https://ntrs.nasa.gov/search.jsp?R=20160006030 2019-08-31T03:00:53+00:00Z

Comparing Anisotropic Output-Based Grid Adaptation Methods by Decomposition

Michael A. Park*
NASA Langley Research Center, Hampton, VA 23681, USA

Adrien Loseille†
INRIA Paris-Rocquencourt, 78153, Le Chesnay, France

Joshua A. Krakos‡ and Todd Michal§
The Boeing Company, St. Louis, MO, USA

Anisotropic grid adaptation is examined by decomposing the steps of flow solution, adjoint solution, error estimation, metric construction, and simplex grid adaptation. Multiple implementations of each of these steps are evaluated by comparison to each other and expected analytic results when available. For example, grids are adapted to analytic metric fields and grid measures are computed to illustrate the properties of multiple independent implementations of grid adaptation mechanics. Different implementations of each step in the adaptation process can be evaluated in a system where the other components of the adaptive cycle are fixed. Detailed examination of these properties allows comparison of different methods to identify the current state of the art and where further development should be targeted.

I. Introduction

Output-based grid adaptation¶ methods have been developed to control the discretization error of numerical simulation. Fidkowski and Darmofal[1] provide a review of existing methods that encompass many different discretization schemes. Surprisingly, there are few examples where these methods have been directly compared to each other on the same geometry. One example of this comparison is when anisotropic tetrahedral and Cartesian cut-cell methods were applied to predict near field sonic boom.[2]

Output-based grid adaptation is an iterative process that involves multiple steps, see Fig. 1. When the grid adaptation process is separated into these constituent parts, different methods for error estimation, metric creation, and grid adaptation mechanics can be compared at a finer grain level than simply monitoring the convergence of the output with grid optimization. This work will focus on adaptation of simplicies (triangles and tetrahedra) to an anisotropic metric. The specific details of the adaptive process have been examined in a few situations (e.g., Hessian recovery,[3] the formulation of the error estimate.)[4]

Often, output-based approaches are examined by comparing an adapted discrete solution to an analytic solution when available. To address more complex problems without a closed-form solution, an adapted solution can be compared to a solution computed on a much finer uniformly refined grid. While these evaluation methods are good global indicators of the correctness and efficiency of the entire output-based process, they are unable to expose weaknesses or inefficiencies in the individual steps of the process.

This paper describes an approach targeted at improving the robustness and efficiency of the grid adaptation process by decomposing the process and comparing different implementations of each step (i.e., each

*Research Scientist, Computational AeroSciences Branch, AIAA Senior Member.
†Researcher, GAMMA3 Team, AIAA Member.
‡Engineer, AIAA Senior Member
§Technical Fellow, AIAA Senior Member
¶output-based, goal-based, and adjoint-based grid adaptation are common synonyms for each other.

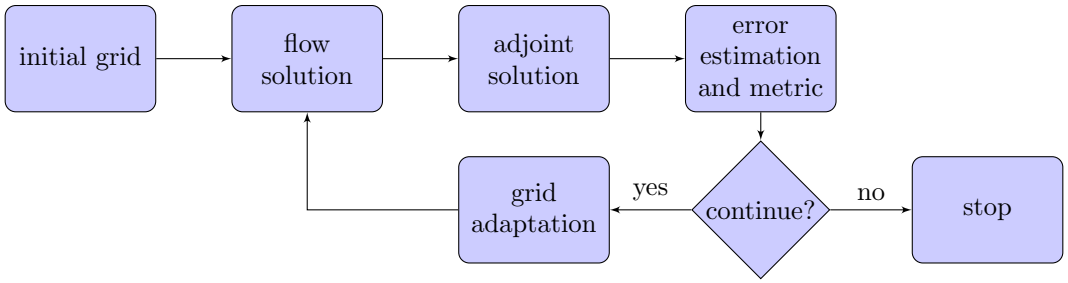


Figure 1. Output-based grid adaptation process.

box in Fig. 1) to understand the implications of implementation choices. Once the process is decomposed, each step can be verified by comparison to alternative implementations or an analytic result when available. Descriptions of each implementation and the results of this comparison are documented to ensure correct implementation and set the stage for further method development. This verification by comparison approach is also employed by the Turbulence Modeling Resource Website.[5] "What makes the current website unique is that it focuses on providing ready access to equations, grids, and flow solution details from previously verified codes as an aid to users who wish to verify their own implementations of models on relatively simple cases."[6] The goal of this work is to define a framework for rigorous examination of the entire output-based adaptation process and each of its constitutive parts that can guide the implementation and further development of solution adaptive methods.

II. Definitions

A metric tensor $\mathcal{M}$ is commonly employed to define the desired multidimensional grid resolution. This symmetric positive definite matrix can be diagonalized:

$$\mathcal{M} = X\Lambda X^T, \tag{1}$$

where the eigenvectors X define an orthonormal basis with length specifications h_i in this basis,

$$\Lambda = \begin{bmatrix} h_1^{-2} & 0 & 0 \\ 0 & h_2^{-2} & 0 \\ 0 & 0 & h_3^{-2} \end{bmatrix}. \tag{2}$$

The metric can be interpreted as an ellipsoid with axes of lengths h_i in directions of X_i.[7]

In a continuous setting, $\mathcal{M}$ is defined throughout the domain Ω and the edge length $\bar{\ell}$ of an edge parameterized between nodes A and B is given as,

$$\bar{\ell}_{\mathcal{M}}(A,B) = \int_0^1 \sqrt{AB^T\mathcal{M}((1-t)A+tB)AB}\,\mathrm{d}t, \tag{3}$$

where the parameter t varies linearly between $t=0$ at A and $t=1$ at B. In this study, $\mathcal{M}$ is sampled at each node and interpolated to evaluate edge lengths (and shape measures used internally by the grid adaptation mechanics). The length of an edge in the metric ℓ is either computed with quadrature or geometric interpolation as given by Alauzet.[8] With a metric $\mathcal{M}$ defined at each node in the mesh, the length of an edge defined between nodes A and B is

$$\ell_{\mathcal{M}}(A,B) \equiv \frac{r-1}{r\ln r}\sqrt{AB^T\mathcal{M}(A)AB}, \tag{4}$$

where

$$r = \frac{\sqrt{AB^T\mathcal{M}(A)AB}}{\sqrt{AB^T\mathcal{M}(B)AB}}. \tag{5}$$

As $r \to 1$, the $\mathcal{M}(A)$ and $\mathcal{M}(B)$ become equivalent. For $|r-1| < 10^{-12}$,

$$\ell_{\mathcal{M}}(A,B) = \frac{\sqrt{AB^T\mathcal{M}(A)AB} + \sqrt{AB^T\mathcal{M}(B)AB}}{2}, \tag{6}$$

to prevent the $r \ln r$ term from producing a divide-by-zero in finite precision arithmetic. The goal of adaptation is to produce a grid with all edges of unit length $\ell = 1$.

The statistical moments are computed in log space by

$$\hat{\mu} = \frac{1}{E} \sum_{i=1}^{E} \log_2(\ell_i), \tag{7}$$

where ℓ_i is the length of edge i in $\mathcal{M}$. There are E edges in the grid. The mean edge length reported in the table of statistics is $2^{\hat{\mu}}$. The higher order moments are given by

$$\hat{\mu}_j = \frac{1}{E} \sum_{i=1}^{E} \left| \log_2(\ell_i) - \hat{\mu} \right|^j . \tag{8}$$

A series of adapted anisotropic grids do not have the a uniform scaling of element sizes, which are present in uniformly refined grids. For adapted grids, the complexity C can be used as a continuous counterpart of the classical parameter h from uniformly refined grid studies.[9,10] In the continuous setting, the complexity of a grid is defined as

$$\bar{C} = \int_{\Omega} \sqrt{\det(\mathcal{M})} \, \mathrm{d}\Omega, \tag{9}$$

for the domain Ω. In the discrete setting, $\mathcal{M}$ is sampled at each node i and

$$C \equiv \sum_{i=1}^{N} \sqrt{\det(\mathcal{M}_i)} V_i, \tag{10}$$

where V_i is the volume of the Voronoi dual surrounding each node. The complexity has a linear dependency with respect to the number of nodes and elements. It is used in error estimates to specify the desired accuracy via the size of the adapted grid.

III. Output-Based Grid Adaption

The output-based grid adaption cycle described in Fig. 1 is composed of various steps detailed in this section. Given an initial grid, the process starts by computing a flow solution.

III.A. Flow and Adjoint Solution

FUN3D[11,12] is used to compute the flow solution. An exact discrete adjoint[13] is computed with a solution scheme described by Nielsen et al.[14] that is wrapped with the Generalized Conjugate Residual (CGR)[15] iterative scheme to accelerate convergence. Some of the viscous cases required the Hierarchical Adaptive Nonlinear Iterative Method (HANIM) to reach a converged steady-state residual on the high aspect ratio adapted grids. (HANIM was formerly referred to as HANIS by Pandya et al.[16])

III.B. Error Estimation and Metric Construction

Two error estimation, localization, and metric construction methods are compared in this study. These include the anisotropic optimal goal-based (Opt-Goal) metric of Loseille, Dervieux, and Alauzet[17] and the output-based anisotropic metric of Venditti.[18] Both of these methods require Hessian recovery[19] from the discrete solution.

Hessian recovery is performed with the "double L^2-projection" method of Alauzet and Loseille.[3] This method computes the gradient in each simplex based on the solution at the nodes. Then this reconstructed gradient is volume weighted to form an average reconstructed gradient from surrounding simplicies to each node. A second pass of this gradient reconstruction operator is used to recover second and mixed partial derivatives. Off-diagonal Hessian terms are averaged to yield a symmetric Hessian.

Vallet et al.[20] provides analysis that shows that the reconstructed gradient is less accurate on the boundaries than the interior. To prevent the inaccurate reconstructed boundary Hessian from negatively impacting

the adaptive process, the boundary values of the reconstructed Hessian are discarded and the boundary Hessian is averaged from interior nodes connected to the boundary nodes by edges. This averaging technique dramatically improved results and the convergence of the adaptive process.

Details of the 3D implementation of the Venditti[18] metric are provided by Park.[21–23] Anisotropy of the metric is set by the Mach number Hessian, which is scaled so that its largest eigenvalue matches a spacing derived from the current spacing and an adjoint error estimate. This current spacing is estimated by computing the metric of each simplex[24] and averaging the implied metric to the nodes in log-Euclidean space. At the nodes, the largest eigenvalue is computed to estimate the local spacing of the grid. This process can produce noise in the current spacing estimate and is negatively impacted by sliver cells. So, two passes of isotropic gradation control are used to limit the local spacing to not less than 3/2 times its neighbor's spacing. This prevents an isolated sliver element from indicating an inappropriately small local spacing.

The Opt-Goal metric has the benefit of no direct dependency on the current grid, making it less sensitive to the size and shapes of the current grid elements. This metric depends only on the Hessian of the Euler fluxes weighted by the gradient of the adjoint state,

$$\mathcal{M}_{\text{Opt-Goal}} \;=\; \left(\det\left|H_{\text{goal}}\right|\right)^{-\frac{1}{2p+d}} \left|H_{\text{goal}}\right|, \tag{11}$$

with the norm order of $p = 1$, the dimensionality d, and the goal based Hessian,

$$H_{\text{goal}} = \sum_{j=1}^{5} \left([\Delta x]_j(\mathbf{x}) + [\Delta y]_j(\mathbf{x}) + [\Delta z]_j(\mathbf{x}) \right). \tag{12}$$

Where

$$[\Delta x]_j = \left|\frac{\partial W_j^*}{\partial x}\right| \cdot \left|H(\mathcal{F}_1(W_j))\right|, \quad [\Delta y]_j = \left|\frac{\partial W_j^*}{\partial y}\right| \cdot \left|H(\mathcal{F}_2(W_j))\right|, \quad [\Delta z]_j = \left|\frac{\partial W_j^*}{\partial z}\right| \cdot \left|H(\mathcal{F}_3(W_j))\right|, \tag{13}$$

with W_j^* denoting the j^{th} component of the adjoint vector W^* and $H(\mathcal{F}_i(W_j))$ the Hessian of the j^{th} component of the vector $\mathcal{F}_i(W)$.

The aspect ratio AR of the metric can be optionally controlled by replacing h_2 with $\hat{h}_2$ and h_3 with $\hat{h}_3$ in Eq. (2):

$$\hat{h}_2 = \max(h_2, AR h_1) \tag{14}$$

$$\hat{h}_3 = \max(h_3, AR h_1). \tag{15}$$

Gradation control is implemented in the form of Eq. (9) from Alauzet[8] and applied to both solution adaptive metrics. A gradation[8] of $\beta = 1.5$ is used for all cases. The metric is rescaled to maintain the desired complexity after each of ten passes of gradation control. This is important for constructing a metric that is realizable by the grid adaption mechanics.

III.C. Grid Adaptation Mechanics

The mesh mechanics from four grid adaption tools are considered: `BAMG`[25] (Bidimensional Anisotropic Mesh Generator), `refine`,[26,27] `Feflo.a`,[28] and `EPIC`[29] (Edge Primitive Insertion and Collapse). Each of these codes attempts to produce a simplex (triangles in 2D and tetrahedra in 3D) mesh from a supplied anisotropic metric field. The focus of this study is to compare grid mechanic implementations at the elemental operator level. Variations in implementation near curved boundaries is avoided by only examining geometry models with linear surfaces. The impact of curved boundaries will be investigated in a future study. Results for 2D and 3D test cases with prescribed and computed sizing metrics are presented.

The `BAMG`[25] (Bidimensional Anisotropic Mesh Generator) is used for 2D examples. The original version of `refine`, `refine/one`,[26] has been modified to target edges with a length in metric space outside of [0.3,1.8]. The original intent[26] of the tool was to always produce edges that are equal or smaller than the metric specification. This study is focused on metric conformity or producing a unit mesh in the metric space, which requires a distribution of edge lengths centered about unity. The metric supplied to the tool is used as provided with no modifications (e.g., smoothing, limiting). A second version, `refine/two`,[27] is also evaluated, which only utilizes split and collapse operators.[29] The 2D algorithm has been extended to include the vertex smoothing of Alauzet.[30] The metric is used by `refine/two` as provided with no modifications.

`Feflo.a` belongs to the class of metric-based local remeshing software. It uses the classical operators of insertion, collapse, swap, and smoothing described by Frey and George.[28] Surface and volume meshes are adapted in a couple of ways to ensure a high-level of robustness. It is a based on a unique cavity-operator, where each of the standard operators (e.g., insertion, collapse, swap, smoothing) are recast within an insertion procedure. The operation that results from this insertion depends only on the choice of the cavity. Cavity optimizations (e.g., reduction, enlargement) are performed to improve the quality of mesh after the insertion. `Feflo.a` handles 2D, surface, and 3D meshes.

