Verification of Anisotropic Mesh Adaptation for RANS Simulations over ONERA M6 Wing

Aravind Balan,[*] Michael A. Park[†] and William K. Anderson[‡]
NASA Langley Research Center, Hampton, VA 23681, USA

Dmitry S. Kamenetskiy[§]
Boeing Commercial Airplanes, Seattle, WA, USA

Joshua A. Krakos[¶] and Todd Michal[‖]
The Boeing Company, St. Louis, MO, USA

Frédéric Alauzet[**]
INRIA Paris-Saclay, Alan Turing Building, 91120 Palaiseau, France

Unstructured anisotropic mesh adaptation is known to be an efficient way to control discretization errors in Computational Fluid Dynamics (CFD) simulations. Method verification is required to provide the confidence for routine use in production analysis. The current work aims at verification of anisotropic mesh adaptation for RANS simulations over the ONERA M6 wing. The present verification study is performed using four different flow solvers, three different implementations of the metric field, and three mesh mechanics packages. Two of the flow solvers use stabilized finite-element discretizations (FUN3D-SFE and GGNS), one uses finite-volume discretization (FUN3D-FV), and the last one uses mixed finite-volume and finite-element discretizations (WOLF). The mesh adaptation is based on an error estimator that aims to control the quadratic error term in the linear interpolation of Mach number. Two sets of adaptations were performed; the first one controls the interpolation error in L^2 norm and the second one controls the interpolation error in L^4 norm. Convergence studies were performed on the forces and the pitching moment using all four solvers, and the results are compared with previously verified convergence studies on fixed (nonadapted) meshes. Both forces and pitching moment on adapted meshes are found to be converging to the fine mesh values faster than those on fixed meshes. In addition to forces and moments, convergence of surface pressure and skin friction coefficients at various measurement locations on the wing are also presented. Adapted-mesh surface pressure distributions agree with the fine fixed mesh pressure distribu-

[*] NASA Postdoctoral Fellow, Computational AeroSciences Branch, AIAA Member.
[†] Research Scientist, Computational AeroSciences Branch, AIAA Associate Fellow.
[‡] Research Scientist, Computational AeroSciences Branch, AIAA Associate Fellow.
[§] Engineer, AIAA Senior Member.
[¶] Engineer, AIAA Senior Member.
[‖] Technical Fellow, AIAA Senior Member.
[**] Researcher, GAMMA3 Team, AIAA Member.

tions. Adapted-mesh skin friction distributions contain high frequency noise with mean values approaching the fixed mesh pressure skin friction distributions.

I. Introduction

The CFD Vision 2030 Study: A Path to Revolutionary Computational Aerosciences [1] proposes a research strategy for developing visionary CFD capability by the year 2030. The CFD Vision 2030 study specifically identifies mesh generation and adaptive-mesh technology to be significant bottlenecks in the current CFD work flow. There has been an ongoing effort to address the issues that prevent the use of mesh adaptation in routine production analysis. One of the issues being addressed is the verification of the end-to-end mesh adaptation process for problems of increasing complexity, including the benchmark test cases listed on the Turbulence Modeling Resource (TMR) web site [2]. The present work is part of such a verification effort, and focusses on unstructured anisotropic mesh adaptation for RANS simulations over the ONERA M6 wing.

Mesh adaptation is a means for automating turbulent flow simulations by providing an effective tool for systematically reducing discretization errors. In this context, metric-based mesh adaptation that relies on the concept of continuous mesh and metric fields to control interpolation and output errors has gained broader application. Loseille et al. [3] demonstrated the potential of metric-based anisotropic mesh adaptation using several three-dimensional test cases. Fidkowski and Darmofal [4] provide an excellent summary of various techniques for doing anisotropic mesh adaptation based on output error estimators of CFD simulations. Alauzet and Loseille [5] review the progress made in anisotropic mesh adaptation for CFD during the last decade. In a more recent paper, Park et al. [6] review the status of unstructured mesh adaptation, identify the impact of robust automated unstructured mesh technologies, and recommend the adoption of newly developed and matured unstructured mesh adaptation technologies with an intent of contributing to the CFD 2030 Vision Study capabilities mentioned in Ref.[1].

Park et al. [7] examined the anisotropic mesh adaptation by decomposing the adaptation cycle into individual steps consisting of flow solution, error estimation, metric construction and mesh generation (see Fig. 1). Multiple implementations of each of these steps were compared to each other. The informal Unstructured Grid Adaptation Working Group (UGAWG) has been formed to continue this process as described in their first benchmark [8]. The work focused on evaluating adaptive mesh mechanics for analytic metric fields on planar and simple curved domains. UGAWG's efforts continued with the verification of various anisotropic mesh adaptation tools using several test cases including the benchmark cases listed on the TMR wesbsite [9, 10]. TMR provides the results of multiple codes on a set of uniformly-refined meshes to evaluate the convergence of integrated forces. Independently implemented anisotropic mesh adaptation procedures showed consistent convergence to fine-mesh forces and moments for each of those test cases.

The current work is a continuation of the verification efforts of UGAWG, and focusses on the anisotropic mesh adaptation for RANS simulations over the ONERA M6 wing. The mesh adaptation is based on a metric field that aims to control the linear interpolation error in Mach number. The TMR website has detailed documentation for the ONERA M6 test case and provides results from uniform mesh refinement using multiple numerical schemes that were originally published by Diskin et al. [11]. The results from the uniformly refined meshes are also used here to compare those from adapted meshes. The general adaptation methodology and specific implementations evaluated are described in Section II, adaptive results are then presented for the RANS simulations over ONERA M6 wing.

II. Anisotropic Mesh Adaptation Processes

The components of metric-based anisotropic unstructured mesh adaptation are shown on Fig. 1. Starting with an initial mesh, a flow solution is computed. The information from the flow solution is used to estimate error and specify a new mesh resolution request via an anisotropic metric field $\mathcal{M}$. If the estimated errors are larger than limits specified by the practitioner, the current mesh system is modified by mesh mechanics such as edge swap, node movement, and adding nodes if needed, etc., to conform to the anisotropic metric $\mathcal{M}$. 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 exit criteria are met (e.g., accuracy requirement, resource limit). Details of the specific implementations of these components is detailed in the following sections.

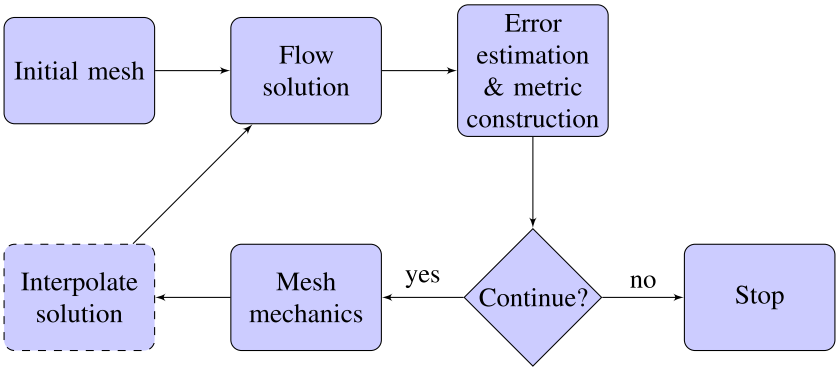


Fig. 1 Solution-based mesh adaptation process with optional components indicated by dashed outlines.

There are several ways a metric field, $\mathcal{M}$, can be constructed that encodes anisotropic mesh information. Loseille and Alauzet [12] provide a thorough introduction to the metric tensor field. Here we consider what is known as the multiscale metric field. The multiscale metric controls the L^p-norm of the interpolation error of a solution scalar field [13]. The multiscale metric balances refinement of smooth and nonsmooth regions of the solution, which differs from classic Hessian-based metric approaches [14] that primarily focus on nonsmooth regions by controlling error in the L^∞-norm. The metric field $\mathcal{M}_{L^p}$ is constructed from the (reconstructed) Hessian, $\mathcal{H}$, of the scalar field that is locally scaled by the Hessian determinant, $\det(\mathcal{H})$, and globally scaled to a specified target to get,

$$\mathcal{M}_{L^p} = D_{L^p} \det(\mathcal{H})^{\frac{-1}{2p+d}} |\mathcal{H}|, \tag{1}$$

where the global scaling D_{L^p},

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

corrects the complexity of the locally scaled Hessian to produce $\mathcal{M}_{L^p}$ with specified target complexity C_t. $|\mathcal{H}|$ is derived from $\mathcal{H}$ by taking the absolute value of the eigenvalues. Both scaling operations depend on the dimensionality of the domain, which is $d = 3$ in the present case. The complexity, C, of a continuous metric field, $\mathcal{M}$, is defined as the integral,

$$C(\mathcal{M}) = \int_{\Omega} \sqrt{\det(\mathcal{M}(x))}\, dx. \tag{3}$$

The complexity can be interpreted as the continuous counterpart of the number of vertices in the discrete mesh. The relationship between C and the number of vertices and elements in the adapted mesh is shown theoretically by [12] and experimentally by [7, 15]. A mesh conforming to $\mathcal{M}_{L^p}$ provides optimal control of the scalar field interpolation error in the L^p-norm. A lower value of p targets weaker variations of the scalar field and a larger value of p targets rapid variations of the scalar field [13]. Mesh adaptations using the multiscale metric have been performed for various problems of increasing complexity, and have proven its viability [16, 17]. The various implementations of the multiscale metric have also been verified for turbulent RANS benchmark problems [10]. For the numerical simulations presented in this paper, Mach number is chosen as the scalar field.

Mesh adaptation is an inherently nonlinear process. The robustness of the adaptive procedure is enhanced by optimizing the mesh at a fixed-complexity, which allows for control of Degrees of Freedom (DOF). An adaptive series of meshes is created by optimizing the metric at a fixed complexity and then increasing the target complexity in a series of steps. This is done by evaluating Eq. 1 on each mesh after the flow solve to get the next mesh for a fixed complexity. Typically, 5 to 10 fixed-complexity adaptation iterations are performed before increasing the complexity to the next target.

III. Flow Solvers, Metric Formulations and Mesh Mechanics

The description of the solution-based mesh adaptation process begins with the flow solver component. Multiple flow solvers are employed to compute the flow field. 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. The 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. The various flow solvers used in the verification effort are described below, along with the details of the corresponding metric field implementation and mesh mechanics packages used.

A. FUN3D-FV

FUN3D-FV (Fully-Unstructured Navier-Stokes $3D$) [18, 19] is a finite-volume Navier-Stokes solver in which the flow variables are stored at the vertices of the mesh. FUN3D-FV can solve the equations on mixed-element meshes, including tetrahedra, pyramids, prisms and hexahedra. The adaptive meshes in this study contain only tetrahedra. At interfaces between neighboring control volumes, the inviscid fluxes are computed using the Roe [20] approximate Riemann solver based on the values on either side of the interface. For 2^{nd}-order accuracy, interface values are extrapolated from the gradients computed at the mesh vertices. These gradients are reconstructed with an unweighted least-squares technique [18].

For the discretization of the full viscous fluxes, the required velocity gradients on the dual faces are computed using the Green-Gauss theorem. On tetrahedral meshes this is equivalent to a Galerkin-type approximation. The solution at each time step is updated with a backward Euler time-integration scheme. At each time step, the linear system of equations is approximately solved with a multicolor point-implicit procedure [21]. Local time-step scaling is employed to accelerate convergence to steady state. The negative Spalart-Allmaras (SA-neg) turbulence model [22] is loosely-coupled to the meanflow equations, where the meanflow and turbulence model equations are relaxed in an alternating sequence.

