AIAA JOURNAL
Vol. 46, No. 6, June 2008

Unstructured-Mesh Discretizations and Solvers for Computational Aerodynamics

Dimitri J. Mavriplis*
University of Wyoming, Laramie, Wyoming 82071-3295

DOI: 10.2514/1.34681

An overview of current unstructured-mesh discretization and solution techniques is given. Cell-centered vs vertex-centered discretizations are discussed, as well as issues of grid alignment with flow features, higher-order reconstruction, and viscous-term formulation. With regard 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 1980s, 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, in which 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 [3–5], for which approximate Riemann solver technology could be applied [5,6] and existing solution strategies could be adopted [7–11]. 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 strategies also became one of the often-quoted advantages of the unstructured-mesh approach [12–18].

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.

In hindsight, on one hand, it is surprising 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 [19–21] or cut-cell Cartesian approaches [22–25]. 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 due to the lack of reliable error-estimation techniques for driving the refinement criterion.

Although it is difficult and somewhat artificial to separate 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. Before 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 that separate unstructured-mesh approaches from competing methods.

As previously stated, 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 regard to complex geometries, the principal alternatives have been the use of block-structured grids, or overset grids. With regard to adaptive meshing strategies, these can be performed on any type of mesh, and the defining difference is whether the adapted mesh is

Dimitri Mavriplis is a Professor of mechanical engineering at the University of Wyoming. He obtained his Ph.D. in mechanical and aerospace engineering from Princeton University in 1987. He joined the Institute for Computer Applications in Science and Engineering (ICASE), located at the NASA Langley Research Center in Hampton, VA, as a Staff Scientist in 1987, and he became an ICASE Research Fellow in 1997. In January 2003, he joined the National Institute of Aerospace in Hampton, VA, as a Research Fellow before moving to the University of Wyoming in August 2003. His research has focused on the development of computational fluid dynamics (CFD) methods using unstructured meshes, and he has published over 125 papers through the AIAA and other venues. He is a frequent consultant to industry on CFD matters, he serves as a Member of the AIAA Drag Prediction Workshop Committee, and he is an Associate Fellow of the AIAA.

Presented as Paper 3955 at the 18th AIAA Computational Fluid Dynamics Conference, Miami, FL, 25–28 June 2007; received 19 September 2007; revision received 17 January 2008; accepted for publication 20 January 2008.

*Professor, Department of Mechanical Engineering; mavripl@uwyo.edu. Associate Fellow AIAA.

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 fully 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 lie, on 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 to assess the strengths and weaknesses of unstructured-mesh approaches. Briefly stated, the element type principally affects spatial discretization accuracy, and 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 viscous-term discretizations.

A. Cell or Vertex Approaches

Perhaps another surprising situation in hindsight is that after 20 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.

Although 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 five or six 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 an average of 14 neighbors, based on the number of edges in the mesh, which can be shown to be on 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, because 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 that which provides the highest accuracy at the lowest cost. Although numerical experiments conducted by the author and others [7,26] have verified the superior accuracy of cell-centered approaches vs 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 and Thacker [27] found that matching the number of surface-grid variables for both grids achieved similar accuracy for aerodynamic quantities in transonic flow cases. Because there are twice as many triangular elements on the surface as there are vertices in a given grid, this results in vertex-based grid sizes that, 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) [28–30] 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 [28,31,32], in which 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).

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

The appearance of edge-based and face-based data structures for vertex- and cell-centered discretizations, respectively, has made it

Table 1 Detailed description of three equivalent unstructured meshes for computation of flow over DLR-F4 wing–body configuration; reproduced from [31]

Grid characteristics	Vertex-based grid (full viscous)	Cell-based grid (full viscous)	Cell-based grid (full viscous)
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 the wing trailing edge	5	5	5
Chordwise grid spacing at the leading edge	≈0.250 mm	≈0.450 mm	≈0.450 mm
Chordwise grid spacing at the trailing edge	≈0.500 mm	≈0.800 mm	≈0.800 mm
Maximum spanwise spacing at the leading edge	≈2.500 mm	≈6.000 mm	≈6.000 mm
Maximum spanwise spacing at the trailing edge	≈3.500 mm	≈3.500 mm	≈3.500 mm
Grid spacing on the fuselage	≈10.00 mm	≈10.00 mm	≈10.00 mm
Grid spacing at the outer boundary	≈3000.00 mm	≈3000.00 mm	≈3000.00 mm
Normal spacing at the 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

possible to construct a solver that can be operated in either mode, with relatively minor modifications that mostly concern boundary conditions, at least for inviscid flows. Although this has seldom been done, it would provide a more solid basis for comparing cell-vs-vertex types of discretizations.

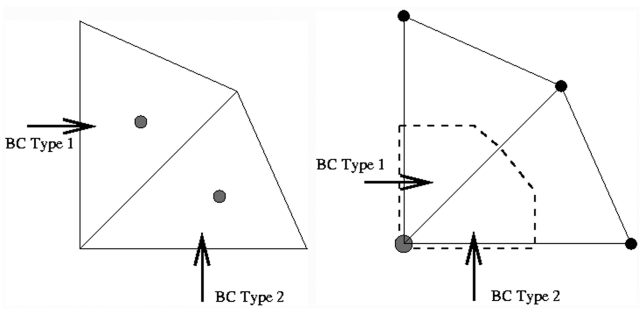
Note that this issue is most important for tetrahedral meshes and is only of secondary importance on hexahedral meshes. For prismatic-element meshes, such as those often used in boundary-layer regions, the differences between cell-centered and vertex-based approaches are less pronounced than for tetrahedral 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 [33], corresponds to a vertex-based discretization, whereas the discontinuous Galerkin approach [34] 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. Because 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, whereas the cell-centered approach is well suited for Neumann boundary conditions. However, a more fundamental problem arises in the vertex-based discretization, because 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 Fig. 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. To implement such a boundary condition in the case of vertex-based discretizations, a weak formulation must be used, in which 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 that are similar for both the cell-centered and vertex-centered discretizations, as shown in Fig. 1.

B. Inviscid Discretization

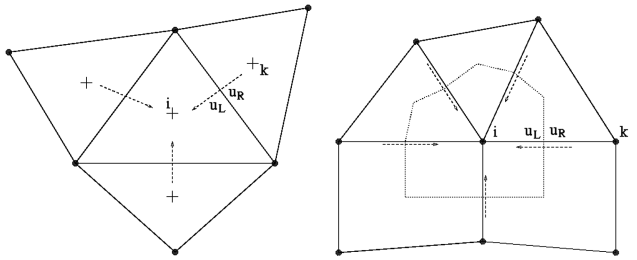
1. *First Order*

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 Fig. 2a, 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 midpoints,



a) b)

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



a) b)

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

known as the centroidal dual control volume, as shown in Fig. 2b, although other constructions are possible [35,36]. 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 [8,37]. 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. Because this strategy almost exactly mimics 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 is generally simpler with structured-grid approaches than with 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 that 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 multidimensional approximate Riemann solvers, which would be insensitive to the orientation of the face along which the flux is computed [38,39]. Although progress has been made in this area, this remains a difficult problem that, to date, has 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 that are aligned with the flow direction for boundary-layer resolution [40–43]. For flows with strong shocks, nonaligned 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 [44,45]. Current ongoing 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 Fig. 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 in which the grid-point

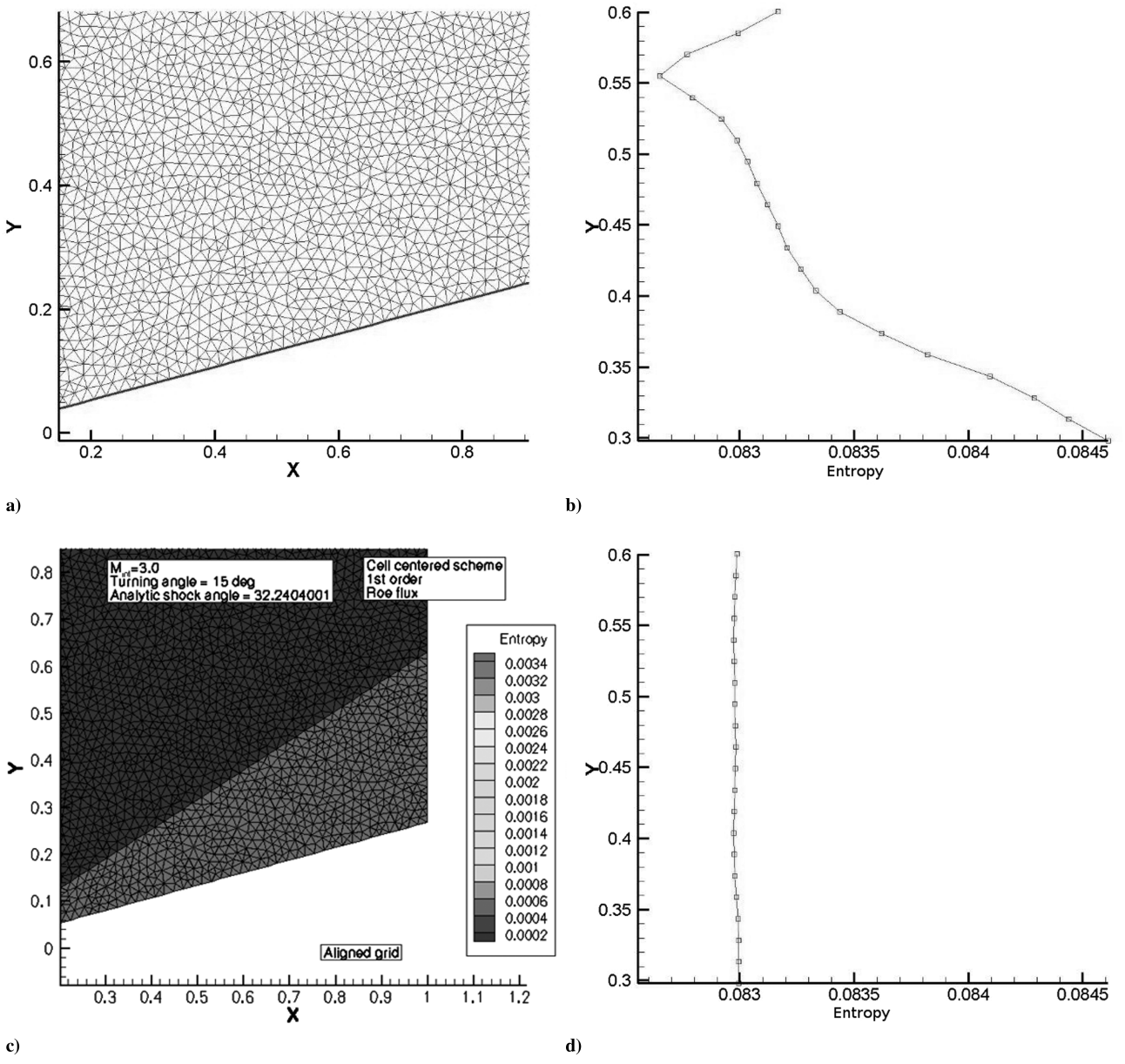


Fig. 3 Illustration of shock-fitting through mesh-face alignment in two dimensions.