Grids from the `EPIC` adaptation code are presented for the 3D examples. The grid mechanics in `EPIC` have been extended to include element reconnection (swapping) and node smoothing operators. Element swapping is performed using the generalized swapping technique of Alauzet.[30] Node movement is performed using a local optimization technique to maximize a metric-based cell quality objective function. Edge lengths in `EPIC` are computed with numerical integration of Eq. (3) instead of assuming a linear variation of the metric $\mathcal{M}$ in log-Euclidean space. `EPIC` grids using three sets of mesh operators are presented: only insertion and collapse (`EPIC-IC`); insertion, collapse, and swap (`EPIC-ICS`); insertion, collapse, swap, and node movement (`EPIC-ICSM`).

IV. Prescribed Metric Fields

To examine the grid adaption step of Fig. 1 in isolation, 2D and 3D versions of an analytic metric field are used to evaluate the grid adaptation mechanics on a unit Cartesian domain in the following subsections.

IV.A. Analytic 2D Metric Field

The `BAMG`, `refine/two`, and `Feflo.a` grid adaptation processes are applied to an analytically defined (on the domain $x = [0, 1]$ and $z = [0, 1]$) 2D metric field given by

$$\mathcal{M} = \begin{pmatrix} h_1^{-2} & 0 \\ 0 & h_2^{-2} \end{pmatrix} \text{ where } \begin{cases} h_1 = 0.1 \\ h_2 = 0.001 + 0.198\,|z - 0.5| \end{cases} . \tag{16}$$

The metric is provided at the nodes of an initial coarse grid. The grid is adapted to the metric on this grid and then the analytic metric is evaluated on the new grid. This process, where the analytic metric is evaluated at the nodes and the grid is adapted, is repeated nine times. By this point, the grid adaptation cycle has reached steady state. The final grids are shown in Fig. 2. Histograms of edge length ℓ in $\mathcal{M}$ evaluated on the final grids are provided for these methods in Fig. 3 and statistics on these final grid edge lengths are shown in Table 1. Distributions in Fig. 3 that appear more narrow and peaked produce lower values of the moments in Table 1.

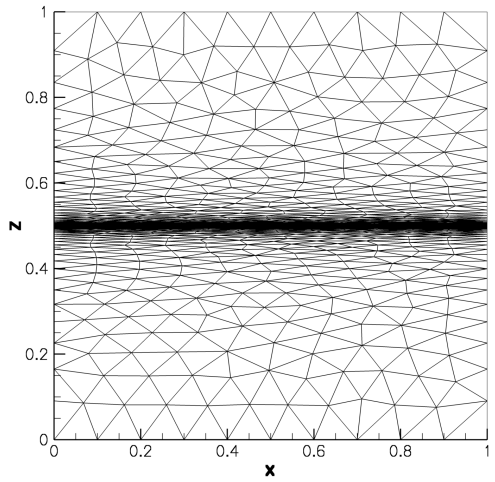
The minimum and maximum edge lengths are comparable for all methods, and bounded by the range of `refine/two` between 0.601 and 1.472. `BAMG` has a slightly smaller mesh with a vast majority of the edges near unit length in the metric. The methods produce grids with node counts of 0.597 to 0.723 of the complexity for this 2D metric. The theoretical minimum for the ratio of nodes to complexity is

$$\frac{1}{2}\frac{\pi}{\sqrt{12}} = 0.45345, \tag{17}$$

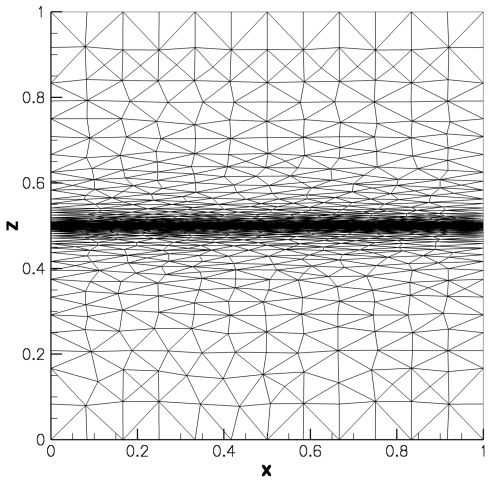
where 1/2 is the ratio of nodes to triangles in an infinite 2D grid with an average node degree of six and $\pi/\sqrt{12}$ is the density of triangles in an ideal Delaunay grid (also true for a constant metric field and ideal mapped Delaunay grid).[31] `BAMG` produced a grid with a node count that is closest to this theoretical limit. It also had the best statistics in terms of mean and higher order moments, which can be observed qualitatively in the Fig. 3 histograms.

Table 1. 2D linear metric properties.

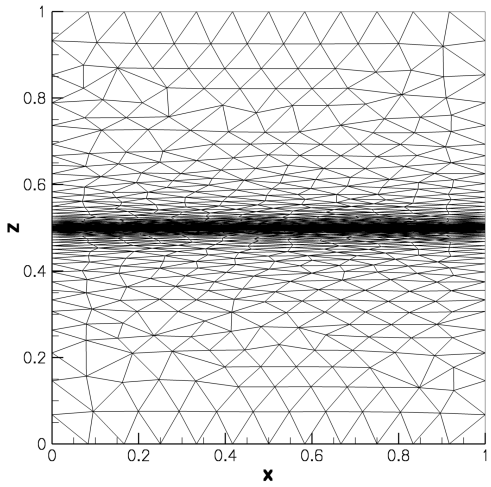
Mechanics	Nodes	Complexity	Ratio	Min Edge	Max Edge	Mean	Higher-order statistics		
	N	C	N/C	$\min(\ell)$	$\max(\ell)$	2^{μ}	μ_2	μ_3	μ_4
`BAMG`	830	1390.2	0.597	0.683	1.436	1.039	0.0277	-0.0007	0.0024
`refine/two`	1005	1389.9	0.723	0.601	1.472	0.950	0.0797	-0.0023	0.0149
`Feflo.a`	941	1389.1	0.713	0.617	1.413	0.948	0.0459	0.0026	0.0055



(a) `BAMG`.

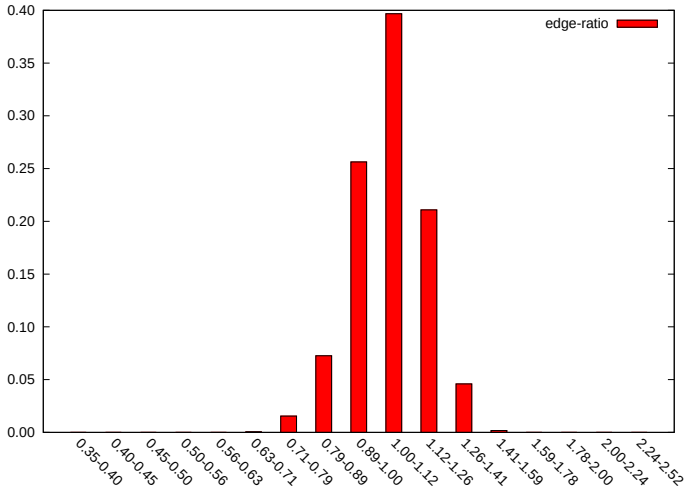


(b) `refine/two`.

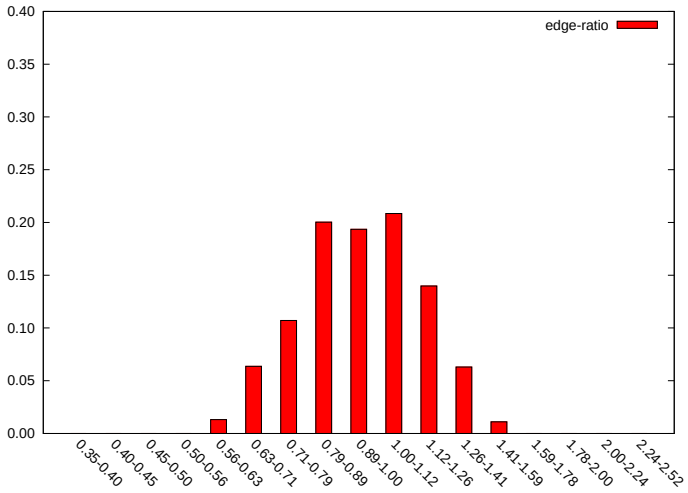


(c) `Feflo.a`.

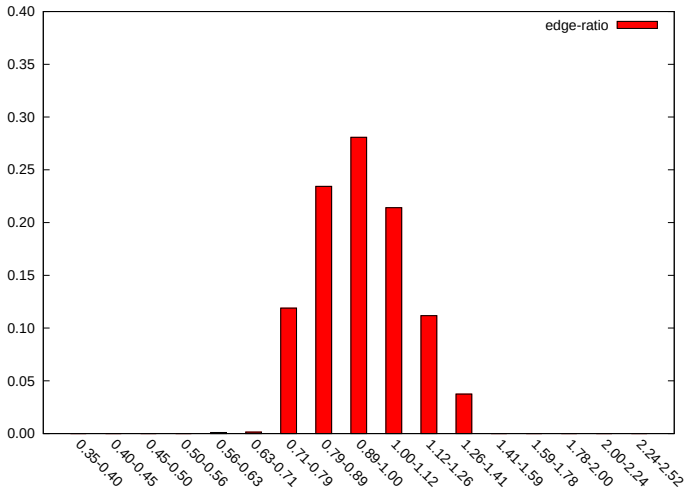
Figure 2. Final adapted grids for the 2D linear metric.



(a) `BAMG`.



(b) `refine/two`.



(c) `Feflo.a`.

Figure 3. Histogram of edge length in 2D linear metric.

IV.B. Analytic 3D Metric Field

The `Feflo.a`, `refine/one`, `refine/two`, and multiple `EPIC` grid adaptation tools are applied to a 3D analytically defined (on the domain $x = [0,1]$, $y = [0,1]$, and $z = [0,1]$) metric field given by

$$\mathcal{M} = \begin{pmatrix} h_x^{-2} & 0 & 0 \\ 0 & h_y^{-2} & 0 \\ 0 & 0 & h_z^{-2} \end{pmatrix} \text{ where } \begin{cases} h_x = 0.1 \\ h_y = 0.1 \\ h_z = 0.001 + 0.198\,|z - 0.5| \end{cases} . \tag{18}$$

As in the 2D case, the metric is provided at the nodes of an initial coarse grid. The grid is adapted to the metric on this grid and the process is repeated until the process converges. The final grids are shown in Fig. 4. Histograms of edge length ℓ in $\mathcal{M}$ are provided for the two methods in Fig. 5. Statistics on these final grids are shown in Table 2.

The ratio of the number of nodes to the complexity is near 2, which is expected by theory[9] and numerical experiments.[10] The `Feflo.a` grid is the smallest and has a large majority of near unity length edges in the metric space. Both versions of `refine` produce grids with more nodes than `Feflo.a` and a significant population of edges shorter than unity are shown in the histogram and minimum edge length in metric space. The ability of `Feflo.a` to prevent the creation of short edges or to remove existing short edges may be enhanced by the cavity operator.[32] The methods that include node movement (`Feflo.a`, `refine/one`, and `EPIC-ICSM`) tend to produce the histograms in Fig. 5 with a more peaked distribution. The methods without node movement (`refine/two`, `EPIC-IC` and `EPIC-ICS`) tend to produce a uniform distribution of edge lengths between their internal edge length limits.

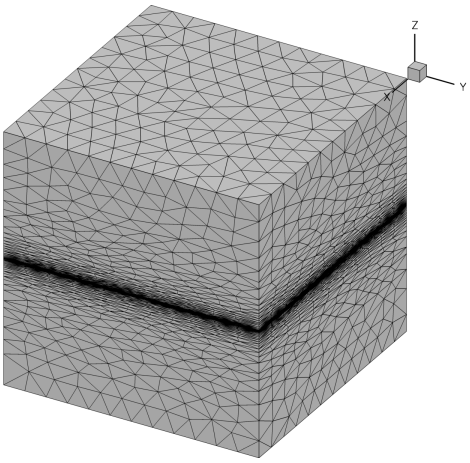
The addition of element swapping and node smoothing improves the statistics of the `EPIC-IC` algorithm. The long edges in the `EPIC` meshes can be attributed to the edge length computation method that `EPIC` uses compared to `Feflo.a` and `refine`. Statistics for the `EPIC` grid using the quadrature based edge lengths are shown in the last three lines in Table 2. When the edge lengths are evaluated using the quadrature based method consistent with the `EPIC` formulation, the maximum edge lengths become consistent with the other methods.

Statistics for the `EPIC` edge lengths reported in Table 2 are computed with two methods. The first set are computed with Eq. (4), which is consistent with `refine` and `Feflo.a`. The second set with the footnote are computed using numerical integration of Eq. (3), which is consistent with `EPIC`'s internal calculations. The numerical integration approach is more accurate where the edge and metric principle directions are not aligned. The maximum and minimum edge lengths produced by `EPIC` are longer than `refine`. This reflects a design choice in the `EPIC` implementation that accepts a slightly longer edge if splitting that edge creates a very short edge. The addition of element swapping and node smoothing improves the statistics relative to `EPIC-IC`.

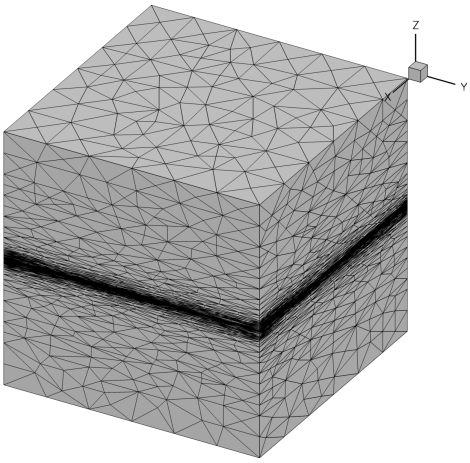
Table 2. 3D linear metric properties.

Mechanics	Nodes	Complexity	Ratio	Min Edge	Max Edge	Mean	Higher-order statistics		
	N	C	N/C	$\min(\ell)$	$\max(\ell)$	2^μ	μ_2	μ_3	μ_4
`Feflo.a`	8933	4668.6	1.91	0.449	1.804	1.011	0.0597	-0.0031	0.0086
`refine/one`	9981	4670.3	2.14	0.078	1.799	0.952	0.1932	-0.0506	0.1341
`refine/two`	12868	4673.8	2.75	0.065	1.500	0.977	0.1423	-0.0414	0.1217
`EPIC-IC`	9524	4671.4	2.04	0.149	3.483	1.024	0.1227	-0.0112	0.0435
`EPIC-ICS`	9612	4670.2	2.06	0.324	3.051	1.005	0.1120	-0.0054	0.0314
`EPIC-ICSM`	9018	4669.9	1.93	0.391	2.439	1.019	0.0902	-0.0094	0.0215
`EPIC-IC` [#]	9524	4671.4	2.04	0.149	2.201	1.022	0.1209	-0.0113	0.0414
`EPIC-ICS` [#]	9612	4670.2	2.06	0.324	2.137	1.004	0.1113	-0.0059	0.0306
`EPIC-ICSM` [#]	9018	4669.9	1.93	0.391	1.907	1.018	0.0900	-0.0095	0.0214

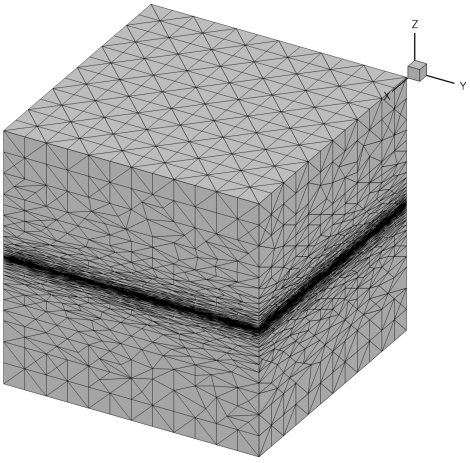
[#] These edge length statistics utilize numerical integration of Eq. (3), which is consistent with `EPIC` formulation. Other lines use Eq. (4).



