50th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition
09 - 12 January 2012, Nashville, Tennessee
AIAA 2012-0078

Discrete Adjoint Based Adaptive Error Control in Unsteady Flow Problems

Bryan T. Flynt* and Dimitri J. Mavriplis†

Department of Mechanical Engineering, University of Wyoming, Laramie WY, 82071, USA

This paper presents a framework for the estimation of spatial, temporal and algebraic error in a time integrated functional (i.e. Lift, Drag, etc.) for problems of aerodynamic interest. The three types of error are separated into distinct contributions that are used to selectively adapt the discretization to reduce the error in the functional in the most efficient manner. A simple unsteady test case is used to demonstrate that the individual components of error can be separated out and properly account for the error in the calculated objective. Adaption of convergence tolerance, time step size and grid resolution are shown to reduce computational expense while improving the accuracy of the functional of interest.

I. Introduction

Achieving satisfactory results with computational fluid dynamics has traditionally been a balancing act between available computational resources and the desired accuracy the practitioner hopes to achieve. This is especially true for time accurate simulations where the number of time steps, step size, grid resolution and convergence criteria all must be chosen to accurately capture the flow physics of interest without resulting in excessive computational cost. An experienced practitioner uses past results to chose these parameters expecting similar results but has no way to directly evaluate the error resulting from their choices. Recently, adjoint based adaptive methods have been gaining popularity to provide answers to this problem for both steady and unsteady problems.

An adjoint variable based approach permits the estimation of error relevant to a functional, and the corresponding distribution of this error in the time and spatial domains. The method relies on applying discrete adjoint equations on a Taylor series expansion of a functional, where a linear approximation of the functional between two different resolutions (space or time) is estimated by solving the flow problem and the adjoint problem at the coarse resolution and then projecting it to the finer resolution. The method only predicts the error in the flow solution that is relevant to the functional of interest between two different resolutions and not against the exact analytical value of the functional. Adjoint based approaches also permit distinct identification of the contributions to this error arising from the three primary sources, namely the spatial/temporal resolution and the effect of partial convergence of the equations. This is different from common adaption or control schemes which are based on algebraic estimates of the local error within the solution, in that the global error in the functional can be estimated and adapted on.

This method is most mature for steady problems where researchers have used the error estimates to adaptively refine the grid[1–3] and put error bounds on solution accuracy.[4–6] Unsteady problems have received considerably less attention and the primary focus has been on performing adaption in time,[7–9] space[9,10] and convergence tolerances[7,8] individually but never all together in the same simulation.

The present study builds on previous work by combining all three into a unified framework for unsteady problems of aerodynamic interest. Furthermore, each of the three types of error can be separated to only adapt the discretization in the parameter needed to reduce the error most effectively. The methodology is demonstrated using an implicit cell-centered finite volume solver with gradient reconstruction and a Roe flux splitting scheme[11] to solve the Euler equations on an unstructured mesh up to 2nd order accuracy in both space and time. The following sections contain the formulations required for calculation of each of the error

*Ph.D. Candidate, AIAA Student Member, email: bflynt@uwyo.edu
†Professor, AIAA Associate Fellow, email: mavripl@uwyo.edu

estimates, implementation details and test case results showing the benefits of adaption using the adjoint based error estimates

II. Governing Equations

II.A. Euler Equations

The Euler equations are the governing equations for inviscid flow where the dissipative phenomena of viscosity and thermal conductivity are ignored. The three-dimensional equations in Cartesian coordinates for a compressible flow minus any source terms are represented by a system of five non-linear partial differential equations.

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \vec{V}) = 0 \tag{1}$$

$$\frac{\partial (\rho u)}{\partial t} + \nabla \cdot (\rho u \vec{V}) = -\frac{\partial P}{\partial x} \tag{2}$$

$$\frac{\partial (\rho v)}{\partial t} + \nabla \cdot (\rho v \vec{V}) = -\frac{\partial P}{\partial y} \tag{3}$$

$$\frac{\partial (\rho w)}{\partial t} + \nabla \cdot (\rho w \vec{V}) = -\frac{\partial P}{\partial z} \tag{4}$$

$$\frac{\partial}{\partial t}\left[\rho\left(e + \frac{V^2}{2}\right)\right] + \nabla \cdot \left[\rho\left(e + \frac{V^2}{2}\right)\vec{V}\right] = -\frac{\partial uP}{\partial x} - \frac{\partial vP}{\partial y} - \frac{\partial wP}{\partial z} + \rho \vec{f} \cdot \vec{V} \tag{5}$$

By examining the equations we see there are five equations but six unknowns (ρ, u, v, w, e, P). Therefore, in this paper an ideal gas is assumed and the equation of state relates the pressure to total energy by.

$$P = (\gamma - 1)\left[E - \frac{1}{2}\rho\left(u^2 + v^2 + w^2\right)\right] \tag{6}$$

To facilitate solution by computer the equations are manipulated into vector form by re-writing the system as:

$$\frac{\partial \vec{U}}{\partial t} + \frac{\partial \vec{F_x}}{\partial x} + \frac{\partial \vec{F_y}}{\partial y} + \frac{\partial \vec{F_z}}{\partial z} = 0 \tag{7}$$

where the vector of conserved quantities becomes:

$$\vec{U} = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ \rho E \end{pmatrix} \tag{8}$$

and the Cartesian inviscid flux vectors reduced to.

$$\vec{F_x} = \begin{pmatrix} \rho u \\ \rho u^2 + P \\ \rho uv \\ \rho uw \\ \rho u\left(E + \frac{P}{\rho}\right) \end{pmatrix} \quad \vec{F_y} = \begin{pmatrix} \rho v \\ \rho vu \\ \rho v^2 + P \\ \rho vw \\ \rho v\left(E + \frac{P}{\rho}\right) \end{pmatrix} \quad \vec{F_z} = \begin{pmatrix} \rho w \\ \rho wu \\ \rho wv \\ \rho w^2 + P \\ \rho w\left(E + \frac{P}{\rho}\right) \end{pmatrix} \tag{9}$$

The Euler equations (Eqn. 7) are solved implicitly using the second order backwards difference formula in time with second order accurate spatial derivatives on an unstructured grid. This is done for each time step by introducing an implicit residual as

$$\vec{R^n}(\vec{U^n}, \vec{U^{n-1}}, \vec{U^{n-2}}, \vec{x}) = 0 \tag{10}$$

where $\vec{U}^n$ is the solution at the current time step, $\vec{U}^{n-1}$ and $\vec{U}^{n-2}$ are the solutions at the two previous time steps and $\vec{x}$ is the vector of grid coordinates. This system is linearized with respect to the unknown solution $\vec{U}^n$ and solved using Newton's method.

$$\begin{aligned}\left[\frac{\partial \vec{R}(\vec{U}^k,\vec{x})}{\partial \vec{U}^k}\right]\delta\vec{U}^k &= -\vec{R}(\vec{U}^k,\vec{x}) \\ \vec{U}^{k+1} &= \vec{U}^k + \delta\vec{U}^k \\ \vec{U}^n &= \vec{U}^{k+1}, \delta\vec{U}^k \to 0\end{aligned} \tag{11}$$

II.B. Unsteady Adjoint

Consider an objective function that is a time integrated quantity dependent on the flow solution and mesh coordinates for all time steps in the simulation. For this case, the adjoint error will be derived by using the method of Lagrange multipliers for constrained optimization. This method is commonly used for optimization problems but we will not be optimizing any metric in this paper. Instead we only use the method to derive the adjoint variables for time accurate flows and how to solve for them in a succinct manner. Using this procedure an augmented objective J is formed by adding a penalty term $\Lambda^T R(U,D)$ onto the objective function L to ensure the governing equations (Euler equations) are solved at each time step. By taking the derivative of this augmented equation with respect to each independent variable and setting it equal to zero we will arrive at a set of equations which must be solved.

For a time integrated objective function (i.e. Lift on airfoil from $t = 0$ to $t = t_f$) the objective is a function of all time step solutions and the design variables $\vec{D}$.

$$L(\vec{U}^n, \vec{U}^{n-1}, \vec{U}^{n-2}, ..., \vec{U}^1, D) \tag{12}$$

subject to the constraints that the solution is solved at each time step (using BDF2).

$$\begin{aligned}\vec{R}^n(\vec{U}^n, \vec{U}^{n-1}, \vec{U}^{n-2}, D) &= 0 \\ \vec{R}^{n-1}(\vec{U}^{n-1}, \vec{U}^{n-2}, \vec{U}^{n-3}, D) &= 0 \\ &\vdots \\ \vec{R}^2(\vec{U}^2, \vec{U}^1, \vec{U}^0, D) &= 0 \\ \vec{R}^1(\vec{U}^1, \vec{U}^0, D) &= 0\end{aligned} \tag{13}$$

Expanding this out using the augmented equation we arrive at.

$$\begin{aligned}J(\vec{U}^n, \vec{U}^{n-1}, \ldots, \vec{U}^0, \Lambda^n, \Lambda^{n-1}, \ldots, \Lambda^0, \vec{D}) = \quad & L(\vec{U}^n, \vec{D}) \\ & +\Lambda^{T^n} R^n(\vec{U}^n, \vec{U}^{n-1}, \vec{U}^{n-2}, \vec{D}) \\ & +\Lambda^{T^{n-1}} R^{n-1}(\vec{U}^{n-1}, \vec{U}^{n-2}, \vec{U}^{n-3}, \vec{D}) \\ & \vdots \\ & +\Lambda^{T^2} R^2(\vec{U}^2, \vec{U}^1, \vec{U}^0, \vec{D}) \\ & +\Lambda^{T^1} R^1(\vec{U}^1, \vec{U}^0, \vec{D})\end{aligned} \tag{14}$$

Taking the derivative with respect to each independent variable and setting equal to zero we get a series of equations which must be satisfied. The first equation ensures that the slope of the objective is zero with

respect to the design variables.

$$\frac{\partial J}{\partial \vec{D}} = \frac{\partial L}{\partial \vec{D}} + \Lambda^{T^n} \frac{\partial R^n}{\partial \vec{D}} + \Lambda^{T^{n-1}} \frac{\partial R^{n-1}}{\partial \vec{D}} + \ldots + \Lambda^{T^1} \frac{\partial R^1}{\partial \vec{D}} = 0 \tag{15}$$

The next set of equations ensures the flow equations are solved at each step.

$$\begin{aligned}
\tfrac{\partial J}{\partial \Lambda^n} &= R^n(\vec{U}^n, \vec{U}^{n-1}, \vec{U}^{n-2}, \vec{D}) &&= 0 \\
\tfrac{\partial J}{\partial \Lambda^{n-1}} &= R^{n-1}(\vec{U}^{n-1}, \vec{U}^{n-2}, \vec{U}^{n-3}, \vec{D}) &&= 0 \\
\vdots &= \vdots &&= \vdots \\
\tfrac{\partial J}{\partial \Lambda^2} &= R^2(\vec{U}^2, \vec{U}^1, \vec{U}^0, \vec{D}) &&= 0 \\
\tfrac{\partial J}{\partial \Lambda^1} &= R^1(\vec{U}^1, \vec{U}^0, \vec{D}) &&= 0
\end{aligned} \tag{16}$$

The final set of equations defines how the unsteady adjoint variable is solved.

$$\begin{aligned}
\tfrac{\partial J}{\partial \vec{U}^n} &= \tfrac{\partial L}{\partial \vec{U}^n} + \Lambda^{T^n} \tfrac{\partial R^n}{\partial \vec{U}^n} &&= 0 \\
\tfrac{\partial J}{\partial \vec{U}^{n-1}} &= \tfrac{\partial L}{\partial \vec{U}^{n-1}} + \Lambda^{T^n} \tfrac{\partial R^n}{\partial \vec{U}^{n-1}} + \Lambda^{T^{n-1}} \tfrac{\partial R^{n-1}}{\partial \vec{U}^{n-1}} &&= 0 \\
\tfrac{\partial J}{\partial \vec{U}^{n-2}} &= \tfrac{\partial L}{\partial \vec{U}^{n-2}} + \Lambda^{T^n} \tfrac{\partial R^n}{\partial \vec{U}^{n-2}} + \Lambda^{T^{n-1}} \tfrac{\partial R^{n-1}}{\partial \vec{U}^{n-2}} + \Lambda^{T^{n-2}} \tfrac{\partial R^{n-2}}{\partial \vec{U}^{n-2}} &&= 0 \\
\vdots &= \vdots &&= 0
\end{aligned} \tag{17}$$

This series of equations can be simplified by re-arranging for a general time step k where the last term on the right hand side is zero for the second to last time step and the last two terms are zero for the last time step (i.e. the first two steps of the adjoint calculations).

$$\left[\frac{\partial R^k}{\partial \vec{U}^k}\right]^T \Lambda^k = -\left[\frac{\partial L}{\partial \vec{U}^k}\right]^T - \left[\frac{\partial R^{k+1}}{\partial \vec{U}^k}\right]^T \Lambda^{k+1} - \left[\frac{\partial R^{k+2}}{\partial \vec{U}^k}\right]^T \Lambda^{k+2} \tag{18}$$

Solving for the adjoint variables starts at the last time step of the flow solution and proceeds in reverse order solving one linear adjoint equation for every time step of the non-linear flow solution. At each time step the linear system of equations to be solved involves the transposed Jacobian matrix of that used in the solution of the analysis problem for the Euler equations (Eqn. 11). Therefore, the same linear solver is used to solve this system as previously used to solve the Euler equations.

II.C. Error due to Spatial Resolution

To understand the connection between grid refinement and the unsteady adjoint variables (Section II.B) we start by using a Taylor series expansion of the true objective calculated on a fine grid $L_s(\vec{U}_s)$.

$$L_s(\vec{U}_s) = L_s(\tilde{U}_s) + \frac{\partial L_s(\tilde{U}_s)}{\partial \tilde{U}_s}\left(\vec{U}_s - \tilde{U}_s\right) + \cdots \tag{19}$$

where $L_s(\tilde{U}_s)$ is the fine grid objective computed using an approximate solution $\tilde{U}_s$. In general, L_s may be a time integrated objective and will depend on the entire time history of the solution $\vec{U}_s$. Therefore the time level superscripts are omitted with the understanding that unsuperscripted variables refer to the entire time history.