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 in which 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 that 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 using, for example, the Roe scheme [46], as

$$\mathbf{F}_{ik} = \mathbf{F}(u_L, u_R) = \tfrac{1}{2}[F(u_L) + F(u_R) + T|\Lambda|T^{-1}(u_L - u_R)] \quad (1)$$

where $F(u)$ represents the convective flux, and 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:

$$u_L = u_i \quad (2)$$

$$u_R = u_k \quad (3)$$

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

$$u_L = u_i + \nabla u_i . \boldsymbol{r}_{if} \quad (4)$$

$$u_R = u_k + \nabla u_k . \boldsymbol{r}_{kf} \quad (5)$$

where $\boldsymbol{r}_{if}$ 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

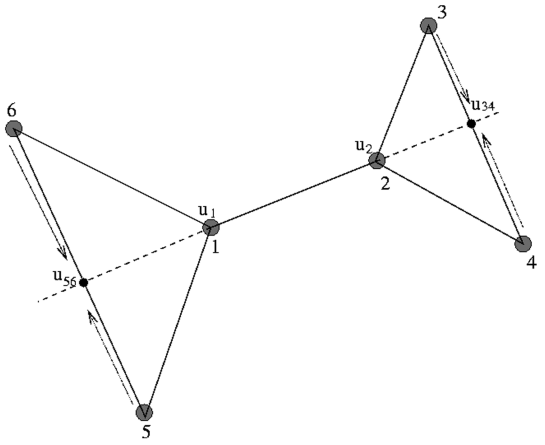


Fig. 4 Illustration of one-dimensional reconstruction technique used in [5] and by CUSP schemes in [47].

information is obtained by locating the intersection of lines drawn tangent to the current edge with surrounding triangles, from which solution values are interpolated, as shown in Fig. 4. A similar approach was later adopted by the convective upwind and split pressure (CUSP) schemes developed by Jameson [47]. However, the requirement of additional data structures to represent the cells intersected by the extended edges remains a disadvantage of these approaches.

By far, the most common approach today is based on the multidimensional 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, because this leads to overestimation of solution variations and potentially nonphysical 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 [36,48–50]. However, in the context of the least-squares 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 [51,52], as will be shown in subsequent examples.

For flows with strong shock waves, the use of limiters becomes necessary for preventing solution overshoots that 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 and to prevent robust convergence, due to nonsmooth application and limit-cycle behavior. These problems can be mitigated to some degree using total-variation-bounded limiters [53], 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 precomputed second differences or undivided Laplacian operators on each side of the interface and are then multiplied by the appropriate (largest) eigenvalue of the flowfield at the interface condition to create a dissipative term as

$$\mathbf{F}_{ik} = \tfrac{1}{2}[F(u_i) + F(u_k) + \alpha(L_i(u) - L_k(u))] \tag{6}$$

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

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

This approach is known as the *scalar* artificial-dissipation approach, which corresponds to a Lax–Friedrichs scheme [54] and is known to result in excessively diffusive results. An improvement over this approach consists of taking α as a matrix, rather than as a scalar. In this case, a natural choice for α, by analogy with Eq. (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, because 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 and 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.

The T matrices on the right-hand side of Eq. (1) represent the eigenvectors associated with the linearization of the equations of inviscid compressible flow normal to the control-volume face ik [55], and 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 that are proportional to u, $u + c$, and $u - c$, 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

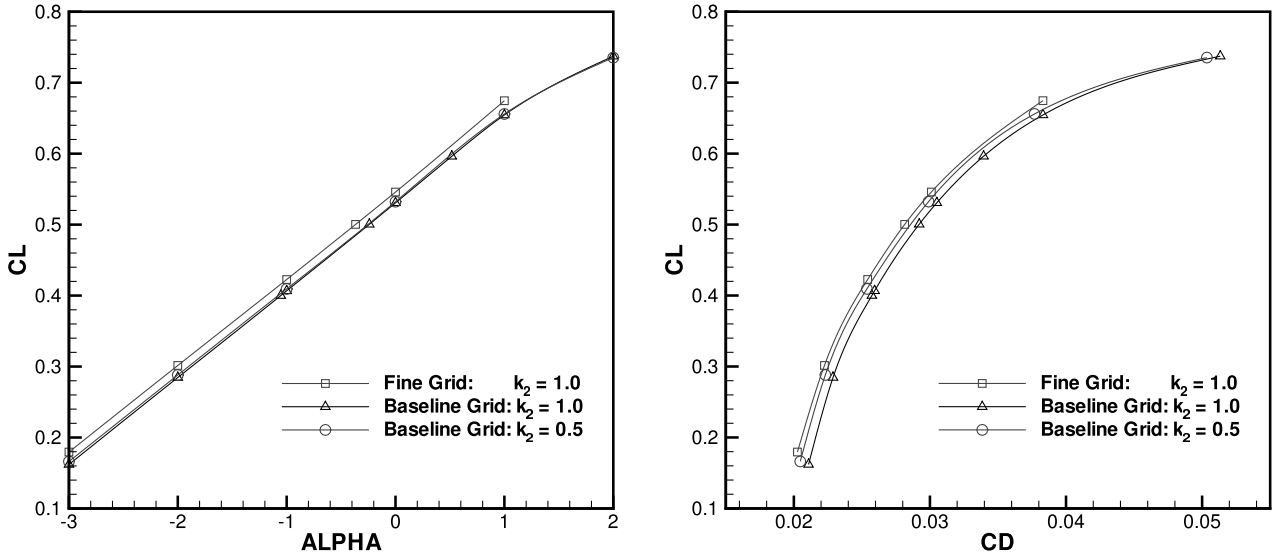
$$u = \text{sign}(u) \times \max(|u|, \delta(|u| + c)) \tag{9}$$

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

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

where $|u| + c$ is the maximum eigenvalue, and δ is a parameter that varies between 0 and 1, to be chosen empirically. 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 [54]. Small values of δ on the order of 0.1 are common in many production codes, and this process is often referred to as an *entropy fix* [56].

These various forms of the convection-term discretization 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



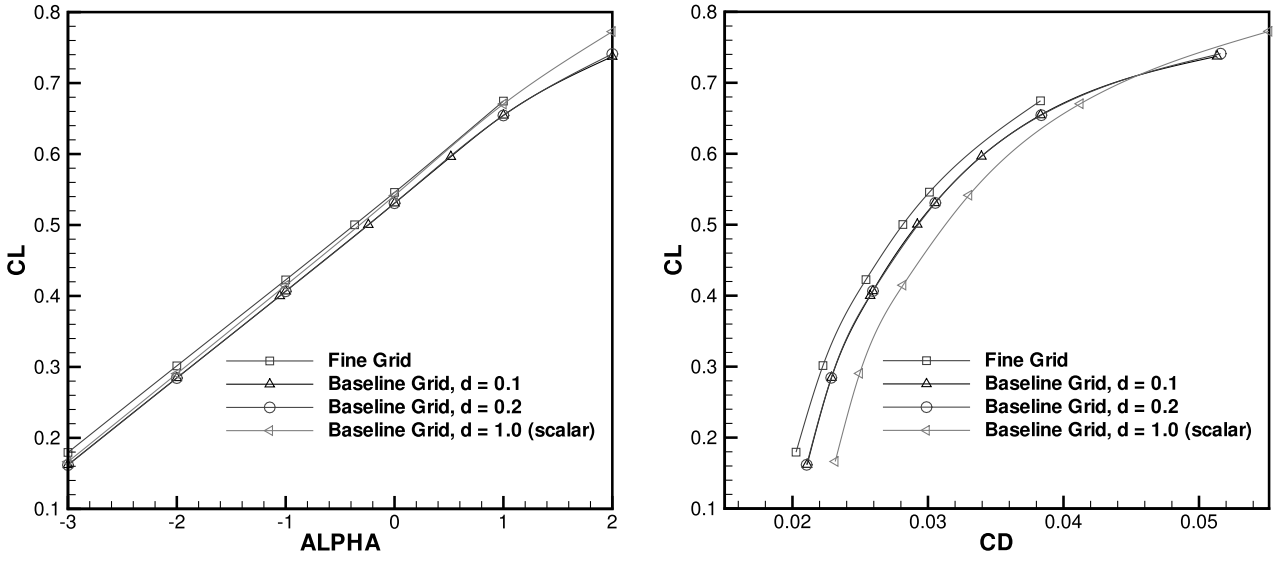
a) b)

Fig. 5 Comparison of computed lift and drag coefficients on the DLR-F4 configuration at transonic conditions for different values of the artificial-dissipation scaling coefficient.

should accentuate the differences between the discretizations that we are seeking, because all the considered discretizations are consistent and should converge toward 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 5 and 6 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 deg. 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, and the drag decreases by six counts at 0-deg incidence. The lower dissipation value appears to yield higher accuracy, because the results tend toward those computed on the finer grid. This is expected, because one of the main characteristics of increased grid resolution is the reduction of artificial-dissipation effects. Although 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 be acceptable in a production environment.

The sensitivity due to the value of the entropy fix in the artificial-dissipation discretization is depicted in Fig. 6. Doubling the entropy-fix value from $\delta = 0.1$ to 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 nonzero 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-deg 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 with the matrix-dissipation formulation. However, these savings are



a) b)

Fig. 6 Comparison of computed lift and drag coefficients on the DLR-F4 configuration at transonic conditions for different values of the entropy fix for matrix artificial-dissipation formulation.

Table 2 Variations of computed lift and drag values at a Mach number of 0.75, a Reynolds number of 3×10^6, and a 0-deg incidence for the DLR-F4 test case as a function of variations in artificial-dissipation parameters

Discretization	C_L	C_D
Fine mesh (13 million points)	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

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, because the small increase in lift values associated with the scalar dissipation computations may, in isolation, convey an impression of higher accuracy with regard to the baseline case.

