18th AIAA Computational Fluid Dynamics Conference
25 - 28 June 2007, Miami, FL

AIAA 2007-3955

Unstructured Mesh Discretizations and Solvers for Computational Aerodynamics

Dimitri J. Mavriplis *

Department of Mechanical Engineering, University of Wyoming, Laramie, Wyoming 82071-3295

An overview of current unstructured mesh discretization and solution techniques is given. Cell-centered versus vertex-centered discretizations are discussed, as well as issues of grid alignment with flow features, higher-order reconstruction, and viscous term formulation. With regards to solution techniques, implicit and multigrid methods are discussed, as well as efficient implementation on massively parallel computer hardware. Sample calculations are used to illustrate the importance of grid resolution and issues in solution accuracy. The examples are taken mostly from transonic flow calculations, and are used to argue that unstructured mesh technology has reached a level of maturity on a par with block-structured or overset mesh methods in this area, although important outstanding issues remain for other application areas such as high-speed flows.

I. Introduction

Unstructured mesh methods for computational fluid dynamics have been under development for over 25 years. The original attraction of this approach was based on their success at handling complex geometries, as demonstrated through finite-element-based approaches mainly in the field of structural analysis.[1] Thus, not surprisingly, the earliest applications of unstructured mesh technology to fluid dynamics came from within the finite-element community. In the early 1980's, using a least-squares Galerkin discretization of the full potential equation, what may well have been the first calculation of compressible transonic flow over a complete aircraft configuration was demonstrated by Bristeau et. al.[2] Despite these early successes, unstructured mesh techniques largely remained a curiosity within the aerodynamics community, where they were viewed as either too complicated or too costly to replace established structured mesh techniques. However, the turning point came once unstructured mesh methods were recast into the more familiar (for the aerodynamics community) finite-volume framework[3,4,5] for which approximate Riemann solver technology could be applied[5,6] and existing solution strategies could be adopted.[7,8,9,10,11] Since then, the development of unstructured mesh methods in computational aerodynamics has proceeded at a rapid pace. In addition to the flexibility for dealing with complex geometries, the ability to easily incorporate adaptive mesh refinement (AMR) strategies also became one of the often quoted advantages of the unstructured mesh approach.[12,13,14,15,16,17,18]

In the current day environment, unstructured mesh methods constitute one of the most prevalent approaches in computational aerodynamics. However, many outstanding issues remain with these approaches, and the promise of fully automated complex geometry handling with effortless adaptive meshing capabilities has yet to be realized.

On the one hand, it is surprising in hind-sight that one of the principal barriers to complex geometry handling remains the availability of robust high-quality unstructured mesh generation procedures. This in itself has left the door open for advances in alternate competitive approaches such as overset meshes or cut-cell Cartesian approaches. On the other hand, adaptive mesh refinement methods have not seen widespread adoption in production environments, as may have been predicted 20 years ago, partly due to the difficulty in dynamically load balancing such operations on massively parallel computer architectures, but also in large part to the lack of reliable error estimation techniques for driving the refinement criterion.

*Professor, AIAA Associate Fellow; email: mavripl@uwyo.edu.

Although it is difficult and somewhat artificial to separate out all the individual effects of mesh generation, adaptivity, and flow solution, the current paper will only focus on the effects of the flow solver itself, which include discretization, solution, and parallel implementation. Prior to discussing the strengths and weaknesses of unstructured mesh approaches in these areas, it is worth examining the alternative available approaches and identifying the defining strategies which separate unstructured mesh approaches from competing methods.

As stated above, the principal motivators for the adoption of unstructured mesh methods have been the handling of complex geometries, and the use of adaptive mesh refinement techniques. With regards to complex geometries, the principal alternatives have been the use of block structured grids, or overset grids. With regards to adaptive meshing strategies, these can be performed on any type of mesh, and the defining difference is whether the adapted mesh is described as a list of refined patches, each containing a regular structure within, or as a fully unstructured list of mesh elements. Similarly, a block-structured mesh can be represented easily using a full unstructured data-set. On the other hand, a fully unstructured tetrahedral mesh must employ an unstructured data-set, and the tetrahedral element type is not typically amenable for use in structured or block-structured meshes. However, unstructured meshes can be constructed using any element type, including hexahedral elements typical of structured mesh topologies, as well as arbitrary mixtures of various polyhedral cells.

Thus, the defining differences between unstructured meshes and structured or block-structured meshes lies on the one hand in the use of an unstructured data-set, and on the other hand, in the potential use of different element types. These two aspects must be examined separately in order to assess the strengths and weaknesses of unstructured mesh approaches. Briefly stated, the element type principally affects spatial discretization accuracy, while the use of unstructured data-sets is principally responsible for efficiency concerns.

II. Discretization Issues

We will first consider the differences between cell-centered and vertex-based discretizations, and then examine the individual aspects of current discretization approaches, including first-order inviscid discretizations, strategies for achieving second-order accuracy for these terms, and finally, viscous term discretizations.

A. Cell or Vertex Approaches

Perhaps another surprising situation in hindsight, is that after twenty years of debate, the determination of which discretization approach, cell-based or vertex-based, is most effective for unstructured mesh methods, has still not been resolved definitively.

Whereas for purely hexahedral meshes, the numbers of cells and vertices are equivalent, excluding boundary effects, a tetrahedral mesh of N vertices contains from 5N to 6N cells. This can be seen simply by constructing a tetrahedral mesh by subdividing each cell of a hexahedral mesh into 5 or 6 tetrahedra, with no additional vertices. Thus, a cell-centered approach on a tetrahedral mesh will contain many more degrees of freedom than a vertex-centered discretization on the same mesh, and can therefore be expected to yield higher accuracy and require higher computational expense. However, the cell-centered discretization results in a relatively sparse stencil, with each tetrahedron having only four neighbors, whereas in the vertex-based discretization each vertex has on average 14 neighbors, based on the number of edges in the mesh, which can be shown to of the order of 7N, for a mesh of N vertices. The vertex-based discretization can therefore be expected to be more accurate than a cell-based discretization using equivalent numbers of unknowns, since the former approach will result in a larger number of flux calculations. Additionally, the larger stencil has the potential for more robust reconstruction techniques and limiting procedures. In the final assessment, the most effective discretization is the one which provides the highest accuracy at the lowest cost. Although numerical experiments conducted by the author and others[7,19,20,21] have verified the superior accuracy of cell-centered approaches versus vertex-based approaches on identical grids, but also suggested the vertex-based approach to be the most efficient approach overall, the issue has never been decided conclusively, in large part due to the lack of fully consistent comparisons between the two approaches using identical discretizations and solvers. On the other hand, the continuing lack of progress in robust unstructured mesh generation technology has resulted in an unexpected advantage for the cell-centered approach, due to the ability of these discretizations to achieve acceptable accuracy on relatively coarse

meshes.

The definition of an "equivalent" grid for comparing cell-based and vertex-based discretizations at equivalent accuracy levels remains an open question. A study by Levy[22] found that matching the number of surface grid variables for both grids achieved similar accuracy for aerodynamic quantities in transonic flow cases. Since there are twice as many triangular elements on the surface as vertices in a given grid, this results in vertex-based grid sizes which on average contain approximately three times more vertices overall than the equivalent meshes for cell-centered discretizations. This approach was used in the recent Drag Prediction Workshop series (DPW)[23,24,25] for establishing baseline "equivalent" grids for the two different discretization methods. An example is given in Table 1, for a set of meshes generated on the DLR-F4 wing body configuration, which was the subject of the first workshop,[23,26,27] where "equivalent" vertex-based and cell-based meshes are described, including a version of the cell-centered discretization based mesh for use with wall functions (which requires lower normal wall resolution with no change in the tangential resolution).

While this approach for defining "equivalent" resolutions for cell-based and vertex-based meshes has been studied for transonic flows, it is understood that the equivalency may not apply equally for other types of flows, particularly for cases where off-body flow-physics are of primary importance.

Table 1. *Detailed description of three "equivalent" unstructured meshes for computation of flow over DLR-F4 wing-body configuration. Reproduced from*[27]

Grid Characteristics	Vertex-Based Grid (Full Viscous)	Cell Based Grid (Full Viscous)	Cell Based Grid (Wall Function)
Boundary Points	48,339	23,290	25,175
Surface Triangles	96,674	46,576	50,346
Triangles on no-slip surfaces	72,902	30,037	38,571
Total Grid Points	1,647,810	470,427	414,347
Points in Viscous Layers	1,129,427	389,753	238,301
Tetrahedral Cells	9,686,802	2,743,386	2,390,089
Cells in Viscous Layers	6,495,828	2,208,260	1,281,854
Maximum Number of Viscous Layers	35	35	12
Number of Complete Viscous Layers	24	24	7
Grid Points Across Wing T.E.	5	5	5
Chordwise Grid Spacing at L.E.	≈0.250mm	≈0.450mm	≈0.450mm
Chordwise Grid Spacing at T.E.	≈0.500mm	≈0.800mm	≈0.800mm
Maximum Spanwise Spacing at L.E.	≈2.500mm	≈6.000mm	≈6.000mm
Maximum Spanwise Spacing at T.E.	≈3.500mm	≈3.500mm	≈3.500mm
Grid Spacing on Fuselage	≈10.00mm	≈10.00mm	≈10.00mm
Grid Spacing at Outer Boundary	≈3000.00mm	≈3000.00mm	≈3000.00mm
Normal Spacing at No-Slip Walls	0.001 mm	0.003 mm	0.0549 mm
Rate of geometric stretching (viscous layers)	1.2	1.2	-
Outer Boundary Box Size	50 mean chords	50 mean chords	50 mean chords

The appearance of edge-based and face-based data-structures for vertex and cell-centered discretizations, respectively, has made it possible to construct a solver which can be operated in either mode, with relatively minor modifications mostly concerning boundary conditions, at least for inviscid flows. Although this has seldom been done, it would provide a more solid basis for comparing cell versus vertex-type discretizations.

Note that this issue is most important for tetrahedral meshes, and is only of secondary importance on hexahedral meshes. For prismatic element meshes, such as those often used in boundary layer regions, the differences between cell-centered and vertex-based approaches are less pronounced than for tetrahedral meshes, but still significant. It is also worth noting that in the consideration of higher-order methods, the traditional continuous finite-element strategy, which extends to higher order through streamwise-diffusion and Petrov-Galerkin methods,[28] corresponds to a vertex-based discretization, whereas the Discontinuous Galerkin approach[29] can be viewed as a straightforward extension of the cell-centered approach to orders higher than one.

Another principal difference between cell-centered and vertex-centered discretizations relates to the application of boundary conditions. Since a vertex-based approach results in degrees of freedom being located directly on the boundary, this would seem to facilitate the implementation of Dirichlet boundary conditions, while the cell-centered approach is well suited for Newman boundary conditions. However, a more fundamental problem arises in the vertex-

based discretization, since individual boundary vertices may have ill-defined boundary conditions if they are located at the intersection of two faces with different boundary conditions, as shown in Figure 1. Note that this situation never occurs for the cell-centered discretization. The is due to the fact that the elemental unit of the computational boundary corresponds to a mesh face, rather than a mesh vertex. Thus it is much more appropriate to define boundary conditions based on mesh faces rather than mesh vertices. In order to implement such a boundary condition in the case of vertex-based discretizations, a weak formulation must be used, where the boundary condition is introduced into the residual through a modified boundary flux. This obviates the need to assign boundary conditions to the mesh vertices, and results in boundary condition formulations which are similar for both the cell-centered and vertex-centered discretizations, as shown in Figure 1.

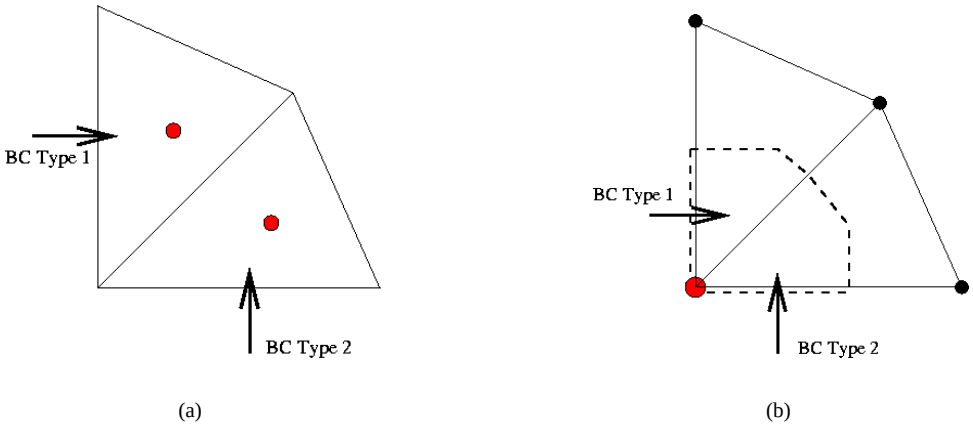
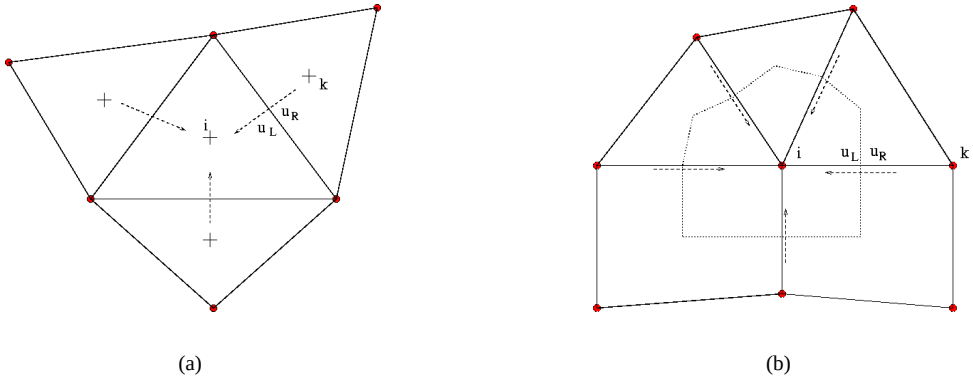


Figure 1. Illustration of boundary condition implementation for (a) cell-based and (b) vertex-based discretizations.

