ECCOMAS Congress 2016
VII European Congress on Computational Methods in Applied Sciences and Engineering
M. Papadrakakis, V. Papadopoulos, G. Stefanou, V. Plevris (eds.)
Crete Island, Greece, 5–10 June 2016

OUTPUT-BASED SPACE-TIME ADAPTATION WITH NON-VARIATIONAL TIME INTEGRATION

Krzysztof J. Fidkowski[1]

[1]University of Michigan
1320 Beal Avenue, Ann Arbor, MI, 48109
e-mail: kfid@umich.edu

Keywords: Error Estimation, Adaptation, Unsteady, Non-Variational, Discontinuous Galerkin.

Abstract. *In this paper we present an output-based method for estimating numerical errors in unsteady simulations. This method applies to both variational and non-variational time integrators, and we focus on the latter. Key to the method is the solution of a continuous-in-time adjoint; that is, the unsteady adjoint system is derived from the semi-discrete form of the primal using a discretization of choice that needs not be the same as the primal. This provides flexibility in choosing the most appropriate and efficient time integration methods for the primal and adjoint problems separately. We also present a general approach for reconstructing the solution within a time interval, which is required for an accurate adjoint solution when using high-order methods. We then show how the output error estimate can be separated into contributions from spatial and temporal discretizations using a spatial down-projection of the adjoint. Finally, we present a space-time mesh adaptation procedure that appropriately targets spatial and temporal errors by setting equal ratios of marginal error to marginal cost of each refinement option, space or time. We demonstrate the error estimation and adaptation method for a simple prototypical problem: unsteady linear advection in one spatial dimension.*

1 INTRODUCTION

In an output-based setting, mesh resolution is automatically dictated by an error estimate of an output of interest. Much work has been done in this area for steady problems with finite volume and finite element methods [13, 2, 9, 16, 15, 12, 6, 17]. However, unsteady problems pose additional challenges and computational costs, namely in the unsteady adjoint solution. Yet output-based adaptive methods have also been explored for such problems, with various mechanics of adaptation, including static-mesh, dynamic-mesh, space-only, and combined space-time [14, 1, 4, 11, 3, 8, 7, 5, 10].

A variational discretization, such as a finite element method in space and time, puts output error estimation on a sound theoretical foundation and simplifies the error calculation and localization [6]. In particular, 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 [14, 1, 7, 5, 10]. 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 or simple to work with, especially if employing 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 methods. There is therefore a need to extend output-based methods to such discretizations, to increase their utility and prevalence, and this is the topic of the present work. Some work in this area has already been done [11, 8], 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, even of high order, and fine-space choices. We also show how to separate the error into contributions of the spatial and temporal discretization.

In the remainder of this paper we first introduce our discretization in Section 2, which is variational in space but could be non-variational in time. In Section 3 we derive a continuous-in-time adjoint solution and show how to use it to estimate temporal errors, and spatial errors when also using spatial refinement. A key part of the adjoint solution and the error estimate is a reconstruction of the primal and adjoint solution in time, for which we present a scheme-independent approach in Section 3.2. Section 3.4 outlines the space-time adaptive procedure, and Section 4 presents results that demonstrate the methods for an advection problem.

2 DISCRETIZATION

We are interested in the simulation of phenomena governed by partial differential equations in space and time. We write the PDE in canonical form as

$$\frac{\partial u}{\partial t} + L(u) = 0, \tag{1}$$

where u is the state, and $L(\cdot)$ is the continuous spatial residual operator. We discretize this equation in space and time separately, using a semi-discrete approach, as outlined in this section.

2.1 Spatial

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 on each element, with no continuity constraints imposed on the approximations on adjacent elements. Formally, for a scalar equation, we write that $u_h \in \mathcal{V}_h$, where $\mathcal{V}_h = \left\{u \in L_2(\Omega) : u|_{\Omega_e} \in \mathcal{P}^p \ \forall \Omega_e \in T_h\right\}$, and $\mathcal{P}^p$ denotes polynomials of order p on an element Ω_e. The weak-form of (1) follows from multiplying the PDE by test functions in the same approximation space, integrating by parts, and coupling elements via common fluxes. Choosing a basis for the test and trial spaces yields a system of ordinary differential equations,

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

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

We consider general marching schemes for advancing the system of ODEs in (2) in time. 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 DG-in-time variational discretization [7] (DG1 and DG2), standard backwards-difference multi-step methods (BDF1 and BDF2), and diagonally-implicit Runge-Kutta methods (DIRK3 and DIRK4). For example, an n_{stage} DIRK method takes the form

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

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 example, a fourth-order accurate scheme (DIRK4) with $n_{\text{stage}} = 5$ has

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

3 OUTPUT ERROR ESTIMATION

In previous work, the author and collaborators have presented output error estimation for unsteady problems using the discrete space-time adjoint and a variational discretization in time. In the present work, to accommodate general time integration methods that do not necessarily possess a temporal weak form, we rely on a continuous-in-time adjoint, as described in this section.

