47M$

(f) MOESS, m0, $p = 2$, BDF2, $N_t^0 = 30$

Figure 15: Gust encounter simulation: adapted meshes at similar degrees of freedom.

Results for three test cases demonstrate the accuracy of the error estimates and the relative merits of the various adaptation strategies. A simple scalar test case allows for a head-to-head comparison of multiple strategies, which are

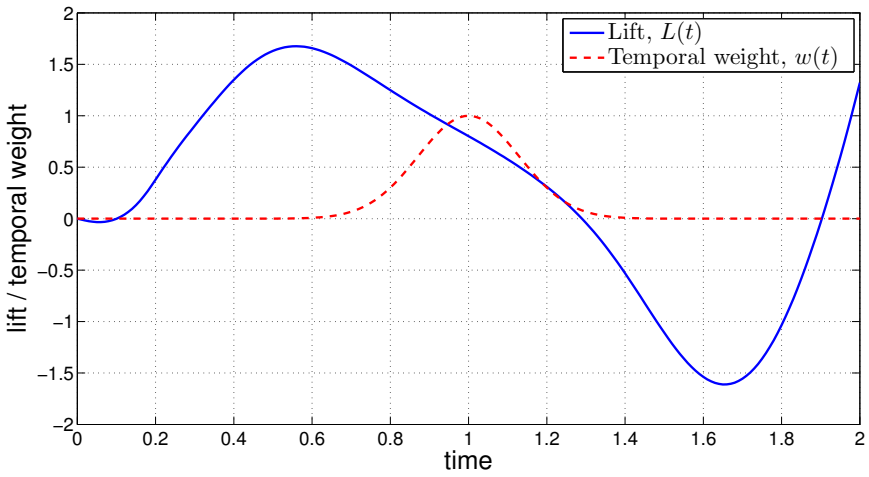
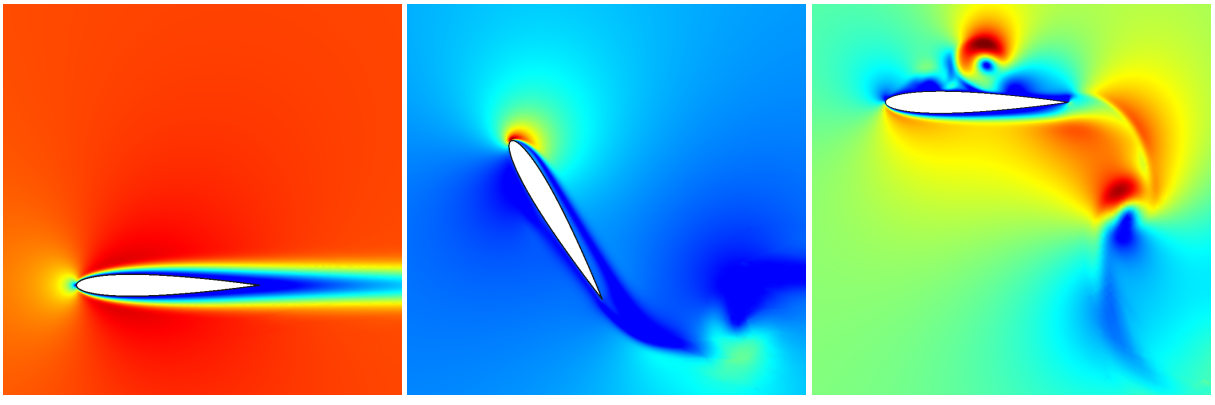
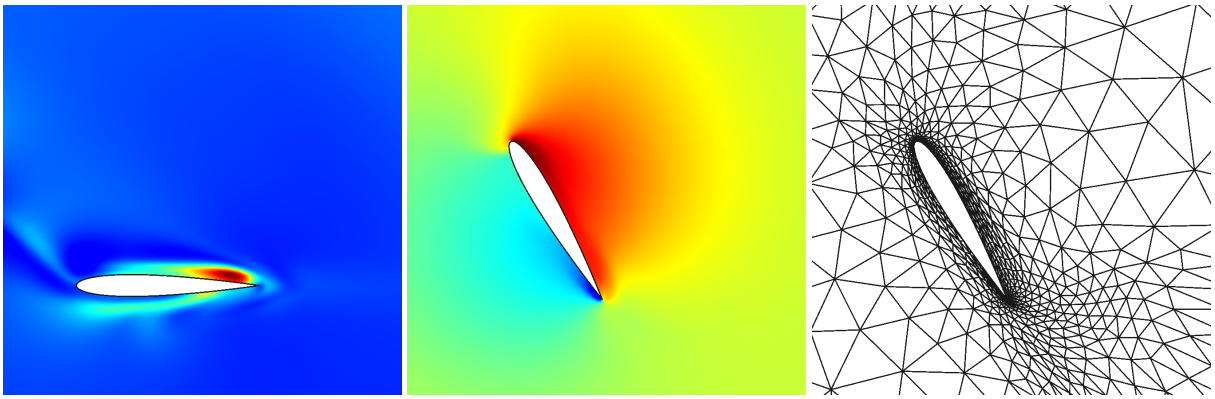