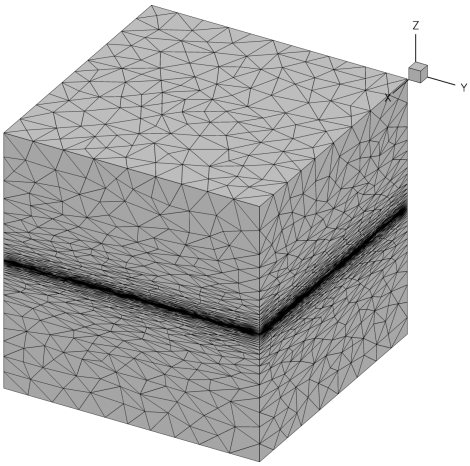
(a) `Feflo.a`.



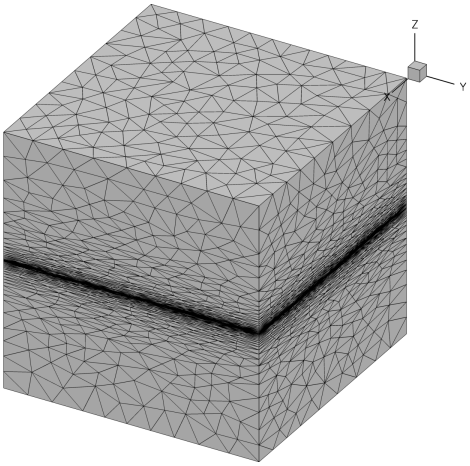
(b) `refine/one`.



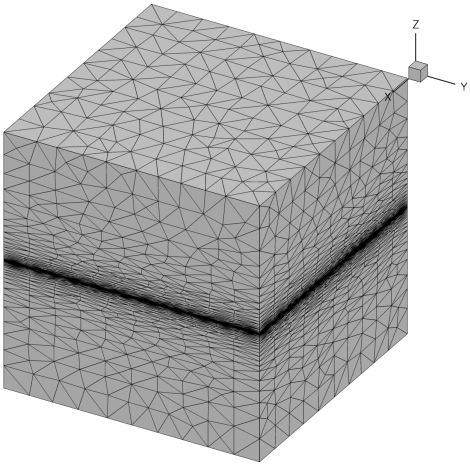
(c) `refine/two`.



(d) `EPIC-IC`.

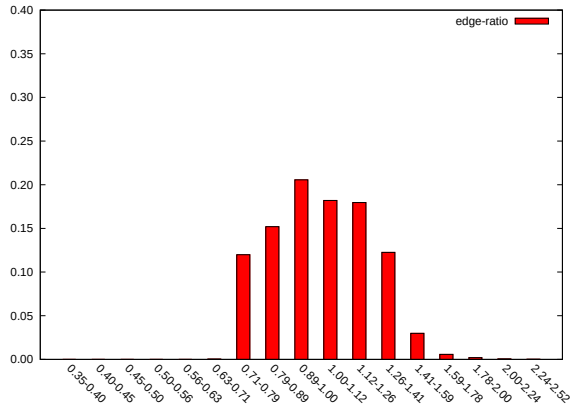


(e) `EPIC-ICS`.

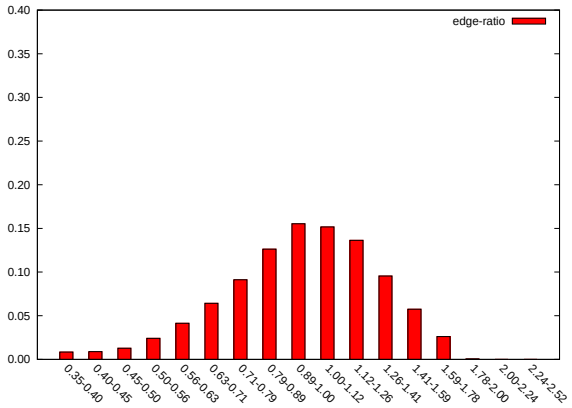


(f) `EPIC-ICSM`.

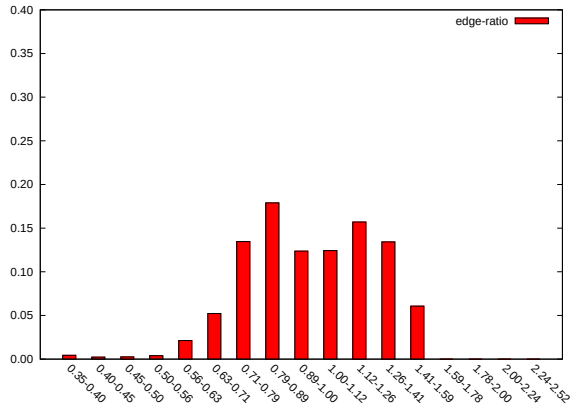
Figure 4. Final adapted grids for the 3D linear metric.



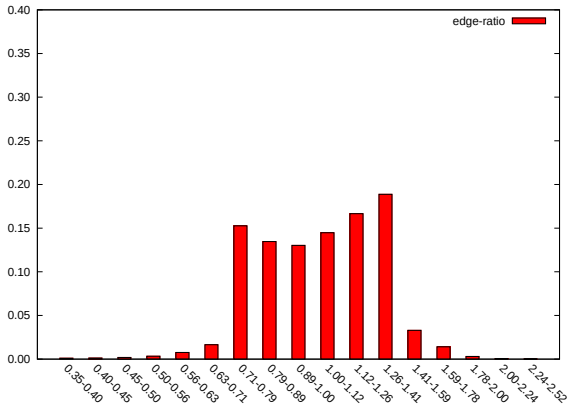
(a) `Feflo.a`.



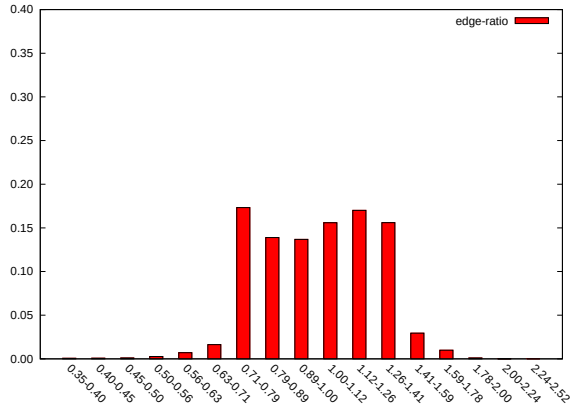
(b) `refine/one`.



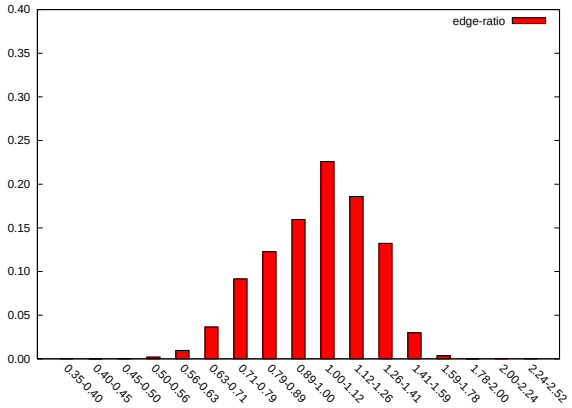
(c) `refine/two`.



(d) `EPIC-IC`.



(e) `EPIC-ICS`.



(f) `EPIC-ICSM`.

Figure 5. Histogram of edge length in 3D linear metric.

V. Output-Based Adaptation

Now that the adaptive mechanics has been evaluated with an analytic metric, the solution adaptive cases are examined. In this study, we consider only peace-wise linear geometries in 2D and 3D to avoid the surface approximation differences of these mesh generators.

V.A. Diamond Airfoil in 2D Supersonic Flow

A 2D diamond airfoil with a thickness of 0.07 is simulated in a supersonic inviscid flow with a Mach number of 2. The `BAMG` method is used to construct an initial coarse grid and a sequence of 20 grids with increasing complexity is computed with `BAMG` and `refine/two`. The complexity C_{2Diam} and maximum aspect ratio $\mathcal{AR}_{\text{2Diam}}$ are scheduled as

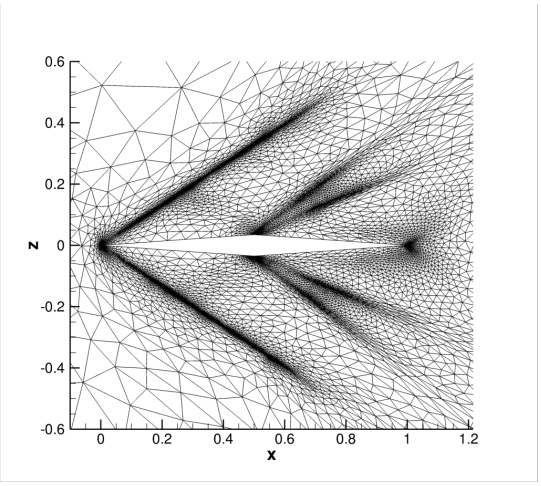
$$C_{\text{2Diam}} = \min(1000c, 10000) \tag{19}$$

and

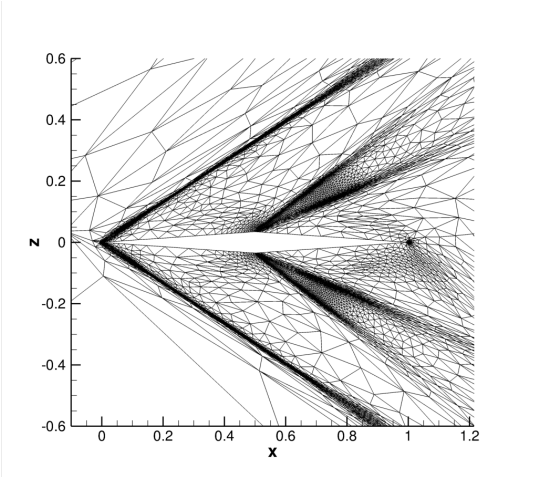
$$\mathcal{AR}_{\text{2Diam}} = 10, \tag{20}$$

where c is the adaptive cycle index

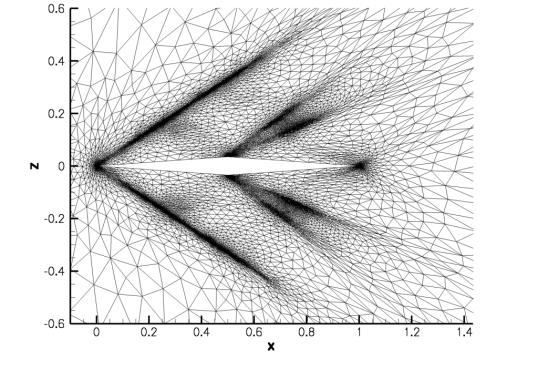
The final grids for the drag adaptation are shown in Fig. 6. The difference in drag of the final grid between the four methods is 0.00000174 (0.015 percent), see Table 3. The drag for this series of grids with increasing complexity are shown in Fig. 7. Drag computed on the INRIA Opt-Goal metric adapted grids approach the final drag more rapidly than drag computed on the Venditti metric adapted grids, and the `BAMG` method is able to produce grids that are closer to the requested complexity.



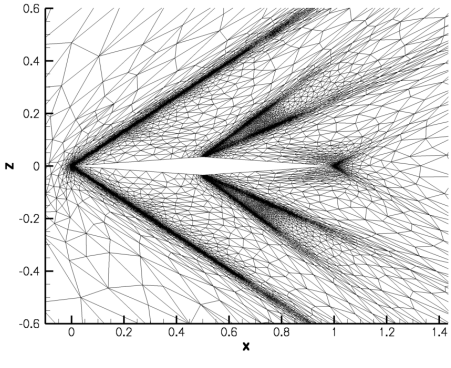
(a) `BAMG` with Venditti metric.



(b) `BAMG` with Opt-Goal metric.



(c) `refine/two` with Venditti metric.

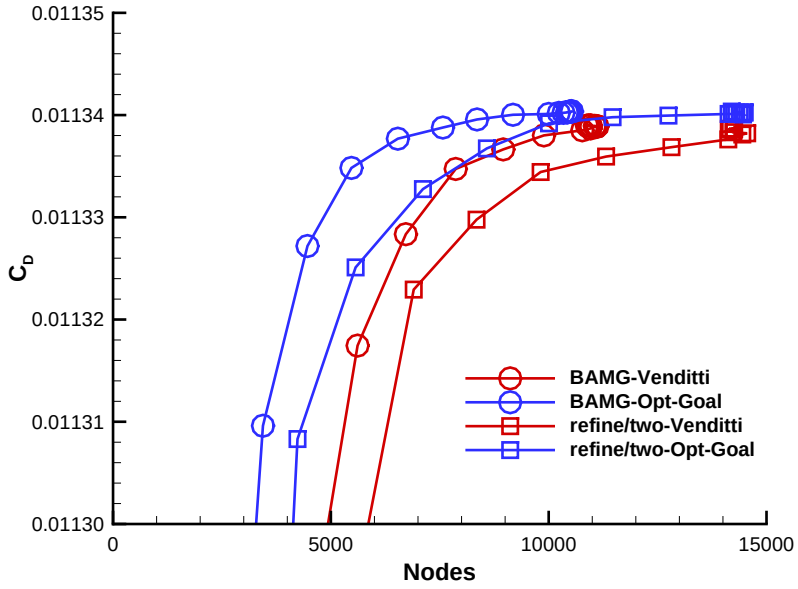


(d) `refine/two` with Opt-Goal metric.