3.1 The continuous-in-time adjoint

Consider a time-integral output of a simulation,

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

where $J(\mathbf{U}, t)$ is a function of the spatial distribution of the state, via the discrete coefficients $\mathbf{U}$, and of time [1]. The continuous-in-time adjoint $\mathbf{\Psi}(t)$ is the sensitivity of $\bar{J}$ to source perturbations in the unsteady residual, $\bar{\mathbf{R}}$, (2). To derive the equation for $\mathbf{\Psi}(t)$, we define a Lagrangian as

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

Integrating the first term in the integral by parts, we have

$$\mathcal{L} = \bar{J} + \left[\mathbf{\Psi} \mathbf{M} \mathbf{U} \right]_0^T + \int_0^T \left[-\frac{d\mathbf{\Psi}^T}{dt} \mathbf{M} \mathbf{U} + \mathbf{\Psi}^T \mathbf{R}(\mathbf{U}) \right] dt. \tag{5}$$

Now requiring stationarity of the Lagrangian with respect to permissible state variations gives

$$\mathcal{L}_\mathbf{U}(\delta\mathbf{U}) = \bar{J}_\mathbf{U}(\delta\mathbf{U}) + \left[\mathbf{\Psi} \mathbf{M} \delta\mathbf{U} \right]_0^T + \int_0^T \left[-\frac{d\mathbf{\Psi}^T}{dt} \mathbf{M} \delta\mathbf{U} + \mathbf{\Psi}^T \mathbf{R}_\mathbf{U}(\delta\mathbf{U}) \right] dt = \mathbf{0}, \tag{6}$$

where the subscript denotes differentiation with respect to the discrete state vector $\mathbf{U}$. The middle term drops out since $\delta\mathbf{U} = 0$ at $t = 0$, and $\mathbf{\Psi} = 0$ at $t = T$ ($\bar{J}$ does not depend on the terminal state). Combining the remaining terms,

$$\int_0^T \left[J_\mathbf{U} - \frac{d\mathbf{\Psi}^T}{dt} \mathbf{M} + \mathbf{\Psi}^T \mathbf{R}_\mathbf{U} \right] \delta\mathbf{U}\, dt = \mathbf{0}. \tag{7}$$

This equation must hold for all $\delta\mathbf{U}$, which forces the term in the square brackets to zero. Transposing this term produces the continuous-in-time adjoint equation,

$$-\mathbf{M} \frac{d\mathbf{\Psi}}{dt} + \mathbf{R}_\mathbf{U}^T \mathbf{\Psi} + J_\mathbf{U}^T = \mathbf{0}, \tag{8}$$

with the terminal condition $\mathbf{\Psi}(T) = \mathbf{0}$. Due to the terminal condition, the adjoint is solved backwards in time, from $t = T$ to $t = 0$. In the present work, we do not restrict the choice of time integration for the adjoint equation to be variational or even the same as the primal. We note that when applying standard time integrators for marching backwards in time, it is useful to define $\tau = T - t$ and to rewrite (8) in a form similar to the primal,

$$\mathbf{M} \frac{d\mathbf{\Psi}}{d\tau} + \underbrace{\mathbf{R}_\mathbf{U}^T \mathbf{\Psi} + J_\mathbf{U}^T}_{\mathbf{R}^\mathbf{\Psi}} = \mathbf{0}, \tag{9}$$

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

[1] More general outputs that involve end-of-time states could also be considered in a straightforward manner.

3.2 Temporal reconstruction

Solving the adjoint using (9) requires evaluating the adjoint residual, which in general depends on the primal state, at not only time nodes but possibly also on the interval in between – for example, if using a multi-stage method. These in-between evaluations are not directly available for general non-variational time integrators. 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 section, we describe how to reconstruct the solution in time, independent of the time scheme used.