Figure 16: Pitching and plunging airfoil: instantaneous lift and temporal weight function for the time-integral lift output definition.



(a) Mach number at $t = 0$ (0 to 0.25) (b) Mach number at $t = 1$ (0 to 0.7) (c) Mach number at $t = 2$ (0 to 0.4)



(d) Adjoint at $t = 0$ (e) Adjoint at $t = 1$ (f) Mesh at $t = 1$

Figure 17: Pitching and plunging airfoil: primal and adjoint (energy component) solutions at various times.

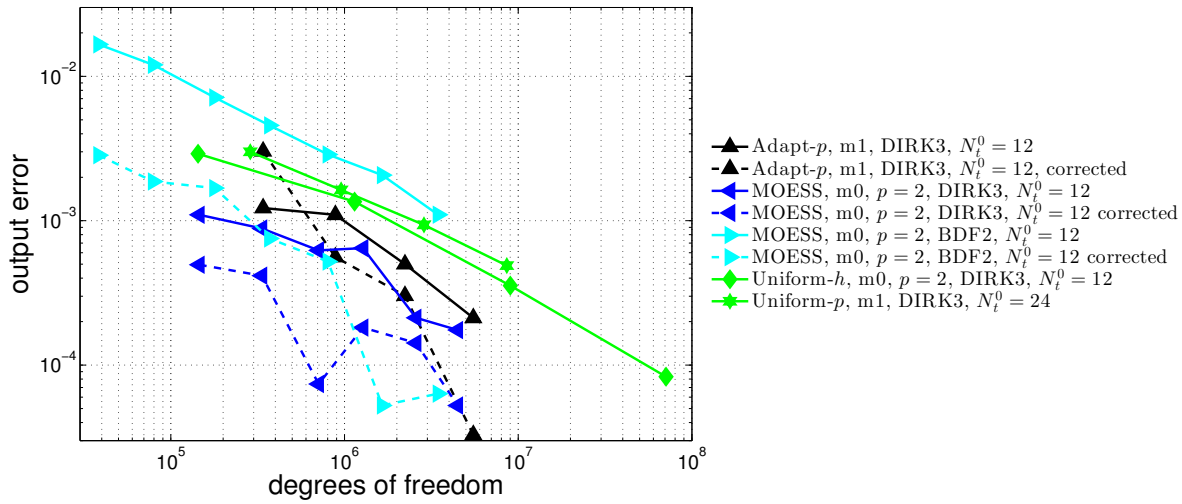


Figure 18: Pitching and plunging airfoil: output convergence versus degrees of freedom for different adaptation schemes.

found to perform similarly for a given choice of spatial approximation order and time-marching scheme. Low-order time marching severely limits accuracy for a given cost, as shown in a comparison between second-order backwards differencing and third-order implicit Runge-Kutta. Plots of the degree-of-freedom allocations show trends consistent with expectations: matched spatial and temporal orders yield similar growth in the spatial and temporal costs, whereas a mismatch of orders forces the adaptive scheme to compensate by allocating more degrees of freedom to the lower-order accurate discretization.

