Comparing Unstructured Adaptive Mesh Solutions for the High Lift Common Research Model Airfoil

Todd Michal[1] and Joshua Krakos[2]
Boeing Research and Technology, St Louis, MO, USA

Dmitry Kamenetskiy[2]
Boeing Research and Technology, Seattle, WA, USA

Marshall Galbraith[3] and Carmen-Ioana Ursachi[4]
Massachusetts Institute of Technology, Cambridge, MA, USA

Michael A. Park[5] and W. Kyle Anderson[6]
NASA Langley, Hampton, VA, USA

Frederic Alauzet[7] and Adrien Loseille[7]
INRIA Paris-Saclay, Palaiseau, France

Discretization error is a common source of uncertainty in Computational Fluid Dynamics (CFD) analyses. Traditional means of controlling discretization error through fixed-mesh refinement studies has proven to be difficult particularly when modeling complex geometries and flow fields. One reason for this is that mesh generation in today's production CFD workflow is often a labor intensive process that is heavily dependent on user judgment. Unstructured mesh adaptation is known to be an efficient way to control discretization errors in CFD. Adaptive methods replace user based decision making with automated processes that optimize a mesh to reduce discretization error.

This paper compares the application of multiple solution adaptive techniques in combination with multiple flow solvers to solve for the flow field about a 2D airfoil section of the NASA High-Lift Common Research Model (HL-CRM). By driving the adaptive mesh processes to a similar level of mesh convergence, the ability to achieve consistent results between multiple adaptive techniques and flow solvers is demonstrated. Mesh convergence for the various adaptive mesh approaches is compared identifying potential areas for improvement and providing mesh generation guidance for future workshops.

I. Introduction

Accurate prediction of the forces acting on a transport aircraft approaching wing stall has proven to be challenging as evidenced by results of the AIAA series of High-Lift Prediction Workshops [1,2]. An example of a high-lift flow field is shown in Figure 1. This type of analysis has a number of characteristics that make numerical analysis difficult. It is crucial that the analysis capture the interactions of multiple geometry induced shear layers (attached and detached) and vortices. Adequately resolving these features presents a difficult challenge to mesh generation as it is impossible to know the exact strength and location of these features prior to running a flow solution. In addition, feature locations change with each new flow condition (i.e. angle of attack or airspeed). As a result an educated guess of the importance

[1] Technical Fellow, AIAA Associate Technical Fellow.
[2] Senior Engineer, AIAA Member.
[3] Research Engineer, Department of Aeronautics & Astronautics, AIAA Senior Member.
[4] Graduate Student, Department of Aeronautics & Astronautics, AIAA Student Member.
[5] Research Scientist, Computational AeroSciences Branch, AIAA Associate Fellow.
[6] Senior Research Scientist, Computational AeroSciences Branch, AIAA Associate Fellow.
[7] Researcher, GAMMA3 Team, AIAA Member.

and location of off body features is typically made and broad areas of the mesh are refined to attempt to resolve them. This approach often misses or under resolves important features increasing solution error and resulting in costly rework to modify the mesh and rerun the analysis.

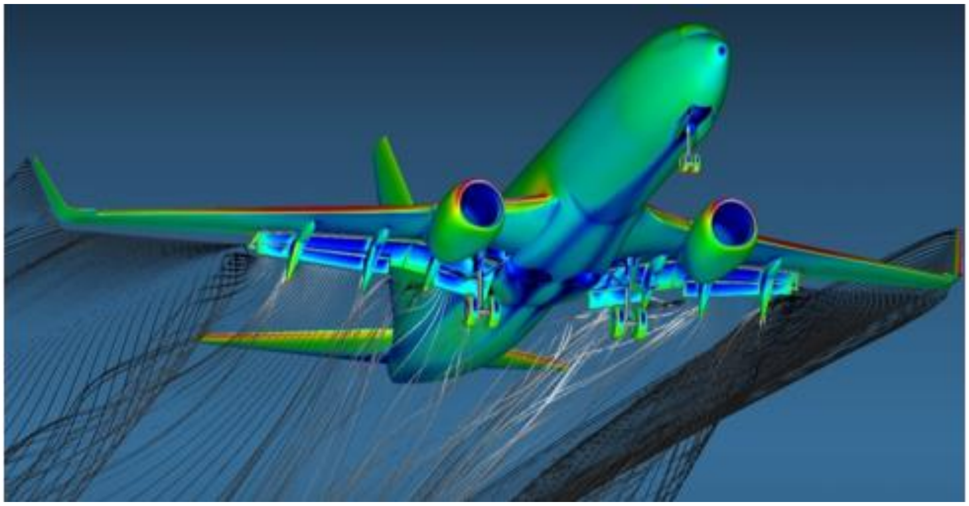

Figure 1: Computational Analysis of High-Lift Flow Field

Recently adaptive meshing techniques for unstructured meshes have matured to the point that adaptive Reynolds-averaged Navier-Stokes (RANS) solutions on full aircraft configurations are now practical [3-6]. These methods automatically adapt the mesh to minimize an error measure for a given number of Degrees of Freedom (DOF). By directly targeting solution error, these methods can achieve a given level of accuracy more efficiently than the fixed mesh approach. All characteristics of the adapted mesh including element size and orientation are determined automatically from the error analysis: removing the need for meshing guidelines or for an expert user to guide the mesh generation process.

A common approach for evaluating discretization error in a numerical simulation is to compute solutions on a sequence of meshes with increasing density. The solution variation between meshes gives an assessment of the sensitivity to mesh resolution. Meshes in the sequence are typically generated to match a set of meshing guidelines in an effort to make the meshes parametrically equivalent. Meshing guidelines are typically based on best practices gained over many years of CFD application experience on similar problems. Richardson extrapolation is then used to estimate the output of interest such as lift on an infinitely fine mesh. This approach has been used in studies such as the AIAA Drag Prediction and High-Lift Prediction workshops to assess mesh convergence. These efforts have exposed a number of issues with this approach. Generation of a consistent family of meshes that properly exhibits asymptotic convergence in the output of interest has proven to be difficult. This is particularly true for unstructured meshes were it is nearly impossible to maintain consistent mesh topology in the family of meshes. A common practice is to build a series of fixed meshes by varying inputs to the mesh generator to achieve a desired growth in the number of mesh elements between mesh levels. Characteristics of the resulting meshes are governed by the mesh generation algorithm and may vary between mesh levels resulting in a loss of consistency.

While achieving mesh convergence via fixed mesh refinement studies has proven difficult in 3D, better success has been achieved in 2D where it is practical to continue refinement of the mesh families to very high element counts. For example NASA's turbulence modeling resource [7] has demonstrated near mesh convergence on several 2D problems. While reduced dimension problems lack many of the complex physical interactions found in real applications, 2D test cases can be useful for assessing algorithmic and mesh convergence properties of computational approaches.

In this paper adaptive mesh simulations about a 2D cross section of the NASA High Lift Common Research Model wing are compared for a range of angles of attack. Simulations are computed using a combination of four flow solvers, four mesh mechanics methods and four separate error estimates. Variations in the adaptive mesh processes are examined. Solutions with a similar level of mesh convergence from each method are compared to demonstrate the level of consistency that is achievable with current adaptive meshing techniques. Mesh convergence properties of the adaptive approaches are analyzed to develop recommendations for future applications to high lift applications.

II. Adaptive Mesh Process

The components of metric-based anisotropic unstructured mesh adaptation are shown on Figure 2. Starting with an initial mesh, a flow solution (and optionally, an adjoint solution) is computed. The information from these flow solutions are used to estimate error and specify a new mesh resolution request via an anisotropic metric field M. If the estimated errors are larger than limits specified by the practitioner, the current mesh system is modified by mesh mechanics to conform to the anisotropic metric M. Once the adapted mesh is available, the previous flow solution is optionally interpolated to the new mesh to provide an initial condition for the flow solver that approximates the converged solution. This improved initial condition can decrease the execution time and improve the robustness of the flow solution calculation, but standard initialization is also possible. The process is repeated until an exit criteria is met (e.g., accuracy requirement, resource limit). There are potential interactions between each of these elements that impact the overall convergence and efficiency of the adaptation process. Details of the specific implementations of these components is detailed in the following sections.

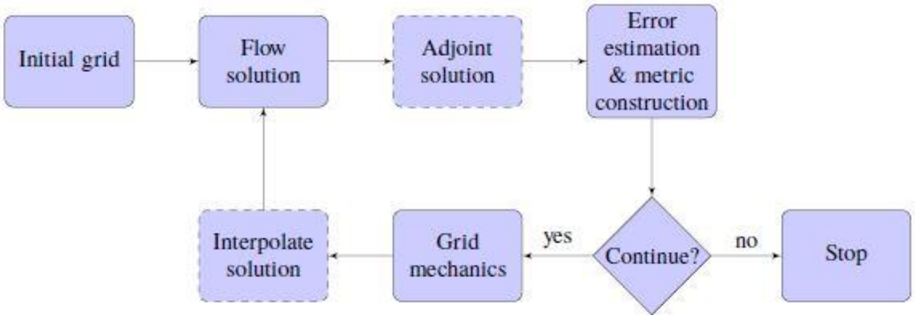

Figure 2: Solution-based mesh adaptation process with optional components indicated by dashed outlines.

There are several ways a metric field, M, can be constructed that encodes anisotropic information. Loseille and Alauzet [8] provide a thorough introduction to the metric tensor field. Here we consider two kinds of metric field construction techniques: feature-based and output-based. Two types of feature-based metrics will be compared in this study, a heuristic metric constructed by the Anisotropic Hierarchical Adapation code (AHA) and the multiscale metric. The multiscale metric controls the L^p norm of the interpolation error of a solution scalar field and forms the foundation of some output-based metrics [8-10]. Output-based metric formulations include the adjoint solution, which allows the targeting of a goal or functional output (e.g., lift, drag).

The number of desired vertices or elements for a mesh generated from a metric field is specified by scaling the metric field to match a target complexity. The complexity, C, of a continuous metric field, M, is defined as the integral

$$C(M) = \int_\Omega \sqrt{\det(M(x))}\, dx \tag{1}$$

and is evaluated on the discrete mesh and metric. The relationship between C and the number of vertices and elements in the adapted mesh is shown theoretically by [8] experimentally by [9-11]. Details of the metric construction for each code is provided in Section V.

Mesh adaptation is an inherently nonlinear process. The robustness of the adaptive procedure is enhanced and the execution time is reduced by optimizing the mesh at a fixed-complexity, which allows for control of DOF. An adaptive series of meshes is created by optimizing a mesh at a fixed complexity and then increasing the target complexity in a series of steps, where the complexity is held fixed and the mesh is optimized at each step. The progression of forces and moments during these steps is referred to as a trajectory in the result sections. Typically, 5 to 10 fixed-complexity adaptation iterations are performed before increasing the complexity to the next target. These fixed-complexity adaptations are referred to as subiterations in Michal et al. [12], where only the final force or moment value is plotted for the fixed-complexity target. Alternatively, an average of a number of fixed-complexity adaptations can be used to reduce the jitter in trajectories.

III. Flow Solvers

Four flow solvers are utilized to compute the high-lift airfoil flow solutions. The details of the discretization and nonlinear solution scheme can impact the performance of the flow solver component on the integrated mesh adaptation process, in particular, on highly anisotropic meshes. A strong nonlinear update scheme can positively impact the overall robustness of the process and a low-diffusion discretization can approximate numerical solutions on coarser meshes better than higher-diffusion schemes.

A. FUN3D-SFE

FUN3D-SFE is a continuous Stabilized Finite-Element discretization within FUN3D [13]. The discretizationoptions include the Streamlined Upwind Petrov-Galerkin (SUPG) scheme [14,15], Galerkin Least-Squares (GLS) [16], and variational multiscale methods [17]. In the results shown here, only the SUPG scheme is considered. In the current implementation, the SA-neg turbulence model is tightly coupled with the flow equations, yielding a nonlinear algebraic system of equations with six variables at each vertex. A linear nodal basis is used in this study, which is designed to be 2nd-order accurate in space. The current implementation includes the capability for computing on tetrahedra, hexahedra, pyramids, and prisms in 3D. All the results shown in the present paper are for prismatic meshes, which are 2D triangular meshes extruded to a single layer of prisms in 3D.

To advance the solution toward a steady state, the density, velocities, temperature, and the turbulence working variable are updated in a tightly-coupled Newton-type solver described by Anderson, Newman, and Karman [13]. Here, an initial update to the flow variables is computed using a locally varying time-step parameter that is later multiplied by the current CFL number, which is adjusted during the iterative process as described in the next paragraph.

At each iteration, a linearized residual matrix is formed and solved using the GMRES algorithm with a preconditioner based on an ILU decomposition with two levels of fill [18] and a Krylov subspace dimension of 300. Using the full update of the variables, the L^2 norm of the unsteady residual is compared to its value at the beginning of the iteration. If the L^2 norm after the update is less than one half of the original value, the CFL number is doubled and the iterative process continues to the next iterative cycle. If the L^2 reduction target for the residual is not met, a line search is conducted to determine an appropriate relaxation factor. Here, the L^2 norm of the residual is determined at four locations along the search direction and the optimal relaxation factor is determined by locating the minimum of a cubic polynomial curve fit through the samples. After the line search, the solution is updated using the relaxation factor and the CFL number is neither increased nor decreased.

B. GGNS

GGNS (General Geometry Navier-Stokes) is a Boeing-developed flow solver built upon the SUPG finite-element discretization. The code uses piecewise linear finite elements resulting in a 2nd-order accurate discretization. Additional 1st-order artificial viscosity built upon the DG discretization is added for shock capturing. The indicator triggering this additional stabilization is based on the oscillation of the Mach number across a cell. The solver uses unstructured meshes of mixed-element type (tetrahedrons, prisms, and pyramids) as well as purely-tetrahedral meshes. The number of DOF for the 2nd-order SUPG scheme is equal to the number of vertices in the computational mesh. The discretization is vertex-based in the sense that it is conservative over the dual volumes of an unstructured mesh. More details on discretization used in the GGNS solver, including the particular choices of discretization variables and special treatment of the essential boundary conditions via the Lagrange-multiplier based technique [19], can be found in Kamenetskiy et al. [20]. Forces are computed using the residual balance at the boundary.

The discrete nonlinear solver in the GGNS code implements a variant of the Newton-Krylov-Schwarz algorithm using PETSc. Time stepping is employed to drive to the steady state solution. On each time step, an exact Jacobian matrix for the discretization is formed by an automatic differentiation technique. The linear system arising from the Newton's method is approximately solved using GMRES with a drop-tolerance-based block-ILU preconditioner (locally on subdomains) implemented in the context of the additive Schwarz method with minimal overlap [21]. Right preconditioning is employed to maintain consistency between the nonlinear and linear residuals. The compact stencil property of the SUPG scheme helps to reduce the fill-in levels in the approximate factorization, thereby reducing the memory footprint.

A line search is applied along the direction provided by the approximate solution of the linear system. Residual decrease and physical realizability of the updated state are tracked during the line search. A heuristic feedback algorithm is implemented to communicate failure of the line search back to the time-stepping algorithm, so that the CFL number can be increased or decreased as necessary. There is no upper preset limit for the CFL number in the time-marching algorithm; so Newton-type quadratic convergence (or, at least, superlinear, due to inexact linear solves) is routinely achieved at steady state.

C. SANS

Solution Adaptive Numerical Simulator (SANS) [22], currently under development at the Massachusetts Institute of Technology, is a general framework for solving discrete finite-element approximations to advection-diffusionreaction type PDEs, such as scalar advection-diffusion, Navier-Stokes, and RANS equations. A range of finite-element methods are currently implemented in SANS, including high-order discontinuous (DG), hybridized discontinuous (HDG), and continuous (CG) Galerkin finite-element methods. The Variational Multiscale Method with Discontinuous Subscales (VMSD) [23,24] discretization is used to compute all solutions to the RANS equations presented here. Boundary conditions are weakly enforced and forces are computed using the residual balance at the boundary.

The nonlinear system of equations is solved using pseudotime continuation (PTC) damped Newton's method with a line search to ensure residuals decrease. The complete linearization of the residual is computed via operator overloaded automatic differentiation [25]. The PTC algorithm computes an element local time step based on the characteristic speed, element size, and a Courant-Friedrichs-Lewy (CFL) number. The inverse CFL is driven toward zero such that a Newton-like convergence rate is recovered. The Portable, Extensible Toolkit for Scientific Computation (PETSc) [26-28] framework is used to solve the linear system for each PTC iteration with restarted generalized minimal residual (GMRES) [21] preconditioned with a Direct Lower Upper (LU) factorization. Parallel computations use the restricted additive Schwarz preconditioner with three layers of overlap. The LU preconditioner is applied to each subdomain, and restarted GMRES is applied to the global system. Adjoint systems are solved using the same linear solver as the primal.

D. Wolf

Wolf is a vertex-centered (flow variables are stored at vertices of the mesh), hybrid finite-volume and finite-element Navier-Stokes solver on unstructured meshes composed of triangles in 2D and tetrahedra in 3D. The convective terms are formed by the finite-volume method on the dual mesh composed of median cells. The HLLC approximate Riemann solver [29] computes the flux at the cell interface. Piecewise linear interpolation is based on the Monotonic Upwind Scheme for Conservation Law (MUSCL) procedure, which uses a particular edge-based formulation with upwind elements to achieve 2nd-order accuracy in space. A low dissipation scheme uses the combination of centered (edge) and upwind (element) gradients. A dedicated slope limiter is employed to dampen or eliminate spurious oscillations that may occur in the vicinity of discontinuities. The viscous terms are formed by the Continuous Galerkin (CG) method, which also provides 2nd-order accuracy.

The implicit temporal discretization considers the backward Euler time-integration scheme. At each time step, the linear system of equations is approximately solved using a Symmetric Gauss-Seidel (SGS) implicit solver and local time stepping to accelerate the convergence to steady state. A Newton method based on the SGS relaxation is very attractive, because it uses an edge-based data structure that can be efficiently parallelized. Empirically, the following *crucial* choices to solve the compressible Navier-Stokes equations have been made. The residual of the linear system is reduced by two orders of magnitude by SGS relaxation. Breadth-first search renumbering improves the convergence rate of the implicit method and increases overall efficiency. Fully differentiating the HLLC approximate Riemann solver and the CG viscous terms is very advantageous. Automatically adjusting the time step is required to achieve high efficiency, automation, and robustness in the solution of the nonlinear system of algebraic equations to steady-state. The SA turbulence model is loosely-coupled to the mean-flow equations, where the mean-flow and turbulence model equations are relaxed in an alternating sequence. Complete details of the Wolf flow solver are provided in [8,30].

