AIAA Aviation
16-20 June 2014, Atlanta, GA
32nd AIAA Applied Aerodynamics Conference

Feature-Based Grid Adaption for the Study of Dynamic Stall

Kyle Hord* and Yongsheng Lian†

University of Louisville, Louisville, Kentucky 40292

In this paper we study dynamic stall phenomenon of a pitching NACA0015 airfoil using feature based grid adaption technique. The fluid solver is based on FUN3D which solves the three-dimensional, compressible, unsteady Reynolds-Averaged Naiver-Stokes equations. The one equation Spalart-Allmaras is used a the turbulence closure. The governing equations are discretized spatially using second-order finite volume methods and temporally using an optimized second order backward difference scheme. The grid adaption is based the anisotropic tetrahedral adaptation approach in which grid is adapted to match a desired quality via an anisotropic metric calculated throughout the simulation. Particularly we track vorticity throughout the pitching cycle and adapt the grid in areas where vorticity is damped. Comparisons are also made with results using uniformly refined grids. Results suggest feature-based adaptation has potential in refining the mesh in the wake of the airfoil, allowing vorticity to be carried out several chords behind the airfoil without excessive dissipation. Our study also shows that great care must be placed in allowing the grid to be adapted in the vicinity of the airfoil as grid resolution can be lost.

I. Introduction

Research has shown that grid quality can dramatically affect simulation results. Kravchenko[1] showed that incorrect flow features were produced in the wake of a cylinder due to large grid spacing. As a result, the separated flow region was shorter and the recirculation region immediately behind the cylinder became smaller, resulting in incorrect flow features and poor force prediction. Similarly, in a review paper by Komerath,[2] it was noted that the wake grid resolution behind helicopter blades can impact the accuracy of the flow simulation. In the study of dynamic stall, the grid quality also plays a critical role. Dynamic stall

*PhD Candidate, Mechanical Engineering Department, 200 Sackett Hall.
†Assistant Professor, Mechanical Engineering Department, 200 Sackett Hall. Senior Member AIAA.

is a complex fluid mechanics problem plagued with vortex formation and shedding. To accurately predict the force coefficients, it is critical to capture the vortex formation and its propagation in the flow. However, without a priori knowledge of the vortex structures and their trajectories, creating a computationally efficient grid to capture these features is challenging.

Feature-based grid adaption is a promising technique to automatically adapt grid at regions of interests. It can reduce the manual interaction with the CFD tools and the required expertise of the user to obtain accurate solutions.[3] It can also significantly alleviate the need for spatial convergence verification of computational results by balancing the modeling accuracy and computational efficiency for engineering analysis. Kang et al.[4] and Park et al.[5] have successively implemented feature-based grid adaptation schemes to resolve vortex flow structures produced by a rotor-craft in hovering flight. Their studies showed that feature-based adaptation methods are able to improve the prediction of lift and drag coefficients by refining grid where vortices propagated through and interacted with vehicle structures. By improving spatial accuracy of the vortex structures, they showed that feature-based grid adaption can better model vortex interactions which are critical in hovering flight.

To fulfill the potential of feature-based grid adaption method, it is critical to calibrate the initial grid in order to simulate flow features correctly.[6,7] Without proper evaluation of the initial grid, the adapt mesh can capture incorrect features far from their true locations. Further adaption can "force" features into a refined area and hence steadily enhance an erroneous solution with each grid adaption. It was hypothesized that this erroneous behavior was the result of an insufficient grid resolution of the initial uniform mesh on which the feature-based grid adaption was based.[3,6]

If properly addressed, such issues should not inhibit the application of mesh adaptation to problems in flow fields.[8] In order to confirm the effectiveness and viability of feature-based adaptation for dynamic stall flows, a study will be conducted on the flow over a 2D pitching NACA0015 airfoil. In particular, the study will compare the results of an uniformly refined grid to that of a feature adapted grid. To learn from previous studies, an initial spatial convergence study will be conducted to evaluate the effectiveness of uniformly refining a grid. This will be determined by investigating the change in coefficients of lift and drag. Next, one uniformly refined grid will be used as the initial grid for the adaptation study. We investigate if the mesh can be improved to better capture the wake structure of the pitching airfoil.

II. Numerical Methods

FUN3D[a] is an unstructured finite-volume solver that has been developed and supported by the NASA Langley Research Center.[9,10] FUN3D solves the three-dimensional, unsteady, compressible Reynolds-Averaged

[a] http://fun3d.larc.nasa.gov/

Naiver-Stokes (RANS) equations in integral form:

$$\frac{\partial}{\partial t}\int_V qdV + \int_{\partial V}(F^* - F_V)\cdot\hat{n}dS = 0 \tag{1}$$

where V is the control volume which is bounded by δV. F^* and F_v are the inviscid and viscous fluxes of conservative variables. q represents the conserved variable vector

$$q = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ E \end{bmatrix} \tag{2}$$

For the case of a moving control volume, the convective fluxes must take in account the relative velocity of the control volume. The inviscid flux is adjusted for the control volume face speed $|W|$ such that:

$$F^* = F - qW \tag{3}$$

where F is a stationary control volume flex vector. The volume average (Q) of q must be also taken to account for the moving volume:

$$Q = \frac{\int_V qdV}{V} \tag{4}$$

and the conservation equations become:

$$\frac{\partial(QV)}{\partial t} + \oint_{\partial V}(F^* - F_v)\cdot\hat{n}dS = 0 \tag{5}$$

The convective inviscid and viscous fluxes are then defined as follows where F^* now accounts for the relative motion of the control volume.

$$F^* = \begin{bmatrix} \rho(u-W_x) \\ \rho u(u-W_x)+p \\ \rho v(u-W_x) \\ \rho w(u-W_x) \\ (E+p)(u-W_x)+W_x p \end{bmatrix}\hat{i} + \begin{bmatrix} \rho(v-W_y) \\ \rho u(v-W_y)+p \\ \rho v(v-W_y) \\ \rho w(v-W_y) \\ (E+p)(v-W_y)+W_y p \end{bmatrix}\hat{j} + \begin{bmatrix} \rho(w-W_z) \\ \rho u(w-W_z)+p \\ \rho v(w-W_z) \\ \rho w(w-W_z) \\ (E+p)(w-W_z)+W_y p \end{bmatrix}\hat{k} \tag{6}$$

$$F_v = \begin{bmatrix} 0 \\ \tau_{xx} \\ \tau_{yx} \\ \tau_{zx} \\ u\tau_{xx}+v\tau_{xy}+w\tau_{xz}-q_x \end{bmatrix}\hat{i} + \begin{bmatrix} 0 \\ \tau_{xy} \\ \tau_{yy}) \\ \tau_{zy} \\ u\tau_{yx}+v\tau_{yy}+w\tau_{yz}-q_y \end{bmatrix}\hat{j} + \begin{bmatrix} 0 \\ \tau_{xz} \\ \tau_{yz} \\ \tau_{zz} \\ u\tau_{zx}+v\tau_{zy}+w\tau_{zz}-q_z \end{bmatrix}\hat{k} \tag{7}$$

Within FUN3D, the RANS equations are discretized using flux splitting scheme where the inviscid and vis-

cous fluxes are handled separately. For second order accuracy, the values at the cell interfaces are calculated using gradients at the mesh nodes and computed using a least-squares technique. Currently, feature-based grid adaptation is only implemented on fully tetrahedral based computational domains. For tetrahedral meshes, the full viscous fluxes are discretized using a finite-volume formulation in which the required velocity gradients at the cell faces are computing using the Green-Guass theorem, which is equivalent to a Galerkin approximation.