To avoid the need for computing the true solution $\vec{U}_s$ on the fine mesh in the proceeding equation we will use a Taylor series expansion about the fine level residual equation

$$R_s(\vec{U}_s) = 0 = R_s(\tilde{U}_s) + \frac{\partial R_s(\tilde{U}_s)}{\partial \tilde{U}_s}\left(\vec{U}_s - \tilde{U}_s\right) + \cdots \tag{20}$$

where the unsuperscripted residual denotes the residuals over all time steps and all mesh cells of the simulation. Using this expression to substitute for the term $\left(\vec{U}_s - \tilde{U}_s\right)$ we can rewrite the objective as:

$$L_s(\vec{U}_s) \cong L_s(\tilde{U}_s) - \left[\frac{\partial L_s(\tilde{U}_s)}{\partial \tilde{U}_s}\right]\left[\frac{\partial R_s(\tilde{U}_s)}{\partial \tilde{U}_s}\right]^{-1} R_s(\tilde{U}_s) \tag{21}$$

Denoting the matrix product with the variable ${\Lambda_s}^T$

$${\Lambda_s}^T = -\left[\frac{\partial L_s(\tilde{U}_s)}{\partial \tilde{U}_s}\right]\left[\frac{\partial R_s(\tilde{U}_s)}{\partial \tilde{U}_s}\right]^{-1} \tag{22}$$

leads to the following expression for Λ_s:

$$\left[\frac{\partial R_s(\tilde{U}_s)}{\partial \tilde{U}_s}\right]^T \Lambda_s = -\left[\frac{\partial L_s(\tilde{U}_s)}{\partial \tilde{U}_s}\right]^T \tag{23}$$

This indicates that Λ_s is the solution of the discrete adjoint problem corresponding to the time integrated objective L_s. Therefore, Λ_s may be solved using the backwards recurrence relation in time given by equation (18). Using the adjoint variable, the linear approximation to the fine grid objective can be re-written as

$$L_s(\vec{U}_s) \cong L_s(\tilde{U}_s) + {\Lambda_s}^T R_s(\tilde{U}_s) \tag{24}$$

where the inner product given by the last term is taken over all mesh cells and all time steps of the simulation. The values in this expansion are obtained by constructing the approximate solution $\tilde{U}_s$ by interpolating a coarse grid solution onto the fine grid at each time step and evaluating $L_s(\tilde{U}_s)$ and $R_s(\tilde{U}_s)$. The difficulty lies in obtaining the adjoint value Λ_s^T. A direct calculation of this quantity costs as much as obtaining the true solution $\vec{U}_s$, in which case, we might as well evaluate $L_s(\vec{U}_s)$ directly. To alleviate this expense we approximate the fine grid adjoint value by calculating the coarse grid adjoint denoted as Λ and interpolating onto the fine grid using a projection matrix (I_S^s) to arrive at the final linear approximation to the fine grid objective function.

$$\begin{gathered} \tilde{\Lambda}_s^T = I_S^s \Lambda^T \\ L_s(\vec{U}_s) \cong L_s(\tilde{U}_s) + \tilde{\Lambda}_s^T R_s(\tilde{U}_s) \end{gathered} \tag{25}$$

From this derivation we see that the term $\tilde{\Lambda}_s^T R_s(\tilde{U}_s)$ is a direct estimation of the linear error in our calculation from the lack of spatial resolution. Therefore, the elements with the largest value can be flagged for future refinement.

II.D. Error due to Temporal Resolution

The use of the adjoint variable to identify errors in temporal resolution follows a similar path to that of spatial error (Section II.C). We start by using a Taylor series expansion of the true time integrated objective calculated using a small time step $L_t(\vec{U}_t)$

$$L_t(\vec{U}_t) = L_t(\tilde{U}_t) + \frac{\partial L_t(\tilde{U}_t)}{\partial \tilde{U}_t}\left(\vec{U}_t - \tilde{U}_t\right) + \cdots \tag{26}$$

where $L_t(\tilde{U}_t)$ is the small time step objective calculated with an approximate solution. The need to calculate the true small time-step solution $\vec{U}_t$ is eliminated by using a Taylor series expansion of the small time step residual equation:

$$R_t(\vec{U}_t) = 0 = R_t(\tilde{U}_t) + \frac{\partial R_t(\tilde{U}_t)}{\partial \tilde{U}_t}\left(\vec{U}_t - \tilde{U}_t\right) + \cdots \tag{27}$$

Following the steps previously presented we can arrive at the expression for the small time step objective using an approximate small time step solution $\tilde{U}_t$ obtained by projecting a coarse time step solution to the finer time step space:

$$L_t(\vec{U}_t) \cong L_t(\tilde{U}_t) + {\Lambda_t}^T R_t(\tilde{U}_t) \tag{28}$$

The requirement for the small time step adjoint solution (Λ_t) is relaxed by approximating it using the adjoint solution from a coarse time step solution. This results in an equation that is a linear approximation to the small time step objective using only solutions from the coarse time step simulation.

$$L_t(\vec{U}_t) \cong L_t(\tilde{U}_t) + \tilde{\Lambda}_t^T R_t(\tilde{U}_t) \tag{29}$$

II.E. Error due to Partial Convergence

In the former sections we assumed the solution on the coarse space and time domain was obtained by full convergence of the flow equations at each implicit time step resulting in a coarse level residual equation that evaluates to zero. If the flow equations were only partially converged at each time step, the fully converged objective can be linearly approximated using the the Taylor series expansion:

$$L(\vec{U}) = L(\tilde{U}_c) + \frac{\partial L(\tilde{U}_c)}{\partial \tilde{U}_c}\left(\vec{U} - \tilde{U}_c\right) + \cdots \tag{30}$$

where the objective $L(\tilde{U}_c)$ is obtained through partially converging the flow equations to an approximate solution $\tilde{U}_c$. Again, we use the expansion of the residual equation to eliminate the need for the fully converged solution $\vec{U}$ and introduce an approximate adjoint variable into the equation.

$$L(\vec{U}) \cong L(\tilde{U}_c) + \tilde{\Lambda}_c^T R(\tilde{U}_c) \tag{31}$$

The approximate adjoint $\tilde{\Lambda}_c$ is obtained by partially converging the adjoint equations using the same mesh and time step distribution as for the analysis problem. The contribution to the integrated linear error (ε_c) for the partially converged solution can be obtained by re-arranging the equation.

$$\underbrace{L(\vec{U}) - L(\tilde{U}_c)}_{\varepsilon_c} \cong \tilde{\Lambda}_c^T R(\tilde{U}_c) \tag{32}$$

It is of interest to note the residual $R(\tilde{U}_c)$ is non-zero only because the flow equations are not fully converged. If the system is fully converged, the residual is zero and the approximate objective is no longer approximate but the true objective.

II.F. Combined Error

Our goal is to estimate the total error in a time dependent simulation objective and to determine the respective contributions to the total error from spatial discretization, temporal discretization, and incomplete convergence in order that adaptive refinement methods may be used to efficiently reduce the error. Using a Taylor series expansion, we can estimate the exact objective L_{st} computed on a fine time step and fine mesh with full convergence using an approximate solution $\tilde{U}_{cst}$ computed on a coarser mesh with a larger time step with partial convergence as:

$$L_{st}(\vec{U}_{cst}) = L_{st}(\tilde{U}_{cst}) + \frac{\partial L_{st}(\tilde{U}_{cst})}{\partial \tilde{U}_{cst}}\left(\vec{U}_{cst} - \tilde{U}_{cst}\right) + ... \tag{33}$$

In the above expression, $\tilde{U}_{cst}$ is obtained from the partially converged approximate solution $\tilde{U}_c$ computed on a coarser grid with a larger time step, projected onto the finer mesh and time step space. Although the above equation could be used to obtain an expression for the total error, it does not provide a mechanism for separating out the various error components. However, since the adjoint error estimation procedure relies on a linearization, we can assume these different error components are additive. Therefore, the error in the solution may be written as:

$$\vec{U}_{cst} - \tilde{U}_{cst} = \vec{U}_s - \tilde{U}_s + \vec{U}_t - \tilde{U}_t + \vec{U} - \tilde{U}_c \tag{34}$$

Here $\vec{U}_s - \tilde{U}_s$ corresponds to the difference between the solution computed on the fine grid and the approximate coarse level solution $\tilde{U}_c$ projected onto the fine grid with a fixed time step value. Similarly, $\vec{U}_t - \tilde{U}_t$ corresponds to the change in the solution when the time step is refined, with all other simulation parameters (mesh size, convergence tolerance) remaining fixed, and $\vec{U} - \tilde{U}_c$ corresponds to the change in the coarse mesh and time step solution that would be observed if full convergence was enforced at each time step. Clearly,

interactions between these various sources of error will exist in actual simulations, but these will be higher order non-linear error interactions that cannot be accounted for in an adjoint formulation. Furthermore, as the errors are reduced, the additive assumption will become asymptotically more exact. Inserting the above expression into equation (33), and making use of an adjoint solution computed on the coarse space and time discretization level at the partially converged state leads to the following expression for the objective error:

$$L_{st}(\vec{U}_{cst}) \cong L_{st}(\tilde{U}_{cst}) + \tilde{\Lambda}_s^T R_s(\tilde{U}_s) + \tilde{\Lambda}_t^T R_t(\tilde{U}_t) + \tilde{\Lambda}_c^T R(\tilde{U}_c) \tag{35}$$

where the adjoint variables $\tilde{\Lambda}_s$, and $\tilde{\Lambda}_t$ correspond to the approximate coarse level adjoint $\tilde{\Lambda}_c$ projected onto the fine mesh and fine time step domain, respectively. The various residual operators R, R_s and R_t correspond to the space-time residuals evaluated on the coarse mesh and time step domain, the fine mesh domain (with coarse time step), and the fine time step domain (with coarse mesh), respectively. The one remaining term that is not readily available is the fine time and space approximate objective value $L_{st}(\tilde{U}_{cst})$. Instead of separately constructing this term it is formed using a linear combination of the terms already used for each individual error correction. To do this we let the variables $(\delta s, \delta t)$ represent the difference between the individual fine domain objective values calculated using approximate solutions and the coarse approximate objective value $L(U_c)$ already calculated.

$$\begin{aligned} \delta s &= L(U_c) - L_s(\tilde{U}_s) \\ \delta t &= L(U_c) - L_t(\tilde{U}_t) \end{aligned} \tag{36}$$

Using these two values we can linearly predict what the approximation would be for the combined approximation $L_{st}(\tilde{U}_{cst})$:

$$L_{st}(\tilde{U}_{cst}) = L(U_c) - \delta s - \delta t \tag{37}$$

Inserting this approximation into the combined linear approximation from earlier (Eqn. 35) we get the final form of the combined approximation.

$$L_{st}(\vec{U}_{cst}) \cong L_s(\tilde{U}_s) + L_t(\tilde{U}_t) - L(U_c) + \tilde{\Lambda}_s^T R_s(\tilde{U}_s) + \tilde{\Lambda}_t^T R_t(\tilde{U}_t) + \tilde{\Lambda}_c^T R(\tilde{U}_c) \tag{38}$$

III. Implementation Details

III.A. Mesh Refinement

In the current work the conformal nature of a mixed element grid is maintained by using element subdivision for refinement and un-refinement.[12] This is accomplished using a set of allowable subdivision types for each element as shown in Figure 1. An element that is flagged for refinement is isotropically split into 4 elements. The non-flagged neighboring element is partially subdivided using an allowable pattern to keep the grid conformal. On subsequent steps, if one of the elements resulting from a partial subdivision is flagged for refinement, the partial element is collapsed back into the parent element which is then isotropically split into 4 smaller elements. This method of refinement limits the formation of sliver elements within the grid and lends itself to a simple tree data structure which can be unrolled to un-refine elements within the grid. Furthermore, no smoothing of the adaptive grid is performed so that un-refinement of elements will result in the starting grid.

III.B. Spatial Projection

To project the solutions to grids of different spatial refinement levels a general interpolation scheme is used based on compact radial basis functions.[13] Specifically, the compact functions of Wendland[14] are used with an additional polynomial constraint to ensure 2nd order accuracy for all cases. The approximations have the general form

$$s(\vec{x}) = \sum_{j=1}^{N} \alpha_j \phi\left(\vec{x} - \vec{x_j}\right) + p(\vec{x}) \tag{39}$$

where α_j is the weight of the known value at element j, $\phi\left(\vec{x} - \vec{x_j}\right)$ is the radial function evaluated between the two centers and $p(\vec{x})$ is a polynomial of the required order of accuracy. The required number of nearest neighbors is found using an alternating digital tree[15] followed by a linear search over the remaining candidates.