IV. Mesh Mechanics

The following anisotropic mesh mechanics packages are used to modify the mesh to conform to a given metric field M. The goal is to create a unit mesh [8], where the edges are unit-length and the elements are unit-volume with respect to the given metric. These tools adopt a local modification approach, where a valid input mesh is transformed into a more metric conforming valid output mesh through a series of local operators. These tools differ in the local operations that are considered (e.g., element split, element collapse, element reconnection, vertex relocation) and the criteria used to apply these local operations.

A. AHA

AHA (Anisotropic Heuristic Adaptation) is an adaptive remeshing tool developed by Boeing for production 2D simulations, and is the only mesh mechanics tool considered here that does not use a metric field. Instead, a coarse initial triangular mesh serves as a framework for mesh refinement and coarsening. The initial mesh elements form macro triangles that are refined by inserting nodes on the triangle edges. Any number of nodes may be added and removed to each macro triangle edge and to the interior of each macro triangle. The original nodes and edges of the macro triangles remain in the mesh and form a limit on how coarse the mesh can become. New boundary nodes added to the airfoil surface are automatically projected to the geometry. Geometry is represented by splines through a provided discrete set of points. Maintaining the original macro element edges results in very large area ratios between neighboring triangles particularly where features are not aligned with the macro element boundaries.

B. EPIC

The EPIC (Edge Primitive Insertion and Collapse) anisotropic mesh adaptation package developed at Boeing provides a modular framework for anisotropic mesh adaptation that can be linked with external flow solvers [5]. EPIC relies on repeated application of edge break, edge collapse, element reconnection and vertex movement operations to modify a mesh such that element edge lengths match a given anisotropic metric tensor field. The metric field on the adapted mesh is continuously interpolated from the initial metric field. Several methods are available to preprocess the metric so as to limit minimum and maximum local metric sizes, control stretching rates of metric size and/or anisotropy, and ensure smoothness of the resulting distribution. In addition, the metric distribution can be limited relative to the initial mesh and/or to the local geometry surface curvature. The surface mesh is maintained on an IGES geometry definition with geometric projections and a local remeshing procedure.

C. FEFLO.A

FEFLO.A is a 2D, 3D, and surface mesh adaptation tool. It uses a combination of generalized standard operators (e.g., insertion, collapse, swap of edges and faces). The generalized operators are based on recasting the standard operators in a cavity framework [31,32]. The cavity operator allows a simultaneous application of multiple standard operator combinations. Quality improvements are attained with the cavity operator that are not possible through a sequential application of standard operators. To increase robustness, the surface and volume mesh are modified simultaneously and each local modification is checked to verify that a valid mesh is maintained. For the volume, validity consists of checking that each newly created element has a strictly positive volume. For the surface, validity is checked by ensuring that the deviation of the geometric approximation with respect to a reference surface mesh remains within a given tolerance. During surface remeshing, new vertex locations are either evaluated with a cubic surface representation or an EGADS [33] geometry query.

D. refine

The refine open source mesh adaptation mechanics package was developed by NASA. It is available via github.com/NASA/refine under the Apache License, Version 2.0. It is designed to output a unit mesh [8] with repsect to a provided metric field. The current version under development uses the combination of edge split and collapse operations proposed by Michal and Krakos [5]. Node relocation is performed to improve adjacent element shape. A new ideal node location of the node is created for each adjacent element. A convex combination of these ideal node locations is chosen to yield a new node location update that improves the element shape measure in the anisotropic metric [34]. Geometry is accessed through the EGADS [33] application program interface.

V. Metric Construction

A. AHA Heuristic Error Indicator

AHA constructs a heuristic error indicator E_t for each edge of the mesh based on a weighted combination of first and second derivatives of the Mach Number, M, along each mesh edge, e.

$$E_t = \left(w_1 \left| \frac{\partial M}{\partial \mathrm{e}} \right|^p + w_2 \left| \frac{\partial^2 M}{\partial \mathrm{e}^2} \right|^{p/2} \right) \; |\mathrm{e}| \tag{2}$$

The weights w_1 and w_2 are selected to guarantee a set fraction of edges are refined by the second derivative error term. Edges are sorted by the magnitude of E_t and a fraction of mesh edges are selected for refinement and coarsening during each adaptive step. Multiple sweeps of refinement and coarsening are performed.

B. Multiscale Metric

The multiscale metric controls the L^p-norm of the interpolation error of a solution scalar field [35]. The (reconstructed) Hessian, H, of a scalar field, is locally scaled by the Hessian determinant and globally scaled to a specified target to get the metric field,

$$\mathrm{M}_{LP}=D_{LP}\left(\det|H|\right)^{\frac{-1}{2p+d}}|H| \tag{3}$$

where the global scaling D_{LP} ,

$$D_{LP}=\left(\frac{C_t}{C\left(\det(H)^{\frac{-1}{2p+d}}|H|\right)}\right)^{2/d} \tag{4}$$

corrects the complexity of the locally scaled Hessian to produce $\mathrm{M}_{LP}$with specified target complexity C_t . Both scaling operations depend on the dimensionality of the domain, which is d = 2 in this case. A mesh conforming to M_{LP} provides optimal control of the scalar field interpolation error in the p-norm. A lower p-norm targets weaker variations of the scalar field and a larger p-norm targets rapid variations of the scalar field. A number of multiscale metric calculation methods are described below that have subtle differences in how the multiscale metric is formed (e.g., Hessian reconstruction, limits on metric gradation).

GGNS+EPIC Multiscale Metric

The Mach Hessian for each element is evaluated from the flow solution using a least-squares approach on an extended stencil in GGNS. GGNS then passes the Hessian at each element to EPIC, which converts it to adaptation metrics via an element-centered modification of Alauzet and Loseille [10], which minimizes the L^p-norm of interpolation error of the scalar field for a given mesh complexity. In this modification, each elemental Hessian is scaled to control the L^p-norm with Eq. (3). The global scaling factor, D_{LP}, is initialized as Eq. (4). When enabled, the metric gradation is limited as detailed in the EPIC description, Section IV.B. The complexity, Eq. (1), of the resulting elemental adaptive metric is computed and the global scale factor, D_{LP} , is adjusted to better match the requested value. The metric is then iteratively recomputed until the computed complexity is within a specified tolerance of the requested value. A continuous metric field is generated by Log-Euclidean [36] interpolation of the elemental metrics to the mesh vertices.

refine Multiscale Metric

To form the metric, a Hessian of the scalar field is reconstructed by a recursive application of L^2-projection [10]. The gradient is computed in each element and a volume-weighted average is collected at each vertex [10]. The 2nd-derivative Hessian terms are formed by computing the reconstructed gradients of the gradients formed in the first pass. The mixed derivative terms of the Hessian are averaged. A special boundary treatment is employed. The reconstructred Hessian on the boundary is replaced with an extrapolation from neighboring interior vertices, which have a well-formed stencil.

The reconstructed Hessian is then diagonalized into eigenvalues and eigenvectors. The absolute value of the Hessian is formed by recombining the absolute value of the eigenvalues with eigenvectors to ensure the Hessian is symmetric positive definite. The Hessian at each vertex is scaled to control the L^p-norm [10] with Eq. (3) The gradation of the metric field is limited isotropically in the metric space with the “metric-space-gradation” of [37]. The complexity is computed, and the metric is globally scaled to set its complexity to a specified value. The complexity Eq. (1) is evaluated discretely by assuming it is piece-wise constant in each median dual.

Wolf+FEFLO.A Multiscale Metric

Recursive L^2-projection [10] is used for Hessian reconstruction. This recursive L^2-projection method is preferred over other reconstruction methods for improved robustness, efficiency, and accuracy.

Alternate methods are available to compute the Hessian (i.e., double weighted least-square and k-exact weighted least square), but are not used in this study. In the presented results, the local Mach number is the scalar field and the multiscale metric [10] controls the interpolation error in the L^4-norm. A metric gradation process is applied to smooth the metric field following the “mixed-space-gradation” approach of [37], which is the most significant difference between the Wolf+FEFLO.A multiscale metric and the refine multiscale metric implementations.

C. Output-based Metric

The metric field can be formulated to target a specific goal or output of the simulation by using information from the dual or adjoint problem. Output-based mesh adaptation approaches are reviewed by Fidkowski and Darmofal [38] and a number of optimal-goal or output-based metric construction methods are examined here.

MOESS Output-based Metric

The Metric Optimization via Error Sampling and Synthesis (MOESS) [39,40] adaptation algorithm is based on the continuous mesh framework developed by Loseille and Alauzet [8,9]. Carson et al [41] details recent improvements to the MOESS algorithm and the implementation of MOESS in SANS. The output error estimation method used within MOESS is the Dual Weighted Residual (DWR) method, as originally devised by Becker and Rannacher [42]. DWR consists of computing an adjoint solution for an output functional in an enriched solution space, and weighting it against the residual to give an estimate of $\mathcal{J}(u) - \mathcal{J}(u_{h,p})$ for some output functional $\mathcal{J}(u)$. Originally devised for the CG discretization, this approach was extended by Carson et al. [43] and Huang et al. [23] for the VMSD discretization.

The MOESS algorithm constructs a set of elemental local models that approximate the change in a localized error estimate for an output of interest as a function of a step matrix change to the implied metric of the mesh. These local models are fit from *local solves* where an element $\varkappa$ is locally refined, either isotropically or by splitting an edge, and an approximate solve is performed. A *local solve* consists of fixing the DOF outside of a patch, $\omega_{\varkappa} \supseteq \varkappa$ whilst allowing those in the patch to vary. For a DG *local solve*, $\omega_{\varkappa} \equiv \varkappa$ but for a CG *local solve*, $\omega_{\varkappa}$ consists of the elements attached to the vertices of $\varkappa$. The localized error estimate is then reevaluated using the result of this *local solve*, which produces a change in the error estimate as a function of the step matrix change to the element. The *local solve* is repeated for multiple different refinements of the element and the results for all the local refinements are synthesized into a local error estimate model for that element. This is repeated for all the elements of the mesh and the sum of these local models is optimized subject to a maximum DOF constraint as well as constraints on the magnitude of the step matrices. The optimized step matrices are applied to the implied metric of the current mesh to create the new metric request that is supplied to a mesh adaptation tool.

GGNS+EPIC Output-based Metric

The GGNS+EPIC output metric was introduced by Michal et al. [12]. For the evaluation of the local error estimate, piecewise linear reconstruction is used to represent the adjoint solution and to obtain its gradient. For the 2nd-order derivatives, Zienkiewicz-Zhu type patch recovery reconstruction for the gradients [44] is first obtained at the vertices of the mesh and then these are linearly interpolated inside the elements. The interpolation error for the primal solution is not evaluated directly, instead, we rely on a semiheuristic approach when the Hessian of the Mach number field is provided to the adaptation module. The Mach field is the only source of metric anisotropy in the present approach; in particular, any information about the adjoint solution enters the error indicator through the isotropic weight. To evaluate the Hessian of the primal solution, the k-exact least-square approach based on an extended stencil reconstruction is involved resulting in the quadratic reconstruction. The local error indicator along with the described implementation has strong similarities to the well known Venditti–Darmofal [45] approach but is not exactly equivalent to the latter.

Wolf+FEFLO.A Output-based Metric

A goal-oriented error estimate based on an a priori error analysis has initially been proposed for the inviscid Euler equations in [8]. The main idea was to translate the error on the considered output functional into a weighted interpolation error estimate. Weights are given by derivatives of the adjoint state and interpolation errors are on the Euler fluxes. As we are left with weighted interpolation errors, we can use the continuous mesh framework [8,9] to obtain an analytical expression of the optimal metric field. An extension of this goal-oriented error estimate has been proposed for the laminar Navier-Stokes equations in [46]. The main advantage of these error estimates in comparison to other goal-oriented error estimates is that the anisotropy of the mesh appears naturally.

From the analysis of the behavior several error estimates for the Reynolds-averaged Navier-Stokes equations [12], we devised the following new error estimate for RANS using integration by part and linearization [47].

$$
\begin{aligned}
\|J(W)-J(W_h)\|_{L^1(\Omega_h)} &\approx \int_\Omega |W^*|\,\left|\nabla\cdot(\mathcal{F}^E(W)-\mathcal{F}^E(\Pi_h W))-\nabla\cdot(\mathcal{F}^V(W)-\mathcal{F}^V(\Pi_h W))\right|\,\mathrm{d}\Omega \\
&\leq \int_\Omega \left(|\nabla W^*|\,\left|\mathcal{F}^E(W)-\Pi_h\mathcal{F}^E(W)\right|+|W^*|\,\left|\sum_i \frac{\partial S_i}{\partial x_i}(W)-\frac{\partial S_i}{\partial x_i}(\Pi_h W)\right|\right)\mathrm{d}\Omega \qquad (5)\\
&\approx \int_\Omega \left|\sum_i \frac{\partial F_i}{\partial W}\cdot\nabla_{x_i}W^*+K_{ij}\,H(W^*)\right|\cdot|W-\Pi_h W|\,\mathrm{d}\Omega
\end{aligned}
$$

where J is the considered output functional, W is the conservative variables vector, W^* the associated adjoint state, $\mathcal{F}^E=(F_i)_i$ the convective fluxes, $\mathcal{F}^V=(S_i)_i$ the viscous fluxes, K_{ij} the viscous terms under matrix form and $H(W^*)$ the hessian of the adjoint state. The error estimate is a weighted sum of L^1 interpolation error on the conservative variables where the weights depend on the gradient and the hessian of the adjoint state and on the convective and viscous fluxes. Therefore, we can directly apply the continuous mesh framework to obtain an analytical expression of the optimal metric field.

VI. Flow Solver Verification

All simulations in this paper were performed using the Spalart Almaras turbulence model. The turbulence model implementation was verified by comparing adapted mesh convergence for each flow solver with published verification results from the Turbulence Modeling Resource (TMR) sponsored by the AIAA Fluid Dynamics Technical Committee Turbulence Model Benchmarking Working Group [7]. The verification case chosen for this study is the 2D NACA0012 Airfoil. Flow conditions are: Mach 0.15; Reynolds number of 6 Million; 10 degrees angle of attack and a freestream reference temperature of 540 degrees Rankine. Solutions were computed on sequences of adapted meshes with increasing resolution with the intent of predicting the mesh converged result. In Figure 3, drag convergence from the GGNS, SANS, SFE and Wolf flow solvers is compared with fixed-mesh results from the TMR site. The x-axis represents an average element length scale (h) given by 1 over the square root of the number of degrees of freedom (N). All solvers are converging to a computed drag coefficient very near the expected value of 0.1227.

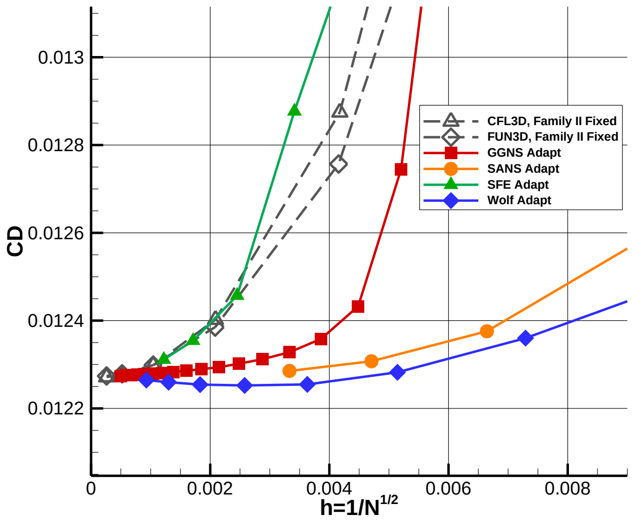

Figure 3: Turbulence Model Verification Results for NACA 0012 Test Case

VII. Results

The flow simulations performed for this study were consistent with the geometry and flow conditions used for the AIAA special session on Mesh Effects for CFD Solutions [48] sponsored by the AIAA Geometry and Mesh Generation Workshop (GMGW) [49]. The geometry model is shown in Figure 4 and consists of a three element high lift airfoil derived from the NASA High Lift Common Research Model [50]. The model is available for download on the GMGW special session website [48].

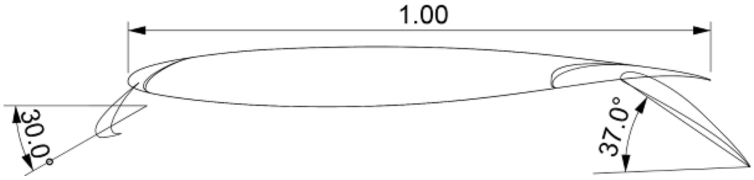

Figure 4: HL-CRM Airfoil Geometry

Solutions about the 2D HL-CRM airfoil were generated at standard sea level conditions with Mach number of 0.2, and Reynolds Number of 5.0 million based on the reference chord and angles of attack from -2.0 to 26 degrees. All flow solutions were computed with the Spalart Allmaras turbulence model. Each solver determined solution converged based on it's own criteria. GGNS solution convergence was measure by a drop in the dependent variable residuals to a level of 1.0e-09. FUN3D-SFE solutions are converged to a machine precision (i.e. 1.0e-15). The DOF averaged L^2-norm in the residuals was converged below a tolerance of 8.0e-12 for all computations using SANS. Wolf solution convergence was considered achieved when the averaged L^2-norm in the residual of the conservative variables was below a tolerance of 1.e-8.

An adaptive simulation consists of a series of iterations with each iteration compromising the following steps: flow solution; error estimation/metric construction; mesh generation; and solution interpolation. For this study, adaptive simulations were performed with several combinations of flow solver, mesh mechanics and error estimation methods. Each combination will be referred to as an adaptive method and is given a unique identifier shown in Table 1. These identifiers will be referenced in the presentation of results.

Table 1: Adaptive Mesh Methods

Results Identifier	Flow Solver	Mesh Mechanics	Error Estimate	Solution Continuation
GGNS-AHA	GGNS	AHA	AHA Heuristic with corrector	Complexity Continuation
GGNS-MS	GGNS	EPIC	GGNS+EPIC multiscale	Complexity Continuation
GGNS-Lift	GGNS	EPIC	GGNS+EPIC output-based (lift output)	Complexity Continuation
SANS-MOESS EPIC	SANS	EPIC	MOESS output-based (lift output)	Complexity Continuation
SANS-MOESS refine	SANS	refine	MOESS output-based (lift output)	Complexity Continuation
SFE-MS	SFE	refine	refine multiscale	Complexity Continuation
Wolf-MS-Cmplx	Wolf	FEFLO.A	Wolf+FEFLOA.A multiscale	Complexity Continuation
Wolf-Lift-Cmplx	Wolf	FEFLO.A	Wolf+FEFLOA.A output-based (lift output)	Complexity Continuation
Wolf-MS-Alpha	Wolf	FEFLO.A	Wolf+FEFLO.A multiscale	Alpha Continuation
Wolf-Lift-Alpha	Wolf	FEFLO.A	Goal Oriented output-based (lift output)	Alpha Continuation

The size of the mesh at each iteration is set by a target complexity specified by the adaptive method. The GGNS-AHA method increased target complexity by a percentage with each iteration. All of the other adaptive methods specified complexity steps doubling the complexity with each step. Several iterations (meshes) were performed at a fixed complexity before increasing the complexity for the next step. A comparison of the mesh complexity schedules for each of the adaptive methods is shown in Figure 5. There was a wide variation in the number of iterations performed at each step. The impact of complexity schedule on mesh convergence trajectory will be investigated in the results section.

