Aerodynamic Drag Prediction using Unstructured Mesh Solvers

Dimitri J. Mavriplis

mavriplis@nianet.org

Research Fellow
National Institute of Aerospace
144 Research Drive
Hampton, VA 23666
United States

Contents

1 Introduction

The ability to accurately predict aerodynamic forces and moments is of the utmost importance to the aircraft designer. Whether this is achieved through flight testing, wind-tunnel experiments, or computational techniques, the prediction of lift, drag, and moments, and their changes as a function of design parameters is critical for reducing risk in the conception stage and optimizing performance of the final product. For example, for long range transport aircraft, small changes in cruise drag can have appreciable overall effects on direct operating costs and mission range. Similarly, in the design of high-lift systems, simplified mechanical designs which maintain of exceed required levels of aerodynamic performance can lead to substantially reduced manufacturing and maintenance costs. For these reasons, it is essential to be able to predict forces and moments in general, and drag in particular, very accurately throughout the design process.

For subsonic transport aircraft, accuracy requirements for drag prediction through wind tunnel experiments are generally taken to be of the order of 1 drag count [1, 2]. This is the level of accuracy and reliability which must be achieved from computational techniques, if computational fluid dynamics (CFD) is to fulfill the promise of reduced design cycle costs, and improved optimization procedures, by gradually displacing wind tunnel testing and automating the design optimization process. This level of accuracy (1 drag count = 1.e-04) is exceptionally stringent, and distinguishes the special subfield of computational aerodynamics from the larger field of computational fluid dynamics or computational physics.

Because of these requirements, it is unlikely that other generic or specialized computation simulation capabilities (such as techniques for the simulation of automotive flows or explosive weapons flows, two disparate example areas of computational fluid dynamics which have received much attention over the last decade) will be viable for aerodynamic drag prediction. Computational aerodynamics involves not only stringent accuracy requirements, but includes unique characteristics, such as very thin boundary layer and wake regions, and turbulence and transitional effects. The former can only be resolved efficiently through the use of highly stretched meshes, which can have adverse effects on solution efficiency if not dealt with explicitly in the solver formulation, while the latter requires the use of physical models specialized for aerodynamic applications. Because of these unique characteristics, and because aerodynamics is considered a core competency of most airframers, virtually all aircraft companies worldwide today maintain some degree of in-house expertise in computational aerodynamics. While the actual simulation capability may either be developed in-house or brought in from the outside, the validation and certification of this capability must be performed in-house and retained as corporate knowledge. In this respect, it is not the absolute performance of a simulation capability which is of value, but the knowledge of the extent, limitations, and region of applicability of the simulation capability. In general, a suite of simulation codes is assembled, with low fidelity models such as panel methods being used for rapid analyses, providing intuitive or empirically correlated feedback, and higher fidelity models such as inviscid flow coupled with boundary layers or Reynolds averaged Navier-Stokes methods providing either incremental information, or ultimately, absolute predictions.

The field of computational fluid dynamics has made great strides over the last 20 years through combined hardware advances and algorithmic improvements. Over this period of

time, the scientific literature has concentrated mostly on documenting algorithmic improvements. Demonstrations on relatively simple test problems and qualitative assessments have generally been the outcome of such studies. However, detailed validation studies, particularly for drag prediction, seldomly have been reported. Now that computational aerodynamics has become affordable to most industry organizations, it must earn credibility through detailed validation and certification studies in order to provide value in the design process. This fact has been recognized over the last several years, and various activities have taken place or are under way both in the United States and in Europe, with the undertaking of programs such as the AIAA Drag Prediction Workshops in June 2001 and June 2003, and the European AIRDATA and HiReTT programs, ERCOFTAC validation activities, and the formation of the QNET CFD project (see [3] for an overview of European activities). It is in this spirit that the current VKI lecture series on Drag Prediction has been organized.

It is generally acknowledged that for cruise conditions, with minimal regions of separated flow, Reynolds-averaged Navier-Stokes methods can provide reliable predictions of surface pressures. However, this in itself does not imply accurate drag prediction. One study found that accurate drag prediction required much finer grids and lower levels of convergence than previously used for surface pressure prediction [4, 5], resulting in an order of magnitude more computational expense. Furthermore, the ability to predict Reynolds number effects, in order to enable reliable scaling from wind-tunnel to flight, is a current area of research on both sides of the Atlantic [6, 7]. For off-design conditions, such as post-stall buffet, and high-lift conditions, due to the increased geometric complexity and more complicated flow physics, the reliable prediction of surface pressures in itself currently presents a serious challenge.

The difficulty of the task of assessing the state-of-the-art in computational drag prediction is compounded by the multitude of Navier-Stokes solver formulations available, as well as the effects on accuracy of turbulence and transition modeling, and grid resolution. For the CFD practitioner, the important issues are less related to the knowledge of the algorithmic details of a given approach, but more related to understanding the interplay between grid resolution, discretization, physical modeling, and the manners in which these characteristics can affect the accuracy and reliability of drag prediction. It is with this purpose in mind that these lecture notes have been written. While a basic level description of the algorithms comprising current-day unstructured grid RANS solvers will be given, emphasis is directed towards the effect various characteristics of current-day grid generation and solver technologies can have on accurate drag prediction, and the types of numerical experiments required to validate these findings.

These lecture notes are concerned exclusively with the use of unstructured mesh solvers for aerodynamic drag prediction. Unlike many other fields of computational physics, computational aerodynamics has been dominated for most of its existence by the use of structured mesh techniques, which have been extended to block-structured and overlapping structured methods for dealing with ever increasingly complex geometries. This has perhaps been a result of the high cost and accuracy requirements of computational aerodynamics from the outset. Over the last decade, unstructured mesh methods have gained more acceptance in this field, as they have demonstrated increasingly competitive accuracy and efficiency compared to their structured grid counterparts. Currently, it can be said that for similar grid

resolutions and equivalent discretizations, structured and unstructured grid formulations can deliver equivalent levels of accuracy on simple problems. However, unstructured grid solvers can require up to two to three times more memory and cpu time than structured grid solvers for equivalent problems, due to the extra overhead involved in indirect addressing, which is required to describe an unstructured mesh. On the other hand, these differences are reduced by the extra overhead inherent in block-structured and overset-structured techniques, required for more complex geometries, and by often superior parallel scalability achieved by unstructured mesh methods due to the homogeneous nature of the data structures involved. In fact, it is often the case that efficient state-of-the-art unstructured grid solvers are more computationally efficient than many well established block- or overset-structured grid solvers. The main advantages of the unstructured grid approach include substantial reduction of the grid generation time required for complicated geometries, and a relatively straight-forward implementation of adaptive meshing techniques, which can be used to optimize grid resolution while providing a bound on the level of discretization error. Unfortunately, adaptive meshing techniques have seldom achieved this level of automation and reliability, but this potential remains to be exploited in the future.

In the following lecture notes, issues related to unstructured mesh CFD drag prediction are discussed in detail. This work concentrates exclusively on aspects of drag prediction which are specific to unstructured mesh methodologies, choosing to relegate to other works, issues which are not specifically related to unstructured mesh approaches, such as turbulence and transition modeling, or far-field drag integration methods. In section 2, the relative capabilities and costs of various model fidelities are discussed, including inviscid (Euler) solvers, Euler with boundary layer corrections, Reynolds-averaged Navier-Stokes (RANS) methods, and large eddy simulations (LES) on unstructured grids. Section 3 considers discretization and grid resolution issues, which are naturally inter-related. An examination of various unstructured mesh discretizations, including some of the subtleties involved and their effect on drag prediction is given in section 3.1. Grid resolution, which represents perhaps the most important issue in achieving accurate drag prediction is covered in section 3.2, where its effect on accuracy is examined, and guidelines for adequate grid resolution are formulated.

In section 4, the principal techniques involved in the construction of an efficient parallel unstructured grid solver are overviewed. Various drag prediction studies are outlined in section 5, detailing the successes and failures of current day methods at predicting absolute cruise drag, engine installation drag, and lift and drag for high-lift systems.

Finally, in section 6, the potential impact of existing and new technologies on the drag prediction capability are discussed, including adaptive meshing, and higher-order methods.

2 Physical Model Fidelity

Although current state-of-the-art aerodynamic drag prediction methods are centered on Reynolds-Averaged Navier-Stokes (RANS) methods, it is useful to examine the capabilities and relative merits of other models and to describe how these may complement current RANS models. The aerodynamicist has at his disposal an entire suite of analysis tools, ranging from simple panel methods, up to full Navier-Stokes simulations. Unstructured mesh-based simulation methods may range from simple inviscid flow analyses, to complex large-eddy simulations (LES), with RANS methods occupying the center ground. The various levels of approximation possible for unstructured mesh methods are depicted in Table 1 along with their relative cost.

Table 1: *Unstructured mesh based physical models and required computational resources*

Physical Model	Nominal Grid Size (Pts)	(MG) Cycles for Convergence	Normalized Run Time
Euler	$10^5 - 10^6$	50 - 100	1.0
Euler + IBL	$10^5 - 10^6$	100 - 300	2 - 3
RANS	$10^6 - 10^7$	250 - 1000	50 - 100
DES	$10^7 - 10^8$	5000 - 10000	5000 - 10000
LES	$10^9 - 10^{11}$	$O(10^6)$	$O(10^8)$

2.1 Inviscid Flow Analysis

The technology for computing inviscid flows on unstructured meshes is currently well developed. Several unstructured and adaptive cartesian mesh approaches have demonstrated rapid turn-around and good utility for various aerodynamic problems, including important applications such as missile aerodynamics, where viscous effects are secondary [8]. The Tranair code developed by Boeing [9], based on potential flow with additional boundary-layer corrections, has been used successfully for drag prediction of transport aircraft for over one decade. Tetrahedral unstructured mesh Euler solvers have been available for over 15 years [10, 11, 12, 13], and have evolved into very efficient, robust, and scalable solution methodologies.

The solution of the Euler equations avoids many of the difficulties associated with RANS calculations. In particular, since boundary layers and wakes are not present, grid stretching is not required, and isotropic tetrahedral meshes can be employed. The generation of isotropic tetrahedral meshes is nowadays a well established robust process, for which many efficient commercial packages exist [14]. Since boundary-layer regions need not be resolved, overall mesh resolution can be be considerably lower than that required in RANS calculations, and rapid solver convergence is generally attainable in the absence of appreciable grid stretching [12, 13]. For these reasons, unstructured mesh Euler solvers generally require 50 to 100 times less computational resources than equivalent unstructured mesh RANS solvers. As an example, consider the case of transonic flow analysis over the DLR-F4 wing body configuration shown in Figure 1.

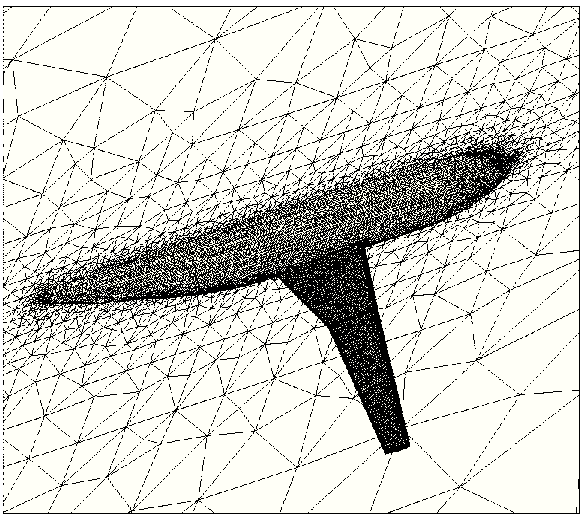

Figure 1: *Unstructured mesh for inviscid flow computations about DLR-F4 configuration. Number of vertices = 250,000*

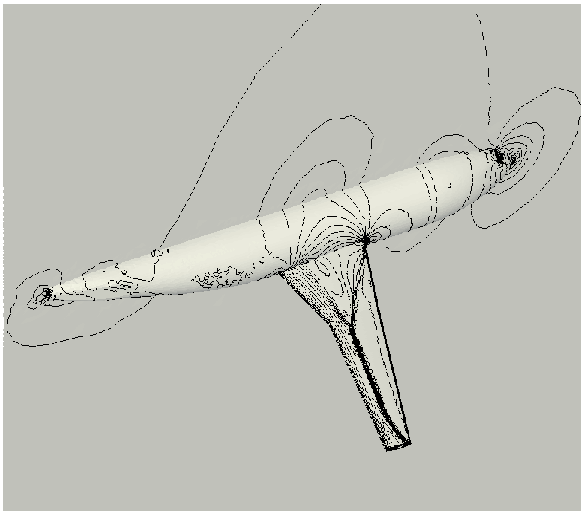

Figure 2: *Computed inviscid flow pressure contours about DLR-F4 configuration. Mach=0.75, Incidence=*1^o

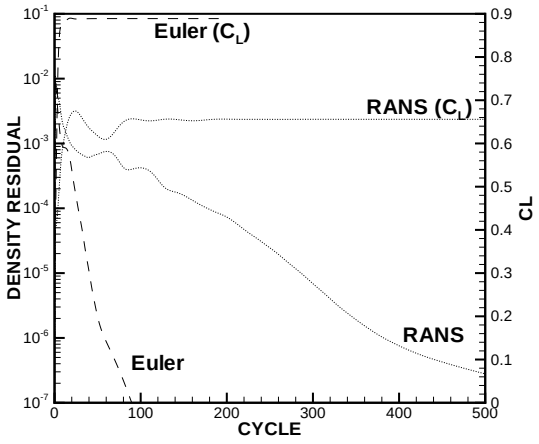

Figure 3: *Comparison of convergence history for inviscid flow and viscous flow simulations in terms of density residual and lift coefficient versus number of multigrid cycles*

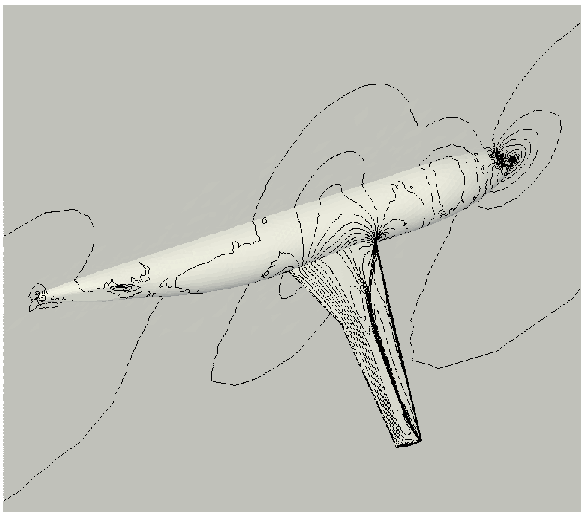

Figure 4: *Computed viscous flow pressure contours about DLR-F4 configuration using NSU3D solver on mesh of Figure 50. Mach=0.75, Incidence=*1^o*, Reynolds number = 3 million*

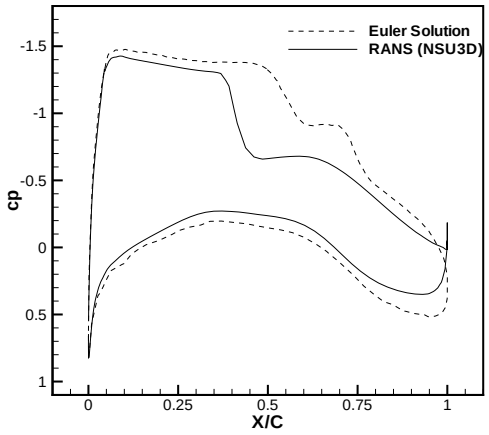

Figure 5: *Comparison of computed surface pressure coefficients at 40.9% span location for inviscid flow and viscous flow computations*

This geometry formed the basis of the 1st AIAA Drag Prediction workshop in June 2001 [15]. The minimum grid size that was found to provide reliable drag values using an unstructured RANS simulation capability contained 1.65 million points, with spacing at the wall of 7.e-06 chords, resulting in aspect ratios of 10,000 near the wall. On the other hand, a suitable Euler solution on this geometry can be achieved using the same solver with 235,000 mesh points, involving exclusively isotropic mesh elements throughout the domain, as shown in Figures 1 and 2. Convergence of the unstructured Euler solver for this case is shown as a function of multigrid cycles in Figure 3, where it is compared with the convergence history of the equivalent RANS solver, illustrating the convergence penalty incurred by the RANS solver due to grid anisotropy. Converging the drag values to 5 significant figures requires a total of 3 minutes on 8 Pentium IV 1700 GHz processors for the inviscid flow simulation (50 cycles, 3.5 seconds per cycle), while the viscous flow simulation required 75 minutes on 16 processors, (250 multigrid cycles, 18 seconds per cycle) or 50 times more resources.

Unfortunately, for transport aircraft in transonic cruise conditions, inviscid flow simulations are of little use in predicting lift and drag values, due to the exclusion of all viscous effects. For example, large differences between the RANS and Euler results can be seen in the contour plots of Figures 2 and 4. Such differences are particularly notable for modern-day supercritical airfoil sections, where the shock position is greatly influenced by the boundary layer thickness, as can be seen be the overprediction of shock strength and aft location in the inviscid analysis as compared to the RANS calculation in Figure 5.

2.2 Coupled Inviscid-Viscous Methods

One approach for overcoming these deficiencies, while maintaining the efficiency attributes of the inviscid flow approximation, consists of coupling the unstructured mesh Euler solver with a boundary-layer solution technique, which can be used to model the displacement thickness

of the boundary layer, and its influence on shock strength and location. This approach has been used successfully on unstructured grids by various practitioners [16, 17], as well as by the current author. Using a two-dimensional boundary-layer solution approach, surface pressure data is gathered along chordwise strips at various spanwise stations, and used to compute a boundary-layer displacement thickness along each strip, which is then coupled back into the Euler solution as a transpiration condition. In the context of unstructured meshes, the principal difficulty resides in interpolating pressures from the unstructured surface mesh onto the strips, and redistributing the resulting transpiration velocities calculated at the boundary-layer strips back onto the global surface unstructured mesh. The former is achieved by calculating the intersection of all mesh edges with the prescribed boundary-layer strips, and interpolating the presures from the two ends of the intersected edges onto the intersection point. The latter is achieved by solving a Poisson equation on the unstructured surface mesh between every pair of adjacent boundary-layer strips, thus approximating a linear interpolation of values between adjacent strips.

A direct boundary-layer method due to Nash and McDonald [18] has been coupled to the NSU3D unstructured mesh solver developed by the author, and has been used to recompute the flow over the DLR-F4 configuration shown in Figure 1. A total of 18 spanwise boundary-layer strips were employed for these computations are shown in Figure 6. The transpiration condition is implemented by modifying the inviscid flow boundary condition of slip velocity at an impermeable wall by prescribing a mass flux normal to the wall which is given as:

$$\rho_w v_w \;=\; \frac{\partial(\rho_e U_e \delta^*)}{\partial x} \tag{1}$$

where δ^* is the boundary-layer displacement thickness, v_w represents the blowing velocity, U_e denotes the boundary-layer edge velocity, ρ_w and ρ_e refer to the density at the wall and at the boundary-layer edge (which are taken to be the same), and x denotes the chordwise coordinate. Figure 7 depicts the surface pressure contours computed using the coupled inviscid-boundary-layer approach, showing a more forward shock position with weaker strength than in the inviscid computations (c.f. Figure 2), which is more consistent with the RANS solution depicted in Figure 4. A comparison of the computed lift versus incidence for the inviscid, coupled, and RANS methods are shown in Figure 8. The inviscid computations overpredict lift by a large amount, while the coupled method agrees closely with the RANS approach, both of which slightly overpredict the lift compared to the experimental data. Figure 9 compares the drag polars produced by the three methods. Because the viscous drag component is not available in the inviscid and coupled approaches, and because this quantity is nearly constant throughout the polar range in the RANS calculations, the viscous drag computed at 0 degrees incidence has been added to the inviscid method and coupled method pressure drag results, for the purposes of this comparison.

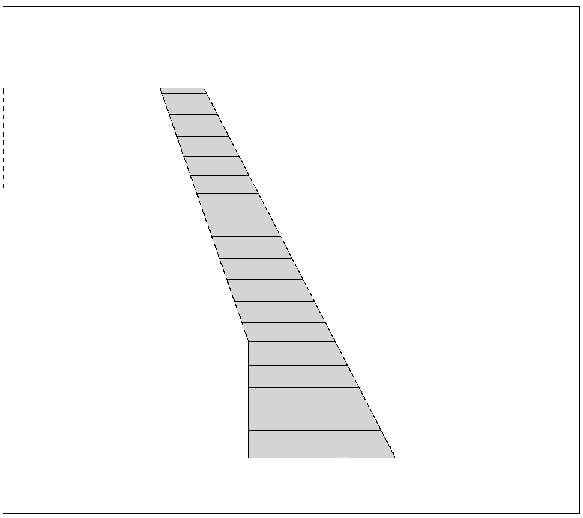

Figure 6: *Illustration of spanwise location of 18 stations for interactive boundary layer computation on wing component of DLR-F4 configuration*

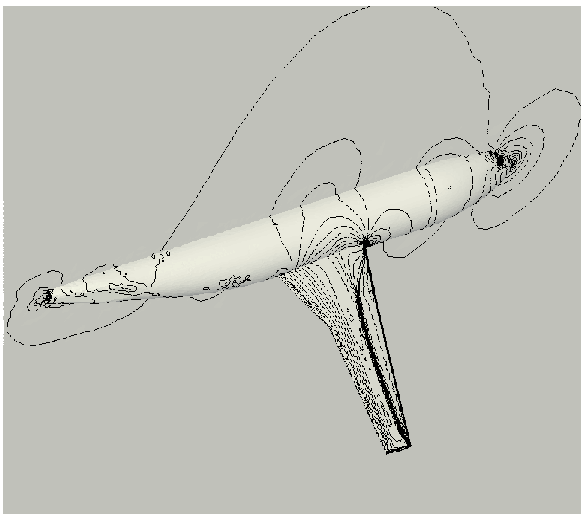

Figure 7: *Computed pressure contours about DLR-F4 configuration using coupled inviscid flow - boundary layer approach on mesh of Figure 1*

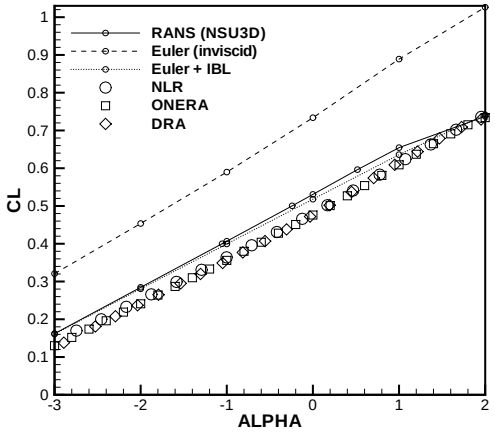

Figure 8: *Comparison of computed lift versus incidence for DLR-F4 configuration using inviscid flow (Euler), coupled inviscid boundary layer approach (Euler+IBL), and Reynolds-averaged Navier-Stokes method (RANS)*

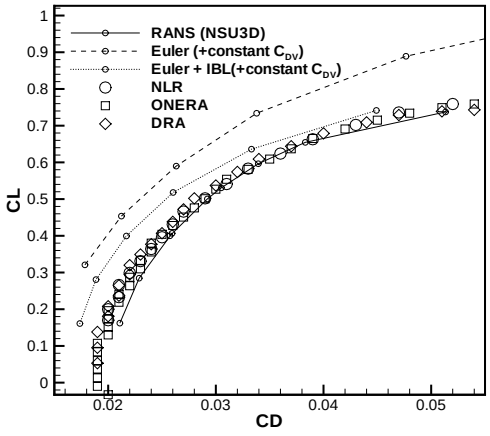

Figure 9: *Comparison of computed drag polars for DLR-F4 configuration using inviscid flow (Euler), coupled inviscid boundary layer approach (Euler+IBL), and Reynolds averaged Navier-Stokes method (RANS). Constant skin friction drag coefficients (from RANS at 0 degrees incidence) have been added to the inviscid and coupled approaches*

Alternatively, the skin friction computed by the boundary layer code [18] could be integrated over the wing for the coupled method, but since no viscous drag data is available for the fuselage, this approach would still require empirical corrections to obtain absolute drag numbers for the coupled method. From the polar plot, the predicted drag values are seen to be substantially improved by the addition of the boundary-layer correction, although at the lower incidences larger discrepancies with the RANS simulation are still observed. Figure 10 illustrates the improved prediction of the surface pressure profile at the 40.9% spanwise station, using the coupled inviscid-viscous approach. The prediction of shock location and strength are greatly improved from the inviscid flow simulation and agree closely with the RANS simulation.