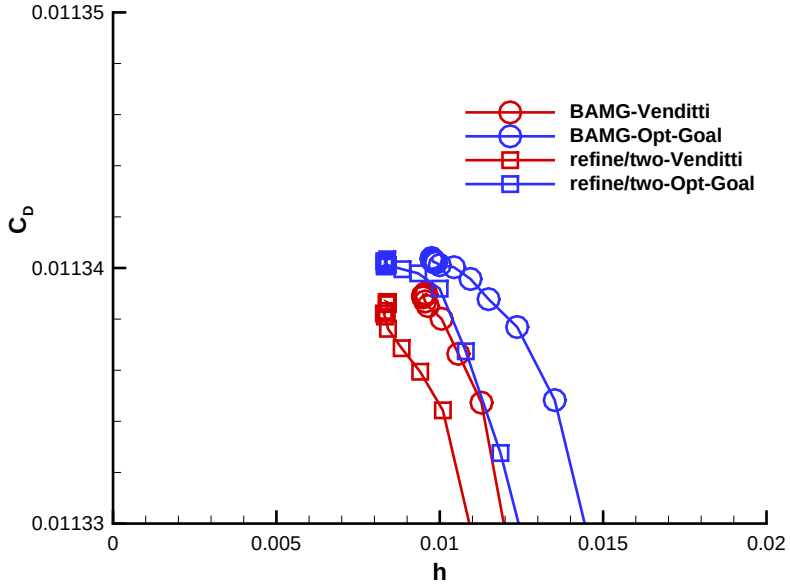
Figure 6. Final adapted grids for the 2D diamond airfoil.

Table 3. 2D diamond airfoil.

Mechanics	Error Estimation	Drag C_D	Nodes N	Complexity C	Ratio N/C
`BAMG`	Venditti	0.01133900	10964	10000	1.10
`BAMG`	Opt-Goal	0.01134039	10518	10000	1.05
`refine/two`	Venditti	0.01133865	14136	10000	1.41
`refine/two`	Opt-Goal	0.01134025	14499	10000	1.45



(a) Drag as function of degrees of freedom (N).



(b) Drag as function of $h = N^{-1/2}$.

Figure 7. Drag convergence for the 2D diamond airfoil.

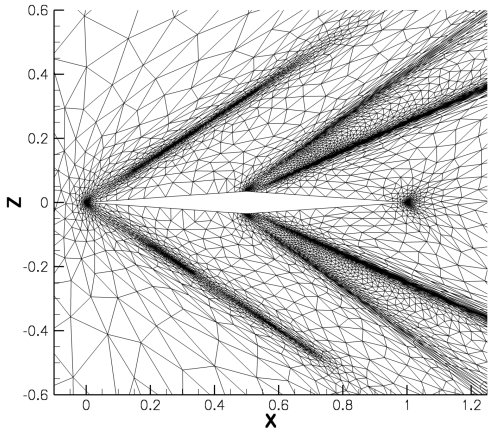
V.B. Diamond Airfoil Extruded to 3D in Supersonic Flow

A diamond airfoil with a span of one and a thickness of 0.07 is extruded to unit span in 3D supersonic inviscid flow with a Mach number of 2. The initial `BAMG` grid from the 2D case is extruded unit span to form two prisms from each triangle. This results in two symmetry planes and a center plane in the domain interior. This center plane enhances Hessian reconstruction accuracy, which degrades on boundaries. The extruded prisms are subdivided into the tetrahedra required by the simplex adaptation mechanics. The final grid sizes and drag values are shown in Table 4. The final symmetry plane grids for the drag adaptation are shown in Fig. 8. The drag for this series of grids with increasing complexity are shown in Fig. 9. The initial complexity is specified as 10,000 for the first ten adaptive cycles. This complexity is raised to 20,000, 40,000, and 80,000 after every 10 subsequent adaptive cycles. Aspect ratio is not limited. The Venditti metric and the Opt-Goal metric are used for these cases. `Feflo.a` and `refine/two` are run sequentially, but `refine/one` is run on 8 cores with a domain decomposition scheme[26] and `EPIC` is run with 127 cores.

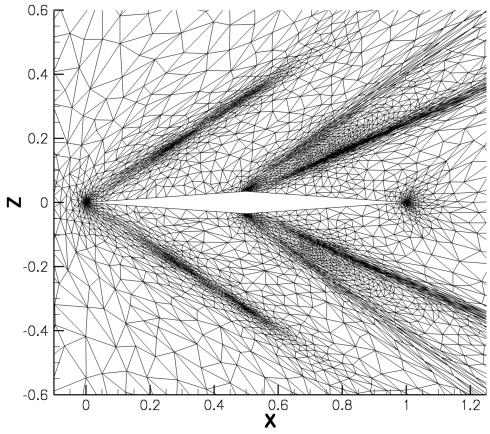
As seen in Fig. 9, `refine/one` converged more slowly than the other methods to a specified grid size and drag. The `Feflo.a` method produces smaller grids for a given complexity and has a drag values closer to the 2D results. The `EPIC-ICS` method produces results similar to `refine/two`. There is a larger variation in drag (0.044 percent) between the adapted 3D grids than the 2D adaptive grids (0.015 percent). Finer grids could be attempted to verify that these methods are converging to the same drag value, because convergence to a common drag value is not obvious at this error level. Figure 9(b) indicates that the drag is increasing more rapidly for the Venditti error estimate adapted grids. As in the 3D analytic metric case, the ratio of the number of nodes to the complexity is near 2, which is expected by theory[9] and numerical experiments.[10]

Table 4. 3D extruded diamond airfoil.

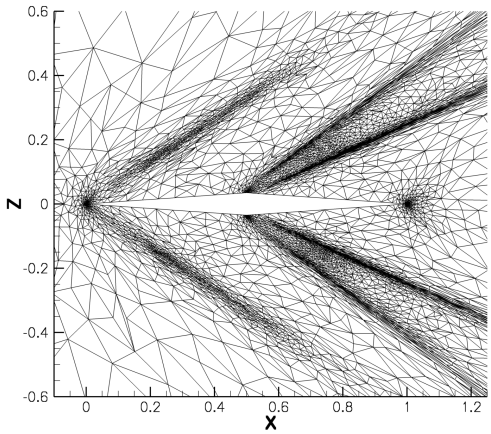
Mechanics	Error Estimation	Drag C_D	Nodes N	Complexity C	Ratio N/C
`Feflo.a`	Venditti	0.01133797	142026	80000	1.78
`refine/one`	Venditti	0.01133649	198679	80000	2.48
`refine/two`	Venditti	0.01133657	177023	80000	2.21
`EPIC-ICS`	Venditti	0.01133698	156656	80000	1.96
`EPIC-ICSM`	Venditti	0.01133738	154286	80000	1.93
`Feflo.a`	Opt-Goal	0.01134097	159740	80000	2.00



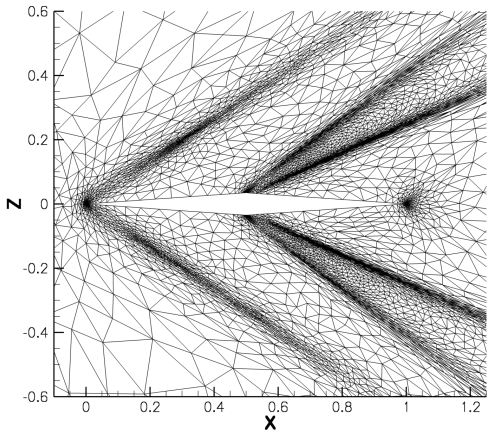
(a) `Feflo.a` with Venditti metric.



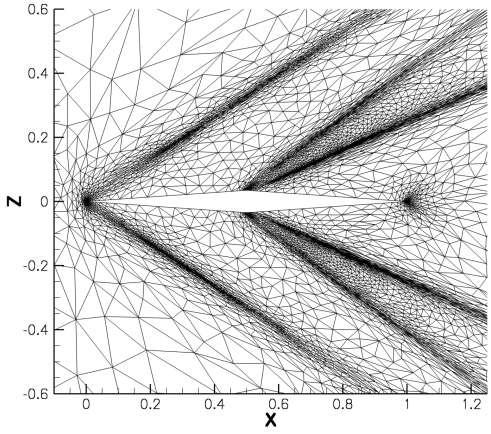
(b) `refine/one` with Venditti metric.



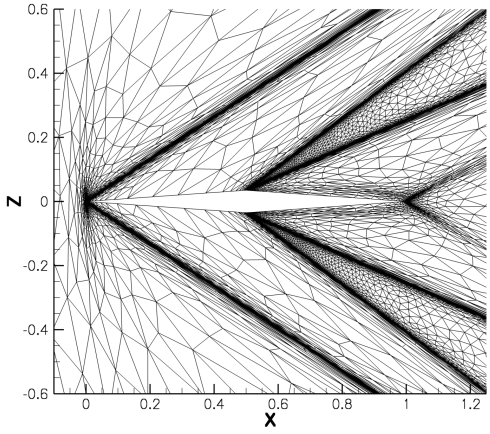
(c) `refine/two` with Venditti metric.



(d) `EPIC-ICS` with Venditti metric.

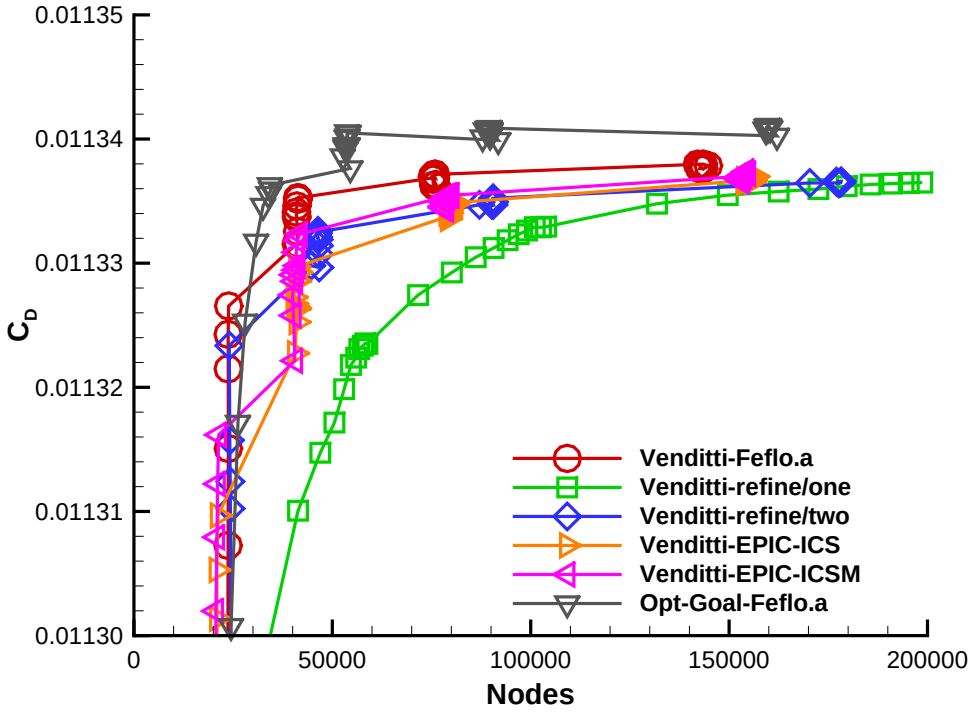


(e) `EPIC-ICSM` with Venditti metric.

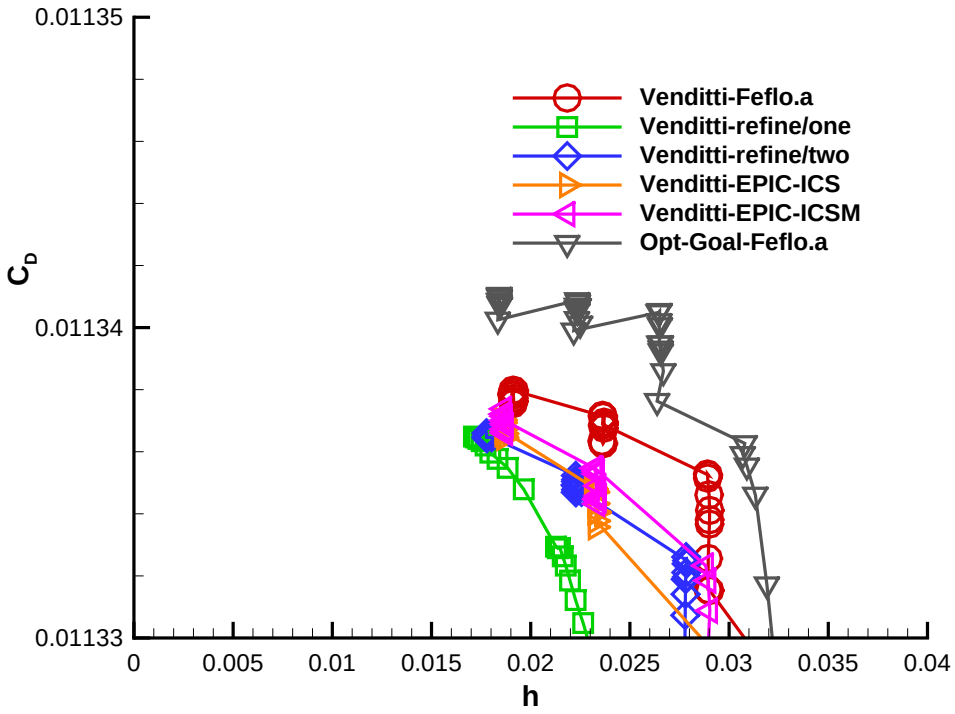


(f) `Feflo.a` with Opt-Goal metric.

Figure 8. Final adapted symmetry plane grids for the 3D extruded diamond airfoil.



(a) Drag as function of degrees of freedom (N).



(b) Drag as function of $h = N^{-1/3}$.

Figure 9. Drag convergence for the 3D extruded diamond airfoil.

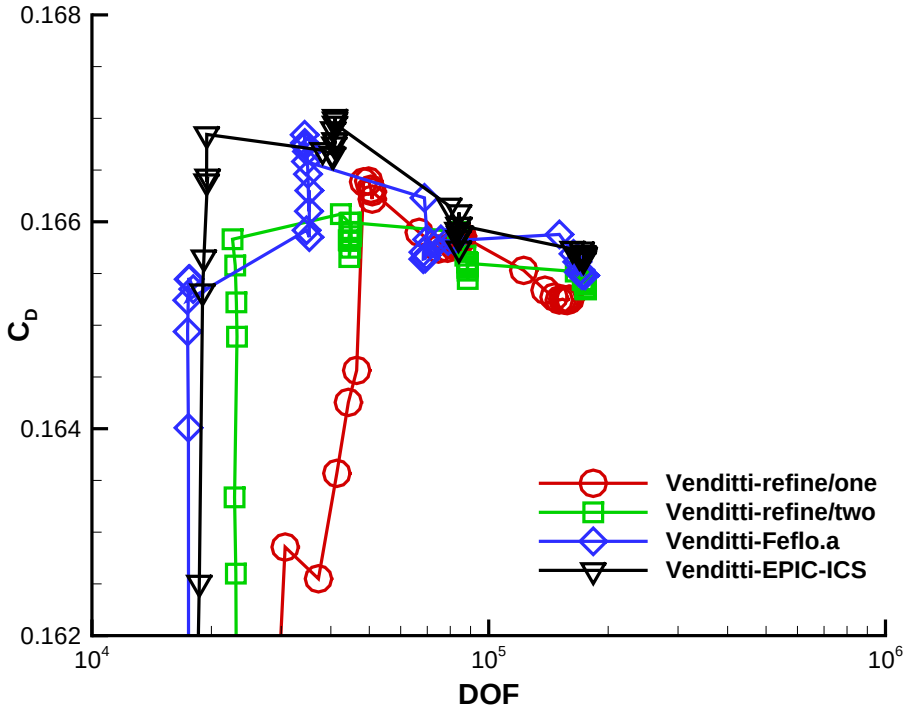
V.C. Laminar Delta Wing (3D)

The 3D Laminar Delta Wing case is described by Wang et al.[33] as part of its inclusion in all three of the International Workshops on High-Order CFD Methods to date. Adaptive results where originally presented by Leicht and Hartmann.[34] Drag and drag error are shown in Fig. 10. The reference drag used to compute drag error, $C_D^{\text{reference}} = 0.1656025$, is the drag computed on a twice embedded final `Feflo.a` grid with 10,853,343 nodes. The spacing h in the drag error plot is estimated using the suggestion of workshop problem description, $h = N^{-1/3}$.