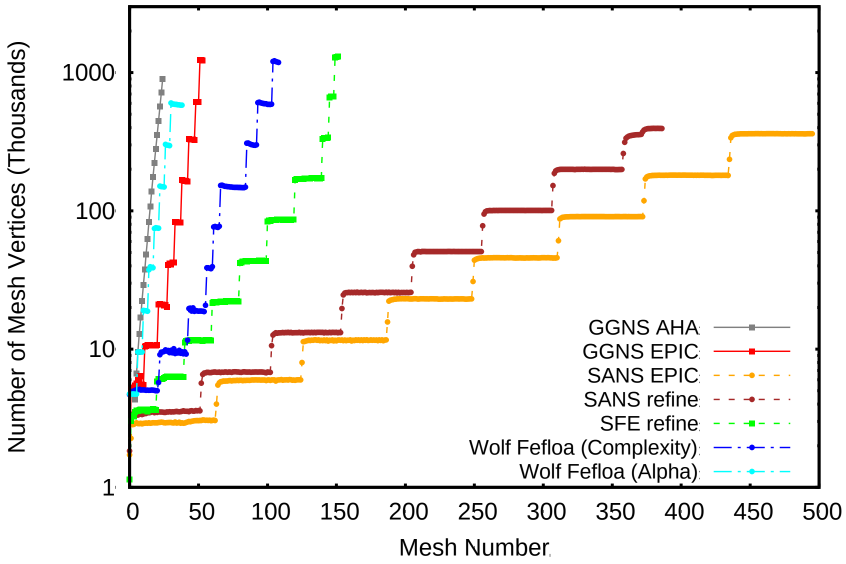

Figure 5: Adapted Mesh Size Sequence for Each Adaptive Method, 8 Degrees Angle of Attack

The adaptive methods that began the solution process at each angle of attack on a coarse mesh from freestream conditions are refered to as complexity continuation. Multiple steady state solutions to the governing equations can exist particularly at high angles of attack. Hysteresis effects can impact which solution the flow solver convergences to. To investigate this effect a series of adaptive simulations were performed with the Wolf/FEFLO.A codes where the adaptive simulation at each angle of attack and each mesh level was initiated from a solution at the previous angle of attack. These simulations are referred to as alpha continuation. For angles of attack 20 and above the solutions generated with SANS started with the coarsest adapted mesh/solution from the previous angle of attack as the initial mesh/solution.

In the following sections, detailed comparisons of mesh convergence, integrated forces, flow solutions, and adaptive mesh properties are presented for simulations at 8 and 16 degrees angle of attack. Mesh converged force and moment properties for the full angle of attack range are also presented.

A. 8 Degrees Angle of Attack Results

Simulations at 8 degrees angle of attack were used to compare mesh convergence properties of each method. The adaptation process was initiated on very coarse initial meshes generated independently for each method. The mesh complexity schedule for each method is shown in Figure 5. Trajectories of the lift, drag, and pitching moment convergence with mesh resolution is shown in Figures 6 through 10. Drag convergence is broken into pressure and viscous components in Figure 8 and 9.

In these Figures, the x-axis represents an average element length scale (h) given by one over the square root of the number of degrees of freedom (N). N is equal to the number of mesh nodes for all of the flow solvers in this study. Two scales for the mesh resolution are shown with the left plot showing the adaptation sequence for mesh resolutions above 2,500 nodes (h=0.02) and the right plot focusing on mesh resolution greater 40,000 nodes (h=0.005). Different line colors and line types are used for each flow solver/mesh mechanics combination. Symbol type represents the class of error estimate, with square symbols representing feature-based methods and circle symbols representing output-based methods. Small symbols are used to represent intermediate meshes at each complexity step with a large symbol representing the "optimized" result for a given complexity. The optimized value is simply taken as the final value in the fixed-complexity sequence for all methods except for the methods using the SANS flow solver. For these methods, the final 10 meshes at each complexity level are averaged for the "optimized" result.

A summary of the computed forces and moments on the final adapted mesh for each method is shown in Table 2. The total drag and viscous drag component, where available, are reported in separate columns. The viscous drag component is not reported by the AHA flow solver and hence that entry in the table is marked as NA. Averaging the integrated forces across all the methods results in average lift, drag and moment coefficient values of 3.2562, 0.03845 and -0.5282 with standard deviations of only 2.97e-3, 1.8e-4 and 5.9e-04. Despite the use of different flow solvers, mesh mechanics, error estimates, and adaptive mesh processes, the adaptive simulations approach remarkably similar mesh converged results.

Table 2: Computed Force and Moment Values on Finest Adapted Meshes, 8 Degrees Angle of Attack

Method	C_L	C_D	C_{DV}	C_m
GGNS-AHA	3.25515	0.038427	NA	-0.52795
GGNS-MS	3.25490	0.038473	0.098820	-0.52792
GGNS-Lift	3.25711	0.038404	0.009879	-0.52838
SANS-MOESS EPIC	3.25895	0.038422	0.009878	-0.52881
SANS-MOESS refine	3.25841	0.038428	0.009878	-0.52870
SFE-MS	3.25844	0.038409	0.009887	-0.52869
Wolf-Lift-Alpha	3.25401	0.038319	0.0098710	-0.52762
Wolf-MS-Alpha	3.25910	0.038260	0.0099154	-0.52876
Wolf-Lift -Cmplx	3.25668	0.038384	0.0098650	-0.52817
Wolf-MS-Cmplx	3.24949	0.038929	0.0098724	-0.52698
Average	3.25622	0.038445	0.0098809	-0.52820
Std Deviation	0.00297	0.000180	0.0000144	0.00059

One way to compare the efficiency of adaptive methods is to measure how quickly the output forces approach the final mesh converged value. As expected, the adaptation methods using output-based error estimates approach the mesh converged results faster than feature-based methods. As an example, at a mesh resolution of only 10,000 points ($1/N^{1/2}$ = 0.01) the average computed lift of the output-based methods is within 1.7 percent of the mesh converged value of 3.2572 while the average of the feature-base methods is only within 14 percent. The feature-based methods eventually approach the same mesh converged forces and moments but require much higher mesh density to achieve the same error levels. Similar behavior is seen in the drag and moment plots in Figure 7 and 10. Among the output-based error methods, the MOESS methods approach the mesh converged results the fastest. The advantage is most notable on the coarsest mesh levels. At tighter error levels the difference between the output-error methods becomes less noticeable. For instance at a mesh resolution of 63,000 nodes all the output-based methods have a similar drag error of ~ 0.7% or 0.0003 (3 counts). The SANS-MOESS adaptive process used a much larger number of meshes for each complexity level compared to the other methods (Figure 5). This may have improved the optimal value for the MOESS results particularly at the coarse complexity levels. The MOESS local error estimate model incorporates multi-directional information differently from the output error estimates used by GGNS and Wolf which also may have contributed to the better convergence properties on coarse meshes.

The impact of restarting adaptive solutions from the previous mesh and interpolated solution (complexity continuation) vs starting adaptive solutions from the mesh at a previous angle of attack (alpha continuation) can be seen in the Wolf-MS-Cmplx (blue) versus Wolf-MS-Alpha (cyan) lines. While both methods approach similar mesh converged answers, alpha continuation has a slight mesh convergence efficiency advantage for the MS metric. Examining the Wolf output based results (Wolf-Lift-Cmplx vs Wolf-Lift-Alpha) shows that alpha continuation at 8 degrees angle of attack has little or no advantage when adapting to an output-based metric. At angles of attack close to stall alpha continuation provided a clear benefit for both MS and output-based metrics.

Streamlines from the GGNS-Lift solution on the final adapted mesh are shown in Figure 11. Several important features of the flow are visible including the separation in the slat and wing flap coves. These features correspond well with increased mesh density in the adapted meshes. A small separation region is present on the upper flap trailing edge. This separation region introduces a non-linearity in the lift curve slope very near 8 degrees angle of attack. Resolution of the flap trailing edge separation was found to be critical to accurately predict the solution at 8 degrees angle of attack. The feature-based adaptive methods required much higher DOF's to adequately resolve this feature compared to the output-based methods. This is one reason for the better performance of the output-based methods at 8 degrees, while the feature and output-based method trajectories were much closer at other angles of attack.

Images of the adapted meshes for each method at a medium mesh resolution of approximately 80,000 nodes are shown in Figure 12 through 19. All of the meshes do a good job of resolving important flow features such as the attached and off-body shear layers, suction peaks on the upper surfaces, gaps between the slat/wing and wing/flap and the stagnation regions on the slat wing and flap. The feature-based methods spend a large number of DOF's resolving the wakes of the slat, wing and flap even far downstream of the geometry where they have little impact on the prediction of forces and moments. The output-based methods resolve the wakes where they interact with downstream geometry but forgo refinement far downstream of the airfoil. Another feature evident in the output error meshes that is missing in the feature-based meshes is refinement of the stagnation streamlines for each of the airfoil elements. This feature is most notable in the MOESS and GGNS-Lift meshes and to a lesser extent in the Wolf-Lift meshes. This extra resolution is driven by the importance of the stagnation streamline location in accurately predicting the circulation and by resolution requirements for accurate computation of the adjoint solution.

Comparing the adapted meshes with traditional fixed meshes shows that traditional mesh characteristics thought to be critical to the analysis such as geometric measures of element quality, and element stretching rates are less important than placing the DOF's where they best resolve the critical features of the flow. For instance the AHA adapted meshes shown in Figure 12 have very high area ratios between neighboring elements and have elements with dihedral angles near 180 degrees. Yet, when combined with a strong solver, converged solutions that are very accurate are possible on these meshes.

Surface pressure and skin friction coefficients on the final adapted mesh for each of the adaptive methods are compared in Figure 20 and 21. The similarity of the mesh converged solutions for all methods is evident in these plots. There is a slight variation in the surface pressure for the Wolf-MS result compared to the other methods. This solution has the slowest grid convergence and is underresolved even on the finest mesh scale. Other than the Wolf-MS solution, the differences in pressure coefficient are nearly imperceptible at the plotting resolution. With the exception of the stagnation and trailing edges the predicted skin friction is also very similar for all methods. In Figure 22, the normal spacing of the mesh at the wall and the computed local y^+ for the first point off the wall is plotted. Because of the highly anisotropic nature of the tetrahedral meshes near the wall, the scatter in these plots is quite noticeable. However, the mean value as you move along the wing surface provides valuable information about the near wall spacing in the meshes. Surprisingly there is a large variation between methods in the near wall resolution varying from y^+ values of less than 1 for the AHA metric to values around 5 for the MOESS metric methods. All of these methods provide similar accuracy in the lift and drag computations suggesting the normal spacing could be relaxed in some of the methods to improve efficiency. This also suggests that traditional mesh guidelines for wall normal spacing that typically require a wall spacing that provides a $y^+ =1$ may be too stringent.