The SA-neg turbulence model requires the distance from every vertex to the nearest no-slip boundary condition. The standard wall distance calculation in FUN3D-FV finds the nearest surface vertex and then searches adjacent triangles to see if they are closer than the closest surface vertex. The standard wall distance method overestimates the distance to the no-slip boundary if the closest triangle is not adjacent to the closest surface vertex, which is a common case for adapted meshes. An accurate wall distance on adapted meshes is computed by enclosing multiple surface triangles in a bounding box that is stored in an octree that allows rapid identification but that is then exhaustively searched.

To form the metric, a Hessian of the scalar field is reconstructed by recursive application of L^2-projection [23]. The gradient is computed in each element and a volume-weighted average is collected at each vertex [23]. The 2^{nd}-derivative Hessian terms are formed by computing the reconstructed gradients using the gradients formed in the first pass. The mixed derivative terms of the Hessian are averaged. A special boundary treatment is employed. The reconstructed 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 [23] with Eq. (1). The complexity is computed, and the metric is globally scaled to set its complexity to a specified value. The complexity Eq. (3) is evaluated discretely by assuming it is piecewise constant in each median dual. The mesh is adapted by `refine` to conform to the metric.

The `refine` open source mesh adaptation mechanics package fulfills the error estimation and mesh mechanics components. It is available via `https://github.com/NASA/refine` under the Apache License, Version 2.0. The current version under development uses the combination of split, collapse, and element swap operations [24]. Some of these classic operators are undergoing replacement with cavity operators [25]. Vertex relocation is performed to improve adjacent element shape with a convex combination of ideal vertex locations [26] or nonsmooth optimization based on Freitag and Ollivier-Gooch [27]. `refine` requires the domain to be a manifold, and have a one-to-one correspondence of discrete vertex, segment elements, and triangle elements to geometry node, geometry edge, and geometry face entities, i.e., virtual topology, chains, and quilts are not supported. Geometry is accessed through the EGADS application program interface, and parallel execution is facilitated by EGADSlite [28].

B. FUN3D-SFE

FUN3D-SFE is a continuous Stabilized Finite-Element discretization within FUN3D [29] that uses the Streamlined Upwind Petrov-Galerkin (SUPG) scheme [30, 31].

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. The alternative wall distance calculation method, described in the previous FUN3D-FV section, is used with FUN3D-SFE. A linear nodal basis is used in this study, which is designed to be 2$^{\text{nd}}$-order accurate in space. The current implementation includes the capability for computing on tetrahedra, hexahedra, pyramids, and prisms, although all the results shown in the present paper are for purely-tetrahedral meshes.

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 [29]. 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 [32] 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 initial 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.

For the numerical results of the ONERA M6 test case, strong, noslip boundary conditions are used and the finite-element solution variables are transferred to the FUN3D-FV driver, which subsequently computes the forces by using previously developed routines. The metric field is calculated using the same method described in Section A. The mesh is adapted by `refine` to conform to the metric.

C. 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 2$^{\text{nd}}$-order accurate discretization. Additional 1$^{\text{st}}$-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 2$^{\text{nd}}$-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 [33], can be found in Kamenetskiy et al. [34].

The discrete nonlinear solver in the GGNS code implements a variant of the Newton-Krylov-Schwarz algorithm. On the code level, this is accomplished 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 [32]. 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.

The Mach Hessian for each element is evaluated from the flow solution by 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 [23], 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. (1). The global scaling factor, D_{L^p}, is initialized as Eq. (2). When enabled, the metric gradation is limited as detailed in the EPIC description. The complexity, Eq. (3), of the resulting elemental adaptive metric is computed and the global scale factor, D_{L^p}, is modified 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 [35] interpolation of the elemental metrics to the mesh vertices. The metric field and the mesh are passed to EPIC to adapt the mesh to conform to the metric.

The EPIC anisotropic mesh adaptation package developed at Boeing provides a modular framework for anisotropic mesh adaptation that can be linked with external flow solvers [36]. 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 EPIC includes edge insertion, edge collapse, element swaps, and vertex movement. 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.

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 [37] 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 2$^{\text{nd}}$-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 2$^{\text{nd}}$-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-neg 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 [23, 38].

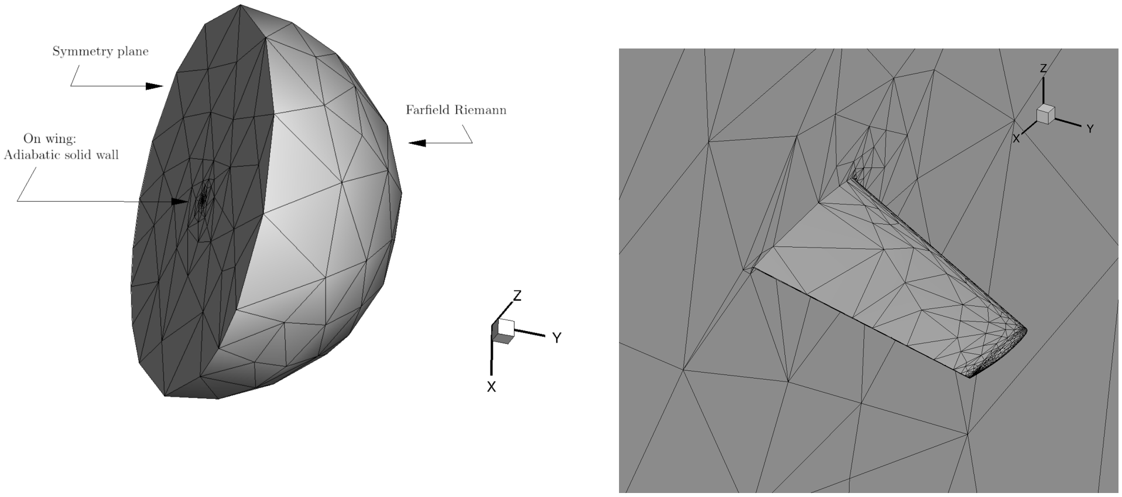
Recursive L^2-projection [23] is used for Hessian reconstruction. This recursive L^2-projection method is preferred over other reconstruction methods for improved robustness, efficiency, and accuracy. On the boundary, the reconstructed Hessian is replaced with an extrapolation from neighboring interior vertices, which have a well-formed stencil. In the presented results, the local Mach number is the scalar field and the multiscale metric [23] controls the interpolation error in the L^2- and L^4-norms. A metric gradation process is applied to smooth the metric field following the "mixed-space-gradation" approach of [39]. Then, the local remesher FEFLO.A is considered to adapt the mesh in order to conform to the final metric field.

FEFLO.A 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 [25, 40]. 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 geometry query.

IV. Numerical results: Viscous Transonic flow around the ONERA M6 wing

The ONERA M6 wing experiment was originally described in an AGARD report by Schmitt et al. [41]. The experimental data set consists of surface pressure coefficients at 7 spanwise locations. The experimental data set has been widely used for CFD validation studies due to the simple geometry of the M6 wing with complex flow features such as shock boundary layer interaction and flow separation. Recently, Mayeur et al. [42, 43] have created a new CAD geometry for the wing by modifying the geometry from the original description that has a moderately thick trailing edge to have a sharp trailing edge and a well-defined wingtip shape for RANS simulations.

For the present numerical study, the test case considered (case number 2308 in [41]) has a free stream Mach number of 0.84, angle-of-attack of 3.06° and a Reynolds number of 14.6×10^6 based on the root chord. This corresponds to an attached transonic flow case with double shocks merging to form a "lambda" shock on the upper surface of the wing. The initial coarse mesh, that has around 1500 vertices, is shown in Fig. 2. The farfield boundary is in the shape of a hemisphere located at 100 unit chords. The root chord is unity, the reference area is 1.1531508, and the reference pitching moment length is 0.80163742 with pitching moment computed about the leading edge of the root chord. Adiabatic no-slip bounday condition is imposed on the wing surface. A symmetry boundary condition is used on the plane extending from the wing root.



(a) Computational domain

(b) Wing and the symmetry plane

Fig. 2 ONERA M6 wing: Initial mesh.

Diskin et al. [11] have performed mesh convergence studies for this test case with various flow solvers on a family of uniformly refined, fixed meshes. The details about the fixed meshes are described by Nishikawa and Diskin [44]. These fixed meshes have well-resolved leading edge, trailing edge, wing tip and boundary layer. The finest fixed mesh consists of around 61 Million vertices. We plot the pressure and skin friction coefficients obtained on our adapted meshes along with those obtained on the finest fixed mesh. Drag, lift and pitching moment convergence are also shown along with those obtained on fixed meshes.

Figure 3 shows the mesh convergence of lift, total drag, pressure drag, viscous drag and the pitching moment on both the adapted meshes and the fixed meshes. The characteristic mesh spacing h is defined as $N^{-1/3}$, where N is the number of vertices in the mesh. In the legend, L2 Adapted (red lines) and L4 Adapted (blue lines) correspond to mesh adaptation based on L^2-norm and L^4-norm multiscale metrics, respectively. The last mesh at each fixed-complexity step is shown. All the forces and the pitching moment trajectories appear to be converging toward the fine fixed mesh values.

All forces and pitching moment from SUPG finite-element discretizations (FUN3D-SFE and GGNS) converge faster than those from finite-volume (FUN3D-FV) discretization for both L^2-norm and L^4-norm metrics. For the L^4-norm metric, the mixed finite-volume and finite-element (Wolf) discretization results are comparable to the FUN3D-SFE results. FUN3D-FV and Wolf solvers with L^2-norm adapted meshes are the slowest to converge for lift and pitching moment coefficients. In the case of lift, viscous drag and pitching moment, L^4-norm adapted meshes converge faster than both the L^2-norm adapted meshes and the fixed meshes for all the solvers. An exception is the pressure drag, where L^2-norm adapted meshes have a slightly better convergence than L^4-norm adapted ones for FUN3D-FV and FUN3D-SFE solvers. GGNS and Wolf show clear superiority of L^4-norm over L^2-norm adaptation for the convergence of all the forces and pitching moment coefficients. For all solvers, the most prominent difference between L^2- and L^4-norm adaptation is in the case of viscous drag convergence. This can be explained due to the fact that a higher norm in the multiscale metric targets regions with higher variation in the solution [13]. The viscous layer, with high gradients, get refined more with the L^4-norm metric resulting in faster viscous drag convergence. Fig. 4 shows meshes, having almost the same number of nodes (122k), from adaptations using the L^2-norm (left) and L^4-norm (right) metric. With the L^4-norm, there is a more aggresive refinement near and on the wing surface. In other words, at similar complexity levels, more mesh resolution is available to support the solution near the wing if it is not being used to resolve the wake multiple chord lengths downstream of the wing as in the case of L^2-norm adapted meshes.