The temporal reconstruction is based on the semi-discrete form of the PDE, (2) [2], which we can re-arrange to solve for the state slope,

$$\frac{d\mathbf{U}}{dt} = -\mathbf{M}^{-1}\mathbf{R}(\mathbf{U}). \tag{10}$$

That is, we can obtain the slope in time through a residual evaluation at a known state. Suppose that we know the states at the endpoints of a time interval, t^n and t^{n+1} – this is what a general time-marching scheme would give us. A very simple reconstruction would be to just connect these states linearly, but that would only give us second-order accuracy. We can do much better if we apply (10) 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.

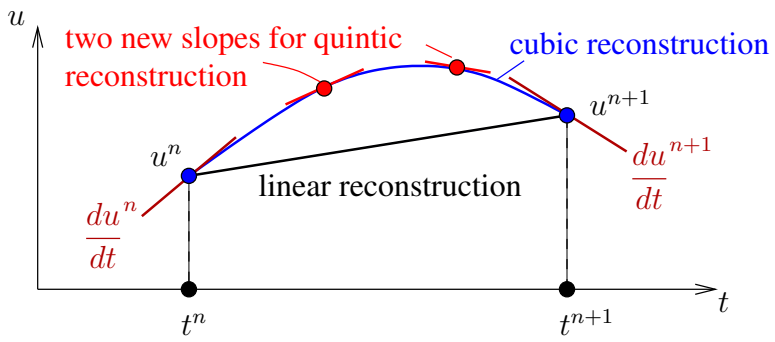


Figure 1: Illustration of solution reconstruction on a time interval. High-order representations can be constructed using slope information readily available from residual evaluations.

Furthermore, we can gain additional accuracy by applying (10) yet again, in between the time nodes, using the reconstructed cubic solution. In particular, for error estimation we will be computing integrals over the time interval via quadrature. For two quadrature points, we evaluate the cubic at those two points and apply (10) to obtain the slopes there. With these additional two pieces of information, we can reconstruct a quintic solution in time (i.e. the unique quintic in time that matches the two endpoint values, two endpoint slopes, and two in-between slopes). 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. To obtain true quintic accuracy, we simply iterate the process, using the current quintic approximation to re-evaluate the slopes at the midpoints, which then yields a new quintic. Just one such iteration will yield the expected convergence rates, but an additional iteration will further lower the errors.

Table 1 shows the results from a numerical test of the reconstruction. The system of interest is a scalar ordinary differential equation, $\frac{du}{dt} = u^2$. Reconstruction is performed over one

[2] For reconstructing the adjoint, we would use (9).

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.

	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).

3.3 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, 6]. Let's first consider only temporal errors. 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 $\bar{J}$,

$$\delta\bar{J} \equiv \bar{J}(\mathbf{U}_H) - \bar{J}(\mathbf{U}) \approx \bar{J}_{\mathbf{U}}(\delta\mathbf{U}) \approx -\int_0^T \mathbf{\Psi}^T\bar{\mathbf{R}}(\mathbf{U}_H)\,dt, \tag{11}$$

where $\delta\mathbf{U} \equiv \mathbf{U}_H - \mathbf{U}$ is the state error, and $\bar{\mathbf{R}}(\mathbf{U}_H) \approx \bar{\mathbf{R}}_{\mathbf{U}}(\delta\mathbf{U})$ 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 will be a higher-order time integration method. Numerical error due to the spatial discretization can be measured by also making the fine space higher order in space. 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

$$\delta\bar{J} \approx -\int_0^T \mathbf{\Psi}_h^T\bar{\mathbf{R}}_h(\mathbf{U}_h^H)\,dt, \tag{12}$$

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. For the time integration schemes considered in this work, we employ a cubic time reconstruction for this injection. The integral in (12) is a summation of integrals over all time intervals, and within each time interval we perform the integral using quadrature – typically between 2 and 4 points, depending on the accuracy of the time integration. For this integral we need to evaluate the adjoint inside the time intervals, and we use a reconstruction appropriate for the fine-space temporal discretization – typically quintic.

Finally, we note that the output error estimate in (12) 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, $\mathbf{\Psi}_h$, spatially

back down to order p and recalculating the error via (12) 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 simply define $\delta\bar{J}^{\text{space}} \equiv \delta\bar{J} - \delta\bar{J}^{\text{time}}$.

3.4 Space-time adaptation

The error estimate in (12) can be localized to contributions of individual time intervals and spatial elements, which could then be used to drive targeted mesh and time-step refinements. However, in the present work we confine adaptation to very simple mechanics: we assume independent but uniform spatial and temporal meshes, and we use the error estimates to guide adaptation either in space or in time (or a combination of both).