B. Inviscid Discretization

1. First order

In computational aerodynamics, the basis for most unstructured grid discretizations of the convective terms relies on the finite-volume analogy, although many of these can be shown to be analogous to finite-element approaches. For a cell-centered scheme, the unknowns are located at the cell centroids, and the control-volume is taken as the mesh element itself. Therefore, the control-volume flux balance requires the computation of fluxes across the faces of the mesh element, as shown in Figure 2(a), for the two-dimensional case. For vertex-based approaches, the unknowns are located at the mesh vertices, and a control-volume surrounding each vertex must be constructed. These are most often taken as the volume formed by joining the centroids of the surrounding cells to the edge mid-points, known as the centroidal-dual control-volume, as shown in Figure 2(b), although other constructions are possible.[30,31] Fluxes are then calculated across each composite face of these dual control-volumes, which can be associated with an edge in the original mesh, thus leading to the edge-based data-structure for vertex-based discretizations. The advantage of using face-based or edge-based data-structures (for the cell-centered or vertex-based schemes respectively) is that meshes involving mixtures of arbitrary polyhedral cell types may be represented using a single homogeneous data-structure, which has implications for computational efficiency and memory footprint.[8,32] Fluxes across control-volume faces are generally computed using approximate Riemann solvers, at least for density-based compressible flow formulations prevalent in aerodynamics, and taking the left and right values in the Riemann solver as the values at the cell centers or vertices on either side of the control-volume face results in a first-order accurate spatial scheme. Since this strategy mimics almost exactly what is done on structured meshes as far as first-order accurate schemes are concerned, there is little difference between the behavior of such schemes on structured and unstructured grids. However, one important difference lies in the possibility of aligning the grid with flow features such as boundary layers and shock waves, which



(a) (b)

Figure 2. Illustration of (a) cell-based and (b) vertex-based control-volume constructions.

is generally simpler with structured grid approaches compared to unstructured grid methods. Thus, accuracy may be degraded with unstructured meshes due to large numbers of faces which find themselves at arbitrary angles to these flow features, as opposed to a well "fitted" structured hexahedral grid which can be constructed with most faces either tangent or normal to the important flow features.

There are two approaches for dealing with these difficulties with unstructured meshes. The first approach seeks to improve the discretization scheme through the development of multi-dimensional approximate Riemann solvers, which would be insensitive to the orientation of the face along which the flux is computed.[33,34] Although progress has been made in this area, this remains a difficult problem which has to date not been resolved satisfactorily. The alternate approach consists of attempting to better align the unstructured mesh with the important flow features, either adaptively, or through a more careful generation procedure using alternate element types. This latter approach is reflected in the common practice of the use of prismatic elements in near-wall regions which are aligned with the flow direction for boundary layer resolution.[35,36,37,38] For flows with strong shocks, non-aligned grid faces have been shown to produce entropy waves which can have a compromising effect on important downstream objectives such as surface heating. One approach consists of adaptively aligning the grid faces to the developing shock front, through mesh movement techniques, as previously demonstrated in two dimensions.[39,40] Current on-going work in the author's research group has concentrated on using adjoint-based sensitivities to guide the shock alignment problem for unstructured meshes, as shown in Figure 3. In this two-dimensional oblique shock wave example, for which the exact analytical solution is known, an arbitrarily oriented unstructured mesh is seen to produce relatively large entropy errors downstream of the shock. By defining an objective based on the difference between the computed solution and the analytic solution, and minimizing this objective through an optimization procedure where the grid point coordinates constitute the design variables, the mesh is seen to align with the shock, and the exact solution is closely recovered. One of the problems with this approach is the formulation of a suitable objective function or error estimator for more realistic cases where the exact solution is not known. Additionally, the extension of this approach to three dimensions has not been investigated.

2. Second order

Useful engineering calculations generally require second-order spatial accuracy (at a minimum), and this is most often achieved through reconstruction techniques or other approaches which result in extended stencils involving neighbors of neighbors. In general the approximate Riemann problem to be solved at each mesh face or edge can be written as:

$$\mathbf{F}_{ik} = \mathbf{F}(u_L, u_R) = F(u_L) + F(u_R) + T|\Lambda|T^{-1}(u_L - u_R) \tag{1}$$

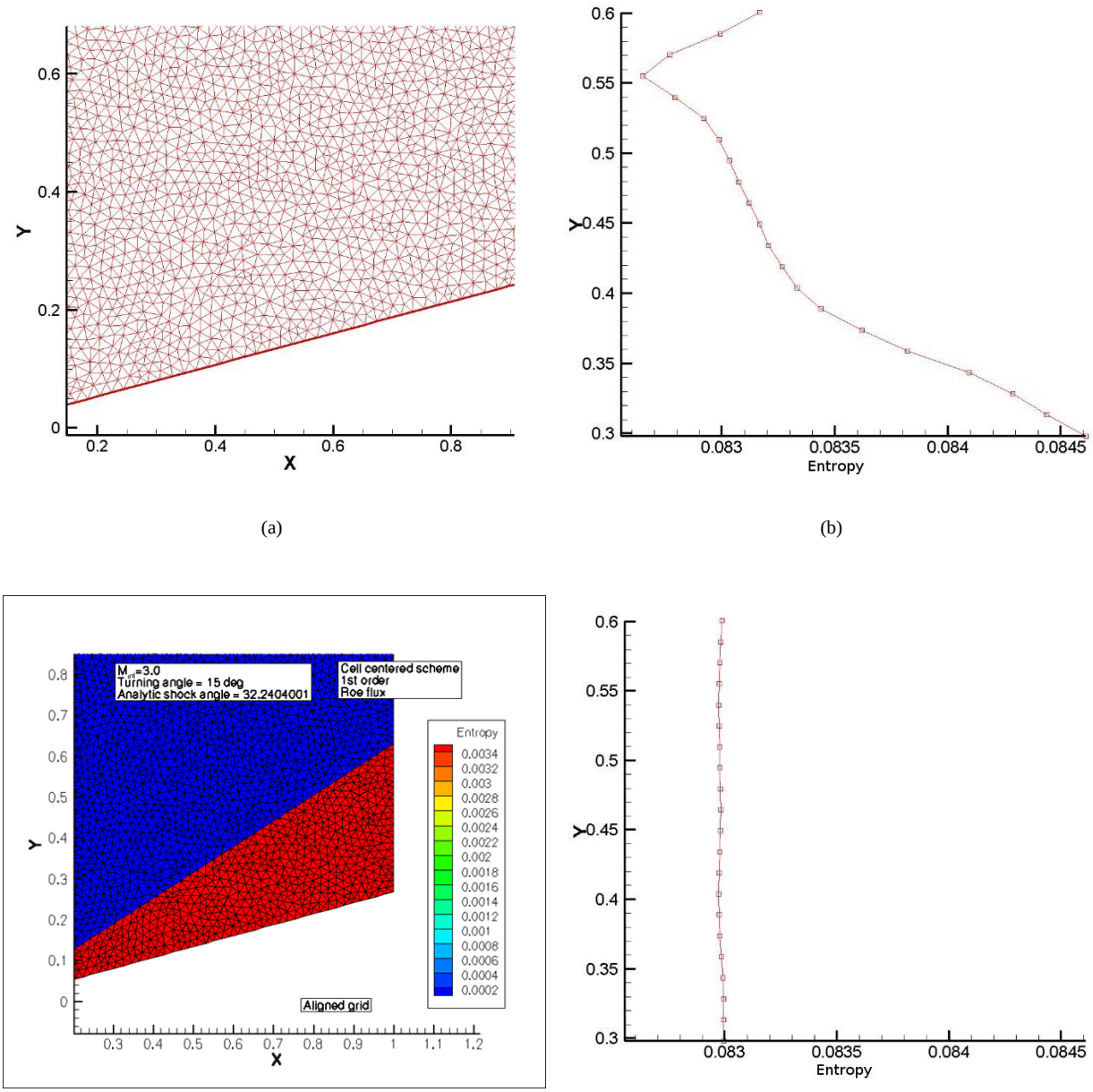


Figure 3. Illustration of shock fitting through mesh face alignment in two-dimensions.

where F(u) represents the convective flux, and where u_L and u_R represent the values of the flow variables at the left and right sides of the control volume interface. For a first-order scheme, these are simply taken as the values at the vertices corresponding to the control volume on either side of the face:

$$u_L = u_i \tag{2}$$

$$u_R = u_k \tag{3}$$

To obtain second-order accuracy, the left and right states must be obtained by extrapolating the control volume values based on a reconstructed gradient. Thus, the second-order accurate scheme is obtained using:

$$u_L = u_i + \nabla u_i . \vec{r}_{if} \tag{4}$$

$$u_R = u_k + \nabla u_k . \vec{r}_{kf} \tag{5}$$

where $\vec{r}_{if}$ denotes the position vector extending from vertex i to the control volume interface, and the gradients ∇u are to be evaluated at the mesh vertices or cell centers, for a vertex or cell-based scheme, respectively.

For vertex-based discretizations, the earliest approaches to achieving higher-order accuracy on unstructured meshes involved the use of one-dimensional solution gradient information obtained in the direction of the current edge for which the flux is to be computed, in a manner similar to that used for structured grids. This slope information is obtained by locating the the intersection of lines drawn tangent to the current edge with surrounding triangles, from which solution values are interpolated, as shown in Figure 4. A similar approach was later adopted by the CUSP schemes developed by Jameson.[41] However, the requirement of additional data-structures to represent the cells intersected by the extended edges remains a disadvantage of these approaches.

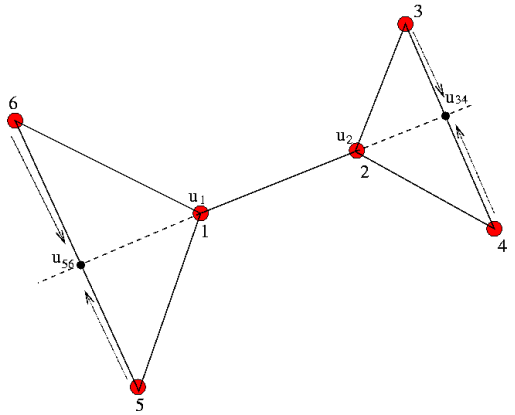


Figure 4. Illustration of one-dimensional reconstruction technique used in reference[5] and by CUSP schemes in reference.[41]

By far the most common approach today is based on the multi-dimensional reconstruction ideas advanced by Barth and Jespersen.[6] In this approach, a gradient at each control-volume center (cell or vertex) is constructed using all available nearest neighbor surrounding information, either by a Green-Gauss contour integral, or using a least-squares fit. Several issues within the gradient reconstruction process have been identified as having potential important impacts on overall solution accuracy and robustness. Solution robustness may be adversely affected by noisy gradients or gradients with inappropriate peaks, since this leads to over-estimation of solution variations and potentially non-physical states. This has been the motivation for the adoption of the least-squares gradient reconstruction approach over the Green-Gauss approach, which is seen to be less sensitive to grid irregularities and more reliable on highly stretched meshes.[42,31,43,44] However, in the context of the least-square gradient reconstruction, various weightings of the least-squares approach may be considered, and the unweighted approach has often been favored over the potentially more accurate distance weighted approach, for robustness reasons. However, serious drawbacks of the unweighted approach have been exposed, particularly for highly-stretched meshes in the presence of surface curvature[45,46] as will be shown in subsequent examples.

For flows with strong shock waves, the use of limiters becomes necessary for preventing solution overshoots which may compromise accuracy and stability. Limiters may also be used to improve robustness in the presence of poor gradient reconstruction accuracy caused by a lack of sufficient grid quality. The function of a limiter is to reduce the magnitude of the reconstructed gradient to ensure stability, ultimately zeroing out the gradient value in extreme cases, and reducing the method to a first-order accurate scheme locally. However, current limiter use is known to inadvertently impact global accuracy as well as prevent robust convergence due to non-smooth application and limit-cycle behavior. These problems can be mitigated to some degree using TVB limiters,[47] although further work on limiter accuracy and convergence is still warranted today.

An alternate strategy for achieving second-order accuracy is through the use of artificial dissipation terms. In this approach, a third difference is constructed at each control-volume interface, as a difference between two pre-computed second differences or undivided Laplacian operators on each side of the interface, and then multiplied by the appropriate (largest) eigenvalue of the flow field at the interface condition to create a dissipative term as:

$$\mathbf{F}_{ik} = F(u_i) + F(u_k) + \alpha\,(L_i(u) - L_k(u)) \tag{6}$$

where $L_i(u)$ represents an undivided Laplacian operator, taken as:

$$L_i(u) = \sum_{k=1}^{neighbors} (u_k - u_i) \tag{7}$$

This approach is known as the *scalar* artificial dissipation approach, which corresponds to a Lax-Friedrichs scheme,[48] and is known to result in excessively diffusive results. An improvement over this approach consists of taking α as a matrix rather than a scalar. In this case, a natural choice for α, by analogy with equation (1) is:

$$\alpha = \kappa_2\, T|\Lambda|T^{-1} \tag{8}$$

where κ_2 is a constant. Thus, the second-order accurate matrix dissipation scheme can be obtained by replacing the difference of reconstructed states in the projection-evolution scheme by a difference of undivided Laplacian operators.[9] Although these quantities are of the same order, they are not directly proportional to each other, and therefore the parameter κ_2 cannot be taken as unity in this case, but must be determined empirically. There are also discrepancies between the centrally differenced convective fluxes in both schemes, since these are evaluated at reconstructed states in the upwind scheme, rather than at vertex values as in the artificial dissipation scheme, although this does not reduce the formal accuracy of the scheme. The advantage of the matrix dissipation scheme is that it bypasses the need for gradient reconstruction with all the problems involved in that procedure, although the inexact correspondence to an approximate Riemann flux function makes the method questionable for high-speed flows. However, for transonic flows, the matrix dissipation formulation has been found to be more accurate and often more robust than the upwind-reconstruction approach described above.