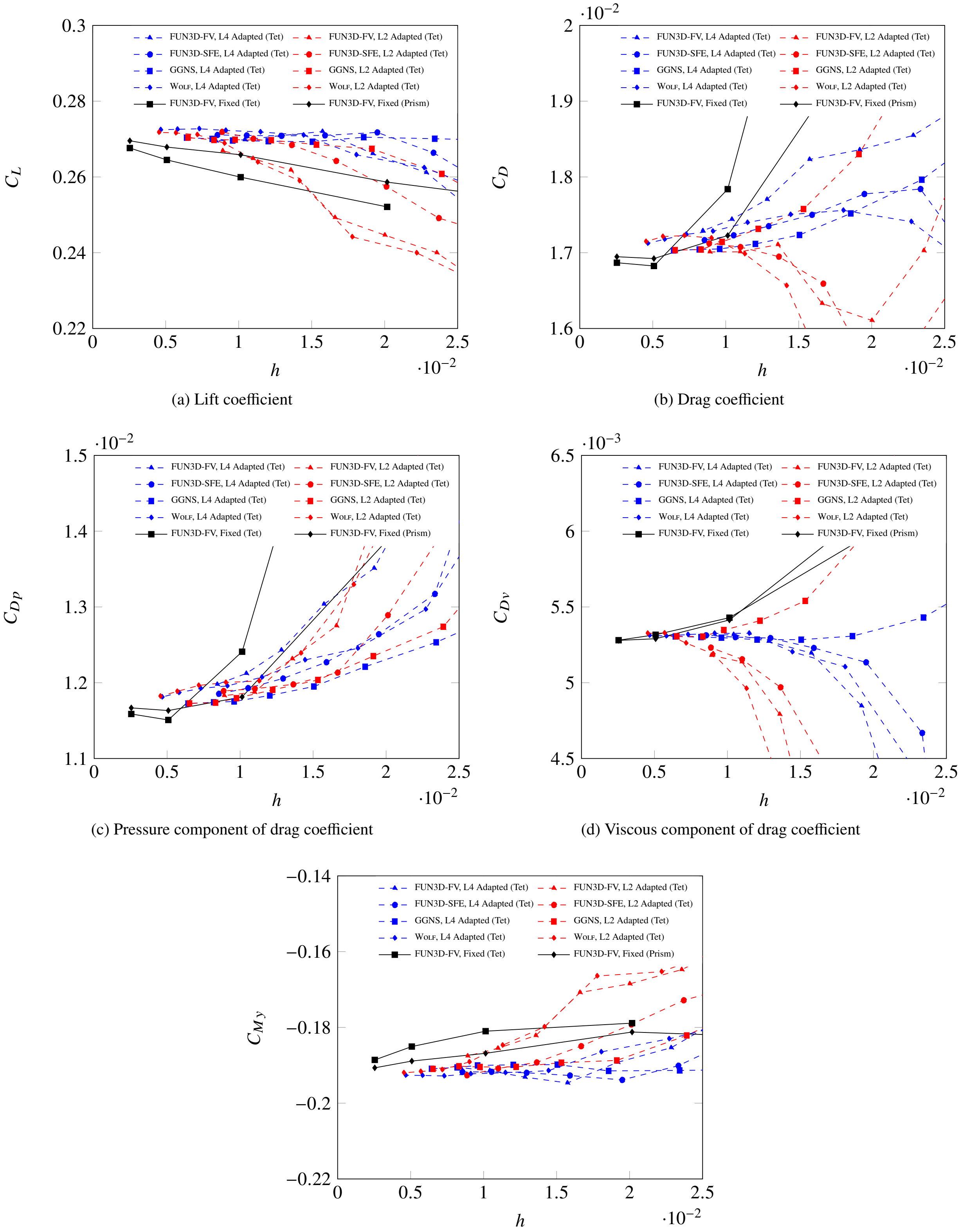
For each solver, three adapted meshes of increasing resolution are chosen to plot surface pressure and skin friction coefficients. Table 1 shows the number of vertices for each mesh for each solver. G1 is the finest and G3 is the coarsest among the chosen meshes. Fig. 5 shows the surface mesh on the wing and symmetry plane, and the pressure contours on these meshes for the FUN3D-SFE solver. With adaptation, all the critical regions including the wing surface, wake and the shock region are refined.

Table 1 Mesh statistics for plotting pressure and skin friction. Values shown are the number of vertices in the meshes.

Mesh	FUN3D-FV	FUN3D-SFE	GGNS	WOLF
G1	1,667,340	1,607,681	1,820,907	1,328,707
G2	473,629	462,074	574,179	659,005
G3	141,725	135,073	156,313	169,826

Table 2 Meaurement locations for plotting coefficients of pressure and skin friction. Wingspan, b = 1.47601797621980.

$\eta = y/b$	y
0.20	0.295203595243960
0.44	0.649447909536712
0.65	0.959411684542870
0.80	1.18081438097584
0.90	1.32841617859782
0.96	1.41697725717101
0.99	1.46125779645760



(a) Lift coefficient

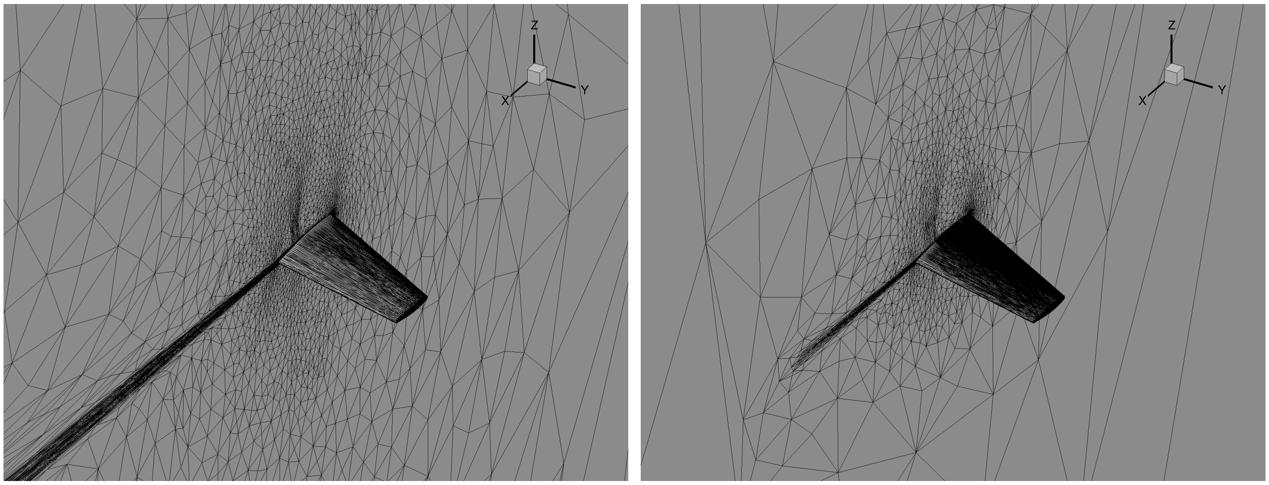
(b) Drag coefficient

(c) Pressure component of drag coefficient

(d) Viscous component of drag coefficient

(e) Pitching moment

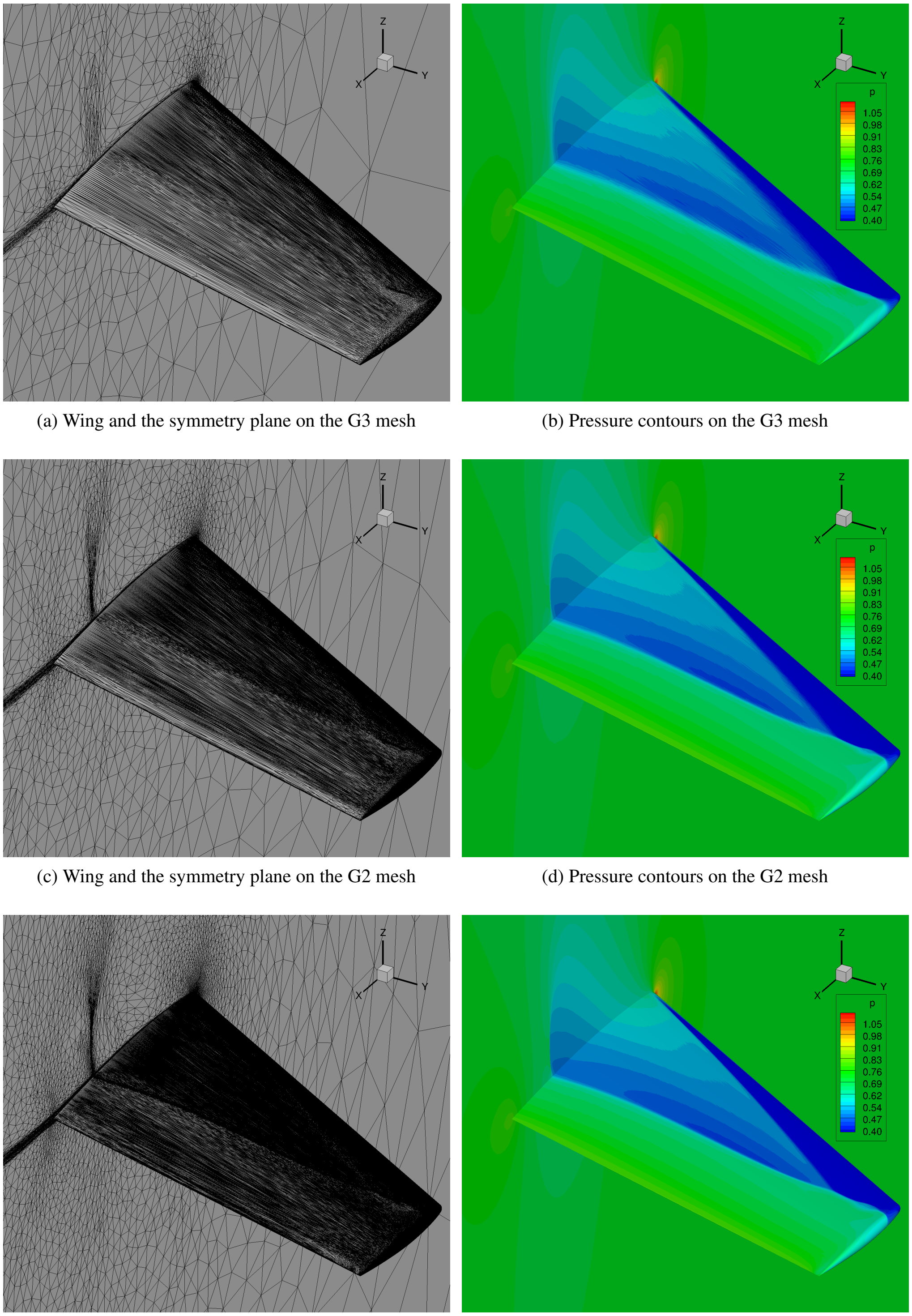
Fig. 3 ONERA M6 wing: Mesh convergence of aerodynamic force coefficients and pitching moment. Dashed lines: Adapted Meshes, Solid Lines: Fixed Meshes.



(a) L^2-norm adapted mesh

(b) L^4-norm adapted mesh

Fig. 4 ONERA M6 wing: Comparison of L^2-norm and L^4-norm multiscale metric adapted meshes. Both meshes have almost same number of nodes (122k), and FUN3D-SFE is used as solver.