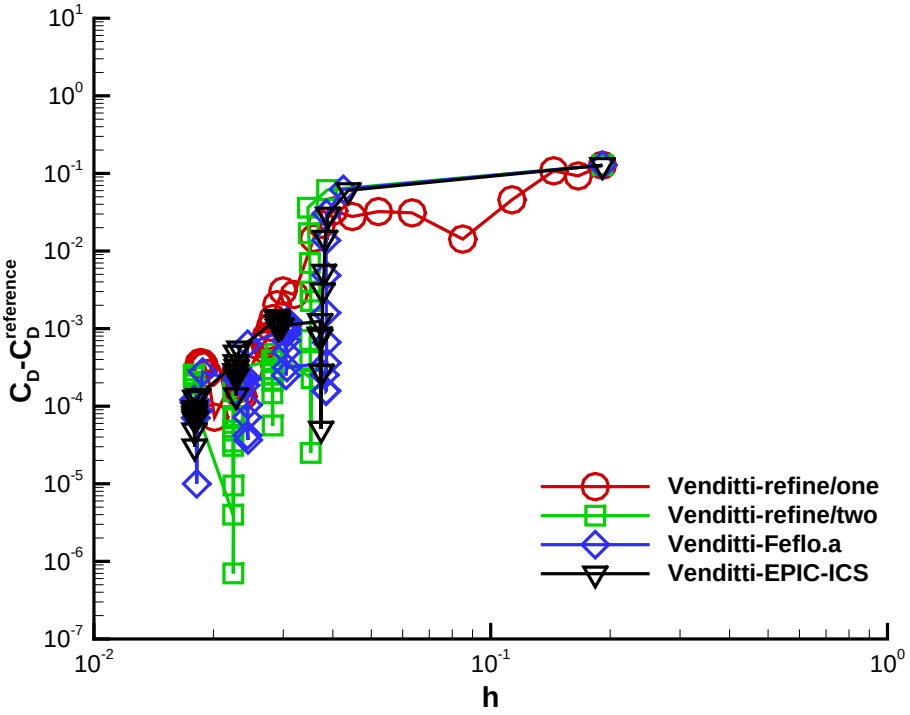
The same complexity schedule of the 3D extruded diamond airfoil is used in this example, which is 10,000, 20,000, 40,000, and 80,000 for ten cycles each. The HANIM solution scheme is required to reach a converted steady-state residual on the two finest complexities. `Feflo.a` and `refine/two` are run sequentially, but `refine/one` is run on 8 cores with a domain decomposition scheme[26] and `EPIC-ICS` is run with 191 cores. Drag (Fig. 10) converges less rapidly in this case than the 3D Extruded Diamond Airfoil, which may be caused by the strong nonlinearities of this problem. The drag error is lower for a given number of degrees of freedom (DOF) than the errors of the higher-order methods summarized by Wang et al.[33] One possible explanation is the nominally second-order FUN3D method on adaptive grids may be resolving singularities that are not resolved by the workshop provided grids.

Views of the final surface grid are provided for `refine/one` (Fig. 11), `refine/two` (Fig. 12), `Feflo.a` (Fig. 13), and `EPIC-ICS` (Fig. 14). These grids look very similar and clearly resolve the footprint of the leading-edge vortex on the upper surface of the delta wing. There is some slight clumping of the grid in the vicinity of the wing and symmetry plane intersection, which is suspected to be caused by noise from Hessian reconstruction.

Overall, the final anisotropic adapted surface grids have a very similar appearance for all four methods. The drag error for this case is converging and is lower (for a given number of degrees of freedom) than the uniform grids of the High-Order CFD workshop.

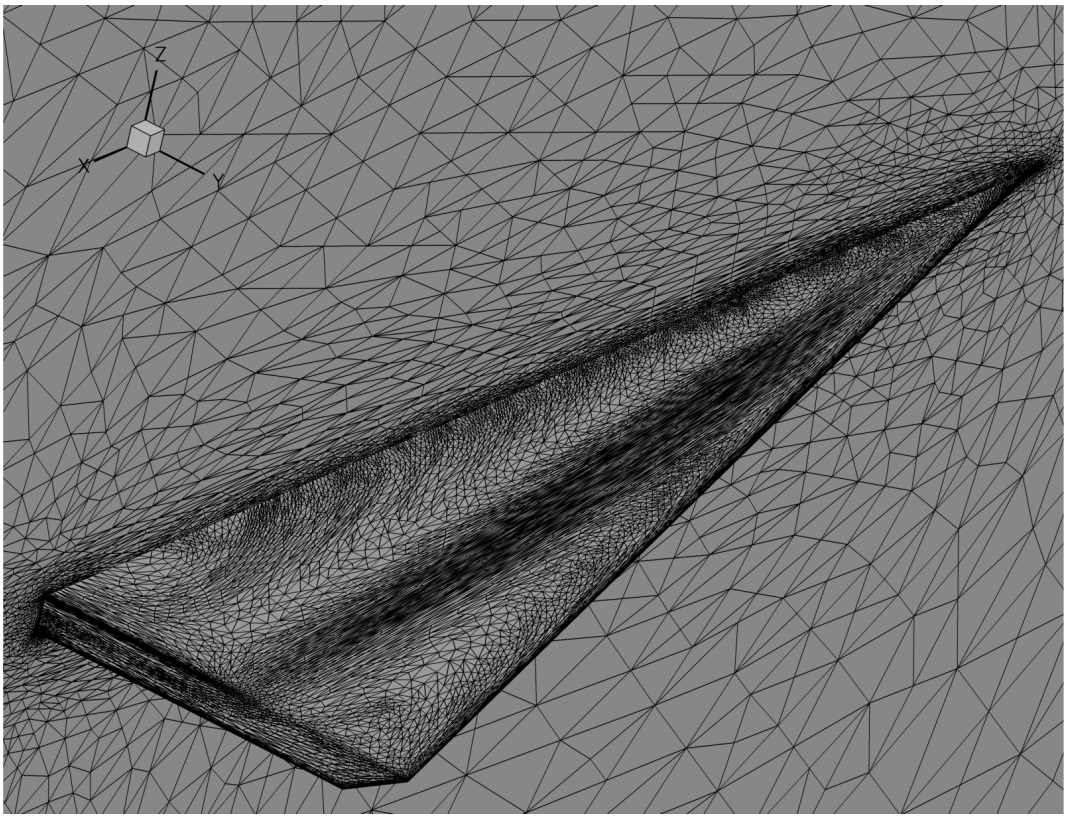


(a) Drag as function of degrees of freedom.

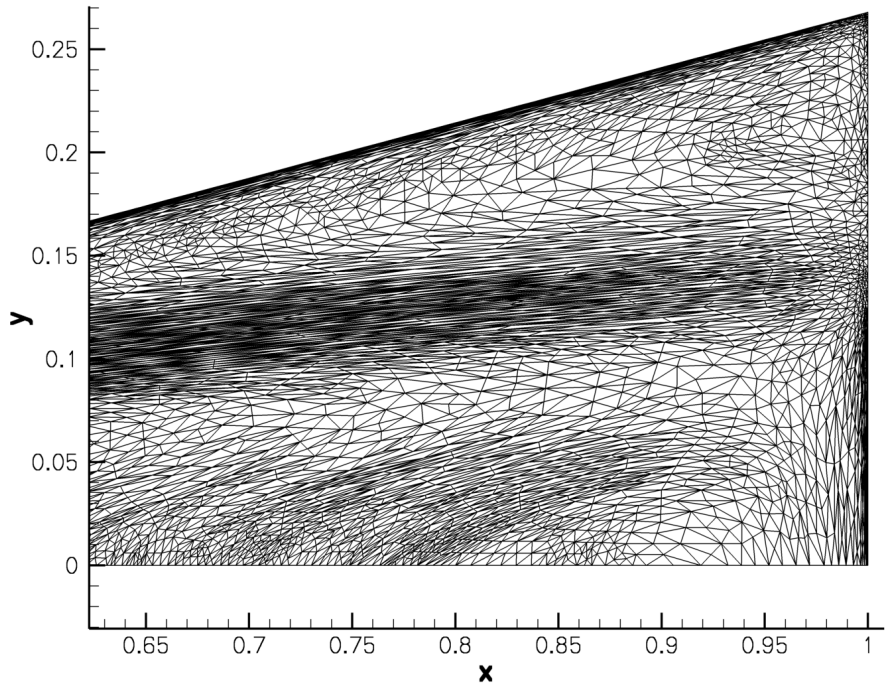


(b) Drag as function of h.

Figure 10. Drag and drag error convergence for the Laminar Delta Wing.

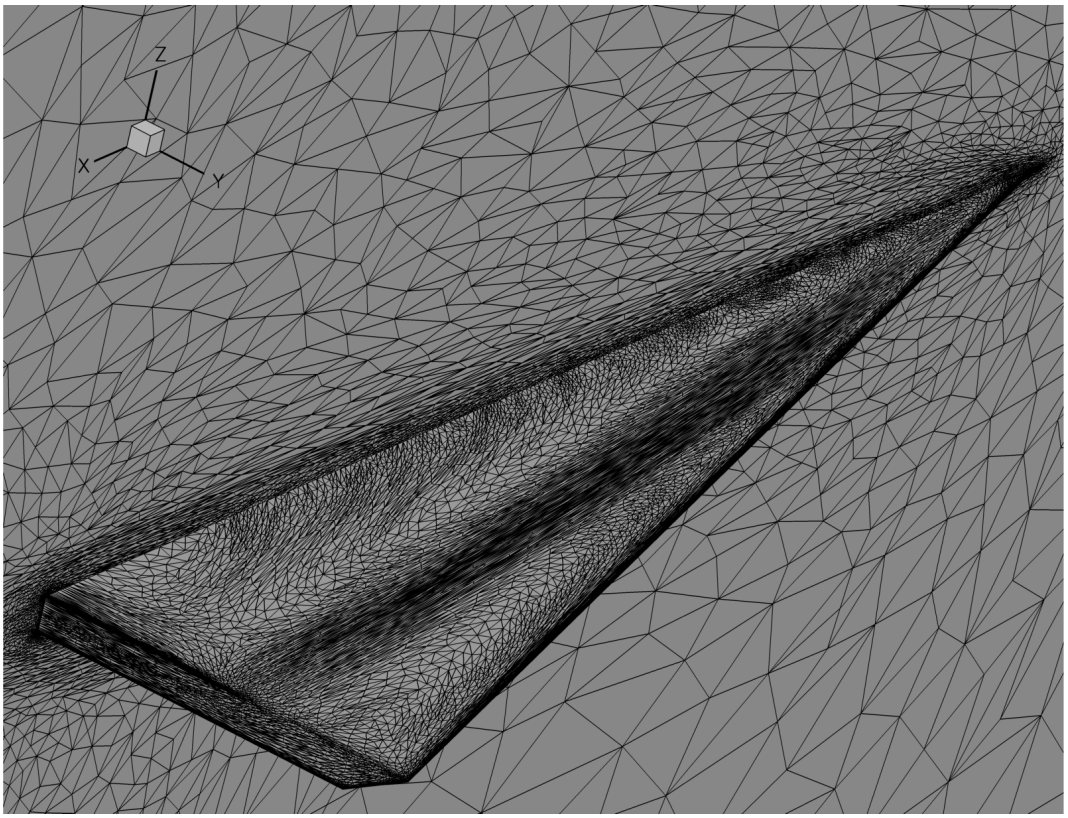


(a) Upper wing surface intersection with symmetry plane.

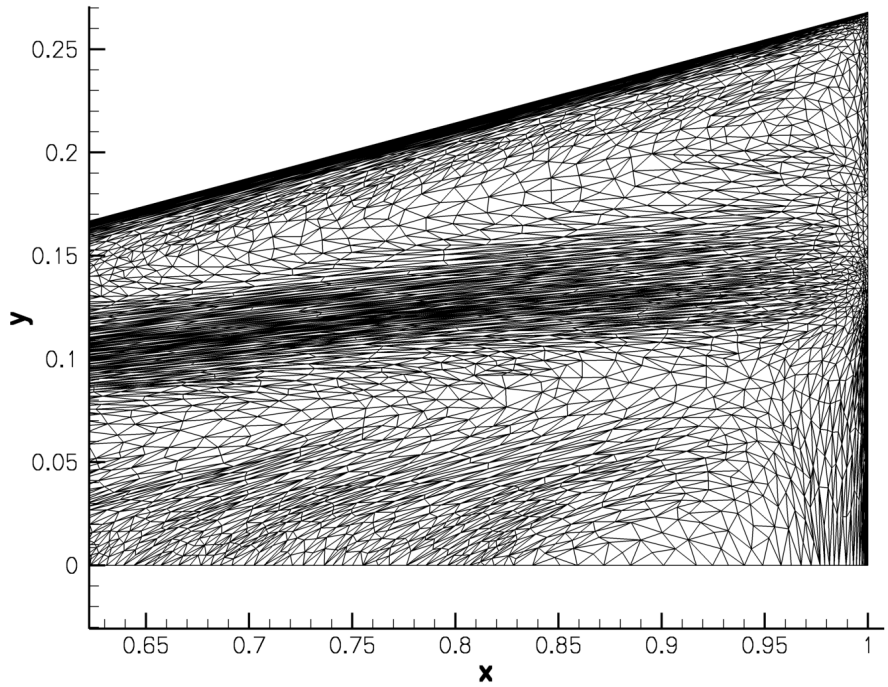


(b) Upper wing surface.

Figure 11. Final `refine/one` adapted grids for the Laminar Delta Wing.