The T matrices on the right hand side of equation (1) represent the eigenvectors associated with the linearization of the equations of inviscid compressible flow normal to the control volume face ik,[49] while the $|\Lambda|$ matrix is a diagonal matrix containing the absolute values of the five eigenvalues associated with these equations. Of these five eigenvalues, three are repeated, leaving three distinct eigenvalues which are proportional to: u, u+c, u-c, where u is the velocity normal to the control volume face, and c is the speed of sound. When one of these eigenvalues vanishes, the dissipation for that component at that location also vanishes, which may lead to numerical instabilities. For this reason, it is common to limit the eigenvalues to a minimum fraction of the maximum eigenvalue, such as:

$$u = sign(u) * max(|u|, \delta(|u| + c)) \tag{9}$$

$$u + c = sign(u + c) * max(|u + c|, \delta(|u| + c)) \tag{10}$$

$$u - c = sign(u - c) * max(|u - c|, \delta(|u| + c)) \tag{11}$$

where $|u| + c$ is the maximum eigenvalue, and δ is a parameter to be chosen empirically which varies between 0 and 1. When δ is taken as 0, no eigenvalue limiting is applied. When δ is taken as 1, the $|\Lambda|$ matrix reverts to a scaled identity

matrix, and the scalar artificial dissipation or the Lax-Friedrichs flux is recovered.[48] Small values of δ of the order of 0.1 are common in many production codes, and this process is often referred to as an *entropy fix*.[50]

These various forms of the convection term discretization discussed above are compared on a sample transonic aerodynamic test case. The baseline test case consists of the transonic flow over the DLR-F4 wing-body configuration at a Mach number of 0.75 and a Reynolds number of 3 million, computed on the 1.65 million point unstructured mesh detailed in Table 1. Although results on finer grids are available, the computational results on this mesh agree reasonably well with the experimental values, and the use of meshes of this resolution level should accentuate the differences between discretizations which we are seeking, since all the considered discretizations are consistent, and should converge towards the same result as mesh resolution is continually increased.

For the purposes of this study, the baseline discretization consists of the matrix artificial dissipation scheme with the parameter settings $\kappa_2 = 1.0$ and $\delta = 0.1$. Figures 6 and 7 illustrate the sensitivity of computed lift and drag coefficients throughout the range of incidences to the variation in these parameters in the artificial dissipation discretization. Precise numerical values are recorded in Table 2, for the single incidence of 0 degrees. In all cases, a more accurate baseline computation using a finer grid of 13 million points is given for comparison purposes. The results show that lowering the nominal dissipation scaling factor ($\kappa_2 = 0.5$) produces slightly higher lift values, and moderately lower drag values. From Table 2, the lift is seen to increase by 16 counts, while the drag decreases by 6 counts at 0 degrees incidence. The lower dissipation value appears to yield higher accuracy, as the results tend towards those computed on the finer grid. This is expected, since one of the main characteristics of increased grid resolution is the reduction of artificial dissipation effects. While lowering the dissipation coefficient κ_2 below its nominal level can lead to more accurate solutions at little additional cost, numerical instabilities may develop due to insufficient dissipation levels, and the loss of robustness associated with this approach may not acceptable in a production environment.

The sensitivity due to the value of the entropy fix in the artificial dissipation discretization is depicted in Figure 7. Doubling the entropy fix value from $\delta = 0.1$ to $\delta = 0.2$ has very little effect on the computed results. Differences of less than one count in lift and drag are observed in Table 3. This indicates that the solutions are not very sensitive to small values of the entropy fix parameter, and non-zero values are acceptable for increasing robustness while minimally impacting accuracy. On the other hand, when the value of the entropy fix parameter is increased to $\delta = 1.0$, the drag increases by a substantial amount (25 counts at 0 degrees incidence from Table 2). The setting $\delta = 1.0$ corresponds to a purely scalar artificial dissipation scheme, which can be evaluated at reduced cost compared to the matrix dissipation formulation. However, these savings are counterbalanced by the lower accuracy of the scalar approach, which in turn may require the use of finer meshes to regain acceptable accuracy levels. Note the importance of considering both lift and drag values in this case, since the small increase in lift values associated with the scalar dissipation computations may, in isolation, convey an impression of higher accuracy with regards to the baseline case.

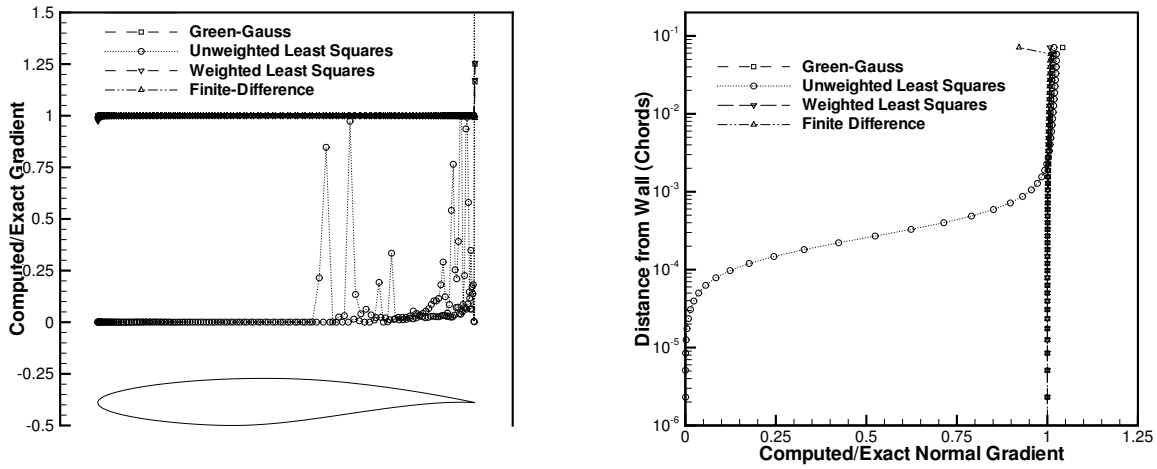
Table 2. *Variations of computed lift and drag values at Mach=0.75, Reynolds = 3 million and 0^o incidence for DLR-F4 test case as a function of variations in artificial dissipation parameters*

Discretization	C_L	C_D
Fine Mesh (13M pts)	0.5459	0.03011
Baseline Mesh : $\kappa_2 = 1.0, \delta = 0.1$	0.5307	0.03051
Baseline Mesh : $\kappa_2 = 0.5, \delta = 0.1$	0.5323	0.02990
Baseline Mesh : $\kappa_2 = 1.0, \delta = 0.1$	0.5308	0.03054
Baseline Mesh : $\kappa_2 = 1.0, \delta = 0.2$	0.5307	0.03054
Baseline Mesh : $\kappa_2 = 1.0, \delta = 1.0$	0.5416	0.03302

In Figure 8 and Table 3, several variations of the upwind discretization scheme are compared with the matrix dissipation scheme discussed previously. The baseline case for the upwind discretization scheme involves gradient reconstruction using the unweighted least-squares procedure,[42,51] a vanishing entropy fix parameter $\delta = 0.0$, and

no limiting. The lift values produced by this discretization scheme are slightly lower than those computed with the artificial dissipation discretization, and since the lift values increase with lower dissipation levels and finer grid resolution, it may be inferred that the least-squares gradient-based upwind discretization is slightly more diffusive than the matrix dissipation scheme. Because the nominal value of the κ_2 coefficient in the matrix dissipation scheme has been determined empirically, it is conceivable that a simple rescaling of the dissipation terms could be used to improve the accuracy in the upwind scheme as well, although at the risk of reduced robustness. On the other hand, the drag values produced by this scheme are slightly lower than those obtained with the matrix dissipation scheme. From Table 3, the drag value at 0 degrees incidence is seen to be 8 counts lower than that computed using the matrix dissipation scheme on the same grid. Therefore, there are more substantial differences between these two schemes which extend beyond the simple scaling of the final dissipative terms.

Returning to the baseline unlimited case, the entropy fix parameter is now increased from 0.0 to $\delta = 0.1$, which is the level used in the baseline matrix dissipation settings. In this case, the computed lift values are substantially lower than those obtained with a vanishing entropy fix, and the drag values are substantially higher. From Table 3, the drag at 0^o incidence is seen to increase by 17 counts, with much larger variations in lift. In essence, the accuracy of the scheme has been completely compromised by this small value of the entropy fix, which had little effect on the accuracy of the matrix dissipation scheme. This behavior is attributed to a poor estimate of the gradients in the boundary layer region using the least-squares procedure and has been extensively investigated in reference.[45,46] This is illustrated in Figure 5, where a linear test function is prescribed over a typical stretched-unstructured mesh used for viscous flow calculations about an RAE 2822 airfoil, and the gradient of this function is computed using various reconstruction techniques. As can be seen, the unweighted least-squares approach severely under-predicts the gradient values in the near wall regions. This behavior has been attributed to the effect of surface curvature which results dominant contributions to the normal gradient from misaligned streamwise stencil points.[45] By under-estimating the gradients in the boundary layer region, the least-squares approach effectively increases the dissipation and reduces the accuracy of the discretization in these regions. This is evident in the solutions produced by the $\delta = 0.1$ case, which resemble first-order accurate solutions.



(a) (b)

Figure 5. Computed gradient of test function (normalized by exact value) for typical viscous-type grid about RAE 2822 airfoil illustrating poor accuracy of unweighted least-squares approach in the presence of surface curvature. Reproduced from[45]

However, this argument does not explain how good accuracy was achieved in the baseline upwind discretization case when using a vanishing entropy fix. The answer lies in the alignment of the grid with the flow direction in the boundary layer region.[45] Because the highly stretched thin mesh elements in the boundary layer region are closely aligned with the flow direction, the normal flow velocity for the control volume interfaces aligned with the wall direction essentially vanishes. Since this normal velocity represents one of the eigenvalues of the dissipation matrix (c.f. equation (1)), it is seen that the additional dissipative effect due to the poor gradient estimate is counterbalanced by a vanishing eigenvalue precisely at the same location. Thus, any limit on the minimum size of this eigenvalue, through the use of an entropy fix, triggers the excessive dissipative effect of the poor gradient estimate.[45] Therefore, although the unweighted least-squares gradient is used extensively by production codes with success, the poor accuracy of this gradient estimate in the boundary layer region has the potential to corrupt the solution in unforeseen manners and must be considered carefully, particularly for new untested applications.

Returning to the baseline upwind discretization with vanishing entropy fix, the application of limiters is invoked in order to examine the effect on solution accuracy. The multi-dimensional monotonicity preserving limiter due to Barth and Jespersen[6] is employed. A small increase in lift and a moderate increase in drag are observed in Figures 8(c) and (d) , respectively. At 0^o incidence, the drag is increased by 8 counts, as shown in Table 3. For flows with strong shocks, the use of limiters may be required to guarantee stability. Unfortunately, as the current example illustrates, the use of limiters has an adverse effect on overall solution accuracy for production type grids of this resolution. Therefore, for transonic flows where shock strengths are not severe, the use of limiters is generally avoided when possible. However, it should also be noted that alternate limiter formulations are available, such as the non-strictly monotone limiter of Venkatakrishnan,[47] which result in less accuracy degradation, and the current results should be regarded as establishing an outer bounds to limiter sensitivity.

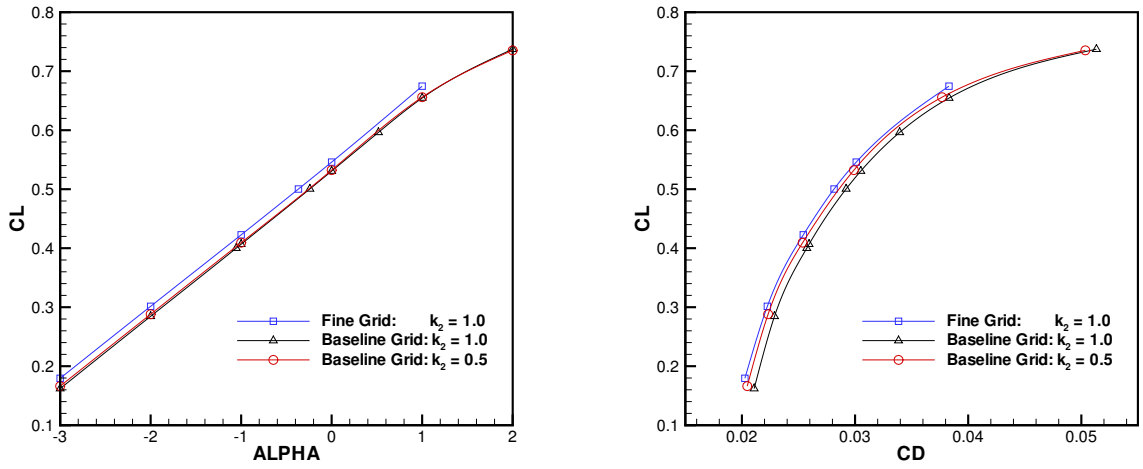
Table 3. *Variations of computed lift and drag values at Mach=0.75, Reynolds = 3 million and 0^o incidence for DLR-F4 test case as a function of variations in least-squares gradient-based upwind discretization.*

Discretization	C_L	C_D
Fine Mesh (13M pts)	0.5459	0.03011
Baseline Mesh : Artificial Dissipation	0.5307	0.03051
Baseline Mesh : Least Squares: Limiter OFF: $\delta = 0.0$	0.5161	0.02970
Baseline Mesh : Least Squares: Limiter OFF: $\delta = 0.1$	0.3995	0.02797
Baseline Mesh : Least Squares: Limiter ON: $\delta = 0.0$	0.5235	0.03054

C. Viscous Discretization

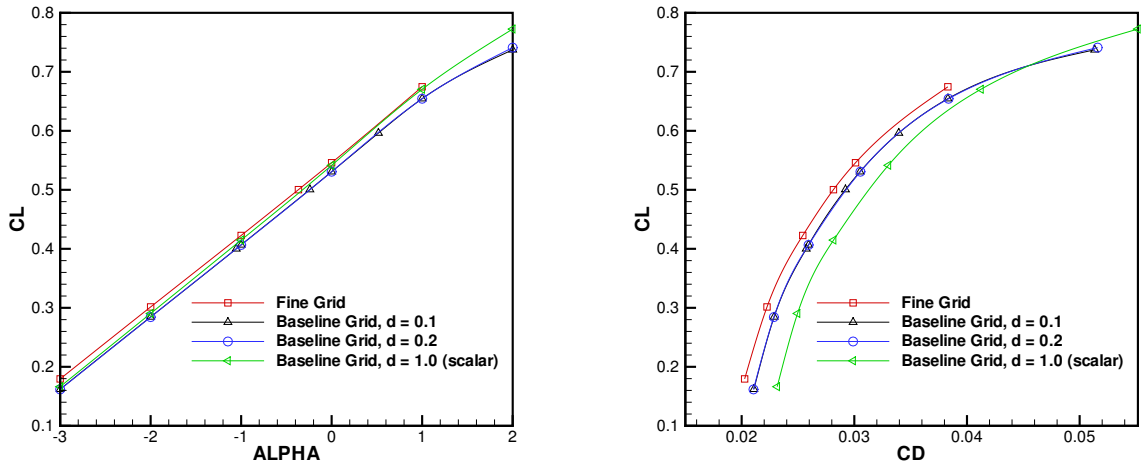
There are two issues related to the accuracy of viscous term discretizations on unstructured meshes. The first issue is a consequence of the use of highly stretched mesh cells in the vicinity of boundary layer and wake regions, for the high-Reynolds number applications typical of aerodynamic calculations. The second relates to the fact that the viscous terms involve second-derivative terms, as opposed to the convective terms which are only first derivatives.