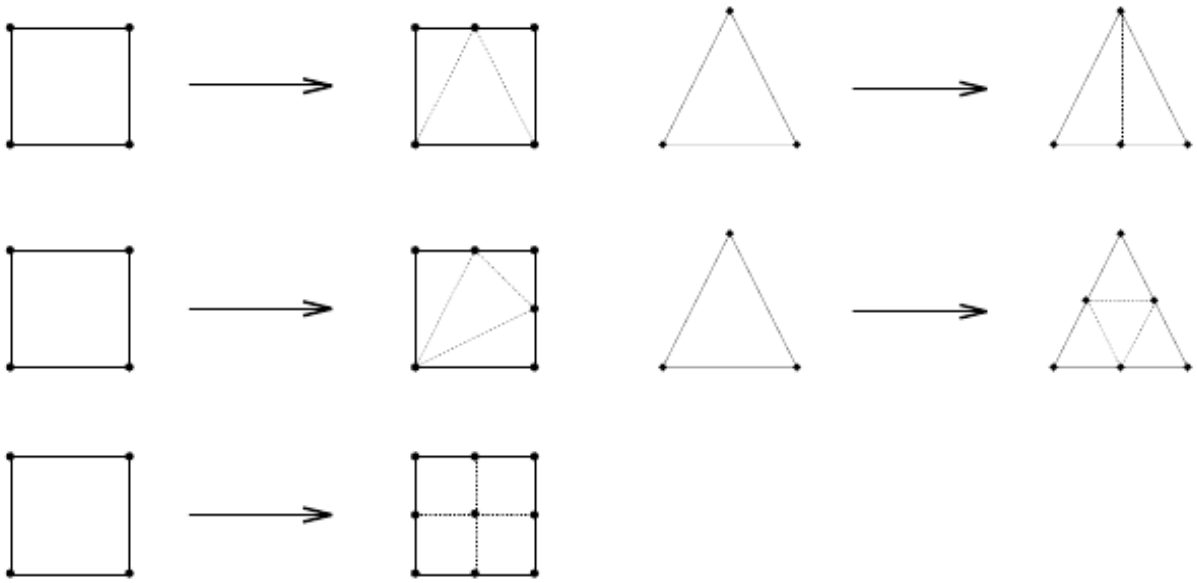

Figure 1. Allowable subdivision patterns for triangle and quadrilateral cells

III.C. Temporal Projection

For the temporal refinement a 2-to-1 refinement pattern is enforced where a single time interval is split into 2 equal intervals. In addition, any adjacent time intervals that vary more than 2 times the size of the current time interval will be flagged for splitting to ensure no large jumps in the time step size exist.

To perform the temporal projections from coarse to fine time steps linear interpolation is used to arrive at the intermediate solution.

$$\begin{aligned} \tilde{\Lambda}_t^{n-\frac{1}{2}} &= \tfrac{1}{2}\left(\Lambda_T^n + \Lambda_T^{n-1}\right) \\ \tilde{\vec{U}}_t^{n-\frac{1}{2}} &= \tfrac{1}{2}\left(\vec{U}_T^n + \vec{U}_T^{n-1}\right) \end{aligned} \tag{40}$$

III.D. Solver Details

The adaptive solver consists of three parts namely the forward flow solver, backward adjoint solver and a time/grid adaption module. The projection operations are done within the adaption module while the error computations are built into the adjoint solver. The simulation starts by the using the adaption module to write out all grids for all time steps to file for the current (first) sweep. The flow solver then performs the forward integration in time by reading in the grid for the new time step and projecting the previous solution to it for the initial condition and time history information. In our current case, no projection is needed since the grid is constant for all time steps but the functionality is included for future research. The adjoint solver reads in the flow solution and associated grid and performs the backwards sweep in time to compute the adjoint variables at each time step and saves them to file. Next the adaption module will create an isotropically refined grid for each of the fine domain time steps and perform the necessary interpolation onto each. These are saved to disk so that the flow solver followed by the adjoint solver can read them in and calculate the fine grid approximate objective solution, evaluate the residual and form the contribution to the error for each grid element of each time step and save them to file. In addition, smoothing of the approximate flow or adjoint solution can be done by their respective solver during this part of the process. For this research only the spatially projected approximate adjoint was smoothed by one order of magnitude in the L_2 norm for each time step. Finally, the individual error contributions are read from disk one time step at a time and the error is used to flag time intervals and grid elements for refinement based on the desired refinement criteria selected. If another more refined simulation is desired the adaption module will again generate and save out a newly refined grid for each adapted time step and the process will repeat again.

IV. Results

IV.A. Validation Test Case

The test case used to verify the developed algorithm involves the time integrated density inside a region of the flow field as a 2-D inviscid isentropic vortex[16,17] convects through uniform flow. In this case the mean flow density (ρ_∞), velocity (u_∞,v_∞), pressure (P_∞), and temperature (T_∞) are taken to be free stream values. We set these flow parameters to be (ρ_∞,u_∞,u_∞,P_∞,T_∞) = (1.0,0.5,0.0,1.0,1.0) with characteristic boundary conditions[18] on all free stream boundaries. At the initial time the flow is perturbed by an isentropic vortex $(\delta u, \delta v, \delta T)$ centered at $(x_o, y_o) = (-15.0, 0.0)$ with the form:

$$\begin{aligned} \delta u &= -\frac{V_m}{R_c}(y - y_o)\, e^{\left[\frac{1}{2}-\left(\frac{r}{2R_c}\right)^2\right]} \\ \delta v &= \frac{V_m}{R_c}(x - x_o)\, e^{\left[\frac{1}{2}-\left(\frac{r}{2R_c}\right)^2\right]} \\ \delta T &= -\frac{V_m^2(\gamma-1)}{2\gamma R_c^2} e^{\left[1-\left(\frac{r}{R_c}\right)^2\right]} \end{aligned} \tag{41}$$

where V_m is the maximum perturbed velocity, R_c is the distance r that this maximum velocity occurs at from the vortex center and $\gamma = 1.4$ is the ratio of specific heats of the fluid. From the relationship for an ideal gas and assuming isentropic flow the density is found for every point in the domain as:

$$\rho = T^{1/(\gamma-1)} = (T_\infty + \delta T)^{1/(\gamma-1)} \tag{42}$$

In this case the strength of the vortex and size were specified by assigning the values of 0.2 to the maximum velocity ($V_m = 0.2$) at a core radius of 0.5 ($R_c = 0.5$) grid units.

To properly validate the results a single quantity is needed as the objective function and for this we defined the objective (Eqn. 43) to be the time and space integrated density inside a square box centered about the origin $(x, y) = (0, 0)$ with equal length sides of 2 grid units. The vortex is convected for 60 non-dimensional time units allowing it to travel from the starting position of $(x_o, y_o) = (-15.0, 0.0)$ to the final position of $(x, y) = (15.0, 0.0)$ and being centered in the integrated region exactly half way through the computation. A portion of the coarsest grid with the prescribed initial condition and highlighted integration region is shown in figure 2. The exact analytic solution to this problem was found using a common computer math program to be 239.52558800471 which we will compare against as the true analytic solution in subsequent analysis.

$$L(U) = \int_0^{60}\int_{-1}^{1}\int_{-1}^{1} \rho \; \mathrm{d}x\mathrm{d}y\mathrm{d}t \tag{43}$$

IV.B. Convergence Threshold Adaption

The convergence threshold computations were run using a grid produced from isotropically refining the base grid (Figure 2) 4 times to produce a grid with 301056 elements and 128 equal sized time steps to complete the vortex convection. The vortex perturbation is assigned as the initial condition and each time step of the flow and adjoint solution is terminated at the same tolerance using the L_2 norm of the residual as the indicator. The true solution for the error calculation is the objective measured at the full convergence of $L_2 \parallel R^n(U^n) \parallel < 10^{-12}$. Figure 3 shows using the adjoint correction reduces the error in the objective function by roughly one order of magnitude. The rate of convergence is slightly better than linear as a result of the adjoint value being partially converged to the same tolerance as the flow solution at each step. If the adjoint values were converged to machine zero while still maintaining the partial convergence of the flow solution, a 2nd order quadratic convergence rate would be expected, as is observed with an exact Newton solver. Furthermore, this plot shows we can set the convergence tolerance to 10^{-7} and by using the adjoint correction achieve an objective solution that is within machine zero of the true solution.

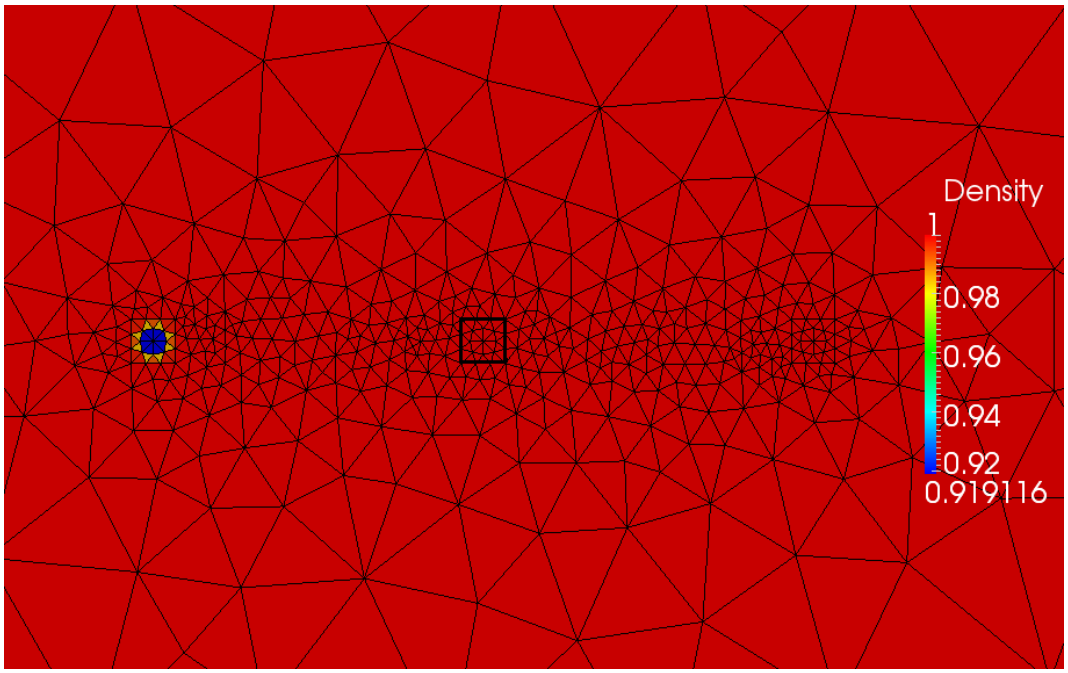

Figure 2. Initial density distribution with highlighted integrated region for grid of 1176 elements