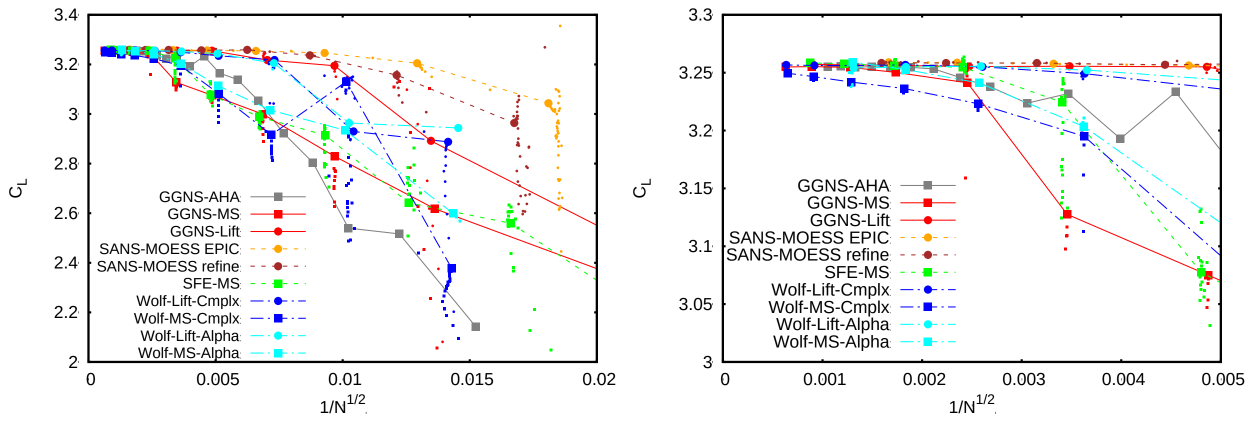

Figure 6: Lift Convergence, 8 Degrees Angle of Attack

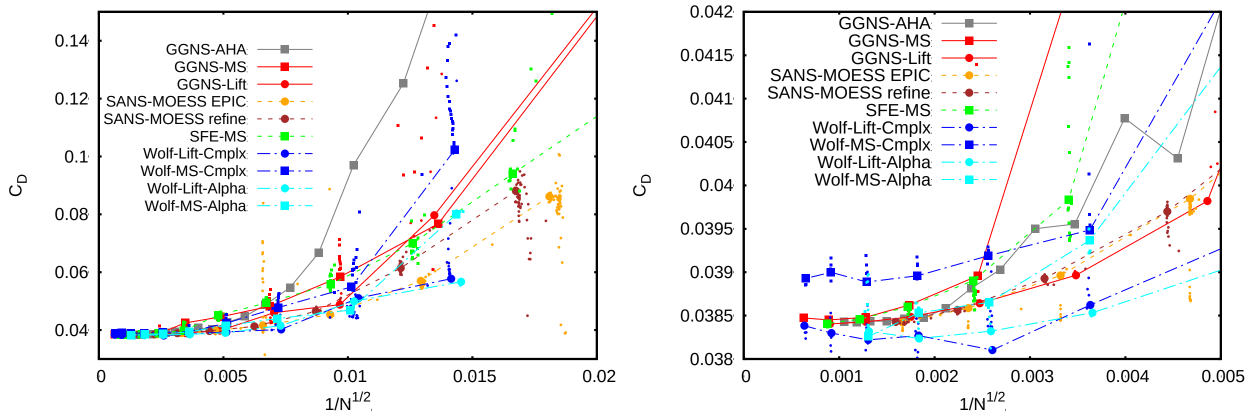

Figure 7: Drag Convergence, 8 Degrees Angle of Attack

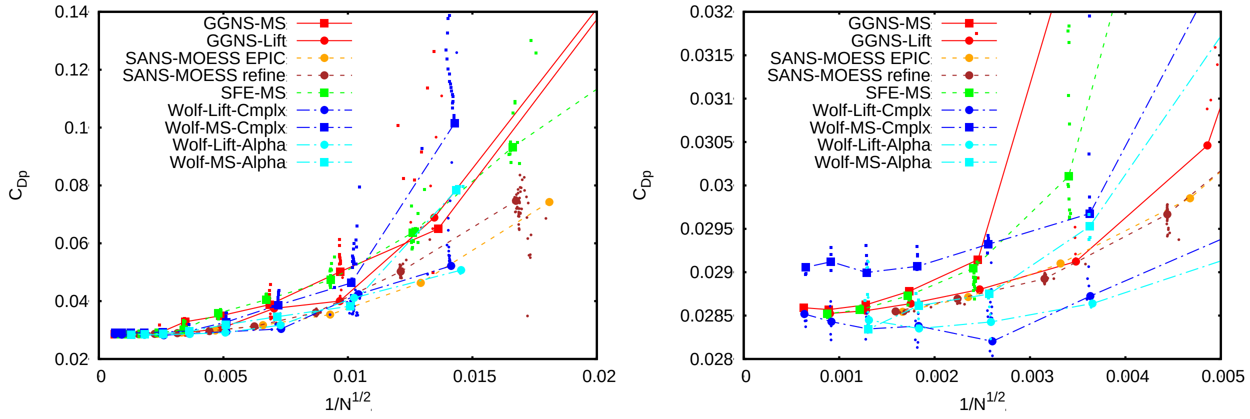

Figure 8 Pressure Drag Convergence, 8 Degrees Angle of Attack

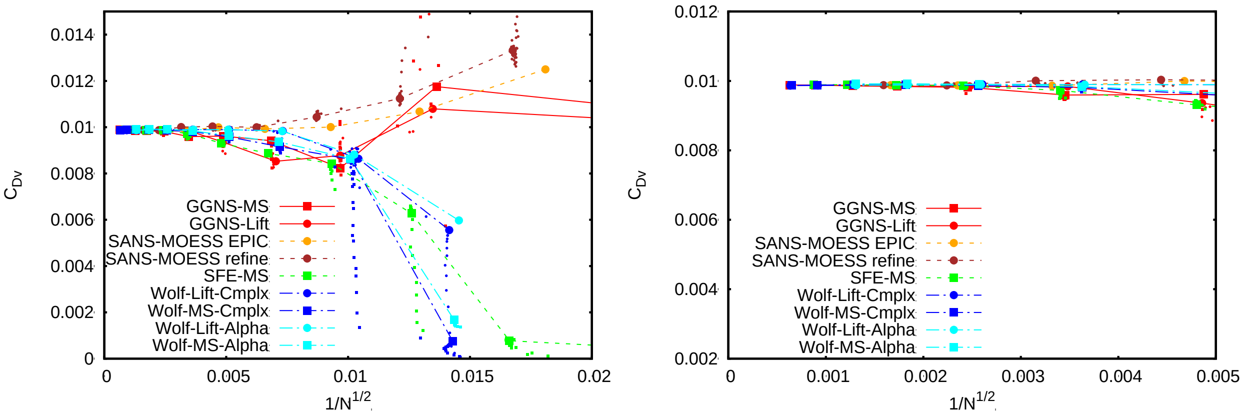

Figure 9: Viscous Drag Convergence, 8 Degrees Angle of Attack

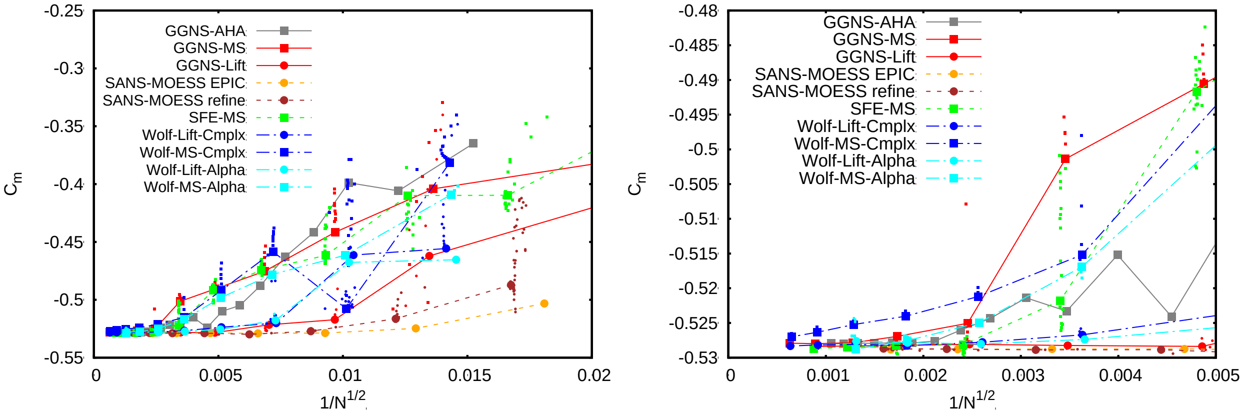

Figure 10: Pitching Moment Convergence, 8 degrees Angle of Attack

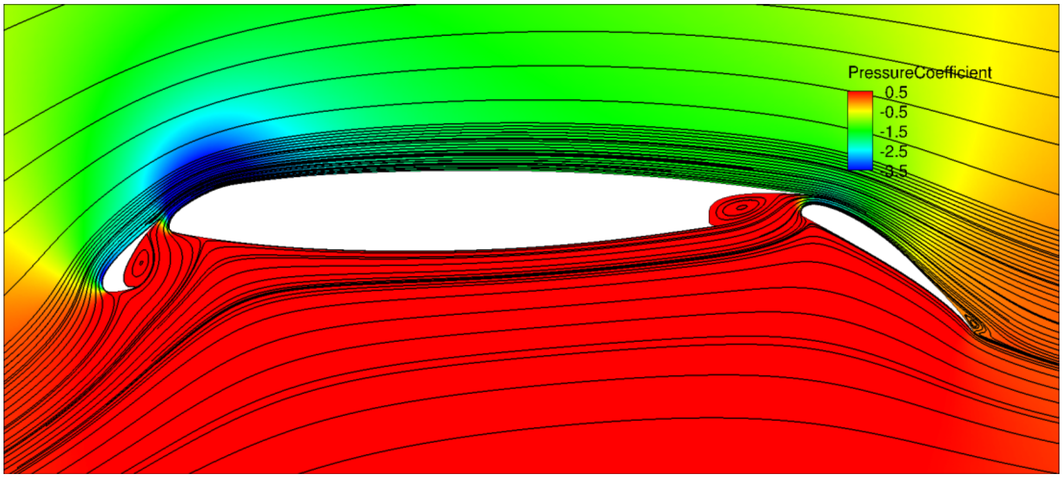

Figure 11: Streamlines from GGNS-Lift Adapted Mesh Solution, 8 Degrees Angle of Attack.

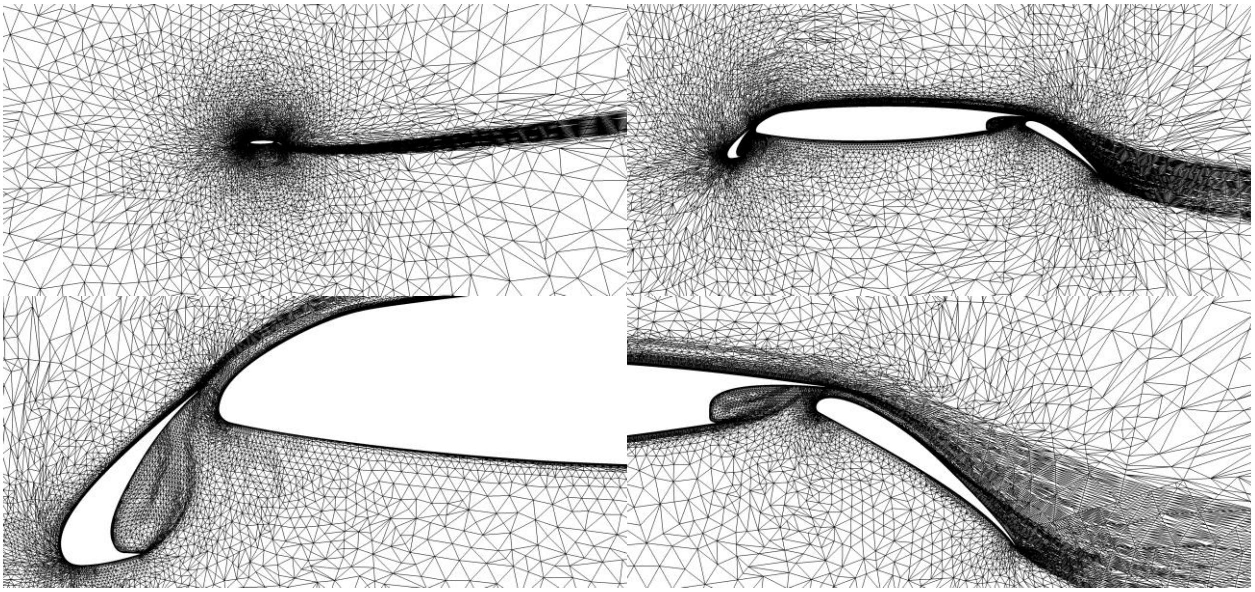

Figure 12: GGNS-AHA Medium Adapted Mesh, 81214 Nodes, 8 Degrees Angle of Attack

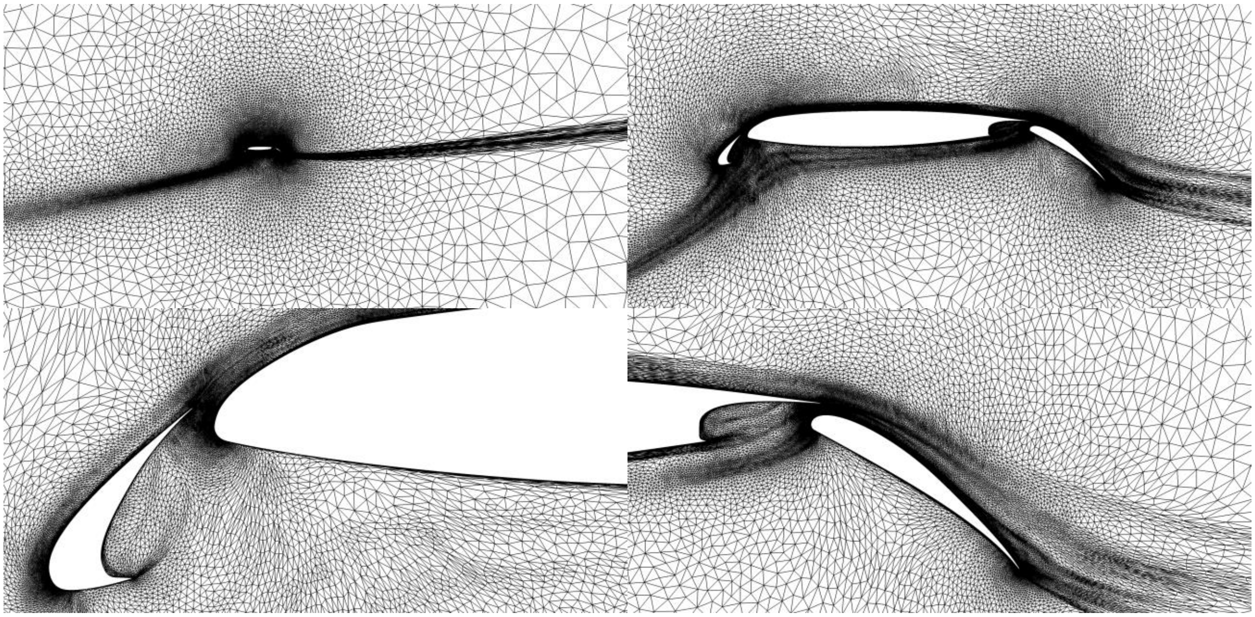

Figure 13: GGNS-Lift Medium Adapted Mesh, 82547 Nodes, 8 Degrees Angle of Attack

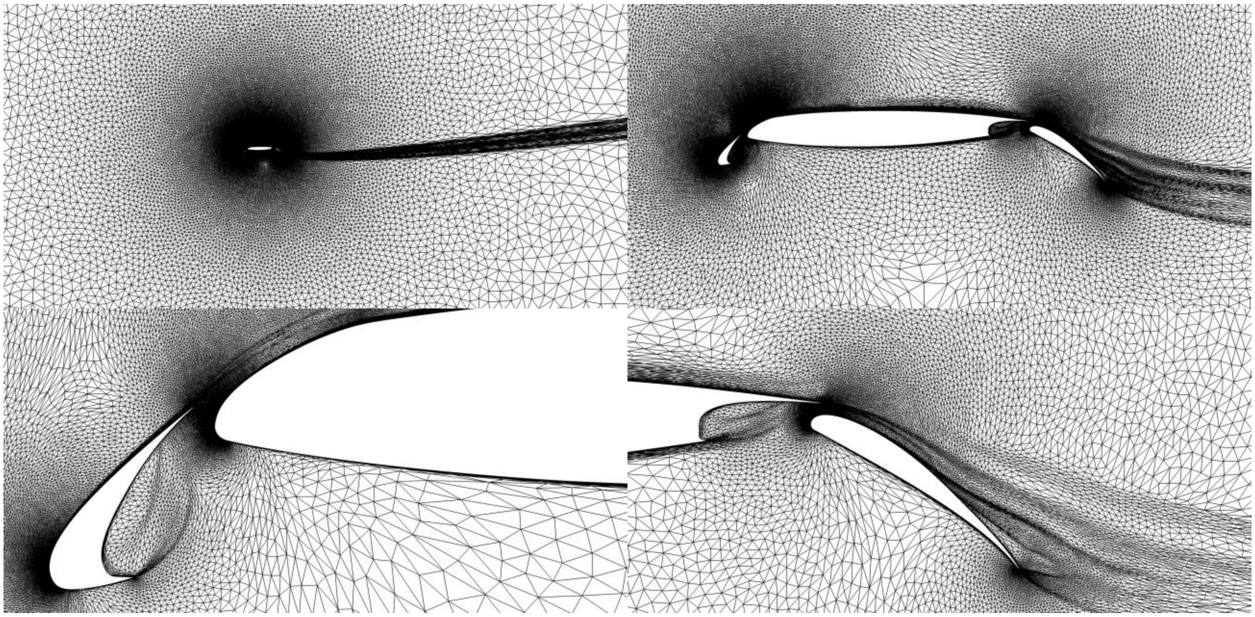

Figure 14: GGNS-MS Medium Adapted Mesh, 83671 Nodes, 8 Degrees Angle of Attack

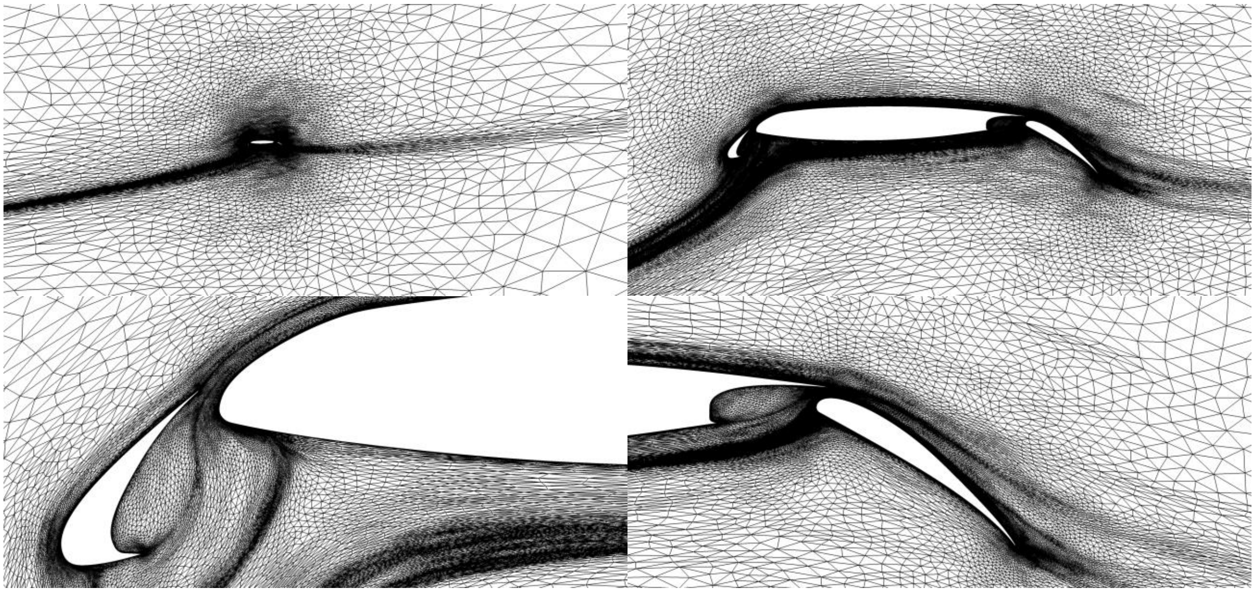

Figure 15: SANS-MOESS EPIC Medium Adapted Mesh, 90819 Nodes, 8 Degrees Angle of Attack

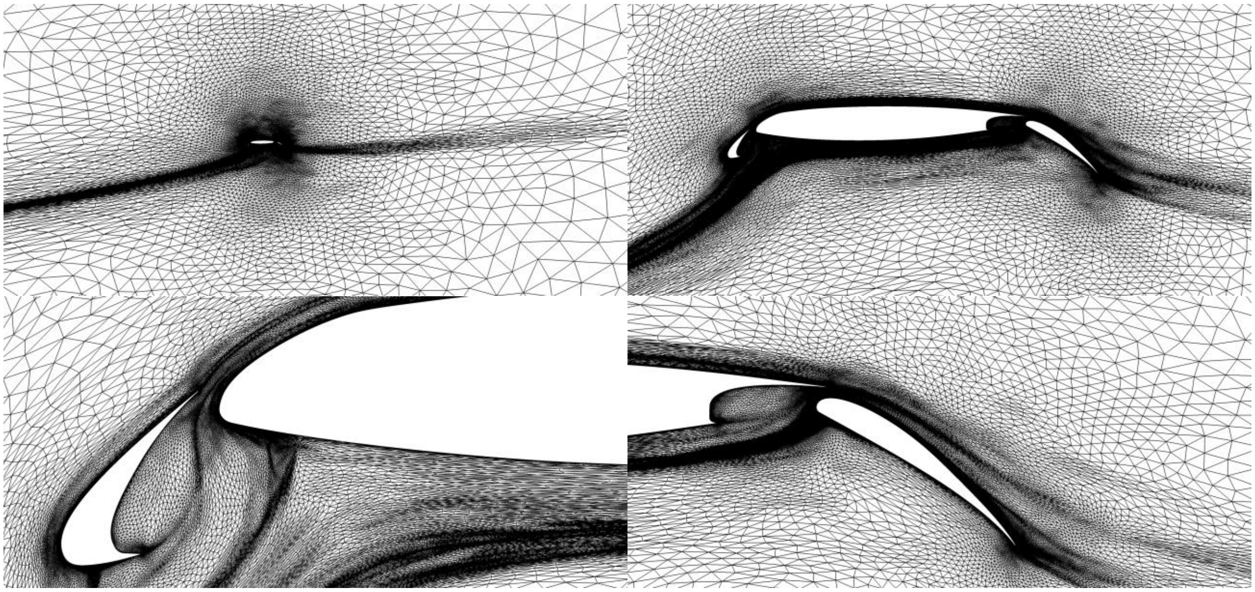

Figure 16: SANS-MOESS refine Medium Adapted Mesh, 100628 Nodes, 8 Degrees Angle of Attack