At present, the use of highly anisotropic meshes in viscous layer regions represents the only plausible approach for adequately capturing the large disparities in normal and streamwise gradients which exist in the high-Reynolds number flows which are of principal interest for aerodynamics. The simple application of an anisotropic mapping to an existing triangular or tetrahedral mesh results in the creation of highly obtuse elements with large angles. It is well known from two-dimensional finite-element theory that the discretization accuracy degrades rapidly with increasing angles for triangular meshes, although the accuracy is not inadvertently affected by the presence of small angles.[52]



(a) (b)

Figure 6. Comparison of computed lift and drag coefficients on DLR-F4 configuration at transonic conditions for different values of the artificial dissipation scaling coefficient.



(a) (b)

Figure 7. Comparison of computed lift and drag coefficients on DLR-F4 configuration at transonic conditions for different values of the entropy fix for matrix artificial dissipation formulation.

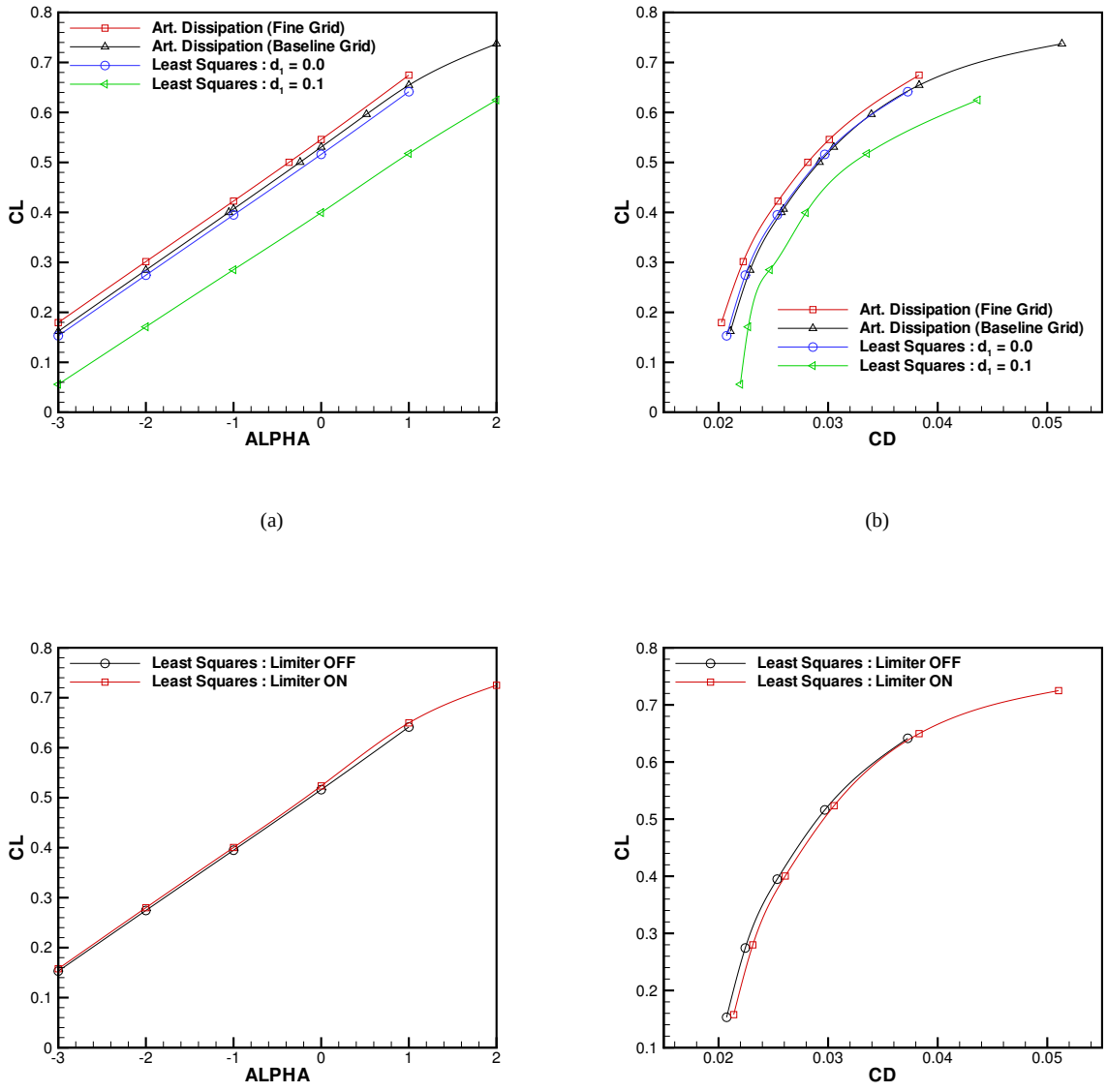


Figure 8. Comparison of computed lift and drag coefficients on DLR-F4 configuration at transonic conditions using the upwind reconstruction approach for various settings of the entropy fix and limiters.

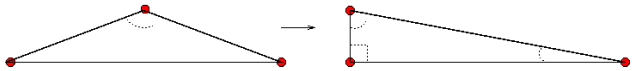


Figure 9. Illustration of two types of anisotropic triangular cells with first configuration containing one large obtuse angle, and second type containing one small angle, and two angles close to 90 degrees.

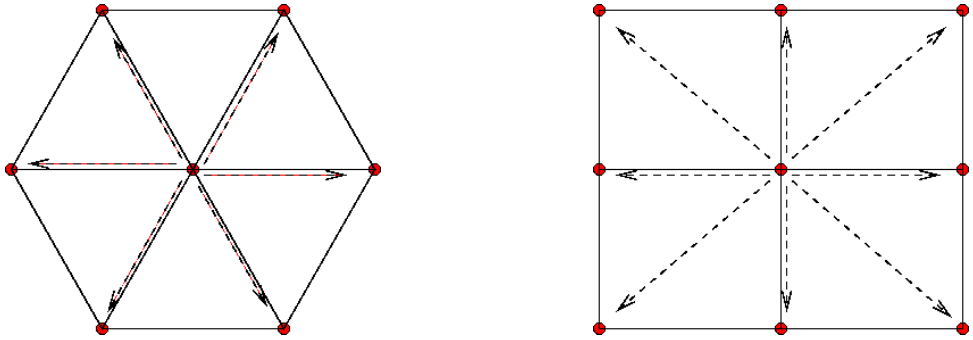
(a) (b)

Figure 10. Illustration of (a) median-dual control-volume on anisotropic stretched mesh configuration and (b) containment-dual control-volume (approximate configuration for illustration purposes) on which de-emphasizes diagonal edge contribution.

The construction of a highly stretched triangular mesh with only small angles and no obtuse angles leads to the generation of a series of nearly right angle triangles, with the small triangle edge normal to the flow feature or direction of stretching, as shown in Figure 9. Although a highly stretched mesh topology with right angle triangles (or tetrahedra in 3D) provides much better accuracy for such flows, the diagonal edge of such cells remains non-aligned with the flow features, and can lead to inaccuracies due to the reasons discussed previously. The use of alternate element types such as quadrilateral elements in two dimensions, or prismatic elements in three dimensions, provides one approach for eliminating the evaluation of fluxes non-aligned with these features. The drawback of this approach is that it requires modification of the grid generation process, as opposed to the solver discretization. Note that alternate control-volume formulations can be used to overcome the deficiencies associated with non-aligned grid faces, such as the use of containment-dual control-volumes[31,53] as shown in Figure 10, which de-emphasize the contributions of such edges. However, if the contributions of such edges are error-prone, and these can be identified either through containment dual constructions, or in the grid generation phase itself, then the outright omission of these edges provides equivalent accuracy at reduced cost,[54] and potentially superior robustness as well. This has been the strategy behind the development of hybrid prismatic/tetrahedral mesh generation schemes. In fact, it may be argued that the simple determination of a stretching direction is equivalent to the identification of a structure in the underlying grid to be generated, and that this can then be used to construct semi-structured meshes using prismatic elements in these regions.[55,56]

Once a suitable mesh has been constructed, the second-derivative viscous terms must be discretized on this topology. While for cell-centered discretizations there is no obvious simple strategy, for vertex-based discretizations, the standard continuous Galerkin finite-element approach using linear basis functions results in a compact stencil discretization for second derivatives for simplicial elements such as triangles in two dimensions and tetrahedra in three dimensions. In spite of the fact that the finite-element discretization has been traditionally assembled on an element basis, the formulation can be recast as a loop over edges, enabling the use of the standard edge-based data-structure already used for the inviscid terms.[57,31] However, for non-simplicial elements, this discretization results in a stencil which contains diagonally located vertices in adjacent elements, as shown in Figure 11, which are not connected to the considered vertex by a mesh edge. This obviates the possibility of using the edge data-structure in these cases, and complicates the construction of the exact Jacobian as well.

For these reasons, an alternate construction has often been advocated, whereby a two-pass procedure is used. In the first pass, gradients at each control-volume are constructed by one of the methods described in the previous section. In the second pass, the required full Navier-Stokes terms, which include mixed second-order derivatives are constructed by integrating the required gradients around the boundary of the control volume, using a face-based gradient defined



(a) (b)

Figure 11. Illustration of stencil for simplicial (triangular) elements and non-simplicial elements in two dimensions, showing stencil dependence on vertices not connected by a mesh edge to considered vertex in latter case.

as:

$$\overline{(\nabla u)}_{ij} = \frac{1}{2}(\nabla u)_i + (\nabla u)_j \tag{12}$$

In other words, the gradient at the control volume face ij is taken as the average of the two pre-computed gradients withing each neighboring control volume i and j. The advantage of this approach is that it enables the construction of these terms using the same edge data-structure (and possibly gradient information) already used in the inviscid term construction. The disadvantage is that it leads to an extended stencil of neighbors of neighbors, and does not suppress odd-even decoupling modes. A simpler approach is to abandon the requirement of multi-dimensional reconstruction of second and mixed derivatives required by the full Navier-Stokes equations, and rely on the thin-layer version of these equations. A slightly more elaborate approach consists of constructing an approximate Laplacian of the velocity, by integrating the normal derivative around the control volume faces as:

$$\int_{\partial\Omega} \nabla u \cdot \mathbf{n} \; d(\partial\Omega) = \sum_{cv\,faces_{ik}} \frac{u_j - u_i}{|\vec{r}_j - \vec{r}_i|^2} \, (\vec{r}_j - \vec{r}_i) \cdot \vec{N} \tag{13}$$

where $\mathbf{n}$ represents the unit face normal, $\vec{N}$ the area weighted face normal, and $\vec{r}_j - \vec{r}_i$ is the vector joining centroids i and j. A consistent discretization of the Laplacian, multiplied by the fluid viscosity, corresponds to the viscous terms for the incompressible version of the Navier-Stokes equations, which is likely a reasonable assumption for most boundary and wake regions in transonic flows. However, the above approach only constitutes a consistent discretization of the Laplacian for cases where the control-volume face normal is aligned with the segment joining the two adjacent control-volume centroids. The full Navier-Stokes terms can be constructed using a hybrid approach, where the estimate of the full gradient at the control volume face given by equation (12) is replaced by:

$$\overline{(\nabla u)}_{ij} = \frac{1}{2}(\nabla u)_i + (\nabla u)_j - \frac{1}{2}\left[(\nabla u)_i + (\nabla u)_j\right] \cdot \frac{\vec{r}_j - \vec{r}_i}{|\vec{r}_j - \vec{r}_i|} + \frac{u_j - u_i}{|\vec{r}_j - \vec{r}_i|} \tag{14}$$

effectively using equation (13) for the contribution to the gradient in the direction of the edge joining control volume centroids i and j, and thus suppressing odd-even decoupling modes. However, this approach retains the extended stencil discussed previously, thus making the construction of an exact Jacobian for the viscous terms impractical, which can have a compromising effect on solver robustness.

Table 4 illustrates a series of calculations performed on the second AIAA Drag-Prediction-Workshop DLR-F6 configuration using the inconsistent Laplacian or multi-dimensional thin-layer approximation given by equation (13),

compared with the inclusion of the full Navier-Stokes viscous terms, using the discretization given by equation (14). For this test case, which includes non-trivial regions of separated flow, the differences in computed force and moment coefficients using the thin-layer and full Navier-Stokes terms is relatively small. These results provide evidence that for transonic flow applications with moderate amounts of separation, the multi-dimensional thin-layer approach represents a valid simplification, although more extensive investigations are still warranted.