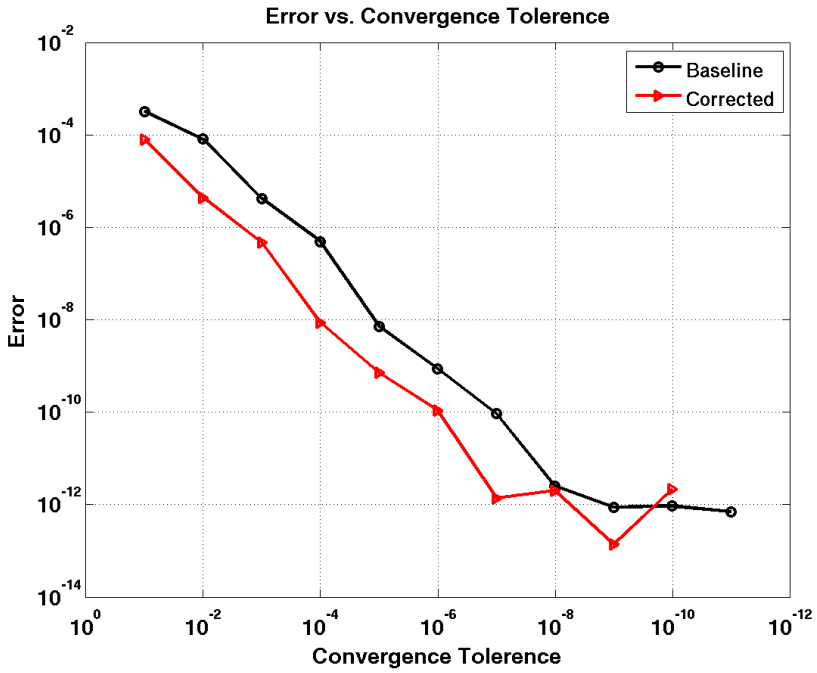

Figure 3. Algebraic Error

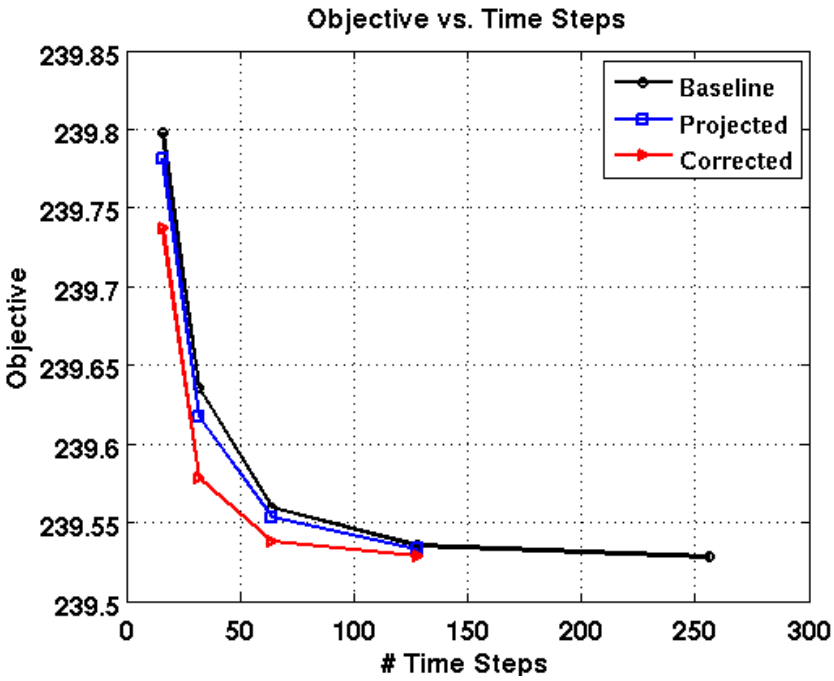

Figure 4. Objective for uniform adapted time steps

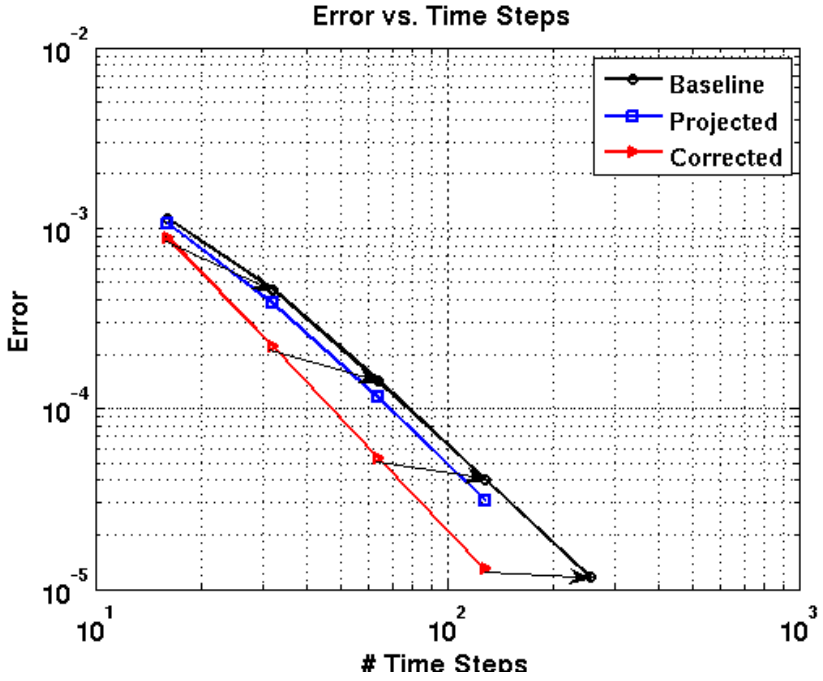

Figure 5. Error for uniform adapted time steps

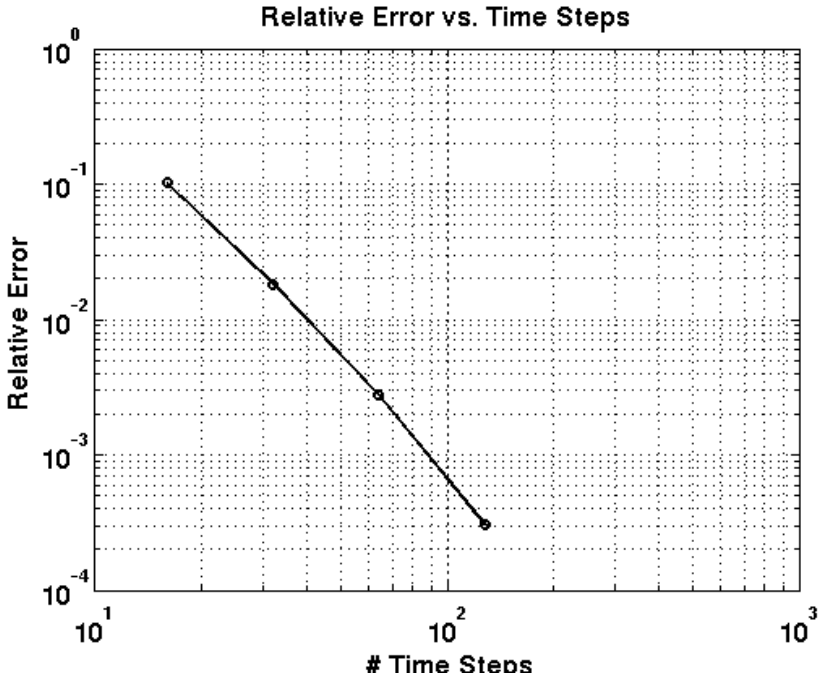

Figure 6. Relative error for uniform adapted time steps

IV.C. Temporal Adaption

IV.C.1. Uniform Refinement

In the case of temporal error estimation the initial calculation is performed using 16 uniform time steps and subsequent refinements equally divide each time step to achieve higher temporal resolution. To reduce the contribution from spatial error and incomplete convergence the 301056 element isotropically refined grid from section IV.B was converged to a tolerance of 10^{-7} at each time step. The calculated baseline objective ($L_t^n(\vec{U}_t^n)$), projected objective ($L_t^n(\tilde{U}_t^n)$) and corrected objective ($L_t^n(\tilde{U}_t^n) + \Lambda_t^{n^T} R_t^n(\tilde{U}_t^n)$) values at each discretization are plotted in Figure 4. The error in these prediction is plotted in Figure 5 where they converge with a slope of 1.81 for the 2nd order accurate scheme.

To quantify the difference in the adjoint correction formula we plot the error as measured with respect to the next refined solution after the adjoint correction in Figure 6. This relationship is represented by arrows in Figure 5 and is referred to as the relative error calculated using the equation:

$$RelativeError = \left| L_t^n(\vec{U}_t^n) - \left[L_t^n(\tilde{U}_t^n) + \Lambda_t^{n^T} R_t^n(\tilde{U}_t^n) \right] \right| \tag{44}$$

From this comparison we observe the relative error is superconvergent with a convergence rate of 3.1 where formally we would expect a 4th order convergence rate. This decrease from the formal superconvergent rate has been observed by others[3] and can be attributed to interpolating regions of the adjoint variable that are non-smooth.

IV.C.2. Non-Uniform Refinement

To test the efficiency gains from using the adaptive temporal refinement procedure we use the same starting discretization as uniform refinement (Section IV.C.1). At the completion of each time sweep we rank all time intervals by the absolute value of the error they contribute to the final integrated objective and refine them until a defined percentage of the error is accounted for. In Figures 7 and 8 we have plotted the objective values and the error with respect to the exact solution as a function of the number of time steps when the percentage of error refined on the subsequent sweep is 100, 99, 95, 85, 50 percent. The plots show the higher refinement percentages (99%, 95%) correct most of the error that uniform refinement (100%) accounts

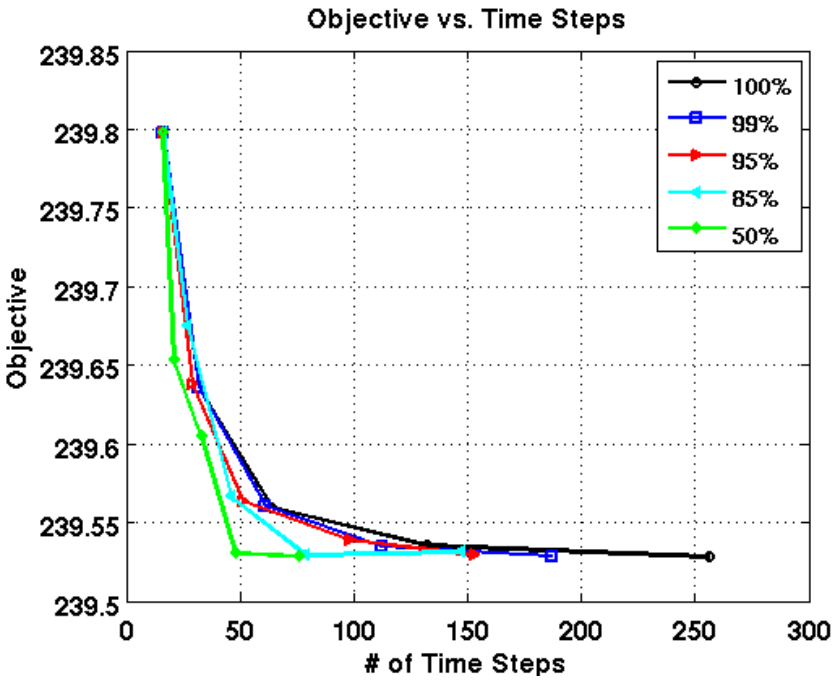

Figure 7. Objective for non-uniform temporal refinement

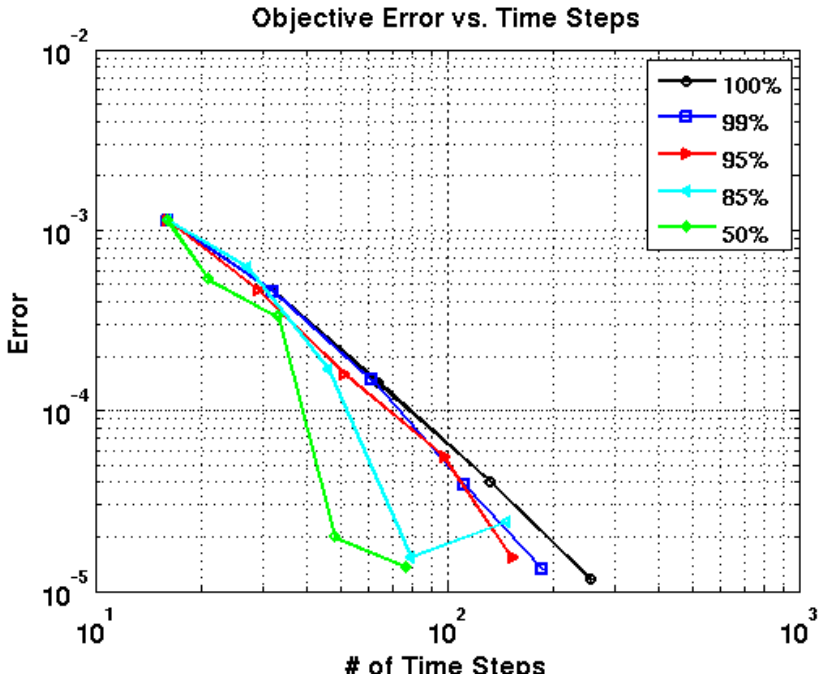

Figure 8. Error in objective for non-uniform temporal refinement

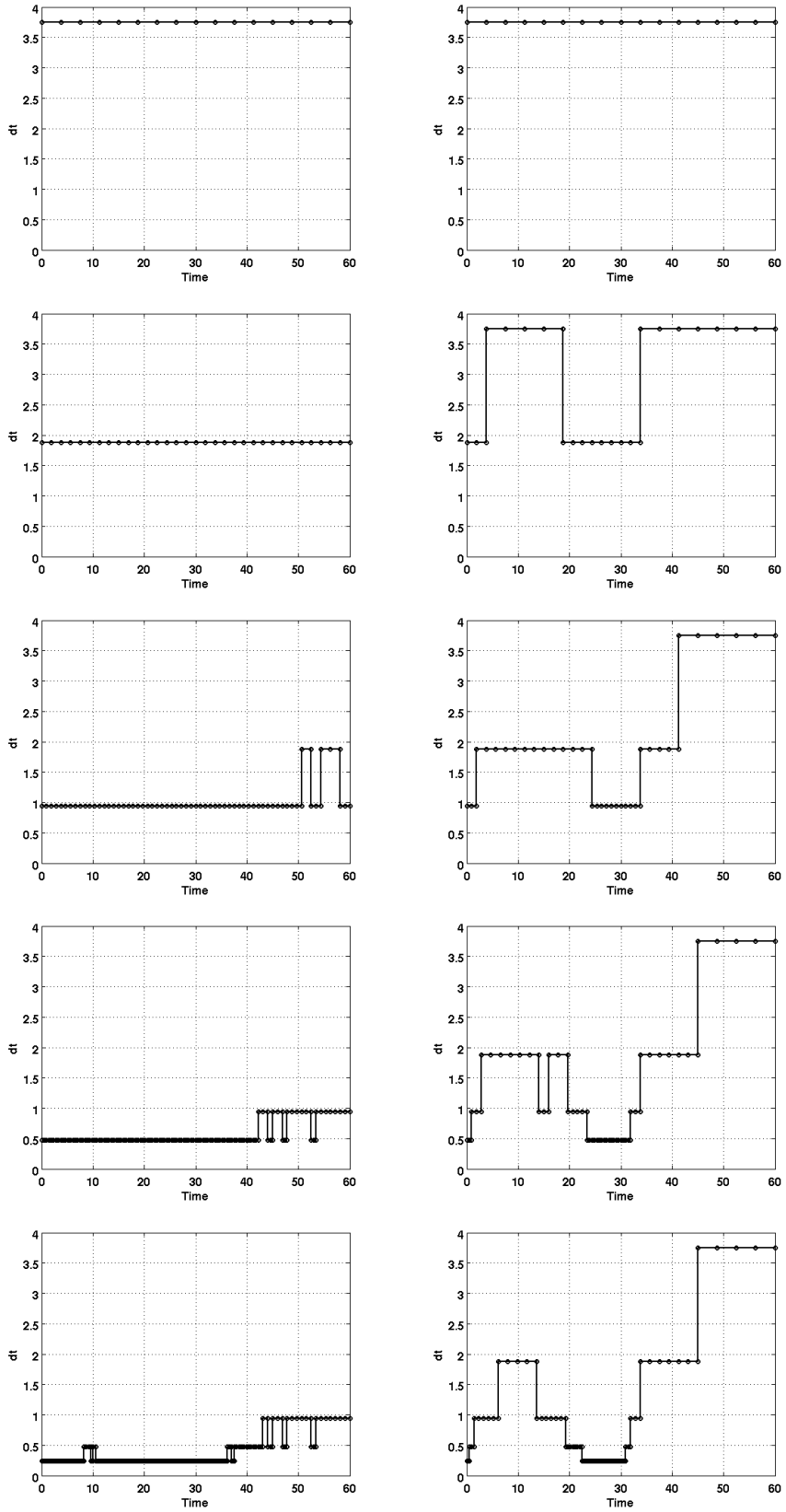

Figure 9. Time step size as a function of simulation time. Left column is for 99% refinement starting at the top with the initial sweep of 16 time steps down through the 5 time refinement sweeps to finish with 187 time steps. Right column is for 50% refinement starting with 16 equal time intervals and finishing with the bottom right image showing 76 non-uniform time steps.

for but requires fewer time steps to complete the simulation. As the percentage of refined error is lowered $(85\%, 50\%)$ the efficiency gains can be more significant but a single refinement sweep doesn't always result in a more accurate solution. This loss in predictability is a result of the cancellation of the errors, since we are using the absolute value of the error to rank the most sensitive time intervals, and doesn't always result in a more efficient solution from one refinement to the next.