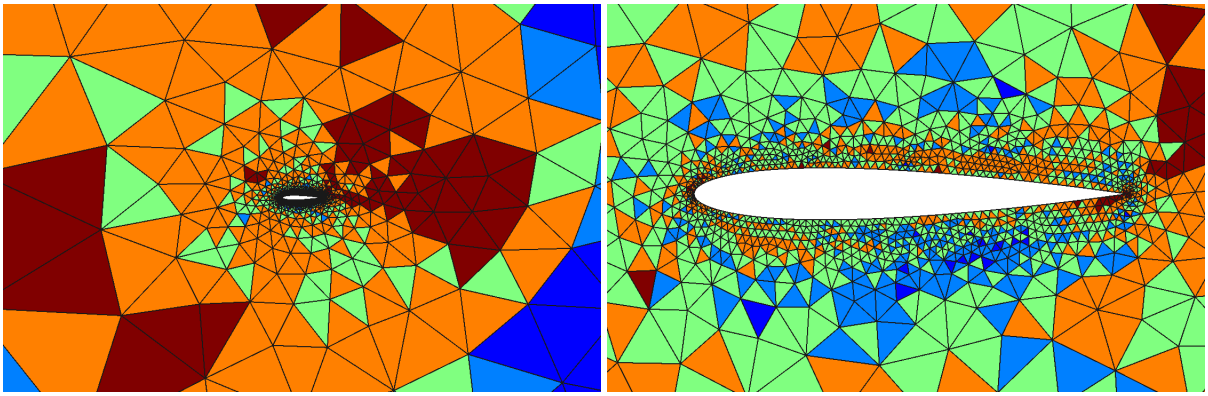
A mesh-independence study for the scalar case indicates that the optimized space-time meshes, at a given degree of freedom cost, are not very sensitive to the starting mesh. Finally, a spatial approximation order study reveals benefits of high-order approximation, e.g. $p = 2$ versus $p = 1$, in the context of unstructured mesh optimization. In these cases, the temporal discretization remained fixed at DIRK3, and even greater benefits are expected if the temporal order were varied as well.

Results for the unsteady compressible Navier-Stokes equations show the applicability of the adaptation strategies to practical systems of equations. In a gust encounter simulation, unstructured mesh optimization yields the best performance out of the schemes tested, and adapted meshes exhibit localized areas of high refinement at the trailing edge and anisotropic elements in regions where the flow exhibits time-averaged anisotropic features. The final example of adaptation for flow undergoing pitching and plunging motion shows that the continuous-in-time adjoint error estimates and mesh adaptation strategies extend to simulations on moving domains, in an arbitrary Lagrangian-Eulerian formulation. In both of these examples, error estimates are shown to be effective at improving the output accuracy, in some cases by over an order of magnitude.

A natural extension of the present work is to combine mesh and order refinement for unsteady flows into one framework, choosing the most efficient approach. Another research direction is dynamic mesh adaptation, through timed order/mesh refinement and node movement. Both of these strategies are expected to improve the adaptation error-cost tradeoff. Finally, we have thus far restricted our presentation to degrees of freedom, rather than computational time, as the cost measure. The reason for this is that the computational time, even in relative terms, depends on the implementation and machine architectures, and we have not tuned requisite aspects of our code, such as the variable order solver or the hanging-node refinement treatment. The degrees of freedom count, on the other hand, provides a deterministic and reasonable metric for method comparison.