The convergence rates of all three methods are compared in Figure 11. In the coupled method, the boundary layer routine is called every 5 multigrid cycles, producing a sudden increase in the flow residuals as the transpiration velocity is updated. The ensuing multigrid cycles rapidly reduce the residuals, but the overall asymptotic convergence is governed by the convergence of the coupling between the viscous and inviscid computations, and thus lags the convergence of the inviscid method alone, although convergence remains substantially faster than that achieved by the RANS approach. The additional cost of solving the boundary layer equations at all 18 strips, every 5 multigrid cycles, is negligible, as the boundary layer calculations are two-dimensional, and individual boundary layer strips are assigned to different processors in a parallel run. Thus, the increased cost over the inviscid method is mainly a result of the slower asymptotic convergence of the coupled system. In this case, the coupled method is approximately twice as costly as the inviscid method alone, but still 25 times less expensive than the full RANS solution with which it is compared.

The predictive ability of the coupled method must be qualified by its substantially lower cost, and by the fact that this represents a relatively simple implementation of viscous effects (direct boundary layer method, strip-wise two-dimensional). More sophisticated techniques for accounting for viscous effects in coupled inviscid solvers can be considered, generally resulting in improved predictive ability [19, 9]. Coupled Euler-boundary-layer methods can provide good surface pressure profile predictions at substantially reduced cost, and are ultimately capable of adequate drag prediction under certain conditions. However, for cases involving appreciable amounts of flow separation, and/or more complex three dimensional geometries, the limitations of the coupling approach become more apparent, and at some point the use of RANS methods, with their increased computational costs, becomes more suitable, particularly when seeking accuracy of the order of a single drag count. On the other hand, coupled Euler-boundary-layer methods are ideal for providing qualitative information about shock location, strength, and flow separation, which can be used intuitively to improve overall aerodynamic characteristics. Because of their low computational cost, coupled Euler-boundary-layer methods are well suited for automated design optimization strategies, which may involve multiple analysis runs for a single optimization. In these cases, the final optimized design may be re-evaluated with a more accurate RANS code after the optimization phase.

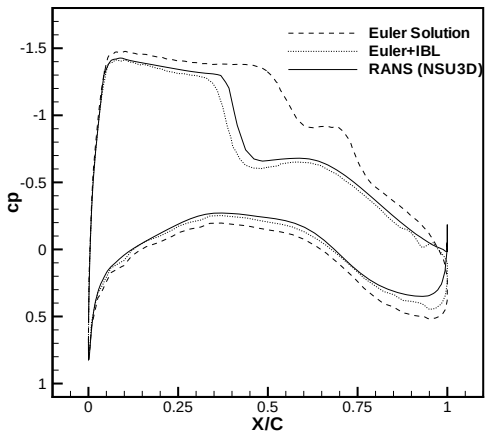

Figure 10: *Comparison of computed surface pressure coefficients at 40.9% span location for inviscid flow, coupled inviscid flow boundary-layer approach, and viscous flow computations*

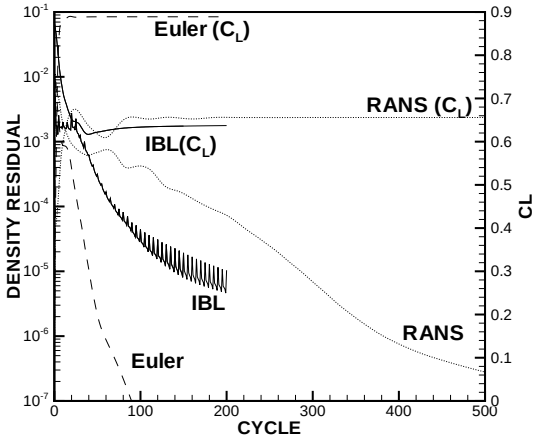

Figure 11: *Comparison of convergence history for inviscid, coupled inviscid flow boundary-layer approach, and viscous flow computations in terms of multigrid cycles*

2.3 Large Eddy Simulations

While RANS methodologies are currently the most suitable approaches for accurate drag prediction, there are many situations where these methods have failed to provide adequate drag prediction. Such cases involve flows with large amounts of separation, including post-stall buffet conditions, and many high-lift flows. While the inclusion of more sophisticated turbulence modeling can be expected to improve predictive ability in certain cases, there is a growing consensus that large eddy simulations (LES) may be required ultimately for accurate simulation of flows with large amounts of separation. Just as RANS simulations are much more costly than inviscid flow simulations, LES for aerodynamic flows represents an increase in computational cost of several orders of magnitude over a steady-state RANS simulation (c.f. Table 1). It is generally acknowledged that complete LES simulations of full aircraft geometries will not be feasible for the foreseeable future [20]. Nevertheless, important advances in LES for aerodynamic flows have been achieved over the last decade [21, 22], with accurate LES calculations over an airfoil near stall having been demonstrated [23]. Hybrid methods which make use of RANS methods in the near wall boundary-layer regions, and LES methods in regions of massively separated flows have also been pursued, in the interest of rendering the cost of LES-type simulations more feasible [20, 24, 25, 26]. While these methods have shown promise, they remaining their infancy, particularly with regards to the accurate prediction of drag.

In this section, the relative advantages and drawbacks of the principal unstructured grid methodologies for computational aerodynamics have been outlined. From this discussion it should be clear that the current state-of-the-art in unstructured mesh drag prediction capability and research lies almost exclusively in the domain of RANS methods, which will comprise the entire focus of the remainder of these lecture notes.

3 Grid Resolution and Discretization Issues

Perhaps the most important requirement for achieving accurate drag prediction centers on the use of adequate grid resolution and suitable discretizations. It is difficult to discuss grid resolution issues separately from discretization issues, since the two are inter-dependent, as an adequate level of grid resolution depends on the discretization at hand. Inadequate grid resolution, even in localized areas, ultimately leads to poor and unreliable drag prediction, while discretizations producing excessive numerical dissipation can render simulations on highly resolved grid useless. In this section, the main unstructured grid discretization alternatives are discussed and their implications for drag prediction are outlined. The particular elements of these discretizations which affect accuracy are given extra consideration. The topic of grid resolution is then discussed, including the choice of element types, grid stretching issues, and boundary layer resolution requirements.

3.1 Choice of Discretization and Effect of Dissipation

3.1.1 Cells versus Points

There exists two major classes of discretizations for unstructured mesh CFD: cell-centered and vertex-based discretizations. Unlike the case for structured meshes, or hexahedral meshes, where the differences between cell-centered and vertex-based discretizations reduces to small differences in accuracy and boundary condition implementation, for unstructured tetrahedral meshes significant differences exist between these two approaches. This is due to the fact that tetrahedral meshes contain on the average 5 to 6 times more tetrahedral cells than vertices, as opposed to hexahedral meshes in which the numbers of cells and vertices are of the same order. Therefore, on a given tetrahedral mesh, a cell-centered approach will result in more degrees of freedom than a vertex approach and can thus be expected to produce higher accuracy, while also incurring higher costs than the vertex-based approach. However, because the discretization stencils resulting from both approaches are vastly different (cell-centered quantities have 4 neighbors, while vertices have a variable number of neighbors averaging 20 to 30), equivalent accuracy and cost are not achieved for equivalent numbers of unknowns in both discretizations. Furthermore, the ratio of cells to vertices is only 2 to 1 on the surface of the aircraft geometry, thus reducing the difference between these two approaches in these regions. The most relevant metric for comparing both methods would involve assessing the cost of both approaches for achieving equivalent levels of accuracy. This involves determining the precise levels of grid resolution which are required by both discretizations for achieving the prescribed accuracy levels. This has proved difficult to quantify with precision, and the debate over cell-centered versus vertex-based unstructured mesh discretizations has never been settled definitively from its inception over 15 years ago [27, 28].

Table 2: *Detailed description of three unstructured meshes for computation of flow over DLR-F4 wing-body configuration. Reproduced from [29]*

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

The fact that cell-centered approaches achieve higher accuracy on coarse grids has been an advantage in cases where highly resolved unstructured meshes have proved difficult to generate. However, the efficiency of cell-centered approaches often lags behind that of vertex-based approaches. A study by Levy [30] found that matching the number of surface grid variables (2 to 1 in this case) achieved similar accuracy for aerodynamic quantities, which resulted in substantial savings for the vertex-based approach over the cell-centered approach. Both cell-centered and vertex-based approaches are in wide use currently, and the main issue for the CFD practitioner is the realization that the level of grid resolution required for accurate drag prediction will be substantially different for either approach. To illustrate this point, the drag polar for the DLR-F4 test case at Mach=0.75 has been computed using the cell-centered code USM3D [31, 29] and the vertex-based code NSU3D [32, 33] on similar grids, designed for cell-centered solvers. These two grids are not identical because the USM3D solver relies on wall functions to model the inner portion of the turbulent boundary layer, while the NSU3D solver resolves the full turbulent boundary layer, and thus requires finer grid resolution near the aircraft surfaces. However, both grids contain similar resolutions on the surface and in the inviscid flow regions. These results are compared with those obtained

by the NSU3D solver on a finer grid designed for vertex-based solvers, where the number of surface grid points approximates the number of surface cells in the previous two grids. Detailed characteristics of these three grids are given in Table 2 (reproduced from [29]).

The computed lift versus incidence for these three cases is shown in Figure 12. The two solvers produce nearly identical lift values on the cell-centered grids, while the finer vertex-based grid yields slightly lower lift with the NSU3D solver. In this respect it is surprising, perhaps fortuitous, that the two solvers agree closely on predicted lift values on the two cell-centered grids. As will be shown in section 5, experimental lift values are overpredicted in all cases, but convergence of lift with mesh resolution for the NSU3D solver is non-monotonic, and further mesh refinement results in even higher lift values. Figure 13 shows substantial differences in drag prediction between NSU3D and USM3D on equivalent cell-centered type meshes, with the baseline NSU3D results (computed on the finer mesh designed for vertex-based schemes) falling in between these two results. Because of the different turbulent boundary layer wall models in both codes (i.e. USM3D employs wall functions), a polar plot of pressure drag is given in Figure 14, in an effort to eliminate any systematic bias in friction drag due to the different wall treatments in both codes. The pressure drag computed by USM3D on the cell-centered grid is seen to agree more closely with that predicted by NSU3D on the vertex-based grid, with the NSU3D results on the cell-centered grid falling below these others.

In addition to accuracy, the expense of these respective computations must be taken into account. From Table 2, it can be seen that the vertex-based grid is approximately three times larger than the cell-centered grid designed for integration to the wall, and almost four times larger than the cell-centered grid designed for wall functions. However, the vertex-based mesh still contains fewer degrees of freedom, i.e. fewer vertices than the number of cells contained in the cell-centered meshes, and therefore, it should not be surprising that the time to solution for the vertex solver NSU3D on the finer vertex-based mesh is still faster than that required by the cell-centered code USM3D on the coarser cell-centered mesh (although precise accuracy equivalence of these solvers on these respective meshes has not been conclusively established). The NSU3D results computed on the mesh designed for cell-centered solvers is almost an order of magnitude faster than those obtained by the USM3D solver, although the produced results have been shown to be of lower accuracy.

While the above comparison is informative, the reader must realize that it is only qualitative in nature. Significant differences exist between the two solvers: USM3D relies on an upwind scheme with gradient reconstruction and turbulent wall function boundary conditions, while NSU3D is based on an artificial dissipation approach, and fully resolves the turbulent boundary layer. As will be shown in the next section, these differences can have a significant effect on drag prediction even when implemented in the same code and run on the same grid. Furthermore, it is difficult to ensure that the ratio of resolution between both cell-centered and vertex-based meshes is similar in all regions of the domain for the above comparison. Finally, the NSU3D solver relies on a multigrid algorithm which provides more rapid convergence to steady-state than the implicit time-stepping algorithm used by USM3D, thus further complicating the meaning of the timing comparisons. The remainder of these lecture notes will focus exclusively on vertex-based schemes, mainly due to the fact that these are the schemes with which the author has the most experience.

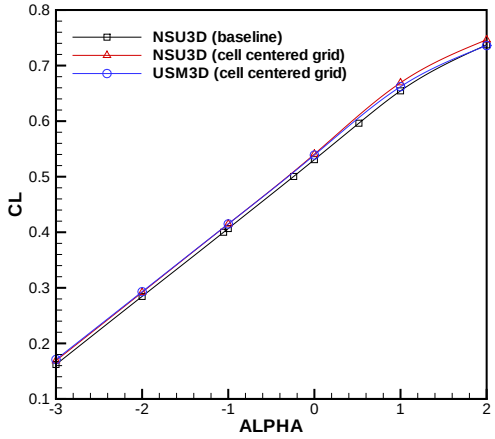

Figure 12: *Comparison of Computed Lift for DLR-F4 Configuration at Mach=0.75 using NSU3D (vertex-based) and USM3D (cell-centered) on Grid Designed for Cell-Centered Scheme, and Comparison with NSU3D results on grid designed for vertex-based scheme.*

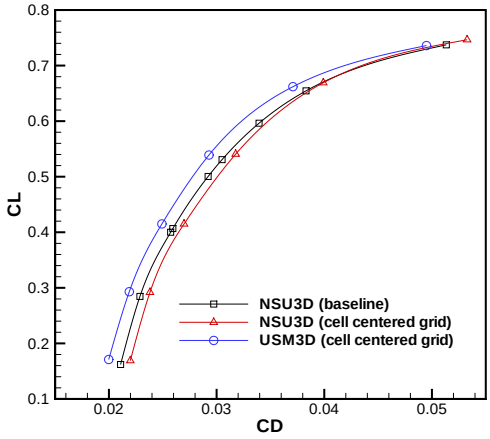

Figure 13: *Comparison of Computed Drag Polar for DLR-F4 Configuration at Mach=0.75 using NSU3D (vertex-based) and USM3D (cell-centered) on Grid Designed for Cell-Centered Scheme, and Comparison with NSU3D results on grid designed for vertex-based scheme.*

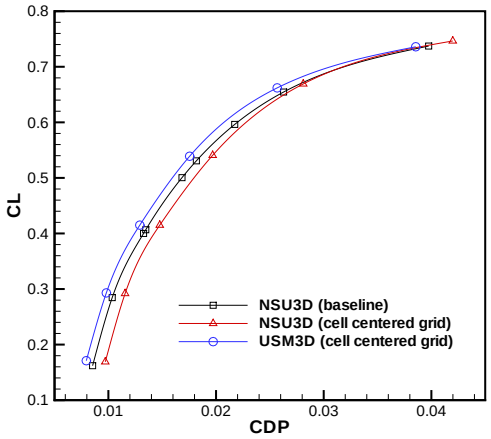

Figure 14: *Comparison of Computed Pressure Drag Polar for DLR-F4 Configuration at Mach=0.75 using NSU3D (vertex-based) and USM3D (cell-centered) on Grid Designed for Cell-Centered Scheme, and Comparison with NSU3D results on grid designed for vertex-based scheme.*

3.1.2 Discretization Formulations

Current-day unstructured mesh aerodynamic production codes rely almost exclusively on formally second-order accurate discretizations. The two main approaches for achieving second-

order accuracy involve centrally-differenced convective terms with added artificial dissipation [34, 33, 35] and projection-evolution schemes using linearly extrapolated values based on gradient reconstruction [28, 29, 36, 37]. Other approaches include fluctuation splitting schemes [38, 39], and streamwise upwind Petrov-Galerkin schemes [40, 41, 42], although these approaches have not seen widespread use in computational aerodynamics and have not been thoroughly validated for drag prediction. Although artificial dissipation schemes and projection evolution schemes have different origins, the final discretizations are closely related. Consider the evaluation of a flux at a control volume interface, as depicted in Figure 15.

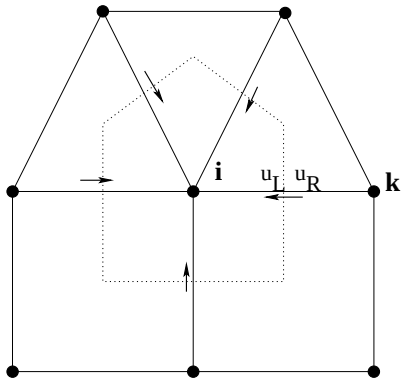

Figure 15: *Illustration of flux evaluation at control volume interface*

The projection evolution scheme requires the solution of an approximate Rieman solver at the interface. For example, the often used Roe Rieman solver can be written as [43]:

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

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

$$u_L = u_i \tag{3}$$
$$u_R = u_k \tag{4}$$

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

$$u_L = u_i + \nabla u_i . \vec{r}_{if} \tag{5}$$
$$u_R = u_k + \nabla u_k . \vec{r}_{kf} \tag{6}$$

where $\vec{r}_{if}$ denotes the position vector extending from vertex i to the control volume interface, and the gradients ∇u are to be evaluated at the mesh vertices. These gradients may be evaluated using a Green-Gauss contour integration around the vertex-based control volumes [28], or by taking a least-squares approximation to the gradient at each vertex which fits the tangent plane constructed using a (least squares) best fit to all neighboring values [44, 45, 37].

Artificial dissipation schemes employ a central difference for the convective terms, and augment these quantities by a dissipative term which is required for stability. The flux at an interface for a first-order accurate artificial dissipation scheme can be written as:

$$\mathbf{F}_{ik} = F(u_i) + F(u_k) + \alpha(u_i - u_k) \tag{7}$$

where α may be a scalar (scalar artificial dissipation) or a matrix (matrix artificial dissipation). In the case where α is a matrix, a natural choice for α, by analogy with equation (2) is:

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

where κ_2 is a constant to be determined empirically. If κ_2 is taken as unity, then the first-order accurate matrix dissipation scheme becomes identical to the first-order accurate projection evolution scheme. On structured meshes, second-order accurate artificial dissipation schemes are obtained by replacing the first difference in equation (7) by a third difference [46]. On unstructured meshes, a second-order accurate artificial dissipation flux can be constructed as [11, 27, 47, 48]:

$$\mathbf{F}_{ik} = F(u_i) + F(u_k) + \alpha \; (L_i(u) - L_k(u)) \tag{9}$$

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

$$L_i(u) = \sum_{k=1}^{neighbors} (u_k - u_i) \tag{10}$$

resulting in an artificial dissipation term which is of the same order as a third difference. 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 [13]. Although these quantities are of the same order, they are not directly proportional to each other, and therefore the parameter κ_2 cannot be taken as unity in this case, but must be determined empirically. There are also discrepancies between the centrally differenced convective fluxes in both schemes, since these are evaluated at reconstructed states in the upwind scheme, rather than at vertex values as in the artificial dissipation scheme. However, numerical experiments reveal that these differences have virtually no effect on drag prediction accuracy in the transonic regime.

The T matrices on the right hand side of equation (2) represent the eigenvectors associated with the linearization of the equations of inviscid compressible flow normal to the control volume face ik [43], while the $|\Lambda|$ matrix is a diagonal matrix containing the absolute values of the five eigenvalues associated with these equations. Of these five eigenvalues, three are

repeated, leaving three distinct eigenvalues which are proportional to: u, u+c, u-c, where u is the velocity normal to the control volume face, and c is the speed of sound. When one of these eigenvalues vanishes, the dissipation for that component at that location also vanishes, which may lead to numerical instabilities. For this reason, it is common to limit the eigenvalues to a minimum fraction of the maximum eigenvalue, such as:

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

$$u + c = sign(u + c) * max(|u + c|, \delta(|u| + c)) \tag{12}$$

$$u - c = sign(u - c) * max(|u - c|, \delta(|u| + c)) \tag{13}$$

where $|u|+c$ is the maximum eigenvalue, and δ is a parameter to be chosen empirically which varies between 0 and 1. When δ is taken as 0, no eigenvalue limiting is applied. When δ is taken as 1, the $|\Lambda|$ matrix reverts to a scaled identity matrix, since all eigenvalues are now taken as $|u| + c$, and the triple matrix product $T|\Lambda|T^{-1}$ reduces to a scalar quantity. For the artificial dissipation discretization, this constitutes the definition of the scalar artificial dissipation, i.e.

$$\alpha = \kappa_2 * max\ eigenvalue \tag{14}$$

which can be computationally cheaper than requiring the evaluation of the full matrices. Small values of δ of the order of 0.1 are common in many production codes, and this process is often referred to as an *entropy fix* [49].

For flows with strong gradients, most notably in the vicinity of shock waves, the above second-order accurate formulations may lead to instabilities, and additional dissipative mechanisms are required. In the upwind scheme, these take the form of limiters applied to the computed gradients [28], while in the artificial dissipation schemes, the difference of undivided Laplacian operators is replaced by a blend of first differences and undivided Laplacian operators [11, 50]. The blending is achieved by multiplying the first difference terms by a pressure switch, (essentially a Laplacian of pressure which becomes large near pressure discontinuities), and scaling these terms with an additional coefficient denoted as κ_1. In both cases, accuracy is reduced from second to first order locally in regions where this additional dissipation is required. For sufficiently fine grids, these regions will be confined to the vicinity of discontinuous flow features (i.e. shocks) and the overall effect on global accuracy will be negligible. However, many practical applications cannot afford the level of grid refinement where limiter/dissipation blending action becomes unobtrusive, and these effects must be monitored closely for accurate drag prediction.

To summarize the above, we have described two alternative forms of dissipative terms, one based on gradient reconstruction, which itself can be done in two manners (Green Gauss or least squares), the other based on undivided Laplacian operators. In both cases, entropy fixes may be applied or omitted, and limiters or difference blending may be used to overcome numerical instabilities at the cost of accuracy. In the following, we will examine the impact all these different discretization details can have on the accuracy of drag prediction.

