50th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition
09 - 12 January 2012, Nashville, Tennessee

AIAA 2012-0077

# An Output-Based Dynamic Order Refinement Strategy for Unsteady Aerodynamics

Krzysztof J. Fidkowski [*]
*Department of Aerospace Engineering, University of Michigan, Ann Arbor, MI 48109*

**An output-based dynamic order refinement strategy is presented for unsteady simulations using the discontinuous Galerkin finite element method in space and time. A discrete unsteady adjoint solution provides scalar output error estimates and drives adaptive refinement of the space-time mesh. Space-time anisotropy is measured using projection of the adjoint onto semi-refined spaces and is used to allocate degrees of freedom to additional time slabs or increased spatial order of individual space-time elements. The spatial refinement is dynamic in that the solution approximation order can change between time slabs. Results for the compressible Euler equations demonstrate benefits of the dynamic order refinement strategy in terms of total degrees of freedom: at strict error tolerances, the savings approaches two orders of magnitude compared to uniform refinement, and a factor of two to three compared to output-based refinement with a static spatial mesh.**

## I. Introduction

Output-based error estimation in computational fluid dynamics improves robustness by providing numerical error bars on quantities of interest and adaptive indicators for driving mesh adaptation. The resulting computational meshes are tailored for the prediction of a desired scalar quantity and often contain orders of magnitude fewer degrees of freedom than meshes constructed a priori. The cost of this tailoring lies in the solution of an adjoint problem for the scalar output, which is non-trivial, especially for unsteady problems. Therefore, to produce a competitive output-based method, we seek to reap the maximum benefit from unsteady adjoint solutions by designing an efficient space-time adaptive strategy.

At present, the development of output-based methods has focused primarily on steady-state problems.[1–7] However, unsteady problems have recently begun to receive increased attention. Topics addressed thus far include temporal error estimation and adaptation,[8,9] static spatial mesh adaptation,[10] combined temporal and static spatial adaptation,[11] and combined temporal and dynamic spatial $h$ refinement.[12,13]

This work considers combined temporal and dynamic order-, "$p$-", refinement for applications that include unsteady aerodynamics. The discontinuous Galerkin finite element method is used for both the spatial and temporal discretizations. The adjoint problem is solved in a discrete fashion, and the error is estimated using an adjoint-weighted residual.

[*]Assistant Professor, AIAA Member

The present research is a continuation of previous work in unsteady output-based adaptation on static meshes.[11,14] The discretization and error estimation extend naturally to the case of dynamic spatial order refinement, and the required modifications are discussed in Sections II and III. Error localization is also modified to increase the granularity of the adaptive indicator for driving dynamic mesh refinement, as described in Section IV. Two sets of adaptive results in Section V demonstrate the benefits of dynamic order refinement over uniform refinement and output-based static mesh/order refinement.

## II. Discretization

### A. Forward Discretization

We consider unsteady flows governed by conservation laws of the form

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

where $\mathbf{u}(\mathbf{x}, t) \in \mathbb{R}^s$ is the state vector, $\mathbf{x} \in \mathbb{R}^d$ is the spatial coordinate, and $t \in \mathbb{R}$ is time. $\mathbf{r} : \mathbb{R}^s \to \mathbb{R}^s$ is a spatial operator, $s$ is the number of governing equations, and $d$ is the spatial dimension.

We use a discontinuous finite element method in both space and time to discretize Eqn. 1. A sample space-time mesh allowable in the present work is illustrated in Figure 1. The temporal dis-

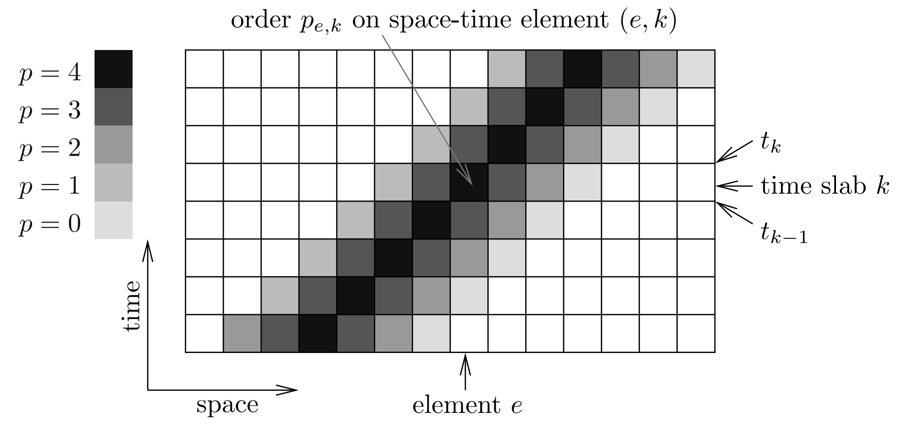


**Figure 1. Sample space-time mesh featuring dynamic order refinement.**

cretization is restricted to slabs on which the temporal variation is approximated with polynomials of order $r$. This means that all elements advance at the same time step, $\Delta t$, which can vary in time. On the other hand, the spatial discretization allows for variable spatial order on individual space-time elements. That is, the spatial approximation order need not be the same for all spatial elements in a given time slab, nor for all time slabs corresponding to a given spatial element.

Aside from the order of approximation, the spatial mesh remains fixed in time. Each space-time element is then identified by two indices, $(e, k)$, where $e$ is the spatial element index and $k$ is the time slab index. $p_{e,k}$ denotes the order of approximation on space-time element $(e, k)$.

On each space-time element, the forward solution is approximated as,

$$\mathbf{u}_H(\mathbf{x},t)\Big|_{e,k} = \underbrace{\mathbf{u}^{k,n}_{H,e,j}}_{\in\mathbb{R}^s}\underbrace{\phi^k_{H,e,j}(\mathbf{x})}_{\text{order } p_{e,k}}\underbrace{\varphi^n_H(t)}_{\text{order r}}, \tag{2}$$

where $j \in [1\ldots \mathrm{dof}(p_{e,k})]$ is the spatial degree-of-freedom index on space-time element $(e,k)$ and $n \in [1\ldots r+1]$ is the temporal degree of freedom index on time slab k. The number of spatial degrees of freedom, $\mathrm{dof}(p_{e,k})$, depends on the approximation basis and the dimension; e.g. for tensor-product approximation in two dimensions, $\mathrm{dof}(p_{e,k}) = (p_{e,k}+1)^2$. Note that the spatial basis functions, $\phi^k_{H,e,j}(\mathbf{x})$, are specific to an element and time slab, while the temporal basis functions, $\varphi^n_H(t)$, are the same for each time slab. We employ a Lagrange basis on equally-spaced nodes for $\varphi^n_H(t)$, and a Lagrange basis on tensor-product Gauss quadrature points for $\phi^k_{H,e,j}(\mathbf{x})$.