The non-uniform step size is plotted in Figure 9 for 5 full sweeps of the simulation when the percentage of error to refine is set at 99% and 50%. The plots show that refining 99% of the error at the completion of each sweep flags nearly all time intervals for refinement. As the time steps of subsequent simulations are refined the period after the vortex passes the integrated region becomes less sensitive to the time step size and the individual time steps are not targeted for refinement as heavily in this section. The time steps for the 50% refinement show similar trends but since the simulation isn't constrained to refine as much error each sweep we see there are actually two regions that develop where the time step can be larger without affecting the integrated objective function.

IV.D. Spatial Adaption

IV.D.1. Uniform Refinement

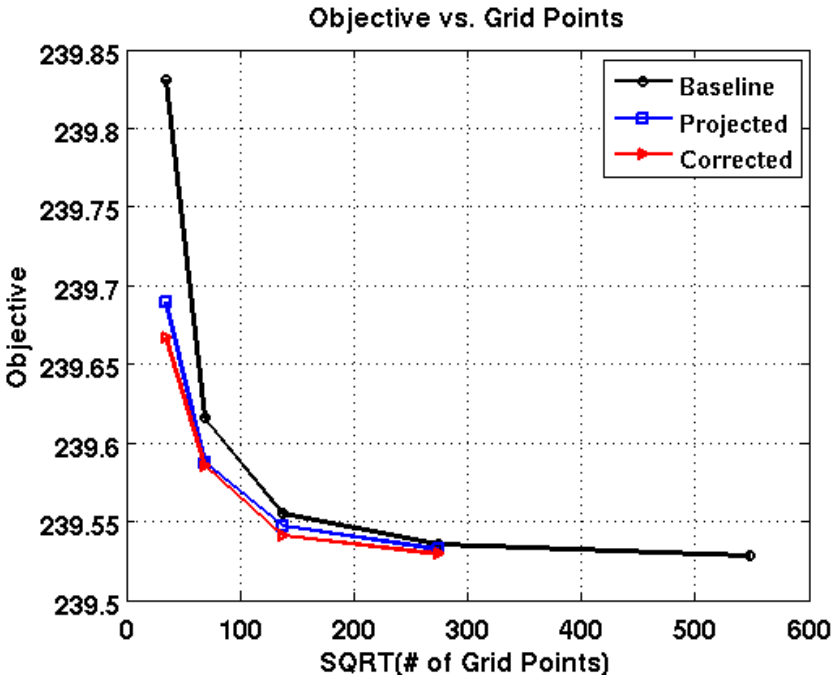

Figure 10. Objective for uniform adapted grid elements

To test the spatial error estimation the initial calculation is performed using a coarse grid of 1176 elements over 256 equal spaced time steps and the L_2 norm of the residual is converged to a tolerance of 10^{-7}. These parameters are chosen to allow the spatial error to be adapted on while minimizing the temporal and incomplete convergence error in the calculations. After a complete solution at one grid resolution is calculated the linear predictions about the next refined level are made. Then the grid is isotropically refined and this new refined grid is marched through the time steps to get the solution on the refined grid. The calculated baseline objective $(L_s^n(\vec{U}_s^n))$, projected objective $(L_s^n(\tilde{U}_s^n))$ and corrected objective $(L_s^n(\tilde{U}_s^n) + \Lambda_s^{nT} R_s^n(\tilde{U}_s^n))$ values at each grid resolution are plotted in Figure 10. The error in these prediction is plotted in Figure 11 where the last segment of the corrected objective converges with a slope of 1.79 for the 2nd order accurate scheme.

As with the temporal error we can quantify the error in the spatial adjoint correction equation (Eqn. 24) by calculating the error with respect to the next refined solution after the adjoint correction. For the spatial

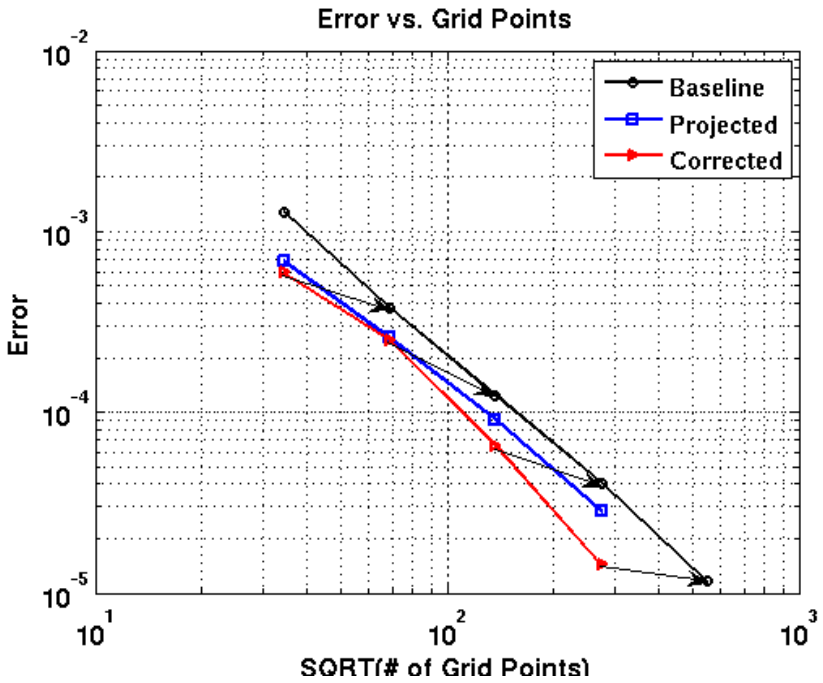

Figure 11. Error for uniform adapted grid elements

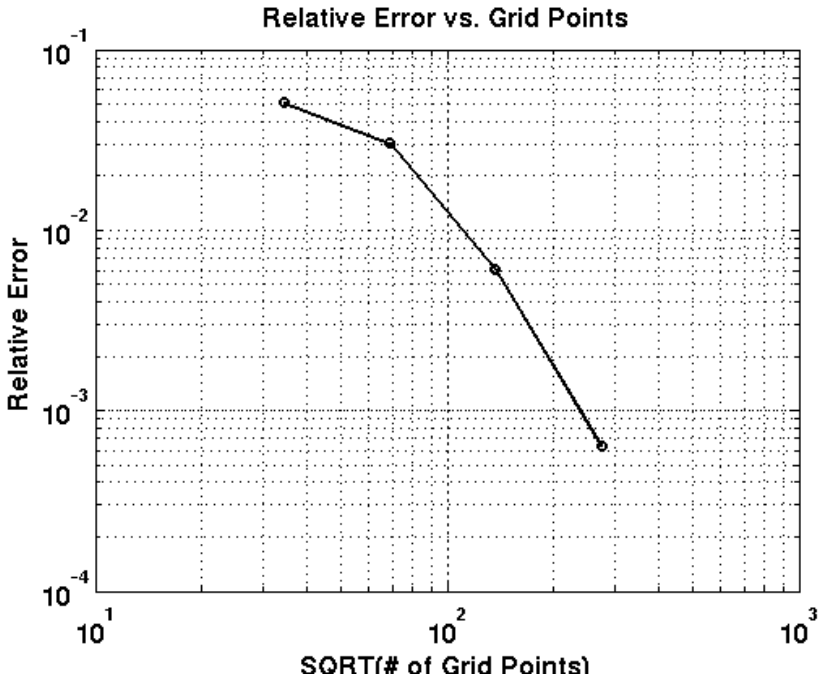

Figure 12. Relative error for uniform adapted grid elements

relative error we calculate this as:

$$RelativeError = \left| L_t^n(\vec{U}_s^n) - \left[L_s^n(\tilde{U}_s^n) + \Lambda_s^{nT} R_s^n(\tilde{U}_s^n) \right] \right| \tag{45}$$

Looking at this plot we observe the relative error for the spatial component is also superconvergent with a convergence rate of 3.2. Again, the decrease from the formal superconvergent rate (4th order) is a result of interpolating regions of the adjoint variable that are non-smooth within the spatial domain.

IV.D.2. Non-Uniform Refinement

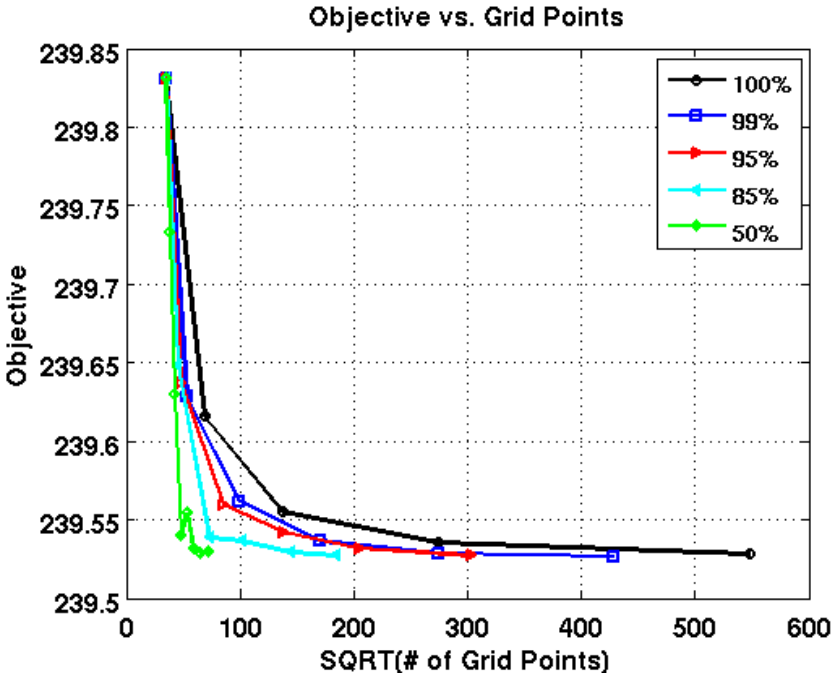

Figure 13. Objective for non-uniform spatial refinement

To examine the efficiency gains from using adaptive spatial refinement we use the same starting discretization as uniform spatial refinement (Section IV.D.1). The grid is held fixed for all time steps of the simulation and at the completion the individual error contributions at each time step are summed within each element to get the total contribution from that element to the error in the integrated objective function. The elements are then ranked by the absolute value of the error that they contribute and those with the largest error are flagged for refinement until a percentage of the total error is accounted for. In Figures 13 and 14 we have plotted the objective values and the error with respect to the exact solution as a function of the number of grid elements when the percentage of error refined on the subsequent sweep is 100, 99, 95, 85, 50 percent.

As in the non-uniform temporal refinement of Section IV.C.2, the higher refinement percentages $(99\%, 95\%)$ correct most of the error that uniform refinement accounts for but requires only 61% of the grid elements to calculate the objective to the same accuracy. As the percentage of refined error is lowered $(85\%, 50\%)$ the efficiency gains can be more significant (5% of isotropic refinement) but a single refinement sweep doesn't always result in a more accurate solution. Again, this loss in predictability is a result of cancellation of the errors based on how the elements are selected for refinement.

The individual grids for each spatial refinement sweep are shown with the initial density distribution in Figures 15 and 16 for 99% and 50% refinement. In both cases the path the vortex travels is heavily targeted for refinement with slightly fewer elements being refined in the region after the vortex convects through the integrated square of the objective function.

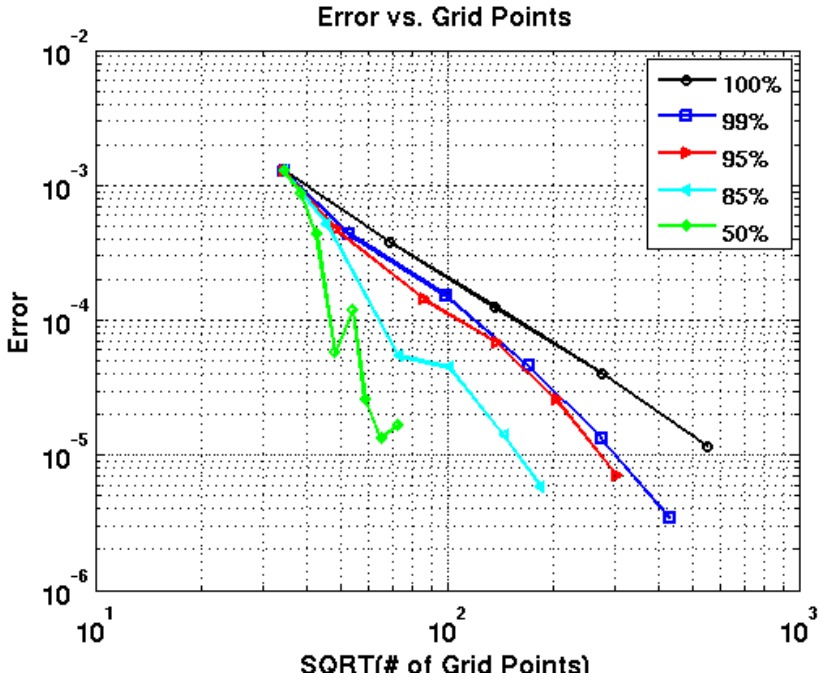

Figure 14. Error in objective for non-uniform spatial refinement

IV.E. Combined Time and Space Adaption

IV.E.1. Uniform Refinement

The combined time and space adaption is performed by using 16 equal interval time steps to march the coarse grid of 1176 elements through to the final time. At each time step the flow and adjoint solution is converged to a tolerance of 10^{-7}. After a complete time series solution at one time and space discretization is calculated the linear predictions about the next refined level are made. Then each time step is split and the grid is isotropically refined and marched through all steps to get the solution on the refined discretization. The objective values for the fine level projected solutions and corrected solutions are shown in Figures 17 and 18 respectively. A new curve has been introduced in the corrected objective plot (Fig. 18) to represent a prediction of the final objective based on the geometric series associated with the 2nd order accurate scheme. In this case, since the error would be expected to be reduced by a factor of 4 at each new refinement level, the total error (with respect to the analytical solution) can be estimated from the error between the two current solution levels as:

$$\epsilon_{Total} = \epsilon_n + \frac{1}{4}\epsilon_n + \frac{1}{16}\epsilon_n + \cdots = \frac{4}{3}\epsilon_n \tag{46}$$

Where ϵ_{Total} is the error with respect to the true analytic solution and ϵ_n is the error with respect to the next refined discretization at any step n. Thus, multiplying the final correction by 4/3 provides an estimate of the total error in the solution as opposed to the relative error between two levels. The error of this final objective value along with the temporal, spatial and combined corrections are plotted in Figure 19.