Table 3 Variations of computed lift and drag values at a Mach number of 0.75, a Reynolds number of 3×10^6, and a 0-deg 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 (13 million points)	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

In Fig. 7 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 [48,57], 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 because the lift values

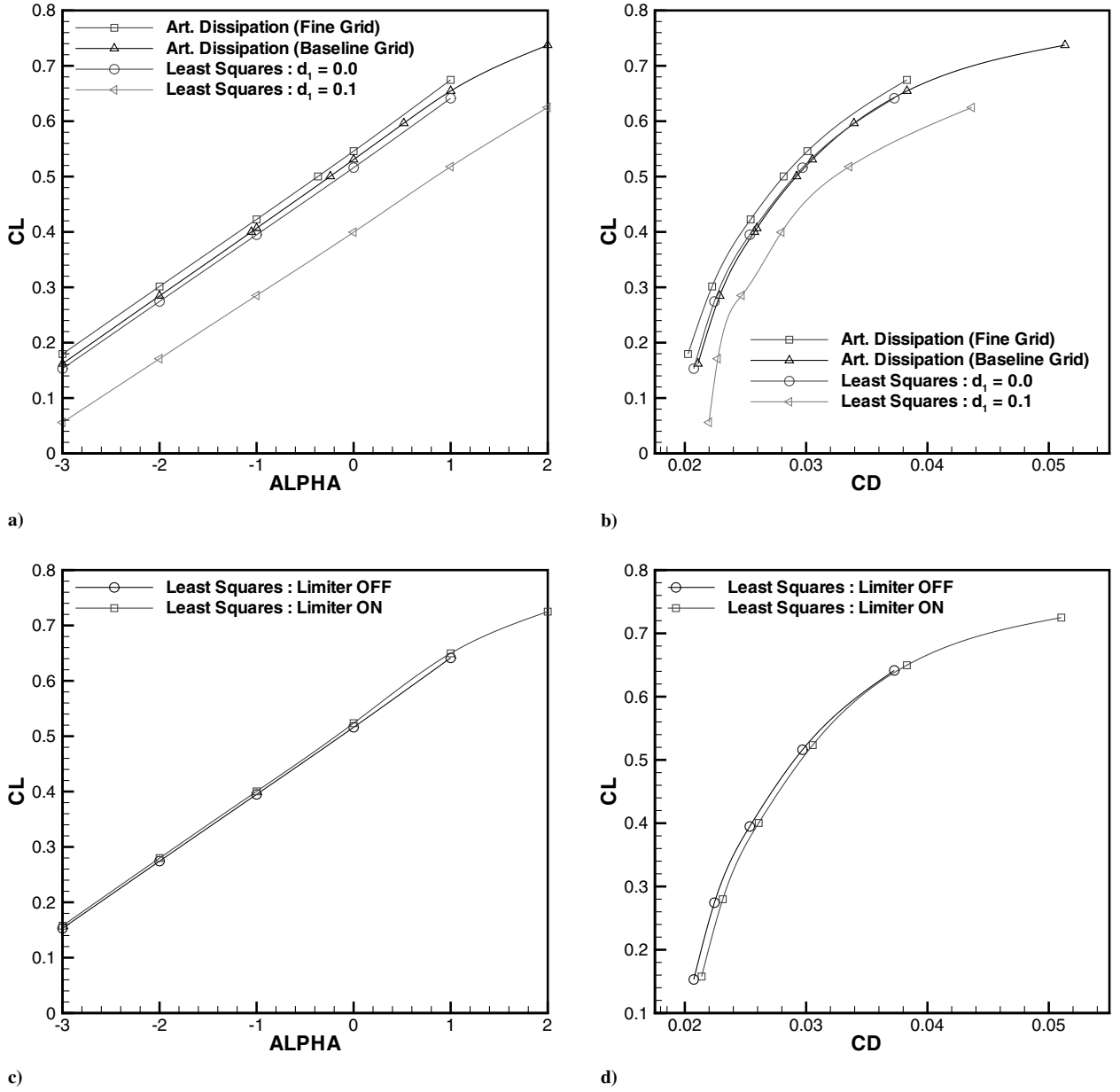


Fig. 7 Comparison of computed lift and drag coefficients on the DLR-F4 configuration at transonic conditions using the upwind-reconstruction approach for various settings of the entropy fix and limiters.

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 was 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-deg incidence is seen to be eight counts lower than that computed using the matrix-dissipation scheme on the same grid. Therefore, there are more substantial differences between these two schemes that extend beyond the simple scaling of the final dissipative terms.

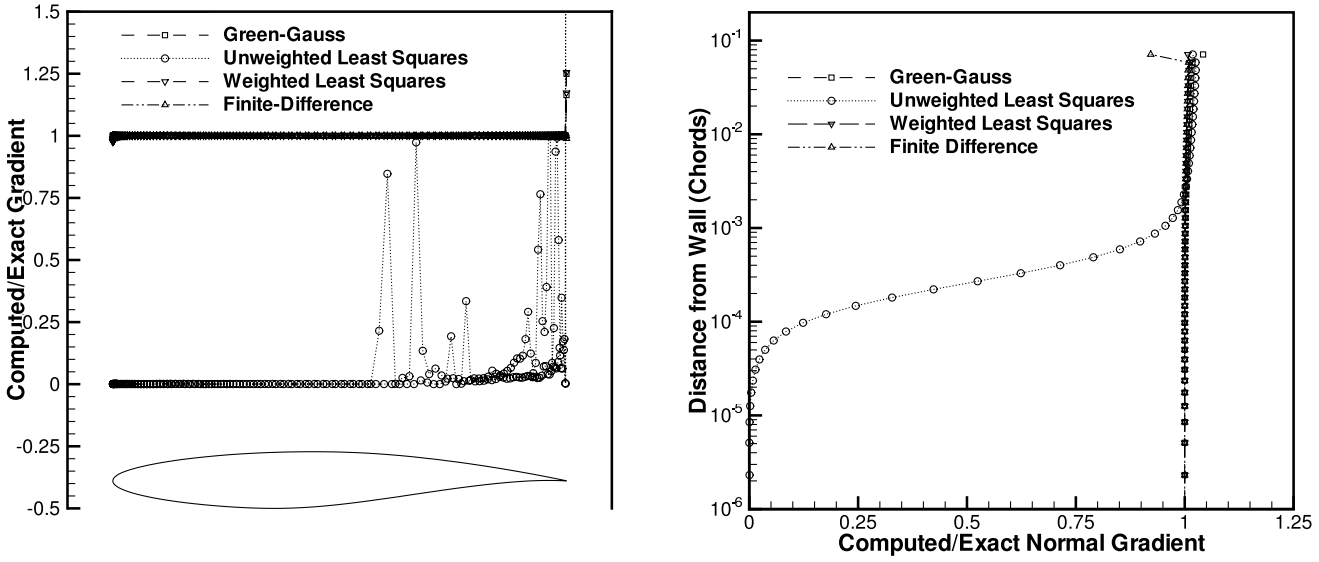
Returning to the baseline unlimited case, the entropy-fix parameter is now increased from 0.0 to $\delta = 0.1$, which is the level used in the baseline matrix-dissipation settings. In this case, the computed lift values are substantially lower than those obtained with a vanishing entropy fix, and the drag values are substantially higher. From Table 3, the drag at 0-deg incidence is seen to increase by 17 counts, with much larger variations in lift. In essence, the accuracy of the scheme was 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 [51,52]. This is illustrated in Fig. 8, in which 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 underpredicts the gradient values in the near-wall regions. This behavior has been attributed to the effect of surface curvature that results in dominant contributions to the normal gradient from misaligned streamwise stencil points [51]. By underestimating 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.

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 [51]. 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 vanish. Because this normal velocity represents one of the eigenvalues of the dissipation matrix [cf. Eq. (1)], it is seen that the additional dissipative effect due to the poor gradient estimate is counterbalanced by a vanishing eigenvalue at precisely 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 [51]. 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 to examine the effect on solution accuracy. The multidimensional monotonicity-preserving limiter from Barth and Jespersen [6] is employed. A small increase in lift and a moderate increase in drag are observed in Figs. 7c and 7d, respectively. At 0-deg incidence, the drag is increased by eight 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 in which 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 nonstrictly monotone limiter of Venkatakrishnan [53], which result in less accuracy degradation, and the current results should be regarded as establishing an outer bound to limiter sensitivity.

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 that are typical of aerodynamic calculations. The second relates to the fact that the viscous terms involve second-derivative terms, as opposed to convective terms, which are only first derivatives.



a) b)

Fig. 8 Computed gradient of test function (normalized by exact value) for a typical viscous-type grid about an RAE 2822 airfoil illustrating poor accuracy of the unweighted least-squares approach in the presence of surface curvature (note that all other methods and their symbols essentially overlap); reproduced from [51].

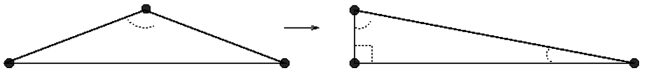
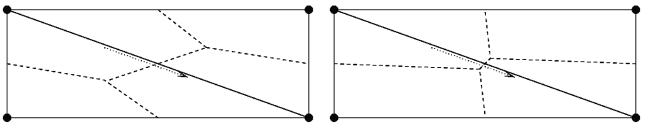


Fig. 9 Illustration of two types of anisotropic triangular cells: the first configuration contains one large obtuse angle and the second contains one small angle and two angles that are close to 90 deg.

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 that exist in the high-Reynolds-number flows that 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 [58].

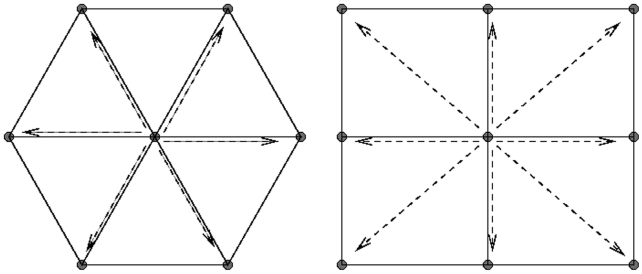
The construction of a highly stretched triangular mesh with only small angles and no obtuse angles leads to the generation of a series of nearly right-angle triangles, with the small triangle edge normal to the flow feature or direction of stretching, as shown in Fig. 9. Although a highly stretched mesh topology with right-angle triangles (or tetrahedra in 3-D) provides much better accuracy for such flows, the diagonal edge of such cells remains nonaligned 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 nonaligned 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 nonaligned grid faces, such as the use of containment dual control volumes [36,59], as shown in Fig. 10, which deemphasize 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 [60] 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 semistructured meshes using prismatic elements in these regions [61,62].