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 2. Although results on finer grids are available, the computational results on this mesh agree reasonably well with the experimental values, and the use of meshes of this resolution level should accentuate the differences between discretizations which we are seeking, since all the considered discretizations are consistent, and should converge towards the same result as mesh resolution is continually increased.

For the purposes of this study, the baseline discretization consists of the matrix artificial dissipation scheme with the parameter settings: $\kappa_1 = 0, \kappa_2 = 1.0, \delta = 0.1$. Figures 16 through 19 illustrate the sensitivity of computed lift and drag throughout the range of incidences to the variation in these parameters in the artificial dissipation discretization. Precise numerical values are recorded in Table 3, for the single incidence of 0 degrees. In all cases, a more accurate baseline computation using a finer grid of 13 million points is given for comparison purposes. Figures 16 and 17 illustrate the sensitivity of the computed force coefficients to the levels of dissipation in the discretization. In the absence of any low-order disspation blending ($\kappa_1 = 0.0$), lowering the nominal dissipation scaling factor ($\kappa_2 = 0.5$) produces slightly higher lift values, and moderately lower drag values. From Table 3, the lift is seen to increase by 16 counts, while the drag decreases by 6 counts at 0 degrees incidence. The lower dissipation value appears to yield higher accuracy, as the results tend towards those computed on the finer grid. This is expected, since one of the main characteristics of increased grid resolution is the reduction of artificial dissipation effects. While lowering the dissipation coefficient κ_2 below its nominal level can lead to more accurate solutions at little additional cost, numerical instabilities may develop due to insufficient dissipation levels, and the loss of robustness associated with this approach is generally not acceptable in a production environment.

The sensitivity due to the value of the entropy fix in the artificial dissipation discretization is depicted in Figures 18 and 19. Doubling the entropy fix value from $\delta = 0.1$ to $\delta = 0.2$ has very little effect on the computed results. Differences of less than one count in lift and drag are observed in Table 3. This indicates that the solutions are not very sensitive to small values of the entropy fix parameter, and non-zero values are acceptable for increasing robustness while minimally impacting accuracy. On the other hand, when the value of the entropy fix parameter is increased to $\delta = 1.0$, the drag increases by a substantial amount (25 counts at 0 degrees incidence from Table 3). The setting $\delta = 1.0$ corresponds to a purely scalar artificial dissipation scheme, which can be evaluated at reduced cost compared to the matrix dissipation formulation. However, these savings are counterbalanced by the lower accuracy of the scalar approach, which in turn may require the use of finer meshes to regain acceptable accuracy levels. Note the importance of considering both lift and drag values in this case, since the small increase in lift values associated with the scalar dissipation computations may, in isolation, convey an impression of higher accuracy with regards to the baseline case. While straight-forward application of scalar dissipation to aerodynamics calculations can be problematic for accurate drag predictions, modified (scaled) forms of this dissipative operator have been developed, which improved drag prediction ability [51, 35, 52].

All the above results have been computed using the fully second-order accurate disspation model. For flows with strong discontinuities, additional lower-order dissipation terms

may be required to ensure stability. Figures 20 and 21 illustrate the effect on computed force coefficients of the addition of lower-order differences by setting the value $\kappa_1 = 1.0$. In general, very little changes are observed. From Table 3, of the order of one count variation in lift and drag is observed at 0 degrees incidence. These results imply that the blending of lower-order difference dissipation is triggered only in confined areas which do not appreciably affect overall force coefficient values. However, past experience has shown that lower-order dissipation terms can be triggered in areas of strong gradients when inadequate grid resolution is employed, such as at leading edge expansions and in the vicinity of trailing edges, with a resulting decrease in solution accuracy. For these reasons, the blending of lower-order dissipation terms is generally omitted for transonic transport aircraft configurations, where the shock strengths are moderate enough to enable robust capturing with purely second-order accurate dissipation models.

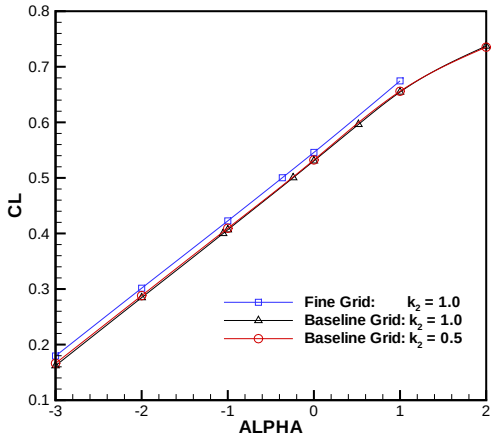

Figure 16: *Effect on computed lift values of variations in the artificial dissipation scaling parameter for DLR-F4 configuration at Mach=0.75*

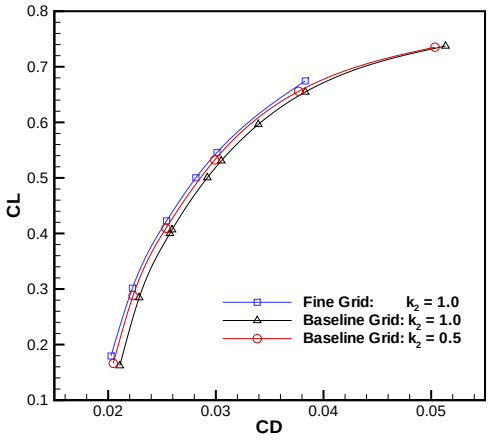

Figure 17: *Effect on computed drag polar of variations in the artificial dissipation scaling parameter for DLR-F4 configuration at Mach=0.75*

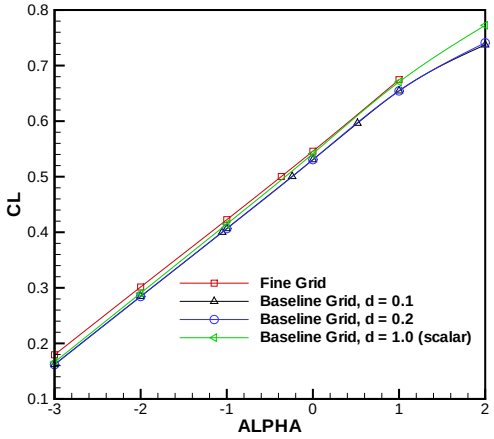

Figure 18: *Effect on computed lift values of variations in the artificial dissipation entropy fix parameter for DLR-F4 configuration at Mach=0.75*

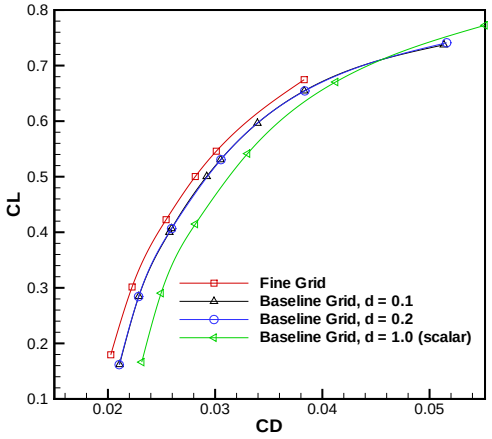

Figure 19: *Effect on computed drag polar of variations in the artificial dissipation entropy fix parameter for DLR-F4 configuration at Mach=0.75*

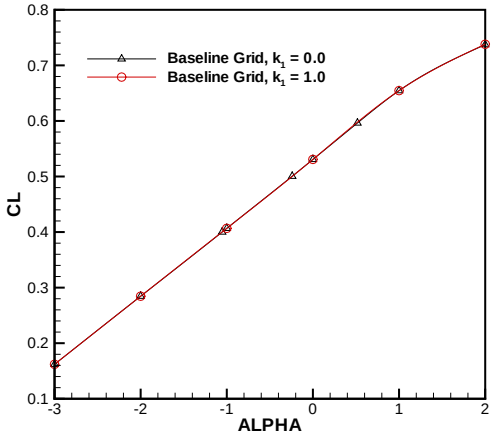

Figure 20: *Effect on computed lift values of application of blended lower-order artificial dissipation for DLR-F4 configuration at Mach=0.75*

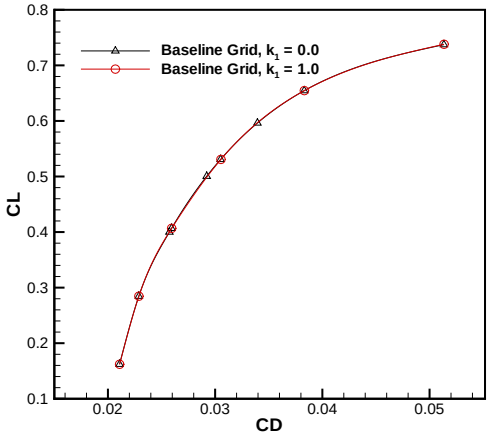

Figure 21: *Effect on computed drag polar of application of blended lower-order artificial dissipation for DLR-F4 configuration at Mach=0.75*

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

Discretization	C_L	C_D
Fine Mesh (13M pts)	0.5459	0.03011
Baseline Mesh : $\kappa_1 = 0.0, \kappa_2 = 1.0, \delta = 0.1$	0.5307	0.03051
Baseline Mesh : $\kappa_1 = 0.0, \kappa_2 = 0.5, \delta = 0.1$	0.5323	0.02990
Baseline Mesh : $\kappa_1 = 0.0, \kappa_2 = 1.0, \delta = 0.2$	0.5307	0.03054
Baseline Mesh : $\kappa_1 = 0.0, \kappa_2 = 1.0, \delta = 1.0$	0.5416	0.03302
Baseline Mesh : $\kappa_1 = 1.0, \kappa_2 = 1.0, \delta = 0.1$	0.5308	0.03054

In Figures 22 through 25 and Table 4, 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 least squares procedure [44, 53], a vanishing entropy fix parameter $\delta = 0.0$, and no limiting. The lift values produced by this discretization scheme are slightly lower than those computed with the artificial dissipation discretization, and since the lift values increase with lower dissipation levels and finer grid resolution, it may be inferred that the least-squares gradient-based upwind discretization is slightly more diffusive than the matrix dissipation scheme. Because the nominal value of the κ_2 coefficient in the matrix dissipation scheme has been determined empirically, it is conceivable that a simple rescaling of the dissipation terms could be used to improve the accuracy in the upwind scheme as well. On the other hand, the drag values produced by this scheme are slightly lower than those obtained with the matrix dissipation scheme. From Table 4, the drag value at 0 degrees incidence is seen to be 8 counts lower than that computed using the matrix dissipation scheme on the same grid. Therefore, there are more substantial differences between these two schemes which extend beyond the simple scaling of the final dissipative terms.

Returning to the baseline unlimited case, the entropy fix parameter is now increased from 0.0 to $\delta = 0.1$, which is the level used in the baseline matrix dissipation settings. In this case, the computed lift values are substantially lower than those obtained with a vanishing entropy fix, and the drag values are substantially higher. From Table 4, the drag at 0^o incidence is seen be increase by 17 counts, with much larger variations in lift. In essence, the accuracy of the scheme has been completely compromised by this small value of the entropy fix, which had little effect on the accuracy of the matrix dissipation scheme. This behavior is attributed to a poor estimate of the gradients in the boundary layer region using the least squares procedure. By under-estimating the gradients in the boundary layer region, the least-squares approach effectively increases the dissipation and reduces the accuracy of the discretization in these regions. This is evident in the solutions produced by the $\delta = 0.1$ case, which resemble first-order accurate solutions. 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. Due to the advancing layers nature of the VGRIDns grid generation package [54] which was used to generate the grids discussed herein, the highly

stretched thin mesh elements in the boundary layer region are closely aligned with the flow direction. This results in near vanishing flow velocity normal to the control volume interfaces in this direction. Since this normal velocity represents one of the eigenvalues of the dissipation matrix (c.f. equation (2)), it is seen that the additional dissipative effect due to the poor gradient estimate is counterbalanced by a vanishing eigenvalue precisely at the same location. Thus, any limit on the minimum size of this eigenvalue, through the use of an entropy fix, triggers the excessive dissipative effect of the poor gradient estimate. It is interesting to note that, although the eigenvalues associated with the normal velocity are small, the two acoustic wave eigenvalues associated with u+c and u-c are not affected by the flow alignment, and yet good accuracy is retained despite the use of inaccurate gradients for these dissipative terms.

Returning to the baseline upwind discretization with vanishing entropy fix, the application of limiters is invoked in order to examine the effect on solution accuracy. The multidimensional monotonicity preserving limiter due to Barth and Jespersen [28] is employed. A small increase in lift and a moderate increase in drag are observed in Figures 24 and 25, respectively. At 0^o incidence, the drag is increased by 8 counts, as shown in Table 4. For flows with strong shocks, the use of limiters may be required to guarantee stability. Unfortunately, as the current example illustrates, the use of limiters has an adverse effect on overall solution accuracy for production type grids of this resolution. Therefore, for transonic flows where shock strengths are not severe, the use of limiters is not recommended. However, it should also be noted that alternate limiter formulations are available, such as the non-strictly monotone limiter of Venkatakrishnan [55], which result in less accuracy degradation, and the current results should be regarded as establishing an outer bounds to limiter sensitivity.

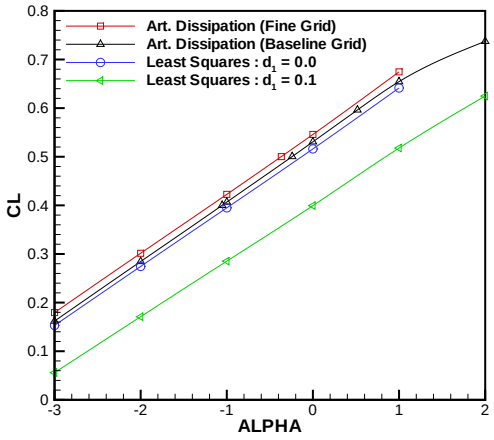

Figure 22: *Comparison of least squares gradient based upwind discretization and matrix artificial dissipation discretization schemes for computed lift values on DLR-F4 configuration at Mach=0.75.*

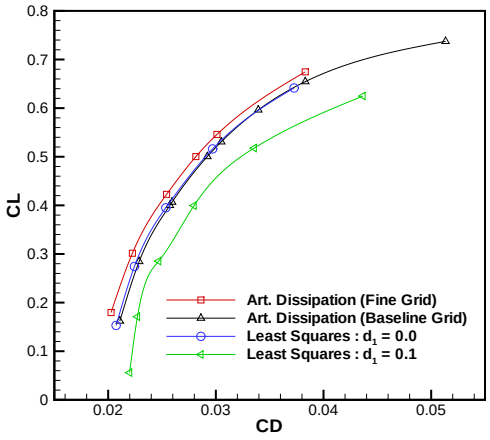

Figure 23: *Comparison of least squares gradient based upwind discretization and matrix artificial dissipation discretization schemes for computed drag polar on DLR-F4 configuration at Mach=0.75.*

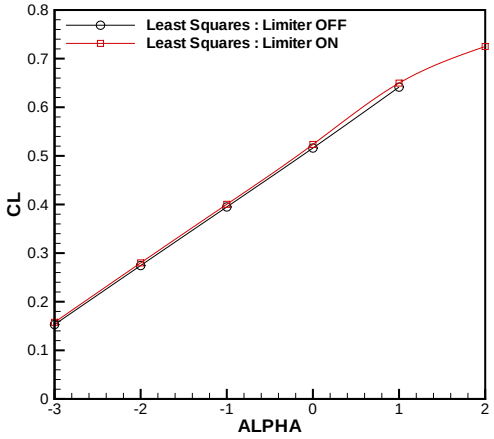

Figure 24: *Effect of limiter on computed lift values for DLR-F4 configuration at Mach=0.75.*

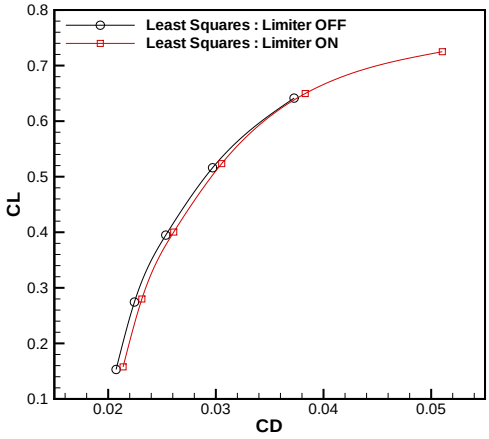

Figure 25: *Effect of limiter on computed drag polar for DLR-F4 configuration at Mach=0.75.*

Table 4: *Variations of computed lift and drag values at Mach=0.75, Reynolds = 3 million and 0^o incidence for DLR-F4 test case as a function of variations in least-squares gradient-based upwind discretization.*

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

3.2 Grid Resolution Requirements

Adequate grid resolution is perhaps the single most important issue for achieving an accurate drag prediction capability. Cost constraints and turn-around time requirements prohibit the use of excessively fine grids, particularly in the course of a drag study, which requires the computation of many single point solutions to populate drag polars and drag rise curves. On the other hand, inadequate grid resolution, even in a single critical isolated region, will likely result in unreliable or even unusable drag results. The problem is rooted in the fact that fully grid-converged RANS solutions are almost never possible in a production environment, and thus changes in grid resolution will produce non-negligible changes in the solution. Sufficient grid resolution is required to enable the capturing of all relevant flow physics (in particular all regions of separated flows), and to ensure that the remaining discretization error is either small enough, or inconsequential. For example, when studying incremental drag differences between two configurations, non-negligible discretization errors may be manageable provided they are of the same magnitude on both configurations, and therefore cancel out in the study of incremental effects. The issue is also particularly complex due to the large range of scales

present in aerodynamic flows, thus requiring vastly different resolutions in different regions of the domain. For example, in the 1.65 million point vertex-based mesh described in Table 1, the ratio of smallest to largest element volume is close to 10^9, and the largest cell-aspect ratio is of the order of 10000. The CFD practitioner must understand the grid resolution requirements of the problem at hand in order to achieve some level of balance between cost and accuracy. Unfortunately, this is still more an art than a science, and is achieved mostly through experience and extensive validation, often building up a set of experience-based rules for grid resolution.

Adaptive meshing techniques are inherently well suited for unstructured mesh methods, and they offer the possibility for ensuring adequate grid resolution in all regions of the domain, by refining and coarsening the mesh locally as the solution evolves. However, successful application of adaptive meshing requires techniques for estimating local discretization error, in order to guide the refinement process. While the mechanics of refining and coarsening unstructured meshes are reasonably well understood today, it is the lack of reliable refinement criteria or error estimators which has held back the potential of adaptive meshing. Most error estimates assume that the solution is asymptotically close to the converged result, an assumption which may not be valid for non-linear problems such as fluid flow, where entire flow patterns(i.e. a separation region) may not be present until sufficient grid resolution is achieved. Because of these difficulties, adaptive meshing strategies have seldom been exploited for demanding problems such as drag prediction. In cases where adaptive meshing has been utilized, heuristic refinement criteria are generally employed, coupled with empirical evidence, and extensive validation [56, 48, 57, 58, 52]. Even in such cases, the construction of well resolved initial meshes remains crucial towards ensuring good final accuracy through the adaptive process.

3.2.1 Choice of Element Type

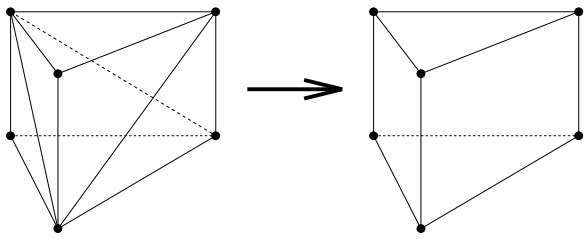

Figure 26: *Merging triplets of tetrahedra into prismatic elements*

As mentioned in the previous section, the appropriate level of grid resolution is dependent on the discretization employed, and the example of Table 1 illustrates how different these requirements can be. A separate issue is related to the type of mesh elements employed, particularly in boundary layer regions. In the boundary layer regions, where the mesh is highly anisotropic, prismatic elements may be substituted in the place of tetrahedral elements. The simultaneous requirements of avoiding large dihedral angle elements for accuracy reasons [59, 60] and high anisotropy for efficiency reasons, leads to right angle tetrahedra in boundary layer regions, as are produced by the advancing layers method [54]. These tetrahedra

can easily be combined into prismatic elements by identifying and grouping together triplets of tetrahedra into prismatic elements [61] as shown in Figure 26. On the other hand, the direct use of prismatic elements constitutes a natural choice in near-wall regions, since there exists an inherent structure in the direction normal to the wall, and semi-structured mesh generation techniques which naturally produce prismatic elements have been proposed [62]. The choice of the type of mesh element to be employed in the near wall regions depends on the delivered accuracy, efficiency, and flexibility of the numerical solver. A two dimensional study by Aftosmis et al. [63] found little difference in accuracy between equivalent grids of quadrilateral and triangular elements. Three dimensional experiments have drawn similar conclusions between prismatic and tetrahedral elements, although published studies examining the particular effect on drag prediction are not known. Figures 27 and 28 illustrate a comparison of predicted lift and drag values between the NSU3D solver, which makes use of prismatic boundary layer elements, and the FUN3D solver which makes use of fully tetrahedral meshes [53], both operating on the same mesh of 1.65 million vertices. In both cases, the discretization is based on a vertex-centered upwind Roe approximate Rieman solver, with least-squares gradient reconstruction. The agreement between the two codes is very close throughout the range of incidences, thus validating the claim that the accuracy is relatively unaffected by the choice of element type in the boundary layer regions. However, this result must be qualified by the fact that there likely exists implementation differences between these two codes whose individual effect on accuracy is not precisely known, and the reported agreement may include fortuitous cancelling of errors. Nonetheless, the displayed agreement lends strong credence to the argument of accuracy insensitivity to element type.