The adaptation relies on models of how the spatial and temporal errors behave with refinement. For this we use a standard a priori model,

$$\delta\bar{J}^{\text{space}} \propto (\Delta x)^{p+1}, \quad \delta\bar{J}^{\text{time}} \propto (\Delta t)^{r+1}, \tag{13}$$

where Δx is the spatial mesh size, Δt is the time step size, and r is the formal order of convergence of the time integration. The adaptation also requires a cost model, which we simply take as the total number of space-time degrees of freedom,

$$C \equiv N_{\text{elem}} N_t n_p n_r, \tag{14}$$

where N_t is the number of time intervals, n_p is the number of spatial degrees of freedom per element ($p+1$ in 1D), and n_r is the number of temporal degrees of freedom, i.e. system solves, per time step. For example, BDF2 would nave $n_r = 1$, while the DIRK schemes would have $n_r = n_{\text{stage}}$.

In our simple adaptive mechanics, at each adaptive iteration we multiply the number of spatial elements by f^{space}, and the number of time steps by f^{time}, with a constraint on the total cost rise, $C = C_0 f^{\text{tot}}$. Here, f^{tot} is prescribed by the user, e.g. a factor of 2. Using the cost formula in (14), the constraint becomes $f^{\text{space}} f^{\text{time}} = f^{\text{tot}}$. To determine f^{space} and f^{time}, we require that the marginal error to cost ratios of each refinement option are equal: $\lambda^{\text{space}} = \lambda^{\text{time}}$, where

$$\lambda^{\text{space}} = \frac{\partial(\delta\bar{J}^{\text{space}})}{\partial f^{\text{space}}} \left[\frac{\partial C}{\partial f^{\text{space}}}\right]^{-1}, \quad \lambda^{\text{time}} = \frac{\partial(\delta\bar{J}^{\text{time}})}{\partial f^{\text{time}}} \left[\frac{\partial C}{\partial f^{\text{time}}}\right]^{-1}. \tag{15}$$

Substituting into the above equations the error and cost models from (13) and (14), and using the constraint $f^{\text{space}} f^{\text{time}} = f^{\text{tot}}$, we obtain the following solution:

$$f^{\text{time}} = \left[\frac{r+1}{p+1} \frac{\delta\bar{J}^{\text{time}}}{\delta\bar{J}^{\text{space}}} (f^{\text{tot}})^{p+2}\right]^{-(r+p+4)}, \quad f^{\text{space}} = \frac{f^{\text{tot}}}{f^{\text{time}}}.$$

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 RESULTS

In this section we present results demonstrating the accuracy of the output error estimate, the accuracy of the spatial/temporal error breakdown, and the efficacy of output-based adaptation

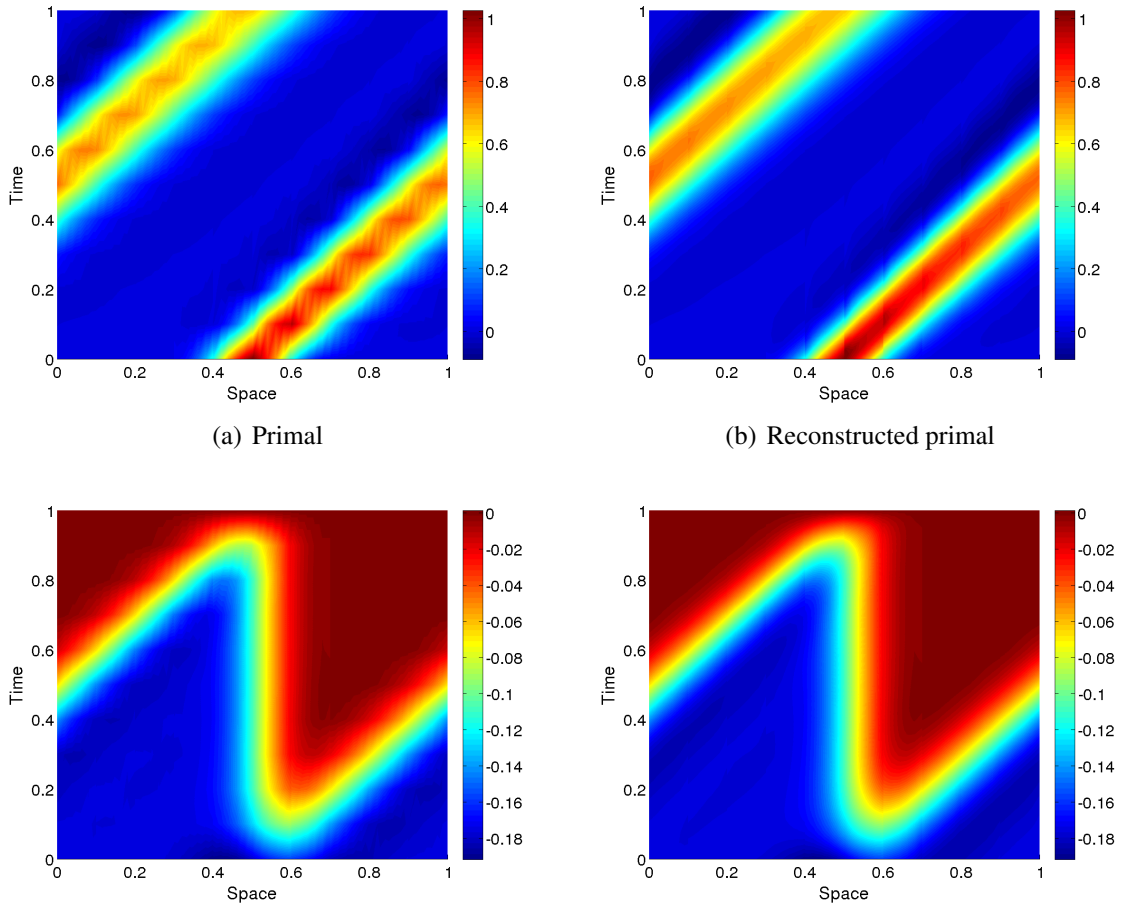
for an unsteady advection simulation. The governing equation is