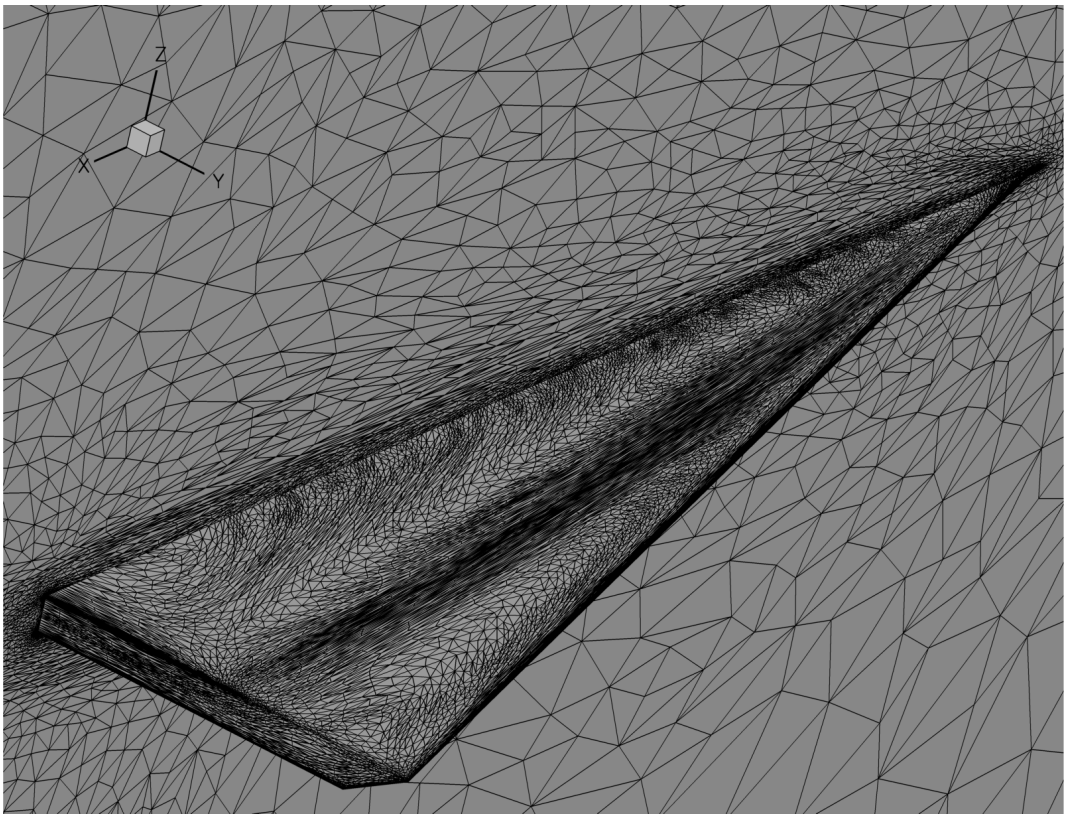


(a) Upper wing surface intersection with symmetry plane.

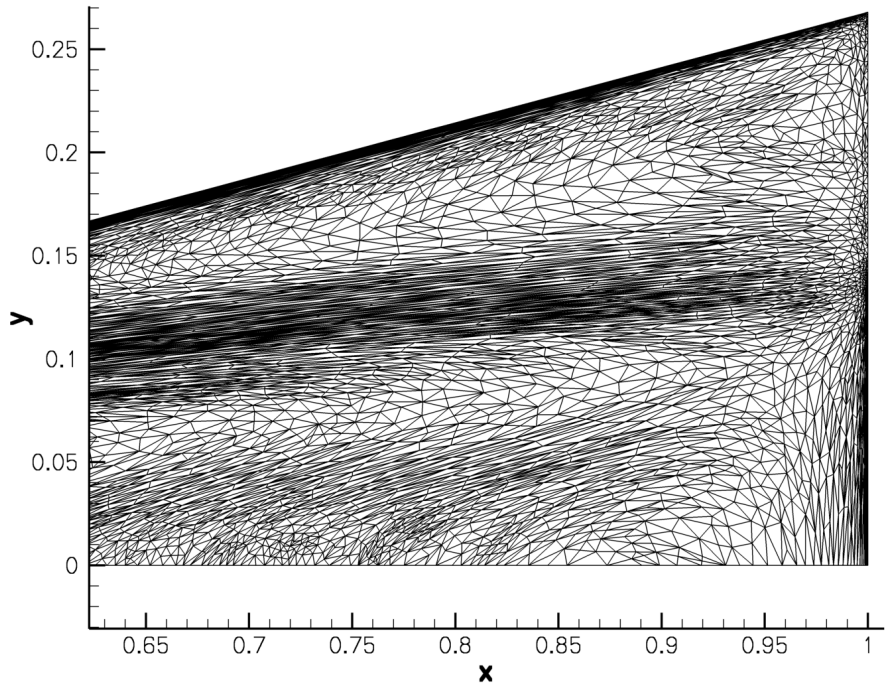


(b) Upper wing surface.

Figure 12. Final `refine/two` adapted grids for the Laminar Delta Wing.

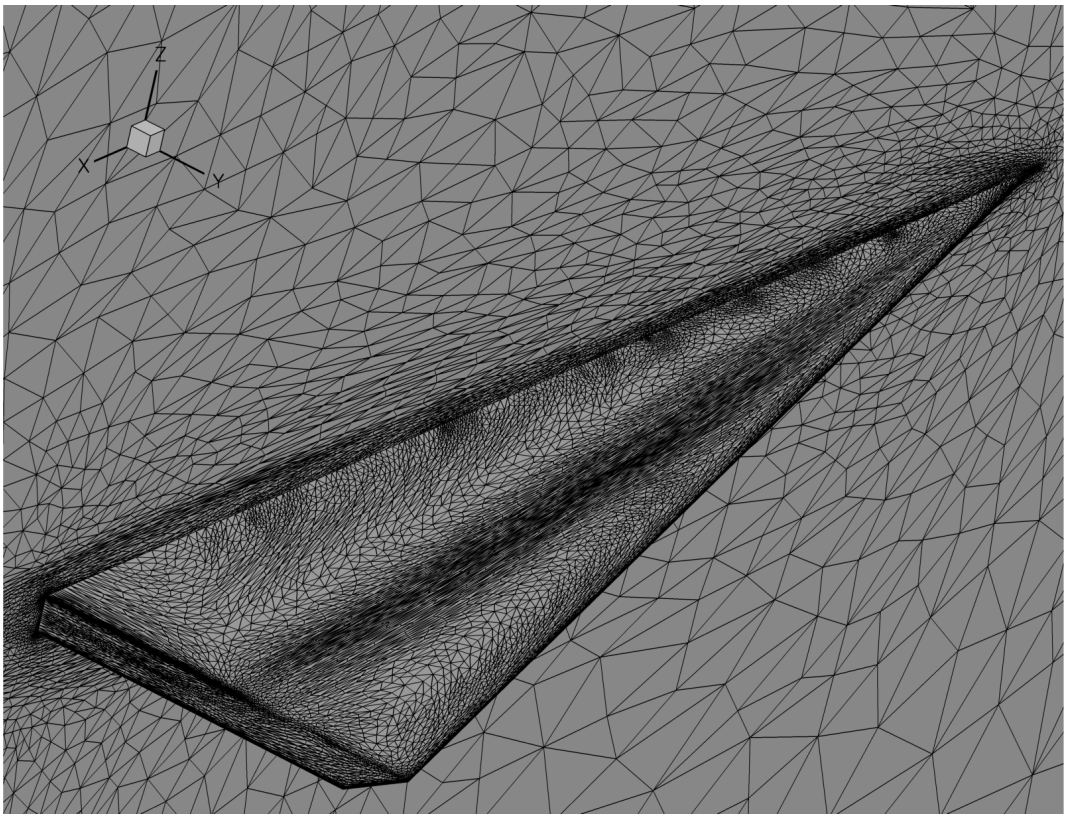


(a) Upper wing surface intersection with symmetry plane.

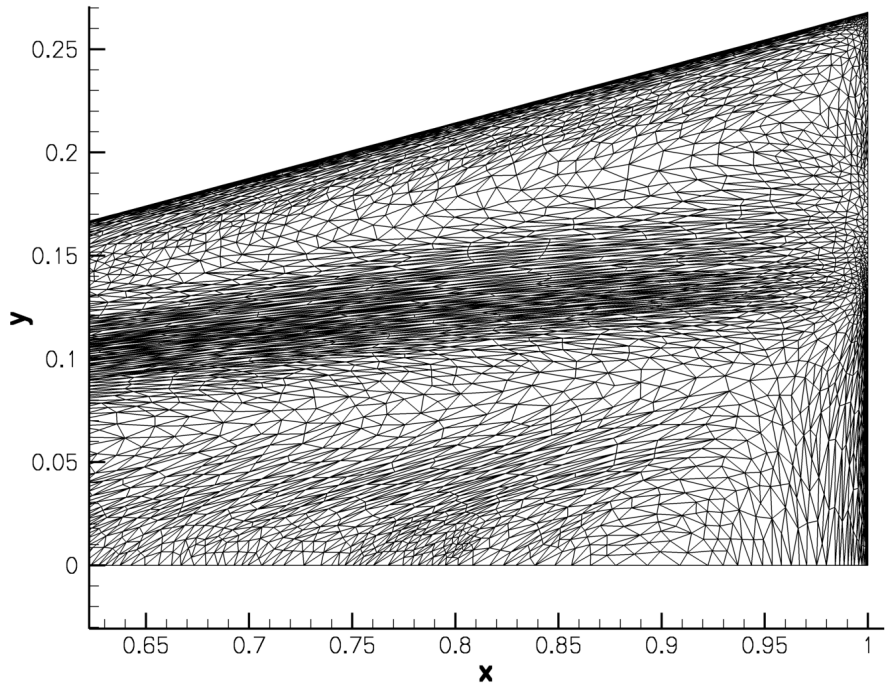


(b) Upper wing surface.

Figure 13. Final `Feflo.a` adapted grids for the Laminar Delta Wing.



(a) Upper wing surface intersection with symmetry plane.



(b) Upper wing surface.

Figure 14. Final EPIC-ICS adapted grids for the Laminar Delta Wing.

V.D. Turbulent Flat Plate (2D)

This case is described on the Turbulence Modeling Resource[5] in the turbulence model numerical analysis section as the 2D Finite Flat Plate Validation Case. The geometry and boundary conditions are reproduced in Fig. 15. The Spalart-Allmaras (SA)[35] turbulence model is used for this case. This case has a length of symmetry boundary conditions before and after the no-slip plate to create more distance between these similarities and the inflow and outflow boundaries. This case has been examined with uniform grid refinement[36] and adaptive grids.[37]

The complexity C_{plate} and maximum aspect ratio AR_{plate} are scheduled as

$$C_{\text{plate}} = \min(4000c, 80000) \tag{21}$$

and

$$AR_{\text{plate}} = 2^{0.5c}, \tag{22}$$

where c is the adaptive cycle index.

Drag convergence is shown in Fig. 16 for the first 23 grids. The series of adaptive grids appear to have a similar trajectory to CFL3D. This is suspected to be a coincidence, because other simulations (not shown) took a slightly different path. The Hessian recovery failed the on the 24th grid, computing not a number (NaN). The standard and 128-bit versions of `BAMG` is also applied (not shown), but failed on 8th grid adaptation when it was unable to find an enclosing triangle for inserting a new point. Skin friction is shown in Fig. 17 and profiles of velocity and eddy viscosity are shown in Fig. 18. The coarse isotropic initial grid and 23rd adapted grid are shown in Fig. 19. Zoom of grid near the end of the plate is shown in Fig. 20. The z-axis as been scaled and AR is limited with Eq. (22), which makes the grid appear nearly isotropic in Fig. 20. This vertical scaling identifies triangles with lower aspect ratio (i.e., vertical stretching on the lower boundary).

The skin friction computed on the adapted grid is similar to the skin friction calculated on structured grids. The adapted grid skin friction is linear closer to the leading edge of the plate than the structured grid results, see Fig. 17(a) and (b). There is a small amount of noise in the adapted grid skin friction. The locations of the noise appear to correspond to the locations of the lower aspect ratio clumps on the lower boundary seen in Fig. 20. These may be a result of Hessian recovery issues. The u-velocity profile is indistinguishable from the structured grid results in Fig. 18(a). The largest deviation in the turbulent eddy viscosity profile is very close to the plate, see Fig. 18(b). The adaptive result shows no undershoot at the edge of the boundary layer. The adaptive scheme uses first-order convection for the turbulence model and the uniformly refined schemes use second-order convection.

The small areas of grid clumping are suspected Hessian recovery issues on the lower boundary of Fig. 20. Without replacing Hessian recovery with averaging from the interior, inconsistent aspect ratios are requested, which produced grids with high amounts of noise in skin friction that stall the convergence of drag (not shown). This may be a contributing issue for the lower aspect ratio grid on the boundary exhibited by the equivalent FFMA method, which was reported by Yano and Darmofal.[38] Previous simulations did not converge the mean flow equations as well as the current simulation presented here. The remaining iterative error in these simulations is amplified by the Hessian recovery method and produces elements with inappropriate stretching and orientation in the interior (not shown). The "double L^2-projection" Hessian recovery method produced less noise than applying an unweighted least squares gradient reconstruction method[11] twice (not shown).