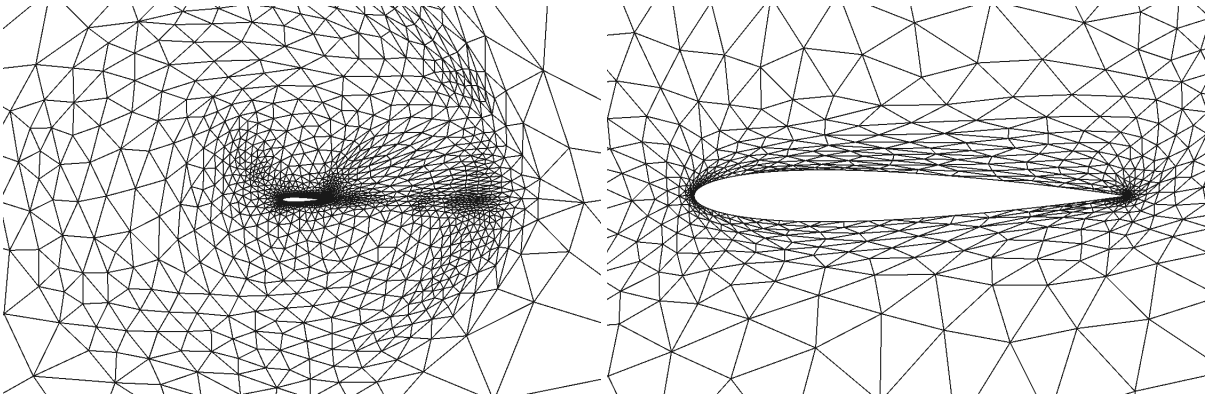
Acknowledgments

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.



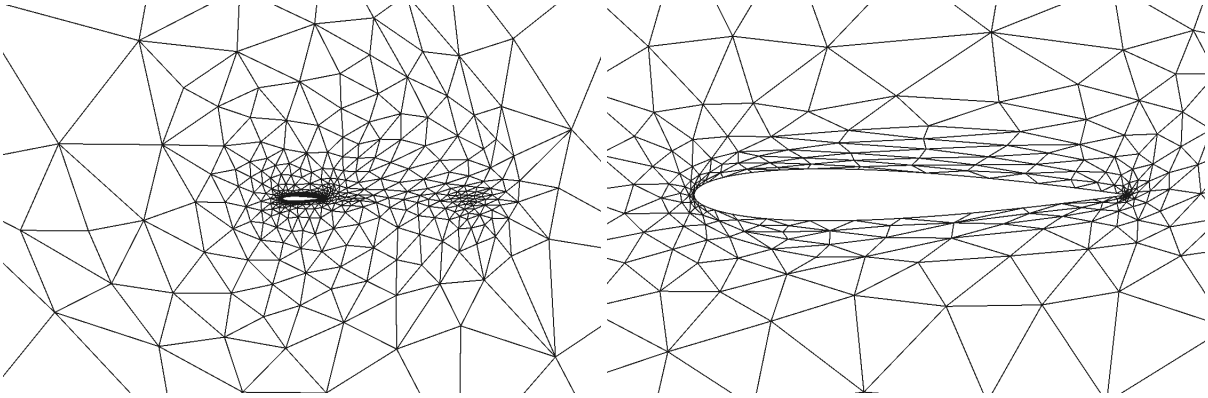
(a) Adapt-p, m1, DIRK3, orders (1-5): $N_t = 65$, $\texttt{dof} = 5.5M$

(b) Adapt-p, m1, DIRK3, orders (zoom)



(c) MOESS, m0, $p = 2$, DIRK3, $N_t^0 = 12$: $N_t = 83$, $\texttt{dof} = 4.5M$

(d) MOESS, m0, $p = 2$, DIRK3, $N_t^0 = 12$ (zoom)



(e) MOESS, m0, $p = 2$, BDF2, $N_t^0 = 12$: $N_t = 589$, $\texttt{dof} = 3.5M$

(f) MOESS, m0, $p = 2$, BDF2, $N_t^0 = 12$

Figure 19: Pitching and plunging airfoil: adapted meshes.

Appendix A. Time Schemes

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

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

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

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

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 are [31]

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

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 $n_{\text{stage}} = 5$ is [32]

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

ESDIRK5 is a seven-stage, fifth-order scheme that takes the same form. Coefficients for this scheme can be found in the work of Kennedy and Carpenter [33].

References

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

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

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

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

[5] S. Sen, K. Veroy, D. Huynh, S. Deparis, N. Nguyen, A. Patera, "Natural norm" a posteriori error estimators for reduced basis approximations, Journal of Computational Physics 217 (2006) 37–62.

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

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

[8] M. Woopen, A. Balan, G. May, J. Schütz, A comparison of hybridized and standard DG methods for target-based hp-adaptive simulation of compressible flow, Computers & Fluids 98 (2014) 3–16.

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

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

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

[12] K. Mani, D. J. Mavriplis, Error estimation and adaptation for functional outputs in time-dependent flow problems, Journal of Computational Physics 229 (2010) 415–440.