Another approach for discretizing the viscous terms on unstructured meshes is afforded by the use of numerical flux functions derived from kinetic theory, such as the BGK scheme.[58] The advantage of the BGK flux function is that it results in the flux contributions from inviscid and viscous effects in a single flux evaluation using only nearest neighbor information. This approach has been demonstrated on unstructured meshes by May and Jameson,[59] and may provide an alternative to the techniques described above.

Table 4. Computed Lift and Drag Coefficients at Mach=0.75, Incidence=0^0, on various grids using the multi-dimensional thin-layer discretization and the extended stencil full Navier-Stokes discretization.

Grid Size	Viscous Terms	C_L	C_D	$CD_{Pressure}$	$CD_{Friction}$
1M pts	Thin Layer Approx.	0.5055	0.02960	0.01661	0.01298
1M pts	Full Navier Stokes	0.4960	0.02921	0.01652	0.01269
3M pts	Thin Layer Approx.	0.5012	0.02859	0.01568	0.01290
3M pts	Full Navier Stokes	0.5020	0.02841	0.01571	0.01270
9M pts	Thin Layer Approx.	0.5141	0.02882	0.01592	0.01290
9M pts	Full Navier Stokes	0.5154	0.02867	0.01594	0.01273

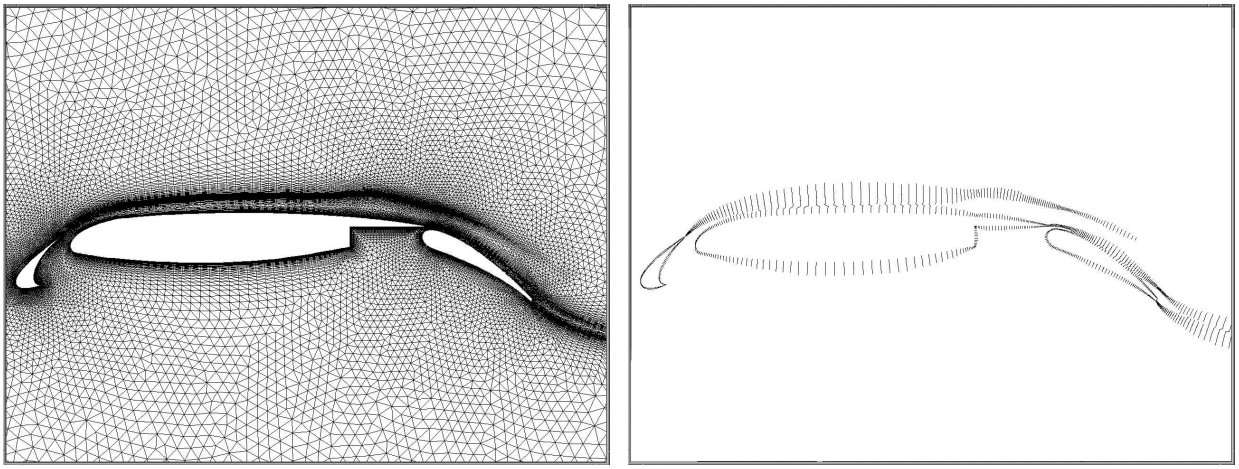
III. Solvers

There have been rapid advances in solver technology for unstructured mesh discretizations over the last two decades. Initially, unstructured mesh solvers were confined to simple explicit solvers[4,6,60,15] since the more advanced solution techniques of the day such as alternating direction implicit (ADI), approximate factorization (AF), or multigrid, which were all based on exploiting the underlying structure of the mesh, were believed to be inapplicable to unstructured meshes. Additionally, the solver technology developed for finite-element structural analysis problems was mostly aimed at direct inversion of sparse matrices, which was not the most appropriate strategy for the large non-linear problems encountered in computational aerodynamics.

However, many of the solution techniques developed for structured meshes were eventually modified, generalized, and often improved for use with unstructured meshes or arbitrary meshes in general.[61,62,63,64,8,9,10,11] For example, the original application of multigrid methods to unstructured meshes was devised by closely following the structured grid approach, whereby a set of coarser physically consistent meshes are constructed, and the solution and residual values are interpolated back and forth between the various meshes of the multigrid sequence. The principal development for enabling the original unstructured multigrid strategy was the formulation of efficient search and interpolation routines for non-nested coarse and fine unstructured meshes.[7,8,9,65,66] However, it was soon realized that algebraic multigrid strategies offered the potential for more robust and general solvers for unstructured mesh problems. Because algebraic multigrid solvers at the time were not competitive with geometric multigrid solvers, the hybrid approach of agglomeration multigrid was developed, which offered the efficiency of geometric multigrid methods combined with the automation of algebraic multigrid methods, for use on unstructured meshes.[67,68,69,70,71,72] Today, agglomeration and algebraic multigrid solver technology is used extensively in various unstructured mesh simulation codes, delivering equivalent or better convergence efficiency than existing structured mesh multigrid methods. In many cases, the ability to construct very coarse mesh levels in the presence of complex geometries and without the requirement of evenly divisible grid coordinate indices provides for a more flexible and efficient unstructured mesh solution mechanism compared with structured multigrid methods.

Other techniques such as implicit line solution methods have also been applied to unstructured mesh methods with

great success.[73,74,75,76,77] Although grid lines are naturally existing structures within a structured, block structured, or overset mesh system, they can be easily created in an unstructured mesh by grouping together subsets of contiguous mesh edges or faces using a graph algorithm or other identification means, as shown in the two-dimensional example depicted in Figure 12. Once these structures have been identified, an implicit line solver or preconditioner can be implemented as a block tridiagonal matrix inversion algorithm in a similar manner to that used for structured meshes, simply using an additional level of indirection. As in the multigrid case, the use of line solvers on unstructured meshes offers the added flexibility of tailoring the line structures to the dominant stiffness components in the computational domain, without the restriction of having to conform to the original mesh structure. Line solvers have proven particularly effective for relieving the stiffness associated with highly anisotropic meshes in boundary layer and wake regions, and have also been used effectively in conjunction with multigrid methods.



(a) Initial unstructured mesh in two dimensions about three-element airfoil configuration.

(b) Subset of mesh edges extracted to form set on non-intersection lines.

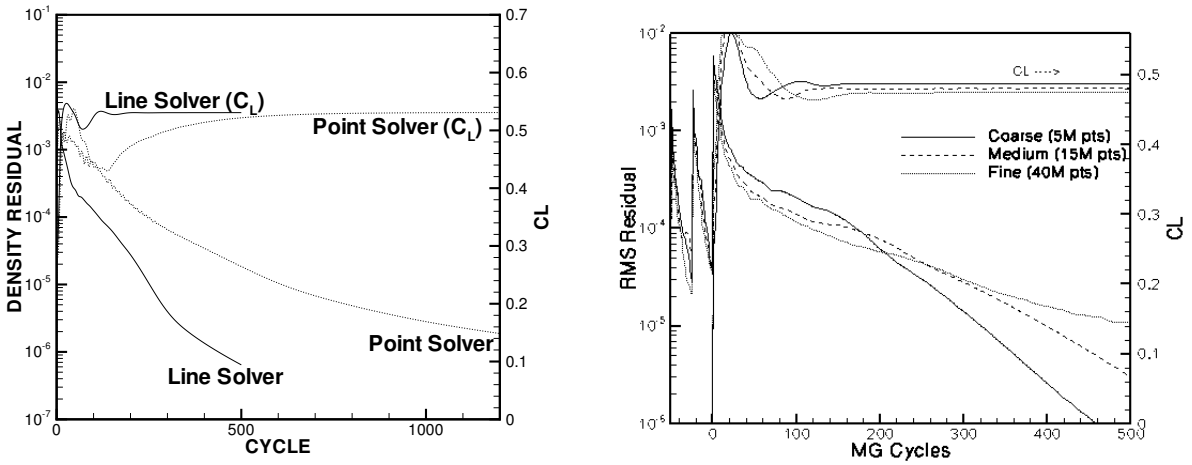
Figure 12. Two dimensional illustration of extraction of set of lines from unstructured mesh for implicit line solver.

Figure 13 provides an example of the effectiveness of line solvers and multigrid for unstructured meshes. The test case is taken from the AIAA drag-prediction-workshop series, and consists of a wing-body configuration at the transonic Mach number of 0.75, 0 degrees incidence, and a Reynolds number of 3 million. In Figure 13(a), using a mesh of 3 million points with 1.e-06 chords spacing near the wall, the line solver driven multigrid algorithm is seen to be more than twice as fast as the point solver multigrid algorithm, reaching steady state values for the force coefficients in well under 500 multigrid cycles. In Figure 13(b), the convergence of the line-implicit multigrid algorithm is shown for various meshes on a similar configuration, illustrating the relative insensitivity of the convergence rate to the degree of mesh resolution, which varies by an order of magnitude in this example.

The success of solution strategies for unstructured mesh problems has been such that most current developments in solver technology have been designed for use with unstructured data-sets (including of course structured data-sets as a subset of these), as exemplified by recent developments in algebraic multigrid methods, graph algorithms, preconditioners, and Krylov methods (see for example reference[78]).

IV. Efficient Hardware Implementation

One of the initial perceived drawbacks of unstructured mesh discretizations is that they require the explicit storage of the mesh connectivity, as well the use of indirect memory references, which can lead to slower computational



(a) (b)

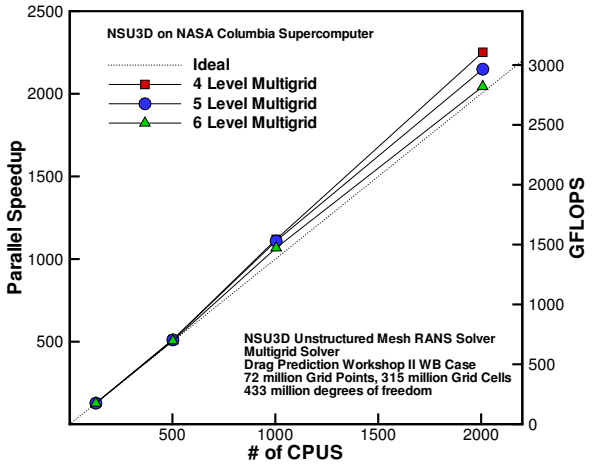
Figure 13. Illustration of (a) effectiveness of line-implicit solver versus point implicit solver on highly anisotropic meshes used as a driver for multigrid and (b) weak dependence of convergence on grid resolution for DPW3 wing-body-fairing configuration at Mach = 0.75, 0 degrees incidence, and Reynolds number of 5 million.

rates and larger memory requirements compared to structured grid methods. Although these were non-trivial issues in the early development of unstructured mesh methods, when vector computers with relatively limited expensive core memory were the norm, many of these initial concerns have been addressed through the development and use of more effective data-structures, data-reordering techniques, and advances in computational hardware. The development of the edge or face-based data-structure has enabled reduced memory requirements and higher computational rates, while the shift to commodity memory and cache-based processors has largely alleviated the memory footprint bottleneck of early unstructured mesh implementations. Furthermore, while structured grid discretizations result naturally in a banded matrix with inherent memory locality, re-ordering techniques for unstructured mesh data-sets for cache locality have been successful in increasing the computational rates achieved by unstructured mesh solvers.

Furthermore, one of the significant advantages of unstructured mesh solvers has been the scalability displayed by these methods on massively parallel computer architectures. Compared to block-structured or overset mesh methods, unstructured mesh methods can be formulated using a single homogeneous data-structure (i.e. edge or face data-structure) which covers the entire computational domain, enabling near optimal partitioning and load-balancing operations. An example is depicted in Figure 14, where the NSU3D unstructured multigrid solver is benchmarked on the NASA Columbia machine showing good scalability up to 2008 cpus for a mesh of 72 million vertices.[79] When partitioned on 2008 cpus, the average partition size of this mesh contained only 36,000 vertices, while the average partition size for the coarsest mesh of the multigrid sequence used in this solution contains well under 100 control-volumes. Nevertheless, the homogeneous nature of the unstructured mesh data, coupled with the use of efficient graph-based partitioners,[80] enables near perfect load balancing on this large number of processors.