For compactness of notation, we lump all of the spatial degrees of freedom associated with time node $n$ on slab $k$ into one vector,

$$\mathbf{U}^{k,n}_H = \left\{\mathbf{u}^{k,n}_{H,e,j}\right\}_{\forall e,j} \in \mathbb{R}^{N^k_H}, \tag{3}$$

where $N^k_H = s\sum_e \mathrm{dof}(p_{e,k})$ is the total number of spatial degrees of freedom, including the state rank, on time slab $k$.

A nonlinear system of equations on each time slab is obtained by substituting the approximation from Eqn. 2 into Eqn. 1, multiplying by test functions in the same space as the approximation functions, and integrating by parts to incorporate discontinuities at time slab and spatial element interfaces. These equations are expressed in terms of $r+1$ residual vectors on time slab $k$,

$$\overline{\mathbf{R}}^{k,m}_H \;\equiv\; a^{m,n}\mathbf{M}^{k,k}_H\mathbf{U}^{k,n}_H - \varphi^m_H(t_{k-1})\mathbf{M}^{k,k-1}_H\mathbf{U}^{k-1,r+1}_H + \int_{t_{k-1}}^{t_k}\varphi^m_H(t)\mathbf{R}_H(\mathbf{U}^k_H(t))\,dt \;=\; \mathbf{0}, \tag{4}$$

where $\overline{\mathbf{R}}^{k,m}_H \in \mathbb{R}^{N_H}$ and $m \in [1\ldots r+1]$. The $a^{m,n}$ are time-slab-independent coefficients that are defined by,

$$a^{m,n} \;=\; -\int_{t_{k-1}}^{t_k}\varphi^n_H\frac{d\varphi^m_H}{dt}\,dt + \varphi^n_H(t_k)\varphi^m_H(t_k). \tag{5}$$

Since the temporal basis is Lagrange on equally-spaced time nodes, $\mathbf{U}^{k-1,r+1}_H$ is the state at the end of the previous time slab; on the first time slab this is just the initial condition. $\mathbf{U}^k_H(t) = \sum_n \mathbf{U}^{k,n}_H\varphi^n_H(t)$ is the temporal approximation of the state on time slab $k$, and $\mathbf{R}_H \in \mathbb{R}^{N_H}$ is the spatial residual. We do not focus on the details of the spatial discretization and only mention that it is a discontinuous Galerkin method employing the Roe inviscid flux[15] and the second form of Bassi and Rebay for the viscous flux.[16]

Given two time slabs $k$ and $l$, $\mathbf{M}^{k,l}_H$ is the mass matrix formed from the spatial basis functions on each slab, which need not be the same due to the possibility of dynamic order refinement. This matrix is element-wise block diagonal, and on each element $e$ it is given by

$$\mathbf{M}^{k,l}_H\Big|_e = \underbrace{\left[\int_{\Omega_e}\phi^k_{H,e,i}(\mathbf{x})\phi^l_{H,e,j}(\mathbf{x})\right]}_{\mathrm{dof}(p_{e,k})\times\mathrm{dof}(p_{e,l})\text{ matrix}}\otimes\mathbf{I}_s, \tag{6}$$

where $\Omega_e$ is the portion of the spatial domain enclosed by spatial element $e$, and $\mathbf{I}_s$ is the $s\times s$ identity matrix. Note, $(\mathbf{M}^{k,l}_H)^T = \mathbf{M}^{l,k}_H$.

### B. Adjoint Discretization

The discrete adjoint associated with a scalar output calculated from the unsteady solution, $J_H(\mathbf{U}_H^{k,n})$, is the sensitivity of $J_H$ to residual source perturbations added to Eqn. 4. Denoting the adjoint at time slab $l$, time node $m$, by $\mathbf{\Psi}_H^{l,m}$, the discrete adjoint equation is[7]

$$\underbrace{\left(\frac{\partial \overline{\mathbf{R}}_H^{l,m}}{\partial \mathbf{U}_H^{k,n}}\right)^T \mathbf{\Psi}_H^{l,m} + \left(\frac{\partial J_H}{\partial \mathbf{U}_H^{k,n}}\right)^T}_{\overline{\mathbf{R}}_H^{\psi,k,n}(\mathbf{\Psi}_H^{l,m})} = 0, \tag{7}$$

where $k, l$ index time slabs and $n, m$ index time nodes. Linearizing the residual expressions in Eqn. 4, the $r+1$ adjoint residual vectors on time slab $k$ are

$$\begin{aligned}\overline{\mathbf{R}}_H^{\psi,k,n} &= a^{m,n}\mathbf{M}_H^{k,k}\mathbf{\Psi}_H^{k,m} - \varphi_H^n(t_k)\mathbf{M}_H^{k,k+1}\mathbf{\Psi}_H^{k+1,1} \\ &\quad + \int_{t_{k-1}}^{t_k} \varphi_H^n \left(\frac{\partial \mathbf{R}}{\partial \mathbf{U}}(\mathbf{U}_H(t))\right)^T \mathbf{\Psi}_H^k(t)\, dt + \left(\frac{\partial J_H}{\partial \mathbf{U}_H^{k,n}}\right)^T,\end{aligned} \tag{8}$$

where $\mathbf{\Psi}_H^k(t) = \sum_m \mathbf{\Psi}_H^{k,m}\varphi_H^m(t)$, and $\mathbf{\Psi}_H^{k+1,1}$ is the adjoint vector associated with the start of the next time slab. When calculating on the last time slab, $\mathbf{\Psi}_H^{k+1,1} = 0$.

Both the forward and adjoint equations are solved using a Newton iteration based on an approximate factorization. This solver is based on solutions of systems that are the same size as a steady-state solution. Details are given in.[11]

## III. Output Error Estimation

### A. The Adjoint-Weighted Residual

Since the approximation space for the forward solution is finite-dimensional, the output $J_H(\mathbf{U}_H)$ will in general be polluted by numerical error. We estimate this error by comparing the output to one computed on a finer space, denoted by the subscript $h$. In this work, the finer space consists of order enrichment in both space and time: i.e. using a spatial order of $p_{e,k}+1$ and a temporal order of $r+1$ for each space-time element $e, k$.