II.A. Temporal Discretization

Time-accuracy is achieved through an optimized second order backward difference scheme (BDF2OPT). It is used to construct a higher order temporal scheme by extending the difference stencil in time.[11] The details of BDF2OPT and the temporal error control schemes are well documented by Biedron[9] and will be briefly discussed here. To begin the derivation of the equations used for time stepping, it is assumed that the control volume is invariant in time (the grid is rigid) which is written as

$$V\frac{\partial Q}{\partial t} = R \tag{8}$$

Expanding on Eq. (8) to evaluate the next time level of $n+1$, and writing the time derived as a series of backward difference levels gives

$$\frac{V}{\Delta t}\left(\phi_{n+1}Q^{n+1} + \phi_n Q^n + \phi_{n-1}Q^{n-1} + \phi_{n-2}Q^{n-2} + \dots\right) = R^{n+1} \tag{9}$$

The series of ϕ_n governs the accuracy of the backwards difference of the temporal discretization. The respective values of ϕ_n varies depending the order of accuracy of the scheme, but must satisfy the requirement of $\sum \phi_n = 0$. As stated before, FUN3D uses an optimized backwards difference scheme, which is a linear combination of second and third order coefficients. With an order of accuracy that is in between second and third order, the temporal equations are stable with any time-step. The coefficients for BDF2OPT are listed in Table 1.

order	ϕ_{n+1}	ϕ_n	ϕ_{n-1}	ϕ_{n-2}
2^{nd}	3/2	-2	1/2	0
3^{rd}	11/16	-3	3/2	-1/3
$BDF2_{opt}$	$3/2 - \phi_{n-2}$	$-2 + 3\phi_{n-2}$	$1/2 - 3\phi_{n-2}$	-0.58/3

Table 1: Coefficients for backwards difference schemes[9]

A pseudo-time term (τ) is introduced into Eq. (9), as in Ref.,[12] to minimize the error due to the linearization about the time level of $n+1$. If δQ is linearized about τ and assumed to be equal to $\delta Q^m = Q^{m+1} - Q^m$, where m is the current pseudo-time level, then the final form of the temporal equation can be written as

$$\left[\left(\frac{V}{\Delta\tau}+\frac{V\phi_{n+1}}{\Delta t}\right)I-\frac{\partial R^m}{\partial Q}\right]\Delta Q^m = R^m - \frac{V}{\Delta t}\left(\phi_{n+1}\left(Q^m-Q^n\right)+\phi_{n-1}\left(Q^{n-1}-Q^n\right)+\phi_{n-2}\left(Q^{n-2}-Q^n\right)+\ldots\right) \quad (10)$$

At each sub-iteration of m, the linear system in Eq. (10)is iteratively solved using a user-specified number of point Jacobi or Gauss-Seidel sweeps. When between time-steps, the equations are advanced in pseudo time with local time stepping to accelerate the solution to a steady state pseudo time. In order to accelerate convergence in pseudo time, the CFL number can be ramped during the sub-iterations.

A temporal error control method[9] has been used to reduce the number of sub-iterative loops by exiting the dual time stepping process when a specified criteria has been met. This helps address the issue of the choice of the number of sub-iterations to use when solving the flow equations. The temporal error is calculated by examining the difference in residual contribution of two different levels of time derivative approximations. The sub-iteration loop will be terminated when the residuals drop below a specific fraction of the temporal error norm.

II.B. Feature-Based Adaptation

FUN3D's feature-based adaptation is a node based approach that uses anisotropic tetrahedral adaptation. The mesh resolution is modified to match a desired quality via a anisotropic metric calculated throughout the simulation.[3,13,14] The approach in this paper focuses on using the vorticity magnitude at each node to determine the local and global adaptation metric. First, the vorticity magnitude is used to form an adaptation key, $K_{l,\omega}$. This key is the vorticity magnitude delta across an edge connecting two adjacent nodes n_1 and n_2, which is then scaled by the edge length, l_e. Scaling $K_{l,\omega}$ by the edge length acts as a filter, emphasizing large changes in vorticity across long edges, while reducing the change across smaller edges. This assists in the adaptation of large cells first.

$$K_{l,\omega} = l_e \frac{|\omega_{n1}| - |\omega_{n2}|}{2} \quad (11)$$

The local error adaptation intensity, $\hat{\text{I}}$, is then computed a for each node in the domain. This is done by finding the maximum scaled key at each node where the adaptation key is normalized by a user specified tolerance,K_t, thus:

$$\hat{I} = \max\left(\frac{K_{l,\omega}}{K_t}\right)_{edge} \quad (12)$$

The selected $\hat{\text{I}}$ is the edge with the greatest scaled vorticity delta connected to the node Next, the new

isotropic mesh size, h_1, is calculated using an estimate of the spacing from the original mesh, h_0, and a coarsening factor C. In this study $C = 1.1$.

$$h_1 = h_0 \min\left(C, \left(\frac{1}{\hat{I}}\right)^{0.2}\right) \tag{13}$$

Finally, the anisotropic adaptation metric is derived using the scalar isotropic mesh size (Eq. (13)) and a vorticity-magnitude Hessian not discussed here. The derivation of the anisotropic metric is detailed thoroughly in references.[15,16] During the computation of the pitching motion, the metric is intersected in time to track the areas that require adaptation. This allows areas vorticity propagates through to be marked for adaptation. The metric is then passed to adaptation modules currently implemented in FUN3D to modify the mesh. The adaptation process provides node insertion and deletion, edge swapping, and node movement to achieve the the desired mesh density and quality necessary to match the derived metrics. The mesh adaptation process has been described in several papers.[3,16,17]

The adaptation modules implemented in FUN3D haven been shown to work well for both static inviscid[3] and viscous flows;[18] however, viscous adaptation lacks the capability to modify the the surface geometry. The mesh can only be modified outside of the cells that define the boundary layer across the surface geometry. Thus it is up to the user to ensure the surface geometry is adequately defined to capture the near wall boundary layer of the geometry. The practitioner can control how much of the boundary layer mesh is adapted by freezing the mesh a specified distance away from the surface. This capability has also been used in Ref.,[18] and will be used in this study.

III. Richardson Extrapolation

Preforming sensitivity studies on the spatial schemes of a numerical simulation can sometimes be prohibitively computational expensive to conduct. Numerical data is normally compared to experimental data to validate the computed solution. However, experimental data is not always available, thus alternate routes must be pursed to ensure that the simulations are converging onto a single answer. One possible method is to use Richardson Extrapolation[19] to estimate the fully converged answer across two or three successively refined grids by a constant factor. The extrapolation technique is used to estimate the order of the solution, and project out a fully converged solution. This can help the practitioner estimate the error associated with the simulation and determine whether further grid refinement is required.

The method to estimate the converge solution is straight forward and can be done two different ways. The first, if the discretization order of solution is known, only two grids with some refinement ratio r (for this study: 1.5) is required. The converged solution can be found with

$$f_{h\approx 0} \cong f_1 + \frac{(f_1 - f_2)}{(r^p - 1)} \tag{14}$$

Where $f_{h\approx 0}$ is the exact solution for an infinitely refined grid or time step, p is the formal order of accuracy of the spatial discretization, and f_1 and f_2 are the flow variables of interest on the fine and coarse grid respectively. In the event that p is not known, it can be estimated with a three refined grids that are refined at a constant refinement ratio (r) value. To estimate p

$$p = \frac{\ln\left(\frac{f_3 - f_2}{f_2 - f_1}\right)}{\ln(r)} \tag{15}$$

where f_1 to f_3 are the flow variables of interest on the finest to coarsest grid. Once p is known, Eq. 14 can be used. To estimate the error associated with the fine grid, Eq. 14 can be rewritten as

$$E_{fine} = \frac{(f_1 - f_2)}{(r^p - 1)} \tag{16}$$

And the error for a coarse grid(s) can be found with:

$$E_{coarse} = \frac{r^p(f_1 - f_2)}{(r^p - 1)} \tag{17}$$

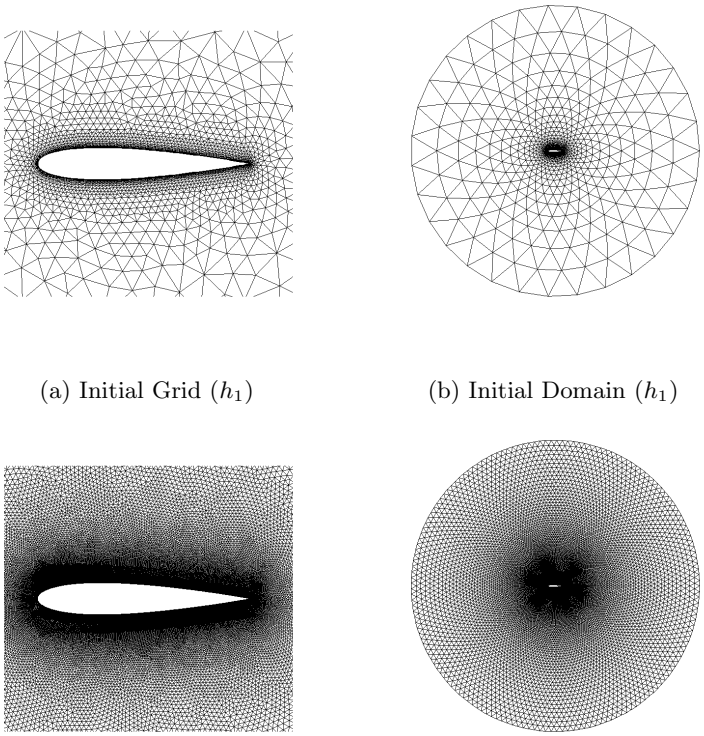
IV. Grid Generation

The NACA0015 geometry was selected for this study as it was used in the referenced experimental work.[20] A family of grids were generated using AFLR3[21] for the spatial convergence study. AFLR3 (Advancing-Front/Local-Reconnection) is a unstructured tetrahedral element grid generation code. An initial grid was uniformly refined by a constant factor of 1.5. Thus each newly created grid had the number of nodes defining the airfoil surface, and outer boundary spacing was increased by a r factor of 1.5: ie if the outer boundary had 32 nodes, the new boundary would have 48. Each grid was generated only with tetrahedral elements due to limitations in the feature-based adaptation in FUN3D. With these settings, the family of grids generated are presented in Table 2.