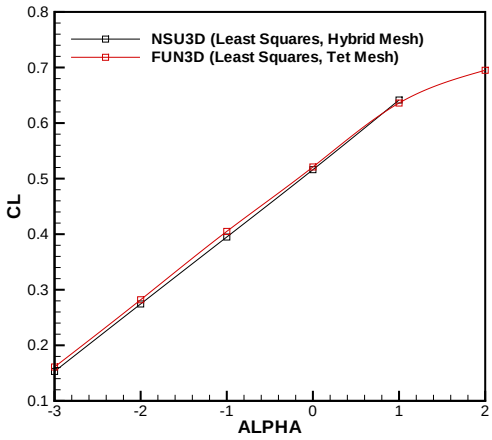

Figure 27: *Comparison of computed lift values using NSU3D solver on hybrid prismatic-tetrahedral mesh and FUN3D solver on fully tetrahedral mesh with identical point distributions for DLR-F4 configuration at Mach=0.75.*

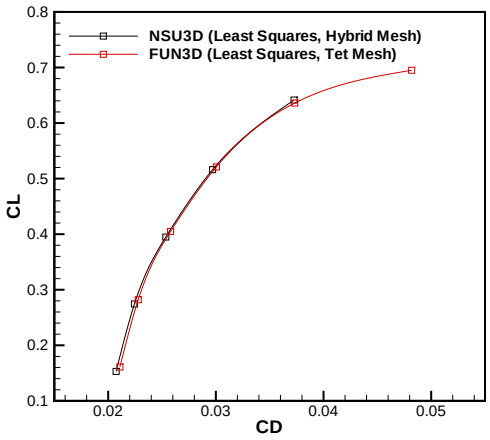

Figure 28: *Comparison of computed drag polars using NSU3D solver on hybrid prismatic-tetrahedral mesh and FUN3D solver on fully tetrahedral mesh with identical point distributions for DLR-F4 configuration at Mach=0.75.*

While accuracy may not be strongly affected by element type, solution efficiency can benefit substantially through the use of prismatic elements. Because each flux evaluation

at each mesh control-volume interface corresponds to a particular edge in the mesh (c.f. Figure 15), the number of flux evaluations required to perform a residual evaluation in a mesh containing prismatic elements is substantially reduced over that required in a fully tetrahedral mesh. This is simply due to the topology of prismatic meshes, which contain on average four times as many edges as vertices (and twice as many cells as vertices) compared with tetrahedral meshes which contain up to 7 times as many edges as vertices (and 6 times as many cells). This can easily be seen by noting that a single hexahedron can be subdivided into two prismatic elements, or six tetrahedral elements, without any addition of new vertices, or by observing the reduction in edges through the element merging operation depicted in Figure 26. Although the boundary layer regions occupy only a small portion of the computational domain, it is not uncommon for more than half of the mesh resolution to be packed into this small region (c.f. Table 1), and thus prismatic element meshes can lead to significant memory and cpu time savings. Another benefit of prismatic element boundary layer meshes derives from the semi-structured nature of such meshes. This facilitates the use of structured mesh techniques such as line solvers in the direction normal to the wall, which can greatly improve solver efficiency [64].

3.2.2 Grid Resolution Issues

As stated previously, the problem of determining and prescribing adequate grid resolution in all regions of the computational domain is central to achieving good drag prediction. The CFD practitioner requires a set of guidelines for determining adequate grid resolution in critical areas such as at leading and trailing edges, as well as general mesh characteristics such as chordwise, spanwise and boundary layer resolution. In the absence of a robust adaptive meshing capability, the approximate location of important flow features must be anticipated and additional grid resolution prescribed in these regions. This can be very rudimentary, such as prescribing higher mesh resolution on the wing upper surface than on the lower surface, in the anticipation of stronger suction peaks and shocks. These issues are not tied to the use of unstructured meshes in particular, but appear equally for structured multi-block and overset mesh techniques. Basic studies in two and three dimensions have demonstrated that the accuracy achieved by vertex-based unstructured mesh discretizations is very close to that achieved by equivalent structured mesh discretizations on similar mesh-point distributions [65, 66, 15]. Thus, mesh resolution guidelines developed for structured mesh methodologies apply equally to vertex-based unstructured mesh methodologies.

Two dimensional studies have shown that of the order of 200 to 300 vertices in the chordwise direction are required to achieve a good predictive ability for viscous airfoil flow in cruise conditions, with high-lift flows often requiring substantially more resolution [67]. Therefore, similar chordwise resolution should be expected in three-dimensional simulations of transport aircraft configurations, noting that this is often not achieved due to resource constraints. Spanwise resolution can be significantly lower than chordwise resolution, since spanwise gradients are generally much weaker, especially for high-aspect ratio wings. However, unlike structured grid methods, the prescription of differing chordwise and spanwise resolutions does not come naturally to unstructured mesh methodologies, which cannot easily distinguish between these geometric coordinate directions. The use of isotropic surface meshes on transport-aircraft-type configurations generally leads to excessive spanwise resolu-

tion and relatively large meshes with inefficient grid-point placement. Various unstructured mesh surface generators, specialized for computational aerodynamics, have the capability for surface mesh stretching in prescribed directions. This capability is illustrated in the surface mesh plots of Figures 29 generated by the VGRIDns program [54], where strong spanwise stretching near the leading edge of a wing is visible. This capability in the VGRIDns program generally results in a reduction of the overall number of grid points by a factor of 3, without sacrificing overall accuracy.

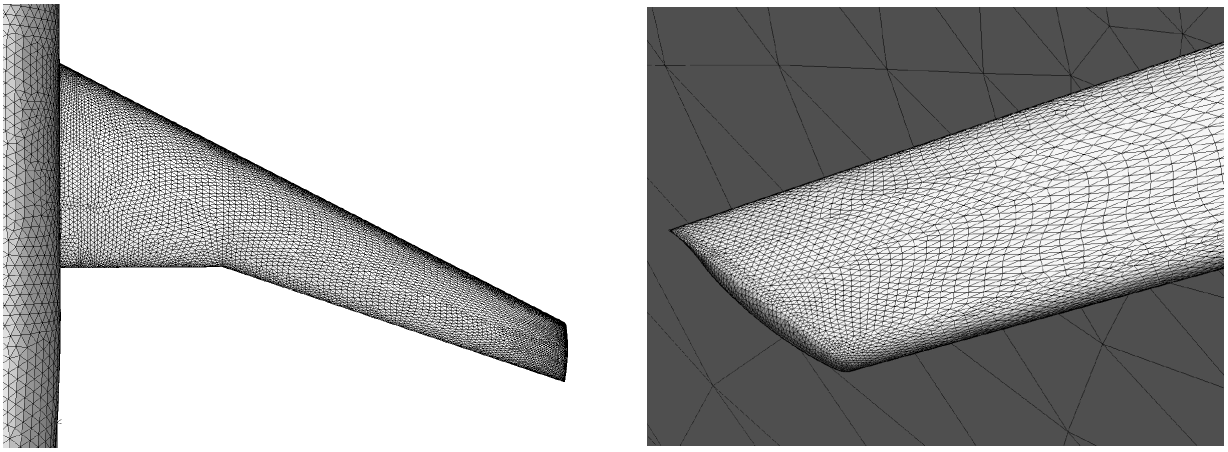

Figure 29: *Illustration of spanwise grid stretching capability of VGRIDns grid generation software (reproduced from Ref. [29])*

Precise control over the spacings in the mesh boundary layer regions is required in order to ensure accurate viscous flow calculations. For example, most turbulence models require that the first grid point off the wall be inside the laminar subregion, usually within the range $y^+ = 1$. Using a flat plat turbulent boundary layer estimate, this distance can easily be calculated as a function of Reynolds number. For Reynolds numbers of the order of 10 million, this requires a wall spacing of approximately 1.e-06 chords, which results in grid aspect ratios of the order of 10000 in these regions. As the mesh traverses the boundary layer, the mesh spacing is allowed to grow in the direction normal to the wall, and the increase in spacing between two adjacent cells is generally accepted to be no more than 1.2 This results in 20 to 30 mesh points spanning the direction normal to the boundary layer. The importance of boundary layer grid resolution on engineering aerodynamic quantities is illustrated by the simple case of flow over a flat plate in two dimensions. The NSU2D unstructured RANS solver [68] has been used to compute this flow on the grid shown in Figure 30, for a Mach number of 0.2 and a plate-based Reynolds number of 10 million, using the Spalart-Allmaras turbulence model [69]. This mesh contains a normal spacing at the plate surface of 1.e-05 plate lengths. Figure 31 depicts the skin friction computed on this mesh as compared with experimental data taken from [70], 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 Figure 31. This simple example illustrates the importance of adequate boundary layer resolution in drag prediction.

While improper boundary layer resolution can hinder drag prediction directly through

poor skin friction prediction, even larger discrepancies can arise for cases involving more complicated flow physics, such as high-lift flows. Figure 32, reproduced from [71], compares the computed versus experimental surface pressures for a three-element high-lift configuration, noting poor agreement over the trailing edge flap upper surface, where an overprediction of the upper surface flow separation is apparent. In Figure 32, the same case has been recomputed using a nearly identical mesh, with the exception that the wall spacing of the mesh was reduced by one order of magnitude, while the growth rate has been kept the same as previously. The agreement with experiment is now much improved, with a smaller separated region occurring on the flap. The fact that the separated region overprediction is simply the result of inadequate grid wall spacing is not necessarily evident, and might otherwise have been attributed to turbulence modeling inadequacies.

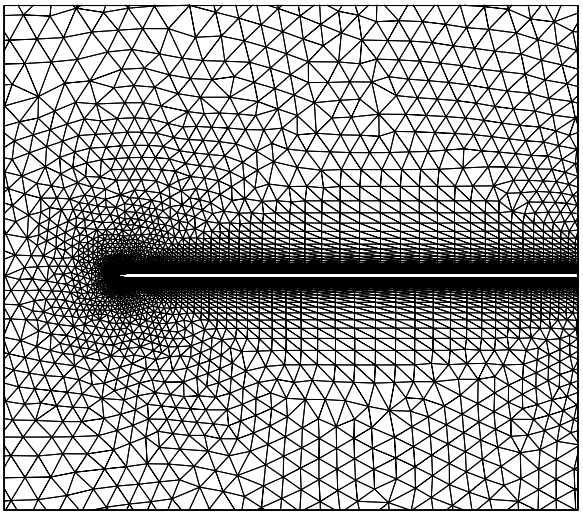

Figure 30: *Unstructured grid used for computation of flow over flat plate in two dimensions*

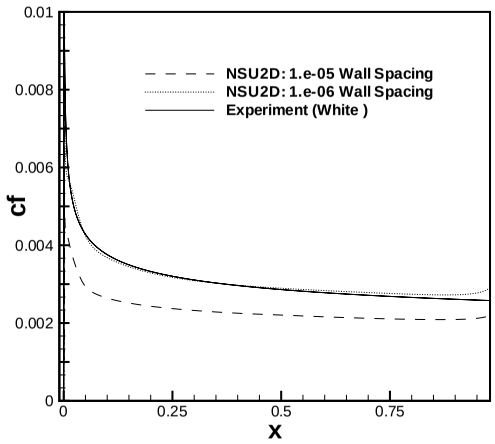

Figure 31: *Effect of normal grid resolution on skin friction for two-dimensional flat plate boundary layer calculation*

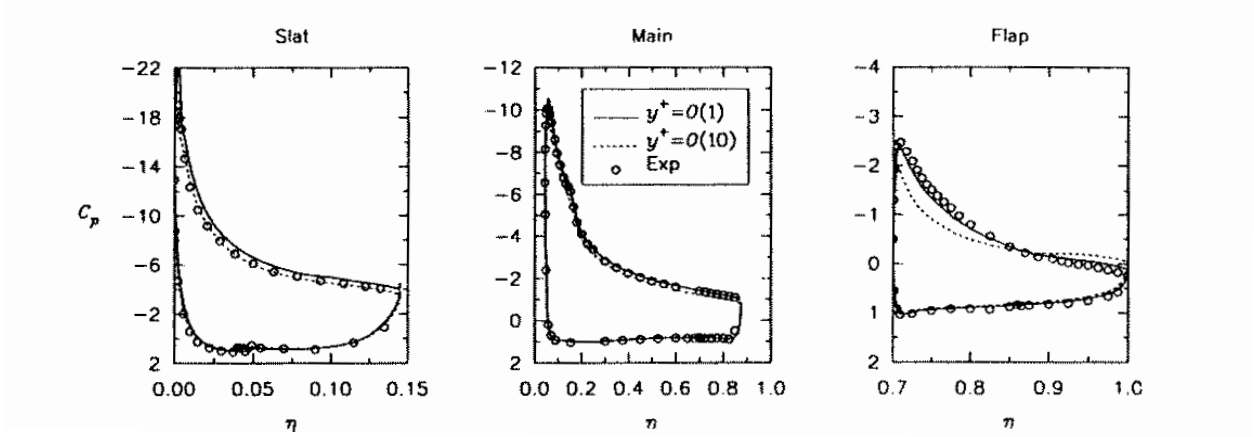

Figure 32: *Effect of normal grid resolution on high-lift three-element airfoil configuration. Mach = 0.2, Incidence = 22 degrees, Reynolds number = 9 million. Reproduced from [71]*

3.3 Grid Convergence

One of the requirements for a viable discretization of any CFD code is that it be consistent. This implies that, as the grid is continually refined, the result of the simulation approaches the solution of the continuous problem, in the limit of infinitely small grid resolution and thus vanishing discretization error. The ultimate goal of any numerical simulation should be the approximation of this continuous result, either directly through extrapolation, or indirectly by certifying that the discretization errors are small enough to be safely neglected. The fully grid converged solution can be estimated through Richardson extrapolation of a sequence of coarser level grid solutions. Figures 33 and 34 illustrate the process for a simple two-dimensional inviscid flow simulation about an RAE2822 airfoil using the NSU2D solver [68]. The flow conditions are subsonic, with a Mach number of 0.3, and an incidence of 2.31 degrees. The simulation is performed on a sequence of progressively finer meshes, and a plot of computed lift and drag as a function of the mesh resolution is constructed. Using this plot, the results of can be extrapolated to the limit of infinite grid resolution. Alternatively, a grid convergence index (GCI) [72] can be estimated. from this sequence of results. For a second-order accurate discretization, the GCI should be equal to 2, although slightly lower values are often observed due to boundary effects, or other phenomena such as accuracy reducing limiters or other shock capturing phenomena. Once this index has been established, solution values in the limit of infinitely fine grid resolution can be estimated through extrapolation along the slope determined by the CGI.

In Figure 34, the results are plotted against a measure of the grid spacing squared (i.e. h^2), normalized by the average spacing (squared) on the coarsest grid. For a second-order accurate scheme, the convergence of the numerical results should appear as a linear function of h^2. This is observed for the lift values, while the drag values display a more non-linear behavior. Note that for this inviscid subsonic case, the theoretically correct behavior of vanishing drag in the continuous limit is observed.

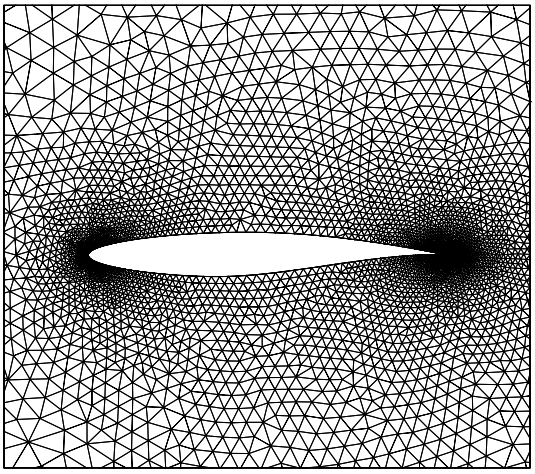

Figure 33: *Intermediate resolution unstructured grid taken from sequence of three grids used for computation of grid converged solution of flow past RAE 2822 airfoil*

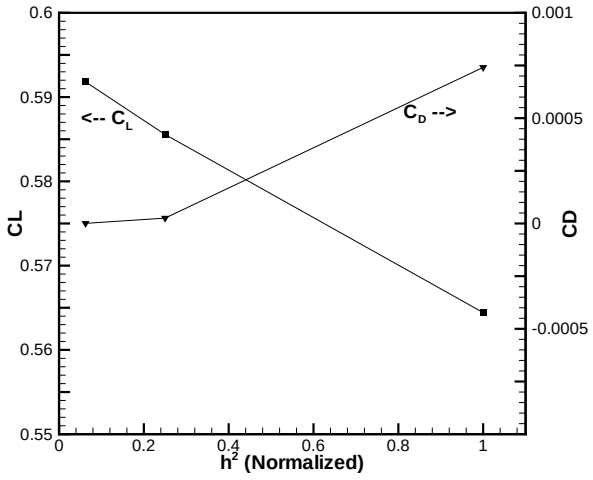

Figure 34: *Illustration of grid convergence for two-dimensional inviscid flow over RAE2822 airfoil using NSU2D solver*

A more practical approach involves observing the results obtained on a sequence of progressively refined grids, and certifying that the sensitivity of the results to changes in resolution on the finest grid are within the acceptable tolerances.

For three-dimensional RANS calculations, the difficulty resides in the affordability of grids fine enough to provide the aforementioned certification. Furthermore, because of the wide variation in scales, (over 10^9 in the grid of Table 1, for example), if the grids are not designed with adequate distributions of resolution in all regions of the domain from the outset, full grid convergence will prove to be intractable. The reality is that very few three-dimensional RANS simulations are adequately grid converged, due to resource constraints. Grid resolution guidelines, as those discussed previously, and careful validation for the specific engineering quantities of interest must be undertaken in order for the simulation to result in a useful engineering tool. Advances in error estimation and adaptive meshing, which hold promise for alleviating these difficulties, are discussed in section 6.

4 Designing an Efficient Unstructured Mesh Solver for Computational Aerodynamics

A useful aerodynamic analysis capability must not only be accurate, but must be efficient and robust as well. This is particularly true for drag prediction studies, where large numbers of analysis cases are required to populate drag polars and drag rise curves, thus requiring full automation with rapid turn-around for individual analysis cases. In order to achieve these characteristics, algorithmic choices specially tailored for aerodynamic problems must be made which enhance accuracy, efficiency and robustness. These include discretization choices, design of appropriate data-structures, development of efficient solution algorithms, and parallelization strategies suited for modern parallel computing hardware. The methodologies discussed in this section consist largely of techniques developed or reproduced by the author and have been implemented in the NSU3D unstructured mesh solver.

4.1 Discretization

One of the first choices to be made by the CFD practitioner involves the use of vertex-based versus cell-centered discretization approaches. As mentioned previously, while cell-centered schemes may prove advantageous when available grid resolution is insufficient for vertex-based schemes, we believe vertex-based schemes offer substantial efficiency advantages over cell-centered approaches for equivalent levels of accuracy. Another design decision involves the choice of appropriate element types. For vertex-based schemes, prismatic elements have been shown to offer similar accuracy characteristics to tetrahedral elements, while providing greater computational efficiency. In fact, for ultimate flexibility, discretizations which are applicable on various types of elements such as tetrahedra, prisms, pyramids, and hexahedra, are most desirable. This can be accomplished using a single edge-based data structure, as shown in Figure 35, rather than a collection of data structures based on the individual elements types. Returning to Figure 15, it can be seen that each control volume interface for a vertex-based scheme is associated with a mesh edge, and hence the flux balance computation for a mesh of arbitrary element types can be computed by looping over the mesh edges [32]. The set of vertices and edges of an unstructured mesh represent the graph of the mesh, which corresponds to the lowest level description of the mesh connectivity, from which all other element information can be retrieved. For this reason, edge data-structures require substantially less memory in order to describe the mesh, and achieve faster computational throughputs than other data-structures, while simplifying the parallelization of the overall solution strategy.

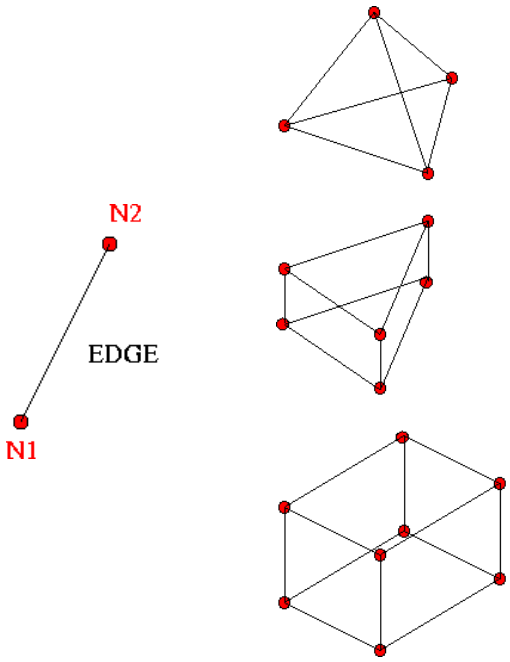

Figure 35: *Illustration of various element types and mesh edge structure which can be used to represent all element types*

While edge-data-structures are well suited for the computation of convective terms on arbitrary element types, consistent discretizations of the full Navier-Stokes viscous terms using edge-based data-structures are only feasible on simplicial (i.e. tetrahedral) elements [44, 66]. For non-simplicial elements, such as prisms, pyramids, and hexahedra, the cross-diffusion terms of the Navier-Stokes equations require a stencil involving neighboring points which are not directly connected by a mesh edge (such as diagonally opposed vertices in a hexahedron). There are various approaches for resolving this difficulty. The NSU3D discretization neglects these cross-terms, which amounts to using a Laplacian operator for the diffusion terms, similar to the terms found in the incompressible (constant viscosity) form of the Navier-Stokes equations [32, 73]. This can also be viewed as equivalent to a thin layer approximation in a compressible boundary layer (in all three coordinate directions). An alternate approach consists of forming the full viscous terms, including all cross-diffusion terms, in a two pass approach, by first computing gradients at all vertices, and then forming the second derivatives as gradients of these gradients [58]. This approach results in a non-nearest neighbor stencil, which may result in lower overall accuracy and may incur instabilities due to odd-even decoupling. Nevertheless, this approach has been used successfully by many authors, although a precise study of the effect on drag prediction of this approach is not known. Hybrid approaches are also possible, where the principal terms are computed by finite differencing along mesh edges, and the cross terms are added in through the double-pass gradient construction method, potentially achieving better accuracy while consistently retaining all cross-derivative terms.

4.2 Solution Methodologies

From the discussion on grid resolution issues, it should be apparent to the reader that one of the main impediments to achieving a more reliable drag prediction capability is related to the ability to employ sufficiently resolved meshes. Limitations on mesh resolution are determined by the available computational resources and required turnaround time, which are directly dependent on the computational efficiency of the solver. Efficient solution algorithms are central not only to reducing the computational the cost of a drag study, but also to improving the accuracy and reliability of the study, by enabling the use of finer meshes. In general, the determination of aerodynamic forces on an aircraft configuration at design conditions constitutes the solution to a steady-state problem. If the spatially discretized equations can be represented by the residual operator $\mathbf{R}(u)$, then the problem to be solved can be written as:

$$\frac{du}{dt} + \mathbf{R}(u) = 0 \tag{15}$$

These equations must be integrated in time until the steady-state is reached, i.e. when the time derivative becomes vanishingly small. An explicit method for advancing these equations in time can be written as:

$$u^{n+1} = u^n - \Delta t\, \mathbf{R}(u^n) \tag{16}$$

where $n+1$ represents the new time level, n represents the old time level, and Δt represents the time step. Explicit methods constitute the simplest techniques for advancing these equations in time. However, because explicit methods rely exclusively on local information to advance the solution in time (through the residual), these methods must obey the Courant-Freidrichs-Lewy condition [74], which states that the maximum stable time step is proportional to the local grid spacing. This results in an order $O(N^2)$ method, where N represents the number of unknowns, since the number of time steps required to reach steady-state increases as the mesh is refined. For highly resolved meshes, explicit methods become extremely inefficient, and more sophisticated solution techniques are required.