$$\frac{\partial u}{\partial t} + a\frac{\partial u}{\partial x} = 0, \quad u(x, t = 0) = u_0(x), \tag{16}$$

where $a = 1$ is the advection speed. The boundary conditions are periodic, and the initial condition is $u_0(x) = \exp\left[100(x/L - .5)^2\right]$, where $L = 1$ is the domain length. The output of interest is a weighted function of the state,

$$\bar{J} = \int_0^T \int_0^L w(x)u(x, t)\, dx\, dt, \tag{17}$$

where $w(x) = \exp\left[100(x/L - .52)^2\right]$, and the final simulation time is $T = 1$. Figure 2 shows the primal and adjoint solutions computed using 10 elements in space and 10 intervals in time, with $p = 2$ and DIRK3. The space-time fields without reconstruction are visualized by linearly interpolating the time node states, and these are visibly "choppy", masking the true high-order temporal accuracy. Reconstructing the primal and adjoint solutions using cubic functions over the time steps produces visible improvements in the space-time solutions in between the time nodes.



(a) Primal

(b) Reconstructed primal

(c) Adjoint

(d) Reconstructed adjoint

Figure 2: Primal and adjoint space-time solutions for the advection problem, with and without temporal reconstruction (cubic), shown for the case of $N_{\text{elem}} = 10$, $N_t = 10$, $p = 2$, DIRK3.

We first examine the effectivity of the total error estimate and the breakdown into spatial and temporal contributions. Table 2 lists these errors for various combinations of element and time step numbers, all for $p = 2$ and DIRK3 as the coarse space, and $p = 3$ and DG2 as the fine space. We see that the total error estimates are very accurate, with a maximum error of less than 7%. In addition, the spatial and temporal error estimates behave as expected: the spatial error diminishes when the number of elements increases, and similarly for the temporal error. The "actual" values of these errors were computed by refining the spatial and temporal discretizations independently, using concurrent uniform refinement and order increase, and the estimates agree very well with the actual values.

		δJ		δJ^{time}		δJ^{space}	
N_{elem}	N_t	Predicted	Actual	Predicted	Actual	Predicted	Actual
10	10	-5.66e-05	-6.03e-05	-8.89e-05	-1.11e-04	3.23e-05	1.56e-05
10	100	4.96e-05	5.14e-05	9.98e-07	8.70e-07	4.86e-05	5.01e-05
100	10	-7.00e-05	-7.59e-05	-7.00e-05	-7.65e-05	6.62e-09	1.80e-10
100	100	1.30e-06	1.30e-06	1.30e-06	1.30e-06	5.32e-10	5.28e-10

Table 2: Comparison of estimated and predicted output errors, and their breakdown into spatial and temporal components, for the advection simulation on several space-time meshes.