h	**Airfoil**	**Trailing Edge**	**Outer Boundary**	**Initial Cell Height (10^{-5})**	g_r	**Nodes**
1	128	2	32	5.693	1.4000	5626
2	192	3	48	3.795	1.2515	16878
3	288	5	72	2.530	1.1613	36546
4	432	8	108	1.687	1.1048	82284
5	648	12	162	1.124	1.0687	185292
6	972	18	243	0.749	1.0453	419258

Table 2: Geometric settings for NACA0015 spatial study

Where g_r is the geometric ratio of the boundary layer, and h is the current grid number. An example of the family of grids created is shown in Figure 1.



(a) Initial Grid (h_1) (b) Initial Domain (h_1)

(c) Refined Grid (h_5) (d) Refined Domain (h_5)

Figure 1: Uniform Grid Refinement Around Airfoil

IV.A. Flow Conditions and Kinematics

The flow conditions, as with the pitching kinematics are listed in Table 3 have been taken from Piziali.[20] The airfoil is pitched about the quarter-chord location following a the sinusoidal kinematic described as

$$\alpha = \alpha_m + \alpha_0 sin\left(\frac{2\pi k M_\infty t}{c}\right) \tag{18}$$

where α is the instantaneous angle of attack, α_m is the mean angle of attack, α_0 is the pitch amplitude, k is the reduced pitching frequency, M_∞ is the free stream Mach number, t is the simulation time, and c is the airfoil chord length.

$\alpha_\mathbf{m}$	$\alpha_\mathbf{0}$	$\mathbf{k}$	$\mathbf{M}_\infty$	$\mathbf{Re}$ $(\mathbf{10^6})$
13.07	4.27	0.134	0.289	1.947

Table 3: Flow conditions and kinematics used for pitching airfoil

For the temporal sensitivity analysis, the time step was also varied. If the time step is too large flow

features may be improperly resolved, if too small, then computational resources could be wasted. Like the grid generation, the time step was consecutively refined, but by a factor of 2. Five levels of refinement were chosen to test along side the spatial study, and are listed in Table 4. As i increases, so does the number of steps taken per pitching cycle. The time step Δt varied from 0.05 to 0.8, which corresponds to the flow translating 1% to 23% of the airfoil's chord (c).

i	**1**	**2**	**3**	**4**	**5**
steps	100	200	400	800	1600
Δt	0.8	0.4	0.2	0.1	0.05
Δx/c	0.23	0.12	0.06	0.03	0.01

Table 4: Tested time steps (Δt) used for temporal analysis, and the estimated flow convection distance per time step ($\Delta x/c$).

V. Uniform Grid Refinement

Before beginning the study on the pitching motion, the solver was benchmarked to ensure spatial convergence would be seen with a static airfoil. The flow conditions matched the conditions specified in Table 3 and the selected turbulence solver was the one equation Spalart Allmaras. Figure 2 reports the steady state force coefficients at $\alpha = 13.04°$ against $(N^{-1})^{\frac{1}{2}}$, where N is the number of nodes in the mesh, reported in Table 2. This assumes the characteristic edge length of the mesh, h, varies with the inverse of the square root, $h \approx N^{-1/2}$, for a 2D grid. The spatial discretization is also assumed to be second order, $p = 2$, thus flow variables of interest should vary with h^2.[3]

In Figure 2, the bottom axis can be read from left to right moving from the finest grid to the coarsest ($h_6 \leftarrow h_1$). The two plots show as the grid is refined, lift and drag coefficients begin to converge. For comparison, the Richardson Extrapolation is shown for a $h = 0$ grid. This simply verifies that as the grid is refined, the flow solution begins to converge onto a single answer.

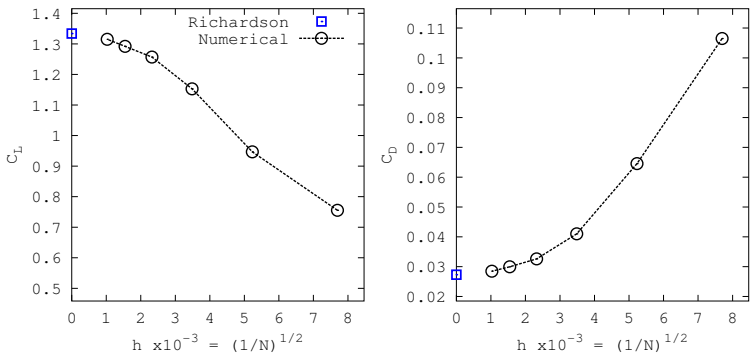
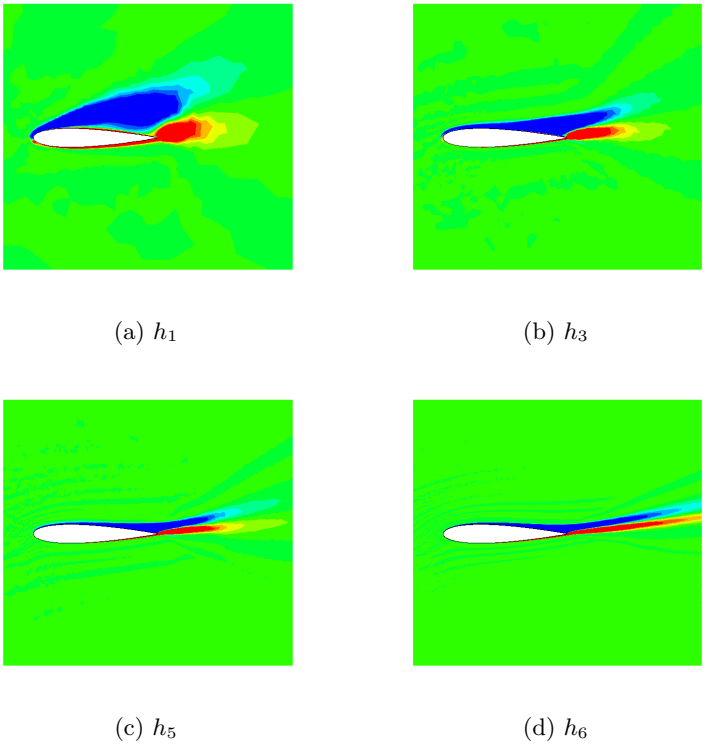


Figure 2: Static lift, drag, and moment coefficients at an angle of 13.04°.

Figure 3 illustrates how the flow changes with uniform refinement. It can be seen that a large separation region is prevalent about the airfoil in Figure 3a and is greatly reduced in Figure 3d. This helps illustrate the importance of grid resolution. The flow features can dramatic change with the grid used.



(a) h_1 (b) h_3

(c) h_5 (d) h_6

Figure 3: Changes in vorticity with grid resolution of a steady state angle of attack at 13.04°

V.A. $13 \pm 4°$ Pitching Convergence

In this section, we will begin to focus on the spatial and temporal convergence of a pitching airfoil. Each grid specified in Table 2 was tested at each time-step listed in Table 4, resulting in 36 unique combinations of grid density and time step sizing. It is important to note: results from grid combinations h_1, h_2, $\triangle t = 0.4, 0.8$ will not be discussed in the future sections.