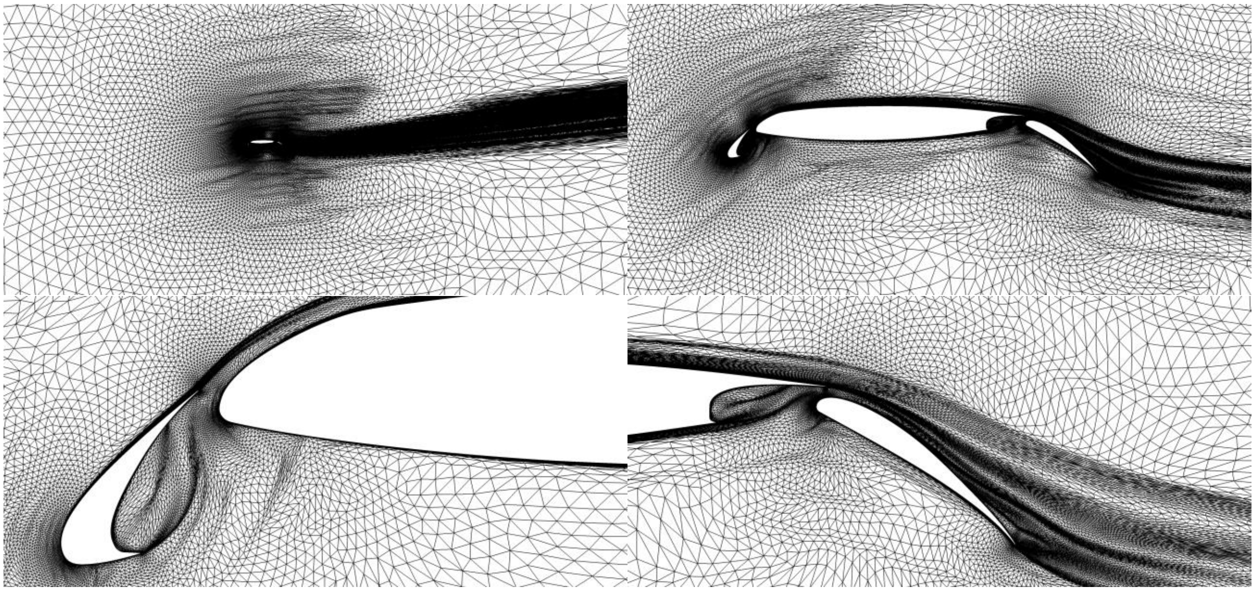

Figure 17: SFE-MS Medium Adapted Mesh, 85976 Nodes, 8 Degrees Angle of Attack

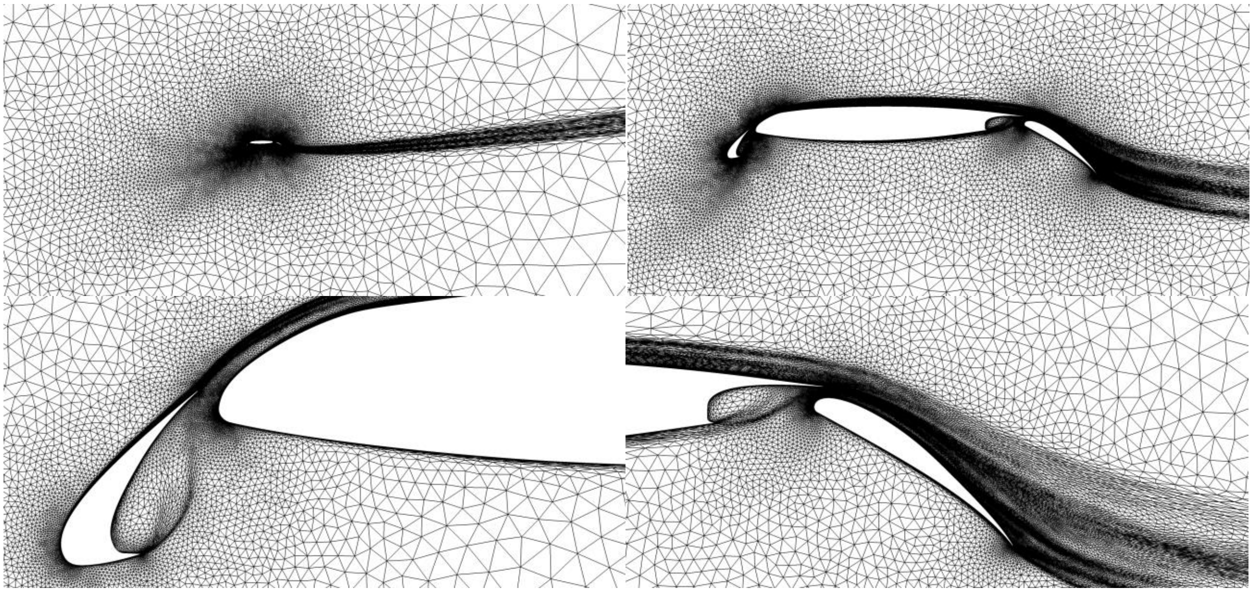

Figure 18: Wolf-Lift Medium Adapted Mesh, 75130 Nodes, 8 Degrees Angle of Attack

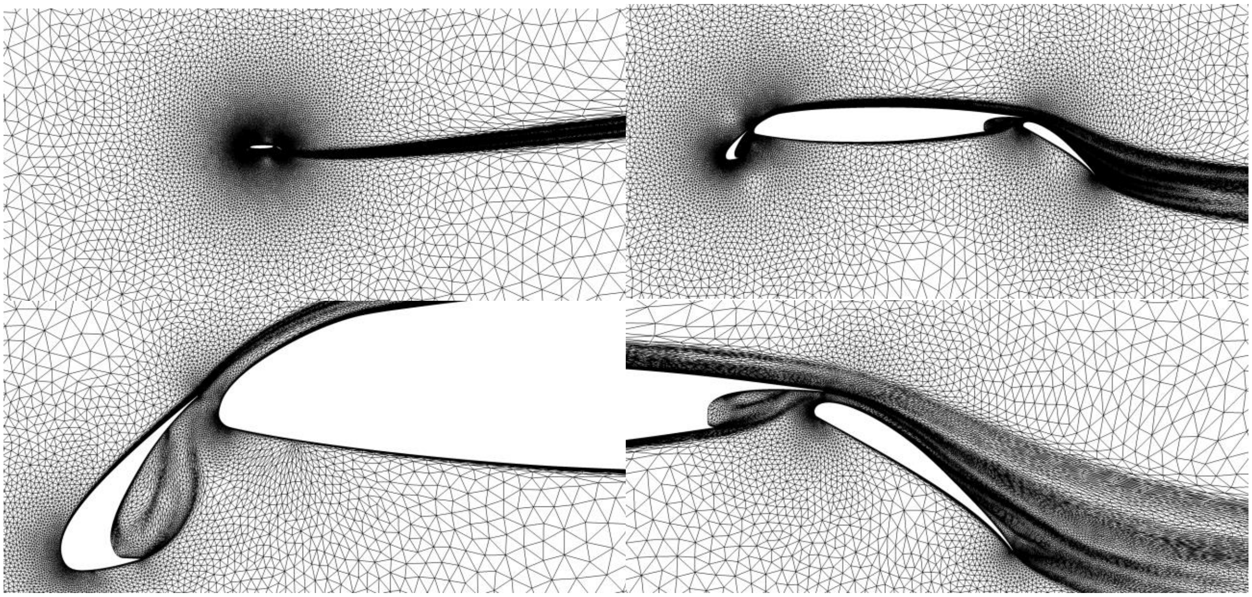

Figure 19: Wolf-MS Medium Adapted Mesh, 76853 Nodes, 8 Degrees Angle of Attack

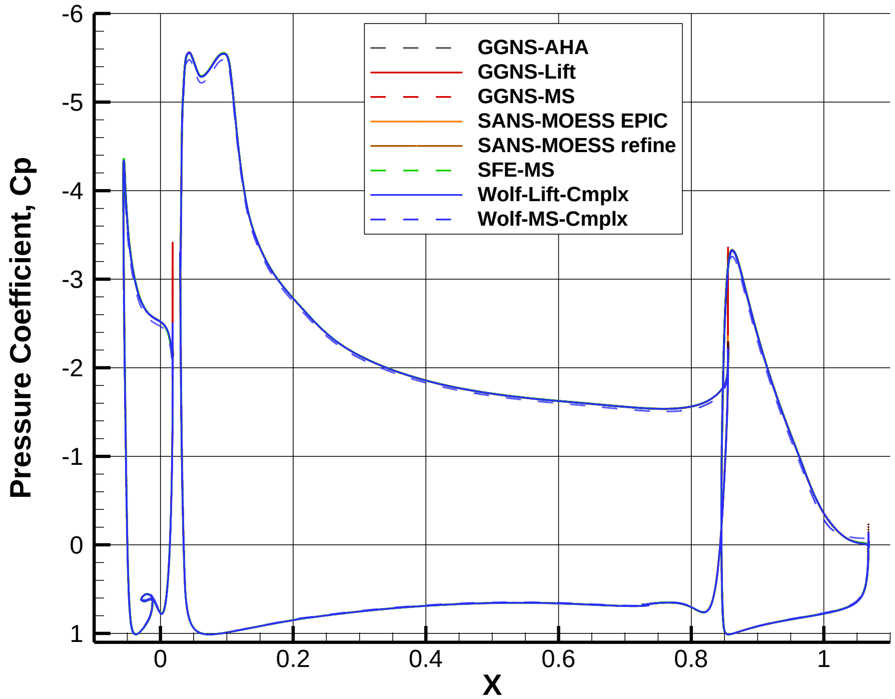

Figure 20: Surface Pressure Coefficient on Finest Adapted Meshes, 8 Degrees Angle of Attack.

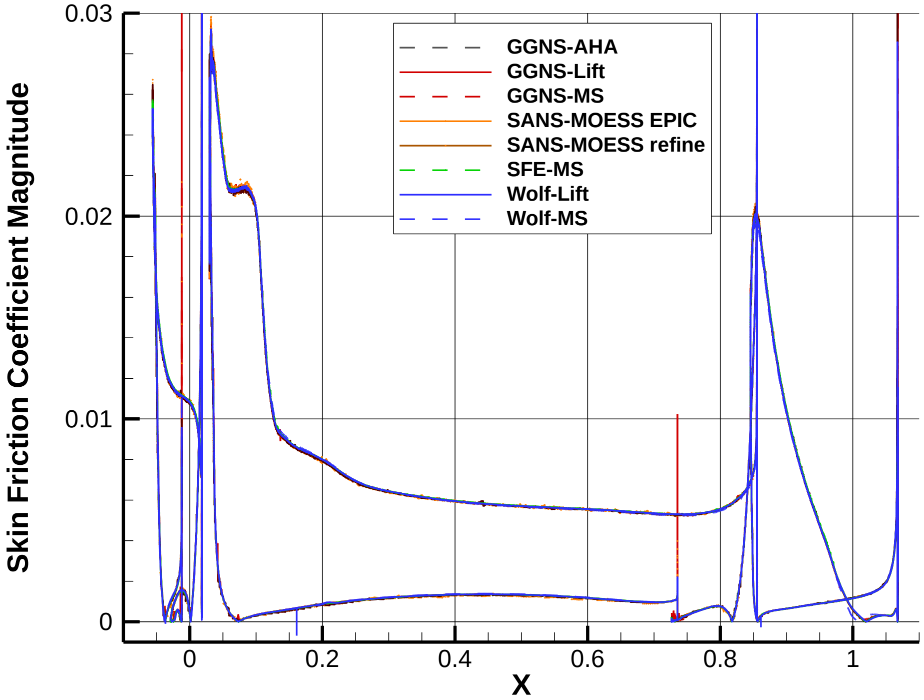

Figure 21: Skin Friction Magnitude on Finest Meshes, 8 Degrees Angle of Attack

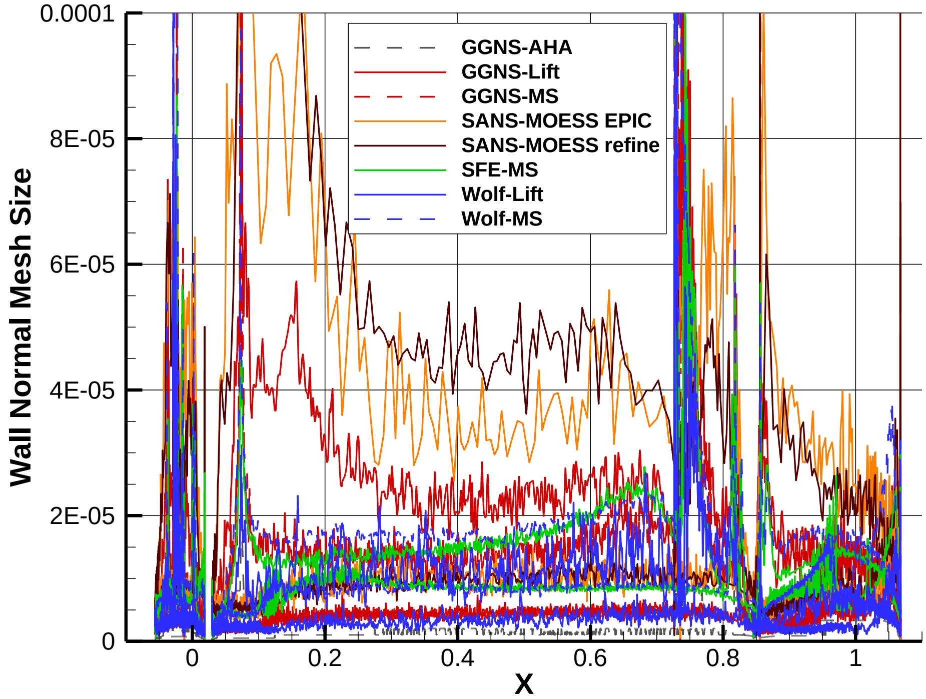

Figure 22: Wall Normal Spacing and Y$^+$ Comparison on Finest Adapted Meshes, 8 Degrees Angle of Attack

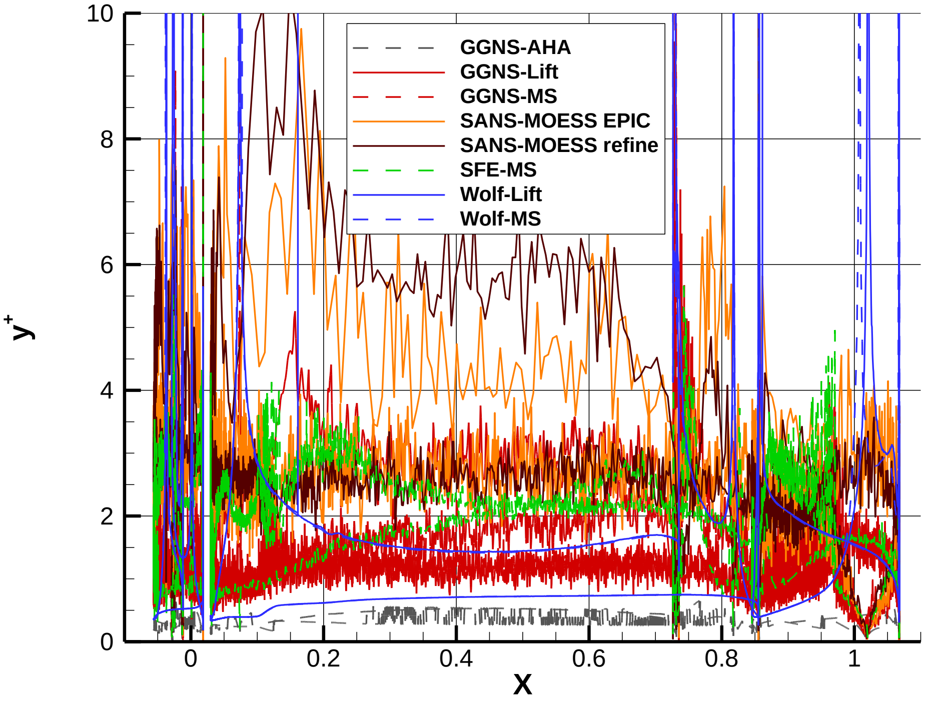

Figure 23: Y+ on Finest Adapted Meshes, 8 Degrees Angle of Attack

B. *16 Degrees Angle of Attack Results*

A second comparison of the methods was performed on the HLCRM airfoil at 16 degrees angle of attack. As before, there was no attempt to coordinate the initial mesh or adaptive mesh size sequence between the methods. Predicted lift, drag and pitching moment convergence with mesh resolution is compared for the various methods in Figure 24 through 28. Once again there is a clear separation between the trajectories of the output-based and feature-based approaches. The output-error estimates are much more efficient at reducing the error on coarser meshes.

A summary of the computed forces and moments on the final adapted mesh for each method is shown in Table 3. All of the adaptive simulations approach very similar mesh converged values for lift, drag and pitching moment. The average computed value for lift drag and moment coefficient is 3.7989, 0.06078 and -0.3824 with standard deviations of 4.1e-3, 1.4e-4 and 7.6e-4.

Table 3: Computed Force and Moment Values on Finest Adapted Meshes, 16 Degrees Angle of Attack

Method	C_L	C_D	C_{Dv}	C_m
GGNS-AHA	3.80197	0.060617	NA	-0.38292
GGNS-MS	3.80170	0.060654	0.009391	-0.38290
GGNS-Lift	3.80195	0.060673	0.009375	-0.38296
SANS-MOESS EPIC	3.80284	0.060718	0.009376	-0.38314
SANS-MOESS refine	3.80307	0.060898	0.009371	-0.38329
SFE-MS	3.80038	0.060841	0.009370	-0.38280
Wolf-Lift-Alpha	3.79564	0.060603	0.009336	-0.38172
Wolf-MS-Alpha	3.79254	0.060980	0.009388	-0.38129
Wolf-Lift-Cmplx	3.79601	0.060825	0.009343	-0.38195
Wolf-MS-Cmplx	3.79323	0.060975	0.009366	-0.38142
Average	3.79893	0.060778	0.009368	-0.38244
Std Deviation	0.00413	0.000144	0.000018	0.00076

Streamlines from the GGNS-Lift solution on the final adapted mesh are shown in Figure 29. Images of the adapted meshes for each method at a medium resolution around 80,000 nodes are shown in Figure 30 and 36. Important features such as the wakes and stagnation streamlines have shifted from their location in the 8 degree angle of attack simulations. These features are well resolved in all the adapted meshes with the output-based methods avoiding the wasted DOF's in the downstream wakes far from the airfoil. Once again, the stagnation streamlines ahead of each element are very noticeably refined in the MOESS and GGNS-Lift adapted meshes.

Surface pressure and skin friction magnitude on the final adapted mesh for each of the adaptive methods are compared in Figure 37 and 38. Differences in the surface pressure and skin friction are nearly imperceptible. The normal mesh spacing and corresponding y^+ values are shown in Figure 39 and 40. These are similar to the 8 degree angle of attack results. One slight difference is that the MOESS results have a slightly smaller normal wall spacing of y^+~3 compared to y^+~5 in the earlier results.