Next, several different orders and time integration schemes were tested in the context of space-time output-based refinement. In all cases, the total space-time degrees of freedom (i.e. the cost) were allowed to grow by a factor of $f^{\text{tot}} = 2$ per adaptive iteration, and 10 adaptive iterations were taken. The fine space consisted of $p + 1$ spatial approximation and time integration of one order higher (e.g. DIRK3 for BDF2). Figure 3 compares the convergence of the output error for different combinations of orders and time marching schemes. We see that the most efficient combination for this smooth problem is $p = 3$ and DIRK4 – as expected, high order in space and time.

To further examine the performance of the output-based adaptation, we compare the results for each run to uniform refinement of the starting space-time mesh. Figure 4 shows this comparison and includes an additional set of curves: the output error resulting after correction of the output with the error estimate. These curves converge the fastest, as expected given the previously observed effectivity of the error estimate. The rate of convergence is generally at least one higher than the rate of convergence of the output error without correction. Performing the worst is uniform refinement, where the convergence rate is limited by the dominant errors in space or in time.

Poor performance of uniform refinement is tied to the starting space-time mesh. For example, a suboptimal ratio of elements to time steps will not be addressed by uniform refinement. Figure 5 illustrates this point by comparing the separate spatial and temporal errors for adaptive runs and uniform refinement. In the case of $p = 2$ and BDF2, the temporal errors will dominate and must be addressed by increased allocation of resources to the temporal discretization (i.e. the number of time steps). This is what the adaptive refinement accomplishes in just a couple iterations, and for the remainder of the simulation the spatial and temporal errors roughly balance (since their marginal cost is identical). In the case of $p = 1$ and DIRK4, the spatial errors dominate and should be addressed by relatively more spatial refinement. Again, the adaptive run accomplishes this, whereas the uniform refinement run is consistently dominated by spatial errors.

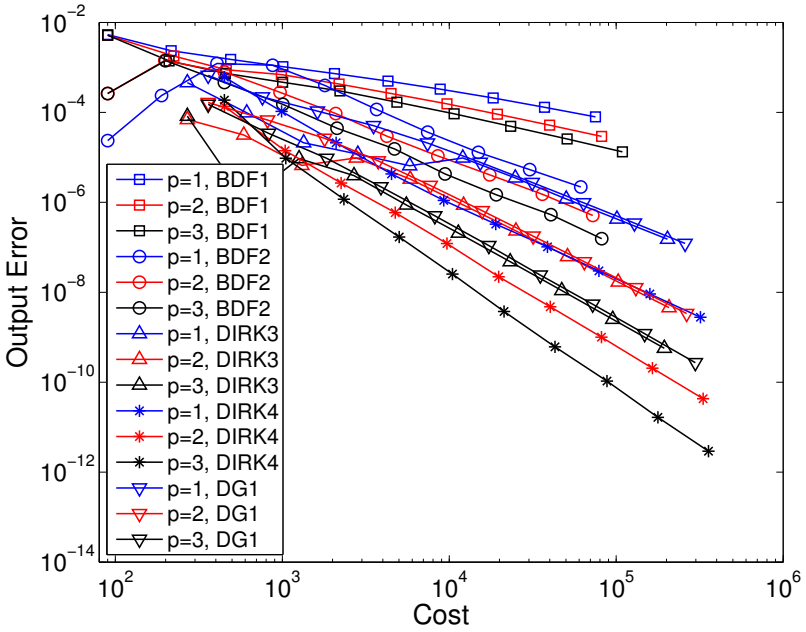


Figure 3: Convergence of the output error with cost, measured as the total space-time degrees of freedom, using output-based space-time adaptation.