(a) Wing and the symmetry plane on the G3 mesh

(b) Pressure contours on the G3 mesh

(c) Wing and the symmetry plane on the G2 mesh

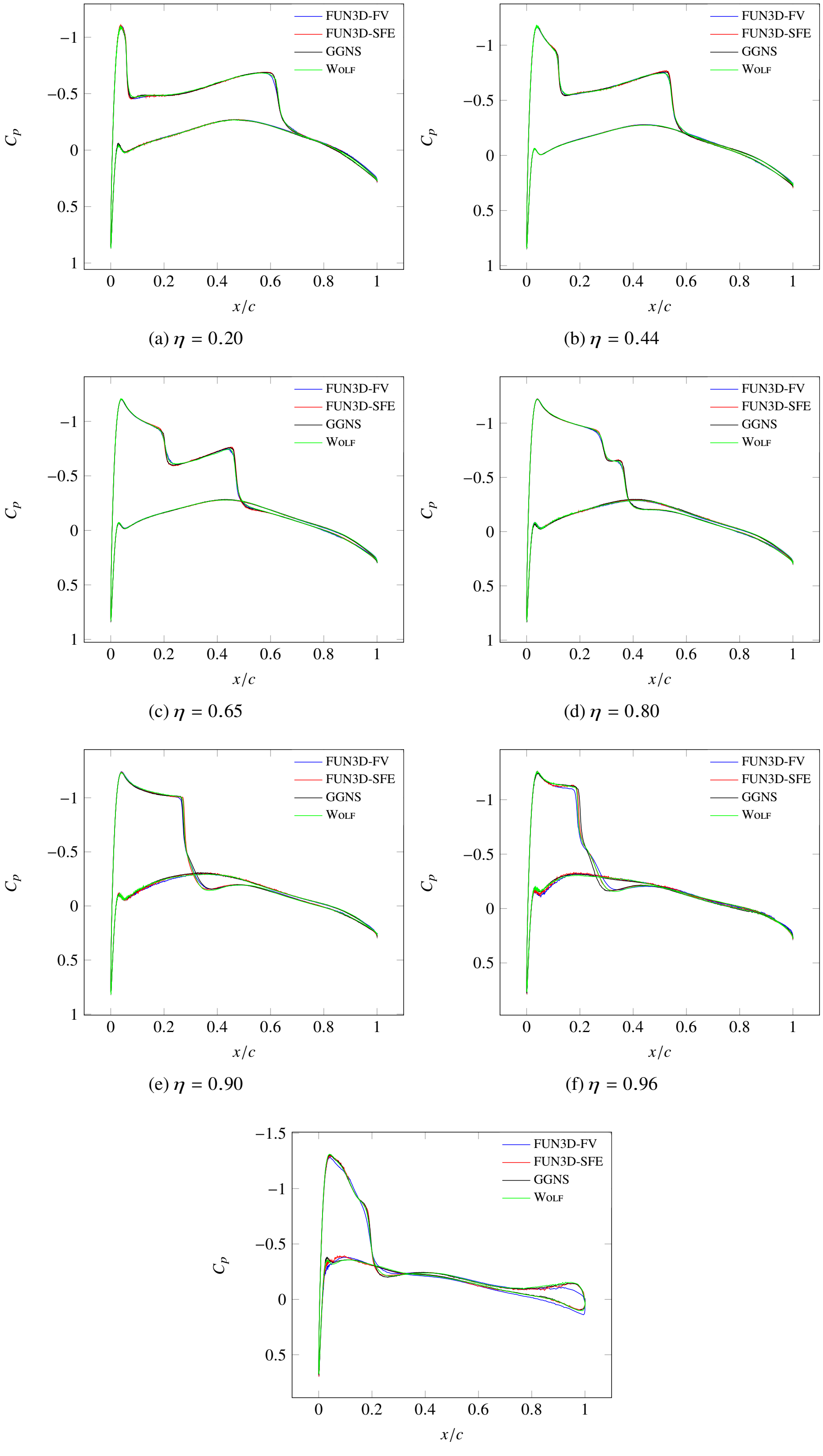
(d) Pressure contours on the G2 mesh

(e) Wing and the symmetry plane on the G1 mesh

(f) Pressure contours on the G1 mesh

Fig. 5 ONERA M6 wing: Surface meshes and pressure contours using FUN3D-SFE on G3 (coarse), G2 (medium) and G1 (fine) adapted meshes.

Figure 6 shows the surface pressure coefficients, obtained using the various solvers on their respective G1 meshes at various spanwise locations that are specified in Table 2 . The values are remarkably close between different solvers for most of the spanwise locations. A slight variation can be seen at the merged shock region on the upper wing at $\eta = 0.96$. At $\eta = 0.99$, the solution from FUN3D-FV shows a slight variation from other solvers, especially at the trailing edge. All four solutions have the same shock locations and pressure minima on the lower and upper wing surfaces. The plateau region between the shocks on the upper surface at $\eta = 0.88$ is well resolved, indicating sufficient mesh resolution at the region.



(a) η = 0.20

(b) η = 0.44

(c) η = 0.65

(d) η = 0.80

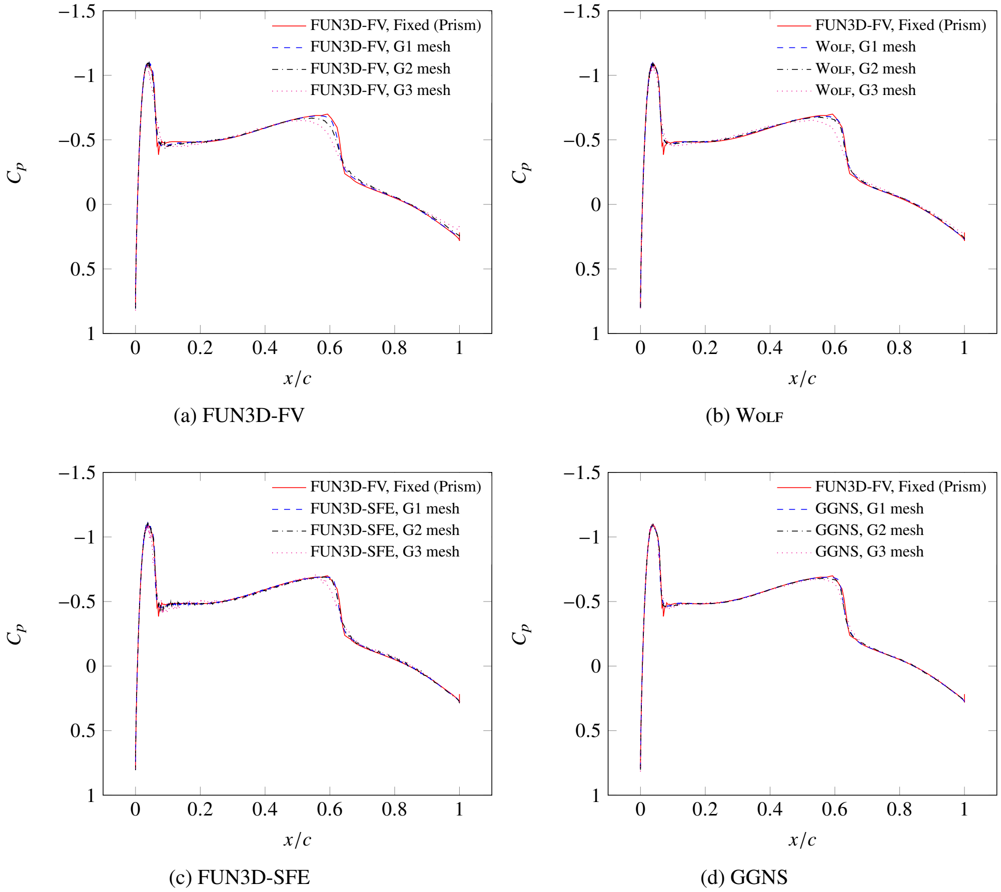
(e) η = 0.90

(f) η = 0.96

(g) η = 0.99

Fig. 6 ONERA M6 wing: Surface pressure distribution at various stations on the wing obtained by various solvers on their respective G1 (fine) adapted meshes.

Figure 7 shows the pressure coefficients on the upper wing surface at $\eta = 0.20$ for all the four solvers on their respective G1, G2 and G3 meshes. The pressure coefficients near the location of the minimum pressure, and near the trailing edge, are shown in Fig. 8 and Fig. 9, respectively. Solutions from a fine fixed mesh (61 million vertices) using FUN3D-FV are also shown for comparison. In the global view, the solutions from all four solvers appear to converge to the fine fixed mesh solution. Solutions on G1 meshes for all the solvers are in close agreement with that on the fixed mesh. Solutions on G3, the coarsest meshes, give diffused shocks on the trailing edge side; however, overall they follow the pressure curve well. Near the minimum pressure location, solutions from both G1 and G2 meshes are very close to the fine fixed mesh results (Fig. 8). On G3 meshes, the solution from GGNS solver is the closest to the fine fixed mesh values. Near the trailing edge (see Fig. 9), the solutions from all solvers seem to be converging to the fine fixed mesh solution. On both G2 and G3 meshes, the solutions from SUPG finite-element solvers (FUN3D-SFE and GGNS) are closer to the fine values compared to those from other two solvers (FUN3D-FV and WOLF).



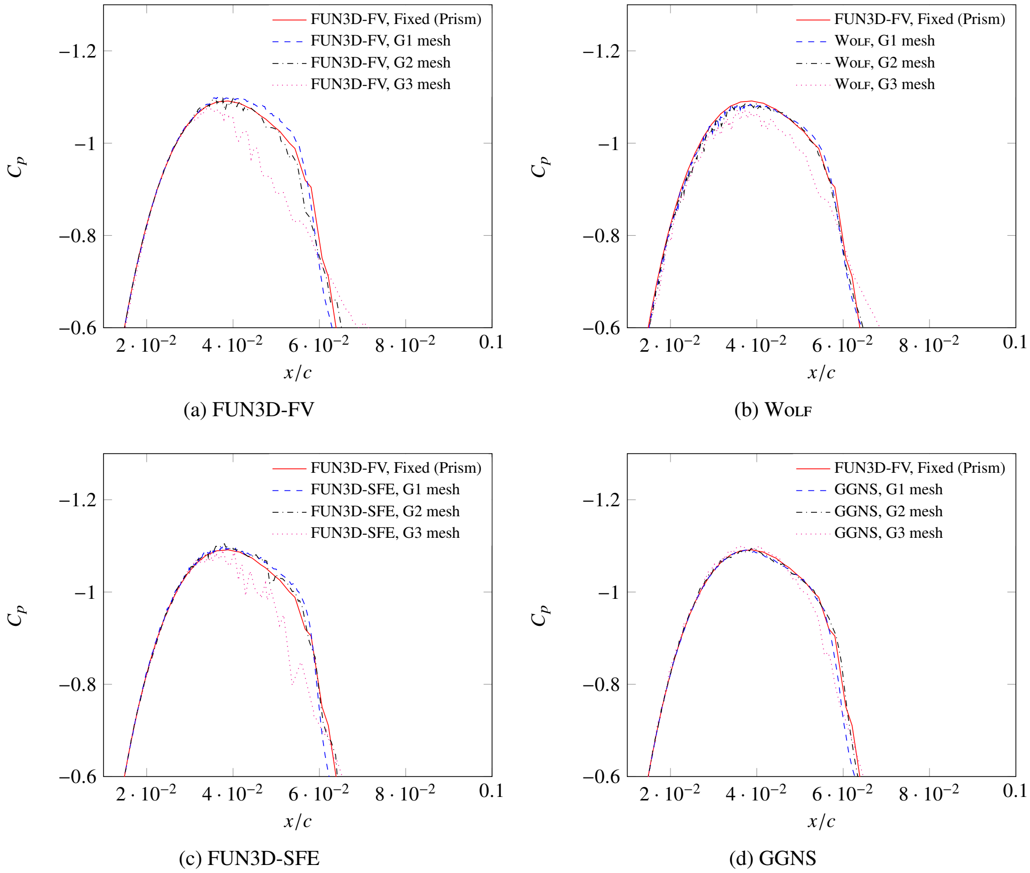
(a) FUN3D-FV

(b) WOLF

(c) FUN3D-SFE

(d) GGNS

Fig. 7 ONERA M6 wing: Surface pressure distribution on the wing at $\eta = 0.20$ obtained by various solvers on their respective G1 (fine), G2 (medium) and G3 (coarse) adapted meshes.