An estimate of the output error is obtained by taking the difference between the output computed with the coarse solution and that computed with the fine solution. This difference is approximated using an adjoint-weighted residual,[7]

$$\begin{aligned}\delta J = \text{output error} &\approx J_H(\mathbf{U}_H) - J_h(\mathbf{U}_h) \\ &= \underbrace{\left(\delta\mathbf{\Psi}_h^{k,m}\right)^T \overline{\mathbf{R}}_h^{k,m}(\mathbf{U}_h^{H,l,n})}_{\text{estimate}} + \underbrace{R^{(2)}\left(||\delta\mathbf{U}_h^{H,l,n}||, ||\delta\mathbf{\Psi}_h^{k,m}||\right)}_{\text{remainder}},\end{aligned} \tag{9}$$

where $\mathbf{U}_h^{H,l,n}$ is an injection of the coarse forward solution into the fine space, $\delta\mathbf{\Psi}_h^{k,m} = \mathbf{\Psi}_h^{H,k,m} - \mathbf{\Psi}_h^{k,m}$ is the difference between the injection of the coarse adjoint into the fine space and the fine adjoint. $\delta\mathbf{U}_h^{H,l,n}$ is defined similarly to $\delta\mathbf{\Psi}_h^{k,m}$, and $R^{(2)}(\cdot,\cdot)$ is a second-order remainder term that we will neglect in our output error estimate calculation.

The error estimate in Eqn. 9 requires an evaluation of the fine-space unsteady residual associated with the coarse solution, and the fine-space adjoint solution $\mathbf{\Psi}_h^{k,m}$. In this work we solve the fine-space adjoint equation exactly (to machine precision) to minimize additional sources of error in our estimates. However, in practice the fine-space adjoint can be computed by smoothing or reconstructing the coarse-space adjoint, in order to minimize computational cost.[17]

## B. Error Localization

The output error estimate in Eqn. 9 can be written as a sum over space-time elements,

$$\delta J = \sum_k \sum_e \varepsilon_{e,k}, \tag{10}$$

where the error contribution of a space-time element $(e, k)$ is

$$\varepsilon_{e,k} = \sum_m \left(\mathbf{Z}_e \delta \mathbf{\Psi}_h^{k,m}\right)^T \mathbf{Z}_e \overline{\mathbf{R}}_h^{k,m} \left(\mathbf{U}_h^{H,l,n}\right). \tag{11}$$

In the above expression, $m \in [1 \ldots r+2]$ is an index over the fine-space temporal degrees of freedom, and $\mathbf{Z}_e$ is a mask matrix that returns the spatial degrees of freedom associated with element $e$. The error indicator for a space-time element is taken as the absolute value of the elemental contribution to the output error,

$$\text{error indicator} = \epsilon_{e,k} = \left|\varepsilon_{e,k}\right|.$$

This indicator identifies space-time elements that contribute most to the output error. However, it does not contain information about the source of the error, i.e. spatial or temporal. This information is obtained from the space-time anisotropy measure described next.

## C. Space-Time Anisotropy

Key to a successful adaptive strategy is determining the space-time anisotropy of the output error. This is a separation of the error into a contribution due to spatial resolution and a contribution due to temporal resolution. In this work we calculate the anisotropy using separate projections of the fine-space adjoint onto semi-coarsened spatial and temporal spaces.[17] The resulting spatial and temporal error estimates for space-time element $(e, k)$ are respectively

$$\varepsilon_{e,k}^{\text{space}} = \sum_m \left(\mathbf{Z}_e \delta \mathbf{\Psi}_{hH}^{k,m}\right)^T \mathbf{Z}_e \overline{\mathbf{R}}_h^{k,m} \left(\mathbf{U}_h^{H,l,n}\right), \tag{12}$$

$$\varepsilon_{e,k}^{\text{time}} = \sum_m \left(\mathbf{Z}_e \delta \mathbf{\Psi}_{Hh}^{k,m}\right)^T \mathbf{Z}_e \overline{\mathbf{R}}_h^{k,m} \left(\mathbf{U}_h^{H,l,n}\right), \tag{13}$$

where $\mathbf{\Psi}_{hH}^{k,m} \in \mathbb{R}^{N_h}$ and $\mathbf{\Psi}_{Hh}^{k,m} \in \mathbb{R}^{N_h}$ are discrete fine-space representations of the adjoints projected into coarse temporal and spatial spaces, respectively. Here $m$ is again an index over the fine temporal degrees of freedom. We do not use these values directly; rather, we only use their ratio to calculate the spatial/temporal error fractions on element $e, k$,

$$\beta_{e,k}^{\text{space}} = \frac{|\varepsilon_{e,k}^{\text{space}}|}{|\varepsilon_{e,k}^{\text{space}}| + |\varepsilon_{e,k}^{\text{time}}|}, \qquad \beta_{e,k}^{\text{time}} = 1 - \beta_{e,k}^{\text{space}}. \tag{14}$$

## IV. Dynamic Mesh Adaptation

The output error estimate drives an adaptive process in which the unsteady problem is solved on successively refined space-time meshes. The process requires, at each adaptive iteration, forward and adjoint solutions, which become more expensive on the finer meshes. This section describes details of the adaptive process, including the calculation of the adaptive indicators, the refinement strategy, and the implementation.

### A. Adaptive Indicators and Refinement Strategy

The adaptive strategy in this work consists of:

- Increasing temporal resolution through time slab bisection.
- Increasing spatial resolution through dynamic order increment: $p_{e,k} \rightarrow p_{e,k} + 1$.

In this strategy, the slab-based temporal discretization is preserved, but the spatial approximation order on each element can vary in time, as illustrated in Figure 1. To drive this refinement, adaptive indicators are necessary for identifying the amount of temporal error associated with each time slab and the amount of spatial error associated with each space-time element. These indicators are obtained from the error estimate, Eqn. 11, and the space-time anisotropy measure, Eqn. 14. Specifically, we have,

$$\text{spatial indicator on space-time element } e,k = \quad \epsilon_{e,k}^{\text{space}} = \epsilon_{e,k}\beta_{e,k}^{\text{space}}, \tag{15}$$

$$\text{temporal indicator on time slab } k = \quad \epsilon_{k}^{\text{time}} = \sum_{e} \epsilon_{e,k}\beta_{e,k}^{\text{time}}, \tag{16}$$

where the sum indexed by $e$ is taken over all spatial elements.