5 CONCLUSIONS

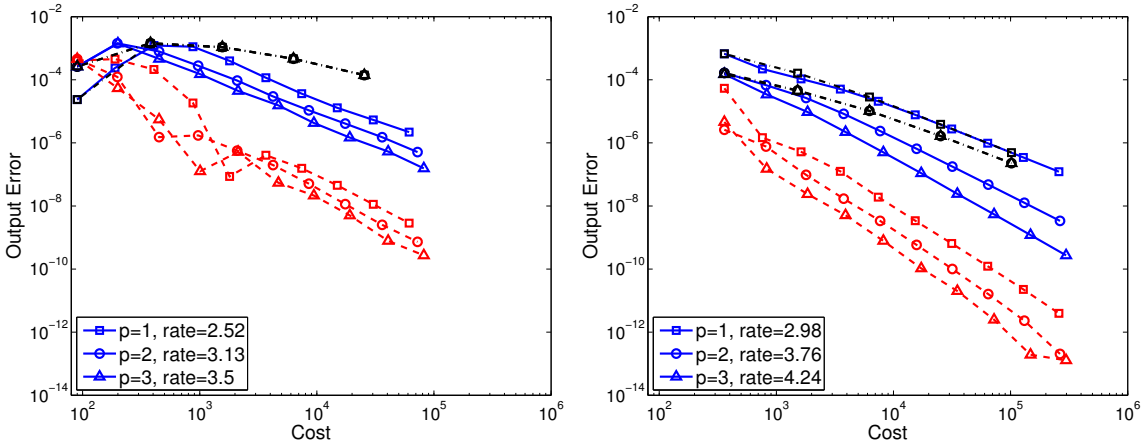
We have presented a general approach for estimating numerical errors in outputs of unsteady simulations when using non-variation time integration methods. The two key ingredients are: (1) a continuous-in-time adjoint solution that simplifies adjoint-consistency considerations and removes constraints on the choice of the adjoint time integration scheme; and (2) a scheme-agnostic temporal reconstruction procedure based on primal or adjoint residual evaluations, which provides a consistent, high-order functional representation of the solution within each time step. When used in conjunction with a spatial discretization, the method allows for separate space and time error estimates via adjoint down-projection. The resulting error estimates can then drive an space-time adaptive procedure, as demonstrated for a prototypical advection problem. Remaining work in this area includes investigation of fine-space adjoint approximations and application to more complex nonlinear simulations, such as those governed by the compressible Navier-Stokes equations.

REFERENCES

[1] Timothy J. Barth. Space-time error representation and estimation in Navier-Stokes calculations. In Stavros C. Kassinos, Carlos A. Langer, Gianluca Iaccarino, and Parviz Moin, editors, *Complex Effects in Large Eddy Simulations*, pages 29–48. Springer Berlin Heidelberg, Lecture Notes in Computational Science and Engineering Vol 26, 2007.

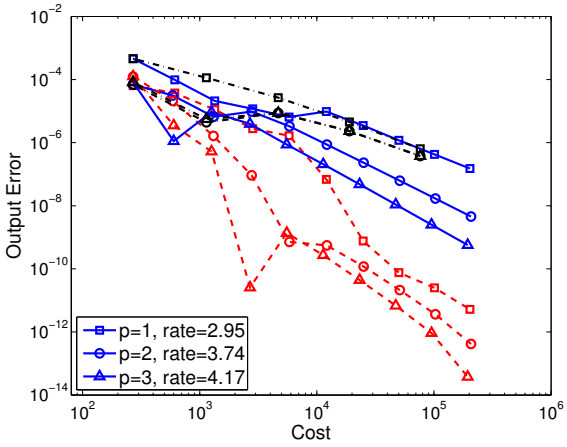
[2] R. Becker and R. Rannacher. An optimal control approach to a posteriori error estimation in finite element methods. In A. Iserles, editor, *Acta Numerica*, pages 1–102. Cambridge University Press, 2001.

[3] A. Belme, A. Dervieux, and F. Alauzet. Error estimation and adaptation for functional outputs in time-dependent flow problems. *Journal of Computational Physics*, 231:6323–