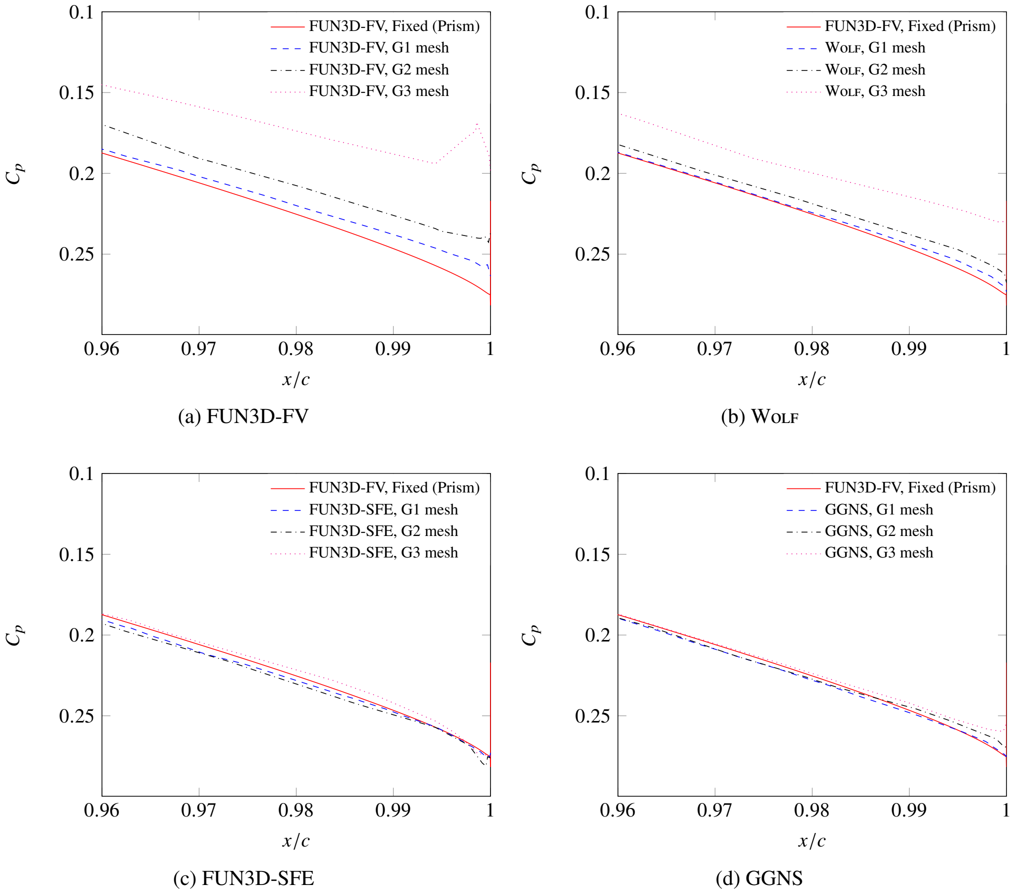
(a) FUN3D-FV

(b) Wolf

(c) FUN3D-SFE

(d) GGNS

Fig. 8 ONERA M6 wing: Surface pressure distribution on the wing at $\eta = 0.20$, near minimum pressure location, obtained by various solvers on their respective G1 (fine), G2 (medium) and G3 (coarse) adapted meshes.



(a) FUN3D-FV

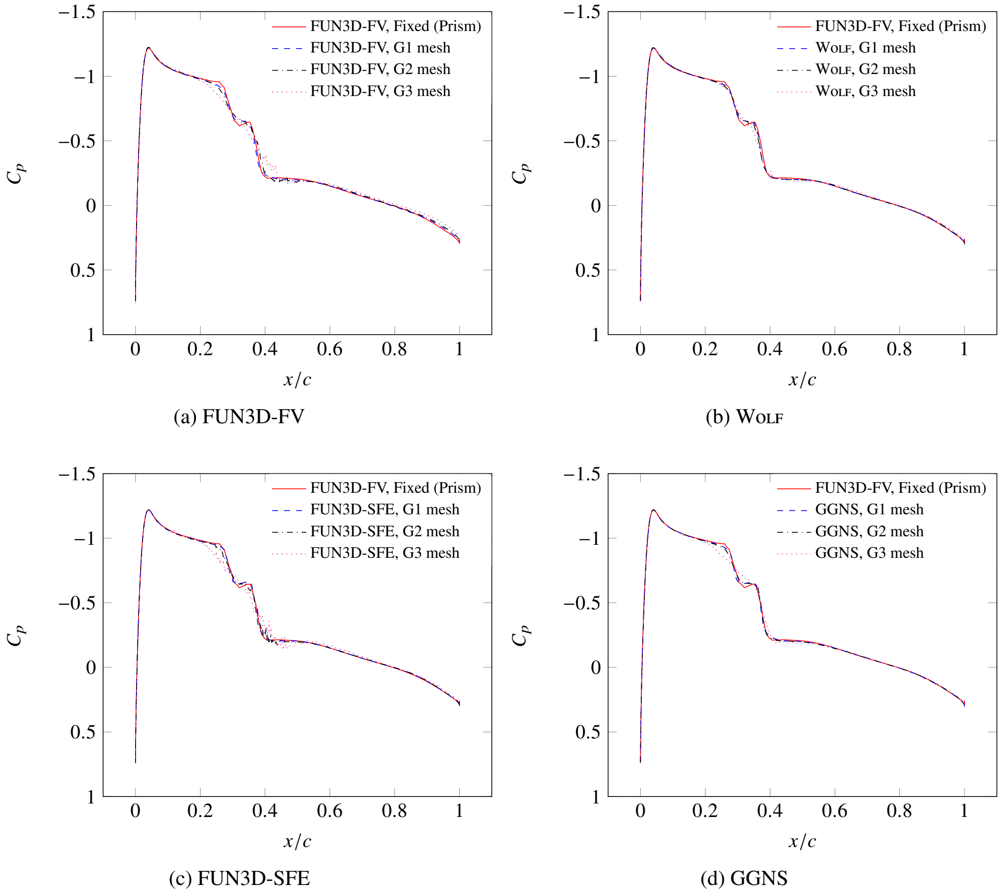
(b) WOLF

(c) FUN3D-SFE

(d) GGNS

Fig. 9 ONERA M6 wing: Surface pressure distribution on the wing at $\eta = 0.20$, near trailing edge, obtained by various solvers on their respective G1 (fine), G2 (medium) and G3 (coarse) adapted meshes.

Figure 10 shows the pressure coefficients on the upper wing surface at $\eta = 0.80$ for all four solvers on their respective G1, G2 and G3 meshes. A closer view of the pressure coefficients near the shock intersection region is shown in Fig. 11. In the global view, all the solvers on both G1 and G2 meshes give solutions that match reasonably well with the fine fixed mesh values. They could capture the double shock structure and the plateau region between the shocks. The solutions on G3 meshes, however, completely miss the double shock and the plateau region for all the solvers (see Fig. 11). On G1 and G2 meshes, both Wolf and GGNS results look similar and follow the fine mesh values closely. On G3 meshes, Wolf results are the closest to the fine mesh values.



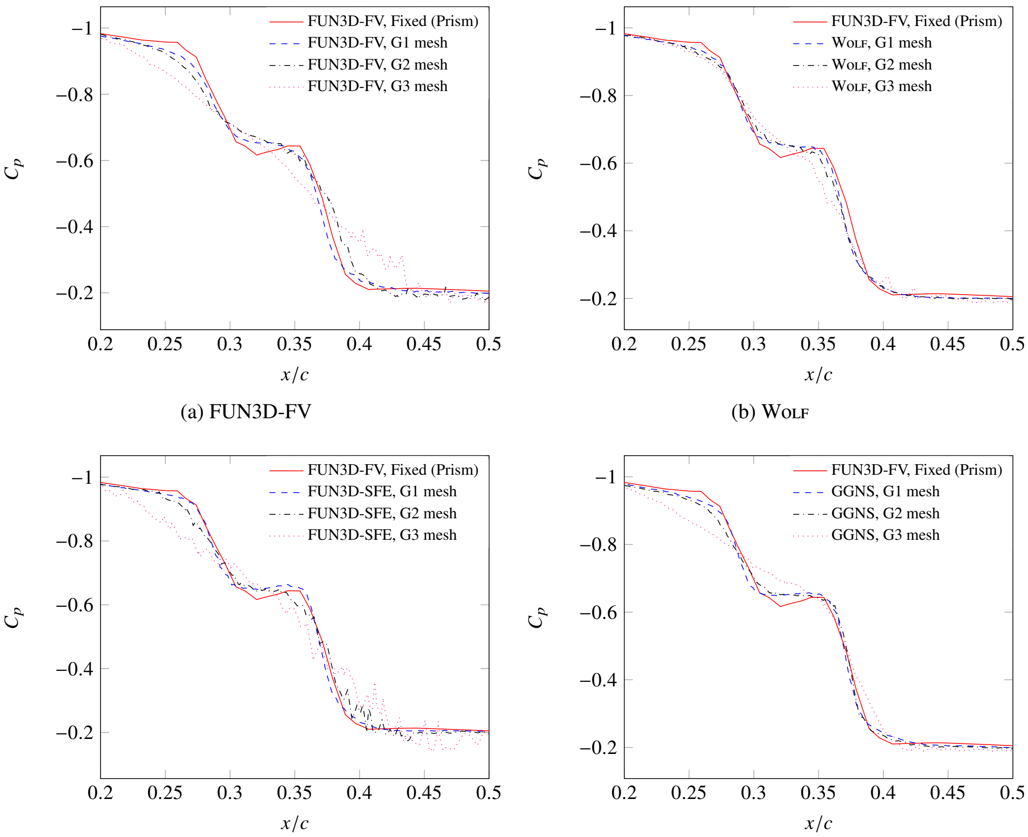
(a) FUN3D-FV

(b) Wolf

(c) FUN3D-SFE

(d) GGNS

Fig. 10 ONERA M6 wing: Surface pressure distribution on the wing at $\eta = 0.80$ obtained by various solvers on their respective G1 (fine), G2 (medium) and G3 (coarse) adapted meshes.



(a) FUN3D-FV

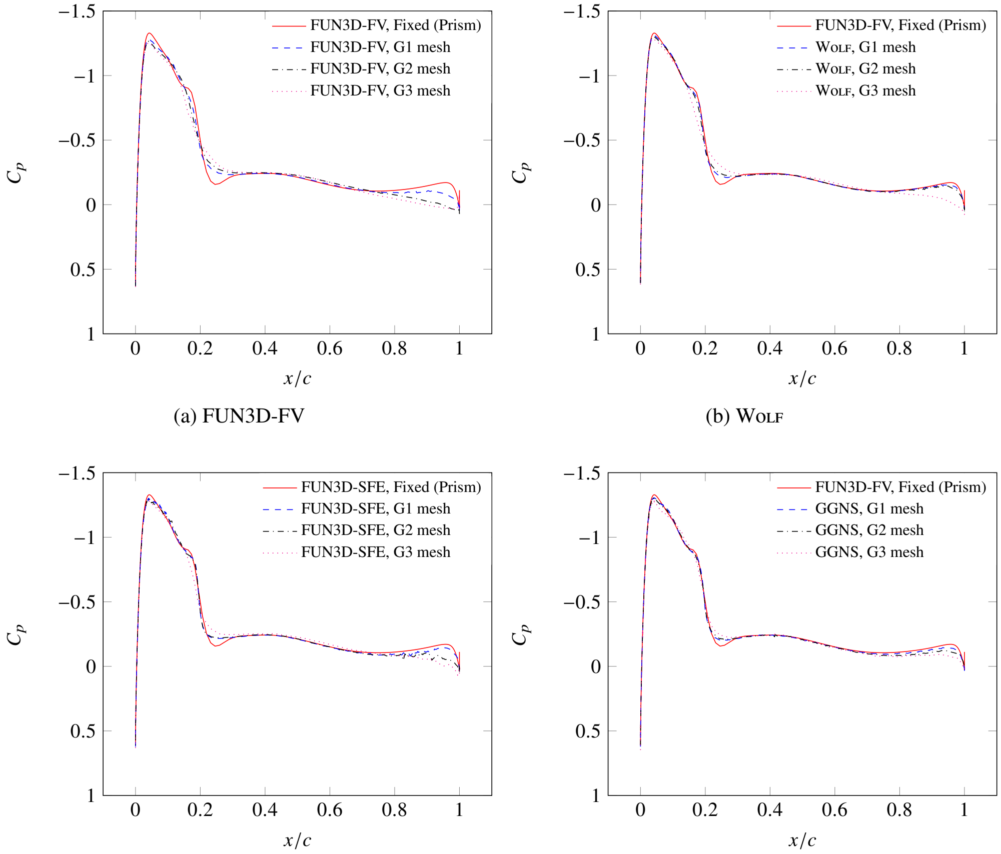
(b) Wolf

(c) FUN3D-SFE

(d) GGNS

Fig. 11 ONERA M6 wing: Surface pressure distribution on the wing at $\eta = 0.80$, near shock intersection, obtained by various solvers on their respective G1 (fine), G2 (medium) and G3 (coarse) adapted meshes.