Figures 4 and 5 show the averaged coefficient of lift and drag when the spatial and temporal resolutions are varied respectively. Figure 4 plots the effects of grid density on the averaged lift and drag coefficients. Lift, and drag show a strong dependence on the number of nodes used. As the spatial resolution of the grid is increased, the magnitude of the averaged forces begins to converge onto the extrapolated value.

Temporal resolution, figure 5, has little effect on the averaged force coefficients. At each grid resolution, reducing the time step modifies the averaged force coefficients little. Examining the instantaneous force coefficients, Figure 6, throughout the pitching cycle reveals a very different story. It can be seen that as the spatial resolution is increased, the magnitude of the forces is modified but the temporal resolution has control

over the shape of the force profile through the cycle. As the spatial resolution is increased, the flow features are better resolved due to increased grid density, but can be damped due to a coarse temporal resolution . Thus the flow field can be adequately resolved spatially, but not properly tracked through time. As time is refined, the flow feature is better projected altering the shape of the force profile through time.

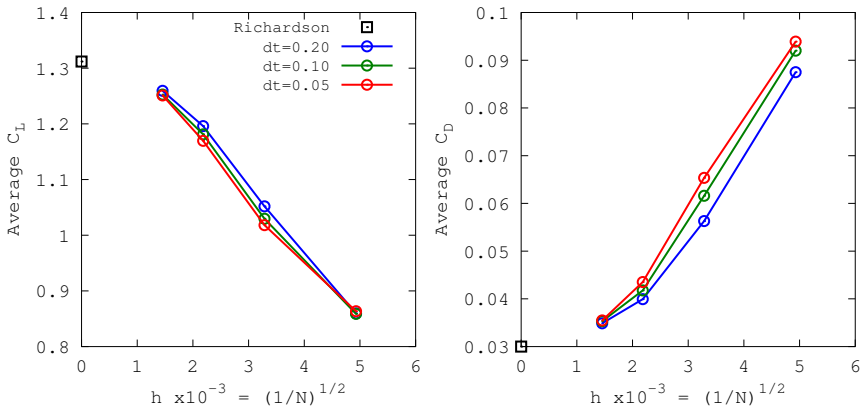


Figure 4: Lift, and drag coefficient variation due to selected spatial resolution of a pitching airfoil between $13 \pm 4°$.

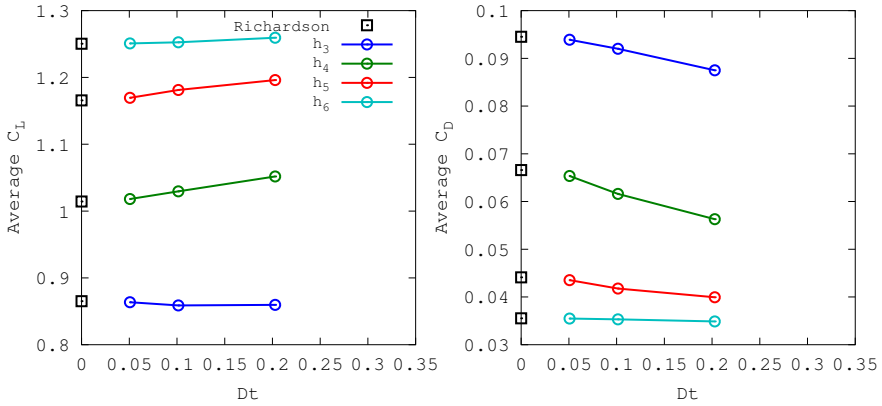


Figure 5: Lift, and drag coefficient variation due to selected temporal resolution of a pitching airfoil between $13 \pm 4°$.

V.B. Convergence Error

In order to conclude which grid and time step combination would give a satisfactory result, the error associated with each spatial and temporal combination must be compared. The error for averaged lift and drag coefficients were calculated using Eq. 16 and 17 are reported in Figure 7. The spatial error of the lift and drag coefficients (figure 7a) is seen to plateau as the grid density approaches h_6. As discussed in the last section, the temporal resolution has a greater impact on the instantaneous force coefficients rather than

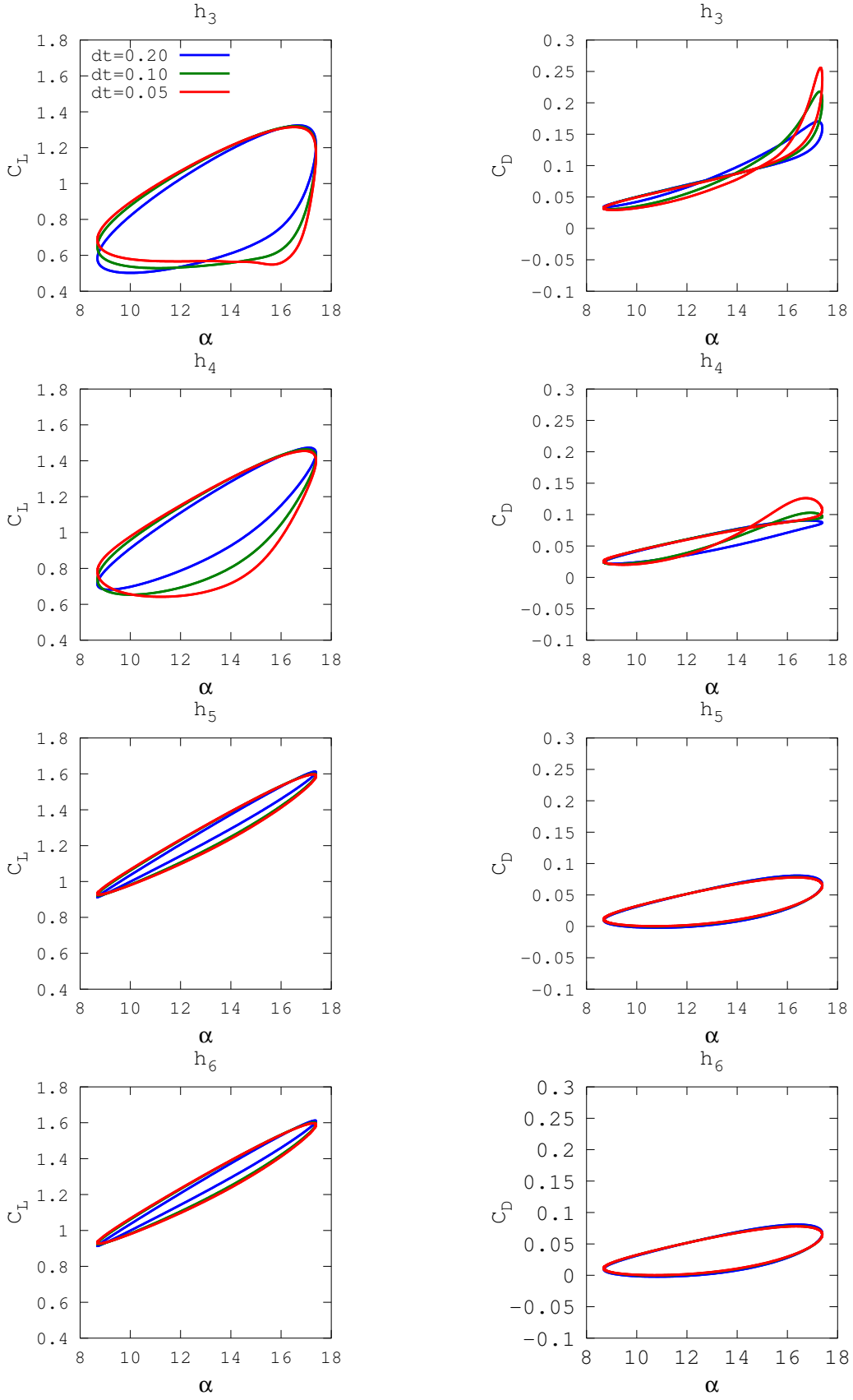
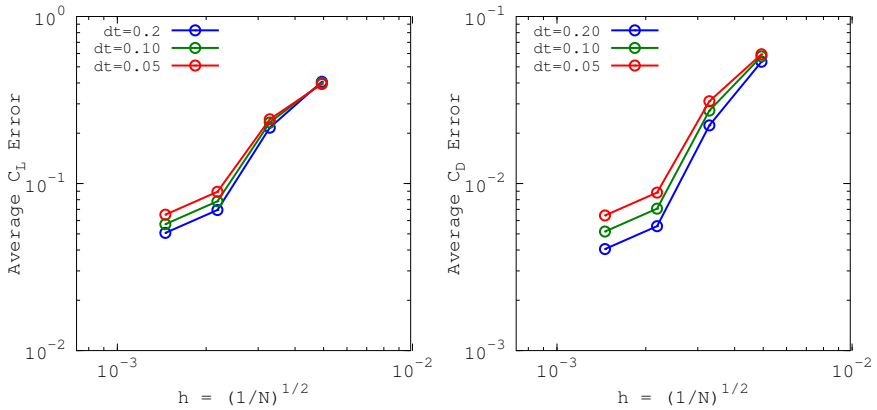
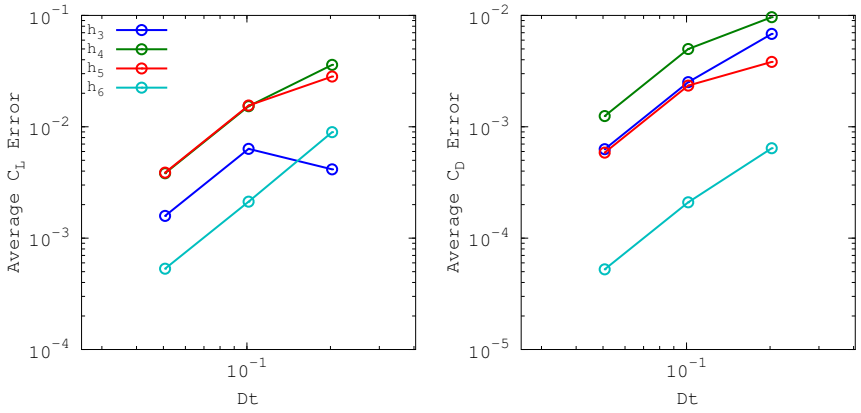