The above indicators are used in a fixed-growth adaptive strategy in which some combination of time slabs and space-time elements are marked for refinement. The increase in the total degrees of freedom, spatial and temporal, at every adaptive iteration is governed by a growth factor, $f^{\text{growth}}$. The budget of new space-time degrees of freedom is $(1-f^{\text{growth}})$ times the current degree of freedom count. Elements/time-slabs are marked for refinement until the degree-of-freedom budget is met or exceeded.

A greedy algorithm is used to decide which space-time elements or time slabs to refine. The figure of merit of each refinement option is the amount of output error addressed divided by the degrees of freedom added. The errors addressed are taken to be the indicators in Eqns. 15 and 16. The number of additional degrees of freedom associated with a slab bisection is approximated as the degrees of freedom in the targeted slab, $\sum_e \text{dof}(p_{e,k})$ for time slab $k$. The number of additional degrees of freedom associated with an order increase of element $e, k$ is $\text{dof}(p_{e,k} + 1) - \text{dof}(p_{e,k})$. The figure of merit is calculated for each space-time element and time slab, after which the set of elements/slabs is sorted according to the figure of merit. The space-time element or time slab with the highest figure of merit is chosen for refinement first, and the process continues until the growth budget is met or exceeded.

### B. Implementation

The implementation of the adaptive solution process is described in the following outline:

1. Start with a coarse spatial and temporal discretization.

2. Calculate the unsteady forward solution $\mathbf{U}_H^n$, $n \in [1 \ldots N_H^{\text{time}}]$. Save the state on each time slab to disk.
3. Begin a loop backwards in time over the time slabs, indexed by $k$.
   (a) Load the coarse solution on time slab $k$ from disk.
   (b) Inject the coarse solution into the fine space to obtain $\mathbf{U}_h^{H,k,m}$.
   (c) Solve the fine adjoint problem using approximation orders of $p_e^k + 1$ in space and $r + 1$ in time.
   (d) Calculate the fine-space residual and form the inner product in Eqn. 9 for the current time slab. Add to $\delta J$.
   (e) Localize the error indicators to space-time elements and time slabs using Eqns. 15 and 16. Store these values to disk.
4. If $\delta J$ is below the user tolerance, stop.
5. Calculate the adaptive figure of merit, error addressed per degree of freedom added, for each slab and space-time element. Flag the space-time elements and slabs with the highest figure of merit for refinement, taking into account the degree-of-freedom growth factor.
6. Bisect time slabs and increment the spatial approximation orders $p_{e,k}$ in the flagged space-time elements. Store the new order time history in files to be read in at the subsequent solve.
7. Return to step 2.

Note that the solution, order information, and error indicators are stored to disk to minimize the memory usage of the code. Disk storage has not been problematic for the cases run, but solution checkpointing can be used to trade-off storage costs against computational time.[18]

Dynamic refinement of the spatial mesh is allowed at every time step, as the refinement itself is not time consuming relative to the implicit solver. However, in a parallel setting load balancing should be addressed. Currently the code is implemented in parallel, but mesh partitioning is oblivious to the element order, which means that the partitioning remains fixed throughout the simulation. Mesh re-partitioning based on order is being addressed in ongoing work, and this will likely involve limiting dynamic refinement to every few time steps.

## V. Results

We present results of the output-based dynamic order refinement strategy applied to two problems: scalar advection in one dimension and the compressible Euler equations in two dimensions. In both examples, $r = 1$ is used for the temporal approximation order. The growth factor used in the adaptation, $f^{\text{growth}}$, ranges from 1.2 to 1.5, and a fixed number of adaptive iterations is run in each case.

The dynamic order refinement strategy is compared to two types of uniform refinement and two static adaptive strategies, described below:

- "uniform h": the temporal mesh is uniformly bisected and the elements of the spatial mesh are uniformly refined into equal (in reference space) sub-elements; i.e. $h$ refinement.
- "uniform p": the temporal mesh is uniformly bisected and the spatial approximation order of each element is incremented by 1; i.e. $p$ refinement.
- "static h": output-driven strategy in which the space-time mesh remains tensor-product in structure, as described in,[11] and in which hanging-node $h$ refinement is used for the spatial mesh.

- "static p": output-driven strategy similar to "static h" except that order refinement is used instead of hanging-node $h$ refinement of the spatial mesh.

## A. Advection in One Spatial Dimension

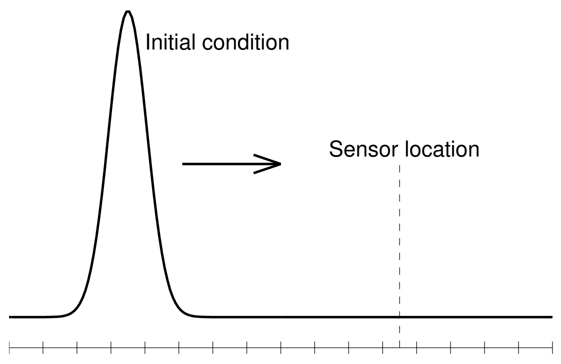
Of interest in this example is one-dimensional advection of a scalar quantity with a nonlinear source term. The governing equation is

$$\frac{\partial u}{\partial t} + V\frac{\partial u}{\partial x} + cu^2 = 0,$$

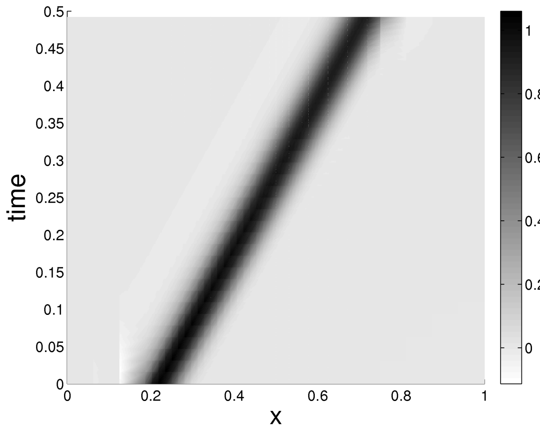
where $V = 1$ is the advection velocity and $c = 0.1$. This equation models a situation in which a scalar concentration is advected in the positive $x$ direction while decaying due to a quadratic source. The spatial domain extends from $x = 0$ to $x = 1$, and the temporal domain from $t = 0$ to $t = 0.5$. The initial condition, illustrated in Figure 2a, is given by

$$u(x, t = 0) = e^{-400(x-7/32)^2}.$$

A sample forward solution is shown in Figure 2b. The initial mesh used for adaptation consists of 4 time slabs and 16 spatial elements of approximation order $p = 1$, as shown in Figure 4a.