Figure 12 shows the pressure coefficients on the upper wing surface at $\eta = 0.99$ for all the four solvers on their respective G1, G2 and G3 meshes. The pressure coefficients near the location of the minimum pressure, and near the trailing edge, are shown in Fig. 13 and Fig. 14, respectively. The solutions for all the solvers appear to be converging to the fine fixed mesh values with adaptation from G3 to G1 meshes. The resolution of the G1 mesh still does not seem to be sufficient to capture the dip in the curve around $x/c \approx 0.23$. Near the minimum pressure location, increased mesh resolution is required to get closer to the fine fixed mesh values. Upstream of the minimum pressure location, Wolf results seem to be matching the most with the fixed mesh. Near the trailing edge, the values are seen to be converging toward the fixed mesh values. On G1 meshes, compared to the fixed mesh values, Wolf is the closest and FUN3D-FV is the farthest.



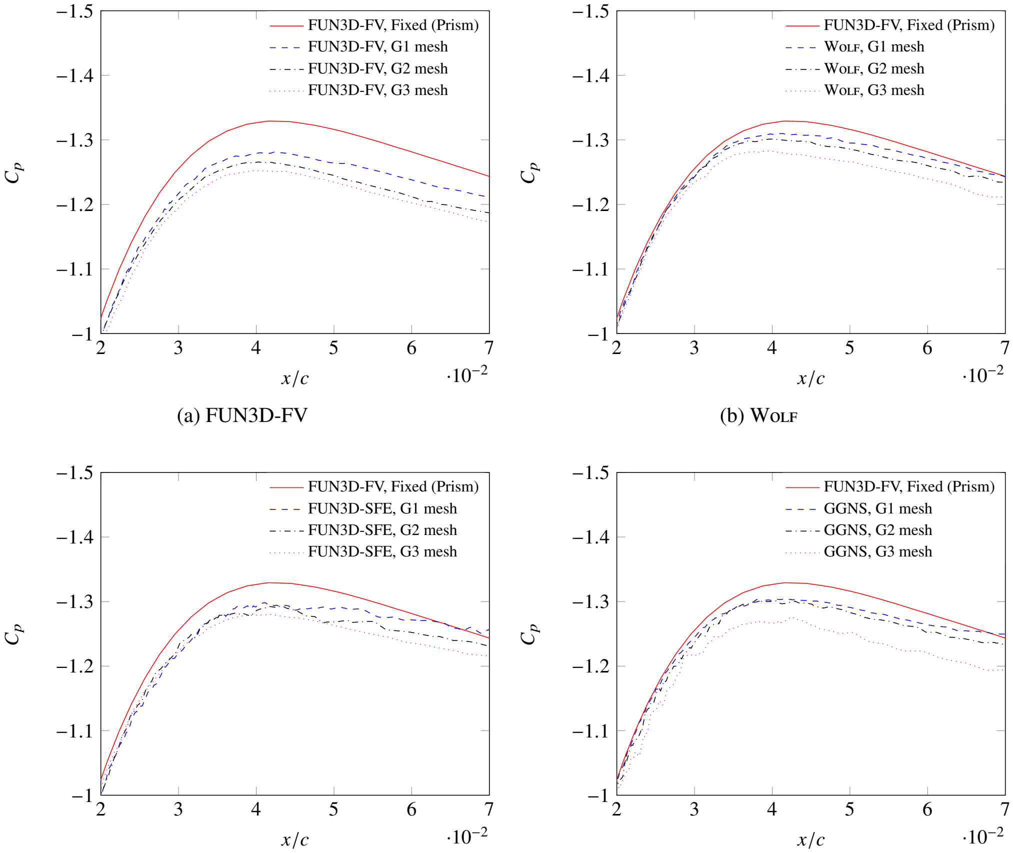
(a) FUN3D-FV

(b) Wolf

(c) FUN3D-SFE

(d) GGNS

Fig. 12 ONERA M6 wing: Surface pressure distribution on the wing at $\eta = 0.99$ obtained by various solvers on their respective G1 (fine), G2 (medium) and G3 (coarse) adapted meshes.



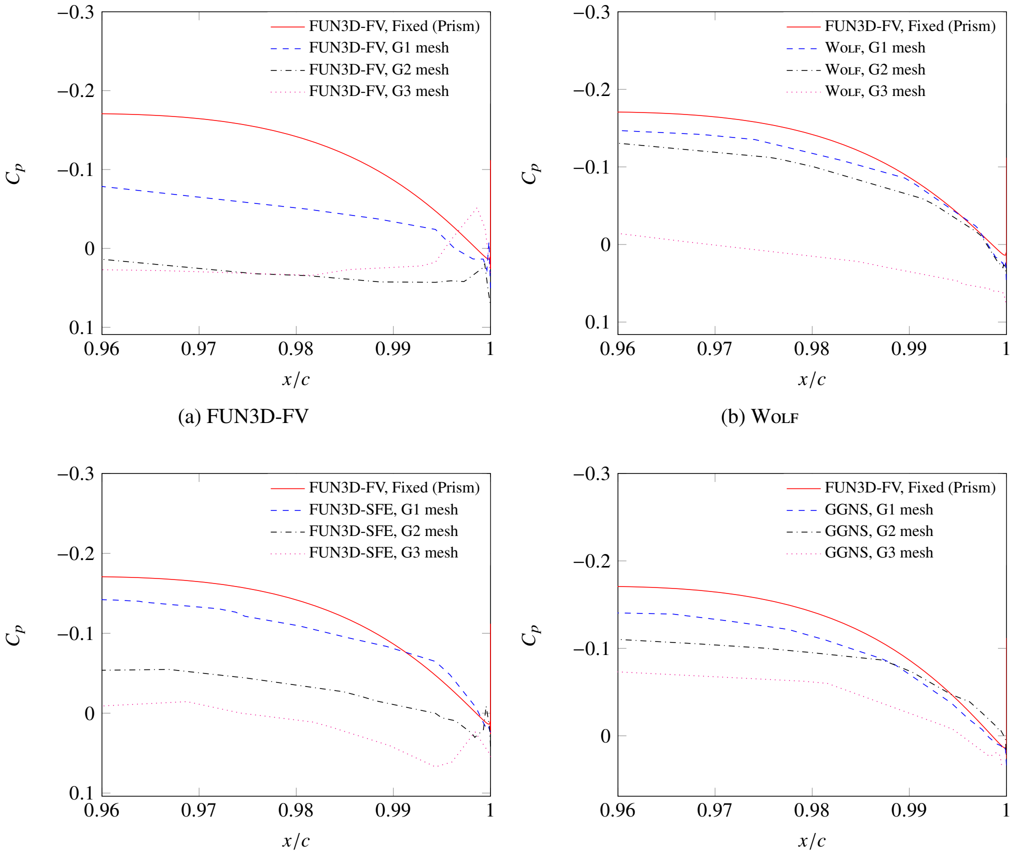
(a) FUN3D-FV

(b) Wolf

(c) FUN3D-SFE

(d) GGNS

Fig. 13 ONERA M6 wing: Surface pressure distribution on the wing at $\eta = 0.99$, near minimum pressure location, obtained by various solvers on their respective G1 (fine), G2 (medium) and G3 (coarse) adapted meshes.



(a) FUN3D-FV

(b) Wolf

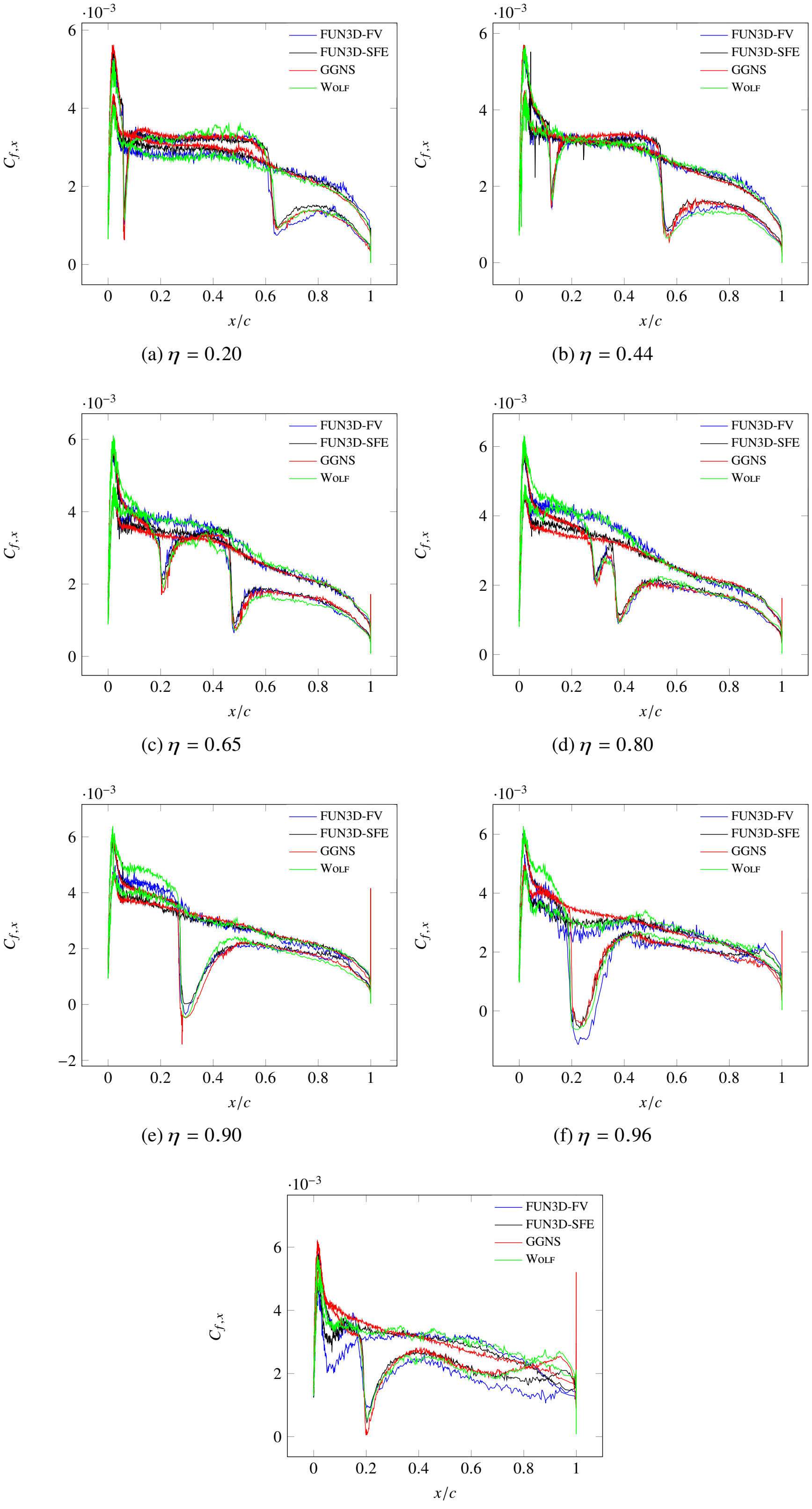
(c) FUN3D-SFE

(d) GGNS

Fig. 14 ONERA M6 wing: Surface pressure distribution on the wing at $\eta = 0.99$, near trailing edge, obtained by various solvers on their respective G1 (fine), G2 (medium) and G3 (coarse) adapted meshes