Figure 6: Lift, and drag coefficient of a pitching airfoil between $13 \pm 4°$ at $\Delta t = 0.05, 0.1$, and 0.2.

the averaged. Examining the temporal error (figure 7b), the decrease in error becomes linear as the grid is refined and the time step is reduced. However, the magnitude of error associated the averaged coefficients due to the temporal resolution is nearly two magnitudes lower than the spatial related error. Thus the time step should be chosen from examining the instantaneous force history and not the averaged error.



(a) Lift and drag coefficient error due to grid density.



(b) Lift and drag coefficient error due to time step.

Figure 7: Computed lift and drag coefficient errors due to selected spatial and temporal resolutions.

VI. Feature-Based Adaptation

In order to investigate whether feature adaption can feature-based adaptation can accelerate the reduction of spatial error using less nodes, a grid outside of the asymptotic convergence of lift and drag was selected. To prevent erroneous solutions from being developed from an under-refined grid, h_4 was selected due to being outside of the lift and drag convergence seen in figure 6 and 7a. Selecting h_4 for refinement will demonstrate whether feature-based adaptation is able to bring a grid (h_4) to a similar solution predicted by (h_6) without input by the user. A time step of $\Delta t = 0.1$ was used, and all flow conditions remained equal to those specified

in Table 3.

Before discussing the settings used, the process used to determine the adaptation metric must be introduced. Figure 8 illustrates the iterative process used. An initial grid, h_4, is supplied to the flow solver. After each time step, the adaptation metric is computed with the current flow solution, and then intersected with the previous calculated metric. The flow solution is advanced in time, and the grid moved. This process repeats until four pitching cycles were completed. The adaptation metric is then passed to the grid adaptation modules implemented in FUN3D. The grid is then iterated until the adaptation metric is satisfied. The grid is then saved, and passed back to the flow solver and the flow entire process repeats.

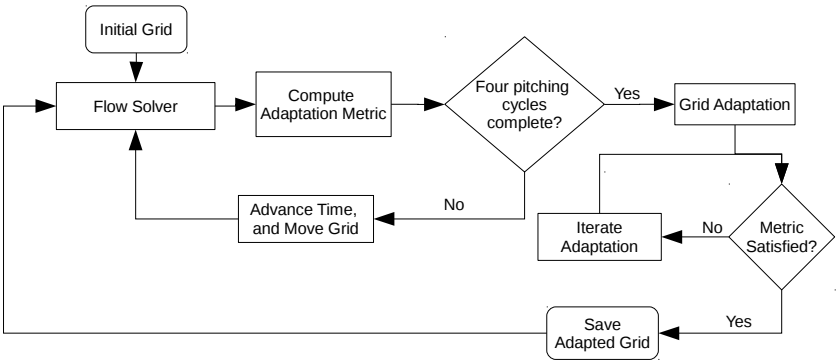


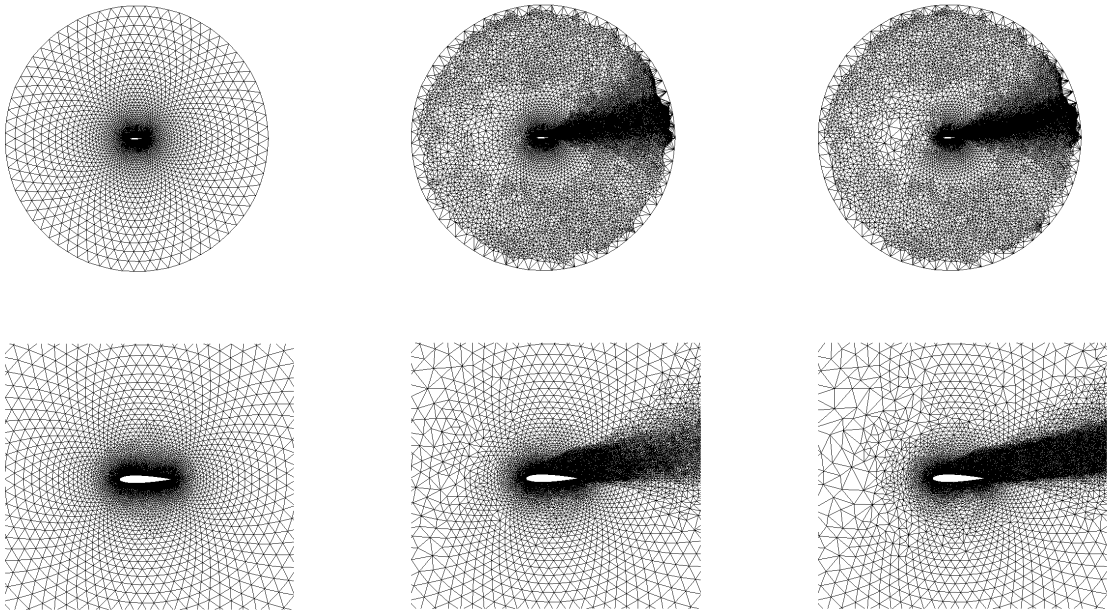
Figure 8: Feature-based adaptation process

Each flow solver and adaptation iteration possessed the same flow settings as previously stated, and was allowed to iterate 6 times. During each iteration, the adaptation was allowed to increase, or decrease cell volume by moving, adding and/or removing nodes, or swapping edges. The maximum desired node count was set to an equivalent node count of h_6 ($\approx 400,000$ nodes) after 6 iterations. Each iteration allowed the grid to increase the global node count in increments of 18%. This did not mean the adapted grid reached this maximum node count. At each adaptation iteration, the grid was only allowed to grow at a specified rate to ensure needless grid growth was avoided. The boundary layer was frozen at a y+ distance of 100 to keep the boundary layer from being modified.

VI.A. Adapted Mesh

Before discussing the results, the adapted mesh produced by the feature adaptation will be examined. Not all meshes will be shown, as it would be superfluous. Figure 9 shows three meshes from a selection of iteration steps 0 (initial),4, and 6. Comparing the initial and the 4^{th} iteration of the mesh, it is easily seen there has been significantly modification. Adaptation has mainly occurred in the wake and around the airfoil, with little adaptation occurring elsewhere. Comparing steps 4, and 6, it is difficult to distinguish any change in

the mesh. It was confirmed through the examination of the global node count that the node density was increased, and can be seen in figure 10. To better observe how the grid was adapted, the change in cell volume was examined.



(a) Initial Grid (b) 4^{th} Adaptation (c) 6^{th} Adaptation

Figure 9: The initial, 4^{th}, and 6^{th} mesh adaptation of the selected h_4 grid.