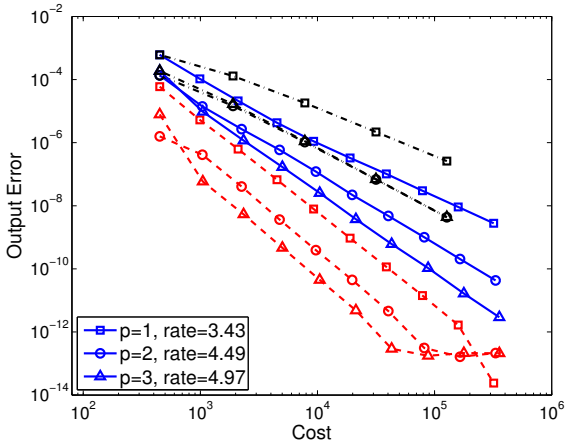


(a) BDF2

(b) DG1

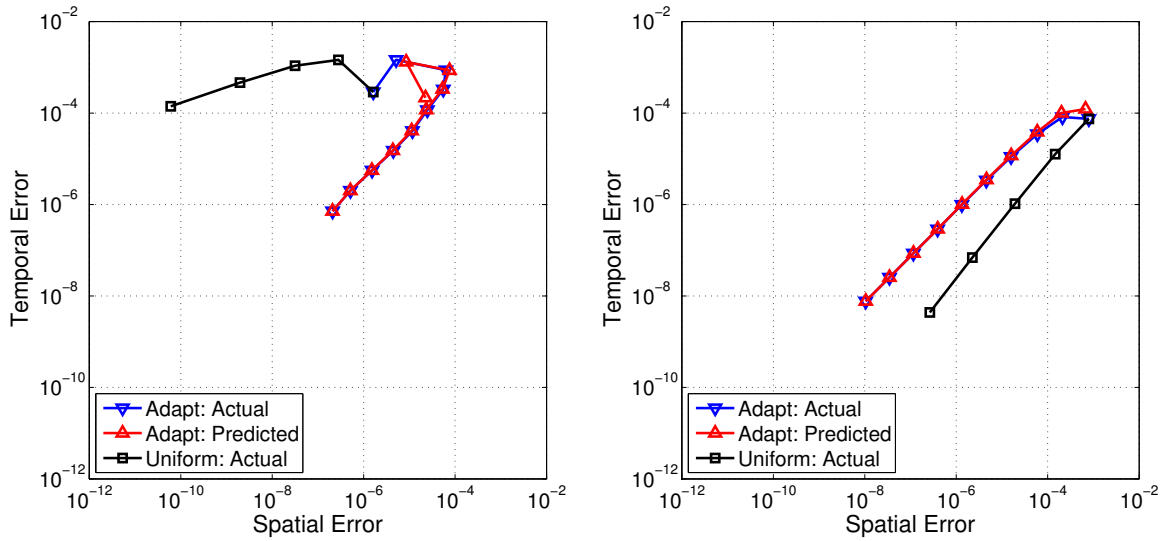


(c) DIRK3



(d) DIRK4

Figure 4: Convergence of the output error with cost, measured as the total space-time degrees of freedom, for various spatial orders and temporal discretizations. Adaptive refinement (solid blue) is compared to uniform refinement (dash-dot black), and to adaptive refinement corrected by the error estimate (dashed red).



(a) $p = 2$, BDF2 (b) $p = 1$, DIRK4

Figure 5: Spatial and temporal error distribution histories for adaptive and uniform refinement of the advection simulation, using two combinations of spatial and temporal discretizations.

6348, 2012.

[4] Michael Besier and Rolf 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*, 70:1139–1166, 2012.

[5] Krzysztof J. Fidkowski. An output-based dynamic order refinement strategy for unsteady aerodynamics. AIAA Paper 2012-77, 2012.

[6] Krzysztof J. Fidkowski and David L. Darmofal. Review of output-based error estimation and mesh adaptation in computational fluid dynamics. *American Institute of Aeronautics and Astronautics Journal*, 49(4):673–694, 2011.

[7] Krzysztof J. Fidkowski and Yuxing Luo. Output-based space-time mesh adaptation for the compressible Navier-Stokes equations. *Journal of Computational Physics*, 230:5753–5773, 2011.

[8] Bryan T. Flynt and Dimitri J. Mavriplis. Discrete adjoint based adaptive error control in unsteady flow problems. AIAA Paper 2012-0078, 2012.

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

[10] Steven M. Kast and Krzysztof J. Fidkowski. Output-based mesh adaptation for high order Navier-Stokes simulations on deformable domains. *Journal of Computational Physics*, 252(1):468–494, 2013.

[11] Karthik Mani and Dimitri J. Mavriplis. Error estimation and adaptation for functional outputs in time-dependent flow problems. *Journal of Computational Physics*, 229:415–440, 2010.

[12] Marian Nemec and Michael J. Aftosmis. Error estimation and adpative refinement for embedded-boundary Cartesian meshes. AIAA Paper 2007-4187, 2007.

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

[14] Michael Schmich and Boris Vexler. Adaptivity with dynamic meshes for space-time finite element discretizations of parabolic equations. *SIAM Journal on Scientific Computing*, 30(1):369–393, 2008.

[15] S. Sen, K. Veroy, D.B.P. Huynh, S. Deparis, N.C. Nguyen, and A.T. Patera. "Natural norm" a posteriori error estimators for reduced basis approximations. *Journal of Computational Physics*, 217:37–62, 2006.

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

[17] Michael Woopen, Aravind Balan, Georg May, and Jochen Schütz. A comparison of hybridized and standard DG methods for target-based hp-adaptive simulation of compressible flow. *Computers & Fluids*, 98:3–16, 2014.