V. Importance of Grid Resolution

A consistent outcome of various studies of accuracy in computational aerodynamics is that mesh resolution remains one of the most important factors in achieving an accurate and reliable simulation outcome. The problem is rooted

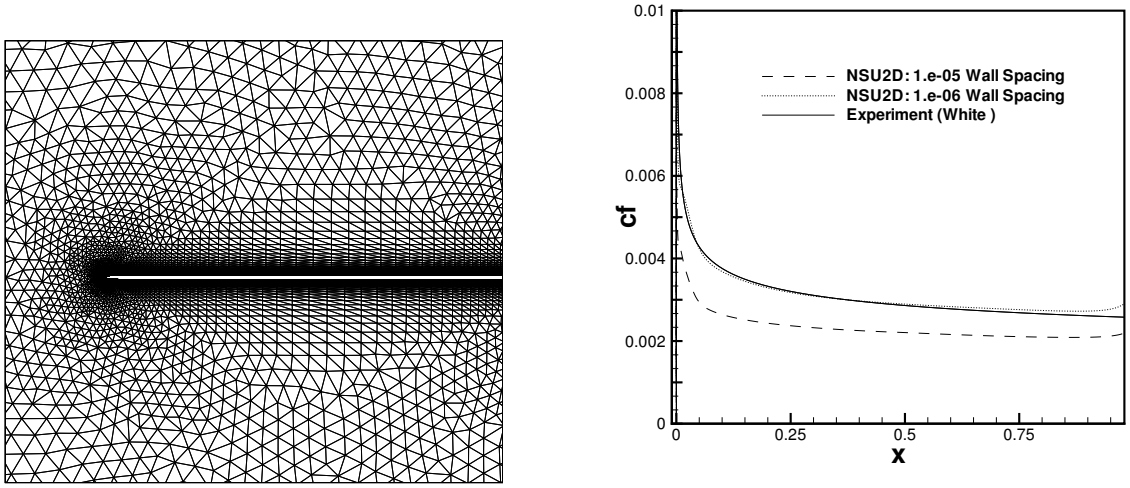


(a)

Figure 14. Scalability of NSU3D multigrid solver on NASA Columbia supercomputer for transonic wing-body configuration using 72 million grid points using up to 2008 cpus.

in the fact that fully grid-converged RANS solutions are almost never possible in a production environment, and thus changes in grid resolution will produce non-negligible changes in the solution. Sufficient grid resolution is required to enable the capturing of all relevant flow physics (in particular all regions of separated flows), and to ensure that the remaining discretization error is either small enough, or inconsequential. For example, when studying incremental drag differences between two configurations, non-negligible discretization errors may be manageable provided they are of the same magnitude on both configurations, and therefore cancel out in the study of incremental effects. The issue is also particularly complex due to the large range of scales present in aerodynamic flows, thus requiring vastly different resolutions in different regions of the domain. For example, in the 1.65 million point vertex-based mesh described in Table 1, the ratio of smallest to largest element volume is close to 10^9, and the largest cell-aspect ratio is of the order of 10000. Precise control over the spacings in the mesh boundary layer regions is also required in order to ensure accurate viscous flow calculations. For example, most turbulence models require that the first grid point off the wall be inside the laminar subregion, usually within the range $y^+ = 1$. Using a flat plat turbulent boundary layer estimate, this distance can easily be calculated as a function of Reynolds number. For Reynolds numbers of the order of 10 million, this requires a wall spacing of approximately 1.e-06 chords, which results in grid aspect ratios of the order of 10000 in these regions. As the mesh traverses the boundary layer, the mesh spacing is allowed to grow in the direction normal to the wall, and the increase in spacing between two adjacent cells is generally accepted to be no more than 1.2 This results in 20 to 30 mesh points spanning the direction normal to the boundary layer. The importance of boundary layer grid resolution on engineering aerodynamic quantities is illustrated by the simple case of flow over a flat plate in two dimensions. The NSU2D unstructured RANS solver[81] has been used to compute this flow on the grid shown in Figure 15(a), for a Mach number of 0.2 and a plate-based Reynolds number of 10 million, using the Spalart-Allmaras turbulence model.[82] This mesh contains a normal spacing at the plate surface of 1.e-05 plate lengths. Figure 15(b) depicts the skin-friction computed on this mesh as compared with experimental data taken from,[83] showing substantial under-prediction of the skin-friction coefficient. Recomputing this flow on a mesh with a normal resolution of 1.e-06 plate lengths (with otherwise identical resolution), results in substantially improved skin-friction correlation, as shown in the figure.

While improper boundary layer resolution can hinder accurate skin-friction values directly, even larger discrepancies can arise for cases involving more complicated flow physics, such as high-lift flows. Figure 16, reproduced from,[84] compares the computed versus experimental surface pressures for a three-element high-lift configuration, noting poor agreement over the trailing-edge flap upper surface, where an over-prediction of the upper surface flow separation is apparent. In Figure 16, the same case has been recomputed using a nearly identical mesh, with the exception that the wall spacing of the mesh was reduced by one order of magnitude, while the growth rate has been kept the same as previously. The agreement with experiment is now much improved, with a smaller separated region occurring on the flap. The fact that the separated region over-prediction is simply the result of inadequate grid wall spacing is not necessarily evident, and might otherwise have been attributed to turbulence modeling inadequacies.



(a) Two-dimensional unstructured grid for flat plate test case.

(b) Computed skin friction for flat plate test case.

Figure 15. Illustration of the importance of normal grid resolution in boundary-layer regions for accurate skin-friction predictions at high Reynolds numbers.

Figure 17 illustrates typical unstructured mesh topologies used in transonic aerodynamic calculations. These meshes have been generated using the VGRID[37] grid generation code, and were made available as baseline grids for the second drag prediction workshop (DPW2) in 2003.[24] Only the coarsest grids are shown, illustrating the boundary layer resolution and detail in areas such as the nacelle for the more complicated geometry. These meshes approximately follow the resolution characteristics given in Table 1 for the vertex-based discretization mesh, and finer meshes using the same topology and relative resolution variations throughout the domain were constructed using up to 72 million vertices, in order to conduct a grid convergence study. The idea of a grid convergence study is to examine the variation of the computed objectives of the simulation, in this case force coefficients, to see if these display second-order accuracy, and converge to a constant value in the limit of infinite grid resolution. For a second-order accurate scheme, plotting the computed force coefficients as a function of the inverse of the square of the grid spacing, should produce a straight line curve. For a sequence of grids with similar relative spatially-varying grid resolutions, a common approach is to plot the computed force coefficients versus the number N of grid points to the -2/3 power,[85,86] where $N^{1/3}$ corresponds to some average measure of the grid spacing in three dimensions.

A grid convergence study was performed in reference[87] on this configuration, and the principal results of this study are reproduced herein. The sensitivity of the solution to the levels of artificial dissipation was also included as part of the grid refinement study. Dissipation errors are often thought to be the dominant errors on coarser grids,

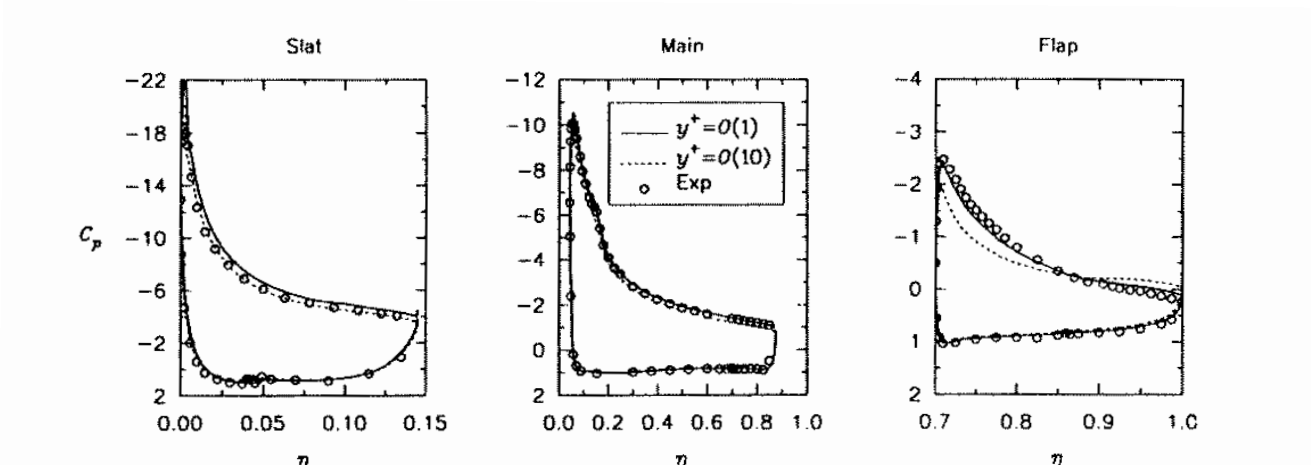


Figure 16. Effect of normal grid resolution on high-lift three-element airfoil configuration. Mach = 0.2, Incidence = 22 degrees, Reynolds number = 9 million. Reproduced from[84]

and increasing the mesh resolution naturally leads to lower dissipation levels. However, the sensitivity of the solution to the prescribed levels of dissipation on any grid level can provide an indication of the level of dissipation errors on that grid level, and the sensitivity to dissipation levels can be expected to decrease for increasingly finer grids. The sensitivity to dissipation levels can also be used to estimate the magnitude of the effects of switching between an artificial dissipation scheme and an upwind scheme, or between various approximate Riemann solvers within an upwind scheme. The sensitivity to dissipation levels is obtained by comparing the results obtained using the nominal values of the dissipation scaling parameter, with the results obtained using half the nominal value. The nominal value has been determined empirically as a compromise between accuracy and robustness. The smaller value provides superior accuracy but may encounter convergence problems for finer grids at higher lift conditions.

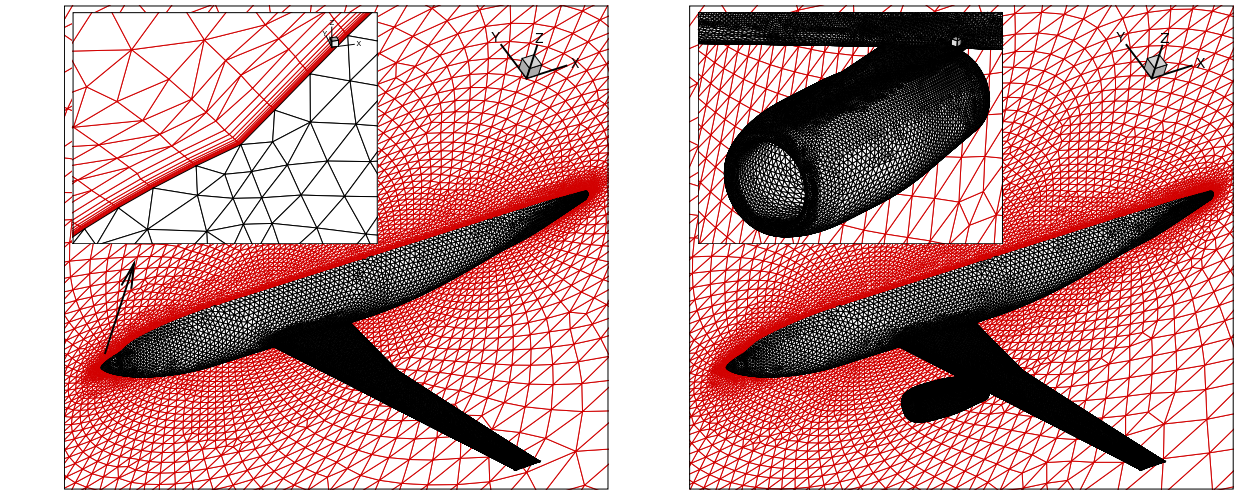
Two conditions were considered for this study, the standard transonic condition of Mach=0.75, and 0 degrees incidence, and a subcritical case of Mach=0.3 and 0 degrees incidence. The second condition is included to provide further validation of the grid convergence process for a simpler subsonic case which contains no discontinuities and (possibly) less separation, and thus may be expected to more readily display asymptotically second-order accurate grid convergence.

Figure 18 illustrates the computed lift and drag values as a function of the number of grid points to the -2/3 power, for the transonic and subsonic conditions, using nominal and reduced dissipation levels. The drag values in particular appear to be converging asymptotically towards an infinite resolution value with second-order accuracy, as evidenced by the straight line behavior of these plots. Furthermore, the *changes* in the computed drag coefficient due to the different dissipation levels decreases monotonically with increasing grid resolution, with very small differences remaining for the 72 million point grid. The computed lift coefficient displays a slightly more erratic behavior, including non-monotone behavior for both the transonic and subsonic cases, although the lower dissipation values are more monotone, and the sensitivity to dissipation is reduced as well on high resolution meshes. For this family of grids(1M, 3M, 9M, and 72M points), the solution thus appears to be converging towards an infinite resolution value.

As part of the study, an alternate fine mesh containing 65M vertices was generated[88] and made available. The topology of this grid was substantially different than the previous grids in the trailing-edge region of the wing, as depicted in Figure 19. While the previous grids all employed spanwise stretching of the elements along the wing in order to reduce the overall number of grid points, the new mesh was designed to produce essentially isotropic surface mesh faces in all regions of the domain, resulting in high spanwise resolution in the trailing-edge region. Surprisingly, when the same cases were computed on the grid of 65M points, the obtained solution values were substantially different than any of the values obtained on any of the other grids, as shown by the additional data points in Figure 18. Note that

it is strictly not valid to plot the results from all five grids on the same $N^{-\frac{2}{3}}$ graph, since the latter grid does not belong to the same family of grids. However, this is done in this case to illustrate the changes in the computed values between the two finest grids. The lift at zero incidence on this latter grid is roughly 10% lower than that obtained on the 72M point grid, representing a much larger variation than seen between any of the other grids, and much larger than any of the computed sensitivities due to dissipation or viscous term formulation (c.f. Table 4). It remains very surprising and unexpected that there could be such a large difference between two solutions computed on such fine grids (65M points vs. 72M points), especially in light of the apparent grid convergence path determined by the original family of grids. It is also worth noting that these trends (lower lift and friction drag) are reproduced for the subsonic M=0.3 case on these grids. Similar characteristics were observed when running these same meshes with the FUN3D unstructured mesh flow solver in reference[89] in the place of the NSU3D solver, verifying that these results are not simply due to anomalous flow solver behavior.

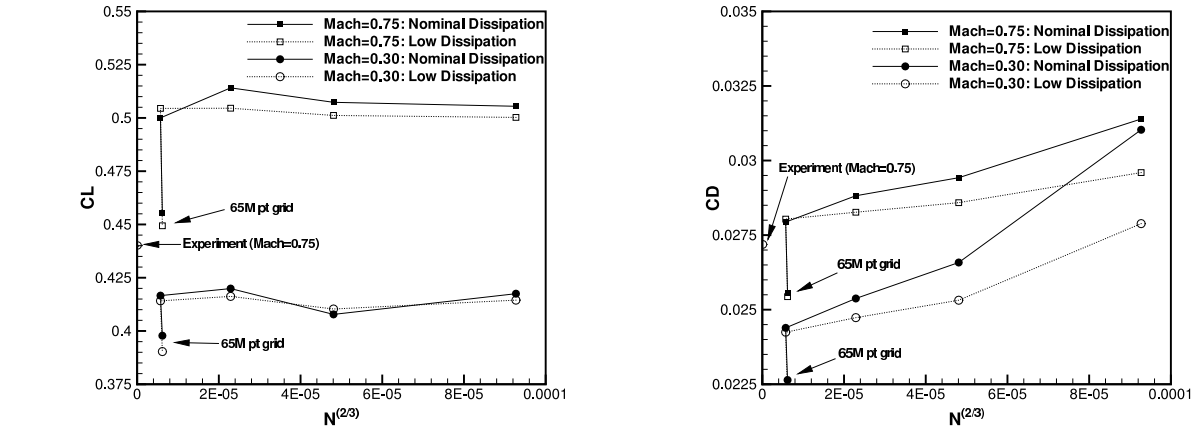
One may speculate that the use of anisotropic cells in the trailing-edge region, were the flow is known to be separated, results in the prediction of smaller trailing-edge separation regions, which produces substantially higher lift, when integrated along the span of the wing. Whether increased grid resolution in both cases would bring the solutions on all meshes together or whether some of these solutions are actually inconsistent with the continuous infinite resolution solution remains an open question, with important implications for the effect of element shape and stretching characteristics in critical regions of the flowfield on overall solution accuracy. However, a reliable determination of the causes of these effects can only be obtained through further study. The current study illustrates the potential for obtaining substantially different solutions on grids of different topologies, even at very fine resolutions, and even in the presence of encouraging grid convergence behavior. This observation, combined with the relatively small sensitivities to other modeling errors such as dissipation levels and viscous term discretization reinforces the notion that discretization errors are often still the dominant errors in many aerodynamic simulations.