(a) problem setup



(b) forward solution, $u(x, t)$

**Figure 2. 1D scalar advection: problem setup and solution. The output is the scalar concentration at the sensor measured at the end of the simulation.**

The output of interest for adaptation is the scalar value measured at $x = 7/32 + 0.5 = 23/32$ at $t = 0.5$, the end of the simulation. This location coincides with the advected peak of the initial condition, and the exact value of the output can be calculated analytically as

$$J = u(x = 23/32, t = 0.5) = \frac{1}{1 + (0.1)(0.5)} = \frac{1}{1.05}.$$

The three output-based adaptive schemes and the two uniform refinement strategies were run on this problem. The growth factor for the output-based strategies was $f^{\text{growth}} = 1.5$. Figure 3 shows the convergence of the output and the output error, relative to the actual output value, versus degrees of freedom. Error bars at $\pm\delta J$, calculated from Eqn. 9, are included on the output-based results in Figure 3a; these are also shown in Figure 3b as dashed lines. We see that the slowest-converging strategy is uniform $h$ refinement, while the fastest strategy is the dynamic order refinement proposed in this work. The difference between these methods, in degrees of freedom for a given accuracy, depends on the desired accuracy; below about 1% error, the difference is about an order of magnitude. Among the other strategies, static refinement in order is the second fastest, followed by uniform refinement in $p$, followed by static refinement in $h$.

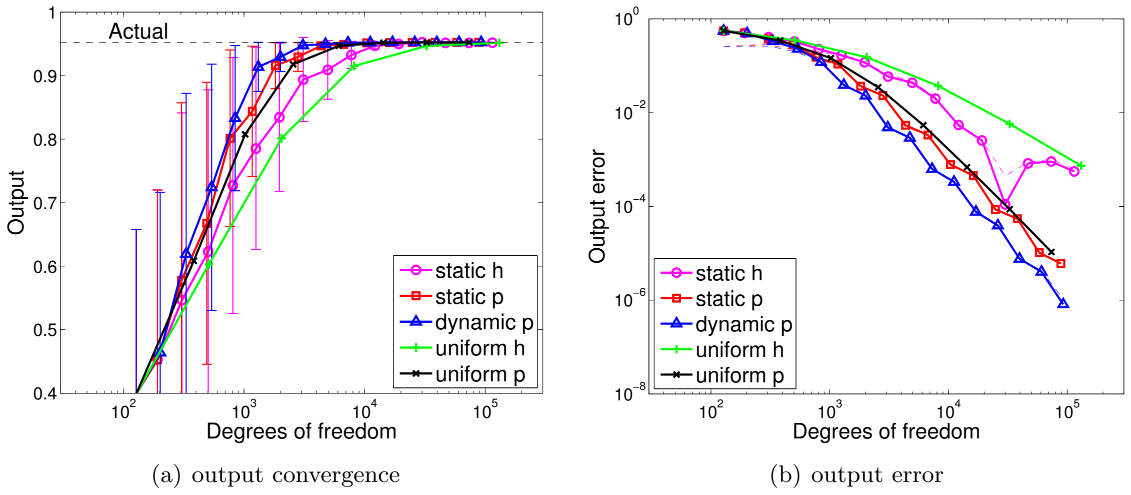
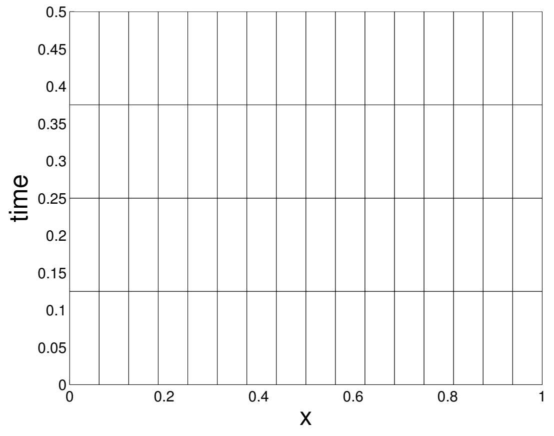


**Figure 3. 1D scalar advection: output convergence using various adaptive strategies. The actual value of 1/1.05 is calculated analytically. Error estimates are included in the output error plot as dashed lines for each output-based adaptive method – these are often coincident with the actual error values.**

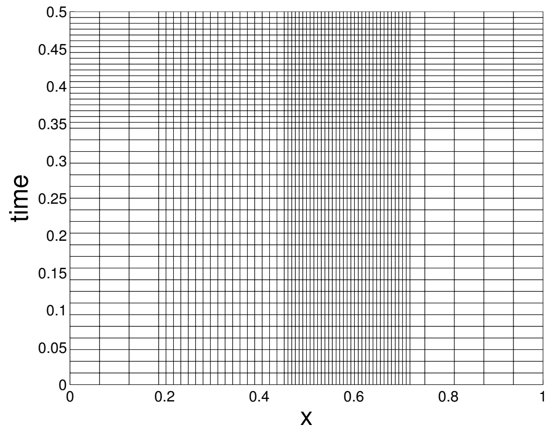
As shown in Figure 3, the degree-of-freedom benefit brought about by the dynamic order refinement strategy relative to the static order refinement strategy is observable but not ground-breaking. This is likely due to a combination of the problem being one-dimensional, the initial mesh being relatively coarse, and the initial condition being relatively diffuse. As a result, there are not too many elements left unaffected by refinement at each time step.

A more useful result from this example is the verification of the error estimation, as indicated by the accuracy of the estimated errors in Figure 3. As expected, the accuracy improves with increasing degrees of freedom. This verifies that the adjoint solution and error estimation are working properly. In addition, the discrete adjoint is verified through sensitivity tests, which are not shown in this paper.

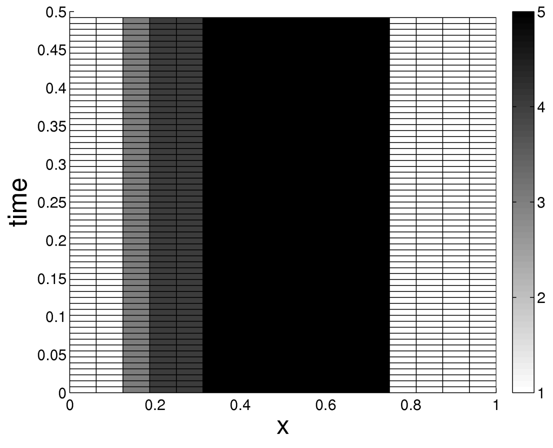
Space-time meshes for the three output-based adaptive strategies are shown in Figure 4. The temporal refinements in all three cases are similar: approximately uniform. In addition, the locations of spatial refinement in the static strategies are comparable. That is, the entire path of the advected scalar distribution is targeted for refinement, with slightly more refinement towards the latter part of the propagation. On the other hand, the dynamic order refinement strategy only targets the region in space-time where the scalar distribution is "active", as illustrated in Figure 2b. Note that this region is automatically identified as important for the prediction of the output, without any user input as to what constitutes an "active" region.