As with the previous temporal and spatial error results (Sects. IV.C.1 and IV.D.1) we quantify the error in the combined adjoint correction (Eqn. 38), minus the incomplete convergence term, by calculating the error with respect to the next refined solution after the adjoint correction. For the combined relative error we calculate this as:

$$RelativeError = \left| L_{st}(\vec{U}_{cst}) - \left[L_s(\tilde{U}_s) + L_t(\tilde{U}_t) - L(U_c) + \tilde{\Lambda}_s^T R_s(\tilde{U}_s) + \tilde{\Lambda}_t^T R_t(\tilde{U}_t) \right] \right| \tag{47}$$

The relative error is plotted in Figure 20 and the slope for the combined temporal and spatial correction is superconvergent with a rate of 2.6, where as, the theoretical is 4th order.

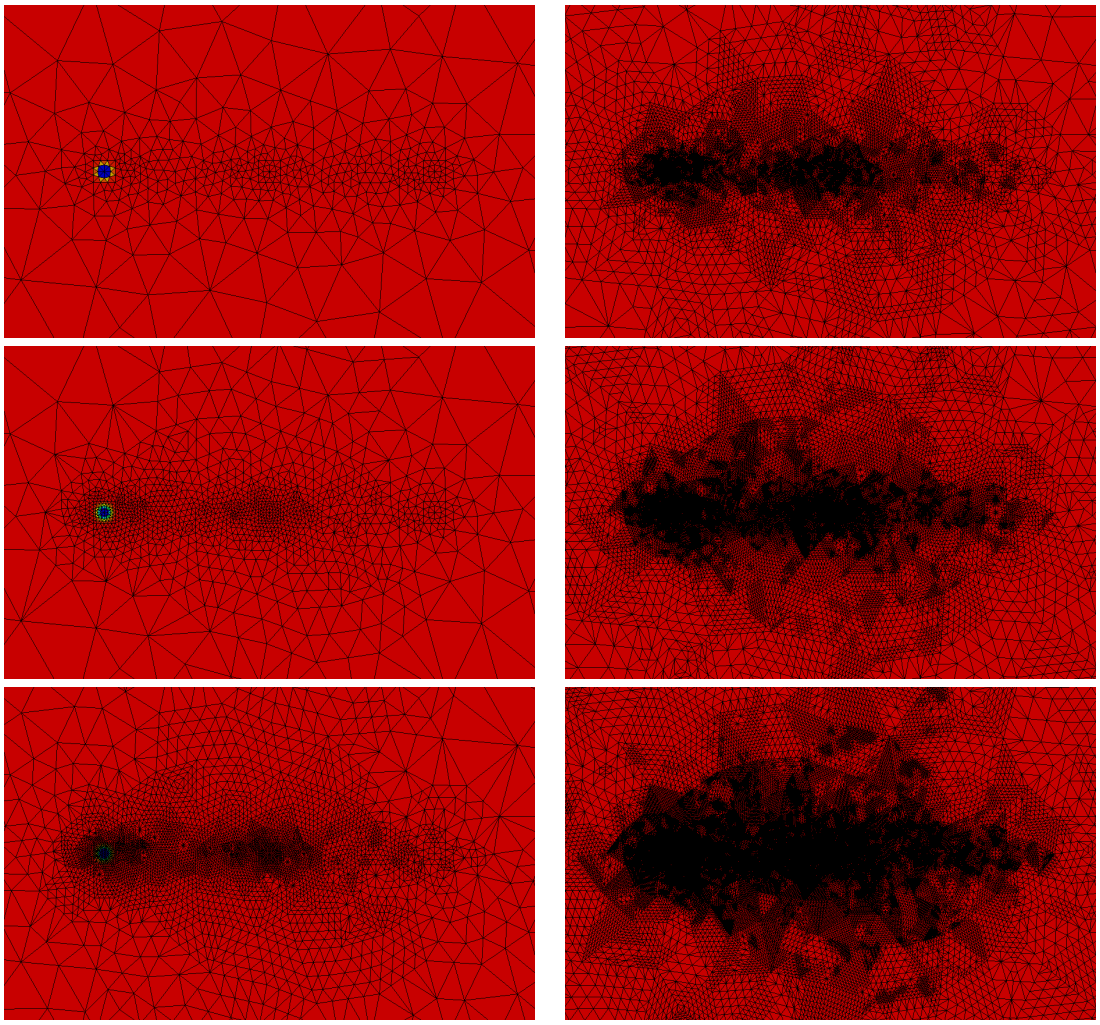

Figure 15. Grid when 99% of the spatial error is refined for each sweep of the simulation. At the top of the left column the image shows the base grid with 1176 elements. Below it is the grid for the 1st and 2nd refinements. The top right image shows the grid after 3 refinement sweeps (29159 elements) and below it are the subsequent grids with a final grid of 183786 elements.

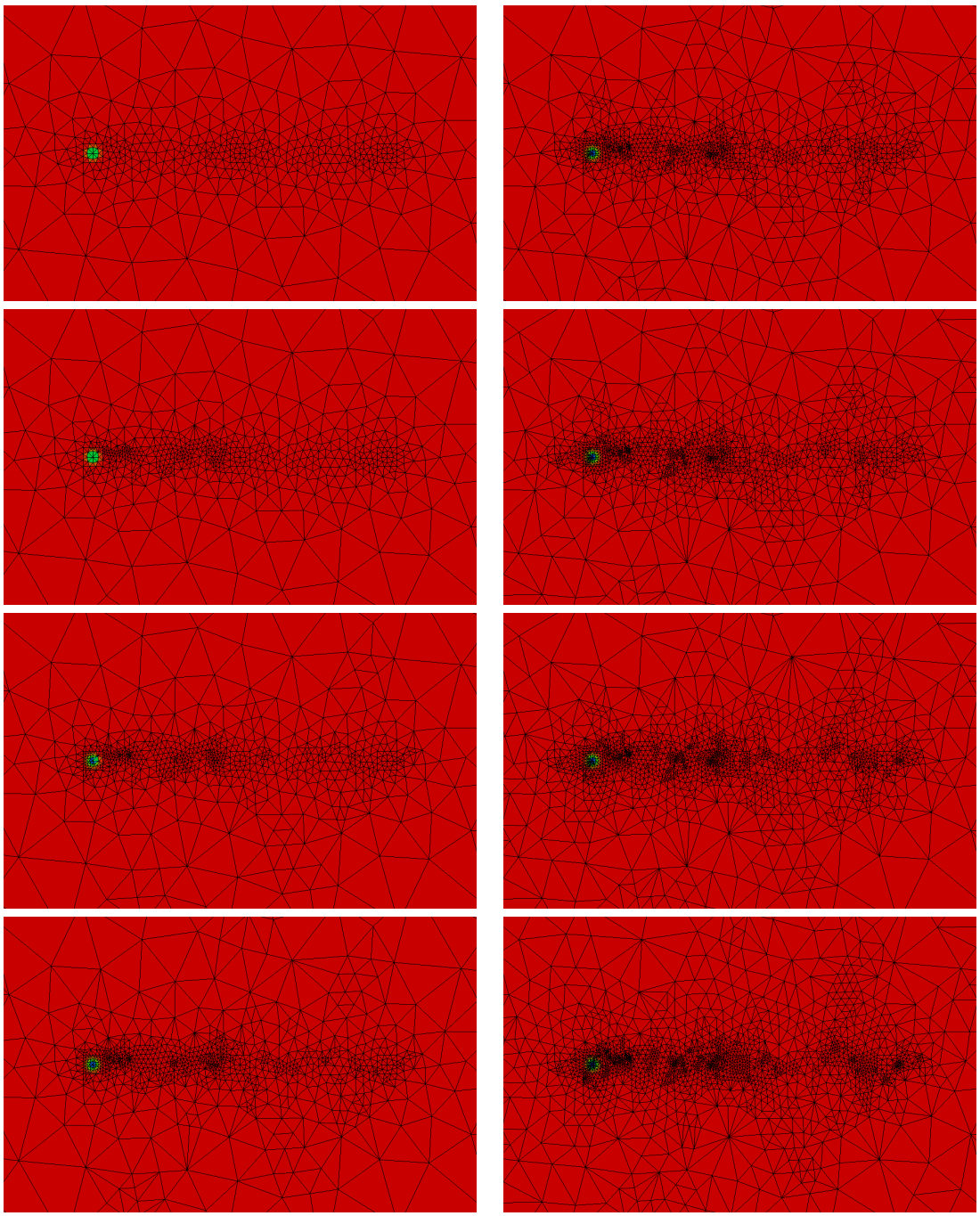

Figure 16. Grid when 50% of the spatial error is refined for each sweep of the simulation. At the top of the left column the image shows the base grid with 1176 elements. Below it is the grid for the 1st through the 3rd refinements. The top right image shows the grid after 4 refinement sweeps (2894 elements) and below it are the subsequent grids with the final grid of 5258 elements in the bottom right.