Once a suitable mesh has been constructed, the second-derivative viscous terms must be discretized on this topology. Although 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 [36,63]. However, for nonsimplicial elements, this discretization results in a



a) b)

Fig. 10 Illustration of a) median dual control volume on an anisotropic stretched mesh configuration and b) containment dual control volume (approximate configuration for illustration purposes) that deemphasizes diagonal-edge contribution.



a) b)

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

stencil that contains diagonally located vertices in adjacent elements, as shown in Fig. 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 as

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

In other words, the gradient at the control-volume face ij is taken as the average of the two precomputed gradients within 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 multidimensional reconstruction of second and mixed derivatives required by the full Navier–Stokes equations and to 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

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

where $\mathbf{n}$ represents the unit face normal, $\boldsymbol{N}$ is the area-weighted face normal, and $\boldsymbol{r}_j - \boldsymbol{r}_i$ 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 preceding approach only constitutes a consistent discretization of the Laplacian for cases in which 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, in which the estimate of the full gradient at the control-volume face given by Eq. (12) is replaced by

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

effectively using Eq. (13) for the contribution to the gradient in the direction of the edge joining control-volume centroids i and j and

Table 4 Computed lift and drag coefficients at a Mach number of 0.75 and a 0-deg incidence on various grids using the multidimensional thin-layer discretization and the extended-stencil full Navier–Stokes discretization

Grid size	Viscous terms	C_L	C_D	$C_{D_{\text{pressure}}}$	$C_{D_{\text{friction}}}$
1 million points	Thin-layer approx.	0.5055	0.02960	0.01661	0.01298
1 million points	Full Navier–Stokes	0.4960	0.02921	0.01652	0.01269
3 million points	Thin-layer approx.	0.5012	0.02859	0.01568	0.01290
3 million points	Full Navier–Stokes	0.5020	0.02841	0.01571	0.01270
9 million points	Thin-layer approx.	0.5141	0.02882	0.01592	0.01290
9 million points	Full Navier–Stokes	0.5154	0.02867	0.01594	0.01273

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 (DPW2) DLR-F6 configuration using the inconsistent Laplacian or multidimensional thin-layer approximation given by Eq. (13), compared with the inclusion of the full Navier–Stokes viscous terms, using the discretization given by Eq. (14). For this test case, which includes nontrivial 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 multidimensional 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 Bhatnagar–Gross–Krook (BGK) scheme [64]. 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 [65] and may provide an alternative to the techniques previously described.

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 [4,6,15,66], because the more advanced solution techniques of the day (such as alternating direction implicit, approximate factorization, 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 nonlinear 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 [8–11,67–70]. 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 nonnested coarse and fine unstructured meshes [7–9,71,72]. 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 [73–78]. 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.

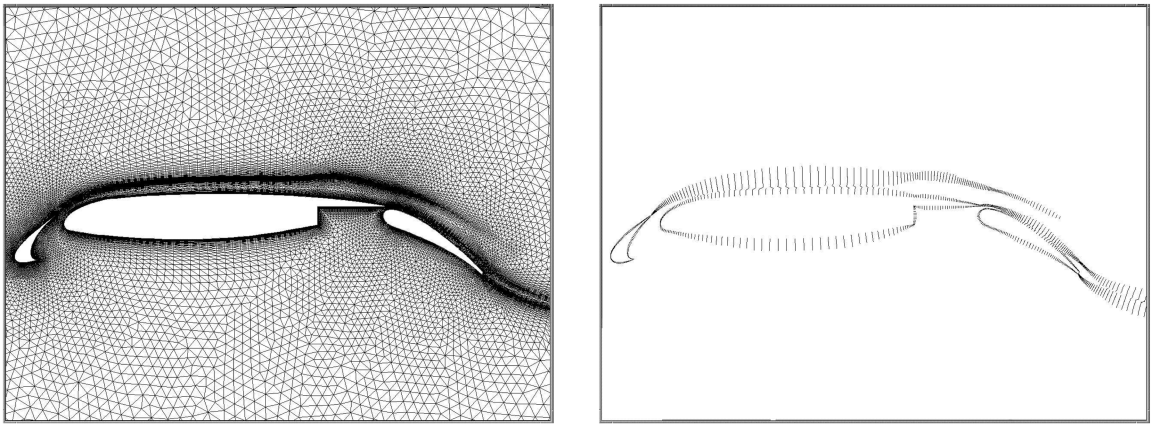
Other techniques, such as implicit line solution methods, have also been applied to unstructured-mesh methods with great success [79–83]. Although grid lines are naturally existing structures within a structured, block-structured, or overset-mesh system, they can be easily created in an unstructured mesh by grouping subsets of contiguous mesh edges or faces using a graph algorithm or other identification means, as shown in the two-dimensional example depicted in Fig. 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 manner similar 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.

Figure 13 provides an example of the effectiveness of line solvers and multigrid for unstructured meshes. The test case is taken from the AIAA DPW series [29] and consists of a wing–body configuration at the transonic Mach number of 0.75, 0-deg incidence, and a Reynolds number of 3×10^6. In Fig. 13a, using a mesh of 3 million points with 1.e-06 chord 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 Fig. 13b, 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 were 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, [84]).

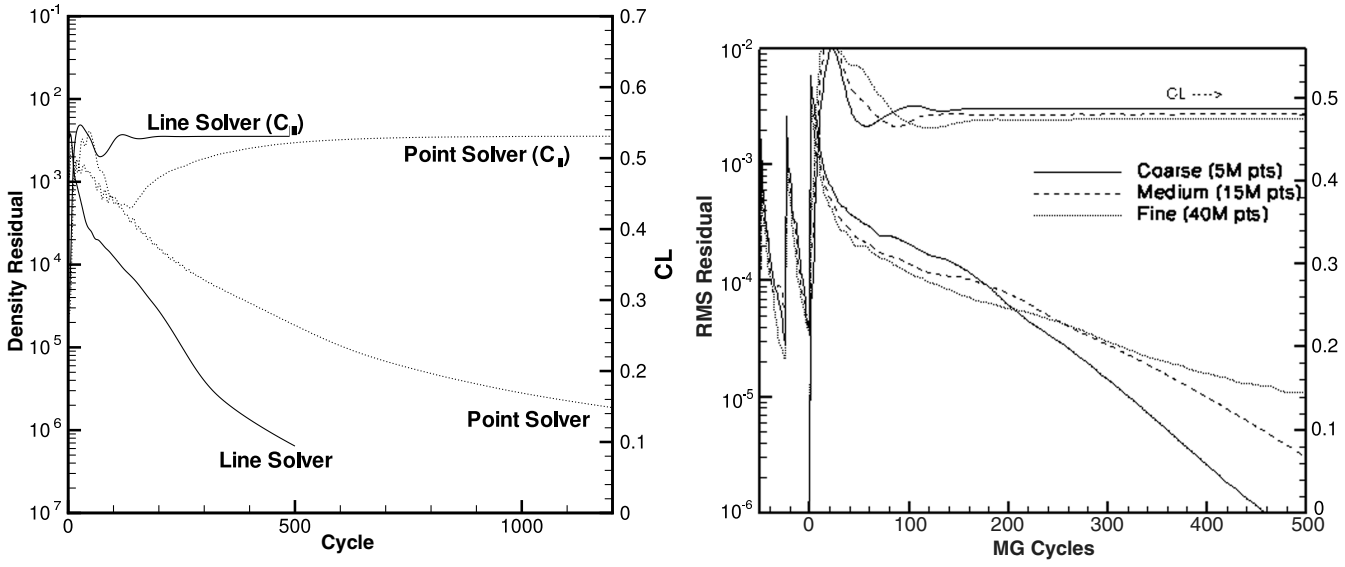
IV. Efficient Hardware Implementation

One of the initial perceived drawbacks of unstructured-mesh discretizations is that they require the explicit storage of the mesh connectivity, as well as the use of indirect memory references, which can lead to slower computational rates and larger memory requirements than with structured-grid methods. Although these were nontrivial issues in the early development of unstructured-mesh methods, when vector computers with relatively limited expensive



a) b)

Fig. 12 Two-dimensional illustration of the extraction of a set of lines from the unstructured mesh for the implicit line solver.



a) b)

Fig. 13 Illustration of the a) effectiveness of the line-implicit solver vs the point-implicit solver on highly anisotropic meshes used as a driver for multigrid and b) weak dependence of convergence on grid resolution for the DPW3 wing–body–fairing configuration at a Mach number of 0.75, a 0-deg incidence, and a Reynolds number of 5×10^6

core memory were the norm, many of these initial concerns were 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, and the shift to commodity memory and cache-based processors has largely alleviated the memory-footprint bottleneck of early unstructured-mesh implementations. Furthermore, although structured-grid discretizations result naturally in a banded matrix with inherent memory locality, reordering 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 with 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 Fig. 14, in which 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 [85]. When partitioned on 2008 CPUs, the average partition size of this mesh contained only 36,000 vertices, and 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 [86], 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 in the fact that fully grid-converged Reynolds-averaged Navier–Stokes solutions are almost never possible in a production environment, and thus changes in grid resolution will produce nonnegligible 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