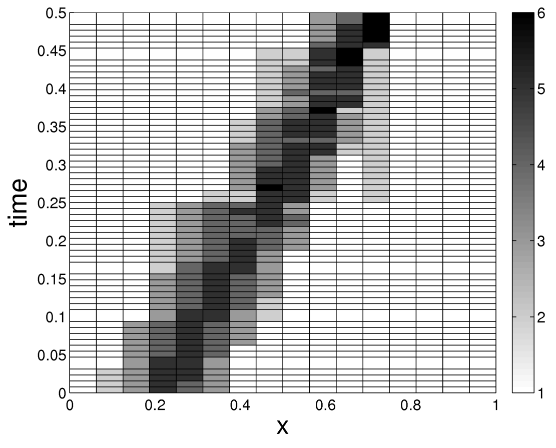
(a) initial mesh



(b) static $h$, iteration 8



(c) static $p$, iteration 8



(d) dynamic $p$, iteration 8

**Figure 4. 1D scalar advection: adapted space-time meshes.**

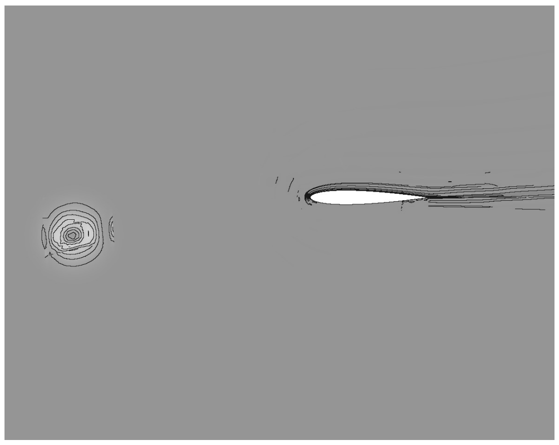
## B. An Airfoil-Vortex Encounter

In this example, the dynamic order refinement strategy is demonstrated for the compressible Euler equations. The case of interest is an encounter between a vortex and a NACA 0012 airfoil at $M = 0.1, \alpha = 5^o$. The spatial domain extends approximately 50 chords away from the airfoil, and freestream boundary conditions are applied at the farfield boundary. The initial condition is created by first converging a steady-state solution without a vortex present and then superimposing a vortex. In adaptive runs, the steady solve for the initial condition is performed on every new mesh.

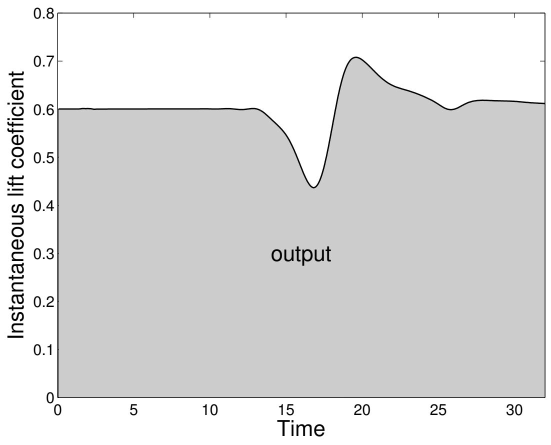
At time $t = 0$, the vortex is centered at $(x, y) = (-2.0c, -0.315c)$, where $c$ is the airfoil chord. The tangential velocity field induced by the vortex, relative to the vortex center, is

$$v_\theta(r) = \frac{\Gamma}{2\pi r}\frac{r^2}{r^2 + r_c^2},$$

where $r$ is the radial distance to the point of interest, $r_c$ is the core radius, and $\Gamma = \Lambda V_\infty r_c$ is the vortex strength. $V_\infty$ is the freestream speed, and $\Lambda$ is a non-dimensional vortex strength. In this example, the values $r_c = 0.1c$ and $\Lambda = 3.0$ were used. Also, the vortex was confined to $r < r_{\max} = 0.5c$ by using $v_\theta(r) - v_\theta(r_{\max})$. A visualization of the vortex at $t = 0$ is shown via entropy contours in Figure 5a.



(a) problem setup: entropy contours



(b) lift coefficient time history

**Figure 5. Airfoil-vortex encounter: problem setup and lift coefficient time history.**

The output of interest is the lift coefficient integral over the course of the simulation, as illustrated in Figure 5b. The temporal domain runs from $t = 0$ to $t = 32$, where time is measured in non-dimensional units in which the end of the simulation corresponds to the vortex having propagated approximately four chord lengths. As in the previous example, three output-based adaptive simulations and two uniform refinement strategies were run for this problem. The initial mesh consisted of 798 quartic curved quadrilateral elements at spatial approximation order $p = 1$ and 32 equally-spaced time slabs.

Figure 6 presents output-convergence results for the various adaptive strategies. In this example, the spread in the performance is quite large. The slowest to converge is uniform $h$ refinement, while the fastest to converge is the dynamic order refinement strategy presented in this work.

The difference in degrees of freedom exceeds two orders of magnitude as the solutions converge to a tight error tolerance. Uniform $p$ refinement is the second slowest to converge, while the static strategies perform significantly better. The large difference in the performance of uniform refinement versus the output-adapted methods is due to the relatively localized nature of the vortex interaction relative to the extent of the computational domain. Dynamic order refinement yields a further improvement by a factor of approximately 2 in degrees of freedom compared to the static refinement strategies.