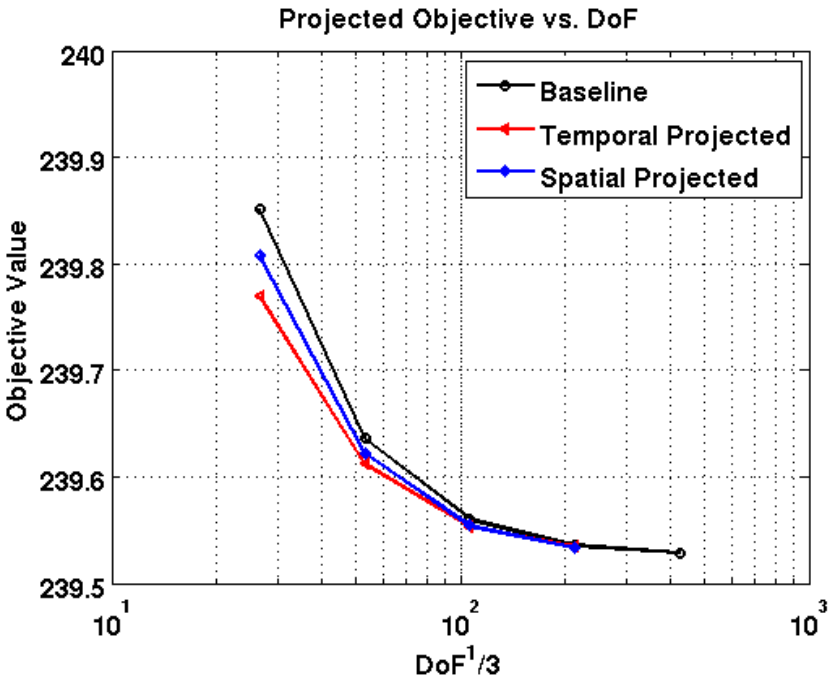

Figure 17. Projected objectives for uniform temporal and spatial refinement

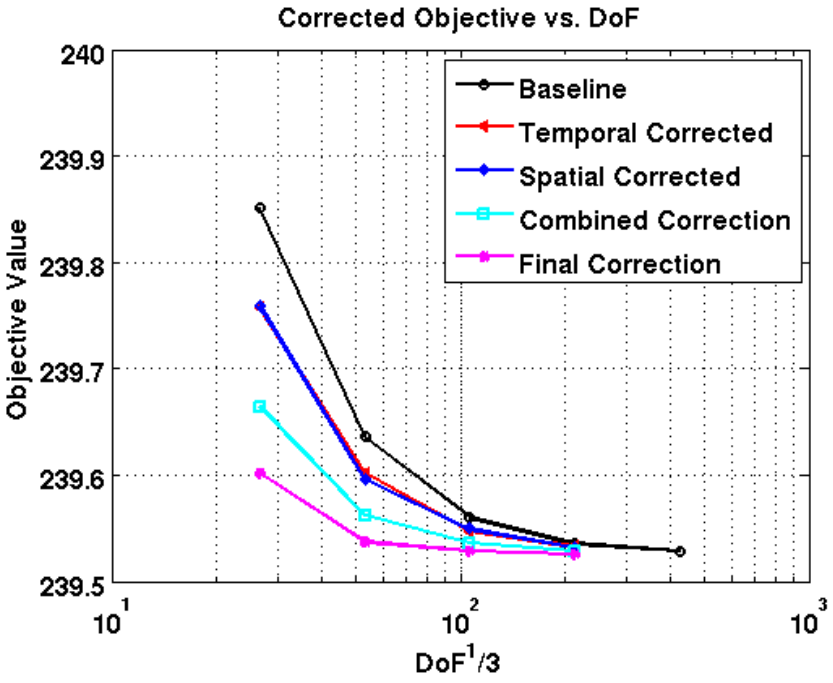

Figure 18. Corrected objectives for uniform temporal and spatial refinement

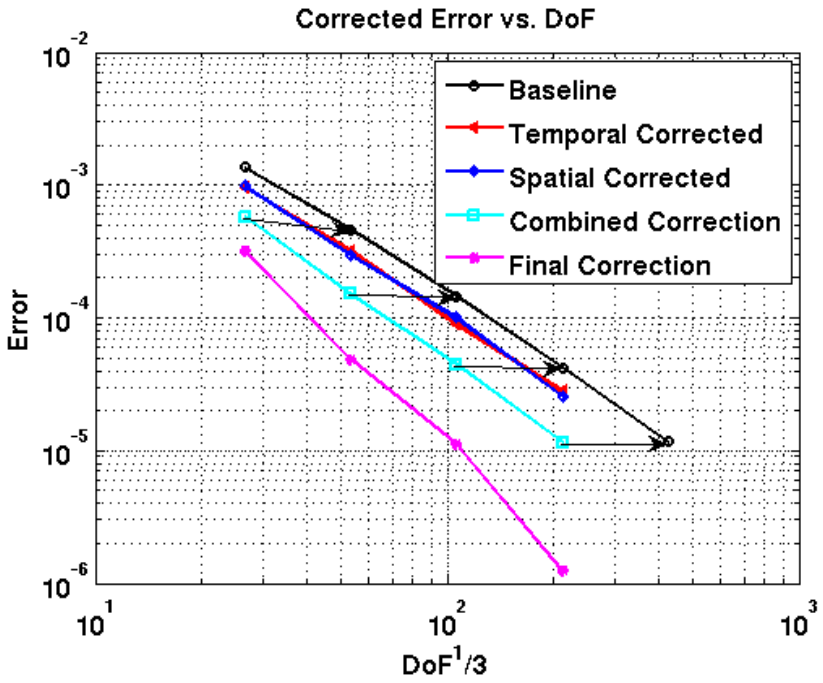

Figure 19. Corrected objective error for uniform temporal and spatial refinement

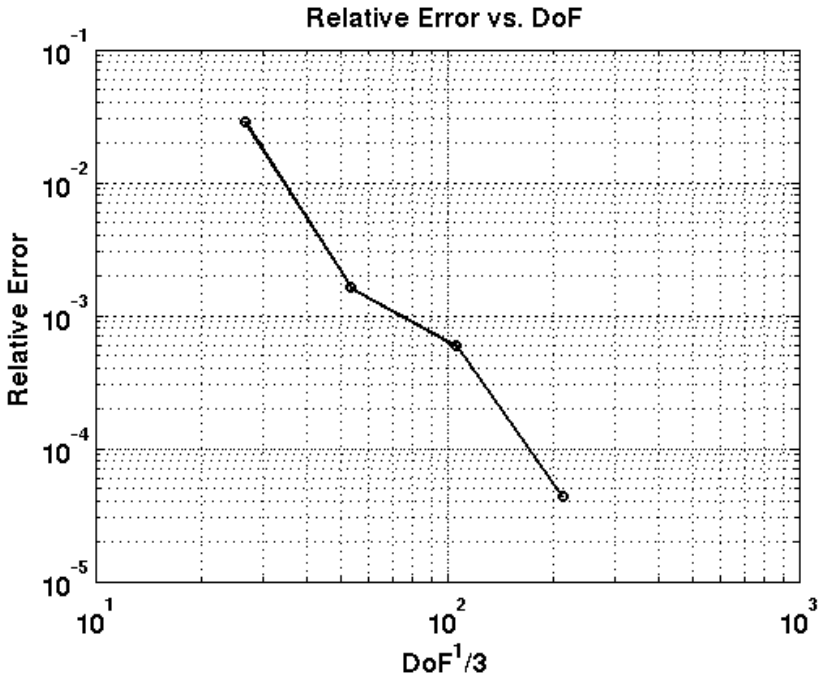

Figure 20. Relative error for uniform temporal and spatial refinement

IV.E.2. Non-Uniform Refinement

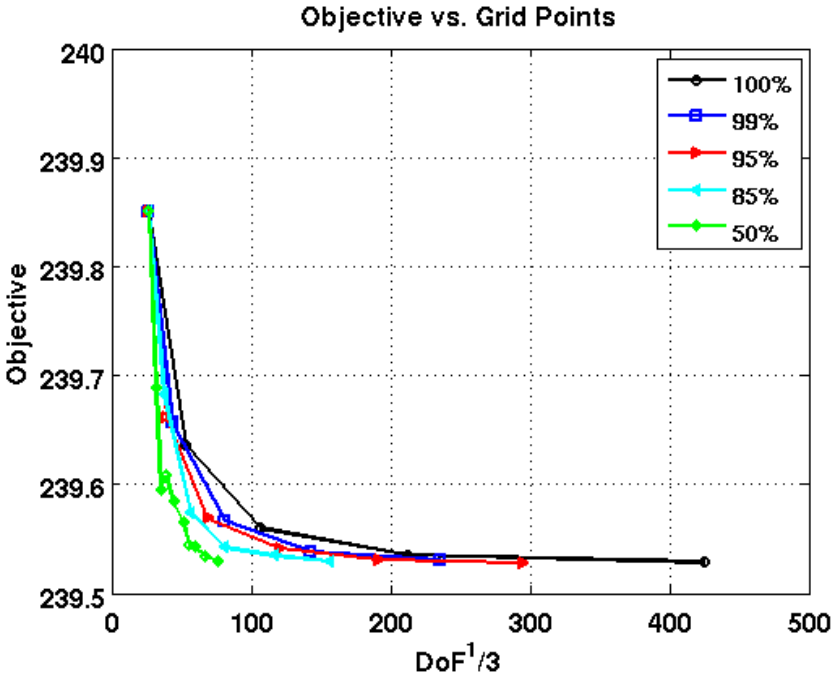

Figure 21. Objective for non-uniform temporal and spatial refinement

To examine the efficiency gains from simultaneously adapting time and space we use the same starting discretization as uniform time and space refinement (Section IV.E.1). The grid is held fixed for all time steps of the simulation and at the completion the temporal error contributions for each time step are calculated along with the spatial error for each element at each time step. The spatial element error values are summed as in section IV.D.2 to get the total contribution from each element to the error in the integrated objective function. The time steps and elements are then separately ranked by the absolute value of the error that they contribute and those with the largest error are flagged for refinement until a percentage of the total error is accounted for. In Figures 21 and 22 the objective values and the error with respect to the exact solution are plotted as a function of the flow solver degrees of freedom ($TimeSteps \times Elements$) when the percentage of error refined on the subsequent sweep is 100, 99, 95, 85, 50 percent.

Again, the higher refinement percentages $(99\%, 95\%)$ correct most of the error that uniform refinement accounts for but requires far less computational effort. In the case of refining 99% of the error at the completion of each sweep the final simulation takes only 17% of the degrees of freedom of the uniform refinement to reach the same error tolerance. As the percentage of refined error is lowered $(85\%, 50\%)$ the efficiency gains can be substantial. In the case of 50% refinement, the final simulation required only 0.58% of the degrees of freedom of the uniform refinement and still achieves roughly the same error. It is important to note, the individual markers on the plots represent a full time series flow and adjoint solution. To achieve the same or lower error the low percentage refinements require more simulations but all the computations are done on a much smaller discretization. A more equal comparison is the total degrees of freedom for all the flow solver sweeps to arrive at the final answer. In this case the uniform (100%) refinement has 88,077,696 total degrees of freedom for the five simulations to reach an error less than 2×10^{-5} while the (50%) refinement takes 1,514,990 total degrees of freedom through 9 refinements to reach the same tolerance. Using this more equal comparison we still find the 50% refinement results in a 98% cost savings in the total degrees of freedom solved over the uniform refinement.

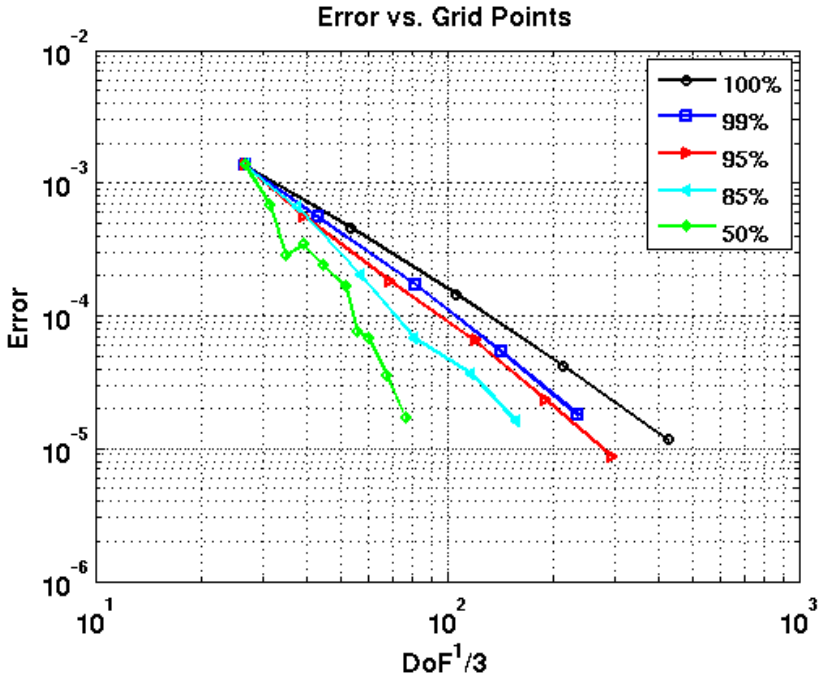

Figure 22. Error in objective for non-uniform temporal and spatial refinement

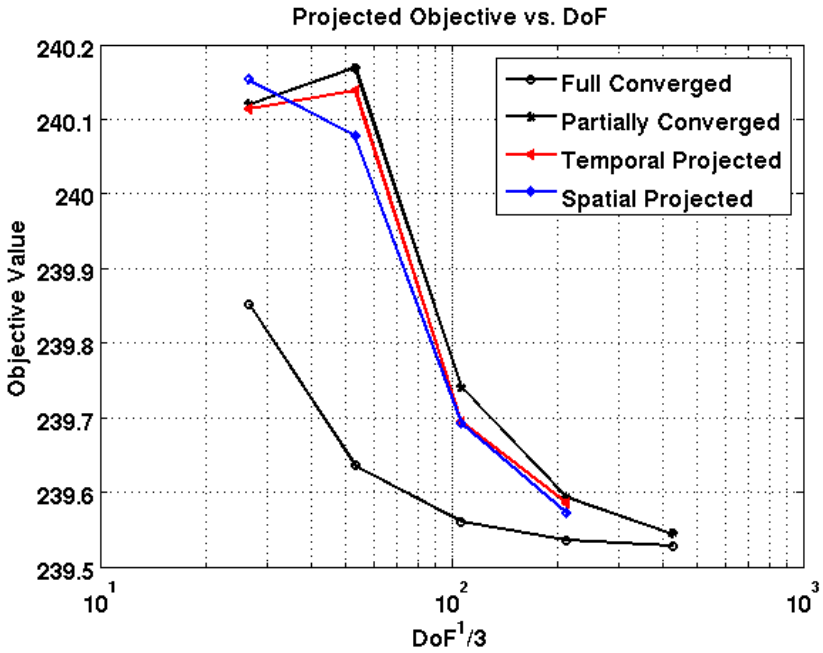

Figure 23. Projected objectives for uniform temporal and spatial refinement with incomplete convergence correction

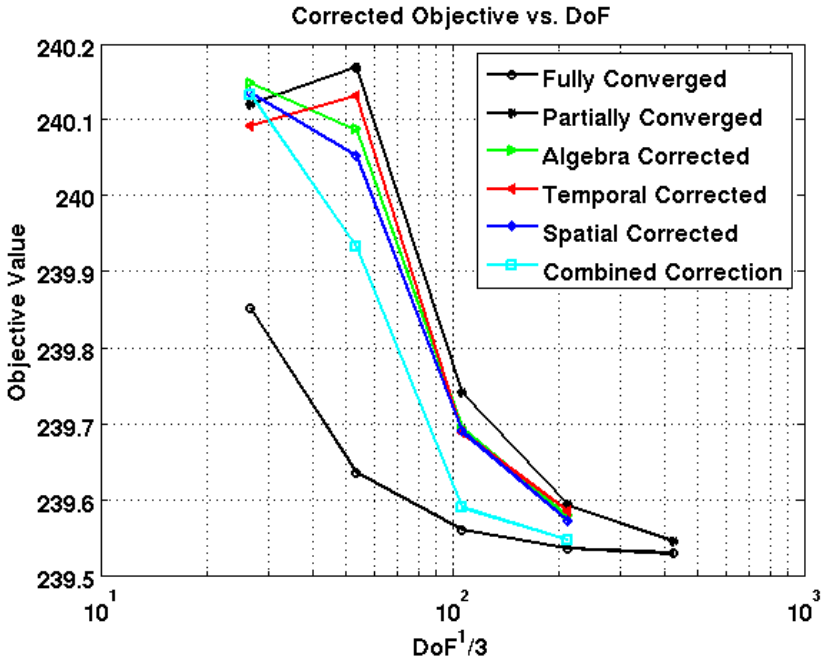

Figure 24. Corrected objectives for uniform temporal and spatial refinement with incomplete convergence correction