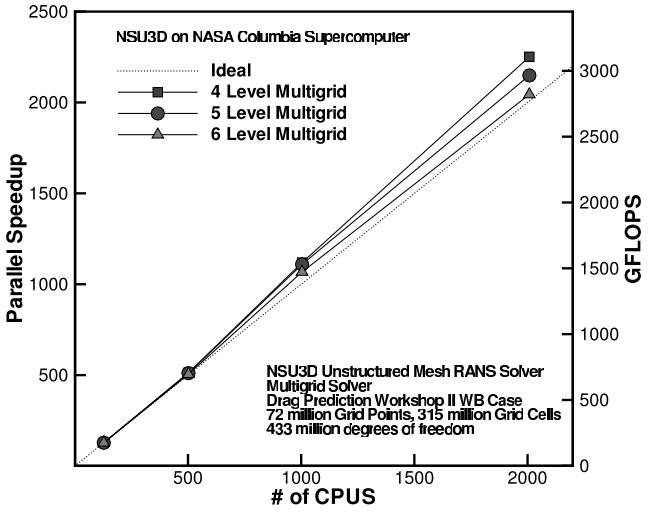
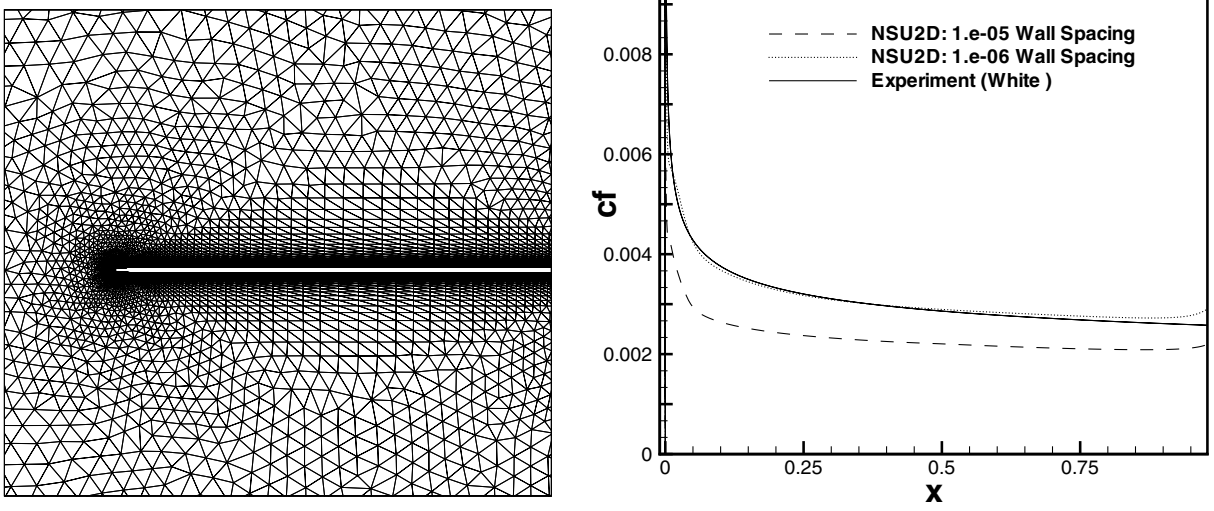


Fig. 14 Scalability of the NSU3D multigrid solver on the NASA Columbia supercomputer for the transonic wing–body configuration using 72 million grid points and using up to 2008 CPUs.

enough or inconsequential. For example, when studying incremental drag differences between two configurations, nonnegligible discretization errors may be manageable if 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 on the order of 10,000. Precise control over the spacings in the mesh boundary-layer regions is also required 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 in the vicinity of $y^+ = 1$. Using a flat-plate turbulent boundary-layer estimate, this distance can easily be calculated as a function of Reynolds number. For Reynolds numbers on the order of 10 million, this requires a wall spacing of approximately 1.e-06 chords, which results in grid aspect ratios on the order of 10,000 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 Reynolds-averaged Navier–Stokes solver [87] was used to compute this flow on the grid shown in Fig. 15a for a Mach number of 0.2 and a plate-based Reynolds number of 10 million, using the Spalart–Allmaras turbulence model [88]. This mesh contains a normal spacing at the plate surface of 1.e-05 plate lengths. Figure 15b depicts the skin friction computed on this mesh compared with experimental data taken from [89], showing substantial underprediction 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.

Although 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 [90], compares the computed-vs-experimental surface pressures for a three-element high-lift configuration, noting poor agreement over the trailing-edge flap upper surface, for which an overprediction of the upper-surface flow separation is apparent. In Fig. 16, the same case was recomputed using a nearly identical mesh, with the exception that the wall spacing of the mesh was reduced by one order of magnitude and the growth rate was kept the same as previously. The agreement with the experiment is now much improved, with a smaller separated region occurring on the flap. The fact that the separated-region overprediction is simply the result of inadequate grid wall spacing is not necessarily evident and might otherwise have been attributed to turbulence-modeling inadequacies.

Figure 17 illustrates typical unstructured-mesh topologies used in transonic aerodynamic calculations. These meshes were generated using the VGRID [42] grid-generation code and were made available as baseline grids for DPW2 in 2003 [29]. 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, to conduct a grid convergence study. The idea of a grid convergence study is to



a) Two-dimensional unstructured grid for flat-plate test case **b) Computed skin friction for flat-plate test case**

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

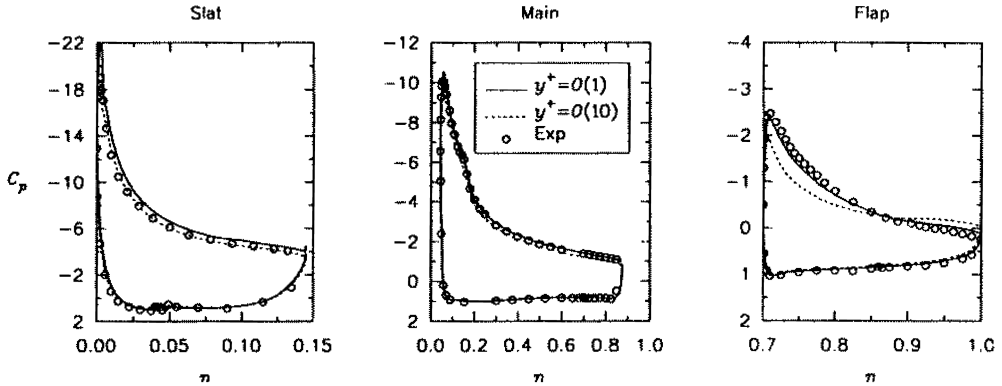
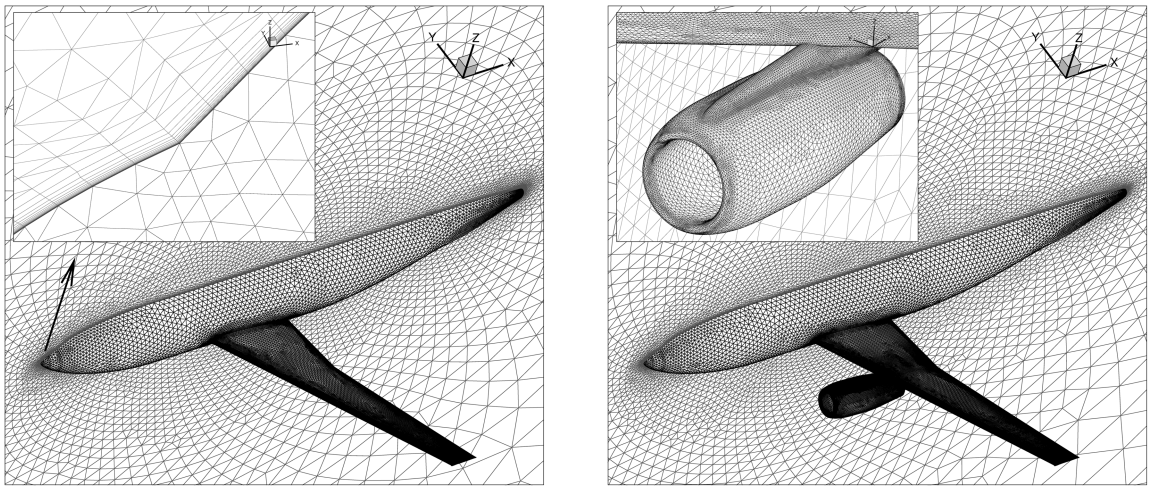


Fig. 16 Effect of normal grid resolution on a high-lift three-element airfoil configuration with a Mach number of 0.2, an incidence of 22 deg, and a Reynolds number of 9×10^6; reproduced from [90].



a) b)

Fig. 17 Illustration of unstructured meshes used for DPW configurations.

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 vs the number N of grid points to the $-2/3$ power [91,92], where $N^{1/3}$ corresponds to some average measure of the grid spacing in three dimensions.

A grid convergence study was performed in [93] 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, 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 was 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 in which the Mach number is 0.75 with a 0-deg incidence and a subcritical case in which the Mach number is 0.5 with a 0-deg incidence. The second condition is included to provide further validation of the grid convergence process for a simpler subsonic case that contains no discontinuities and (possibly) less separation and may thus be expected to more readily display asymptotically second-order-accurate grid convergence.

Figure 7 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 toward 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 nonmonotone behavior for both the transonic and subsonic cases, although the lower dissipation values are more monotone and the sensitivity to dissipation is also reduced on high-resolution meshes. For this family of grids (1, 3, 9, and 72 million points), the solution thus appears to be converging toward an infinite resolution value.