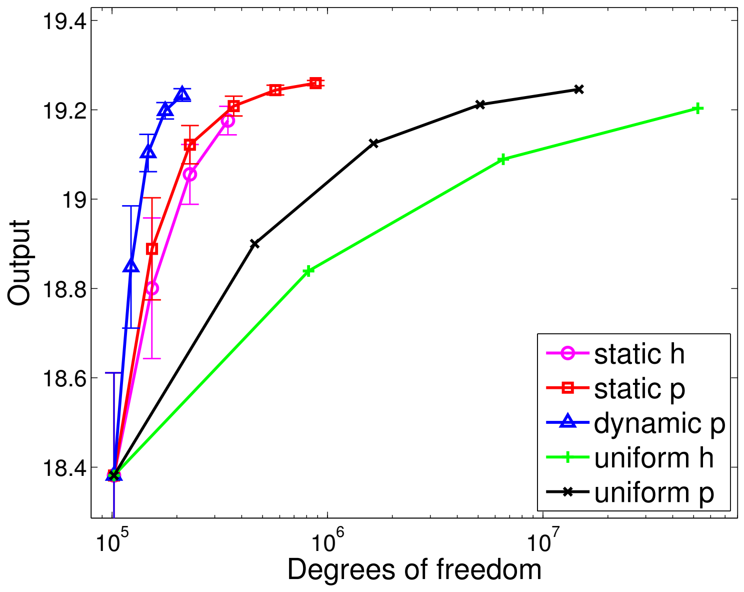
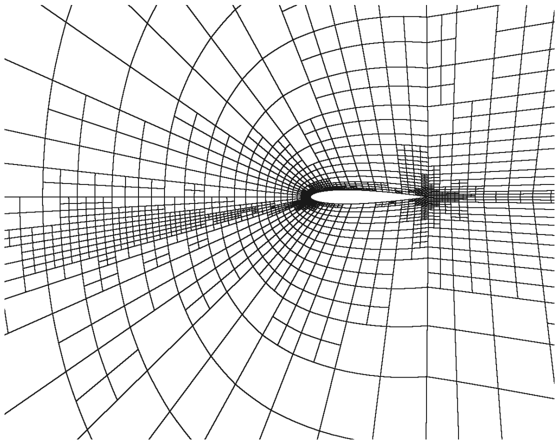


**Figure 6. Airfoil-vortex encounter: output convergence using various adaptive indicators.**

Error bars computed from the adjoint-based error estimates are also shown in Figure 6. These error estimates are not as accurate as in the previous example, underestimating the error by factors of 2-3. This effect is likely due to lower relative mesh resolution and increased non-linear nature of the Euler example.

Figure 7 shows refined spatial meshes for the "static $h$" and the "static $p$" strategies. These meshes are similar in terms of areas targeted for refinement, which include the leading edge, trailing edge, and the path taken by the vortex. These areas are deemed important for the prediction of integrated lift coefficient. However, not all of these areas are important at every time, a fact that is exploited by the dynamic order refinement strategy.

Figure 8 presents the computational mesh and solution for three points in time of the dynamic order refinement strategy: the beginning, middle, and end of the simulation. During the beginning of the simulation, both the vicinity of the airfoil and the vortex initial position are refined. In the middle of the simulation, when the vortex has already convected close to the airfoil, the initial vortex position is no longer refined. The leading and trailing edges of the airfoil are still targeted. Finally, at the end of the simulation, the mesh is relatively coarse (of low order), except at the trailing edge. This is because residual sources further away from the airfoil at this time can no longer affect the force on the airfoil. The flexibility to target different areas at different times gives the dynamic order refinement strategy the observed degree of freedom benefit compared to static refinement.


(a) static $h$, iteration 3

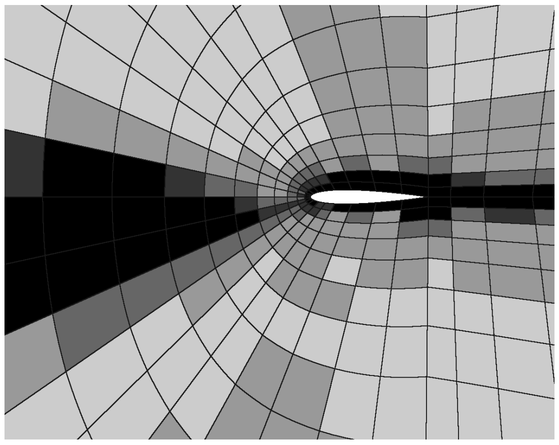

(b) static $p$, iteration 5

**Figure 7. Airfoil-vortex encounter: spatial meshes from output-based adaptive runs using static spatial refinement, in $h$ and in $p$. The order gray scale is from $p = 1$ (white) to $p = 6$ (black).**

Unsteady adjoint solution snapshots, for the $y$-momentum component, are also shown in Figure 8. We note that the areas where the adjoint is large in magnitude are locations at which residual source perturbations will have a large effect on the output.

## VI. Conclusions

This paper presents an output-based dynamic order refinement strategy for unsteady simulations using the discontinuous Galerkin finite element method. DG is chosen in both space and time for the flexibility in the solution space: dynamic order resolution changes require no interpolation or special treatment. Errors in scalar outputs are estimated using an adjoint-weighted residual technique with discrete unsteady adjoints. Adaptive indicators are derived for spatial error at the space-time element level, and for temporal error at time slabs. Results are shown for two examples: a one-dimensional scalar advection problem for verification, and a two-dimensional Euler simulation of an airfoil-vortex encounter. The proposed dynamic order refinement strategy performs favorably compared to uniform refinement and to static spatial mesh refinement, both in $h$ and in $p$. In the airfoil-vortex encounter simulation the dynamic order refinement strategy yields a degree-of-freedom savings of over two orders of magnitude compared to uniform refinement and of a factor of two to three compared to static $p$ and $h$ refinement. Future work will consider problems with mesh motion, combined $h - p$ unsteady refinement, and more sophisticated timing/cost comparisons of the proposed strategy.

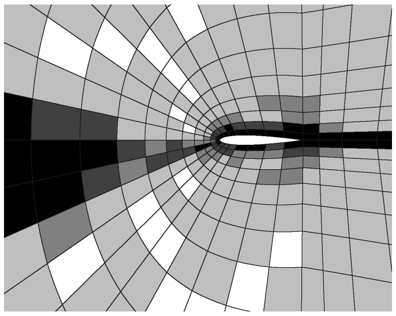
## References

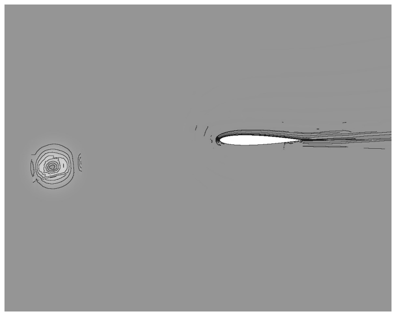
[1]Pierce, N. A. and Giles, M. B., "Adjoint recovery of superconvergent functionals from PDE approximations," *SIAM Review*, Vol. 42, No. 2, 2000, pp. 247–264.