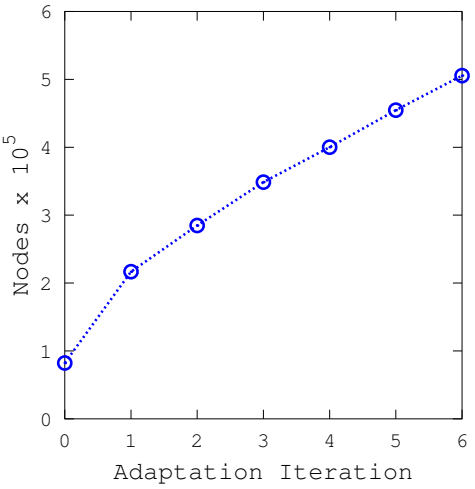


Figure 10: Node increase with adaptation at each iteration

By observing the change in cell volume from each iteration step, how and where the mesh was adapted is easily distinguished. By calculating the cell volume in each mesh, and dividing them, a ratio of the change in cell volume could be observed such that:

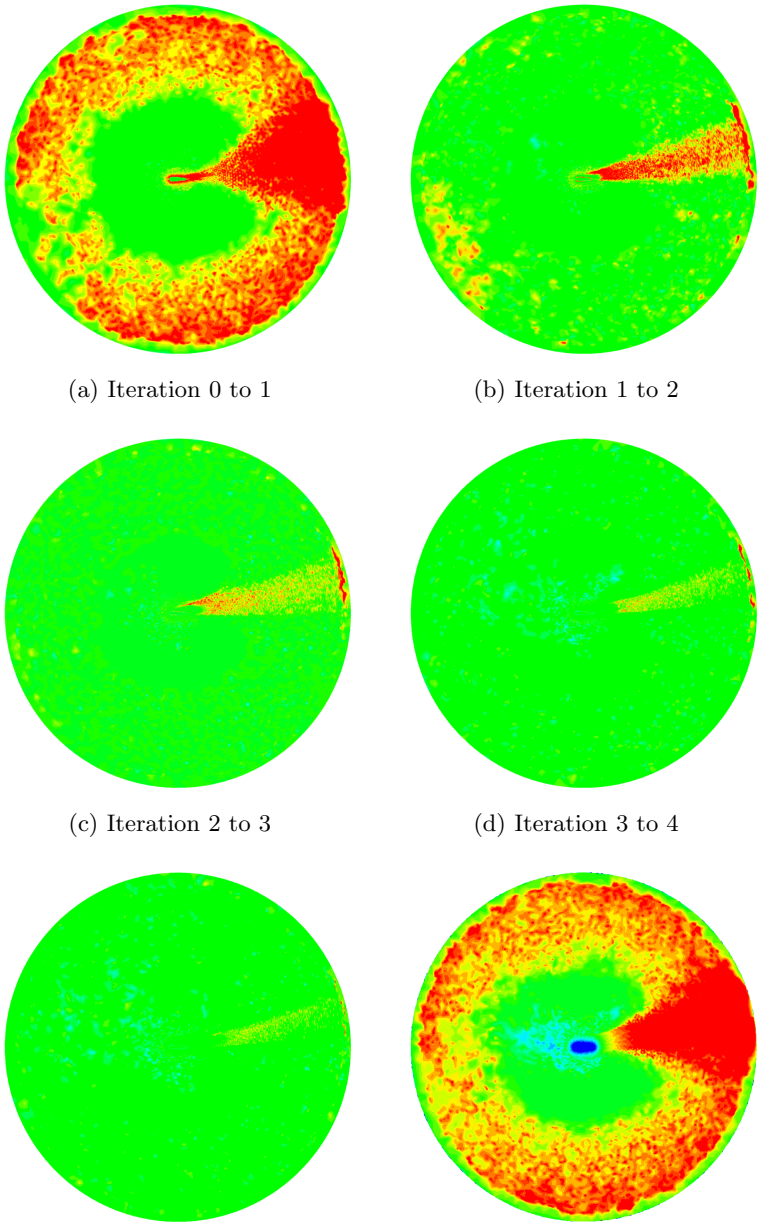
$$R_{cv} = \frac{V_{n-1}}{V_n} \tag{19}$$

Where R_{cv} is the ratio of change in cell volume, V is the cell volume, and n is the iteration number. If $R_{cv} > 1$ the cell volume between the two iterations was reduced (refinement), if $R_{cv} < 1$ cell volume was decreased (coarsening), finally if $R_{cv} \approx 1$ there was little to no change in the cell. This gives a straight forward visualization of where and how the mesh was adapted. Results of the change in cell volume are shown in Figure 11 for iterations 0 through 6, and the cumulative change.

Looking at the overall change in R_{cv} in Figure 11 the largest change in cell volume occur in the first 4 iterations in the wake of the airfoil, similar to Ref.[3] for a hypersonic case. As the iteration count increases towards 6, there is less change observed in the domain. Figure 11f illustrates the overall change in cell volume between iterations 0 and 6 which shows some changes that were not previously observed. Coarsening has appeared around the airfoil itself which did not show in the single consecutive cell ratio plots. This coarsening did however have an affect on the resulting flow field, and will be discussed in the next section.

VI.B. Adapted Flow Field

Immediate improvements to the airfoil wake can be seen from the grid adaptation. Figure 12 plots the cycle averaged vorticity magnitude produced by the pitching airfoil using the initial, 2^{nd}, 4^{th}, and 6^{th} adapted meshes. The cycle average was calculated by averaging the vorticity magnitude at each node during two pitching cycles. Unlike the change in cell volume, where the greatest change was observed in the first few iteration, the vorticity continues to become better resolved as the adaptation continues. After the second adaptation, the wake of the airfoil is carried out twice the distance compared to the initial grid. As the adaptation continues, the average vorticity magnitude increases further downstream, and the wake becomes less diffuse.



(a) Iteration 0 to 1

(b) Iteration 1 to 2

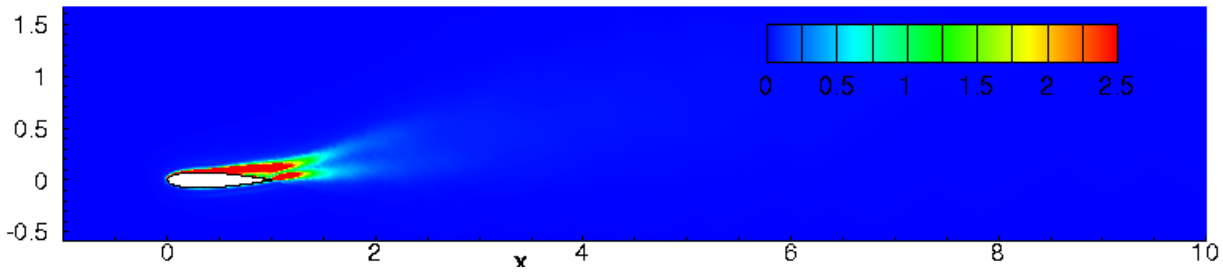
(c) Iteration 2 to 3

(d) Iteration 3 to 4

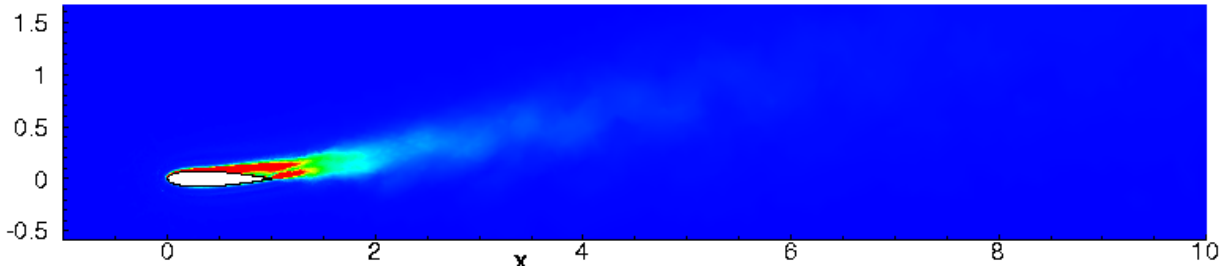
(e) Iteration 4 to 5

(f) Overall Change (0 to 6)