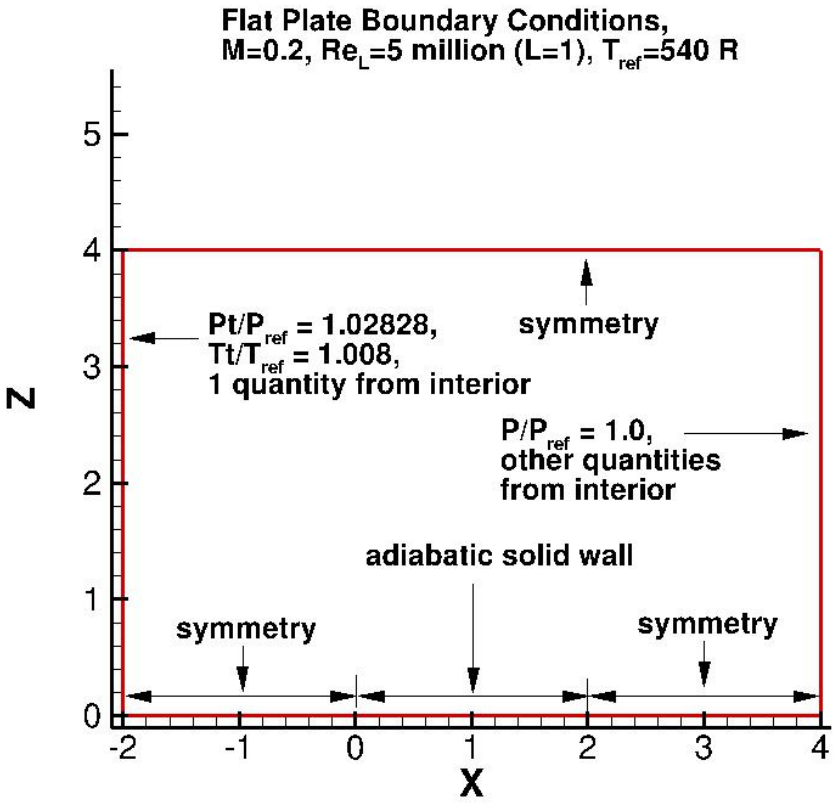


Figure 15. Turbulent flat plate geometry description.[5]

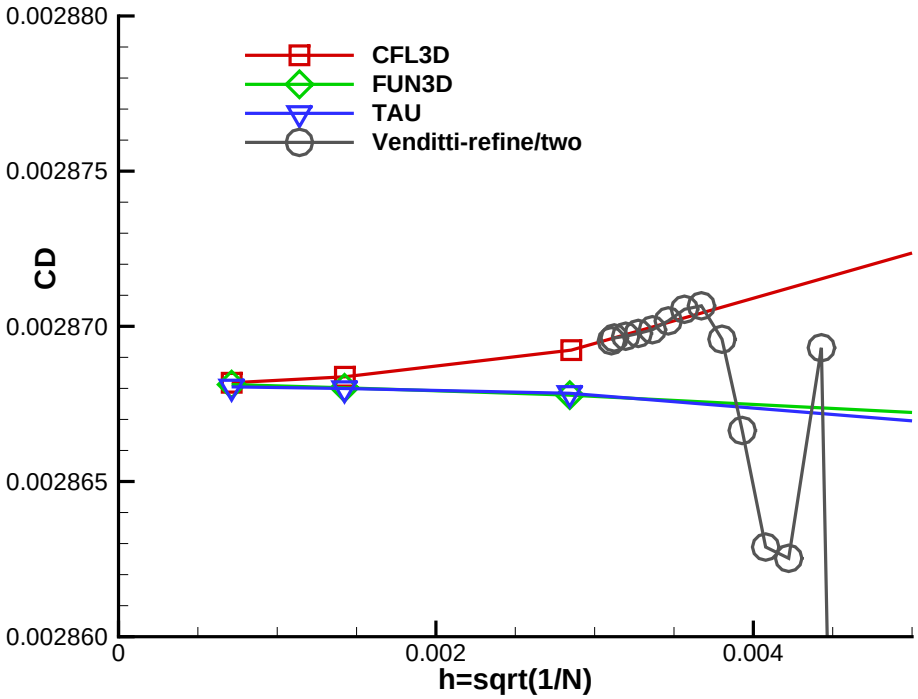
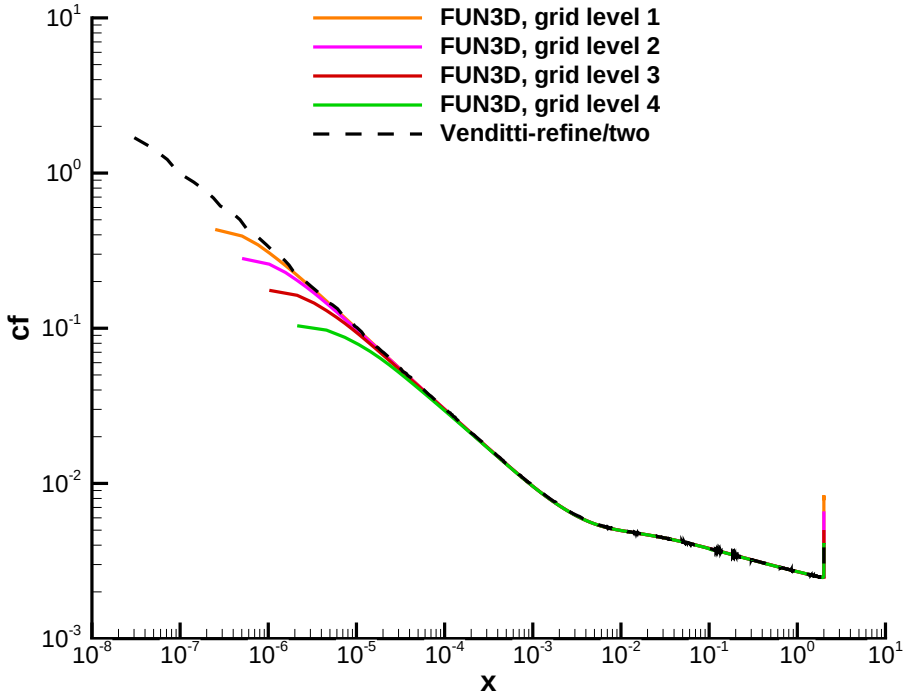
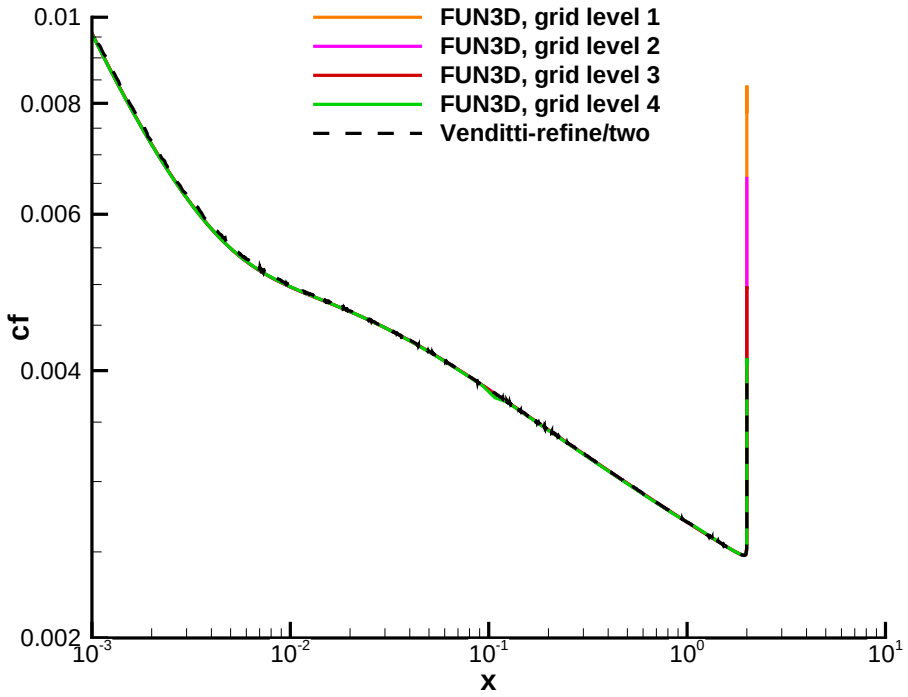


Figure 16. Turbulent flat plate drag convergence.

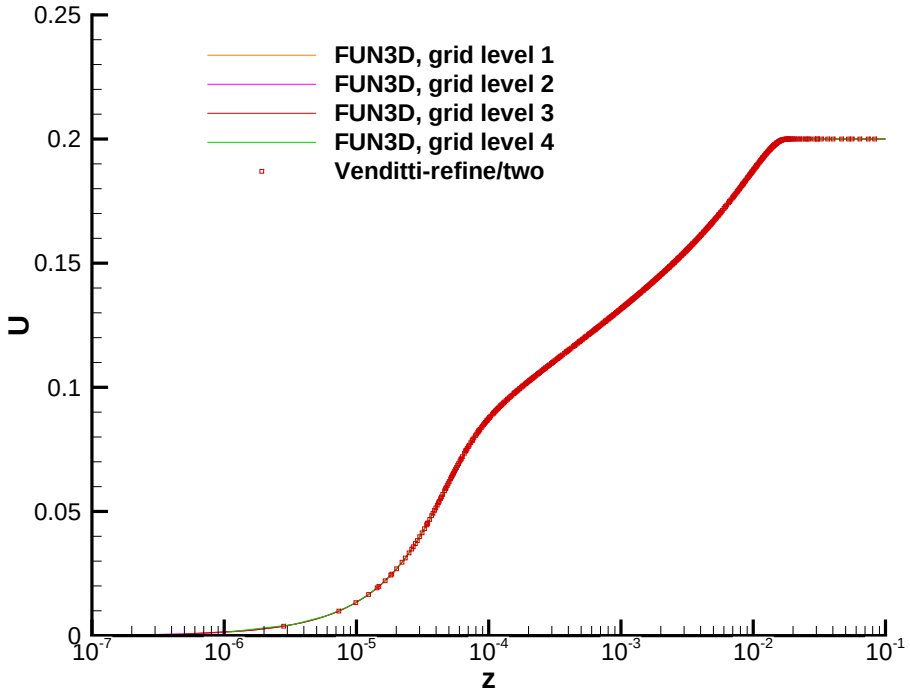


(a) Coefficient of friction.

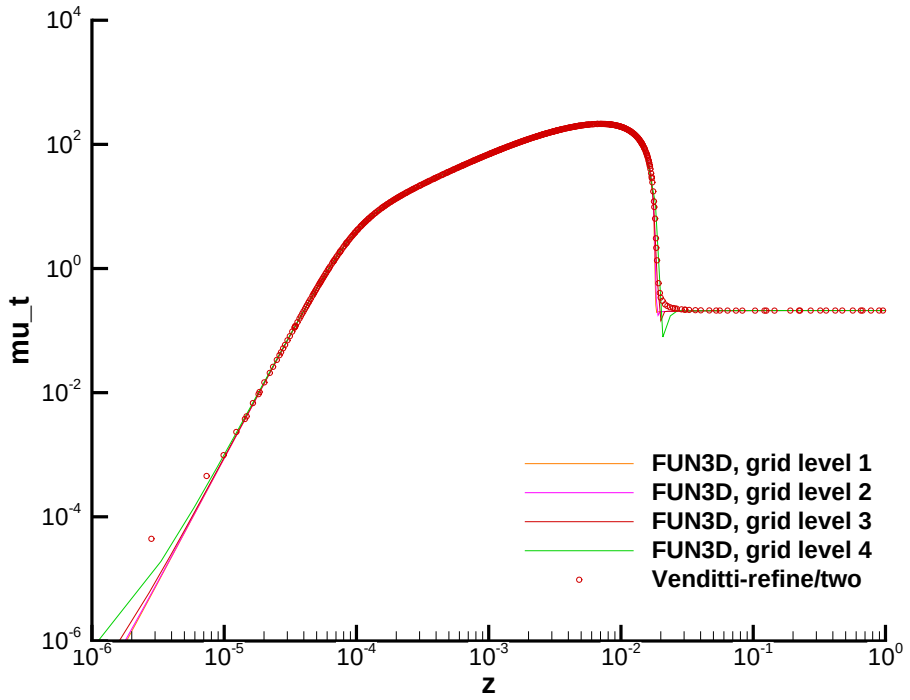


(b) Coefficient of friction (zoom).

Figure 17. Turbulent flat plate skin friction.

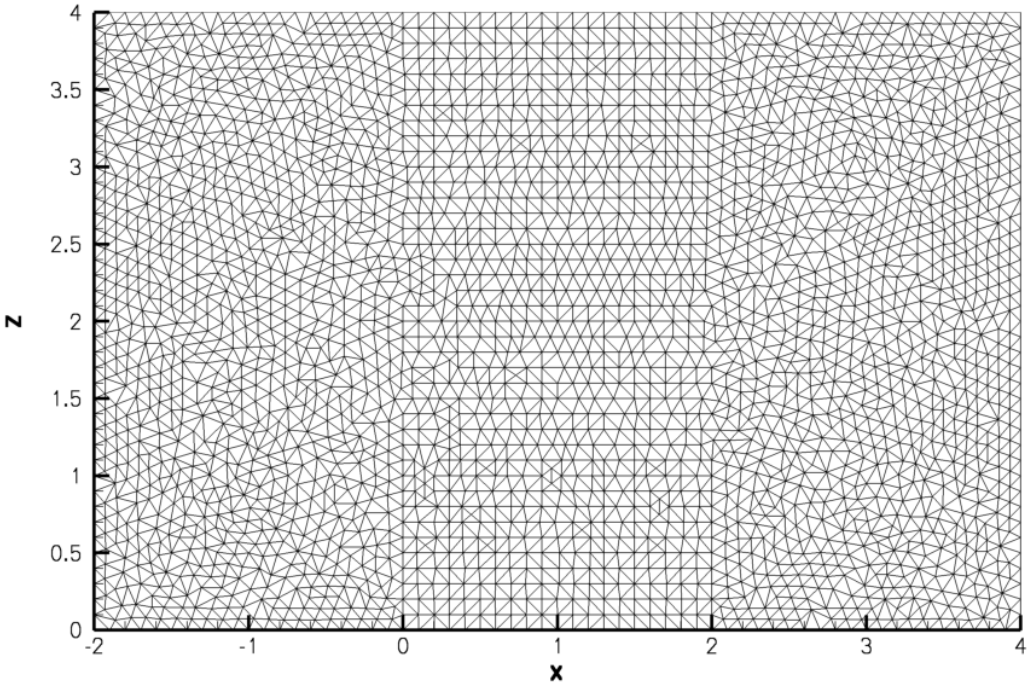


(a) Velocity profile at $X = 1$.

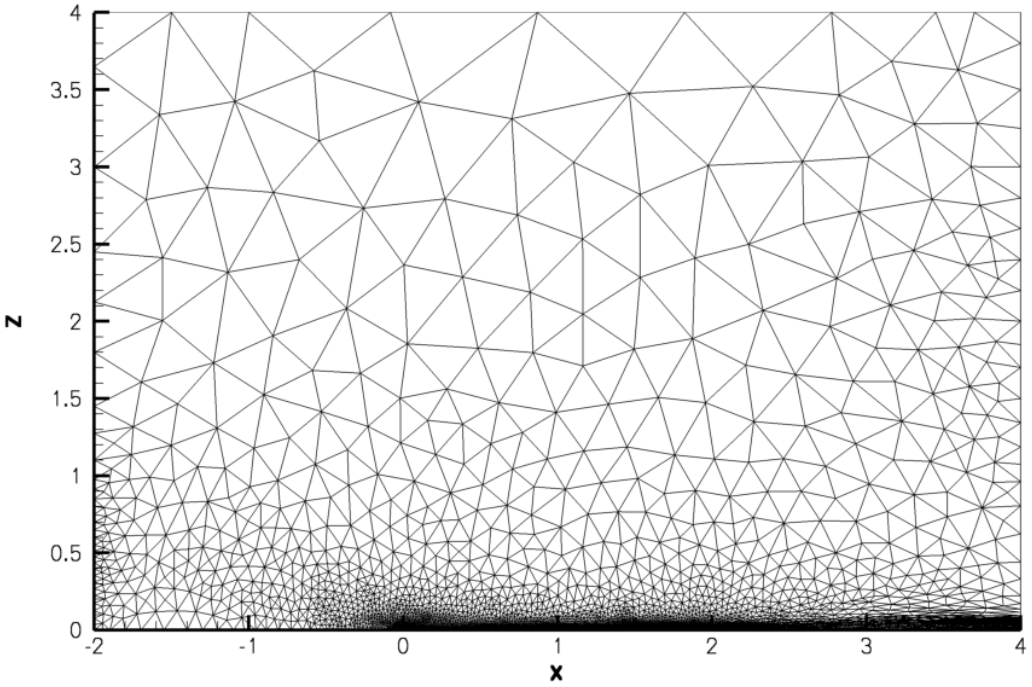


(b) Turbulent eddy viscosity profile at $X = 1$.