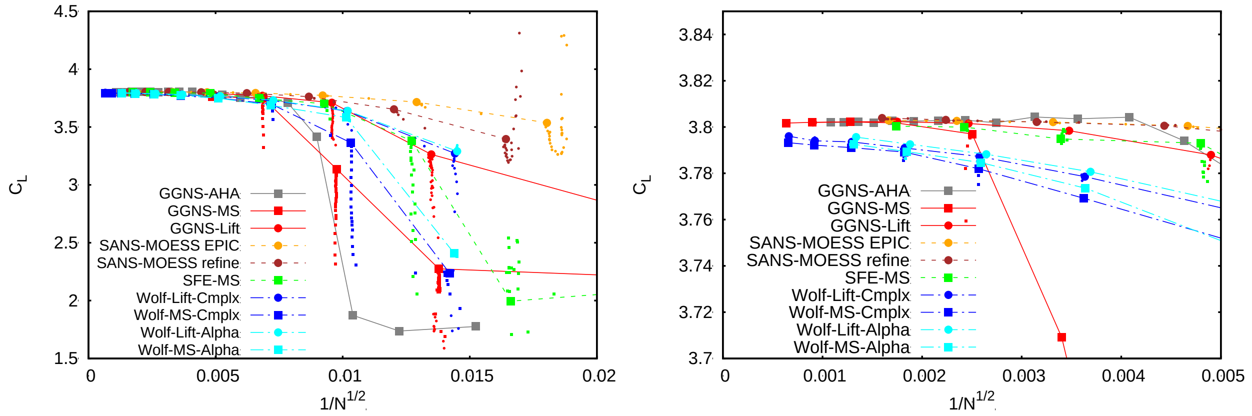

Figure 24: Lift Convergence, 16 Degrees Angle of Attack

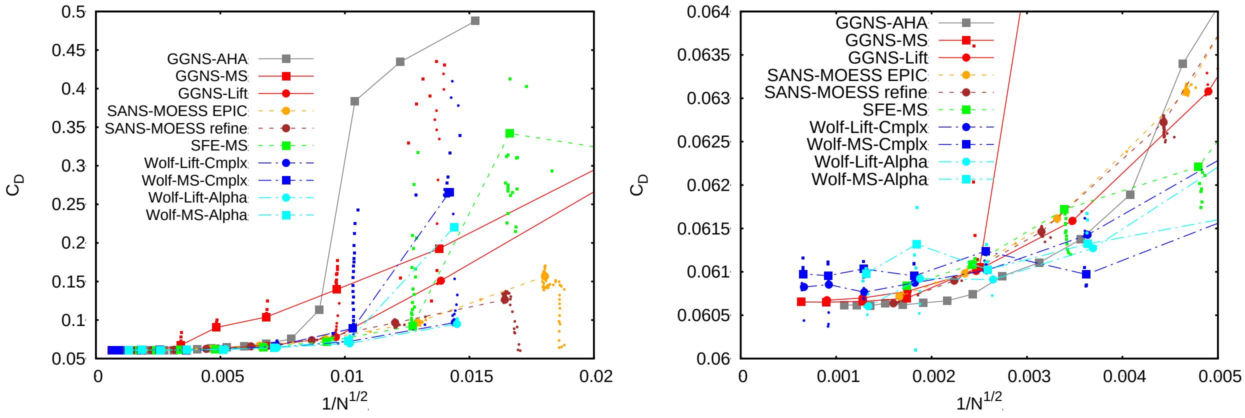

Figure 25: Drag Convergence, 16 Degrees Angle of Attack

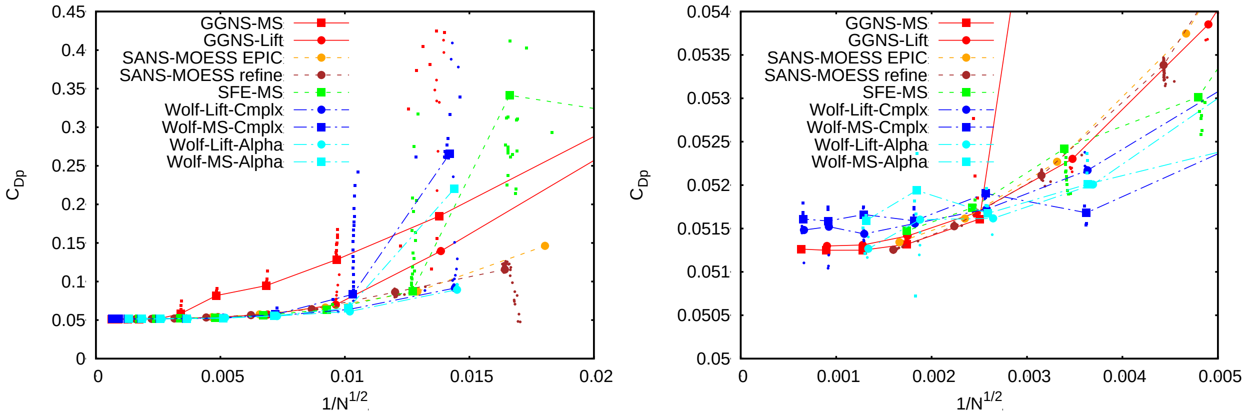

Figure 26: Pressure Drag Convergence, 16 Degrees Angle of Attack

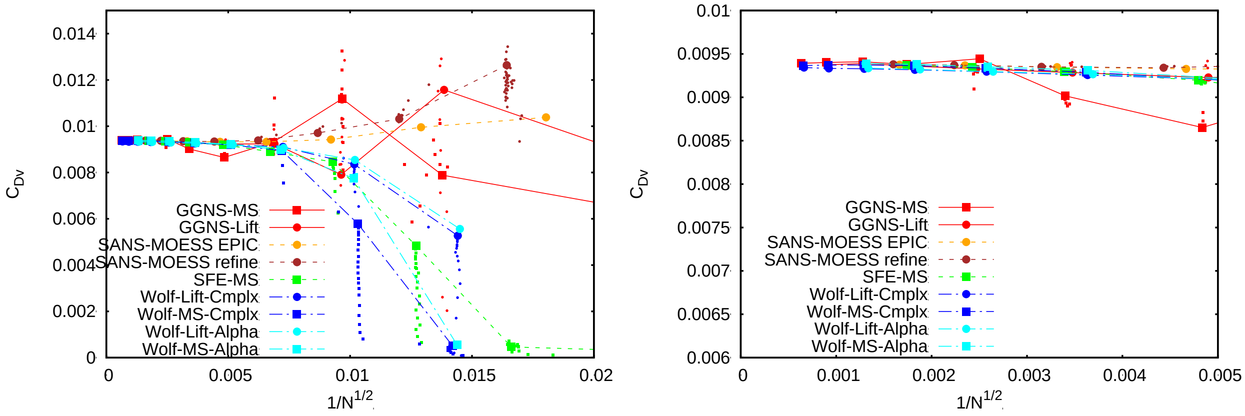

Figure 27: Viscous Drag Convergence, 16 Degrees Angle of Attack

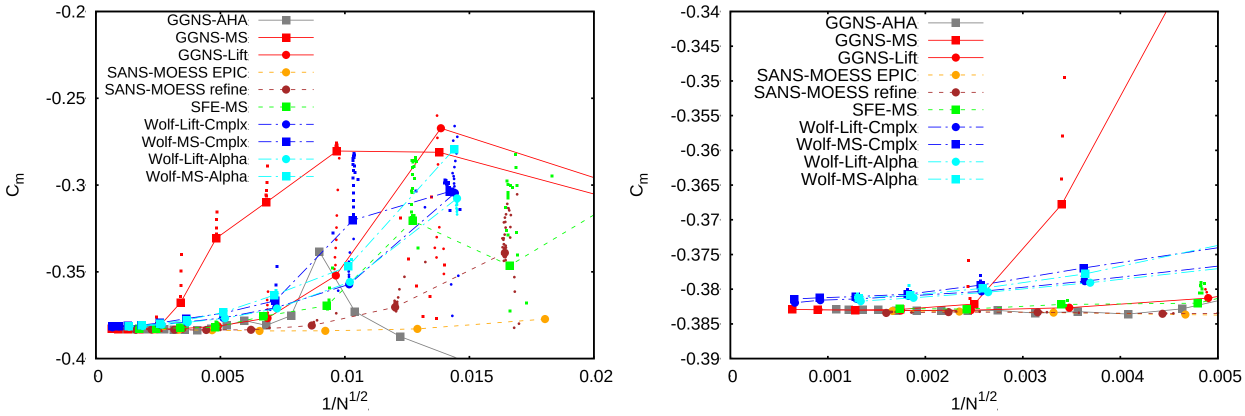

Figure 28: Pitching Moment Convergence, 16 Degrees Angle of Attack

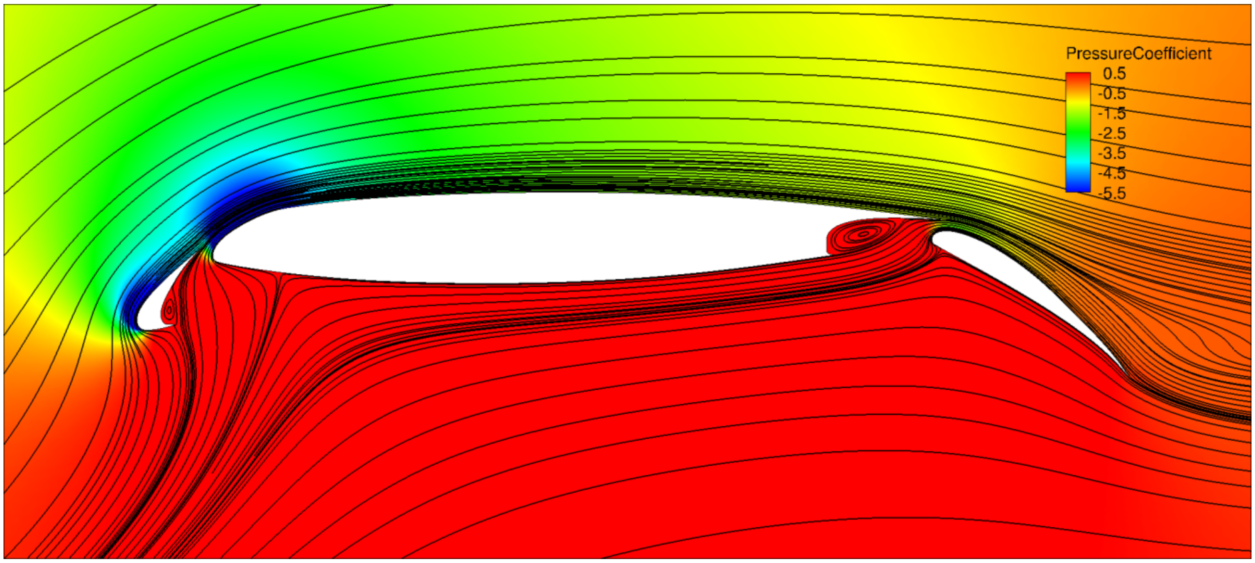

Figure 29: Streamlines on Finest GGNS-Lift Adapted Mesh, 16 Degrees Angle of Attack

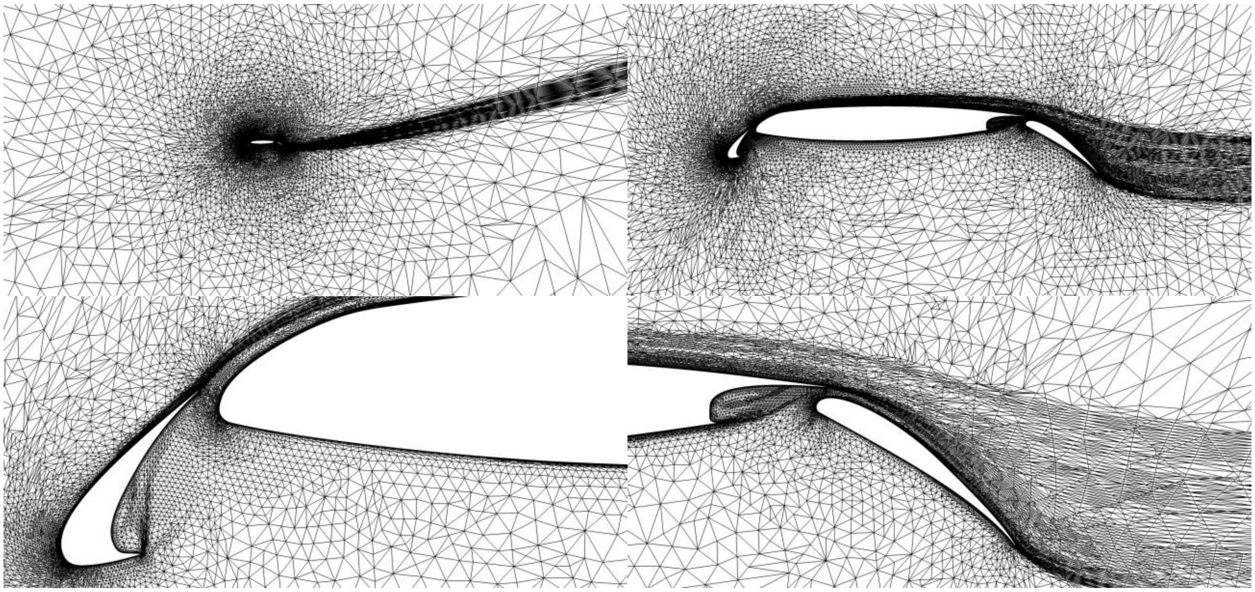

Figure 30: GGNS-AHA Medium Adapted Mesh, 78163 Nodes, 16 Degrees Angle of Attack

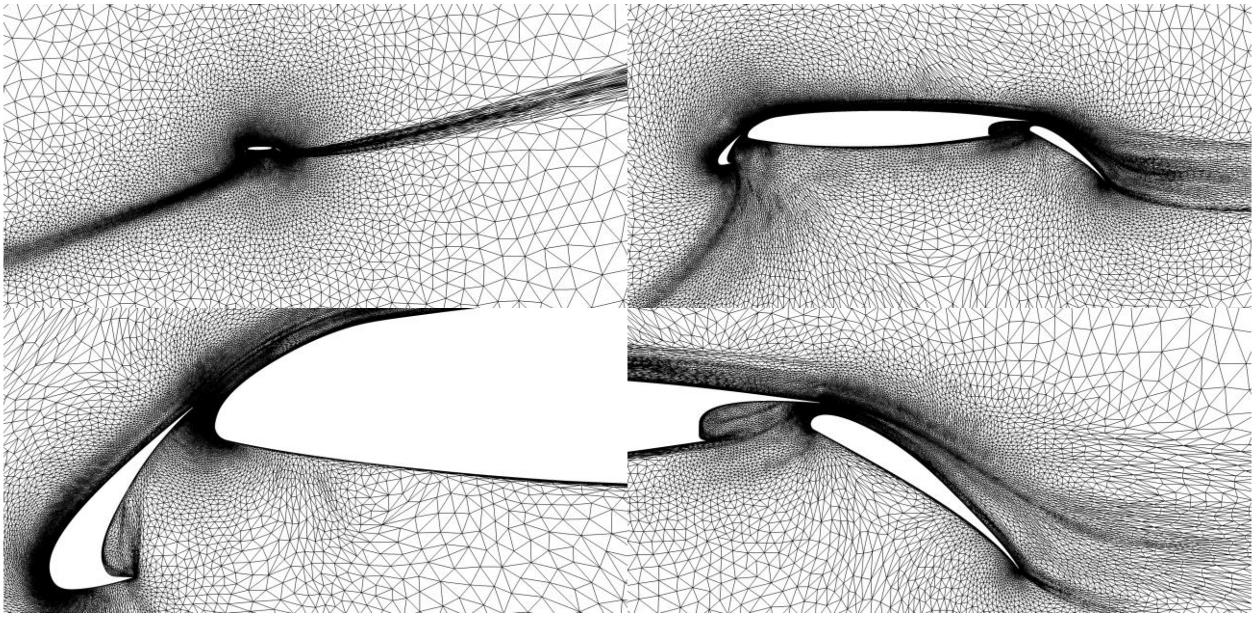

Figure 31: GGNS-Lift Medium Adapted Mesh, 83671 Nodes, 16 Degrees Angle of Attack

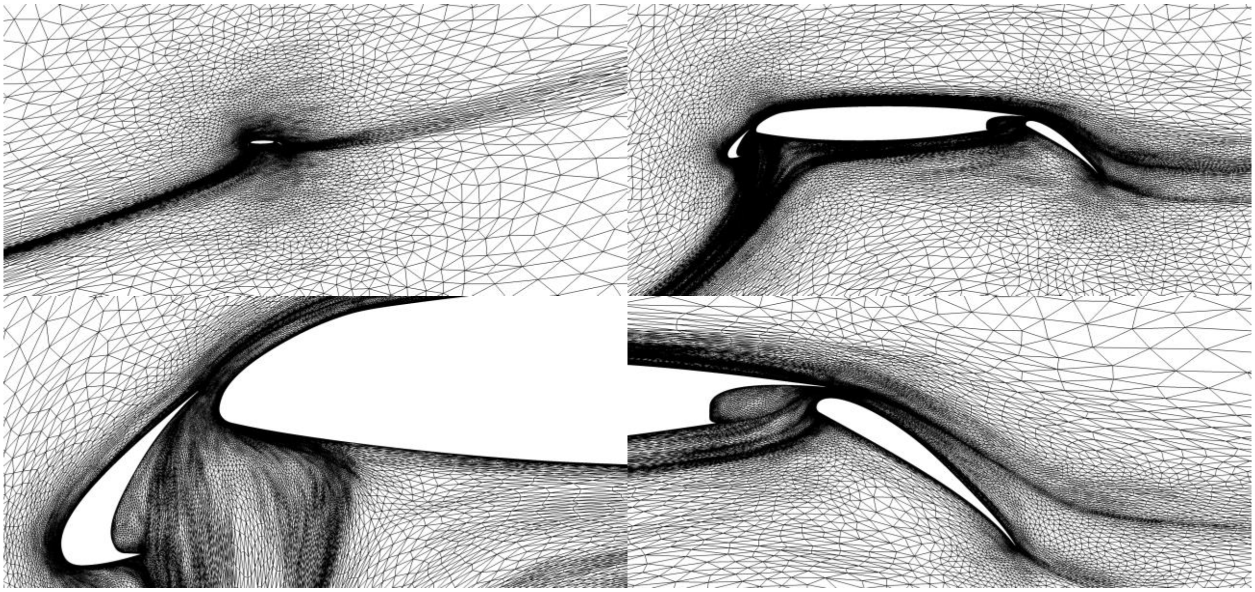

Figure 32: SANS-MOESS (EPIC) Medium Adapted Mesh, 91021 Nodes, 16 Degrees Angle of Attack

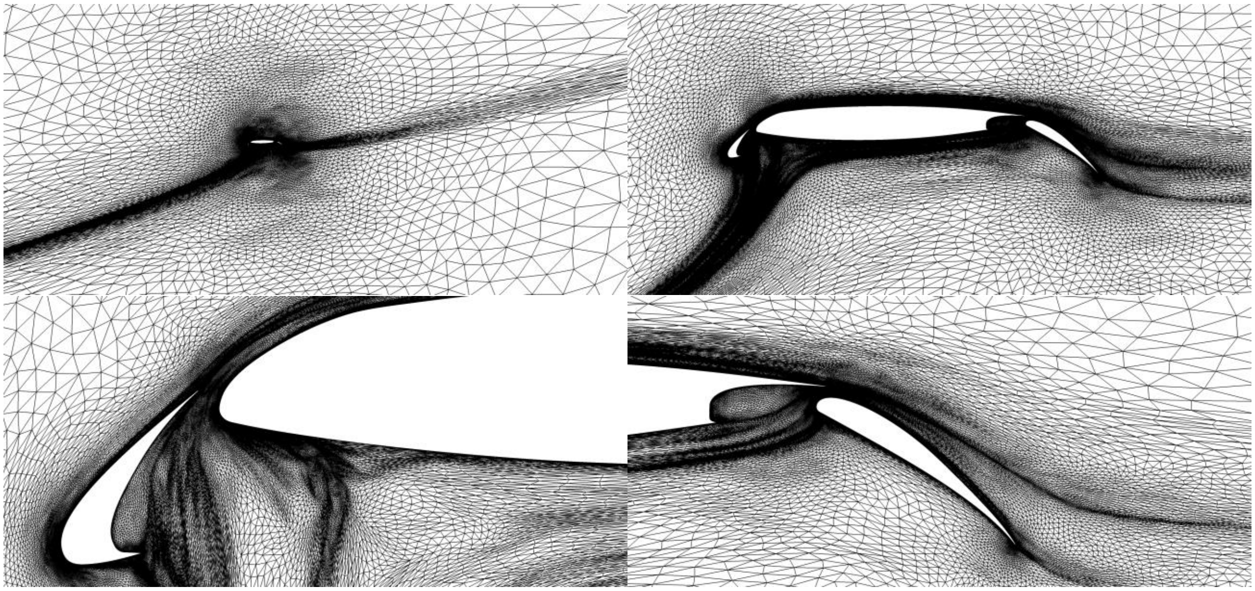

Figure 33: SANS-MOESS (refine) Medium Adapted Mesh, 100728 Nodes, 16 Degrees Angle of Attack