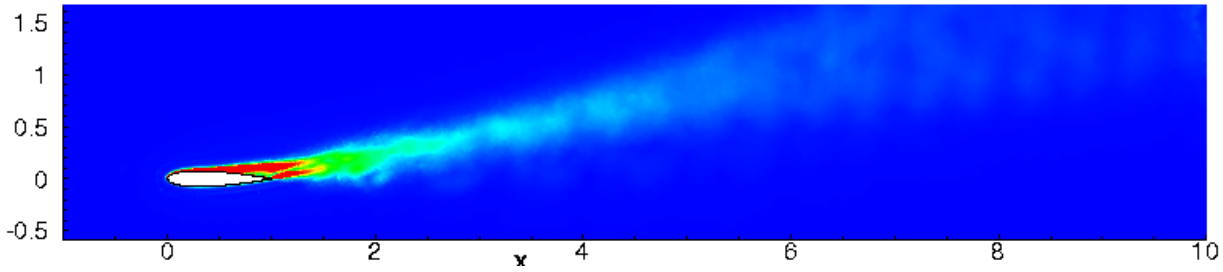
Figure 11: Change in cell volume ratio (R_{cv}) between consecutive adaptation iteration steps. (Red) $R_{cv} > 1$, (Green) $R_{cv} \approx 1$, (Blue) $R_{cv} < 1$



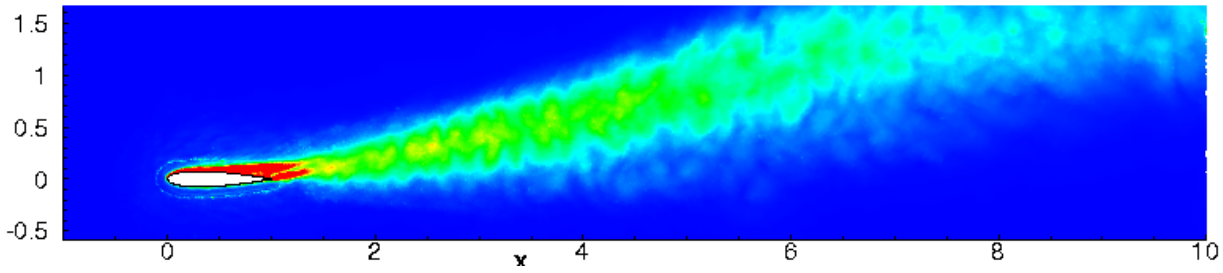
(a) Initial Grid



(b) 2^{th} Adaptation



(c) 4^{th} Adaptation

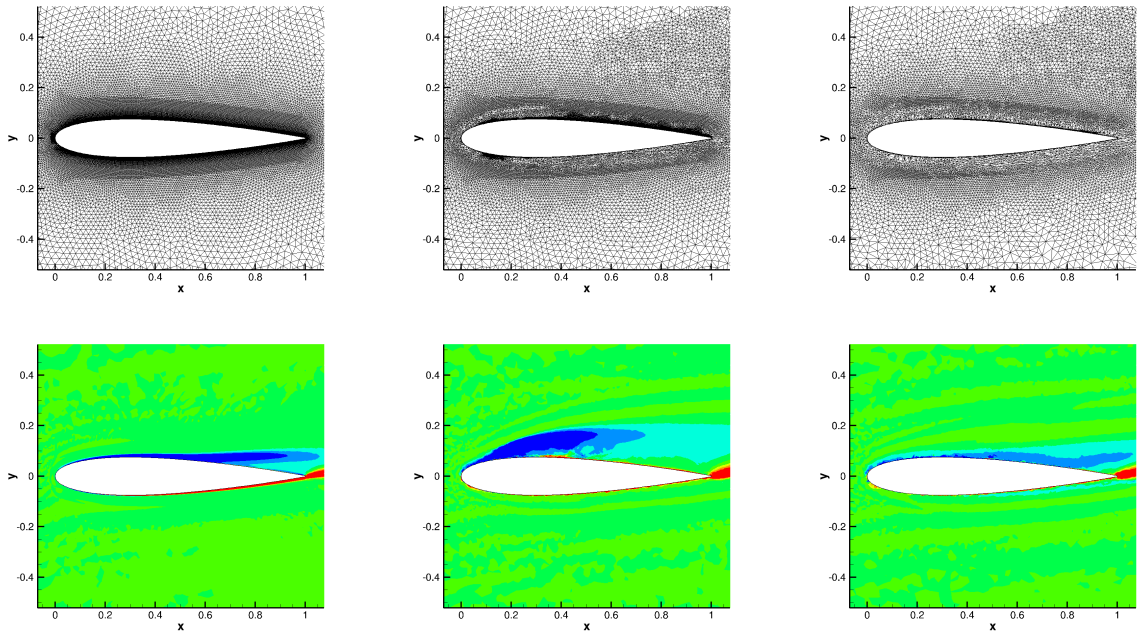


(d) 6^{th} Adaptation

Figure 12: Cycle averaged vorticity magnitude of the initial, 2^{nd}, 4^{th}, and 6^{th} adapted meshes.

As discussed previously, figure 11f revealed coarsening occurring around the airfoil during consecutive adaptation. Snapshots were taken at the mean angle of attack to illustrate the effects of adapted coarsening had on the flow field. The initial mesh (figure 13a) shows vorticity along the upper surface of the airfoil detaching at approximately at $x/c = 0.75$. At the second iteration (figure 13b), coarsening of the mesh can be seen near the surface of the airfoil and vorticity detaches sooner (approximately $x/c = 0.25$). Vorticity

begins to be reattached as the adaptation and coarsening continues (figure 13c) but the vorticity magnitude is weaker compared to the initial grid. This is the result of inadequate grid resolution near the surface of the airfoil, which results in the reduction of force coefficients (shown in figure 14).



(a) Initial Mesh (b) Iteration 2 (c) Iteration 6

Figure 13: Coarsening of mesh and changes in vorticity close to the airfoil surface due to mesh adaptation.

VI.C. Adapted Forces and Error

Figures 14 plots the instantaneous lift and drag coefficients for four selected adaptation iterations: the initial, 2^{nd}, 4^{th}, and 6^{th} adapted meshes listed as I_0, I_2, I_4, and I_6 respectively. Like the grid adaptation, the force coefficients show their largest change in the first few adaptations. This change is due to the effect of slow coarsening around the airfoil that occurs throughout the adaptation history observed in figure 11. However, as the grid continues to adapt, the instantaneous forces first diverge away from I_0 and then begin to converge back towards I_0.

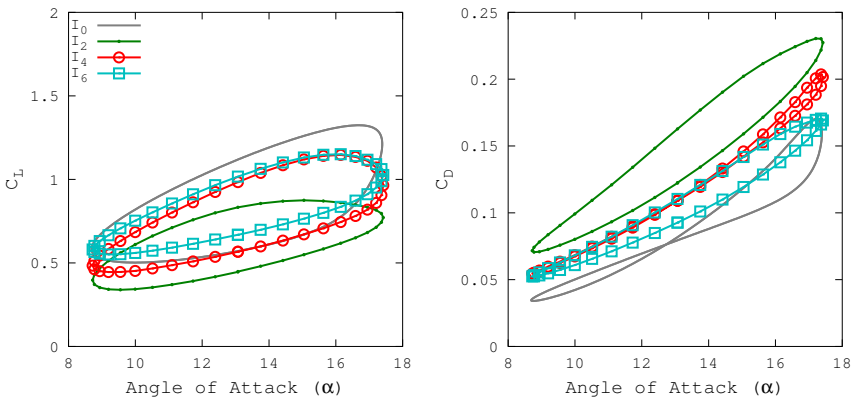


Figure 14: Instantaneous lift, drag, and moment coefficients of the initial, 2^{nd}, 4^{th}, and 6^{th} adapted meshes.

Examining the force averages, and errors (figure 15) show similar results. The two force coefficients were seen to diverge quickly away from the expected trend of approaching the uniform refinement values. The first adaptation results in an approximate 50% reduction in the lift coefficient, with tripling of the average drag. Due forces changing quickest in the first set of adaptations, the spatial error associated with the adapted grids resulted in a "noisy" error plot. However, adaptations 3 through 6 show a rate of error reduction greater than the uniform refinement. While coarsening affected the flow field, the adaptation did continue to reduce the spatial error from the grid.

In summary, the grid adaptation was seen to have it's largest effects in the first few iterations. Slow coarsening of the mesh around the airfoil resulted in altered lift and drag coefficients. However, refinement of the wake of the airfoil allowed vorticity to be carried out significantly further than the initial mesh. Thus feature-based adaptation can be an effective means of wake refinement for dynamic stall when the practitioner has no prior knowledge of the wake characteristics.