(a) (b)

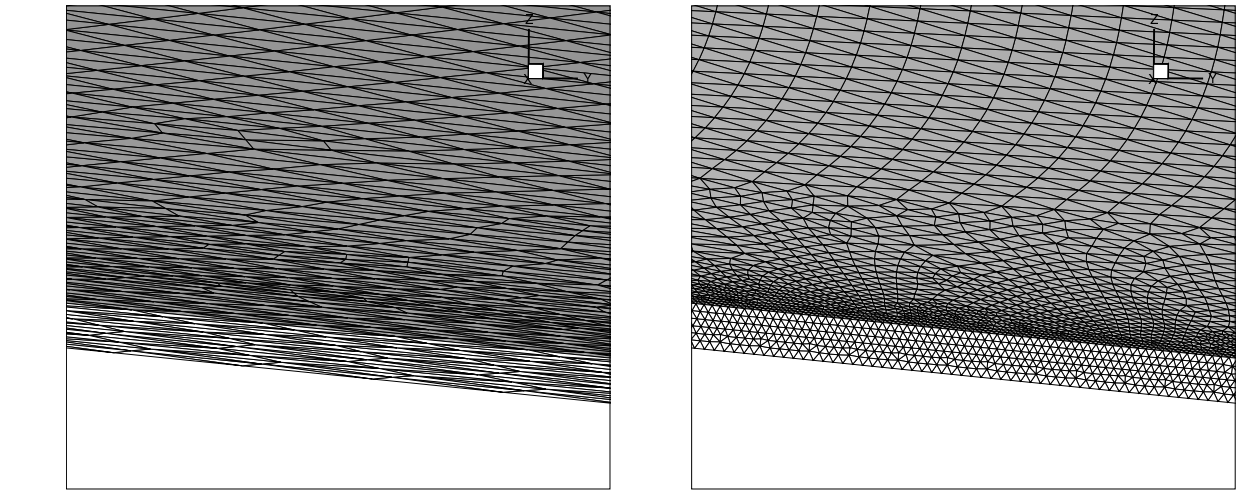
Figure 17. Illustration of unstructured meshes used for Drag Prediction Workshop configurations

In spite of these grid convergence anomalies, this particular case which has been used to highlight remaining accuracy concerns, represents a particularly sensitive condition which contains relatively important regions of separated flow, and which has been problematic for structured as well as unstructured mesh solvers. In addition to the three drag-prediction workshop results,[23,24,25] extensive additional grid convergence studies have been performed using a subset of the solvers participating in the workshops.[85,86,87] When the results of these studies are broken down by grid



(a) (b)

Figure 18. Comparison of computed (a) lift coefficients and (b) drag coefficients versus the number of grid points to the -2/3 power for transonic (M=0.75) and subsonic (M=0.3) cases at 0^o incidence including results computed on the 65 million point grid.



(a) (b)

Figure 19. Comparison between surface resolution for 72 million point grid (a) and 65 million point grid (b) in trailing-edge region in the vicinity of 60% span region illustrating spanwise stretching for the 72 million point grid.

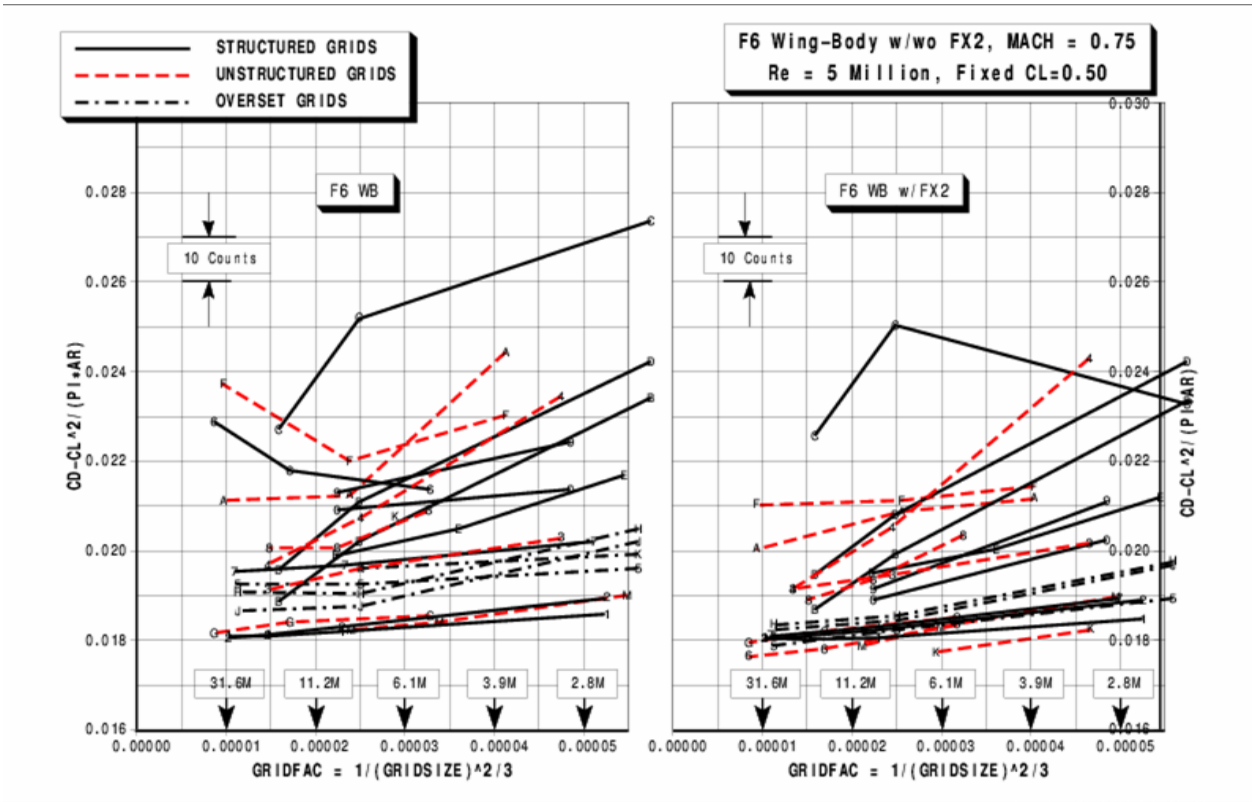


Figure 20. Collective grid refinement study results for third drag prediction workshop broken down by grid type. Reproduced from.[25]

type, as for example in Figure 20, where the collective results of the third Drag Prediction Workshop are shown, no particular advantage is apparent for one approach over the others. While the amount of scatter in the collective results may seem discouraging, it is the wide variety of participating entries with different discretizations, turbulence models, and grid topologies which is major contributor to this scatter. More specifically, there exists a subset of the participating codes, the results of which are grouped near the bottom of these charts, which have been validated extensively for transonic aerodynamics, which cover all three discretization approaches (structured, overset, unstructured), and which display remarkable agreement over the full range of conditions and grid resolutions.[25] This provides good evidence that unstructured and structured mesh solvers are at an equivalent level of sophistication for these types of problems.

However, for other problems of interest, such as high-speed flow, important accuracy concerns remain which need to be addressed.[90] Perhaps it has been the early attention paid to transonic flows which has enabled unstructured mesh methods to perform so well in this area, and similar long-term efforts need to be devoted to resolving the difficult issues in other applications.

VI. Conclusions

In this paper, we have tried to outlined the advantages and drawbacks of unstructured mesh methods compared to the most prevalent competing approaches of block-structured and overset mesh methods. For transonic flow simulations, it is claimed that unstructured mesh methods can be equal or even superior to most block-structured and overset mesh methods in terms of delivered accuracy at fixed computational cost. On the one hand, equivalent accuracy on equivalent unstructured and structured grids can be expected for subsonic and transonic flows, and solver technology for unstructured mesh methods is in no way inferior to that used for structured mesh methods. Furthermore, unstructured mesh approaches generally scale better on massively parallel computer hardware than their structured counterparts.

However, various unresolved accuracy and robustness concerns remain, and these tend to be more important for high-speed flows. Many of these issues, such as the problems due to non-aligned grid faces for shock waves or boundary layer regions, can be attacked either through the development of more sophisticated discretizations, through the use of novel mesh modification techniques, the use of alternate element types, or through the development of advanced mesh generation strategies. The interplay between discretization and mesh topology means that all these approaches deserve consideration.

Although not mentioned extensively in this paper, the drive to higher-order accurate methods can be expected to play an important role in the future for achieving higher fidelity simulations. Discontinuous Galerkin discretizations, or other approaches such as spectral volume methods provide a natural path for extending existing unstructured mesh discretizations to higher order.

Finally, it is worth noting that unstructured meshes may not always provide the best approach for dealing with complex geometries, and/or adaptive meshing requirements. The success of cut-cell Cartesian methods for inviscid flows about complex configurations[91,92,93,94] provides an example of alternate technologies which achieve both of these objectives with at least equal accuracy and reliability. These methods trade the interior mesh complexity of a body-fitted unstructured mesh for the complexity of correctly handling the cell intersections at the geometry boundary. For purely isotropic problems, the cut-cell Cartesian approach may be arguably a more effective strategy, since the problem of intersecting the geometry boundary is a lower dimensional problem which has shown greater promise for robust automation, compared with the slow progress in robust high-quality unstructured volume mesh generation. However, for anisotropic problems such as high-Reynolds number problems or even anisotropic geometries such as large aspect ratio wing planforms, the body-fitted approach (either structured or unstructured) retains the distinct advantage of being able to specify different resolutions in the respective characteristic directions, which can result in greatly increased in computational efficiency.

References

[1]O. C. Zienkiewicz and R. L. Taylor. *The Finite-Element Method for Solid and Structural Mechanics; 6th Edition*. John Wiley and Sons, New York, NY, 2005.

[2]M. O. Bristeau, O. Pironneau, R. Glowinski, J. Periaux, P. Perrier, and G. Poirier. On the numerical solution of non-linear problems in fluid dynamics by least squares and finite-element methods (II), applications to transonic flow simulations. *Comput. Methods Appl. Mech. Engrg*, 51:363–394, 1985.

[3]A. Jameson and D. J. Mavriplis. Finite volume solution of the two-dimensional Euler equations on a regular triangular mesh. *AIAA J.*, 24(4):611–618, 1986.

[4]A. Jameson, T. J. Baker, and N. P. Weatherill. Calculation of inviscid transonic flow over a complete aircraft. AIAA Paper 86-0103, January 1986.

[5]J. A. Desideri and A. Dervieux. Compressible flow solvers using unstructured grids. *VKI Lecture Series*, 1988-05:1–115, 1988.

[6]T. J. Barth and D. C. Jespersen. The design and application of upwind schemes on unstructured meshes. AIAA Paper 89-0366, January 1989.

[7]D. J. Mavriplis. *Solution of the Two-Dimensional Euler Equations on Unstructured Triangular Meshes*. PhD thesis, Princeton University, MAE Department, 1987.

[8]D. J. Mavriplis. Three-dimensional multigrid for the Euler equations. *AIAA Journal*, 30(7):1753–1761, July 1992.

[9]J. Peraire, J. Peirö, and K. Morgan. A 3D finite-element multigrid solver for the Euler equations. AIAA Paper 92-0449, January 1992.

[10]W. K. Anderson. A grid generation and flow solution method for the Euler equations on unstructured grids. *Journal of Computational Physics*, 110(1):23–38, January 1994.

[11]T. Gerhold, M. Galle, O. Friedrich, and J. Evans. Calculation of complex three-dimensional configurations employing the DLR-Tau code. AIAA Paper 97-0167, January 1997.

[12]D. J. Mavriplis. Accurate multigrid solution of the Euler equations on unstructured and adaptive meshes. *AIAA J.*, 28(2):213–221, 1990.

[13]D. J. Mavriplis. Adaptive mesh generation for viscous flows using Delaunay triangulation. *J. Comp. Phys.*, 90(2):271–291, 1990.

[14]S. D. Connell and D. G. Holmes. A 3D unstructured adaptive multigrid scheme for the Euler equations. *AIAA J.*, 32(8):1626–1632, 1994.

[15]R. Löhner and J. D. Baum. Adaptive H-refinement on 3-D unstructured grids for transient problems. *Int. J. Num. Meth. Fluids*, 14:1407–1419, 1992.

[16]Y. Kallinderis and P. Vijayan. Adaptive refinement-coarsening scheme for three-dimensional unstructured meshes. *AIAA J.*, 31(8):1440–1447, 1993.

[17]J. Y. Trepanier, H. Zhang, M. Reggio, and R. Camarero. Adaptive and moving meshes for the computation of unsteady compressible flows. In A. S. Arcilla, J. Hauser, P. R. Eiseman, and J. F. Thompson, editors, *Numerical grid generation in Computational Fluid dynamics and related fields*, pages 43–54, New York, 1991. North-Holland.

[18]G. P. Warren, W. K. Anderson, J. L. Thomas, and S. L. Krist. Grid convergence for adaptive methods. AIAA Paper 91-1592CP, July 1991.

[19]W. K. Anderson. Private communication. 1994.

[20]J. Peraire. Private communication. 1992.