Figure 15 shows the skin friction coefficients, obtained using the various solvers on their respective G1 meshes at various spanwise locations. The coefficients are seen to be highly oscillatory for all the solvers. At all seven spanwise locations, all the solutions place the maximas and minimas almost at the same locations. At $\eta = 0.44$, FUN3D-SFE has a very high oscillation downstream of the leading edge that is not seen in other solutions. Solution variation between the solvers become more pronounced at the outboard locations compared to that of inboard locations. At $\eta = 0.99$, the variation is highest slightly downstream of the leading edge at around $x/c \approx 0.1$, and also near the trailing edge.

Figure 16 shows the convergence of skin friction coefficient for various solvers at $\eta = 0.20$. On the coarse G3 meshes, all solvers except GGNS exhibit very high oscillations. GGNS has solutions that agree to each other well on G1, G2 and G3 meshes (see Fig. 16 d). Both FUN3D-FV and Wolf solutions show an upward trend in skin friction from $x/c \approx 0.1$ to $x/c \approx 0.6$. Figure 17 shows the convergence of skin friction coefficient for various solvers at $\eta = 0.44$. GGNS again has the least oscillations on G3 meshes, and the solutions from G1, G2 and G3 meshes agree well with each other. Downstream of the leading edge from $x/c \approx 0.02$ to $x/c \approx 0.2$, compared to the finite-volume (FUN3D-FV) and mixed finite-volume and finite-element (Wolf) solutions, finite-element solutions match better with the fixed mesh results. Figure 18 shows the convergence of skin friction coefficient for various solvers at $\eta = 0.99$. On G3 meshes, all the solvers except GGNS fail to follow the fine fixed mesh curve in most part. With adaptations, the values, however, get closer to the fixed mesh values. The closest agreement with fixed mesh values is for GGNS, and the largest disagreement is for FUN3D-FV.



(a) $\eta = 0.20$

(b) $\eta = 0.44$

(c) $\eta = 0.65$

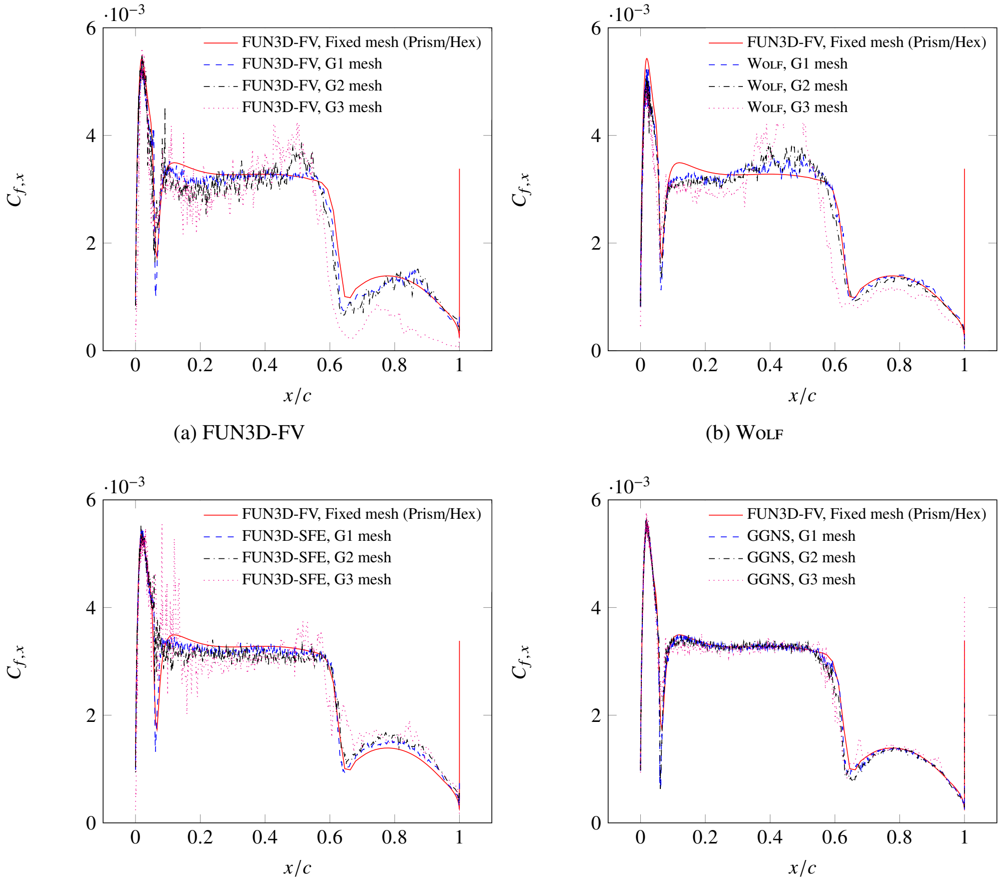
(d) $\eta = 0.80$

(e) $\eta = 0.90$

(f) $\eta = 0.96$

(g) $\eta = 0.99$

Fig. 15 ONERA M6: Skin friction, $C_{f,x}$ distribution at various stations on the wing obtained by various solvers on their respective G1 (fine) adapted meshes.



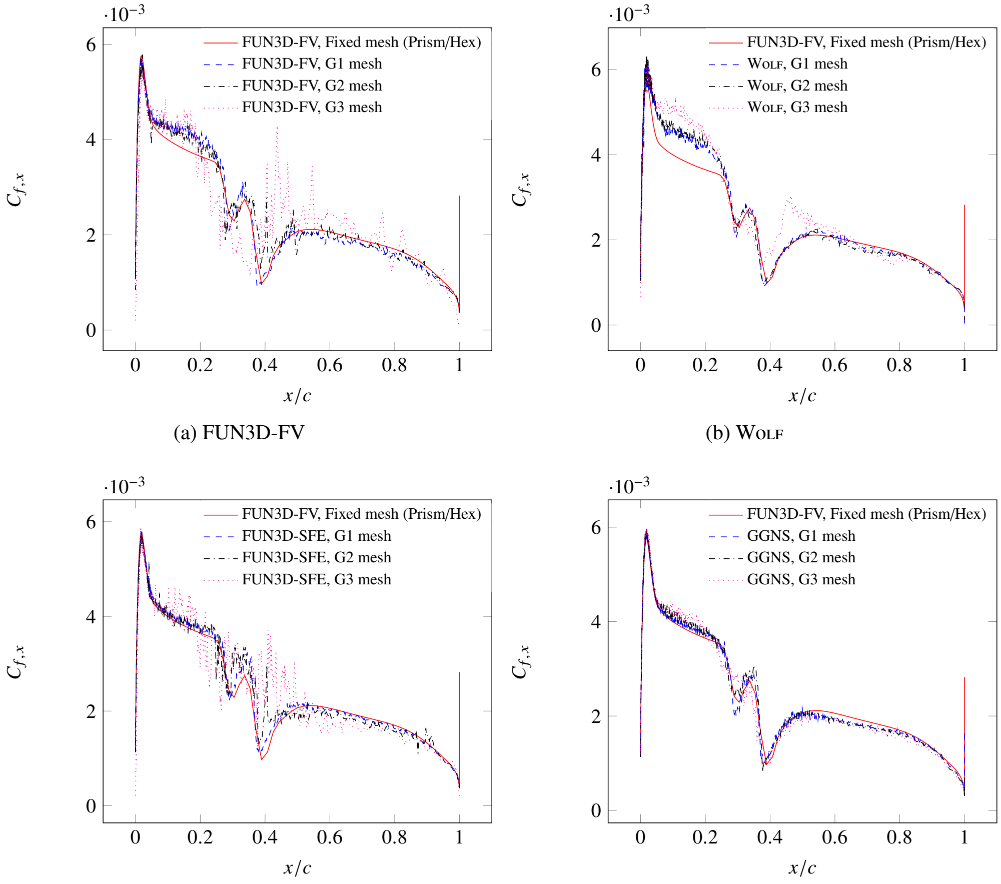
(a) FUN3D-FV

(b) Wolf

(c) FUN3D-SFE

(d) GGNS

Fig. 16 ONERA M6: Skin friction, $C_{f,x}$ distribution on the wing at $\eta = 0.20$ obtained by various solvers on their respective G1 (fine), G2 (medium) and G3 (coarse) adapted meshes.



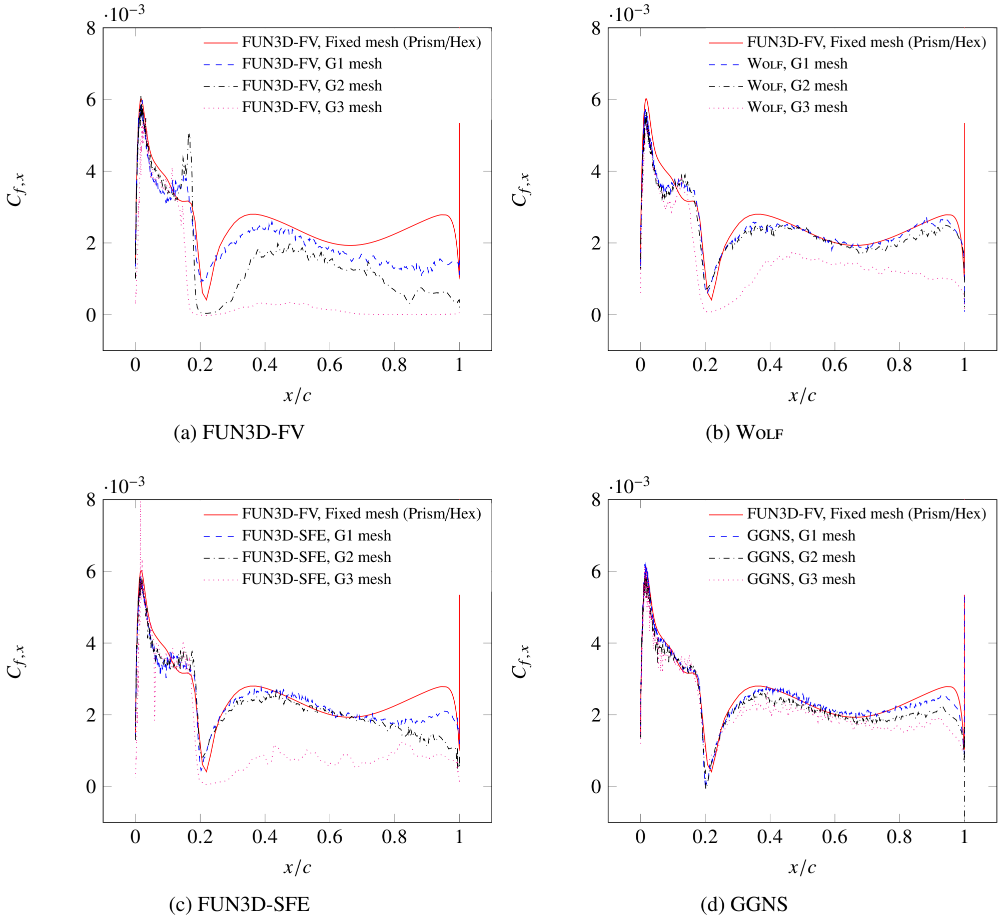
(a) FUN3D-FV

(b) WOLF

(c) FUN3D-SFE

(d) GGNS

Fig. 17 ONERA M6: Skin friction, $C_{f,x}$ distribution on the wing at $\eta = 0.80$ obtained by various solvers on their respective G1 (fine), G2 (medium) and G3 (coarse) adapted meshes.