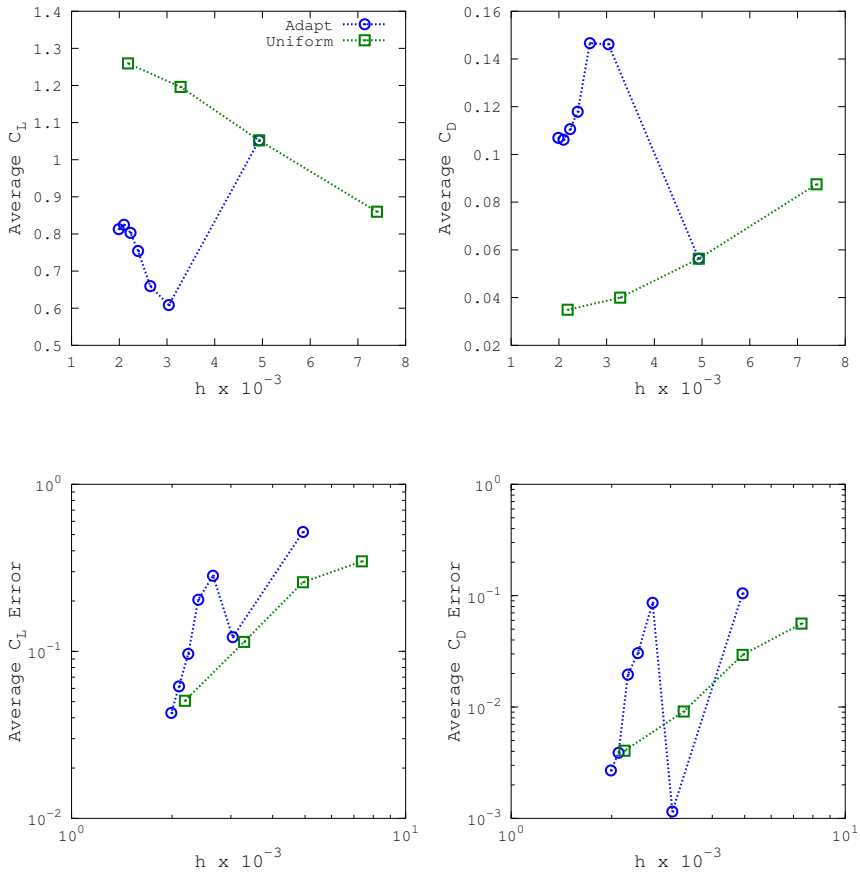


Figure 15: Average and error of lift, drag, and moment coefficients of adapted grids

VII. Conclusion

A dynamic stall motion was selected from a previous experimental study that replicates sinusoidal pitching motion. A family of uniformly refined grids was generated and evaluated to find the reduction in spatial and temporal error associated with each grid. It was found as the grids were uniformly refined, the average spatial error for each grid was reduced. The selected time step was found to impact the average forces little, but had a large effect on the instantaneous force profile; primarily the down-stroke of the motion.

For the feature-based adaptation, a single grid and time stepping combination was selected from the uniformly refined grid family, and tested as to whether feature adaption would improve the rate of convergence as the grid resolution was increased. Coarsening occurred around the airfoil altering the vorticity being generated by the wing. This reduced the strength of vorticity on the surface of the airfoil, and initially reduced the instantaneous force coefficients. However, as the adaptation continues, instantaneous forces were seen to converge near their original values. In the examination of the spatial errors, the rate of error

reduction was seen to be greater than that of the uniform refinement after the second adaptation. Significant wake refine was also observed allowing vorticity to be carried well behind the airfoil.

VIII. Acknowledgments

The authors would like to thank Dr. Mike Park of NASA Langley for mentoring this study. This work has been partially funded by the NASA Langley Research Summer Scholars program.

References

[1]Kravchenko, A. G. and Moin, P., "Numerical studies of flow over a circular cylinder at Re= 3900," *Physics of fluids*, Vol. 12, 2000, pp. 403.

[2]Komerath, N. M., Smith, M. J., and Tung, C., "A Review of Rotor Wake Physics and Modeling," *Journal of the American Helicopter Society*, Vol. 56, No. 2, 2011, pp. 22006–1–22006–19.

[3]Bibb, K. L., Gnoffo, P. A., Park, M. A., and Jones, W. T., "Parallel, Gradient-Based Anisotropic Mesh Adaptation for Re-entry Vehicle Configurations," *9th AIAA/ASME Joint Thermophysics and Heat Transfer Conference*, 2006.

[4]Kang, H. and Kwon, O., "Effect of Wake Adaptation on Rotor Hovering Simulations Using Unstructured Meshes," *Journal of Aircraft*, Vol. 38, No. 5, 2001.

[5]Park, Y. and Kwon, O., "Simulation of Unsteady Rotor-Fuselage Interactions Using Unstructured Adaptive Sliding Meshes," *Journal of Aircraft*, Vol. 49, No. 4, 2006.

[6]Thomas, G. W. W. A. J. and Krist, S., "Grid convergence for adaptive methods," *AIAA*, Vol. 10th Computational Fluid Dynamics Conference, No. 91–1592, 1991.

[7]Sweby, P. and Yee, H., "On the dynamics of some grid adaptation sciences," *Proc. 4th Int. Conf on Numerical Grid Generation in Computational Fluid Dynamics and Related Fields, Swansea, UK*, 1994.

[8]Baker, T. J., "Mesh adaptation strategies for problems in fluid dynamics," *Finite Elements in Analysis and Design*, Vol. 25, No. 3–4, 1997, pp. 243 – 273, ¡ce:title¿Adaptive Meshing, Part 2¡/ce:title¿.

[9]Biedron, R. T., Vatsa, V. N., and Atkins, H. L., "Simulation of unsteady flows using an unstructured navier-stokes solver on moving and stationary grids," *AIAA paper*, Vol. 5093, 2005, pp. 2005.

[10]Anderson, W. K. and Bonhaus, D. L., "An implicit upwind algorithm for computing turbulent flows on unstructured grids," *Computers & Fluids*, Vol. 23, No. 1, 1994, pp. 1–21.

[11]Vatsa, V. N. and Carpenter, M. H., "Higher Order Temporal Schemes with Error Controllers for Unsteady Navier-Stokes Equations," *AIAA Paper*, Vol. 5245, 2005, pp. 2005.

[12]Rumsey, C. L., Sanetrik, M. D., Biedron, R. T., Melson, N. D., and Parlette, E. B., "Efficiency and accuracy of time-accurate turbulent Navier-Stokes computations," *Computers & Fluids*, Vol. 25, No. 2, 1996, pp. 217–236.

[13]Peraire, J., Vahdati, M., Morgan, K., and Zienkiewicz, O. C., "Adaptive remeshing for compressible flow computations," *Journal of computational physics*, Vol. 72, No. 2, 1987, pp. 449–466.

[14]Park, M. A., "Three-dimensional turbulent RANS adjoint-based error correction," *AIAA paper*, Vol. 3849, 2003, pp. 1–14.

[15]Park, M. A. and Darmofal, D. L., "Parallel anisotropic tetrahedral adaptation," *AIAA Paper*, Vol. 917, 2008, pp. 2008.

[16]Park, M. A., *Anisotropic output-based adaptation with tetrahedral cut cells for compressible flows*, Ph.D. thesis, Massachusetts Institute of Technology, 2008.

[17]Park, M. A., "Adjoint-based, three-dimensional error prediction and grid adaptation," *AIAA journal*, Vol. 42, No. 9, 2004, pp. 1854–1862.

[18]Lee-Rausch, E., Park, M., Jones, W., Hammond, D., and Nielsen, E., "Application of parallel adjoint-based error estimation and anisotropic grid adaptation for three-dimensional aerospace configurations," *AIAA paper*, Vol. 4842, 2005, pp. 2005.

[19]Richardson, L. F. and Gaunt, J. A., "The deferred approach to the limit. Part I. Single lattice. Part II. Interpenetrating lattices," *Philosophical Transactions of the Royal Society of London. Series A, containing papers of a mathematical or physical character*, Vol. 226, 1927, pp. 299–361.

[20]Piziali, R., "2-D and 3-D oscillating wing aerodynamics for a range of angles of attack including stall," Tech. rep., NASA, 1994.

[21]State, M., "Advancing-Front/Local-Reconnection," January 2013, Version 12.28.57.