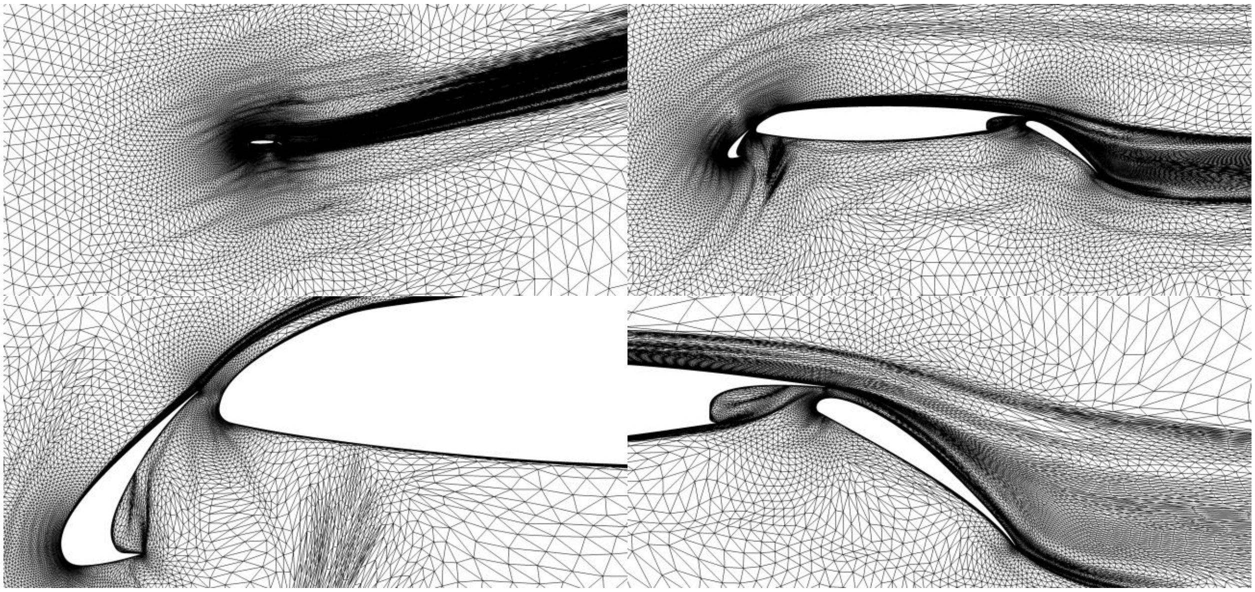

Figure 34: SFE-MS Medium Adapted Mesh, 86935 Nodes, 16 Degrees Angle of Attack

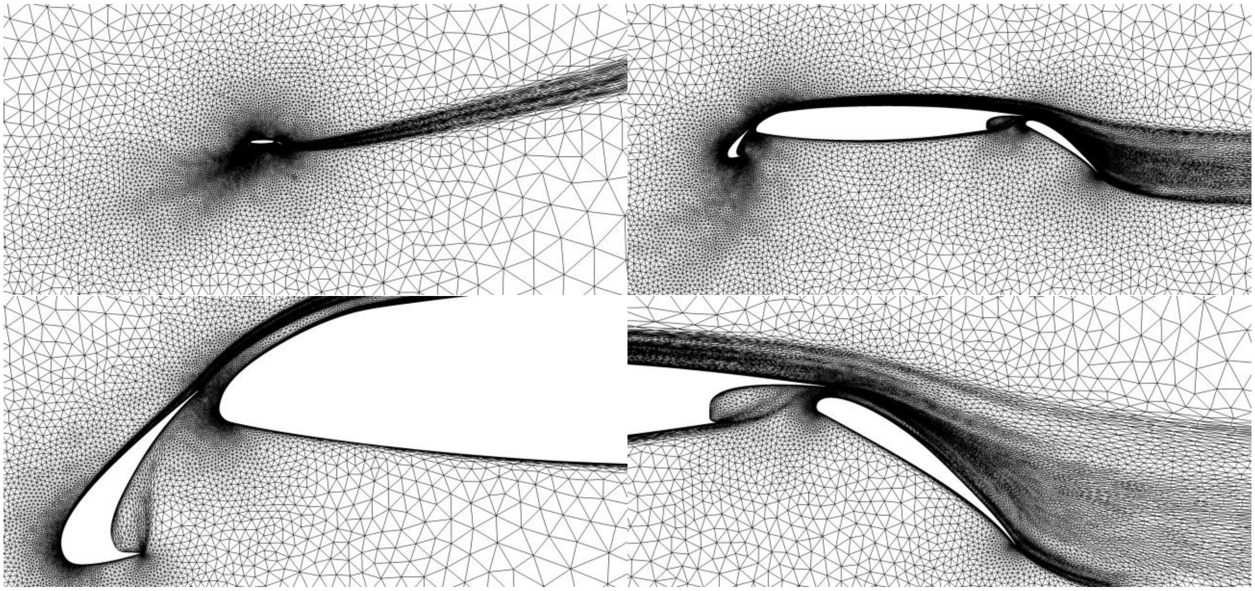

Figure 35: Wolf-GO Medium Adapted Mesh, 75896 Nodes, 16 Degrees Angle of Attack

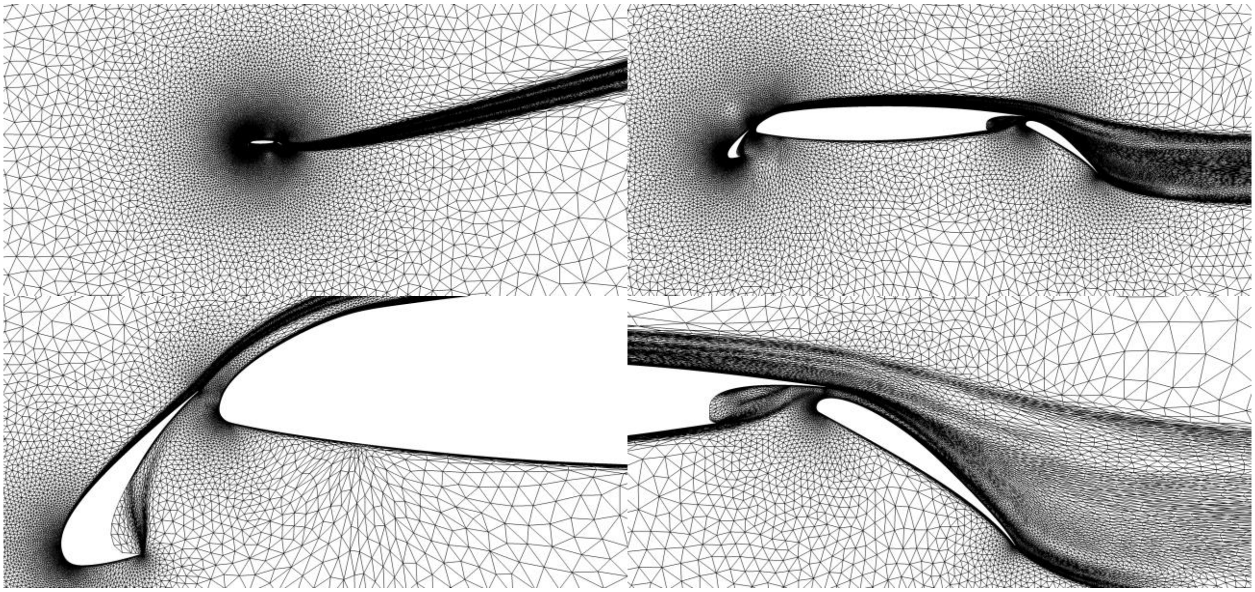

Figure 36: Wolf-MS Medium Adapted Mesh, 75943 Nodes, 16 Degrees Angle of Attack

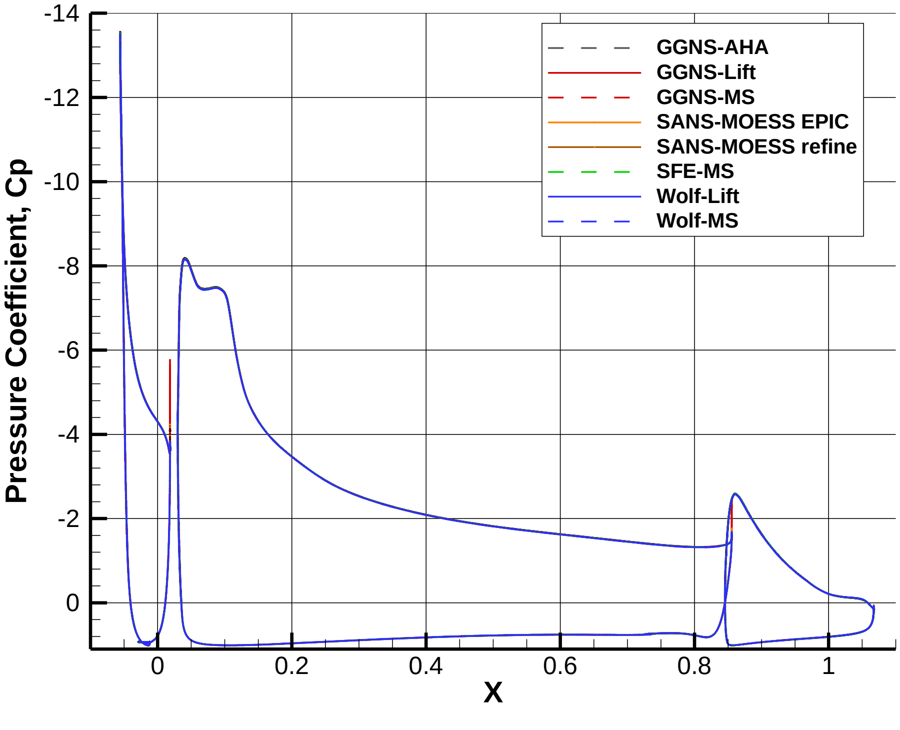

Figure 37: Surface Pressure Coefficient on Finest Adapted Mesh, Angle of Attack=16 degrees.

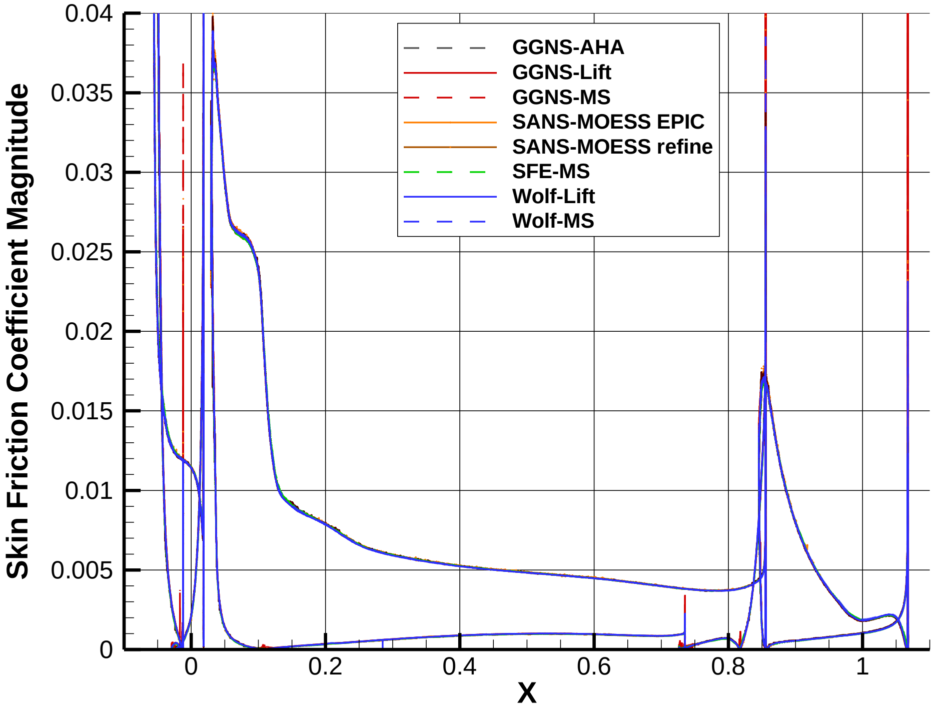

Figure 38: Skin Friction Magnitude on Finest Adapted Mesh, Angle of Attack=16 degrees.

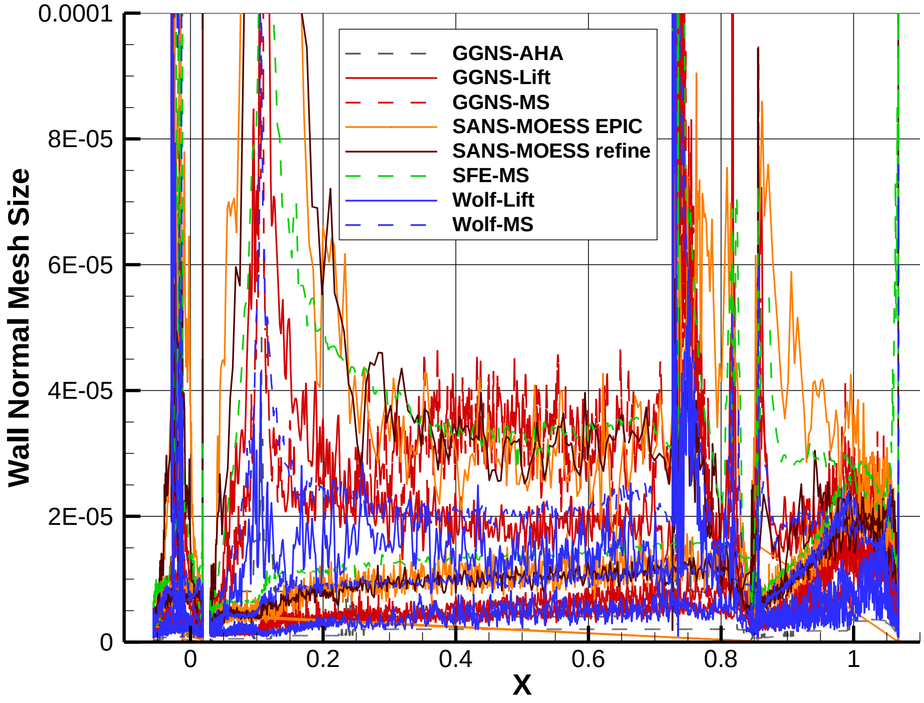

Figure 39: Wall Normal Spacing on Finest Adapted Mesh, 16 Degrees Angle of Attack

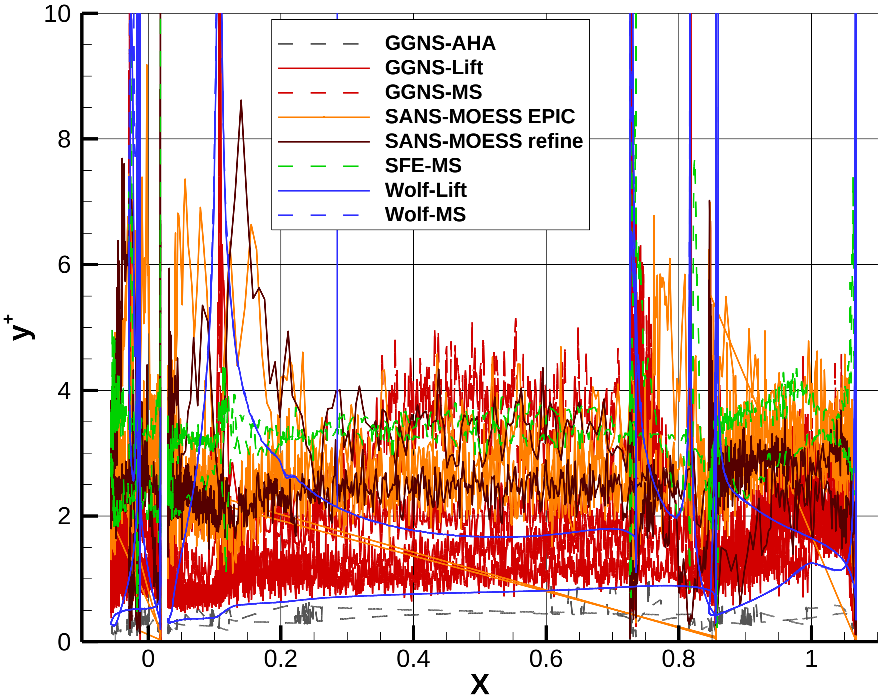

Figure 40: Y+ on Finest Mesh, 16 Degrees Angle of Attack.

C. *Angle of Attack Sweep Results*

Predictions of the HL-CRM airfoil lift, drag and moment were compared across the angle of attack range of -2 to 26 degrees. Adaptive solutions were generated by each method in two degree increments. All simulations were generated by impulsively starting from free stream conditions on a coarse initial mesh for each angle of attack. One exception to this was the Wolf-Lift (Alpha) method. This method used alpha continuation to initialize the solution at each angle of attack from the previous angle of attack. The adaptive simulations for each angle of attack were continued until reaching similar levels of mesh convergence shown at 8 and 16 degrees in the previous sections. Mesh convergence results were obtained at angles of attack up to 24 degrees even for the post stall angles of 22 and 24 degrees. As the angle of attack increased the flow solution computed on the coarse initial mesh showed increasing levels of separation above the wing and flap elements with a corresponding drop in the predicted lift. Increasing the DOF count during the normal adaptive mesh process caused the flow solution to reattach to the wing and flap surfaces with a corresponding increase in the predicted lift. As the angle of attack increased, the number of DOF's required to reattach the flow increased. At 26 degrees angle of attack, the flow solutions for the SFE-MS and AHA methods did not reattach even at the finest mesh level.

A comparison of the predicted lift curve and drag polar for all adaptive methods is shown in Figure 41. Predicted drag and pitching moment versus angle of attack are shown in Figure 42. The methods show excellent agreement in lift, drag and pitching moment up to 26 degrees angle of attack. Divergence of results in the post stall environment is not unexpected for steady state methods where multiple solution branches exist and slight variations in the analysis process can lead to jumps between solution branches.