[13] V. Carey, D. Estep, A. Johansson, M. Larson, S. Tavener, Blockwise adaptivity for time dependent problems based on coarse scale adjoint solutions, SIAM Journal on Scientific Computing 32 (4) (2010) 2121–2145.
[14] A. Belme, A. Dervieux, F. Alauzet, Error estimation and adaptation for functional outputs in time-dependent flow problems, Journal of Computational Physics 231 (2012) 6323–6348.
[15] B. T. Flynt, D. J. Mavriplis, Discrete adjoint based adaptive error control in unsteady flow problems, AIAA Paper 2012-0078 (2012).
[16] K. J. Fidkowski, Y. Luo, Output-based space-time mesh adaptation for the compressible Navier-Stokes equations, Journal of Computational Physics 230 (2011) 5753–5773. doi:10.1016/j.jcp.2011.03.059.
[17] K. J. Fidkowski, An output-based dynamic order refinement strategy for unsteady aerodynamics, AIAA Paper 2012-77 (2012). doi:10.2514/6.2012-77.
[18] S. M. Kast, K. J. Fidkowski, Output-based mesh adaptation for high order Navier-Stokes simulations on deformable domains, Journal of Computational Physics 252 (1) (2013) 468–494. doi:10.1016/j.jcp.2013.06.007.
[19] M. A. Ceze, K. J. Fidkowski, An anisotropic hp-adaptation framework for functional prediction, American Institute of Aeronautics and Astronautics Journal 51 (2013) 492–509. doi:10.2514/1.J051845.
[20] P. Roe, Approximate Riemann solvers, parameter vectors, and difference schemes, Journal of Computational Physics 43 (1981) 357–372.
[21] F. Bassi, S. Rebay, GMRES discontinuous Galerkin solution of the compressible Navier-Stokes equations, in: K. Cockburn, Shu (Eds.), Discontinuous Galerkin Methods: Theory, Computation and Applications, Springer, Berlin, 2000, pp. 197–208.
[22] K. J. Fidkowski, Output error estimation strategies for discontinuous Galerkin discretizations of unsteady convection-dominated flows, International Journal for Numerical Methods in Engineering 88 (12) (2011) 1297–1322. doi:10.1002/nme.3224.
[23] K. J. Fidkowski, D. L. Darmofal, A triangular cut-cell adaptive method for high-order discretizations of the compressible Navier-Stokes equations, Journal of Computational Physics 225 (2007) 1653–1672. doi:10.1016/j.jcp.2007.02.007.
[24] M. Yano, J. Modisette, D. Darmofal, The importance of mesh adaptation for higher-order discretizations of aerodynamics flows, AIAA Paper 2011-3852 (2011).
[25] M. Yano, An optimization framework for adaptive higher-order discretizations of partial differential equations on anisotropic simplex meshes, Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts (2012).
[26] J. A. Krakos, Unsteady adjoint analysis for output sensitivity and mesh adaptation, Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts (2012).
[27] J. A. Krakos, D. L. Darmofal, Anisotropic output-based mesh optimization for unsteady flows, AIAA Paper 2013-3083 (2013).
[28] K. J. Fidkowski, A local sampling approach to anisotropic metric-based mesh optimization, AIAA Paper 2016–0835 (2016). doi:10.2514/6.2016-0835.
[29] X. Pennec, P. Fillard, N. Ayache, A Riemannian framework for tensor computing, International Journal of Computer Vision 66 (1) (2006) 41–66.
[30] P.-O. Persson, J. Bonet, J. Peraire, Discontinuous Galerkin solution of the Navier-Stokes equations on deformable domains, Computer Methods in Applied Mechanics and Engineering 198 (2009) 1585–1595.
[31] R. Alexander, Diagonally implicit Runge-Kutta methods for stiff ODE's, SIAM Journal on Numerical Analysis 14 (6) (1977) 1006–1021.
[32] N. Rattenbury, Almost Runge-Kutta methods for stiff and non-stiff problems, Ph.D. thesis, The University of Auckland (2005).
[33] C. A. Kennedy, M. H. Carpenter, Additive Runge-Kutta schemes for convection-diffusion-reaction equations, Applied Numerical Mathematics 44 (2003) 139–181.