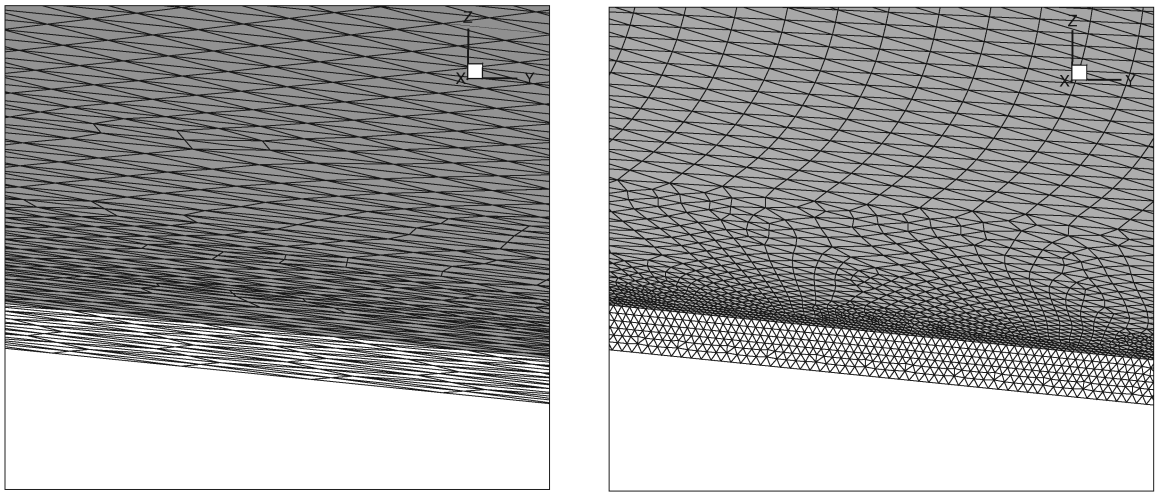
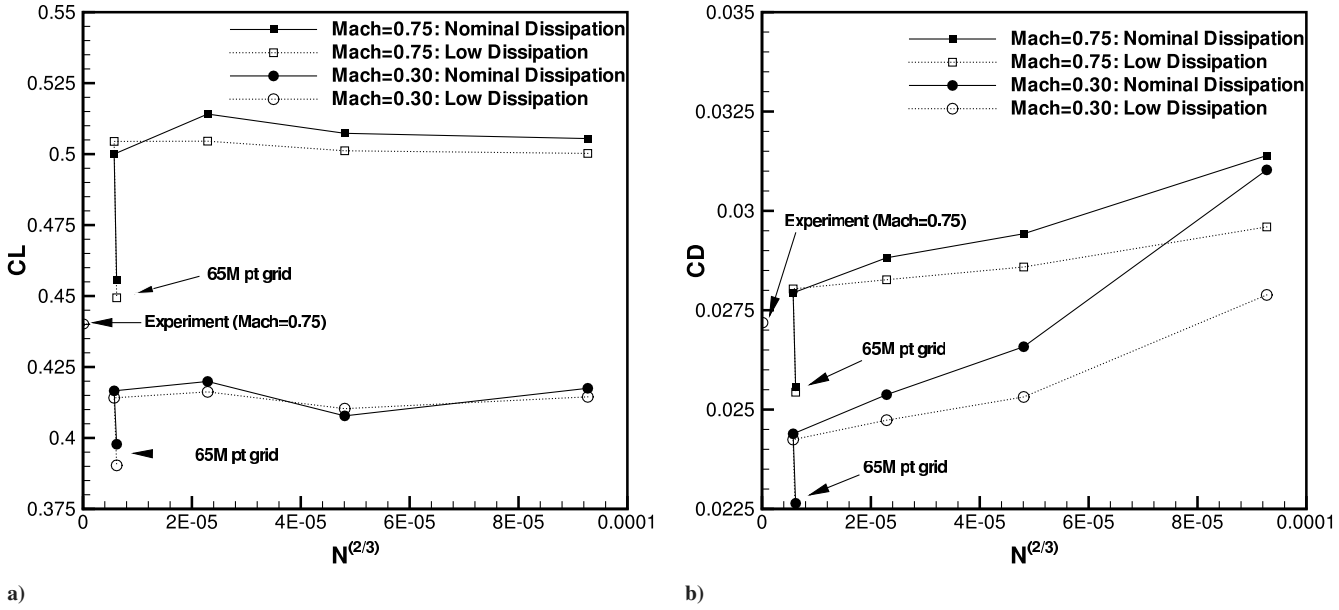


Fig. 18 Comparison between surface resolution for the a) 72-million-point grid and b) 65-million-point grid in the trailing-edge region in the vicinity of the 60% span region illustrating spanwise stretching for the 72-million-point grid.



a) b)

Fig. 19 Comparison of computed a) lift coefficients and b) drag coefficients vs the number of grid points to the $-2/3$ power for transonic ($M = 0.75$) and subsonic ($M = 0.3$) cases at 0-deg incidence, including results computed on the 65-million-point grid.

As part of the study, an alternate fine mesh containing 65 million vertices was generated[†] and made available. The topology of this grid was substantially different from the previous grids in the trailing-edge region of the wing, as depicted in Fig. 18. Although the previous grids all employed spanwise stretching of the elements along the wing 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 65 million points, the obtained solution values were substantially different from any of the values obtained on any of the other grids, as shown by the additional data points in Fig. 19. Note that it is strictly not valid to plot the results from all five grids on the same $N^{-\frac{2}{3}}$ graph, because 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 72-million-point grid, representing a variation that is much larger than that seen between any of the other grids and much larger than any of the computed sensitivities, due to dissipation or viscous-term formulation (cf. Table 4). It remains very surprising and unexpected that there could be such a large difference between two solutions computed on such fine grids (65 million points vs 72 million 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 [94] instead 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, in which the flow is known to be separated, results in the prediction of smaller trailing-edge separation regions, which produce 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

[†]Private communication with S. Pirzadeh, 2005.

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.

In spite of these grid convergence anomalies, this particular case (which was used to highlight the remaining accuracy concerns) represents a particularly sensitive condition that contains relatively important regions of separated flow and that was problematic for structured- and unstructured-mesh solvers. In addition to the three DPW results [28–30], extensive additional grid convergence studies have been performed using a subset of the solvers participating in the workshops [91–93]. When the results of these studies are broken down by grid type [as in Fig. 20, for example, in which the collective results of the third Drag Prediction Workshop (DPW3) are shown], no particular advantage is apparent for one approach over the others. Although 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 that is the 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, and unstructured) and which display remarkable agreement over the full range of conditions and grid resolutions [30]. 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 that need to be addressed [95]. Perhaps it was the early attention paid to transonic flows that 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 tried to outlined the advantages and drawbacks of unstructured-mesh methods compared with 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. 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 are often found to 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 nonaligned 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

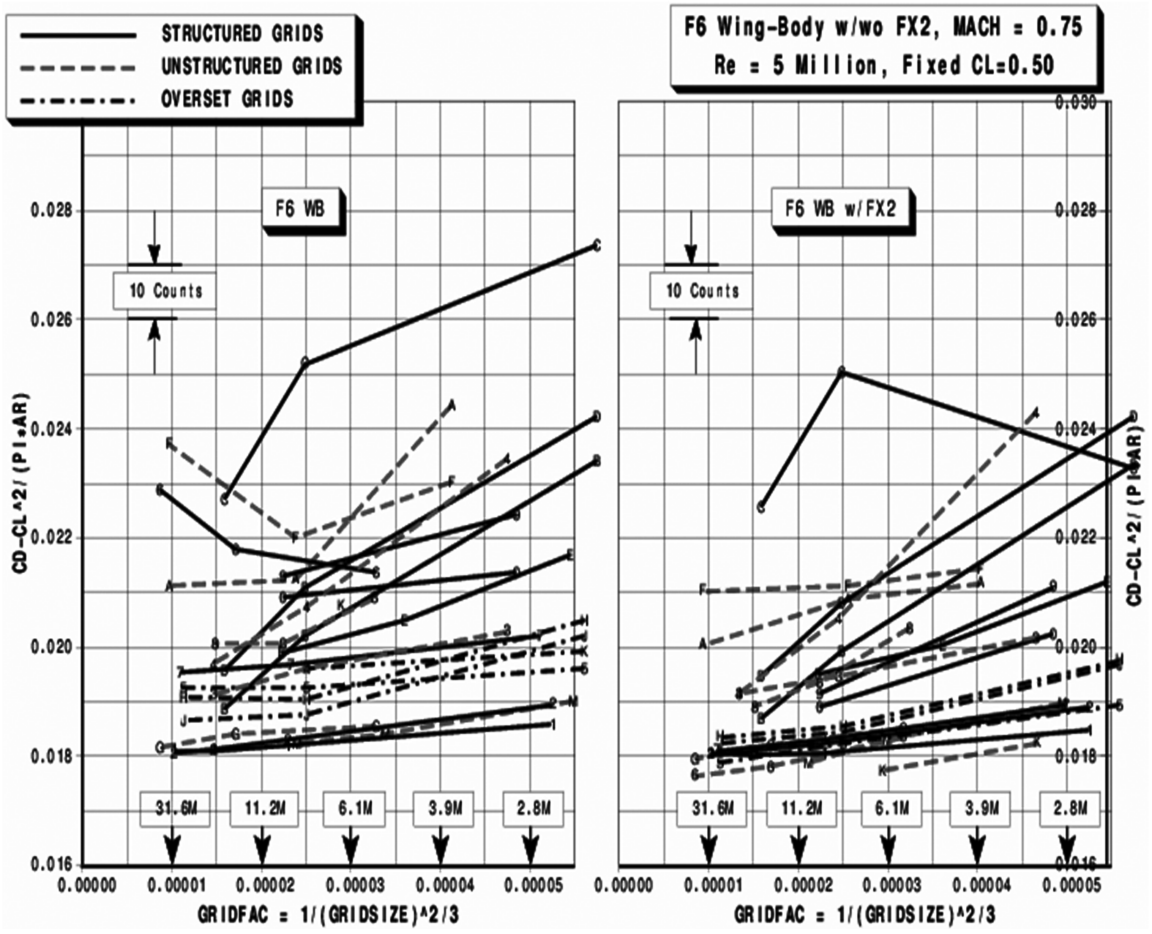


Fig. 20 Collective grid refinement study results for DPW3 broken down by grid type; reproduced from [30].

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 [34] or other approaches such as spectral volume methods [96] 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 [22–25] provides an example of alternate technologies that 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, because the problem of intersecting the geometry boundary is a lower-dimensional problem that has shown greater promise for robust automation than 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] Zienkiewicz, O. C., and Taylor, R. L., *The Finite Element Method for Solid and Structural Mechanics*, 6th ed., Wiley, New York, 2005.

[2] Bristeau, M. O., Pironneau, O., Glowinski, R., Periaux, J., Perrier, P., and Poirier, G., "On the Numerical Solution of Nonlinear Problems in Fluid Dynamics by Least-Squares and Finite Element Methods, 2: Applications to Transonic Flow Simulations," *Computer Methods in Applied Mechanics and Engineering*, Vol. 51, Nos. 1–3, 1985, pp. 363–394.
doi:10.1016/0045-7825(85)90039-8

[3] Jameson, A., and Mavriplis, D. J., "Finite Volume Solution of the Two-Dimensional Euler Equations on a Regular Triangular Mesh," *AIAA Journal*, Vol. 24, No. 4, 1986, pp. 611–618.

[4] Jameson, A., Baker, T. J., and Weatherill, N. P., "Calculation of Inviscid Transonic Flow over a Complete Aircraft," AIAA Paper 86-0103, Jan. 1986.

[5] Desideri, J. A., and Dervieux, A., "Compressible Flow Solvers Using Unstructured Grids," VKI Lecture Series 1995–2002, Vol. 05, Von Kármán Inst. for Fluid Dynamics, Rhode-Saint-Genèse, Belgium, 1988.

[6] Barth, T. J., and Jespersen, D. C., "The Design and Application of Upwind Schemes on Unstructured Meshes," AIAA Paper 89-0366, Jan. 1989.

[7] Mavriplis, D. J., "Solution of the Two-Dimensional Euler Equations on Unstructured Triangular Meshes," Ph.D. Thesis, Mechanical and Aerospace Engineering Dept., Princeton Univ., Princeton, NJ, 1987.

[8] Mavriplis, D. J., "Three-Dimensional Multigrid for the Euler Equations," *AIAA Journal*, Vol. 30, No. 7, July 1992, pp. 1753–1761.

[9] Peraire, J., Peirö, J., and Morgan, K., "A 3D Finite Element Multigrid Solver for the Euler Equations," AIAA Paper 92-0449, Jan. 1992.