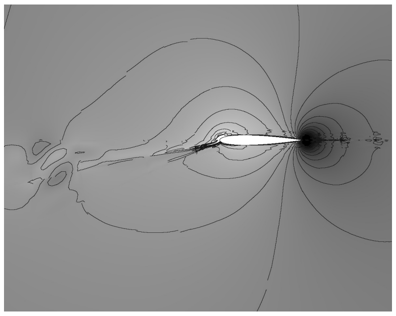
[2]Becker, R. and Rannacher, R., "An optimal control approach to a posteriori error estimation in finite element methods," *Acta Numerica*, edited by A. Iserles, Cambridge University Press, 2001, pp. 1–102.

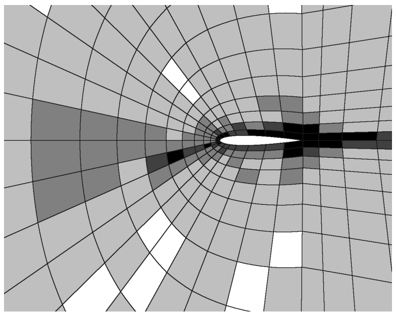
[3]Hartmann, R. and Houston, P., "Adaptive discontinuous Galerkin finite element methods for the compressible Euler equations," *Journal of Computational Physics*, Vol. 183, No. 2, 2002, pp. 508–532.

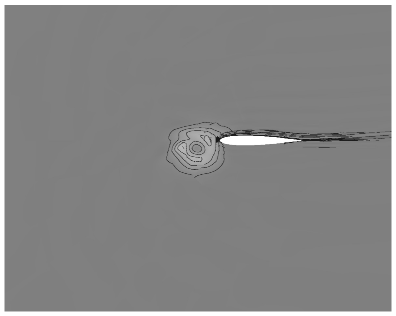
[4]Venditti, D. A. and Darmofal, D. L., "Anisotropic grid adaptation for functional outputs: application to

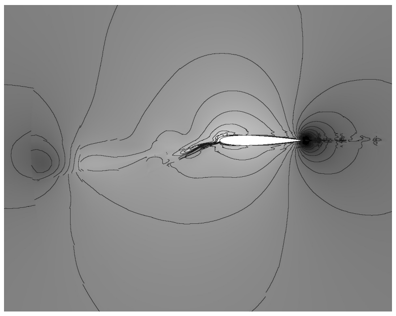

(a) orders, $t = 0$

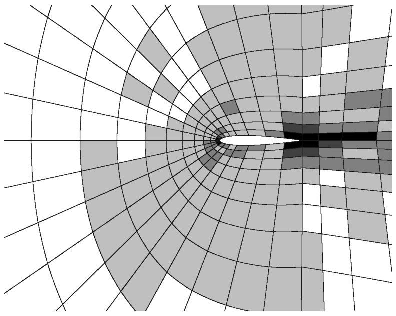

(b) entropy, $t = 0$

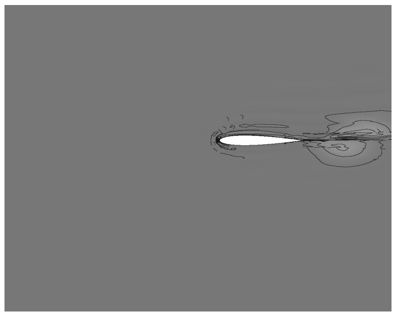

(c) $y$-momentum adjoint, $t = 0$


(d) orders, $t = 16$


(e) entropy, $t = 16$


(f) $y$-momentum adjoint, $t = 16$


(g) orders, $t = 32$


(h) entropy, $t = 32$

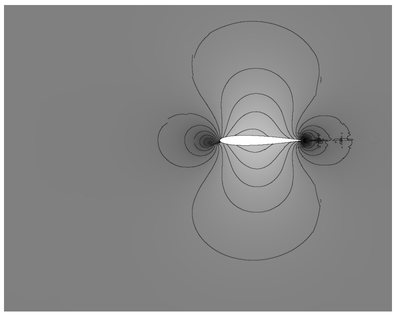

(i) $y$-momentum adjoint, $t = 32$

**Figure 8. Airfoil-vortex encounter: order distribution, entropy contours, and $y$-momentum adjoint contours at three different times for the fifth iteration of dynamic order adaptation. The order gray scale is from $p = 1$ (white) to $p = 5$ (black). The entropy and adjoint gray scales run from $-2$ (white) to 2 (black).**

two-dimensional viscous flows," *Journal of Computational Physics*, Vol. 187, No. 1, 2003, pp. 22–46.

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

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

[7]Fidkowski, K. J. and Darmofal, D. L., "Review of output-based error estimation and mesh adaptation in computational fluid dynamics," *American Institute of Aeronautics and Astronautics Journal*, Vol. 49, No. 4, 2011, pp. 673–694.

[8]Mani, K. and Mavriplis, D. J., "Discrete adjoint based time-step adaptation and error reduction in unsteady flow problems," AIAA Paper 2007-3944, 2007.

[9]Mani, K. and Mavriplis, D. J., "Error estimation and adaptation for functional outputs in time-dependent flow problems," *Journal of Computational Physics*, Vol. 229, 2010, pp. 415–440.

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

[11]Fidkowski, K. J. and Luo, Y., "Output-based space-time mesh adaptation for the compressible Navier-Stokes equations," *Journal of Computational Physics*, Vol. 230, 2011, pp. 5753–5773.

[12]Meidner, D. and Vexler, B., "Adaptive space-time finite element methods for parabolic optimization problems," *SIAM Journal on Control Optimization*, Vol. 46, No. 1, 2007, pp. 116–142.

[13]Schmich, M. and Vexler, B., "Adaptivity with dynamic meshes for space-time finite element discretizations of parabolic equations," *SIAM Journal on Scientific Computing*, Vol. 30, No. 1, 2008, pp. 369–393.

[14]Luo, Y. and Fidkowski, K., "Output-based space time mesh adaptation for unsteady aerodynamics," AIAA Paper 2011-491, 2011.

[15]Roe, P. L., "Approximate Riemann solvers, parameter vectors, and difference schemes," *Journal of Computational Physics*, Vol. 43, 1981, pp. 357–372.

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

[17]Fidkowski, K. J., "Output error estimation strategies for discontinuous Galerkin discretizations of unsteady convection-dominated flows," *International Journal for Numerical Methods in Engineering*, 2011.

[18]Griewank, A. and Walther, A., "Revolve: An implementation of checkpointing for the reverse or adjoint mode of computational differentiation," *ACM Transactions on Mathematical Software*, Vol. 26, No. 1, 2000, pp. 19–45.