[21]D. J. Mavriplis. Aerodynamic drag prediction using unstructured mesh solvers. In *CFD-Based Drag Prediction and Reduction, eds. H. Deconoinck, K. Sermus and C. van Dam, VKI Lecture Series 2003-02, von Karman Institute for Fluid Dynamics, Rhode St-Genese, Belgium*, March 2003.

[22]D. W. Levy and M. D. Thacker. Comparison of unstructured cell- and node-based schemes for the Euler equations. AIAA Paper 99-3185, June 1999.

[23]D. W. Levy, T. Zickuhr, J. Vassberg, S. Agrawal, R. A. Wahls, S. Pirzadeh, and M. J. Hemsch. Summary of data from the first AIAA CFD drag prediction workshop. AIAA Paper 2002-0841, January 2002.

[24]K. Laflin, O. Brodersen, M. Rakowitz, J. Vassberg, R. Wahls, and J. Morrison. Summary of data from the second AIAA CFD drag prediction workshop. AIAA Paper 2004-0555, January 2004.

[25]J. C. Vassberg, E. N. Tinoco, M. Mani, O. P. Brodersen, B. Eisfeld, R. A. Wahls, J. H. Morrison, T. Zickuhr, K. R. Laflin, and D. J. Mavriplis. Summary of the third AIAA CFD drag prediction workshop. AIAA Paper 2007-0260, January 2007.

[26]D. J. Mavriplis and D. W. Levy. Transonic drag prediction using an unstructured multigrid solver. *AIAA Journal of Aircraft*, 42(4):887–893, 2003.

[27]S. Z. Pirzadeh and N. T. Frink. Assessment of the unstructured grid software TetrUSS for drag prediction of the DLR-F4 configuration. AIAA Paper 2002-0839, January 2002.

[28]T. J. R. Hughes. Recent progress in the development and understanding of SUPG methods with special reference to the compressible Euler and Navier-Stokes equations. *Intl. J. for Numer. Meth. in Fluids*, 7:1261–1275, 1987.

[29]B. Cockburn, G. E. Karniadakis, and C. W. Shu. *Discontinuous Galerkin Methods: Theory, Computation and Applications*. Springer, 2000.

[30]M. Aftosmis, D. Gaitonde, and T. S. Tavares. On the accuracy, stability and monotonicity of various reconstruction algorithms for unstructured meshes. AIAA Paper 94-0415, January 1994.

[31]T. J. Barth. Parallel CFD algorithms on unstructured meshes. In *Special Course on Parallel Computing in CFD*, pages 7–1,7–41, May 1995. AGARD Report-807.

[32]H. Luo, J. D. Baum, R. Löhner, and J. Cabello. Adaptive edge-based finite element schemes for the Euler and Navier-Stokes equations on unstructured grids. AIAA Paper 93-0336, January 1993.

[33]D. Sidilkover. A genuinely multidimensional upwind scheme and efficient multigrid solver for the compressible Euler equations. ICASE Report 94-84, October 1994.

[34]H. Deconinck, H. Paillere, R. Struijs, and P. L. Roe. Multidimensional upwind schemes based on fluctuation-splitting of conservation laws. *Comp. Mechanics*, 11(5/6):323–340, 1993.

[35]K. Nakahashi. A finite-element method on prismatic elements for the three-dimensional Navier-Stokes equations. In *Lecture Notes in Physics*, volume 323. Springer Verlag, 1989.

[36]Y. Kallinderis and S. Ward. Hybrid prismatic/tetrahedral grid generation for complex geometries. AIAA Paper 93-0669, January 1993.

[37]S. Pirzadeh. Three-dimensional unstructured viscous grids by the advancing-layers method. *AIAA Journal*, 34(1):43–49, 1996.

[38]D. J. Mavriplis. Unstructured mesh generation and adaptivity. In *VKI Lecture Series VKI-LS 1995-02*, March 1995.

[39]M. Paraschivoiu, J. Y. Trepanier, M. Reggio, and R. Camarero. A conservative dynamic discontinuity tracking algorithm for the Euler equations. AIAA Paper 94-0081, January 1994.

[40]J. van Rosendale. Floating shock fitting via lagrangian adaptive meshes. In *Proceedings of the 12th AIAA CFD Conference, San Diego CA*, June 1995. AIAA Paper 95-1721-CP.

[41]A. Jameson. Analysis and design of numerical schemes for gas dynamics 1, artificial diffusion, upwind biasing, limiters and their effect on multigrid convergence. *Int. J. of Comp. Fluid Dyn.*, 4:171–218, 1995.

[42]T. J. Barth. Numerical aspects of computing viscous high Reynolds number flows on unstructured meshes. AIAA Paper 91-0721, January 1991.

[43]W. K. Anderson and D. L. Bonhaus. An implicit upwind algorithm for computing turbulent flows on unstructured grids. *Computers Fluids*, 23(1):1–21, 1994.

[44]A. Haselbacher and J. Blazek. Accurate and efficient discretization of NavierStokes equations on mixed grids. *AIAA Journal*, 38(11):2094–2102, 2000.

[45]D. J. Mavriplis. Revisiting the least-squares procedure for gradient reconstruction on unstructured meshes. AIAA-Paper 2003-3986, June 2003.

[46]N. B. Petrovskaya. The impact of grid cell geometry on the least-squares gradient reconstruction. Technical Report No. 21, Keldysh Institute of Applied Mathematics, Russian Academy of Sciences, Moscow, April 2003.

[47]V. Venkatakrishnan. On the accuracy of limiters and convergence to steady state solutions. *J. Comp. Phys.*, 118:120–130, 1995.

[48]A Harten, P. D. Lax, and B Van Leer. On upstream differencing and godunov-type schemes for hyperbolic conservation laws. *SIAM Review*, 25(1):35–61, 1983.

[49]P. L. Roe. Approximate Riemann solvers, parameter vectors and difference schemes. *J. Comp. Phys.*, 43(2):357–372, 1981.

[50]A. Harten. High resolution schemes for hyperbolic conservation laws. *J. Comp. Phys.*, 135(2):260–278, 1997.

[51]W. K. Anderson. A grid generation and flow solution method for the Euler equations on unstructured grids. *J. Comp. Phys.*, 110(1):23–38, 1994.

[52]I. Babushka and A. K. Aziz. On the angle condition in the finite-element method. *SIAM Journal of Numerical Analysis*, 13(6), 1976.

[53]A. Haselbacher, J. McGuirk, and G. Page. Finite-volume discretization aspects for viscous flows on unstructured meshes. *AIAA Journal*, 37(2):177–184, 1999.

[54]R. Lohner, H. Luo, and J. D. Baum. Selective edge removal for unstructured grids with cartesian cores. *Journal of Computational Physics*, 206(1):208–226, 2005.

[55]S. D. Connell and M. E. Braaten. Semi-structured mesh generation for three-dimensional Navier-Stokes calculations. *AIAA J.*, 33(6):1017–1024, June 1995.

[56]R. Lohner. Matching semi-structured and unstructured grids for Navier-Stokes calculations. AIAA paper 93-3348, July 1993.

[57]D. J. Mavriplis. A three-dimensional multigrid Reynolds-averaged Navier-Stokes solver for unstructured meshes. *AIAA Journal*, 33(3):445–4531, March 1995.

[58]K. Xu, C. A. Kim, L. Martinelli, and A. Jameson. BGK-based schemes for the simulation of compressible flow. *Int. J. Computational Fluid Dynamics*, 7:213–235, 1996.

[59]G. May and A. Jameson. Improved gaskinetic multigrid method for three-dimensional computation of viscous flow. AIAA Paper 2005-5106, June 2005.

[60]N. T. Frink. Upwind scheme for solving the Euler equations on unstructured tetrahedral meshes. *AIAA J.*, 30(1):70–77, 1992.

[61]V. Venkatakrishnan and T. J. Barth. Application of direct solvers to unstructured meshes for the Euler and Navier-Stokes equations using upwind schemes. AIAA Paper 89-0364, January 1989.

[62]J. T. Batina. Implicit upwind solution algorithms for three-dimensional unstructured meshes. *AIAA J.*, 31(5):801–805, May 1993.

[63]R. R. Thareja, J. R. Stewart, O. Hassan, K. Morgan, and J. Peraire. A point implicit unstructured grid solver for the Euler and Navier-Stokes equations. AIAA Paper 88-0036, January 1988.

[64]V. Venkatakrishnan and D. J. Mavriplis. Implicit method for the computation of unsteady flows on unstructured grids. In *Proceedings of the 12th AIAA CFD Conference, San Diego CA*, June 1995. AIAA Paper 95-1705-CP.

[65]H. Guillard. Node nested multigrid with Delaunay coarsening. INRIA Report No. 1898, 1993.

[66]M. E. Braaten and S. D. Connell. Three dimensional unstructured adaptive multigrid scheme for the Navier-Stokes equations. *AIAA J.*, 34(2):281–290, February 1996.

[67]B. R. Hutchinson, P. F. Galpin, and G. D. Raithby. Application of additive correction multigrid to the coupled fluid flow equations. *Numerical Heat Transfer*, 13:133–147, 1988.

[68]W. A. Smith. Multigrid solution of transonic flow on unstructured grids. In *Recent Advances and Applications in Computational Fluid Dynamics*, November 1990. Proceedings of the ASME Winter Annual Meeting, Ed. O. Baysal.

[69]M. Lallemand, H. Steve, and A. Dervieux. Unstructured multigridding by volume agglomeration: Current status. *Computers and Fluids*, 21(3):397–433, 1992.

[70]V. Venkatakrishnan and D. J. Mavriplis. Agglomeration multigrid for the three-dimensional Euler equations. *AIAA Journal*, 33(4):633–640, April 1995.

[71]D. J. Mavriplis and V. Venkatakrishnan. A unified multigrid solver for the Navier-Stokes equations on mixed element meshes. *International Journal for Computational Fluid Dynamics*, 8:247–263, 1997.

[72]R. S. Tuminaro and C. Tong. Parallel smoothed aggregation multigrid : Aggregation strategies on massively parallel machines. Proceedings of the 2000 ACM/IEEE conference on Supercomputing, November 2000.

[73]D. Martin and Löhner. An implicit linelet-based solver for incompressible flows. AIAA Paper 92-0668, January 1992.

[74]D. J. Mavriplis. Multigrid strategies for viscous flow solvers on anisotropic unstructured meshes. *Journal of Computational Physics*, 145(1):141–165, September 1998.

[75]D. J. Mavriplis. Directional agglomeration multigrid techniques for high-Reynolds number viscous flow solvers. *AIAA Journal*, 37(10):1222–1230, October 1999.

[76]E. J. Nielsen, J. Lu, M. A. Park, and D. L. Darmofal. An implicit exact dual adjoint solution method for turbulent flows on unstructured grids. *Computers and Fluids*, 33(9):1131–1155, November 2004.

[77]I. Nompelis, T. Drayna, and G. Candler. A parallel unstructured implicit solver for hypersonic reacting flow simulation. AIAA Paper 2005-4867, June 2005.

[78]S. Balay, K. Buschelman, V. Eijkhout, W. D. Gropp, D. Kaushik, M. G. Knepley, L. C. McInnes, B. F. Smith, and H. Zhang. PETSc user's manual. ANL-95/11 -Revision 2.1.5,Argonne National Laboratory, 2004.

[79]D. J. Mavriplis, M. Aftosmis, and M. Berger. High-resolution aerospace applications using the NASA Columbia supercomputer. Paper presented at the 2005 Supercomputing Conference, Seattle, WA. Tied for Best paper award at SC05., November 2005.

[80]G. Karypis and V. Kumar. A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM Journal on Scientific Computing*, 20(1):359–392, 1999.

[81]D. J. Mavriplis. A CFD package for multi-element airfoil high-lift analysis. NSU2D User's Manual (Revision 4.0), December 1996.

[82]P. R. Spalart and S. R. Allmaras. A one-equation turbulence model for aerodynamic flows. *La Recherche Aérospatiale*, 1:5–21, 1994.

[83]F. M. White. *Viscous Fluid Flow*. McGraw-Hill, New York, NY, 1991.

[84]W. K. Anderson, D. L. Bonhaus, R. J. McGhee, and B. S. Walker. Navier-Stokes computations and experimental comparisons for multielement airfoil configurations. *AIAA J. of Aircraft*, 32(6):1246–1253, 1995.

[85]E. M. Lee-Rausch, P. B. Buning, D. J. Mavriplis, J. H. Morrison, M. A. Park, S. M. Rivers, and C. L. Rumsey. CFD sensitivity analysis of a drag prediction workshop wing/body transport configuration. AIAA Paper 2003-3400, June 2003.

[86]E. M. Lee-Rausch, N. T. Frink, D. J. Mavriplis, R. D. Rausch, and W. E. Milholen. Transonic drag prediction on a DLR-F6 transport configuration using unstructured grid solvers. AIAA Paper 2004-0554, January 2004.

[87]D. J. Mavriplis. Grid resolution study of a drag prediction workshop configuration using the NSU3D unstructured mesh solver. AIAA-Paper 2005-4729, June 2005.

[88]S. Pirzadeh. Private communication. 2005.

[89]E. M. Lee-Rausch, M. Park, E. Nielsen, W. Jones, and D. Hammond. Parallel adjoint-based error estimation and anisotropic grid adaptation for three-dimensional aerospace applications. AIAA Paper 2005-4842, June 2005.

[90]P. Gnoffo and J. A. White. Computational aerothermodynamic simulation issues on unstructured grids. AIAA Paper 2004-2371, June 2004.

[91]M. Aftosmis M. Berger and J. Melton. Robust and efficient cartesian mesh generation for component-based geometry. AIAA Paper 97-0196, January 1997.

[92]S. Murman, M. Aftosmis, and M. Berger. Numerical simulation of rolling airframes using a multi-level cartesian method. AIAA Paper 2002-2798, June 2002.

[93]S. Z. Tu and S. M. Ruffin. Solution adaptive, unstructured Cartesian-grid methodology for chemically reacting flow. AIAA Paper 2002-3097, June 2002.

[94]K. Fidkowski and D. Darmofal. An adaptive simplex cut-cell method for discontinuous galerkin discretizations of the navier- stokes equations. AIAA Paper 2007-3941, June 2007.