Figure 18. Turbulent flat plate profiles.



(a) Upper wing surface intersection with symmetry plane.



(b) Upper wing surface.

Figure 19. Turbulent flat plate initial and final grids.

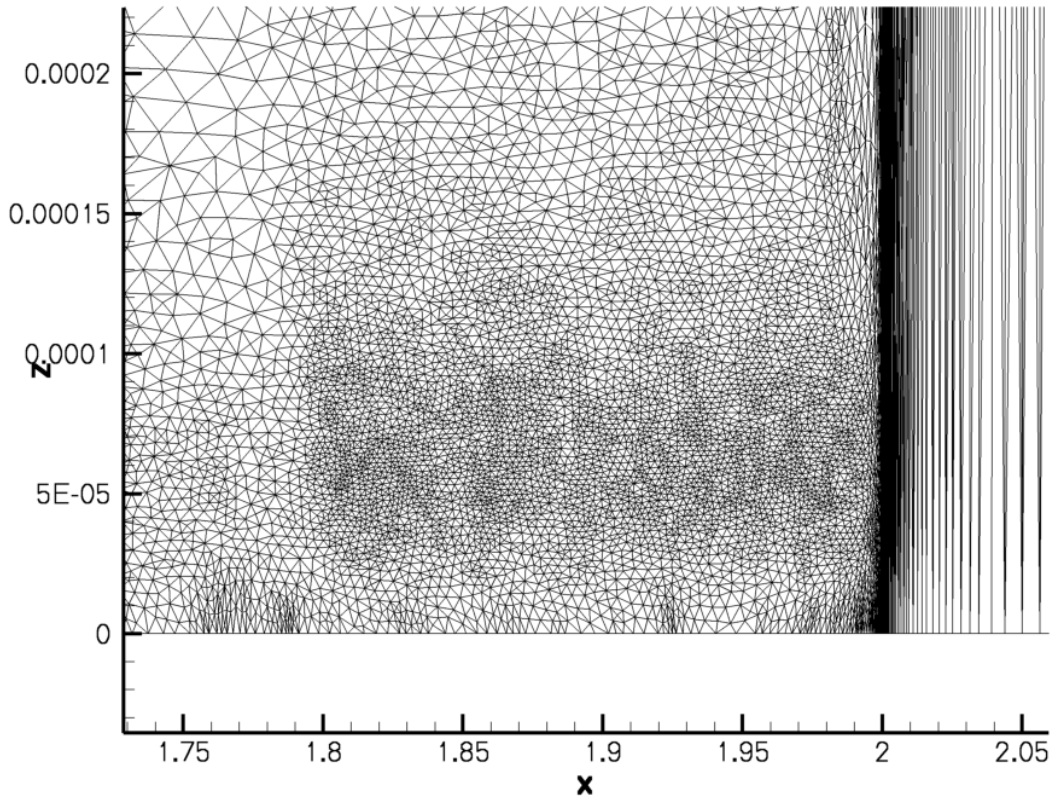


Figure 20. Turbulent flat plate final grid (zoomed and scaled vertically).

VI. Observations

Having a metric field that is realizable (can be reasonably approximated by a discrete grid) is critical to robustness of grid adaptation mechanics. Having a metric field that describes a new mesh that better approximates the continuous solution is critical to robustness of the entire adaptive process. The adaptive process can be made more robust to poorly sized or shaped elements by eliminating the direct dependency on the current grid spacing or the metric of the current grid simplicies (i.e., Opt-Goal metric).

Gradation control reduces discontinuous features and noise in the metric to improve the realizablity of the metric field. Hessian reconstruction, particularly on the boundary of the domain, is important for constructing a high quality metric field. To alleviate the known reduction in Hessian reconstruction accuracy on the boundary, the boundary Hessian reconstruction is replaced with averaging from the interior. This boundary reconstruction created a need for a center node in the extruded airfoil 3D case for averaging the Hessian to the symmetry planes.

A strong iterative solver for the flow and adjoint solution is critical for the robustness of the complete adaptation cycle. The flow solution must be converged well or iterative error can be amplified by Hessian recovery. When this Hessian is used to construct a metric, elements with low aspect ratio and inappropriate orientation can be created for the turbulent flat plate case.

The adaptive methods evaluated have ranges defined for acceptable long and short edges in the metric. These methods modify the grid in the neighborhood of these edges that are outside this acceptable range to produce a grid that conforms to the metric specification. The neighborhood of acceptable length edges are not modified. The choice of the acceptable length range in each scheme affects the statistics of edge length in the resultant adapted grid.

The implementation of the Opt-Goal metric and gradation limiting in FUN3D required detailed discussions with the authors of these methods. The availability of the test cases and results from this paper could make the implementation task easier and confirm correctness for these implementations. A database of grid adaptation examples would encourage the application of the Scientific Method[39] to speed the development of solution adaptive methods. Development of both turbulence modeling[6] and flight vehicle simulations[40] has been significantly impacted by this approach, which justifies the expenditure of effort to build these databases.

VII. Recommendations

Further development of a verified, validated, fast, and robust flow and adjoint solvers is required for the use of high aspect ratio anisotropic grids to compute turbulent simulations. Further examination and development of Hessian recovery is needed (especially near boundaries and in the presence of iterative error). This issue could be circumvented with a metric that does not require Hessian recovery.[4]

Gradation control implementations should be investigated by specifying a discontinuous metric and examining the resulting grids after the application of gradation control. To extend the use of statistics to solution adaptive grids, the metric can be interpolated to the adapted grid and the edge length statistics can be computed. Statistics of the conformity[24] of the adapted grid to the metric field can also be computed with this interpolated metric.

This paper has focused on simplex grid adaptation for planar geometries. There is an acute need for turbulent simulations of geometries with high curvature and complex topologies. New test cases should be constructed to address these needs (i.e., a more challenging analytic metric with high anisotropy intersecting a curved boundary). This paper utilized sequential execution with the exception of `refine/one` and `EPIC`. Evaluation of the scaling and correctness of parallel execution schemes for the grid mechanics are also required for large scale simulation.

VIII. Conclusions

An adaptive process is described and adaptive mechanics are evaluated by statistics of edge lengths in 2D and 3D analytic metric fields. Euler, laminar, and turbulent solution adaptive cases are investigated with the adaptive mechanics. Important components of each step in the process are highlighted: flow solver robustness, Hessian recovery (particularly near boundaries), and gradation control. These concrete test cases, evaluated by each author, fostered detailed discussions about low-level implementation details between the authors. Hopefully, this paper will encourage follow-on papers, future organized paper sessions at conferences, a workshop, a website, or other dissemination of tests cases and results to spur future development of solution adaptive methods.

IX. Acknowledgments

Jim Thomas, NASA Langley Distinguished Research Associate, is thanked for improving the solution scheme for the flat plate and tuning HANIM. Masayuki Yano, Massachusetts Institute of Technology, and David Darmofal, Massachusetts Institute of Technology, provided a version of `BAMG` with 128-bit floating point variables. Boris Diskin, National Institute of Aerospace, Troy Lake, University of Wyoming, and Bil Kleb, NASA Langley, provided insightful feedback on this article.

References

[1]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, April 2011, pp. 673–694.

[2]Park, M. A., Aftosmis, M. J., Campbell, R. L., Carter, M. B., Cliff, S. E., and Bangert, L. S., "Summary of the 2008 NASA Fundamental Aeronautics Program Sonic Boom Prediction Workshop," *AIAA Journal of Aircraft*, Vol. 51, No. 3, May–June 2014, pp. 987–1001.

[3]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.

[4]Yano, M. and Darmofal, D. L., "An Optimization-Based Framework for Anisotropic Simplex Mesh Adaptation," *Journal of Computational Physics*, Vol. 231, No. 22, Sept. 2012, pp. 7626–7649.

[5]http://turbmodels.larc.nasa.gov [cited 21 April 2015].

[6]Rumsey, C. L., Smith, B. R., and Huang, G. P., "Description of a Website Resource for Turbulence Modeling Verification and Validation," AIAA Paper 2010–4742, 2010.

[7]Peraire, J., Peirò, J., and Morgan, K., "Adaptive Remeshing for Three-Dimensional Compressible Flow Computations," *Journal of Computational Physics*, Vol. 103, No. 2, 1992, pp. 269–285.

[8]Alauzet, F., "Size Gradation Control of Anisotropic Meshes," *Finite Elements in Analysis and Design*, Vol. 46, No. 1–2, 2010, pp. 181–202.

[9]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.

[10]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.

[11]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.

[12]Biedron, R. T., 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: 12.6," NASA TM-2015-218690, Langley Research Center, March 2015.

[13]Nielsen, E. J., *Aerodynamic Design Sensitivities on an Unstructured Mesh Using the Navier-Stokes Equations and a Discrete Adjoint Formulation*, Ph.D. thesis, Virginia Polytechnic Institute and State University, 1998.

[14]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.

[15]Eisenstat, S. C., Elman, H. C., and Schultz, M. H., "Variational Iterative Methods for Nonsymmetric Systems of Linear Equations," *SIAM Journal on Numerical Analysis*, Vol. 2, April 1983, pp. 345–357.

[16]Pandya, M. J., Diskin, B., Thomas, J. L., and Frink, N. T., "Improved Convergence and Robustness of USM3D Solutions on Mixed Element Grids," AIAA Paper 2015–1747, 2013.

[17]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.

[18]Venditti, D. A., *Grid Adaptation for Functional Outputs of Compressible Flow Simulations*, Ph.D. thesis, Massachusetts Institute of Technology, 2002.

[19]Picasso, M., Alauzet, F., Borouchaki, H., and George, P.-L., "A Numerical Study of Some Hessian Recovery Techniques on Isotropic and Anisotropic Meshes," *SIAM Journal on Scientific Computing*, Vol. 33, No. 3, 2011, pp. 1058–1076.

[20]Vallet, M.-G., Manole, C.-M., Dompierre, J., Dufour, S., and Guibault, F., "Numerical comparison of some Hessian recovery techniques," *International Journal for Numerical Methods in Engineering*, Vol. 72, No. 8, 2007, pp. 987–1007.

[21]Park, M. A., "Adjoint–Based, Three–Dimensional Error Prediction and Grid Adaptation," *AIAA Journal*, Vol. 42, No. 9, Sept. 2004, pp. 1854–1862.

[22]Park, M. A., "Three–Dimensional Turbulent RANS Adjoint–Based Error Correction," AIAA Paper 2003–3849, 2003.

[23]Park, M. A., *Anisotropic Output-Based Adaptation with Tetrahedral Cut Cells for Compressible Flows*, Ph.D. thesis, Massachusetts Institute of Technology, Sept. 2008.

[24]Labbé, P., Dompierre, J., Vallet, M.-G., Guilault, F., and Trépanier, J.-Y., "A universal measure of the conformity of a mesh with respect to an anisotropic metric field," *International Journal for Numerical Methods in Engineering*, Vol. 61, No. 15, 2004, pp. 2675–2695.

[25]Hecht, F., "The Mesh Adapting Software: BAMG," INRIA Report, 1998, http://www.ann.jussieu.fr/hecht/ftp/bamg/.

[26]Park, M. A. and Darmofal, D. L., "Parallel Anisotropic Tetrahedral Adaptation," AIAA Paper 2008–917, 2008.

[27]Park, M. A., Laflin, K. R., Chaffin, M. S., Powell, N., and Levy, D. W., "CFL3D, FUN3D, and NSU3D Contributions to the Fifth Drag Prediction Workshop," *AIAA Journal of Aircraft*, Vol. 51, No. 4, July–Aug. 2014, pp. 2068–2079.

[28]Frey, P. J. and George, P.-L., *Mesh Generation: Application to Finite Elements*, ISTE Ltd. and John Wiley & Sons, 2nd ed., 2008.

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

[30]Alauzet, F., "A Changing-Topology Moving Mesh Technique for Large Displacements," *Engineering with Computers*, Vol. 30, No. 2, April 2014, pp. 175–200.

[31]Chang, H.-C. and Wang, L.-C., "A Simple Proof of Thue's Theorem on Circle Packing," *arXiv Mathematics*, Vol. Metric Geometry (math.MG), No. arXiv:1009.4322, Sept. 2010.

[32]Loseille, A. and Menier, V., "Serial and Parallel Mesh Modification Through a Unique Cavity-Based Primitive," 22nd International Meshing Roundtable, Sandia National Lab, 2013.

[33]Wang, Z. J., Fidkowski, K., Abgrall, R., Bassi, F., Caraeni, D., Cary, A., Deconinck, H., Hartmann, R., Hillewaert, K., Huynh, H. T., Kroll, N., May, G., Persson, P.-O., van Leer, B., and Visbal, M., "High-Order CFD Methods: Current Status and Perspective," *International Journal for Numerical Methods in Fluids*, Vol. 72, No. 8, July 2013, pp. 811–845.

[34]Leicht, T. and Hartmann, R., "Error Estimation and Anisotropic Mesh Refinement for 3D Laminar Aerodynamic Flow Simulations," *Journal of Computational Physics*, Vol. 229, No. 19, 2010, pp. 7344–7366.

[35]Spalart, P. R. and Allmaras, S. R., "A One-Equation Turbulence Model for Aerodynamic Flows," *La Recherche Aerospatiale*, Vol. 1, No. 1, 1994, pp. 5–21.

[36]Diskin, B., Thomas, J. L., Rumsey, C. L., and Schwöppe, A., "Grid Convergence for Turbulent Flows," AIAA Paper 2015–1746, 2015.

[37]Hu, Y., Wagner, C., Darmofal, D. L., Galbraith, M., and Allmaras, S. R., "Application of a Higher-order Adaptive Method to RANS Test Cases," AIAA Paper 2015–1530, 2015.

[38]Yano, M. and Darmofal, D. L., "An Optimization Framework for Anisotropic Simplex Mesh Adaptation: Application to Aerodynamic Flows," AIAA Paper 2012–79, 2012.

[39]Kleb, W. L. and Wood, W. A., "CFD: A Castle in the Sand?" AIAA Paper 2005–4873, 2005.

[40]Jackson, E. B., Madden, M. M., Shelton, R., Jackson, A. A., Castro, M. P., Noble, D. M., Zimmerman, C. J., Shidner, J. D., White, J. P., Dutta, S., Queen, E. M., Powell, R. W., Sellers, W. A., Striepe, S. A., Aguirre, J., Vuong, N., Reardon, S. E., Weinstein, M. J., Chung, W., and Berndt, J. S., "Further Development of Verification Check-Cases for Six-Degree-of-Freedom Flight Vehicle Simulations," AIAA Paper 2015–1810, 2015.