An implicit time integration method can be derived by evaluating the residual in equation (16) at the new time step $n+1$, rather than at the old time step n. Since values at the new time step are not known explicitly, a linearization of the residual is required, leading to the form:

$$(\frac{\mathbf{I}}{\Delta t} + \frac{\partial \mathbf{R}}{\partial u})\,(u^{n+1} - u^n) = -\Delta t\, \mathbf{R}(u^n) \tag{17}$$

where $\frac{\partial \mathbf{R}}{\partial u}$ represents the Jacobian matrix, and the first bracketed term on the left-hand side represents a large sparse matrix which must be inverted at each time step. When the time step Δt becomes very large, the first term on the left hand-side vanishes, and the above reduces to a Newton scheme, which can achieve convergence to steady-state in a small number (O(10)) of iterations.

The principal challenges associated with implicit schemes relate to the storage and inversion of the Jacobian matrix. For a second-order accurate discretization, the Jacobian matrix

requires over an order of magnitude more storage than the evaluation of the spatial residual in an explicit scheme. In order to alleviate this problem, it is common to use a simplified form of the Jacobian matrix, obtained by considering a first-order accurate discretization in its construction, while full second-order accuracy is retained for the residual construction on the right-hand-side, since this term determines the accuracy of the steady-state solution. Implicit schemes based on first-order accurate Jacobian approximations have been used successfully in many computational fluid dynamics problems [75, 76, 77, 78]. However, these methods still suffer from large memory requirements (3 to 4 times more than the equivalent explicit algorithm), and the key to their efficient implementation resides in the construction of an efficient linear solver for inverting the Jacobian matrix at each implicit time step, a daunting task in itself.

Multigrid methods represent an alternative approach for devising an efficient solution procedure while incurring minimal memory overheads. A properly formulated multigrid method is optimal in the sense that it achieves linear (O(N)) complexity, meaning that the number of multigrid cycles required to converge to steady-state is independent of the level of grid resolution.

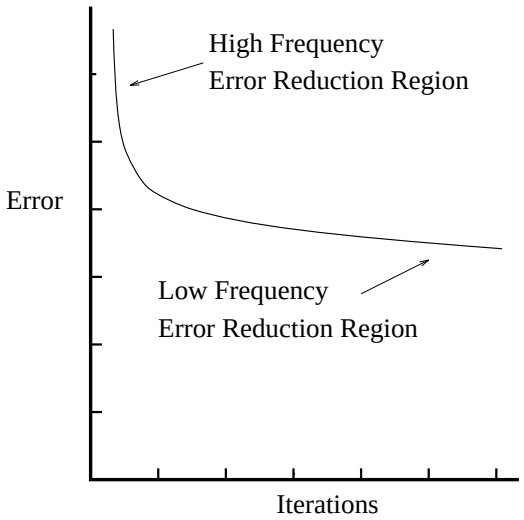

Figure 36: *Typical convergence characteristics of explicit schemes*

The basic idea behind a multigrid algorithm is to accelerate the convergence of a set of equations on a fine grid by time-stepping on coarser grid levels. While explicit schemes are inefficient for solving fine grid problems to completion, they are relatively efficient at eliminating high frequency error modes in the solution (i.e. local error). Figure 36 illustrates the typical convergence behavior of an explicit scheme on a relatively fine mesh. Rapid convergence is initially observed, as the scheme reduces the high frequency error components. However, as the lower frequency errors begin to dominate, convergence slows down dramatically. By transferring the discrete equations to a coarser grid level, once the fine grid high-frequency error modes have been eliminated, the lower frequency modes from the fine grid now appear as higher frequency modes in the multigrid scheme on the coarser grid level,

and are effectively dealt with by the explicit scheme on this grid level. Multigrid methods are applied recursively, on a complete sequence of fine and coarse meshes, with each grid level responsible for a particular bandwidth of error modes, using various inter-grid cycling options as depicted in Figure 37.

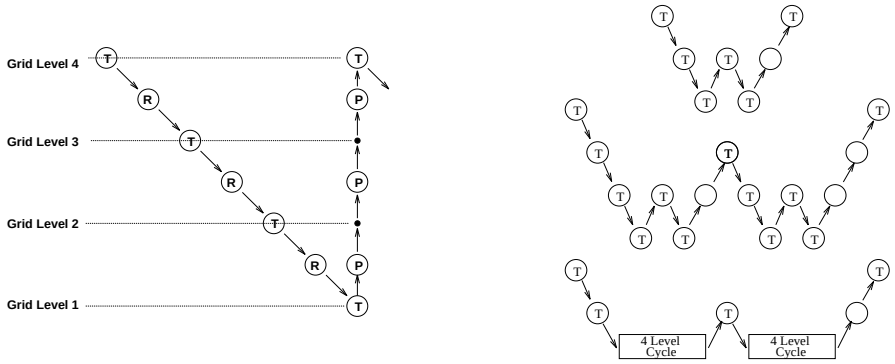

Figure 37: *Illustration of typical multigrid cycling strategies going from fine grid level (top) to coarser levels (bottom): V-cycle (left), W-cycle (right)*

Multigrid methods were originally developed for structured mesh solvers, where coarser levels can be constructed by omitting every second grid line in each coordinate direction. One of the main challenges for unstructured multigrid methods has been the construction of suitable coarse levels. A survey of the various approaches to unstructured mesh multigrid acceleration was given in a previous Von Karman Institute (VKI) lecture series [79]. Initial attempts employed sequences of overlapping but potentially non-nested coarse and fine levels, which are generated independently using the same mesh generation software used to construct the original fine mesh level [12, 13, 80, 79]. While this approach results in an efficient solver, delivering rapid grid-independent convergence rates, the requirement of constructing multiple fine and coarse mesh levels for each geometrical configuration is impractical in a production environment. This requirement lead to the development of agglomeration multigrid methods [81, 82, 83, 84, 35] where the coarse levels are generated in a completely automatic fashion. Agglomeration multigrid methods can most easily be explained in terms of the control volume formulation of an unstructured mesh solver, although they can also be viewed as simplified versions of algebraic multigrid algorithms [85]. In the control volume analogy, coarser levels are formed by fusing together or agglomerating neighboring fine grid control volumes to form a smaller number of larger and more complex control volumes. A graph algorithm is used to partition the initial mesh into coarse level points (i.e. seed points), and neighboring agglomerated points, as depicted in Figure 38, in such a fashion that the coarse agglomerated level represents a maximum independent set of the fine grid graph. Discrete coarse level equations are then obtained by performing flux balances over the agglomerated control volumes, as shown in Figure 39.

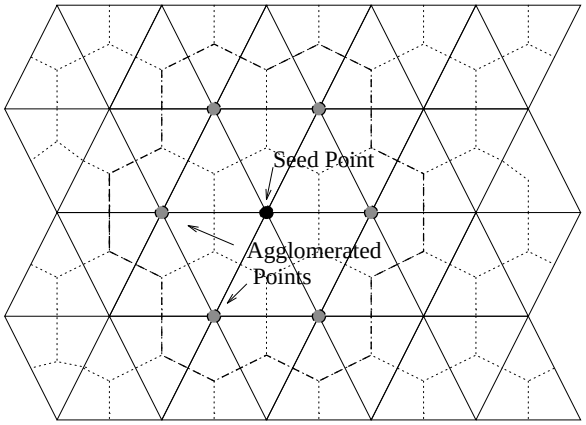

Figure 38: *Illustration of seed point for coarse level control volume and agglomerated points in the agglomeration coarse grid construction strategy*

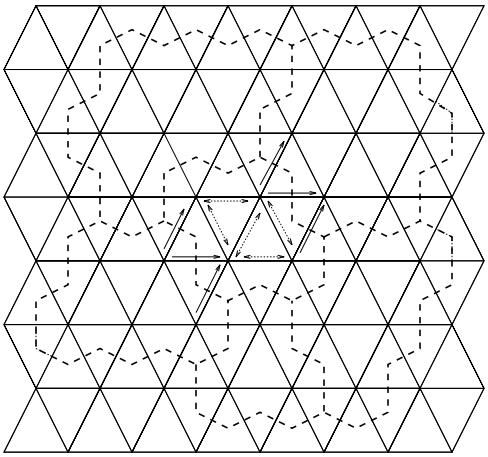

Figure 39: *Illustration of agglomerated coarse level control volume flux balance. Internal fluxes cancel out while boundary fluxes are summed algebraically.*

In the algebraic interpretation, the coarse level equations can be derived by summing the constituent fine grid discrete equations of each coarse level agglomerated control volume [79]. Agglomeration multigrid produces rapid grid independent convergence rates, equivalent to the rates obtained by the overset mesh approach, but with full automation of the coarse level construction. This is demonstrated in the example depicted in Figures 40 and 41. The inviscid flow over an aircraft configuration is computed using the agglomeration multigrid method on the fine mesh shown in Figure 40, using the coarse agglomerated levels shown in the same figure. The observed convergence rate for this case is shown in Figure 41, where the convergence history of the agglomeration multigrid method is compared with that of the overset multigrid method described previously, and the explicit single grid approach. The convergence plots indicate that the agglomeration strategy produces convergence rates which are comparable to the overset multigrid method, and both multigrid approaches result in over an order of magnitude increase in efficiency over the single grid approach, (noting that the cost of a multigrid cycle is less than twice the cost of a single grid cycle). For inviscid flow problems of this type, converged solutions are generally obtained in under 100 multigrid cycles, as seen from Figures 41.

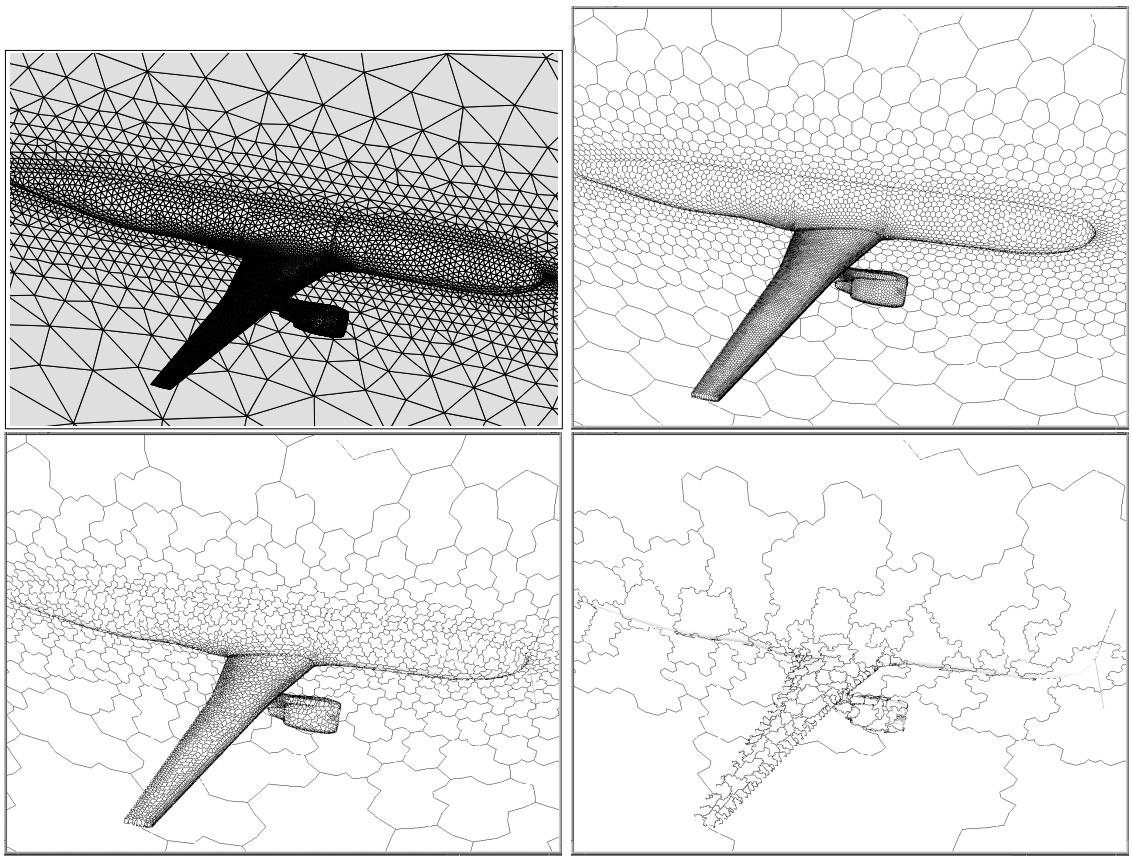

Figure 40: *Illustration of fine grid and coarse agglomerated levels for inviscid flow calculation over aircraft configuration*

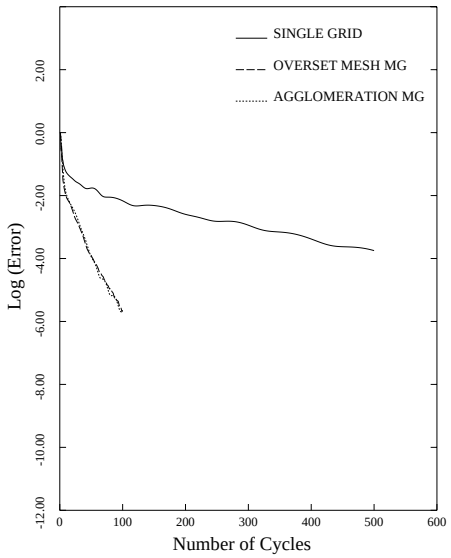

Figure 41: *Convergence rate of agglomeration multigrid algorithm compared with overset-mesh multigrid algorithm and single-grid algorithm for computation of flow over aircraft configuration.*

For viscous flow calculations, which are more relevant for drag prediction studies, convergence to steady-state is much slower than that achieved for inviscid flow problems, mainly as a result of the high degree of grid stretching in the boundary layer regions near the wall. Various strategies for alleviating this stiffness are possible in the context of a multigrid algorithm. These include directional coarsening strategies, and implicit directional solution techniques. Directional coarsening involves constructing the coarse levels by selectively agglomerating in the direction of tight coupling in anisotropic regions, for example normal to the wall in boundary layer regions. Implicit directional solution techniques involve using a stronger implicit-type solver applied in the direction of tight coupling in anisotropic regions. An example of a directional solver is a line solver applied in the normal direction through the boundary layer. This technique has been well known in the context of structured grid methods, where grid lines occur naturally. Although normal coordinate lines do not exist in an unstructured mesh, lines can be formed using a graph algorithm by grouping together mesh edges which represent strong coupling between neighboring vertices in regions of strong mesh anisotropy [86, 87, 64, 88, 89], as illustrated in Figure 42

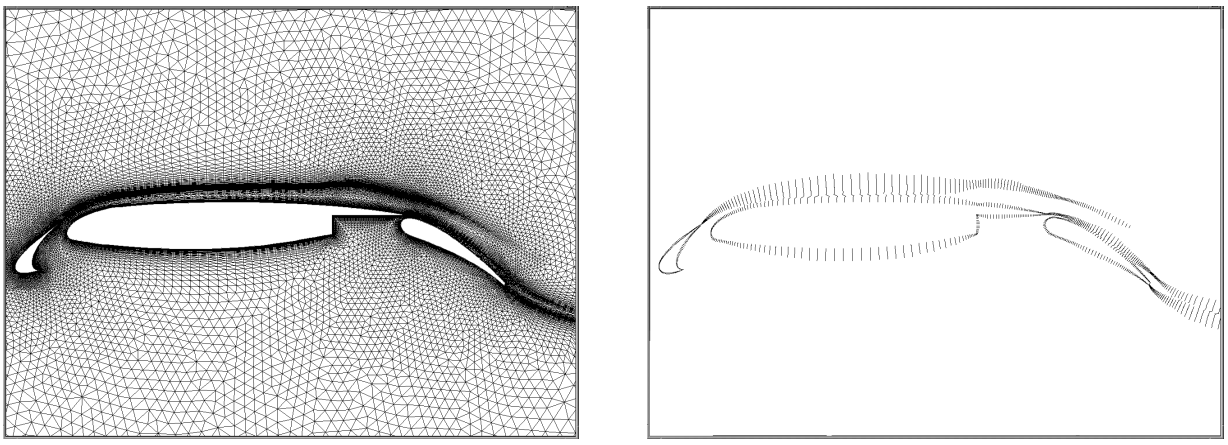

Figure 42: *Two-dimensional unstructured mesh about three-element airfoil configuration and line set constructed using graph algorithm for directional implicit solver*

By retaining the local Jacobian entries associated with these constituent line mesh edges, a locally implicit system is formed, which can be easily inverted using a block tridiagonal algorithm [90]. The combination of directional solution techniques and directional coarsening multigrid has been shown to produce rapid convergence rates which are insensitive to the degree of grid anisotropy [91]. However, in three dimensions, the widespread use of directional coarsening has been plagued by robustness issues, and for production runs the combined approach has been discarded in favor of a more robust but less optimal strategy which makes use of isotropic coarsening coupled with an implicit line solution algorithm on all grid levels [61]. Figure 43 illustrates the convergence rates achieved by this approach for the DLR-F4 configuration mentioned previously, as compared to the unmodified pointwise algorithm previously used for inviscid flows. The figure illustrates well the superior convergence of the directional algorithm, indicating that converged lift and drag values are obtained in under 500 multigrid cycles using the directional line implicit algorithm. Figure 44 depicts the convergence rate obtained by the directional line implicit multigrid algorithm for a three dimensional transport aircraft high-lift flow, using two grids of different resolution, demonstrating the grid independent convergence property of the multigrid algorithm.

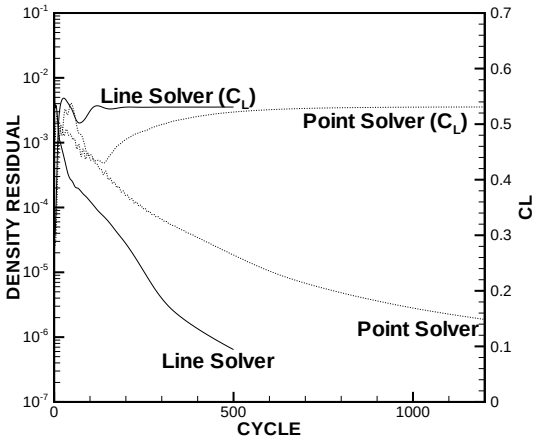

Figure 43: *Comparison of multigrid convergence history obtained with directional implicit line solver versus baseline pointwise solver for transonic flow over DLR-F4 wing-body configuration on baseline mesh (1.65 million points).*

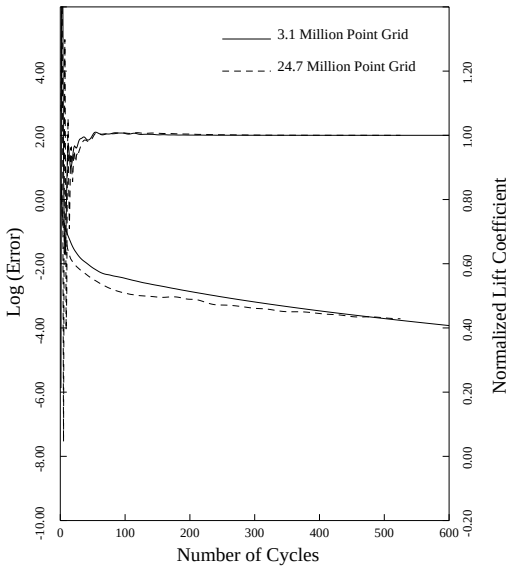

Figure 44: *Illustration of grid independent convergence property of multigrid solver (using line solver) for high-lift flow on two different grids*

4.3 Efficient Hardware Utilization

In addition to algorithmic efficiency, a practical simulation capability must deliver computational efficiency on current day computer architectures. This involves optimization at the local processor level, as well as efficient strategies for running on massively parallel systems involving large numbers of processors. At the individual processor level, cache-based scalar micro-processors and vector-processors represent the two principal architectures which are

used for numerical computations. For scalar micro-processors, efficient cache-utilization is paramount for obtaining good overall performance. Mesh vertex renumbering strategies using bandwidth minimization techniques are often used to produce more cache-local data-access patterns, and judicious numbering of edges in the edge-based data-structure can also be used to improve cache-efficiency, generally resulting in an increase in computational rates by a factor of 2 to 3 [92, 93]. Although cache-based commodity micro-processors have gained popularity over the last decade due to their lower cost and rapid increase in clock-speeds, high-end custom vector processors such as those supplied by NEC, Fujitsu, and the recently announced CRAY X-1, remain serious contenders for cost-effective large-scale numerical simulations. The requirements for effective vectorization are often at odds with those required for efficient cache-optimization (i.e. reversing loop indices, avoiding branching statements within loops, avoiding recurrences) and good vectorization for unstructured mesh solvers generally will not occur with automated compiler flags alone. Thus alternate sorting and loop processing strategies must be incorporated into the code with the option to toggle back and forth between cache and vector optimization strategies, in the event both types of architectures are to be employed. For vectorization of edge loops, the edges must be sorted into groups within which no recurrences occur (i.e. no two edges access the same vertex), and then vectorization can proceed within each group [93]. Vectorization of the implicit line solver is achieved by grouping together sets of lines and initiating a vector loop over all lines within a group for each phase of the tridiagonal line solution algorithm, thus solving all decoupled tridiagonal systems within a group simultaneously [64].

For parallel computations, in order to scale well on large numbers of processors, a domain decomposition approach must be adopted, where the fine mesh is split into multiple partitions, one for each processor, and the edges crossing partition boundaries are redirected to newly created ghost vertices, as shown in Figure 45. After a flux balance has been performed locally on each processor, the ghost vertices are required to add their residual contributions to their vertex image in neighboring processors. This inter-processor communication is handled via MPI [94] on distributed memory machines such as clusters of workstations, or by OpenMP [95] on shared memory machines such as the SGI Origin, or by both techniques on networked clusters of shared memory multi-processors [96]. Mesh partitioning is performed using available graph-based partitioners such as Metis [97] or Chaco [98]. Graph partitioners require only a set of vertices and edges (possibly weighted sets) as input. These methods attempt to generate balanced partitions of sets of vertices, and to minimize the total number of edges which are intersected by the partition boundaries, thus minimizing overall communication. It is important during the partitioning process that the line structures employed in regions of high mesh stretching be confined to individual partitions in order to obviate the need to parallelize the tridiagonal line solution algorithm, which is inherently sequential in nature. This is achieved by contracting the original unweighted graph, which defines the unstructured mesh along the implicit lines, to produce a weighted graph as shown in Figure 46. Unity weights are assigned to the original graph, and any two vertices which are joined by an edge which is part of an implicit line are then merged together to form a new vertex. Merging vertices also produces merged edges, as shown in Figure 46, and the weights associated with the merged vertices and edges are taken as the sum of the weights of the constituent components. The contracted weighted graph is then partitioned using one of

the graph partitioners described in references [97, 98], and the resulting partitioned graph is then decontracted, i.e. all constituent vertices of a merged vertex are assigned the partition number of that vertex. Since the implicit lines reduce to a single point in the contracted graph, they can never be broken by the partitioning process. The weighting associated with the contracted graph ensures load balancing and communication optimization of the final partitioned result.