[10] Anderson, W. K., "A Grid Generation and Flow Solution Method for the Euler Equations on Unstructured Grids," *Journal of Computational Physics*, Vol. 110, No. 1, Jan. 1994, pp. 23–38.
doi:10.1006/jcph.1994.1003

[11] Gerhold, T., Galle, M., Friedrich, O., and Evans, J., "Calculation of Complex Three-Dimensional Configurations Employing the DLR-Tau Code," AIAA Paper 97-0167, Jan. 1997.

[12] Mavriplis, D. J., "Accurate Multigrid Solution of the Euler Equations on Unstructured and Adaptive Meshes," *AIAA Journal*, Vol. 28, No. 2, 1990, pp. 213–221.

[13] Mavriplis, D. J., "Adaptive Mesh Generation for Viscous Flows Using Delaunay Triangulation," *Journal of Computational Physics*, Vol. 90, No. 2, 1990, pp. 271–291.
doi:10.1016/0021-9991(90)90167-Y

[14] Connell, S. D., and Holmes, D. G., "A 3D Unstructured Adaptive Multigrid Scheme for the Euler Equations," *AIAA Journal*, Vol. 32, No. 8, 1994, pp. 1626–1632.

[15] Löhner, R., and Baum, J. D., "Adaptive H-Refinement on 3-D Unstructured Grids for Transient Problems," *International Journal for Numerical Methods in Fluids*, Vol. 14, No. 12, 1992, pp. 1407–1419.
doi:10.1002/fld.1650141204

[16] Kallinderis, Y., and Vijayan, P., "Adaptive Refinement-Coarsening Scheme for Three-Dimensional Unstructured Meshes," *AIAA Journal*, Vol. 31, No. 8, 1993, pp. 1440–1447.

[17] Trepanier, J. Y., Zhang, H., Reggio, M., and Camarero, R., "Adaptive and Moving Meshes for the Computation of Unsteady Compressible Flows," *Numerical Grid Generation in Computational Fluid Dynamics and Related fields*, edited by A. S. Arcilla, J. Hauser, P. R. Eiseman, and J. F. Thompson, North-Holland, New York, 1991, pp. 43–54.

[18] Warren, G. P., Anderson, W. K., Thomas, J. L., and Krist, S. L., "Grid Convergence for Adaptive Methods," AIAA Paper 91-1592CP, July 1991.

[19] Meakin, R. L., "Moving Body Overset Grid Methods for Complete Aircraft Tiltrotor Simulations," AIAA Paper 93-3350CP, July 1993.

[20] Henshaw, W. D., "Overture: An Object-Oriented Framework for Overlapping Grid Applications," AIAA Paper 2002-3189, June 2002.

[21] Sclafani, A., Vassberg, J., Harrison, N., Rumsey, C., DeHaan, M., and Morrison, J., "Drag Prediction for the DLR-F6 Wing/Body and DPW Wing Configurations Using CFL3D and Overflow on an Overset Mesh," AIAA Paper 2007-0257, Jan. 2007.

[22] Aftosmis, M., Berger, M., and Melton, J., "Robust and Efficient Cartesian Mesh Generation for Component-Based Geometry," AIAA Paper 97-0196, Jan. 1997.

[23] Murman, S., Aftosmis, M., and Berger, M., "Numerical Simulation of Rolling Airframes Using a Multilevel Cartesian Method," AIAA Paper 2002-2798, June 2002.

[24] Tu, S. Z., and Ruffin, S. M., "Solution Adaptive, Unstructured Cartesian-Grid Methodology for Chemically Reacting Flow," AIAA Paper 2002-3097, June 2002.

[25] Fidkowski, K., and Darmofal, D., "An Adaptive Simplex Cut-Cell Method for Discontinuous Galerkin Discretizations of the Navier–Stokes Equations," AIAA Paper 2007-3941, June 2007.

[26] Mavriplis, D. J., "Aerodynamic Drag Prediction Using Unstructured Mesh Solvers," *CFD-Based Drag Prediction and Reduction*, edited by H. Deconoinck, K. Sermus, and C. van Dam, VKI Lecture Notes, Von Kármán Inst. for Fluid Dynamics, Rhode-Saint-Genèse, Belgium, Mar. 2003.

[27] Levy, D. W., and Thacker, M. D., "Comparison of Unstructured Cell- and Node-Based Schemes for the Euler Equations," AIAA Paper 99-3185, June 1999.

[28] Levy, D. W., Zickuhr, T., Vassberg, J., Agrawal, S., Wahls, R. A., Pirzadeh, S., and Hemsch, M. J., "Summary of Data from the First AIAA CFD Drag Prediction Workshop," AIAA Paper 2002-0841, Jan. 2002.

[29] Laflin, K., Brodersen, O., Rakowitz, M., Vassberg, J., Wahls, R., and Morrison, J., "Summary of Data from the Second AIAA CFD Drag Prediction Workshop," AIAA Paper 2004-0555, Jan. 2004.

[30] Vassberg, J. C., Tinoco, E. N., Mani, M., Brodersen, O. P., Eisfeld, B., Wahls, R. A., Morrison, J. H., Zickuhr, T., Laflin, K. R., and Mavriplis, D. J., "Summary of the Third AIAA CFD Drag Prediction Workshop," AIAA Paper 2007-0260, Jan. 2007.

[31] Pirzadeh, S. Z., and Frink, N. T., "Assessment of the Unstructured Grid Software TetrUSS for Drag Prediction of the DLR-F4 Configuration," AIAA Paper 2002-0839, Jan. 2002.

[32] Mavriplis, D. J., and Levy, D. W., "Transonic Drag Prediction Using an Unstructured Multigrid Solver," *Journal of Aircraft*, Vol. 42, No. 4, 2003, pp. 887–893.

[33] Hughes, T. J. R., "Recent Progress in the Development and Understanding of SUPG Methods with Special Reference to the Compressible Euler and Navier–Stokes Equations," *International Journal for Numerical Methods in Fluids*, Vol. 7, No. 11, 1987, pp. 1261–1275.
doi:10.1002/fld.1650071108

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

[35] Aftosmis, M., Gaitonde, D., and Tavares, T. S., "On the Accuracy, Stability and monotonicity of Various Reconstruction Algorithms for Unstructured Meshes," AIAA Paper 94-0415, Jan. 1994.

[36] Barth, T. J., "Parallel CFD Algorithms on Unstructured Meshes," AGARD Rept. 807, May 1995, Neuilly sur Seine, France, pp. 7-1–7-41.

[37] Luo, H., Baum, J. D., Löhner, R., and Cabello, J., "Adaptive Edge-Based Finite Element Schemes for the Euler and Navier–Stokes