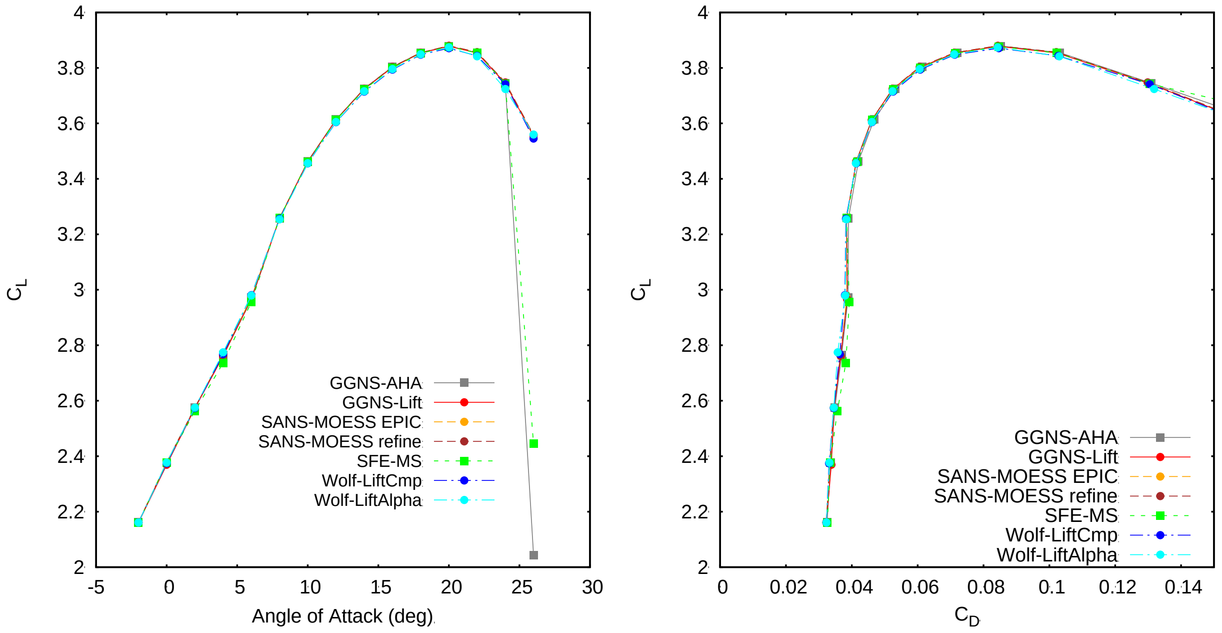

Figure 41: Comparison of Lift Curve and Drag Polar

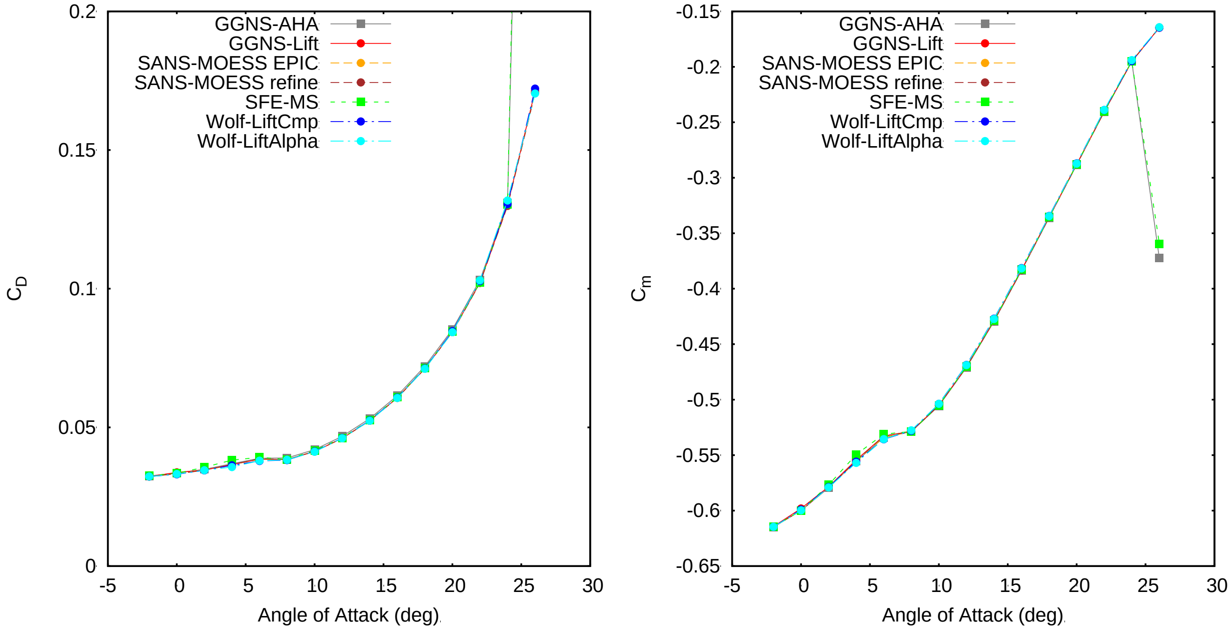

Figure 42: Comparison of Drag and Pitching Moment vs Angle of Attack

VIII. Summary and Conclusions

Adaptive mesh solutions were generated on a 2D cross section taken from the High Lift Common Research Model Wing. Solutions were generated with different combinations of flow solvers, mesh mechanics methods, error estimates, and adaptive mesh processes. The results show that by continuing the adaptation process to a sufficient level of mesh convergence, we are able to reproduce nearly identical results for any of the adaptive methods regardless of flow solver, mesh mechanics method or error estimate. Examination of the adaptive meshes revealed that while all the meshes sufficiently resolved important features of the flow, there were several differences in the resulting meshes depending on the mesh mechanics and error estimate that was used. Meshes generated with adjoint based output-error metrics resolved the stagnation streamlines while meshes generated with the MS and AHA heuristic error estimates placed extra refinement in the downstream wakes. There is a large variation in the wall normal spacing between the methods suggesting that some of the methods may be over resolving the near wall region. This normal spacing recommendation and other reommendations may be used to guide mesh generation for high lift and other 3D problems, particularly in a workshop setting where highly accurate simulations are pursued.

There are significant differences in the mesh convergence trajectories between methods. These differences in trajectory can have a significant impact on process efficiency. Output-error-based methods showed a clear advantage in being able to reduce the error to a given level with coarser meshes. It is expected this advantage will be even more prevalent in 3D where mesh sizes quickly start to encroach on the available computer hardware and clock time.

IX. References

[1] C.L. Rumsey, J. P. Slotnick, M. Long, R. A. Stuever, and T. R. Wayman. Summary of the First AIAA CFD High-Lift Prediction Workshop. *Journal of Aircraft*, Vol. 48, 2011, pp. 2068-2079.

[2] C.L. Rumsey and J. P. Slotnick. Overview and Summary of the Second AIAA High-Lift Prediction Workshop. *Journal of Aircraft*, Vol. 52, 2015, pp. 1006-1025.

[3] Loseille, A., Dervieux, A., and Alauzet, F., “Fully Anisotropic Goal-Oriented Mesh Adaptation for 3D Steady Euler Equations,” *Journal of Computational Physics*, Vol. 229, No. 8. 2010, pp 2866-2897.

[4] Park, MN. A. and Darmofal, D. L., “Parallel Anisotropic Tetrahedral Adaptation,” AIAA Paper 2008-917, 2008.

[5] Michal, T. and Krakos, J., “Anisotropic Mesh Adaptation through Edge Primitive Operations," AIAA Paper 2012-159, 2012.

[6] Michal, T., Babcock, D., Kamenetskiy, D, Krakos, J., Mani, M., Glasby, R., Erwin, T., and Stefanski, D. “Comparison of Fixed and Adaptive Unstructured Grid Results for Drag Prediction Workshop 6”, AIAA *Journal of Aircraft*, Vol. 55, No. 4. , 2017.

[7] Turbulence Model Benchmarking Working Group (TMBWG), AIAA Fluid Dynamics Technical Committee. Turbulence Modeling Resource “https://turbmodels.larc.nasa.gov/naca0012numerics_val_sa_withoutpv.html

[8] A. Loseille and F. Alauzet. Continuous mesh framework. Part I: well-posed continuous interpolation error. *SIAM J. Numer. Anal.*, 49(1):38-60,2011.

[9] Loseille, A., and Alauzet, F., “Continuous Mesh Framework Part II: Validations and Applications,” *SIAM Journal on Numerical Analysis*, Vol. 49, No. 1, 2011, pp. 61–86. doi:10.1137/10078654X.

[10] F. Alauzet and A. Loseille. 2010., “High-order Sonic Boom Modeling Based on Adaptive Methods,” *J. Comput. Phys*. 229, 3 (February 2010), 561-593.

[11] Park, M. A., Loseille, A., Krakos, J. A., and Michal, T., “Comparing Anisotropic Output-Based Grid Adaptation Methods by Decomposition,” AIAA Paper 2015–2292, 2015.

[12] Michal, T., Kamenetskiy, D. S., Marcum, D., Alauzet, F., Frazza, L., and Loseille, A., “Comparing Anisotropic Error Estimates for ONERA M6 Wing RANS Simulations,” AIAA Paper 2018–920, 2018.

[13] Anderson, W. K., Newman, J. C., and Karman, S. L., “Stabilized Finite Elements in FUN3D,” *Journal of Aircraft*, Vol. 55, No. 2, 2018, pp. 696–714. doi:10.2514/1.C034482.

[14] Brooks, A. N., and Hughes, T. J. R., “Streamline Upwind/Petrov-Galerkin Formulation for Convection Dominated Flows with Particular Emphasis on Incompressible Navier-Stokes Equations,” *Computer Methods in Applied Mechanics and Engineering*, Vol. 32, No. 1–3, 1982, pp. 199–259. doi:10.1016/0045-7825(82)90071-8.

[15] Shakib, F., Hughes, T. J. R., and Johan, Z., “A New Finite-Element Formulation for Computational Fluid Dynamics: X. The Compressible Euler and Navier-Stokes Equations,” *Computer Methods in Applied Mechanics and Engineering*, Vol. 89, No. 1–3, 1991, pp. 141–219. doi:10.1016/0045-7825(91)90041-4.

[16] Hughes, T. J. R., Franca, L. P., and Hulbert, G. M., “A New Finite-Element Formulation for Computational Fluid Dynamics: VIII. The Galerkin Least Squares Method for Advective-Diffusion Equations,” *Computer Methods in Applied Mechanics and Engineering*, Vol. 73, No. 2, 1989, pp. 173–189. doi:10.1016/0045-7825(89)90111-4.

[17] Bazilevs, Y., and Akkerman, I., “Large Eddy Simulation of Turbulent Taylor–Couette Flow Using Isogeometric Analysis and the Residual-Based Variational Multiscale Method,” *Journal of Computational Physics*, Vol. 229, No. 9, 2010, pp. 3402–3414. doi:10.1016/j.jcp.2010.01.008.

[18] Saad, Y., *Iterative Methods for Sparse Linear Systems*, 2nd ed., Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2003.

[19] Allmaras, S. R., "Lagrange Multiplier Implementation of Dirichlet Boundary Conditions in Compressible Navier-Stokes Finite Element Methods", AIAA Paper 2005-4714, June 2005.

[20] Dmitry S. Kamenetskiy, John E. Bussoletti, Craig L. Hilmes, Venkat Venkatakrishnan, Laurence B. Wigton, Numerical Evidence of Multiple Solutions for the Reynolds-Averaged Navier–Stokes Equations, AIAA Journal Vol. 52, No. 8 (2014), pp. 1686-1698.

[21] Saad, Y., and Schultz, M. H., "GMRES: A Generalized Minimum Residual Algorithm for Solving Nonsymmetric Linear Systems," *SIAM Journal of Scientific and Statistical Computing*, Vol. 7, 1986, pp. 856–869.

[22] Galbraith, M. C., Allmaras, S. R., and Darmofal, D. L., "SANS RANS Solutions for 3D Benchmark Configurations," AIAA Paper 2018–1570, 2018.

[23] Huang, A. C., "An Adaptive Variational Multiscale Method with Discontinuous Subscales for Aerodynamic Flows," Ph.D. thesis, Massachusetts Institute of Technology, Feb. 2020.

[24] Huang, A. C., Carson, H.A., Allmaras, S. R., Galbraith, M. C., Darmofal, D. L., and Kamenetskiy, D. S., "A Variational Multiscale Method with Discontinuous Subscales for Output-Based Adaptation of Aerodynamic Flows," AIAA Paper 2020-1563, 2020.

[25] Galbraith, M. C., Allmaras, S. R., and Darmofal, D. L., "A Verification Driven Process for Rapid Development of CFD Software," AIAA Paper 2015–818, 2015.

[26] Abhyankar, S., Brown, J., Constantinescu, E. M., Ghosh, D., Smith, B. F., and Zhang, H., "PETSc/TS: A Modern Scalable ODE/DAE Solver Library," *Computing Research Repository (CoRR)*,Vol.Numerical Analysis (math.NA),No. arXiv:1806.01437, 2018. URL http://arxiv.org/abs/1302.6066.

[27] Balay, S., Abhyankar, S., Adams, M. F., Brown, J., Brune, P., Buschelman, K., Dalcin, L., Eijkhout, V., Gropp, W. D., Kaushik, D., Knepley, M. G., McInnes, L. C., Rupp, K., Smith, B. F., Zampini, S., Zhang, H., and Zhang, H., "PETSc Users Manual," Tech. Rep. ANL-95/11 - Revision 3.8, Argonne National Laboratory, 2017.

[28] Balay, S., Gropp, W. D., McInnes, L. C., and Smith, B. F., "Efficient Management of Parallelism in Object Oriented Numerical Software Libraries," *Modern Software Tools in Scientific Computing*, edited by E. Arge, A. M. Bruaset, and H. P. Langtangen, Birkhäuser Press, 1997, pp. 163–202.

[29] Toro, E. F., Spruce, M., and Speares, W., "Restoration of the Contact Surface in the HLL-Riemann Solver," *Shock Waves*, Vol. 4, No. 1, 1994, pp. 25–34. doi:10.1007/BF014146292.

[30] Menier, V., Loseille, A., and Alauzet, F., "CFD Validation and Adaptivity for Viscous Flow Simulations," AIAA Paper 2014–2925, 2014.

[31] Loseille, A., Alauzet, F., and Menier, V., "Unique Cavity-Based Operator and Hierarchical Domain Partitioning for Fast Parallel Generation of Anisotropic Meshes," *Computer-Aided Design*, Vol. 85, 2017, pp. 53–67. doi:10.1016/j.cad.2016.09.008, 24th International Meshing Roundtable Special Issue: Advances in Mesh Generation.

[32] Loseille, A., and Löhner, R., "Cavity-Based Operators for Mesh Adaptation," AIAA Paper 2013–152, 2013.

[33] Haimes, R., and Drela, M., "On The Construction of Aircraft Conceptual Geometry for High-Fidelity Analysis and Design," AIAA Paper 2012-0683, 2012.

[34] Alauzet, F., "A Changing-Topology Moving Mesh Technique for Large Displacements," *Engineering with Computers*, Vol. 30, No. 2, 2014, pp. 175–200. doi:10.1007/s00366-013-0340-z.

[35] Loseille, A., Dervieux, A., Frey, P. J., and Alauzet, F., "Achievement of Global Second Order Mesh Convergence for Discontinuous Flows with Adapted Unstructured Meshes," AIAA Paper 2007–4186, 2007.

[36] Arsigny, V., Fillard, P., Pennec, X., and Ayache, N., "Log-Euclidean Metrics for Fast and Simple Calculus on Diffusion Tensors," *Magnetic Resonance in Medicine*, Vol. 56, No. 2, 2006, pp. 411–421. doi:10.1002/mrm.20965.

[37] Alauzet, F., "Size Gradation Control of Anisotropic Meshes," *Finite Elements in Analysis and Design*, Vol. 46, No. 1–2, 2010, pp. 181–202. doi:10.1016/j.finel.2009.06.028.

[38] Fidkowski, K. J., and Darmofal, D. L., "Review of Output-Based Error Estimation and Mesh Adaptation in Computational Fluid Dynamics," *AIAA Journal*, Vol. 49, No. 4, 2011, pp. 673–694. doi:10.2514/1.J050073.

[39] Yano, M., "An Optimization Framework for Adaptive Higher-Order Discretizations of Partial Differential Equations on Anisotropic Simplex Meshes," Ph.D. thesis, Massachusetts Institute of Technology, Jun. 2012. doi:1721.1/76090.

[40] Yano, M., and Darmofal, D. L., "An Optimization-Based Framework for Anisotropic Simplex Mesh Adaptation," *Journal of Computational Physics*, Vol. 231, No. 22, 2012, pp. 7626–7649. doi:10.1016/j.jcp.2012.06.040.

[41] Carson, H.A., Huang, A. C., Galbraith, M. C., Allmaras, S. R., and Darmofal, D. L., "Mesh optimization via error sampling and synthesis: An update," AIAA Paper 2020-0087, 2020.

[42] Becker, R., and Rannacher, R., "A Feed-Back Approach to Error Control in Finite Element Methods: Basic Analysis and Examples," *East-West Journal of Numerical Mathematics*, Vol. 4, 1996, pp. 237–264.

[43] Carson, H. A., Darmofal, D. L., Galbraith, M. C., and Allmaras, S. R., "Analysis of Output-Based Error Estimation for Finite Element Methods," *Applied Numerical Mathematics*, Vol. 118, 2017, pp. 182–202. doi:10.1016/j.apnum.2017.03.004.

[44] Zienkiewicz, O. C., and Zhu, J. Z., "The Superconvergent Patch Recovery and A Posteriori Error Estimates. Part 1: The Recovery Technique," *International Journal for Numerical Methods in Engineering*, Vol. 33, No. 7, 1992, pp. 1331–1364.

[45] 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. doi:10.1016/S0021-9991(03)00074-3.

[46] Belme, A., Alauzet, F, and Dervieux, A., "An a priori anisotropic Goal-Oriented Estimate for Viscous Compressible Flow and Application to Mesh Adapatation," *Journal of Computational Physics*, Vol. 376, 2019, pp. 1051-1088.

[47] Alauzet, F. and Frazza, L., "3D RANS anisotropic mesh adaptation on the high-lift versionof NASA's Common Research Model (HL-CRM)," AIAA paper 2019-2947, 2019.

[48] Geometry Modeling and Mesh Generation Working Group, AIAA Meshing Visualization and Compuational Environments Technical Committee, Special Session on Mesh Effects for CFD Solutions, http://www.gmgworkshop.com/gmgw25.shtml

[49] Geometry Modeling and Mesh Generation Working Group, AIAA Meshing Visualization and Compuational Environments Technical Committee, AIAA Geometry and Mesh Generation Workshop, http://www.gmgworkshop.com/

[50] Lacy, D. S, and Sclafani, A. J., "Development of the High Lift Common Research Model (HL-CRM): A Representative High Lift Configuration for Transonic Transports," AIAA Paper 2016-0308, 2016.