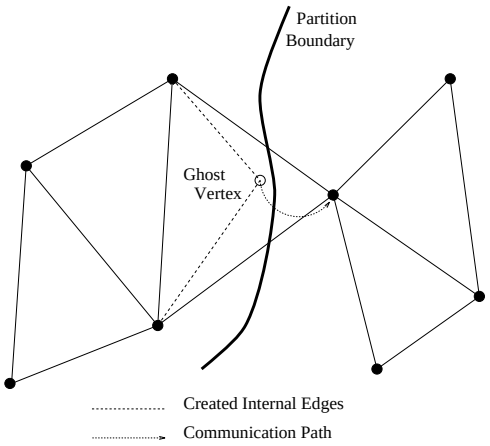

Figure 45: *Illustration of partitioning of unstructured mesh and creation of ghost points with resulting inter-processor communication*

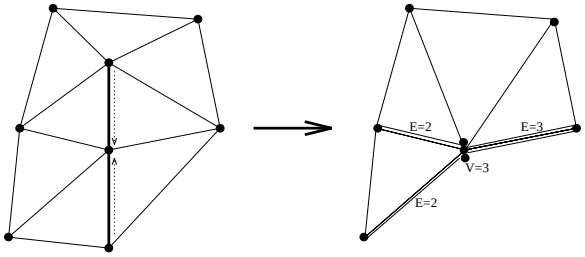

Figure 46: *Illustration of contraction of lines into weighted graph for partitioning process*

For the multigrid algorithm, each agglomerated level must also be partitioned across the processors of the machine. Since the mesh levels of the agglomeration multigrid algorithm are fully nested, a partition of the fine grid could be used to infer a partition of all coarser grid levels. While this would minimize the communication of the inter-grid transfer routines, the amount of intra-grid computation on each level is much more important than the inter-grid computation between each level, and it is essential to optimize the partitions on each grid level rather than between grid levels. Furthermore, inferring coarse level partitions from the fine level would not guarantee the containment of all implicit lines on individual processors for the coarser levels. For these reasons, each grid level is partitioned independently, resulting in unrelated coarse and fine level partitions. In order to minimize inter-grid communication, coarse level partitions are reordered in such a manner that they are assigned to the same processor as the fine grid partition with which they share the most overlap.

The parallel efficiency of the above described implementation is illustrated by the scalability curve of Figure 47, for a small test case of 177,000 points running on a cluster of commodity PCs, and in Figure 48, and a larger 24.7 million point case on a massively parallel CRAY T3E machine. Good speedups are observed in all these cases, with only slight drop-off occurring at the higher numbers of processors. In these cases, the scalability of the single grid algorithm is plotted alongside the scalability of the multigrid algorithm. The moderate deterioration in scalability in going from the single grid algorithm to the multigrid algorithm is a measure of the increased communication generated by the coarser grid levels, which contain fewer vertices, but are distributed across the same number of processors. However, the single grid algorithm is non-competitive in terms of overall cpu time, since

it requires over an order of magnitude more computational effort to achieve steady-state convergence compared with the multigrid algorithm, and is simply shown to quantify the communication penalty of the multigrid method. Figure 49 illustrates the speedups achieved on an SGI Origin 2000 machine using up to 128 processors, for the multigrid algorithm operating on a 3 million point grid, using either MPI or OpenMP for communication, on this shared memory machine. Good speedups are observed, and both communication strategies are seen to offer comparable efficiency.

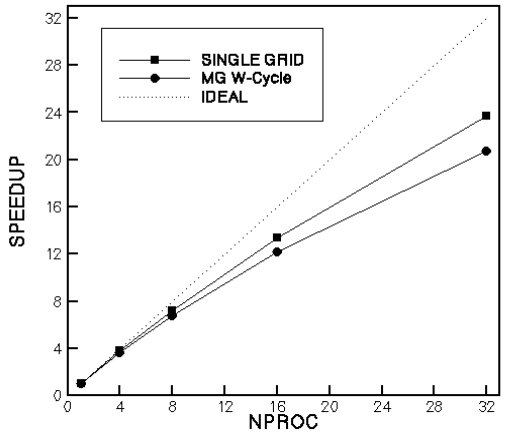

Figure 47: *Observed speedup of multigrid and baseline algorithm for small 177,000 point case on PC cluster*

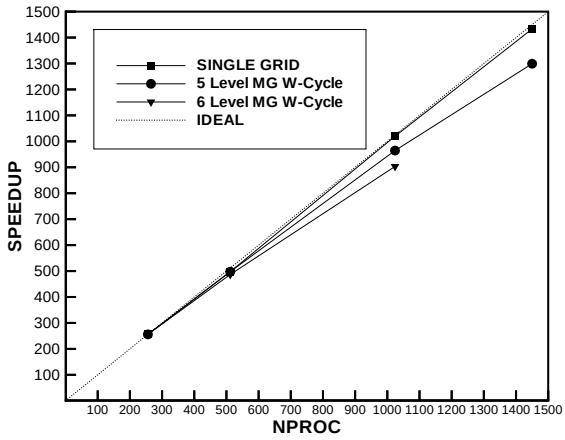

Figure 48: *Observed speedup of multigrid and baseline algorithm for large 24.7 million point case on CRAY-T3E 1200E machine*

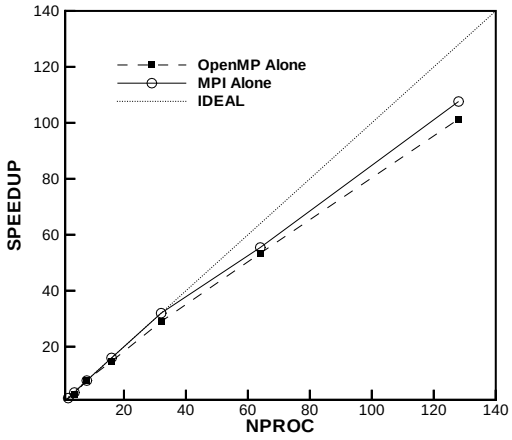

Figure 49: *Observed speedup of multigrid algorithm for 3 million point case on SGI Origin 2000 using MPI and OpenMP communication strategies*

This section has outlined the various considerations which must be taken into account to achieve an efficient solution capability. In summary, it is the combination of efficient discretizations, data-structures, coupled with rapid multigrid convergence and good parallel scalability which enables drag calculations on reasonably fine meshes of the order of several million points to be obtained in several hours on clusters of commodity PCs [33], or for very large cases to be computed on large supercomputers with similar turnaround times [61].

5 Drag Prediction using Unstructured Mesh Solvers

The previous sections have emphasized the various aspects which can affect the drag prediction capability of unstructured mesh solvers. In this section an objective description of the state-of-the-art in drag prediction using unstructured mesh solvers is now given, using several concrete examples, including grid resolution effects, and efficiency considerations, while providing a comparative assessment of unstructured mesh methods with structured mesh methods when possible. Drag prediction is examined for three types of problems: determination of absolute values in the transonic cruise regime, determination of incremental effects, such as nacelle installation drag in the cruise regime, and high lift problems. These examples represent typical areas of study which arise during a transport aircraft design exercise.

5.1 Wing-Body Cruise Drag

The capabilities of unstructured mesh solvers at predicting drag in the transonic cruise regime can only be assessed from the results of careful validation studies. As an example, the AIAA 1st Drag Prediction workshop, held in June 2001, featured the computation and comparison with experiment of the transonic flow over a DLR-F4 wing-body configuration over a range of flow conditions. This particular configuration was chosen because it is representative of a modern supercritical swept-wing transport aircraft, and has been extensively tested in three different wind tunnels. Participants included Reynolds-averaged Navier-Stokes formulations based on block-structured grids, overset grids, and unstructured grids, thus affording an opportunity to compare these methods on an equal basis in terms of accuracy and efficiency. A standard mesh was supplied for each type of methodology, with participants encouraged to produce results on additionally refined meshes, in order to assess the effects of grid resolution. A Mach number versus lift coefficient (C_L) matrix of test cases was defined, including the calculation of drag polars over a range of Mach numbers, and constant C_L drag rise curves, as are typically used in airplane design by industry.

The baseline grid supplied for the workshop was generated using the VGRIDns package [54]. This approach produces fully tetrahedral meshes, although it is capable of generating highly stretched semi-structured tetrahedral elements near the wall in the boundary-layer region, and employs moderate spanwise stretching in order to reduce the total number of points. A semi-span geometry was modeled, with the far-field boundary located 50 chords away from the origin, resulting in a total of 1.65 million grid points, 9.7 million tetrahedra, and 36,000 wing-body surface points. The chordwise grid spacing at the leading edge was prescribed as 0.250 mm and 0.500 mm at the trailing edge, using a dimensional mean chord of 142.1 mm. The trailing edge is blunt, with a base thickness of 0.5% chord, and the baseline mesh contained 5 grid points across the trailing edge. The normal spacing at the wall is 0.001 mm, which is designed to produce a grid spacing corresponding to $y^+ = 1$ for a Reynolds number of 3 million. A stretching rate of 1.2 was prescribed for the growth of cells in the normal direction near the wall, in order to obtain a minimum of 20 points in the boundary layer. The characteristics of this grid are detailed in Table 2 in section 3.

The fully tetrahedral VGRIDns meshes were then merged into hybrid prismatic-tetrahedral meshes as described in section 3. The merging operation results in a total of 2 million created prismatic elements, while the number of tetrahedral cells is reduced to 3.6 million, and

a total of 10090 pyramidal elements are created for the merging of prismatic elements into tetrahedral elements in regions where quadrilateral faces from prismatic elements are adjacent to tetrahedral elements. A higher resolution mesh was also generated using VGRIDns with smaller spacings in the vicinity of the wing root, tip, and trailing edge, resulting in a total of 3 million grid points, and 73,000 wing-body surface points. One of the features of this refined grid is the use of a total of 17 points across the wing trailing edge versus 5 for the baseline grid. After the merging operation, this grid contained a total of 3.7 million prisms and 6.6 million tetrahedra.

An additional fine mesh was obtained through global refinement of the baseline workshop mesh. This strategy operates directly on the mixed prismatic-tetrahedral mesh, and consists of subdividing each element into 8 smaller self-similar elements, thus producing an 8:1 refinement of the original mesh [57]. The final mesh obtained in this manner contained a total of 13.1 million points with 16 million prismatic elements and 28.8 million tetrahedral elements, and 9 points across the blunt trailing edge of the wing. This approach can rapidly generate very large meshes which would otherwise be very time consuming to construct using the original mesh generation software. One drawback of the current approach is that newly generated surface points do not lie exactly on the original surface description of the model geometry, but rather along a linear interpolation between previously existing surface coarse grid points. For a single level of refinement, this drawback is not expected to have a noticeable effect on the results. However, a capability for re-projecting new surface points onto the original surface geometry is ultimately required and is currently under development.

The baseline grid was found to be sufficient to resolve all major flow features. The computed surface pressure coefficient on the baseline grid for a Mach number of 0.75, Reynolds number of 3 million, and C_L= 0.6 is shown in Figure 50, illustrating good resolution of the upper surface shock. A small region of separation is also resolved in the wing root area, as shown by the surface streamlines for the same flow conditions, in Figure 51. Figure 52 depicts the computed y^+ values at the break section for the same flow conditions, indicating values well below unity over the entire lower surface and a majority of the upper surface. The convergence history for this case is shown in Figure 53. The flow is initialized as a uniform flow at freestream conditions, and ten single grid cycles (no multigrid) are employed to smooth the initialization prior to the initiation of the multigrid iteration procedure. A total residual reduction of approximately 5 orders of magnitude is achieved over 500 multigrid cycles. Convergence in the lift coefficient is obtained in as little as 200 multigrid cycles for this case, although all cases are run a minimum of 500 multigrid cycles as a conservative convergence criterion.

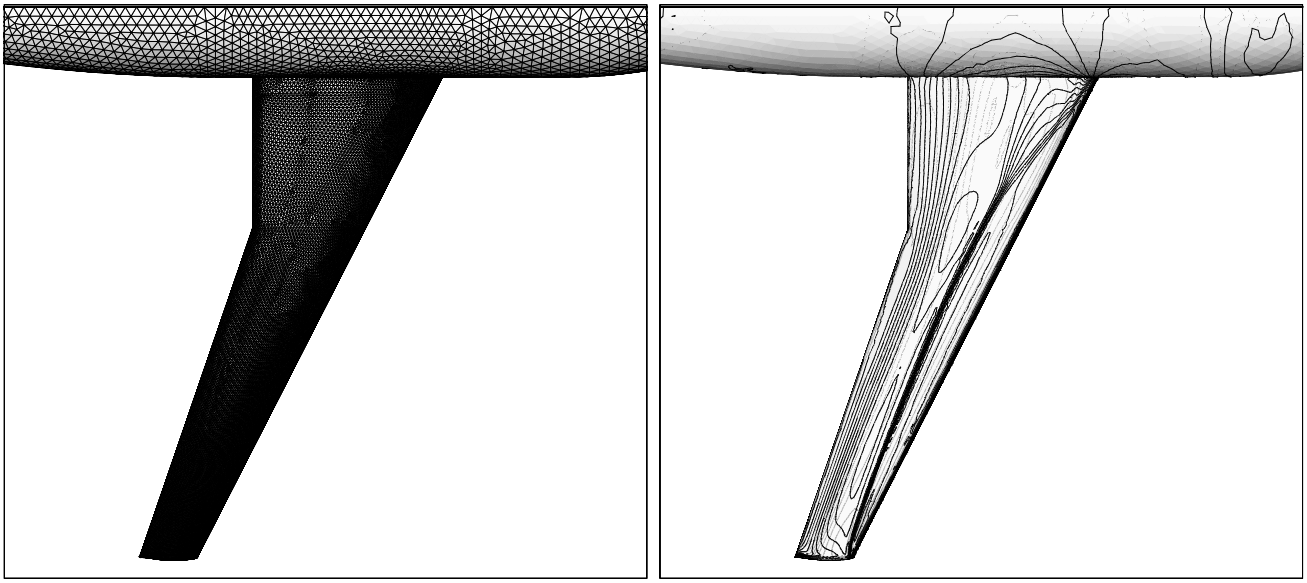

Figure 50: *Baseline grid and computed pressure contours at Mach=0.75, C_L = 0.6 , Re = 3 million*

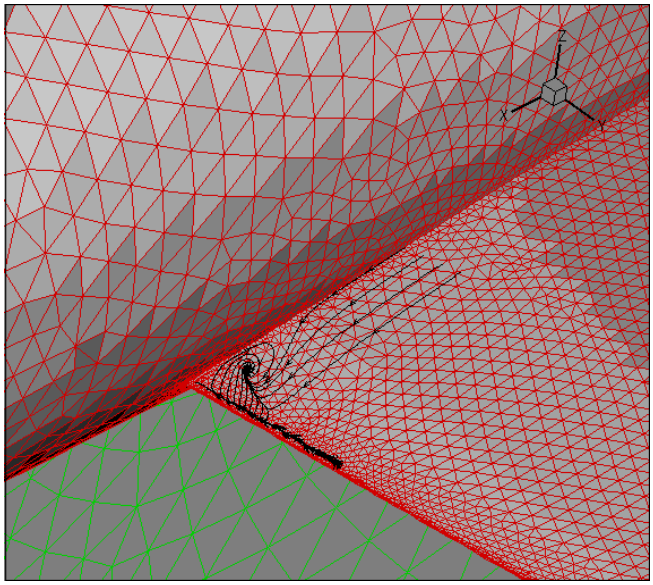

Figure 51: *Computed surface oil flow pattern in wing root area on baseline grid for Mach=0.75, C_L = 0.6, Re=3 million*

This convergence behavior is representative of the majority of cases run, with some of the high Mach number and high C_L cases involving larger regions of separation requiring up to 800 to 1000 multigrid cycles. A flow solution on the baseline grid requires 2.8 Gbytes of memory and a total of 2.6 hours of wall clock time (for 500 multigrid cycles) on a cluster of commodity components using 16 Pentium IV 1.7 GHz processors communicating through 100 Mbit Ethernet. This case was also run on 4 DEC Alpha processors, requiring 2.4 Gbytes of memory and 8 hours of wall clock time. This case was also benchmarked on 64 processors (400MHz) of an SGI Origin 2000, requiring 3 Gbytes of memory and 45 minutes of wall clock time. The memory requirements are independent of the specific hardware and are only a function of the number of partitions used in the calculations. The cases using the 3 million point grid were run on a cluster of 8 DEC Alpha processors communicating through 100 Mbit Ethernet and required approximately 8 hours of wall clock time and 4.2 Gbytes of memory. The 13 million point grid cases were run on an SGI Origin 2000, using 128 processors and required 4 hours of wall clock time and 27 Gbytes of memory. A description of the three grids employed and the associated computational requirements on various hardware platforms is given in Table 5.

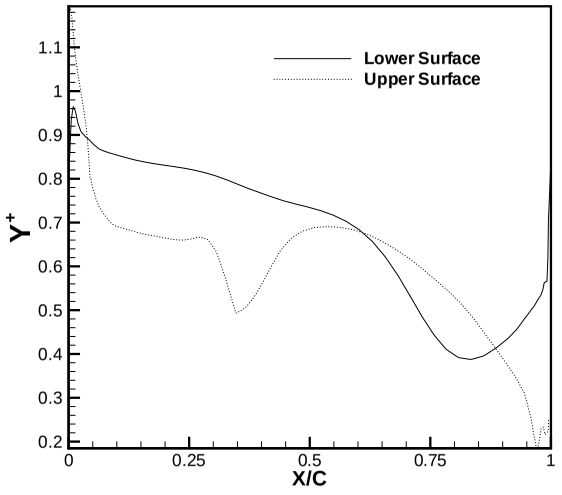

Figure 52: *Computed y^+ on wing surface at span break section on baseline grid for Mach=0.75, C_L = 0.6, Re = 3 million*

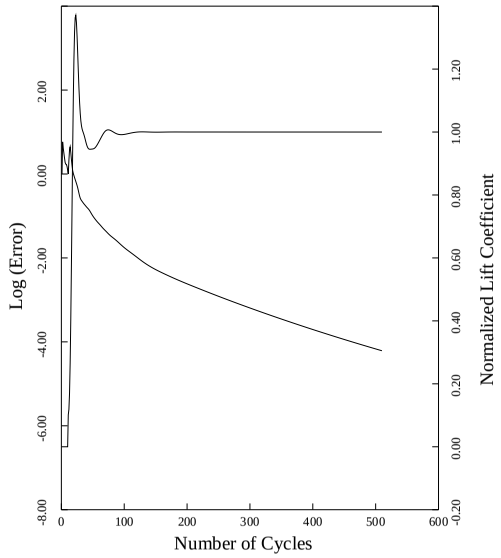

Figure 53: *Density residual and lift coefficient convergence history as a function of multigrid cycles on baseline grid for Mach=0.75, C_L = 0.6, Re = 3 million*

Table 5: *Grids and corresponding run times*

Grid	No. Points	No. Tets	No. Prisms	Memory	Run Time	Hardware
Grid 1	1.65×10^6	2×10^6	3.6×10^6	2.8 Gbytes	2.6 hours	16 Pentium IV 1.7GHz
Grid 1	1.65×10^6	2×10^6	3.6×10^6	2.4 Gbytes	8 hours	4 DEC Alpha 21264 (667MHZ)
Grid 1	1.65×10^6	2×10^6	3.6×10^6	3.0 Gbytes	45 min.	64 SGI Origin 2000 (400MHz)
Grid 2	3.0×10^6	3.7×10^6	6.6×10^6	4.2 Gbyte s	8 hours	8 DEC Alpha 21264 (667MHZ)
Grid 3	13×10^6	16×10^6	28.8×10^6	27 Gbytes	4 hours	128 SGI O2000 (400MHz)

Table 6 compares the computed values at Mach=0.75 and $C_L = 0.5$, on the three different meshes versus experimental data. The drag is seen to be computed relatively accurately by all three grids, although there is a 10.6 count variation between the 3 grids. However, the incidence at which the prescribed $C_L = 0.5$ is achieved is up to 0.6 degrees lower than that observed experimentally. This effect is more evident in the C_L versus incidence plot of Figure 54, where the computed lift values are consistently higher than the experimental values. Since this discrepancy increases with the higher resolution grids, it is most likely not the result of a lack of grid resolution. The slope of the computed lift curve is about 5% higher than the experimentally determined slope, and is largely unaffected by grid resolution.

Figure 55 provides a comparison of computed surface pressure coefficients with experimental values at the experimentally prescribed C_L of 0.6 (where the effects are more dramatic than at $C_L = 0.5$) as well as at the experimentally prescribed incidence of 0.93 degrees, at the 40.9 % span location. When the experimental incidence value is matched, the computed shock location is aft of the experimental values, and the computed lift is higher than the experimental value, while at the prescribed lift condition, the shock is further forward and the suction peak is lower than the experimental values.

This bias in lift versus incidence was observed for a majority of the numerical solutions submitted to the workshop [1], and thus might be attributed to a model geometry effect, a wind tunnel correction effect, or to turbulence modeling effects, although an exact cause has not been determined. When plotted as a drag polar, C_L versus C_D as shown in Figure 56, the results compare favorably with experimental data. The computational results on the baseline grid compare very well with experiment in the mid-range (near $C_L = 0.5$), while a slight overprediction of drag is observed for low lift values, which decreases as the grid is refined.

This behavior suggests an under-prediction of induced drag, possibly due to inadequate grid resolution in the tip region or elsewhere. The absolute drag levels have been found to be sensitive to the degree of grid refinement at the blunt trailing edge of the wing. The drag level is reduced by 4 counts when going from the 1.6 million point grid, which has 5 points on the trailing edge, to the 3 million point grid, which has 17 points on the trailing edge. Additional studies have shown that up to 33 points on the blunt trailing edge are required before the drag does not decrease any further [33]. In the current grid generation environment, and without the aid of adaptive meshing techniques, the generation of highly refined trailing edge unstructured meshes has been found to be problematic, thus limiting the study in this area.