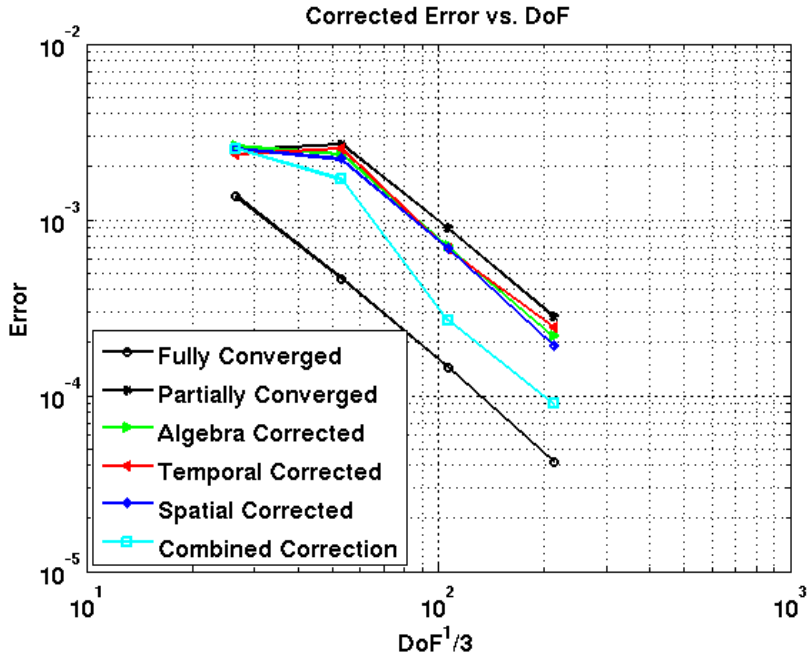

Figure 25. Corrected objective error uniform temporal and spatial refinement with incomplete convergence correction

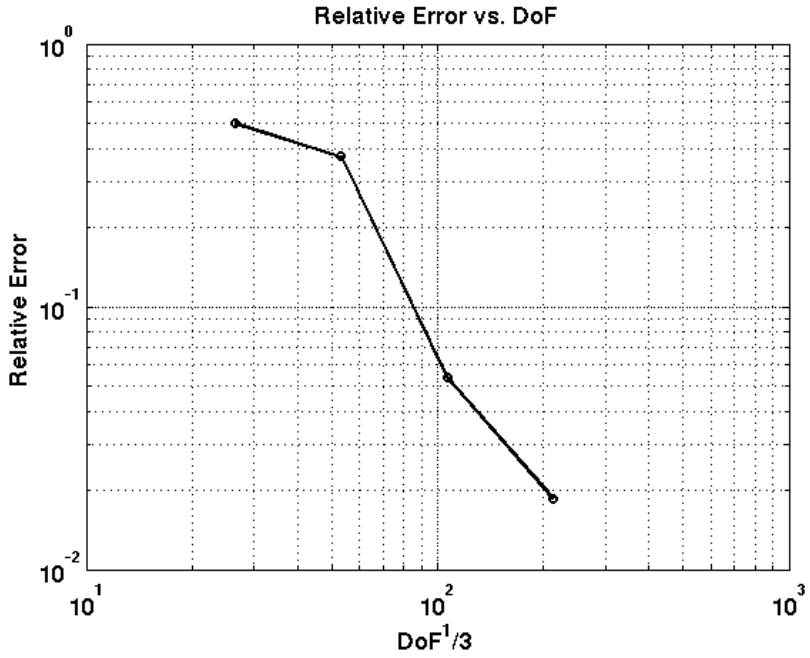

Figure 26. Relative error for uniform temporal and spatial refinement with incomplete convergence correction

IV.F. Combined Convergence, Time and Space Adaption

IV.F.1. Uniform Refinement

To test the combined algebraic, temporal and spatial error correction the calculation is started by using 16 equal time steps on the coarse grid of 1176 elements. For each time step of the calculation the L_2 norm of the residual is converged 1 order of magnitude so an error from incomplete convergence still exists according to Figure 3. At the completion of the required time steps the total correction from the three types of error is combined following section II.F to give a prediction of the next uniformly refined, in both time and space, simulation. The convergence tolerance is not changed from one simulation to the next, therefore, the correction will be used to predict what the answer should be if each step was fully converged.

The objective of the partially and fully converged solutions have been plotted along with the temporal ($L_t^n(\tilde{U}_t^n)$) and spatial ($L_s^n(\tilde{U}_s^n)$) projected objectives for comparison in Figure 23. In the following figures the error corrections (Eqns. 31,28,24) and the combined correction (Eqn. 38) are applied to the projections and the resulting objective (Fig. 24) and error in the objective (Fig. 25) are plotted. The plots show the error due to incomplete convergence is relatively large. Ideally, we would like this to be a smaller contribution since our analysis in Section IV.B showed only a 1 order of magnitude improvement by using the error correction when the adjoint is not fully converged. Even with this large source of error, the combined correction does a better job of predicting the next uniformly refined simulation than any of the individual corrections alone. All corrections result in convergence rates just under 2nd order with the combined correction showing a slope of 1.6 for the last segment. The relative error for the combined correction is plotted in Figure 26 where the slope of the last segments is measured at 2.2, which is below the theoretical superconvergent rate of 4, but still good considering the convergence rate results from a linear approximation to three different types of error.

IV.F.2. Non-Uniform Refinement

The final test case in this paper will examine the efficiency gains from using both temporal and spatial adaptive refinement while partially converging the flow and adjoint solution at each time step.. The starting discretization is the same as the combined uniform refinement of section IV.E.1, 16 equal time steps on the coarse grid of 1176 elements. The grid is held fixed for all time steps of the simulation and at the completion the temporal error contributions for each time step are calculated along with the spatial error for

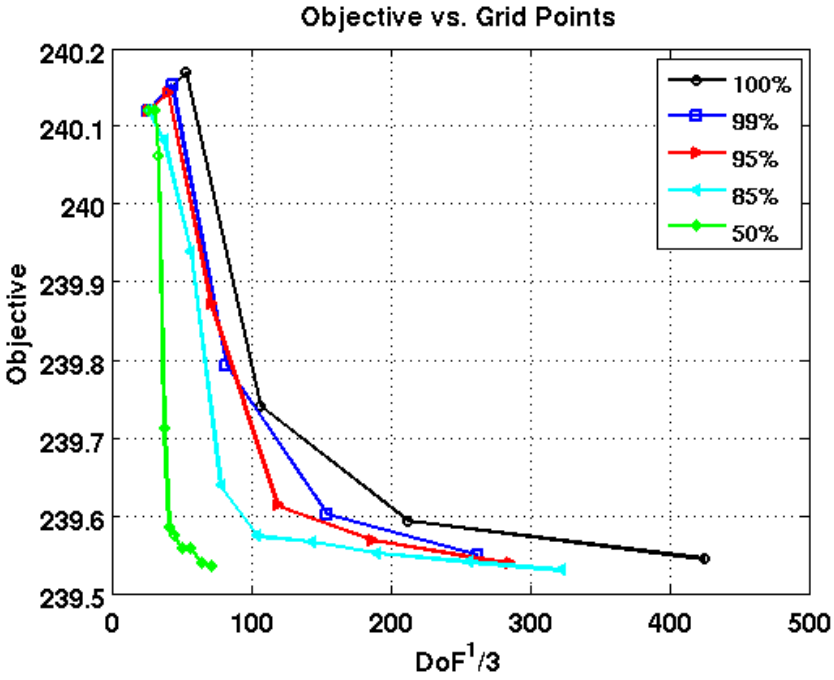

Figure 27. Objective for non-uniform temporal and spatial refinement with incomplete convergence correction

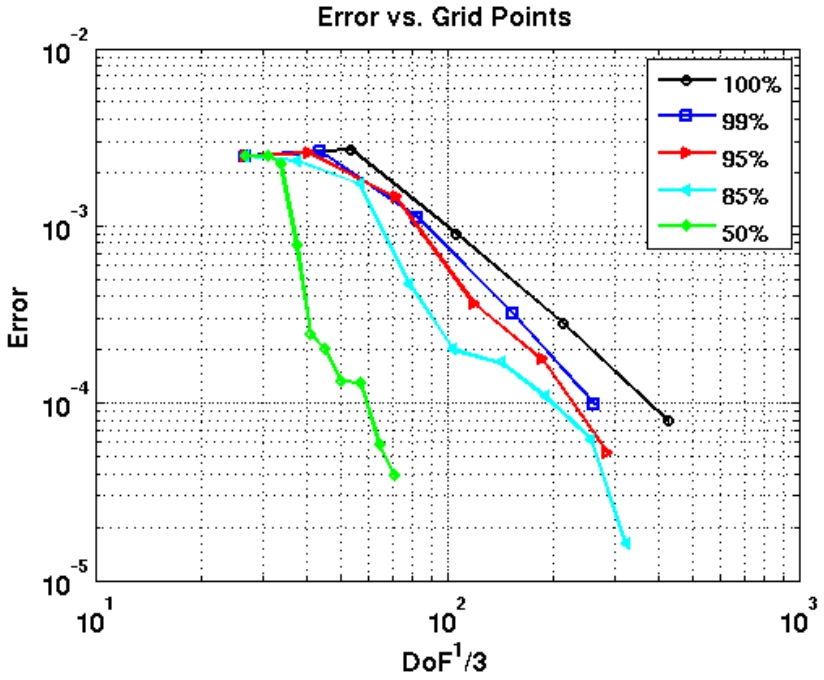

Figure 28. Error in objective for non-uniform temporal and spatial refinement with incomplete convergence correction

each element at each time step. The spatial element error values are summed as in section IV.D.2 to get the total contribution from each element to the error in the integrated objective function. The time steps and elements are then separately ranked by the absolute value of the error that they contribute and those with the largest error are flagged for refinement until a percentage of the total error is accounted for. In Figures 27 and 28 the objective values and the error with respect to the exact solution are plotted as a function of the flow solver degrees of freedom ($TimeSteps \times Elements$) when the percentage of error refined on the subsequent sweep is 100, 99, 95, 85, 50 percent.

Again, the higher refinement percentages ($99\%, 95\%$) correct most of the error that uniform refinement accounts for but requires far less computational effort. In the case of refining 99% of the error at the completion of each sweep the final simulation takes only 23% of the degrees of freedom of the uniform refinement to reach the same error tolerance. As the percentage of refined error is lowered ($85\%, 50\%$) the efficiency gains can be substantial. In the case of 50% refinement, the final simulation required only 0.4% of the degrees of freedom of the uniform refinement and still achieves a lower error. Using the more equal comparison of the total degrees of freedom for all the flow solver sweeps as in Section IV.E.2 the 50% refinement results in a 99% cost savings in the total degrees of freedom solved over the uniform refinement.

V. Conclusions

A method for the correction of algebraic, temporal, spatial and a combined total error in unsteady flow problems was developed and tested. This information was used to replace the ambiguity and guess work typically associated with selecting a convergence tolerance, time step size and grid discretization. Significant gains in efficiency were shown while refining a fixed amount of error in only one dimension (time, space or both) at a time. One obvious extension of this work is to develop a refinement methodology that selects a fraction of the error in each dimension independent of the other which most efficiently reduces the error in the objective. A second extension, is to allow non-uniform grid and convergence tolerance adaption to occur for the individual time steps. This will increase the efficiency by further limiting the calculations to specific regions and enable the tracking of flow features of interest.

References

[1]Venditti, D. A. and Darmofal, D. L., "Grid adaptation for functional outputs: application to two-dimensional inviscid flows," *Journal of Computational Physics*, Vol. 176, No. 1, 2002, pp. 40–69.

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

[3]Nemec, M. and Aftosmis, M., "Adjoint error estimation and adaptive refinement for embedded-boundary cartesian meshes," No. 2007-4187, AIAA Paper, 2007.

[4]Houston, P., Rannacher, R., and Suli, E., "A posteriori error analysis for stabilised finite element approximations of transport problems," *Computer Methods in Applied Mechanics and Engineering*, Vol. 190, No. 11-12, 2000, pp. 1483–1508.

[5]Becker, R. and Rannacher, R., "An optimal control approach to a posteriori error estimation in finite element methods," *Acta Numerica*, , No. 10, 2001, pp. 1–102.

[6]Giles, M. B. and Suli, E., "Adjoint methods for PDEs: a posteriori error analysis and postprocessing by duality," *Acta Numerica*, , No. 11, 2002, pp. 145–236.

[7]Mani, K. and Mavriplis, D. J., "Discrete adjoint based time-step adaptation and error reduction in unsteady flow problems," No. 2007-3944, AIAA Paper, 2007.

[8]Mani, K. and Mavriplis, D. J., "Error estimation and adaptation for functional outputs in time-dependent flow problems," No. 2009-1495, AIAA Paper, 2009.

[9]Luo, Y. and Fidkowski, K. J., "Output-based space-time mesh adaption for unsteady aerodynamics," No. 2011-491, AIAA Paper, 2011.

[10]Li, S. and Petzold, L., "Adjoint sensitivity analysis for time-dependent partial differential equations with adaptive mesh refinement," *Journal of Computational Physics*, Vol. 198, No. 1, 2004, pp. 310–325.

[11]Roe, P., "Approximate Riemann solvers, parameter vectors, and difference schemes," *Journal of Computational Physics*, Vol. 43, No. 1, 1981, pp. 357–372.

[12]Mavriplis, D. J., "Adaptive meshing techniques for viscous flow calculations on mixed element unstructured meshes," *International Journal for Numerical Methods in Fluids*, Vol. 34, No. 2, 2000, pp. 93–111.

[13]Buhmann, M. D., *Radial Basis Functions*, Cambridge University Press, 2004.

[14]Wendland, H., "Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree," *Advances in Computational Mathematics*, Vol. 4, No. 1, 1995, pp. 389–396.

[15]Bonet, J. and Peraire, J., "An alternating digital tree (ADT) algorithm for 3D geometric searching and intersection problems," *International Journal for Numerical Methods in Engineering*, Vol. 31, No. 1, 1991, pp. 1–17.

[16]Davoudzadeh, F., Mcdonald, H., and Thompson, B., "Accuracy evaluation of unsteady cfd numerical schemes by vortex preservation," *Computer in FLuids*, Vol. 24, No. 1, 1995, pp. 1995.

[17]Yee, H., Sandham, N., and Djomehri, M., "Low dissipative high order shock-capturing methods using characteristic-based filters," *Journal of Computational Physics*, Vol. 150, No. 1, 1999, pp. 199–238.

[18]Whitfield, D. and Janus, J., "Three-Dimensional Unsteady Euler Equations Solution Using Flux Vector Splitting," No. 84-1552, AIAA Paper, 1984.