Equations on Unstructured Grids," AIAA Paper 93-0336, Jan. 1993.
[38] Sidilkover, D., "A Genuinely Multidimensional Upwind Scheme and Efficient Multigrid Solver for the Compressible Euler Equations," Inst. for Computer Applications in Science and Engineering, Rept. 94-84, Hampton, VA, Oct. 1994.
[39] Deconinck, H., Paillere, H., Struijs, R., and Roe, P. L., "Multidimensional Upwind Schemes Based on Fluctuation-Splitting of Conservation Laws," *Computational Mechanics*, Vol. 11, Nos. 5–6, 1993, pp. 323–340. doi:10.1007/BF00350091
[40] Nakahashi, K., "A Finite Element Method on Prismatic Elements for the Three-Dimensional Navier–Stokes Equations," *Lecture Notes in Physics*, Vol. 323, Springer–Verlag, Berlin, 1989.
[41] Kallinderis, Y., and Ward, S., "Hybrid Prismatic/Tetrahedral Grid Generation for Complex Geometries," AIAA Paper 93-0669, Jan. 1993.
[42] Pirzadeh, S., "Three-Dimensional Unstructured Viscous Grids by the Advancing-Layers Method," *AIAA Journal*, Vol. 34, No. 1, 1996, pp. 43–49.
[43] Mavriplis, D. J., "Unstructured Mesh Generation and Adaptivity," VKI Lecture Series 1995–2002, Von Kármán Inst. for Fluid Dynamics, Rhode-Saint-Genèse, Belgium, Mar. 1995.
[44] Trepanier, M. J. Y., Reggio, M., and Camarero, R., "A Conservative Dynamic Discontinuity Tracking Algorithm for the Euler Equations," AIAA Paper 94-0081, Jan. 1994.
[45] Van Rosendale, J., "Floating Shock Fitting via Lagrangian Adaptive Meshes," 12th AIAA CFD Conference, San Diego, CA, AIAA Paper 95-1721-CP, June 1995.
[46] Roe, P. L., "Characteristic-based schemes for the Euler equations," *Annual Review of Fluid Mechanics*, Vol. 18, 1986, pp. 337–365.
[47] Jameson, A., "Analysis and Design of Numerical Schemes for Gas Dynamics, 1: Artificial Diffusion, Upwind Biasing, Limiters and Their Effect on Multigrid Convergence," *International Journal of Computational Fluid Dynamics*, Vol. 4, 1995, pp. 171–218. doi:10.1080/10618569508904524
[48] Barth, T. J., "Numerical Aspects of Computing Viscous High Reynolds Number Flows on Unstructured Meshes," AIAA Paper 91-0721, Jan. 1991.
[49] Anderson, W. K., and Bonhaus, D. L., "An Implicit Upwind Algorithm for Computing Turbulent Flows on Unstructured Grids," *Computers and Fluids*, Vol. 23, No. 1, 1994, pp. 1–21. doi:10.1016/0045-7930(94)90023-X
[50] Haselbacher, A., and Blazek, J., "Accurate and Efficient Discretization of Navier–Stokes Equations on Mixed Grids," *AIAA Journal*, Vol. 38, No. 11, 2000, pp. 2094–2102.
[51] Mavriplis, D. J., "Revisiting the Least-Squares Procedure for Gradient Reconstruction on Unstructured Meshes," AIAA Paper 2003-3986, June 2003.
[52] Petrovskaya, N. B., "The Impact of Grid Cell Geometry on the Least-Squares Gradient Reconstruction," Keldysh Inst. of Applied Mathematics, Russian Academy of Sciences, TR 21, Moscow, Apr. 2003.
[53] Venkatakrishnan, V., "Convergence to Steady State Solutions of the Euler Equations on Unstructured Grids with Limiters," *Journal of Computational Physics*, Vol. 118, No. 1, Apr. 1995, pp. 120–130. doi:10.1006/jcph.1995.1084
[54] Harten, A., Lax, P. D., and Van Leer, B., "On Upstream Differencing and Godunov-Type Schemes for Hyperbolic Conservation Laws," *SIAM Review*, Vol. 25, No. 1, 1983, pp. 35–61. doi:10.1137/1025002
[55] Roe, P. L., "Approximate Riemann Solvers, Parameter Vectors and Difference Schemes," *Journal of Computational Physics*, Vol. 43, No. 2, 1981, pp. 357–372. doi:10.1016/0021-9991(81)90128-5
[56] Harten, A., "High Resolution Schemes for Hyperbolic Conservation Laws," *Journal of Computational Physics*, Vol. 135, No. 2, 1997, pp. 260–278. doi:10.1006/jcph.1997.5713
[57] Anderson, W. K., "A Grid Generation and Flow Solution Method for the Euler Equations on Unstructured Grids," *Journal of Computational Physics*, Vol. 110, No. 1, 1994, pp. 23–38. doi:10.1006/jcph.1994.1003
[58] Babuska, I., and Aziz, A. K., "On the Angle Condition in the Finite Element Method," *SIAM Journal on Numerical Analysis*, Vol. 13, No. 6, 1976.
[59] Haselbacher, A., McGuirk, J., and Page, G., "Finite Volume Discretization Aspects for Viscous Flows on Unstructured Meshes," *AIAA Journal*, Vol. 37, No. 2, 1999, pp. 177–184.
[60] Lohner, R., Luo, H., and Baum, J. D., "Selective Edge Removal for Unstructured Grids with Cartesian Cores," *Journal of Computational Physics*, Vol. 206, No. 1, 2005, pp. 208–226. doi:10.1016/j.jcp.2004.11.034
[61] Connell, S. D., and Braaten, M. E., "Semi-Structured Mesh Generation for Three-Dimensional Navier–Stokes Calculations," *AIAA Journal*, Vol. 33, No. 6, June 1995, pp. 1017–1024.
[62] Lohner, R., "Matching Semi-Structured and Unstructured Grids for Navier–Stokes Calculations," AIAA paper 93-3348, July 1993.
[63] Mavriplis, D. J., "A Three-Dimensional Multigrid Reynolds-Averaged Navier–Stokes Solver for Unstructured Meshes," *AIAA Journal*, Vol. 33, No. 3, Mar. 1995, pp. 445–4531.
[64] Xu, K., Kim, C. A., Martinelli, L., and Jameson, A., "BGK-Based Schemes for the Simulation of Compressible Flow," *International Journal of Computational Fluid Dynamics*, Vol. 7, 1996, pp. 213–235. doi:10.1080/10618569608940763
[65] May, G., and Jameson, A., "Improved Gaskinetic Multigrid Method for Three-Dimensional Computation of Viscous Flow," AIAA Paper 2005-5106, June 2005.
[66] Frink, N. T., "Upwind Scheme for Solving the Euler Equations on Unstructured Tetrahedral Meshes," *AIAA Journal*, Vol. 30, No. 1, 1992, pp. 70–77.
[67] Venkatakrishnan, V., and Barth, T. J., "Application of Direct Solvers to Unstructured Meshes for the Euler and Navier Stokes Equations Using Upwind Schemes," AIAA Paper 89-0364, Jan. 1989.
[68] Batina, J. T., "Implicit Upwind Solution Algorithms for Three-Dimensional Unstructured Meshes," *AIAA Journal*, Vol. 31, No. 5, May 1993, pp. 801–805.
[69] Thareja, R. R., Stewart, J. R., Hassan, O., Morgan, K., and Peraire, J., "A Point Implicit Unstructured Grid Solver for the Euler and Navier–Stokes Equations," AIAA Paper 88-0036, Jan. 1988.
[70] Venkatakrishnan, V., and Mavriplis. D. J., "Implicit Method for the Computation of Unsteady Flows on Unstructured Grids," 12th AIAA CFD Conference, San Diego, CA, AIAA Paper 95-1705-CP, June 1995.
[71] Guillard, H., "Node Nested Multigrid with Delaunay Coarsening," Inst. National de Recherche en Informatique et en Automatique, Rept. 1898, Rocquencourt, France, 1993.
[72] Braaten, M. E., and Connell, S. D., "Three Dimensional Unstructured Adaptive Multigrid Scheme for the Navier–Stokes Equations," *AIAA Journal*, Vol. 34, No. 2, Feb. 1996, pp. 281–290.
[73] Hutchinson, B. R., Galpin, P. F., and Raithby, G. D., "Application of Additive Correction Multigrid to the Coupled Fluid Flow Equations," *Numerical Heat Transfer*, Vol. 13, No. 2, 1988, pp. 133–147. doi:10.1080/10407788808913608
[74] Smith, W. A., "Multigrid Solution of Transonic flow on Unstructured Grids," *Recent Advances and Applications in Computational Fluid Dynamics*, edited by O. Baysal, American Society of Mechanical Engineers, New York, Nov. 1990.
[75] Lallemand, M., Steve, H., and Dervieux, A., "Unstructured Multigridding by Volume Agglomeration: Current Status," *Computers and Fluids*, Vol. 21, No. 3, 1992, pp. 397–433. doi:10.1016/0045-7930(92)90047-Y
[76] Venkatakrishnan, V., and Mavriplis, D. J., "Agglomeration Multigrid for the Three-Dimensional Euler Equations," *AIAA Journal*, Vol. 33, No. 4, Apr. 1995, pp. 633–640.
[77] Mavriplis, D. J., and Venkatakrishnan, V., "A Unified Multigrid Solver for the Navier Stokes Equations on Mixed Element Meshes," *International Journal of Computational Fluid Dynamics*, Vol. 8, 1997, pp. 247–263. doi:10.1080/10618569708940807
[78] Tuminaro, R. S., and Tong, C., "Parallel Smoothed Aggregation Multigrid: Aggregation Strategies on Massively Parallel Machines," *Proceedings of the 2000 ACM/IEEE Conference on Supercomputing*, IEEE Computer Society, Washington, D.C., Nov. 2000.
[79] Martin, D., and Löhner, "An Implicit Linelet-Based Solver for Incompressible Flows," AIAA Paper 92-0668, Jan. 1992.
[80] Mavriplis, D. J., "Multigrid Strategies for Viscous Flow Solvers on Anisotropic Unstructured Meshes," *Journal of Computational Physics*, Vol. 145, No. 1, Sept. 1998, pp. 141–165. doi:10.1006/jcph.1998.6036
[81] Mavriplis, D. J., "Directional Agglomeration Multigrid Techniques for High-Reynolds-Number Viscous Flow Solvers," *AIAA Journal*, Vol. 37, No. 10, Oct. 1999, pp. 1222–1230.
[82] Nielsen, E. J., Lu, J., Park, M. A., and Darmofal, D. L., "An Implicit Exact Dual Adjoint Solution Method for Turbulent Flows on Unstructured Grids," *Computers and Fluids*, Vol. 33, No. 9, Nov. 2004, pp. 1131–1155. doi:10.1016/j.compfluid.2003.09.005
[83] Nompelis, I., Drayna, T., and Candler, G., "A Parallel Unstructured

Implicit Solver for Hypersonic Reacting Flow Simulation," AIAA Paper 2005-4867, June 2005.
[84] Balay, S., Buschelman, K., Eijkhout, V., Gropp, W. D., Kaushik, D., Knepley, M. G., McInnes, L. C., Smith, B. F., and Zhang, H., "PETSc User's Manual," Argonne National Lab., Rept. ANL-95/11, Rev. 2.1.5, Argonne, IL, 2004.
[85] Mavriplis, D. J., Aftosmis, M., and Berger, M., "High-Resolution Aerospace Applications Using the NASA Columbia Supercomputer," *The International Journal of High Performance Computing Applications*, Vol. 21, No. 1, 2007, pp. 106–126. doi:10.1177/1094342006074872.
[86] Karypis, G., and Kumar, V., "A Fast and High Quality Multilevel Scheme for Partitioning Irregular Graphs," *SIAM Journal on Scientific Computing*, Vol. 20, No. 1, 1998, pp. 359–392. doi:10.1137/S1064827595287997
[87] Mavriplis, D. J., *A CFD Package for Multi-Element Airfoil High-Lift Analysis: NSU2D User's Manual*, Rev. 4.0, Scientific Simulations, LLC, Laramie, WY, Dec. 1996.
[88] Spalart, P. R., and Allmaras, S. R., "A One-Equation Turbulence Model for Aerodynamic Flows," *La Recherche Aérospatiale*, Vol. 1, 1994, pp. 5–21.
[89] White, F. M., *Viscous Fluid Flow*, McGraw–Hill, New York, 1991.
[90] Anderson, W. K., Bonhaus, D. L., McGhee, R. J., and Walker, B. S., "Navier–Stokes Computations and Experimental Comparisons for Multi-Element Airfoil Configurations," *Journal of Aircraft*, Vol. 32, No. 6, 1995, pp. 1246–1253.
[91] Lee-Rausch, E. M., Buning, P. B., Mavriplis, D. J., Morrison, J. H., Park, M. A., Rivers, S. M., and Rumsey, C. L., "CFD Sensitivity Analysis of a Drag Prediction Workshop Wing/Body Transport Configuration," AIAA Paper 2003-3400, June 2003.
[92] Lee-Rausch, E. M., Frink, N. T., Mavriplis, D. J., Rausch, R. D., and Milholen, W. E., "Transonic Drag Prediction on a DLR-F6 Transport Configuration Using Unstructured Grid Solvers," AIAA Paper 2004-0554, Jan. 2004.
[93] Mavriplis, D. J., "Grid Resolution Study of a Drag Prediction Workshop Configuration Using the NSU3D Unstructured Mesh Solver," AIAA Paper 2005-4729, June 2005.
[94] Lee-Rausch, E. M., Park, M., Nielsen, E., Jones, W., and Hammond, D., "Parallel Adjoint-Based Error Estimation and Anisotropic Grid Adaptation for Three-Dimensional Aerospace Applications," AIAA Paper 2005-4842, June 2005.
[95] Gnoffo, P., and White, J. A., "Computational Aerothermodynamic Simulation Issues on Unstructured Grids," AIAA Paper 2004-2371, June 2004.
[96] Wang, Z. J., "Spectral (Finite) Volume Method for Conservation Laws on Unstructured Grids. Basic Formulation: Basic Formulation," *Journal of Computational Physics*, Vol. 178, No. 1, May 2002, pp. 210–251. doi:10.1006/jcph.2002.7041

E. Oran
Editor-in-Chief