Table 6: *Results for Case 1; Experimental Values 1:ONERA, 2:NLR, 3:DRA, Experimental data and 3 M point grid results are interpolated to specified C_L condition along drag polar.*

Case	C_L	α	C_D	C_M
Experiment[1]	0.5000	$+.192^o$	0.02896	-.1301
Experiment[2]	0.5000	$+.153^o$	0.02889	-.1260
Experiment[3]	0.5000	$+.179^o$	0.02793	-.1371
$Grid1(1.6Mpts)$	0.5004	$-.241^o$	0.02921	-.1549
$Grid2(3.0Mpts)$	0.5000	$-.417^o$	0.02857	-.1643
$Grid3(13Mpts)$	0.5003	$-.367^o$	0.02815	-.1657

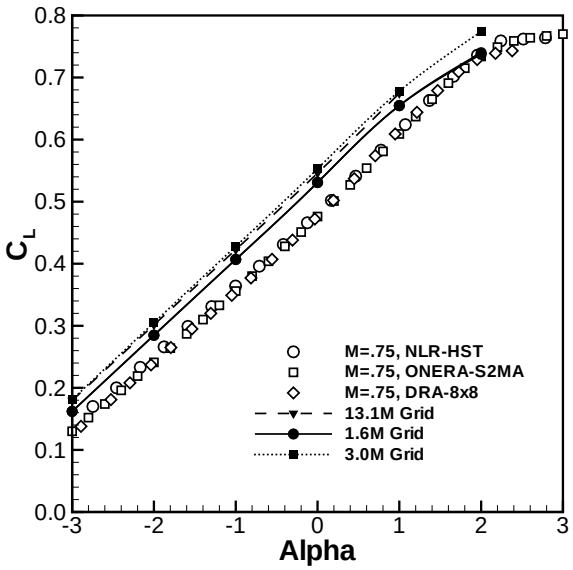

Figure 54: *Comparison of computed lift as a function of incidence for three different grids versus experimental results*

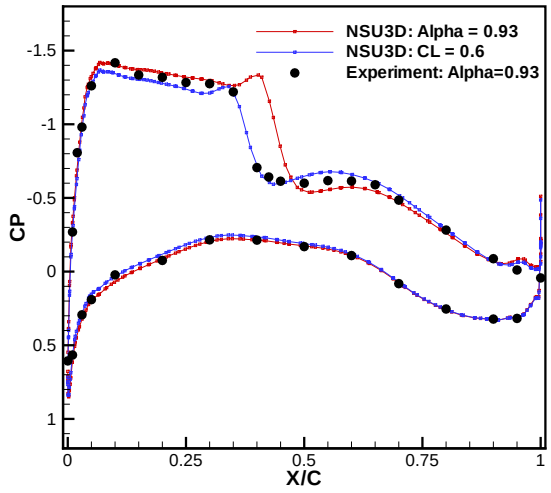

Figure 55: *Comparison of computed surface pressure coefficients at prescribed lift and prescribed incidence versus experimental values for baseline grid at 40.9 % span location*

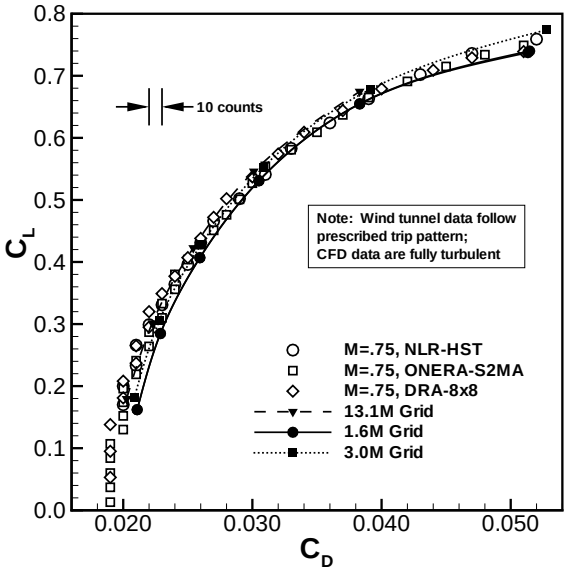

Figure 56: *Comparison of computed versus experimental drag polar for Mach=0.75 using three different grids*

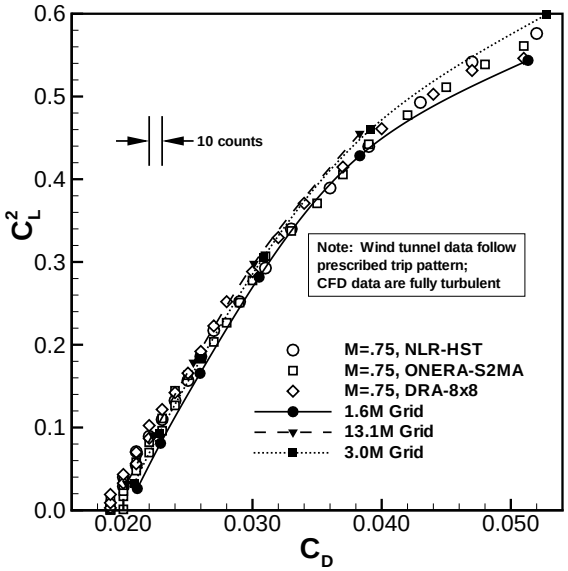

Figure 57: *Comparison of computed versus experimental induced drag factor for Mach=0.75 using three different grids*

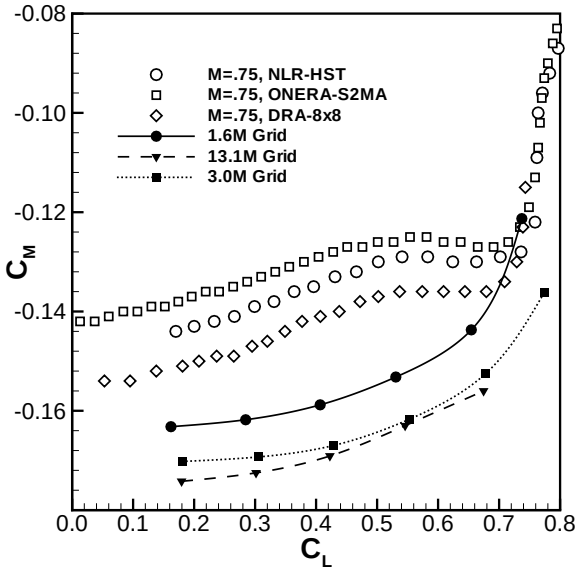

Figure 58: *Comparison of computed versus experimental pitching moment for Mach=0.75 using three different grids*

Figure 57 provides an estimate of the induced drag factor, determined experimentally and computationally on the three meshes. For ${C_L}^2$ up to about 0.36, when the flow is mostly attached, induced drag is underpredicted by approximately 10%, as determined by comparing the slopes of the computational and experimental curves (using a linear curve fit) in this region. Grid refinement appears to have little effect on the induced drag in this region. At the higher lift values, the 3 million point grid yields higher C_L and lower C_D values, which is attributed to a slight delay in the amount of predicted flow separation. Results for the 13 million point grid are not shown at the highest incidence, since a fully converged solution could not be obtained at this condition. It should be noted that the wind tunnel experiments used a boundary layer trip at 15% and 25% chord on the upper and lower surfaces, while all calculations were performed in a fully turbulent mode. Examination of the generated eddy viscosity levels in the calculations reveals appreciable levels beginning between 5% to 7% chord. The exact influence of transition location on overall computed force coefficients has not been quantified and requires further study.

The pitching moment is plotted as a function of C_L in Figure 58 for all three grids versus experimental values. The pitching moment is substantially underpredicted with larger discrepancies observed for the refined grids. This is likely a result of the over-prediction of lift as a function of incidence, as mentioned earlier and illustrated in Figure 54. Because the computed shock location and suction peaks do not line up with experimental values, the predicted pitching moments can not be expected to be in good agreement with experimental values.

Figure 59 depicts the drag rise curves obtained on the baseline grid and the first refined grid (3 million points). Drag values are obtained at four different constant C_L values for a range of Mach numbers. Drag values are predicted reasonably well except at the highest

lift and Mach number conditions. There appears to be no improvement in this area with increased grid resolution, which suggests issues such as transition and turbulence modeling may account for these discrepancies. However, since the two grids have comparable resolution in various areas of the domain, grid resolution issues still cannot be ruled out at this stage.

These results can also be plotted at constant Mach number, as shown in the drag polar plots of Figure 60. The plots show similar trends, with the drag being slightly overpredicted at low lift values on the coarser grid and with the refined grid achieving better agreement in these regions. For the higher Mach numbers, the drag is substantially underpredicted at the higher lift values. These discrepancies at the higher Mach numbers and lift conditions point to an under-prediction of the extent of the separated regions of flow in the numerical simulations. However, the character of the curves also suggest that the error may be due as well to the C_L offset (shown in Figure 54). Additional information concerning the regions of flow separation found in the wind tunnel would be needed to more accurately quantify the nature of the error.

The above results indicate that the NSU3D unstructured mesh Navier-Stokes solver achieves a reasonably good predictive ability for the force coefficients on the baseline grid over the majority of the flow conditions considered. The overall agreement, particularly at the low lift values, is improved with added grid resolution, while the more extreme flow conditions which incur larger amounts of separation are more difficult to predict accurately. On the other hand, the observed bias between computation and experiment in the lift versus incidence values has an adverse affect on the prediction of pitching moment. While the source of this bias is not fully understood, it was observed for a majority of independent numerical simulations undertaken as part of the subject workshop [1] and can likely be attributed to geometrical differences, wind tunnel corrections, or turbulence modeling effects.

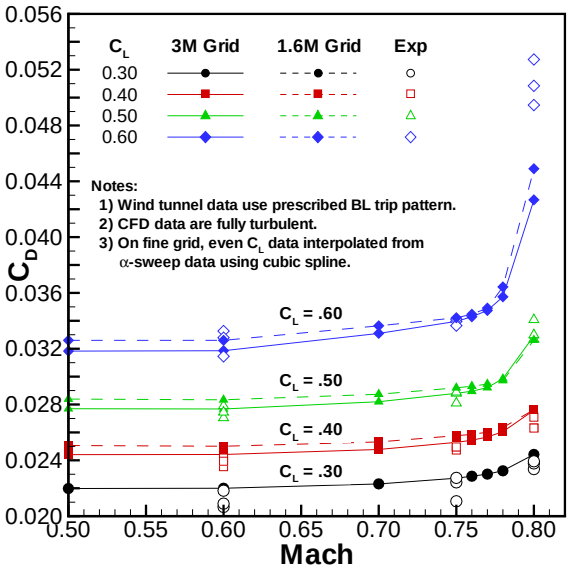

Figure 59: *Comparison of computed versus experimental drag rise curves for three different C_L values on two different grids*

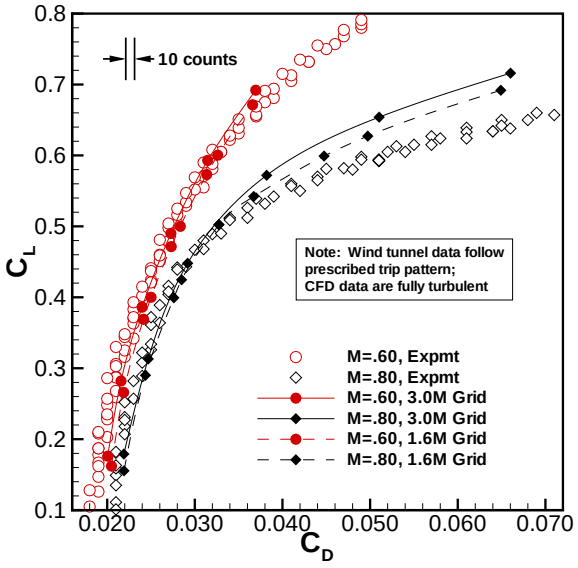

Figure 60: *Comparison of computed versus experimental drag polars for Mach=0.6 and Mach=0.8 on two different grids*

A general assessment of the relative merits of unstructured versus block structured and overset methods can be surmised by a compilation of all the workshop results involving the various different methodologies. The workshop results include submissions from 13 different individual RANS codes, of which six were based on unstructured meshes, six on block-structured meshes, and one on overset meshes. Figure 61 shows plots of idealized profile drag for each of the major code types. Idealized profile drag is defined by the formula [99]:

$$C_{DP} = C_D - C_L{}^2/(\pi AR) \tag{18}$$

where AR is the wing aspect ratio. Plotting C_{DP} generally results in a more compact representation of the data, allowing more expanded scales. It also highlights the characteristics at higher C_L, where the drag polar becomes non-parabolic due to wave drag and separation. The two methods with the most results (block-structured and unstructured) both have considerable scatter, overpredict basic drag levels, and have one or two outliers. The block-structured results contain slightly more scatter than the unstructured results, but represent a wider variety of different codes and turbulence models. Overall, the results from this workshop indicate that well formulated structured grid and unstructured grid methodologies can achieve equivalent predictive ability for drag on cases of practical interest.

Direct efficiency comparisons are complicated by the multitude of different solvers and hardware used for the given test cases. The NSU3D study involved 72 steady-state baseline grid cases which were processed in about one week on a 16 cpu Pentium cluster. For the largest grid, fewer cases were run using a larger SGI Origin 2000 machine with 128 processors. The required resources reported in Table 5 are well inline and in some cases superior to the costs reported for the various structured grid solvers.

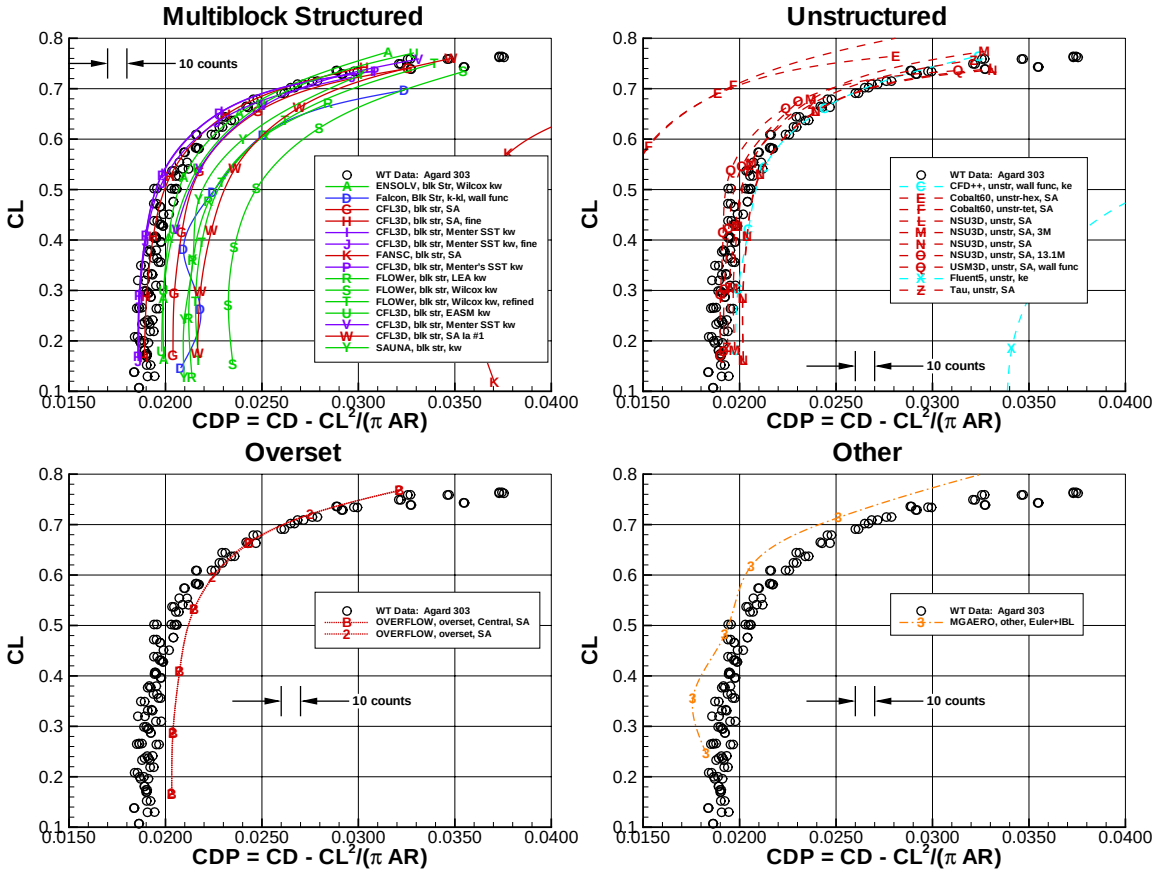

Figure 61: *Comparison of drag prediction performance for DLR-F4 wing-body test case based on methodology (reproduced from Ref [15])*

5.2 Incremental Effects: Engine Installation Drag

While the consistent prediction of absolute cruise drag to within 1 drag count may currently be beyond the practical capabilities of most structured and unstructured meshes solvers, the determination of incremental effects represents a less stringent problem which is of great importance in the design exercise of any aircraft program. The ability to accurately predict changes in drag due to design variations is crucial to guide the selection of competing design configurations, and is a prerequisite for the successful use of any numerical design-optimization procedure.

Prediction of engine airframe interference effects is of particular importance for transport aircraft. With the trend towards higher-bypass ratio power-plants, minimization of installation drag becomes a more difficult task, and optimal nacelle-pylon integration strategies require robust and accurate installation drag prediction capabilities. While extensive validation of block-structured and overset-structured mesh methods has been performed in this area over the last decade, these methods are often plagued by extensive setup time for modeling the complex geometries associated with nacelle-pylon-wing configurations in close

proximity to each other. Unstructured meshes provide a definite advantage in this area, although the predictive ability of these methods is still in question and the necessary validation required to certify unstructured mesh methods for the design process is only beginning to be addressed.

A recent study at the German Aerospace Center (DLR) [52] has demonstrated the potential of an unstructured mesh solver in predicting installation drag. In this work, the installation drag and its variation with nacelle position and shape were studied both computationally and experimentally in the transonic cruise regime on a DLR-F6 configuration. The DLR-F6 is a twin engine configuration derived from the previously mentioned DLR-F4 geometry. It contains a wing aspect ratio of 9.5 and a leading edge sweep angle of 27.1 degrees, and the engines are modeled as flow-through nacelles. For a Mach number of 0.75, and a Reynolds number of 3 million, the study was confined to the range of lift conditions from C_L=0.1 to C_L=0.6. The various geometric configurations are defined in Table 7, with the definition of the nacelle positioning coordinates illustrated in Figure 62.

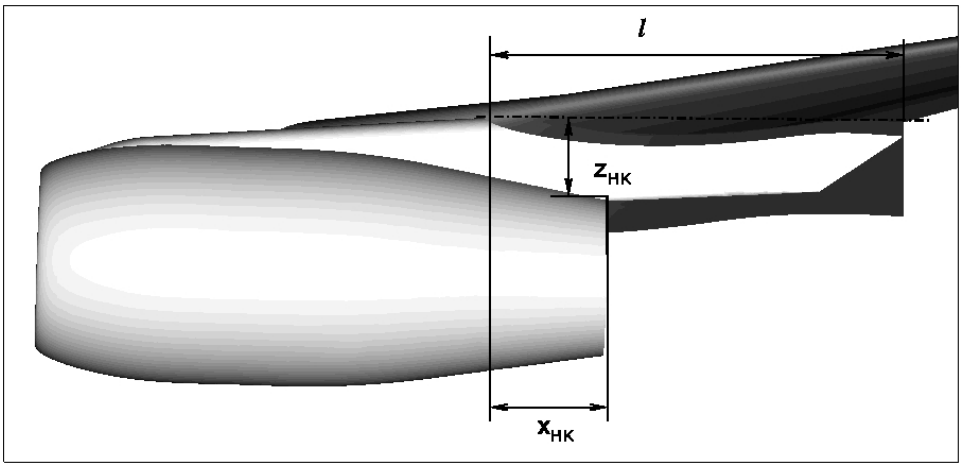

Figure 62: *Definition of parameters describing nacelle position for engine installation drag study. (Reproduced with permission from Ref [52])*

Table 7: *Nacelle Locations (from Ref [52])*

Nacelle	X_{HK}/L	Z_{HK}/L
CFM56-long, position 1	0.49	-0.189
CFM56-long, position 2	0.30	-0.189
CFM56-long, position 3	0.30	-0.250

The DLR **tau** unstructured mesh solver [35, 100, 101] consists of a vertex-based discretization with added artificial dissipation operating on hybrid prismatic-tetrahedral element meshes, and employs an agglomeration multigrid method for accelerating convergence to steady-state. The Spalart-Allmaras turbulence model [69] is used exclusively in this study,

and transition is set to correspond to the specified trip locations in the wind tunnel experiments. This solver is similar in many respects to the aforementioned NSU3D solver [32, 61], although it makes use of a modified form of scalar artificial dissipation as opposed to the matrix form used by NSU3D. A limited adaptive meshing capability is also incorporated, and three levels of adaptive refinement are used for all steady-state solutions, resulting in a near doubling of the number of grid points between the initial and final adapted meshes. Initial mesh resolutions are of the order of 4.5 million points and the final adapted meshes contain of the order of 7.5 million points for the wing-body-pylon-nacelle cases, with the wing-body alone cases comprising 2.9 million and 5.5 million points for initial and final adapted grids respectively.

Figure 63 shows a plot of lift versus incidence for the three nacelle positions and the wing-body alone configuration as reported by the numerical simulations and the wind-tunnel experiments. As in the previous cases, lift as a function of incidence is overpredicted, resulting in a near constant angle shift between the numerical and experimental results. When plotted in drag polar format, as shown in Figure 64, the numerical results are seen to underpredict the drag by approximately 16 counts for all configurations. However, since this offset is reasonably uniform for all cases, the small differences between varying engine positions can be predicted to within an accuracy of less than 2 counts for these lift conditions. Figure 65 shows the comparison of numerically predicted and experimentally observed installation drag, which is defined as:

$$C_{Dinstall} = C_{Dwith\ engine} - C_{Dclean} - C_{Dinternal} \tag{19}$$

where all quantities are obtained both experimentally and numerically. The internal nacelle drag values were measured from calibration tests to be $C_{Dinternal} = 0.00116$ and calculated numerically to be $C_{Dinternal} = 0.00115$. The comparison in Figure 65 demonstrates that the values of $C_{Dinstall}$ and their variations with lift conditions and nacelle positions can be predicted accurately by the unstructured mesh approach. The numerically predicted installation drag values are within 1 to 4 counts of the experimental values, which is within the range of experimental deviations between various wind-tunnel experiments, and the small changes between configurations which are of the order of several drag counts are accurately reproduced, in spite of the fact that the absolute drag numbers are underpredicted for all cases (c.f. Figure 64). The effect of grid convergence on this predictive ability is illustrated in Figures 66 and 67, where the experimental data at C_L=0.5 is compared with numerical drag for three different levels of adaptive meshing refinement. While the offset between numerical and experimental absolute drag values actually increases with additional grid refinement, as seen in Figure 66, the prediction of the installation drag numbers improves with increasing grid resolution, as seen from Figure 67.