(a) FUN3D-FV

(b) WOLF

(c) FUN3D-SFE

(d) GGNS

Fig. 18 ONERA M6: Skin friction, $C_{f,x}$ distribution on the wing at $\eta = 0.99$ obtained by various solvers on their respective G1 (fine), G2 (medium) and G3 (coarse) adapted meshes.

V. Conclusions

The verification of anisotropic mesh adaptation processes using various solvers and mesh-mechanics packages on benchmark CFD problems are critical to establish confidence in these procedures. To that end, we performed anisotropic mesh adaptation for RANS turbulent flow simulations over the ONERA M6 wing, a well-known test case for verification and validation studies, using various solvers and mesh-mechanics packages. The adaptation aimed at controlling the interpolation error in Mach number through the use of a multiscale metric field to generate the meshes. Mesh adaptation automatically establishes appropriate resolution of important smooth features and nonsmooth features (e.g., shocks, boundary layer). We have used two finite-element solvers (FUN3D-SFE and GGNS), one finite-volume solver (FUN3D-FV) and a mixed finite-volume and finite-element solver (Wolf). Two sets of adaptations were done; one that controls the interpolation error in the L^2 norm and the other that controls the interpolation error in the L^4 norm. The forces and pitching moment convergent studies were performed with adaptation using all four solvers. The values were compared against those reported on fine fixed meshes (both prism and tetrahedra) using the FUN3D-FV as the solver. All the forces and pitching moment trajectories appeared to be converging toward the fixed mesh values. Both forces and pitching moment on adapted meshes were found to be converging to the fine mesh values faster than those on fixed meshes. The values from finite-element discretizations converged faster than those from finite-volume and mixed finite-volume and finite-element discretizations. In the case of lift, viscous drag and pitching moment, L^4-norm adapted meshes converge faster than both the L^2-norm adapted meshes and the fixed meshes for all the solvers. For each solver, three adapted meshes (denoted as G1, G2 and G3) of increasing resolution were chosen to study the convergence of surface pressure and skin friction coefficients with adaptation. With adaptation, the values seemed to be converging toward the fine fixed mesh values. On G1 meshes, the finest among three meshes, the pressure coefficients were remarkably close to each other at most of the measurement spanwise location. All four solutions have the same shock locations and pressure minima on the lower and upper wing surfaces. The skin friction coefficients were seen to be oscillatory for all the solvers with mean values that approach the fine fixed mesh solution agreeing on the locations of maximas and the minimas. Future work would aim at the verification of the adaptation process for more complicated geometries, using metrics that control interpolation errors and output functional errors.

Acknowledgements

This work was partially sponsored by the NASA Revolutionary Computational Aerosciences (RCA) Transformational Tools and Technologies (TTT) Project of the NASA Transformative Aeronautics Concepts Program (TACP).

References

[1] Slotnick, J., Khodadoust, A., Alonso, J., Darmofal, D., Gropp, W., Lurie, E., and Mavriplis, D., "CFD Vision 2030 Study: A Path to Revolutionary Computational Aerosciences," NASA CR-2014-218178, Langley Research Center, Mar. 2014.

doi:2060/20140003093.

[2] Rumsey, C. L., “Recent Developments on the Turbulence Modeling Resource Website,” AIAA Paper 2015–2927, 2015.

[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. doi:10.1016/j.jcp.2009.12.021.

[4] 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.

[5] Alauzet, F., and Loseille, A., “A Decade of Progress on Anisotropic Mesh Adaptation for Computational Fluid Dynamics,” *Computer-Aided Design*, Vol. 72, 2016, pp. 13–39. doi:10.1016/j.cad.2015.09.005, 23rd International Meshing Roundtable Special Issue: Advances in Mesh Generation.

[6] Park, M. A., Krakos, J. A., Michal, T., Loseille, A., and Alonso, J. J., “Unstructured Grid Adaptation: Status, Potential Impacts, and Recommended Investments Toward CFD Vision 2030,” AIAA Paper 2016–3323, 2016.

[7] 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.

[8] Ibanez, D., Barral, N., Krakos, J., Loseille, A., Michal, T., and Park, M., “First Benchmark of the Unstructured Grid Adaptation Working Group,” *Procedia Engineering*, Vol. 203, 2017, pp. 154–166. doi:10.1016/j.proeng.2017.09.800, 26th International Meshing Roundtable, IMR26, 18-21 Sept. 2017, Barcelona, Spain.

[9] Park, M. A., Barral, N., Ibanez, D., Kamenetskiy, D. S., Krakos, J., Michal, T., and Loseille, A., “Unstructured Grid Adaptation and Solver Technology for Turbulent Flows,” AIAA Paper 2018–1103, 2018.

[10] Park, M. A., Balan, A., Anderson, W. K., Galbraith, M. C., Caplan, P. C., Carson, H. A., Michal, T., Krakos, J. A., Kamenetskiy, D. S., Loseille, A., Alauzet, F., Frazza, L., and Barral, N., “Verification of Unstructured Grid Adaptation Components,” AIAA Paper 2019–1723, 2019.

[11] Diskin, B., Anderson, W. K., Pandya, M. J., Rumsey, C. L., Thomas, J. L., Liu, Y., and Nishikawa, H., “Grid Convergence for Three Dimensional Benchmark Turbulent Flows,” AIAA Paper 2018–1102, 2018.

[12] Loseille, A., and Alauzet, F., “Continuous Mesh Framework Part I: Well-Posed Continuous Interpolation Error,” *SIAM Journal on Numerical Analysis*, Vol. 49, No. 1, 2011, pp. 38–60. doi:10.1137/090754078.

[13] 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.

[14] 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. doi:10.1016/0021-9991(87)90093-3.

[15] 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.

[16] Michal, T., Krakos, J., and Kamenetskiy, D., "Generation of Anisotropic Adaptive Meshes for the First AIAA Geometry and Mesh Generation Workshop," AIAA Paper 2018–658, 2018.

[17] Michal, T., Kamenetskiy, D., and Krakos, J., "Anisotropic Adaptive Mesh Results for the Third High Lift Prediction Workshop (HiLiftPW-3)," AIAA Paper 2018–1257, 2018.

[18] Anderson, W. K., and Bonhaus, D. L., "An Implicit Upwind Algorithm for Computing Turbulent Flows on Unstructured Grids," *Computers and Fluids*, Vol. 23, No. 1, 1994, pp. 1–22. doi:10.1016/0045-7930(94)90023-X.

[19] Biedron, R. T., Carlson, J.-R., Derlaga, J. M., Gnoffo, P. A., Hammond, D. P., Jones, W. T., Kleb, B., Lee-Rausch, E. M., Nielsen, E. J., Park, M. A., Rumsey, C. L., Thomas, J. L., and Wood, W. A., "FUN3D Manual: 13.3," NASA TM-2018-219808, Langley Research Center, Feb. 2018. doi:2060/20180001971.

[20] Roe, P. L., "Approximate Riemann Solvers, Parameter Vectors, and Difference Schemes," *Journal of Computational Physics*, Vol. 43, No. 2, 1981, pp. 357–372. doi:10.1016/0021-9991(81)90128-5.

[21] Nielsen, E. J., Lu, J. C.-C., Park, M. A., and Darmofal, D. L., "An Implicit, Exact Dual Adjoint Solution Method for Turbulent Flows on Unstructured Grids," *Computers and Fluids*, Vol. 33, No. 9, 2004, pp. 1131–1155.

[22] Allmaras, S. R., Johnson, F. T., and Spalart, P. R., "Modifications and Clarifications for the Implementation of the Spalart-Allmaras Turbulence Model," *Seventh International Conference on Computational Fluid Dynamics (ICCFD7)*, 2012.

[23] Alauzet, F., and Loseille, A., "High-Order Sonic Boom Modeling Based on Adaptive Methods," *Journal of Computational Physics*, Vol. 229, No. 3, 2010, pp. 561–593. doi:10.1016/j.jcp.2009.09.020.

[24] Park, M. A., "Anisotropic Output-Based Adaptation with Tetrahedral Cut Cells for Compressible Flows," Ph.D. thesis, Massachusetts Institute of Technology, Sep. 2008. doi:1721.1/46363.

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

[26] 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.

[27] Freitag, L. A., and Ollivier-Gooch, C., "Tetrahedral Mesh Improvement Using Swapping and Smoothing," *International Journal for Numerical Methods in Engineering*, Vol. 40, 1997, pp. 3979–4002. doi:10.1002/(SICI)1097-0207(19971115)40:21<3979::AID-NME251>3.0.CO;2-9.

[28] Haimes, R., and Dannenhoffer, J. F., III, "EGADSlite: A Lightweight Geometry Kernel for HPC," AIAA Paper 2018–1401, 2018.

[29] 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.

[30] 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.

[31] 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.

[32] Saad, Y., *Iterative Methods for Sparse Linear Systems*, 2$^{\mathrm{nd}}$ ed., Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2003.

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

[34] Kamenetskiy, D. S., Bussoletti, J. E., Hilmes, C. L., Venkatakrishnan, V., and Wigton, L. B., “Numerical Evidence of Multiple Solutions for the Reynolds-Averaged Navier-Stokes Equations,” *AIAA Journal*, Vol. 52, No. 8, 2014, pp. 1686–1698. doi:10.2514/1.J052676.

[35] 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.

[36] Michal, T., and Krakos, J., “Anisotropic Mesh Adaptation Through Edge Primitive Operations,” AIAA Paper 2012–159, 2012.

[37] 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.

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

[39] 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.

[40] 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.

[41] Schmitt, V., and Charpin, F., “Pressure Distributions on the ONEAR-M6-Wing at Transonic Mach Numbers,” *Experimental Data Base for Computer Program Assessment: Report of the Fluid Dynamics Panel Working Group 04*, AR-138, NATO Research and Technology Organisation AGARD, 1979, pp. B1:1–B1:44.

[42] Mayeur, J., Dumont, A., Destarac, D., and Gleize, V., “Reynolds-Averaged Navier–Stokes Simulations on NACA0012 and ONERA-M6 Wing with the ONERA elsA Solver,” *AIAA Journal*, Vol. 54, No. 9, 2016, pp. 2671–2687. doi:10.2514/1.J054512.

[43] Mayeur, J., Dumont, A., Destarac, D., and Gleize, V., “RANS Simulations on TMR 3D Test Cases with the ONERA elsA Flow Solver,” AIAA Paper 2016–1357, 2016.

[44] Nishikawa, H., and Diskin, B., “Customized Grid Generation Codes for Benchmark Three-Dimensional Flows,” AIAA Paper 2018–1101, 2018.