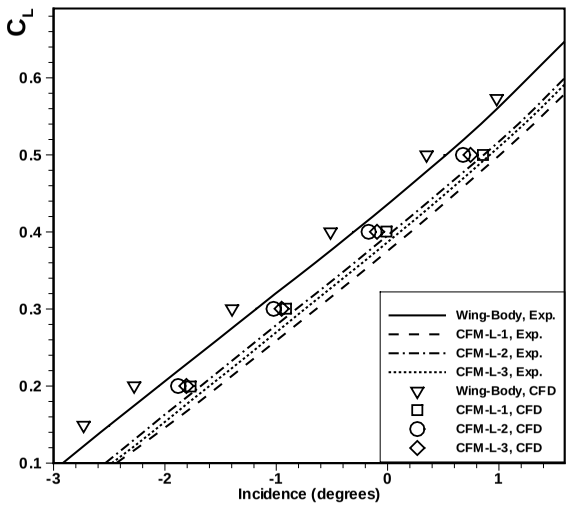

Figure 63: *Comparison of computed and experimental lift as a function of incidence for unpowered configuration and three nacelle positions. (Reproduced with permission from Ref [52])*

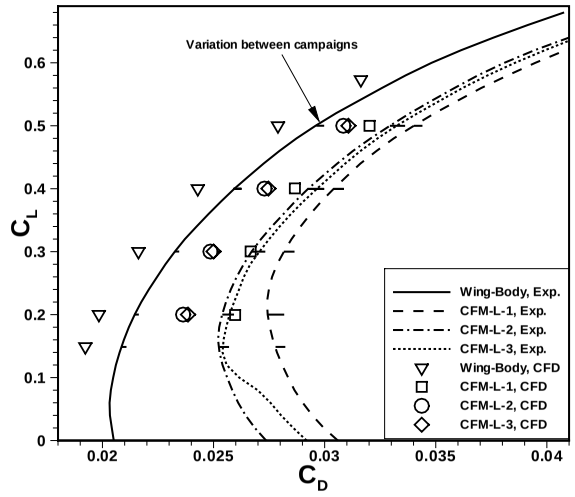

Figure 64: *Comparison of computed and experimental absolute drag polars for unpowered configuration and three nacelle positions. (Reproduced with permission from Ref [52])*

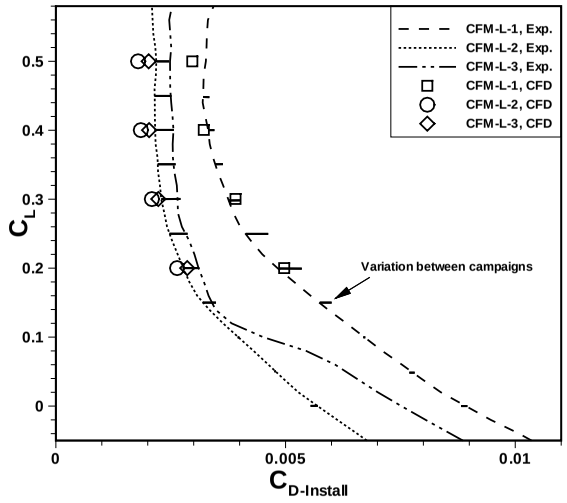

Figure 65: *Comparison of computed and experimental installation drag polars for unpowered configuration and three nacelle positions. (Reproduced with permission from Ref [52])*

These results indicate that installation drag can be predicted to within 1% accuracy compared with wind tunnel data, although total drag is only predicted to within 6% to 8% accuracy. Hence, although a systematic deviation between overall experimental and computational results remains, even with increasing grid resolution, differences of 1 to 2 drag counts in the installation drag for varying engine locations and sizes can clearly be analyzed.

While these results demonstrate the effectiveness of a particular unstructured mesh methodology for an important and demanding aerodynamic problem, this level of fidelity can only be obtained through a concentrated and sustained validation effort involving both experimental data and numerical method development. The AIAA 2nd Drag Prediction Workshop, to be held in June 2003, will be based on the DLR-F6 configuration, and will afford a broader comparison of various CFD solvers and methodologies for predicting installation drag, including grid resolution effects.

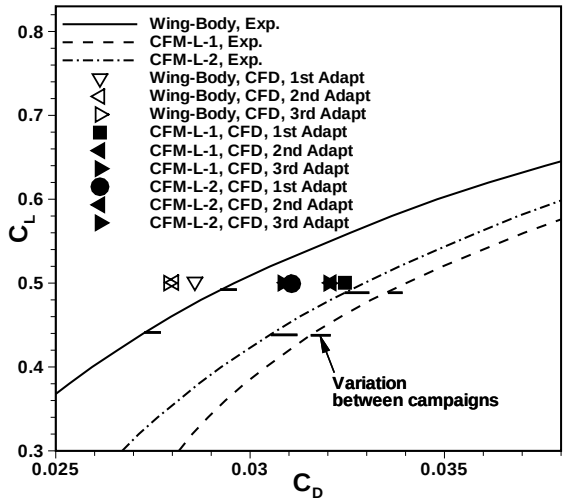

Figure 66: *Effect of grid resolution through sequential levels of mesh adaptation on absolute drag polars for unpowered configuration and three nacelle positions. (Reproduced with permission from Ref [52])*

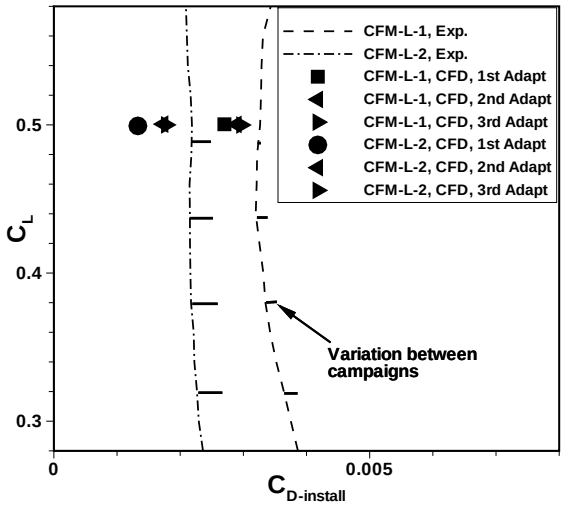

Figure 67: *Effect of grid resolution through sequential levels of mesh adaptation on installation drag polars for unpowered configuration and three nacelle positions. (Reproduced with permission from Ref [52])*

5.3 High Lift Flows

Accurate numerical simulation of high-lift flows represents perhaps the most challenging area of steady-state computational aerodynamics. The multiple slats, flaps, and fairings associated with high-lift configurations constitute an additional level of geometric complexity over cruise configurations for transport aircraft. In fact, it is the complexity of high-lift configurations that has been one of the principal motivating factors in the development of unstructured mesh approaches for aerodynamic problems [47, 102, 65]. At the same time, the physics of flows over high-lift configurations are considerably more complicated than transonic cruise flow physics.

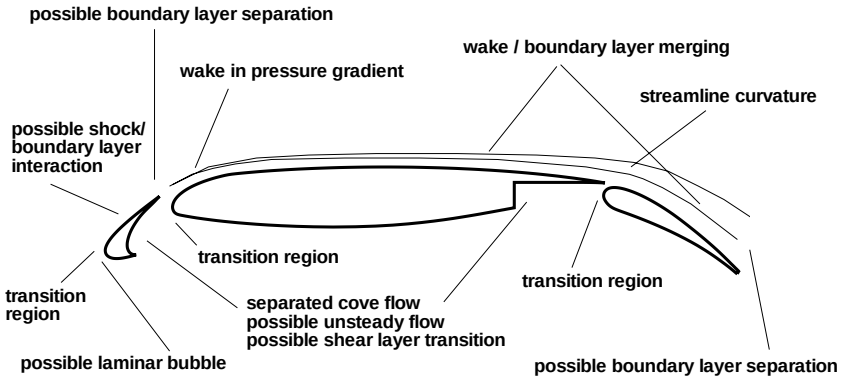

Figure 68: *Illustration of important high-lift flow physics phenomena. (Reproduced from Ref [67])*

Figure 68, reproduced from reference [67], illustrates some important high-lift flow physics characteristics. These include flow separation in the cove regions of the slat and main airfoils, with possible unsteady effects, compressible effects with possible shock boundary-layer interaction on the slat, confluent boundary-layers and wakes on the upper surface of the main and flap elements, transition effects, and adverse pressure gradient separation on the flap and aft portion of the main airfoil.

Contrary to most cruise configurations, many high-lift landing configurations are designed to operate with significant regions of on-body or off-body separated flow. These characteristics represent formidable challenges for accurate numerical simulations of any type. The complicated physics associated with high-lift flows means more sophisticated transition and turbulence models will likely be required for the accurate simulation of such flows. The ability to adequately resolve all important flow features through proper local grid resolution is also a key requirement for successfully predicting high-lift flows. Unstructured mesh methods are well suited for handling the complex geometries associated with high-lift devices, and offer the possibility of using adaptive meshing techniques for addressing the grid resolution issue. On the other hand, unstructured mesh approaches have generally lagged structured mesh methods in the implementation of new turbulence models, and the potential of adaptive meshing has yet to be fully exploited.

Figures 69 through 73 illustrate a highly resolved numerical three-dimensional high-lift study performed using the NSU3D solver [61]. The configuration consists of a twin-engine transport known as the energy efficient transport (EET), which has been tested both as a full span and semi-span model in the NASA Langley $14 \times 22ft$ subsonic wind-tunnel [103]. The geometry studied in this work contains no pylon or nacelle. The wing has an aspect ratio of 10, a leading edge sweep of 28.8 degrees, and consists of a super-critical airfoil section with a slat and double slotted flap. This case consists of a take-off configuration, with a slat deflection of -50 degrees, a vane deflection of 15 degrees, and a flap deflection of 30 degrees, with respect to the main airfoil. The freestream Mach number is 0.2, and the Reynolds number is 1.6 million based on the wing reference chord, and the experimental flow incidence varies over a range of -4 degrees up to 24 degrees.

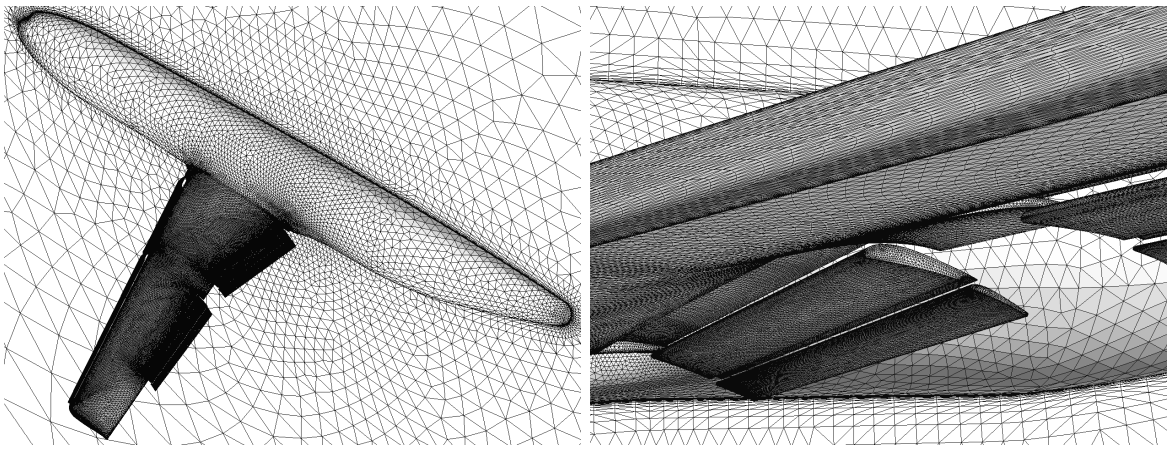

Figure 69: *Illustration of 3.1 million point mesh for three-dimensional high-lift configuration*

Experimental results are available in the form of force and moment coefficients as a function of angle of attack, and chordwise pressure distributions at three spanwise locations. While reference [103] describes the experiments on a semi-span model with pylon and nacelle, the full-span "nacelle-off" results used for comparison herein have not been published separately.

The computations are all performed at zero yaw angle, and therefore only include one half of the symmetric aircraft geometry, delimited by a symmetry plane. The baseline unstructured grid for this geometry was generated using the VGRIDns software package [54] developed at NASA Langley. Figure 69 illustrates the baseline mesh, which contains a total of 3.1 million vertices and 6.6 million tetrahedra, 3.9 million prisms, and 46,899 pyramids, after merging of the boundary layer tetrahedra into prismatic elements. A finer grid of 24.7 million points was also generated, by uniform subdivision of each cell in the baseline mesh into 8 smaller self-similar cells, using a global mesh refinement technique [57]. The position of the additional surface points in the refined mesh were obtained from linear interpolation of the surrounding baseline mesh surface-point coordinates, and therefore do not match the original surface geometry description exactly. A capability for reprojecting these points to the original surface definition is currently under development.

Figure 70 illustrates the convergence history obtained for a typical high-lift calculation. The residuals are reduced by approximately four orders of magnitude over 500 multigrid cycles, for both the baseline and the fine meshes, as shown in the figure, illustrating the grid independence property of the multigrid algorithm. The baseline mesh cases were run on 128 processors of an SGI Origin 2000 which required 80 minutes of will clock time, while the fine grid cases were run on a large CRAY T3E using up to 1450 processors, which required one hour of wall clock time.

Figure 71 provides a comparison of the computed surface pressures on the coarse and fine grids with experimental values at one of the three spanwise locations at which experimental data is available, for an incidence of 10 degrees. The differences between the coarse and fine grid values are rather small, with the fine grid computations providing slightly higher suction peaks at the main and flap leading edges. Both computational results compare favorably with experimental values at all three stations.

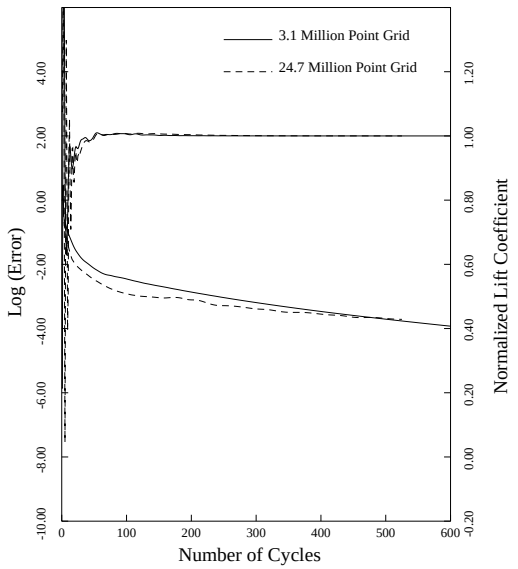

Figure 70: *Observed convergence rates for coarse and fine grids for three-dimensional high-lift case demonstrating grid independence property of multigrid*

A comparison between computed and experimental lift coefficients as a function of angle of attack is given in Figure 72 for both grids. As expected, the fine grid produces slightly higher lift coefficients than the coarse grid. The experimental lift values are over-predicted by both fine and coarse grid results. However the slope of the lift curve is reproduced very accurately by both computations. The maximum lift point, which experimentally occurs at 16 degrees incidence, is well predicted by the coarse grid. The fine grid over-predicts the maximum lift incidence by 1 degree, giving a value of 17 degrees. In both cases the value of C_{Lmax} is over-predicted.

After stall, the computations fail to converge adequately, producing large variations in the lift coefficients. The average, as well as the minimum and maximum of these computed lift coefficients are plotted in the figure. The post-stall computed values averaged in this manner follow the experimental values fairly closely, although the range of computed min-max values is rather large.

The drag results are presented in polar format in Figure 73. The baseline grid results overpredict the drag by approximately 10 counts, whereas the fine grid values are within several counts of the experimental values. While this level of absolute drag prediction is encouraging, it must be considered at least partly fortuitous, in view of the transonic cruise drag prediction capabilities discussed earlier, and from the documented capabilities of two-dimensional high-lift numerical experiments discussed below. Furthermore, this high-lift take-off configuration represents a simpler test case than landing configurations which involve higher flap deflections and more potential for flow separation. Results of this type must be considered as a feasibility study for unstructured meshes in three-dimensional high-lift, and as a first step towards a comprehensive development and validation effort required for the establishment of a reliable high-lift numerical predictive ability.

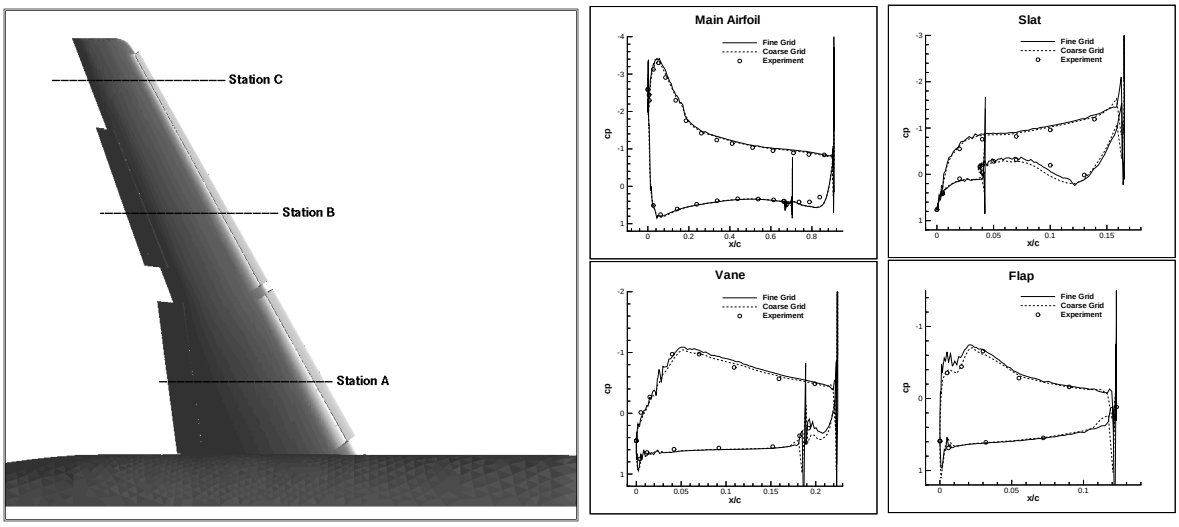

Figure 71: *Illustration of spanwise locations of experimental pressure data and comparison of computed surface pressures with experimental data at (center) station B*

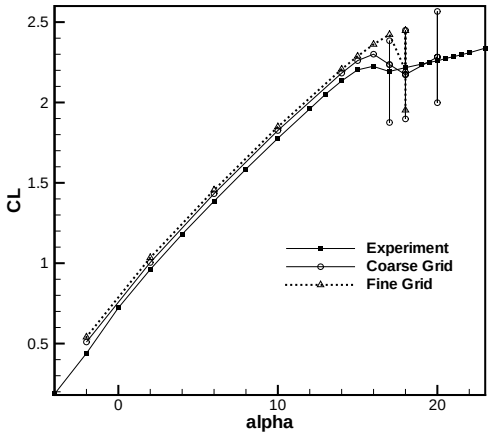

Figure 72: *Comparison of computed lift as a function of incidence on coarse and fine grids versus experimenta l results for 3D high-lift configuration*

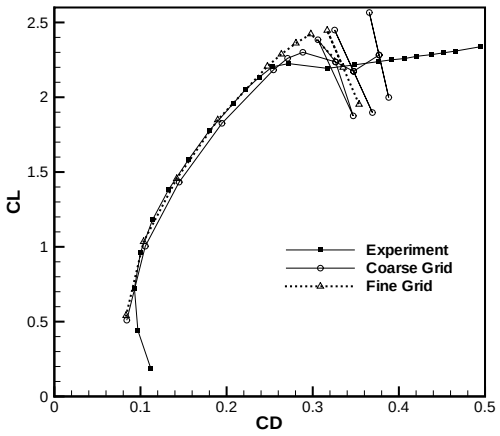

Figure 73: *Comparison of computed drag polars on coarse and fine grids versus experimental results for three- dimensional high-lift configuration*

Considerably more experience has been developed in the two-dimensional setting for assessing the predictive ability of unstructured mesh high-lift computations. In two dimensions, more elaborate experimental data has been obtained, including surface pressure and skin friction measurements, as well as off-body velocity profiles and even Reynolds stress measurements in boundary layer and wake regions [104]. The computational aspects of two-dimensional high-lift problems can also be studied in more detail, for example by using very fine grid resolution which is currently unaffordable in three dimensions, and rapidly testing out various turbulence and transition models. On the other hand, two dimensional experiments have limitations, since at high-lift conditions near stall, even the best two-dimensional experimental setups have been shown to contain appreciable three dimensional effects [105].

An excellent review of the state-of-the-art in numerical high-lift prediction is given by Rumsey and Ying [67], including structured and unstructured grid methods, in both two dimensions, and three dimensions. No particular accuracy benefit between structured or unstructured grid methods is reported in the survey. For two-dimensional problems, grid resolutions of the order of 50,000 points for three-element airfoils were found in many cases to be sufficient to produce good surface pressure and skin friction predictions, but finer grids of up to 200,000 points are required for accurate force calculations, especially near the maximum lift point. The requirement of adequately resolving all wakes with off-body grid refinement has also been demonstrated in various two-dimensional numerical studies [106]. This level of overall resolution in general, and wake resolution in particular, is most often not achieved in reported three-dimensional calculations, and will require future advances in grid generation/adaptivity, and solver/hardware efficiency in order to become commonplace in three dimensions.

Well resolved two-dimensional unstructured mesh simulations have generally been found to predict overall force coefficients reasonably well for high-lift configurations without large areas of separated flows. Good prediction of incremental effects, such as Reynolds number effects and flap positioning has also been demonstrated. Figure 74 illustrates the surface pressure comparison for a two-dimensional high-lift study reproduced from reference [107]. In Figure 75, experimental lift values for a three-element airfoil configuration are compared with numerical values computed using the NSU2D unstructured mesh solver [68]. In this comparison, which is reproduced from reference [107], the lift is slightly overpredicted, although the slope of the lift curve is well reproduced. The location of the maximum lift point is also slightly overpredicted, although this may be due to three-dimensional effects in the experiment, as has subsequently been investigated [105]. Comparisons of off-body velocity profiles are given in Figure 76 for a similar configuration, reproduced from [65]. These results illustrate the potential for predicting important flow physics details, provided adequate grid resolution, accurate discretizations, and suitable turbulence modeling are employed. The exception is the slat wake deficit, which is consistently over-predicted by many computational methods [67], the causes of which are still not well understood.

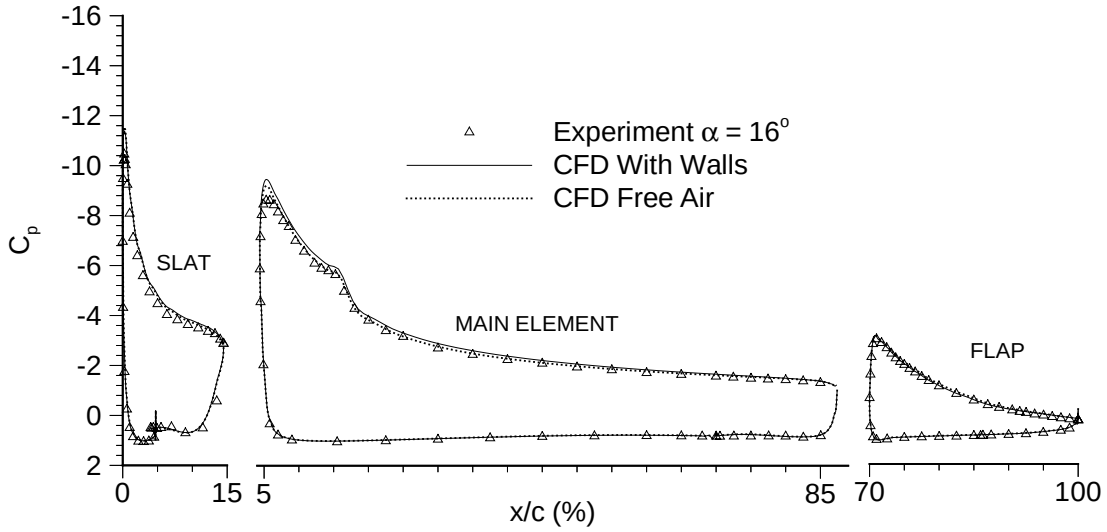

Figure 74: *Comparison of computed and experimental surface pressures for three-element high-lift airfoil configuration in two-dimensions including effects of wind tunnel walls (Reproduced from Ref [107]).*

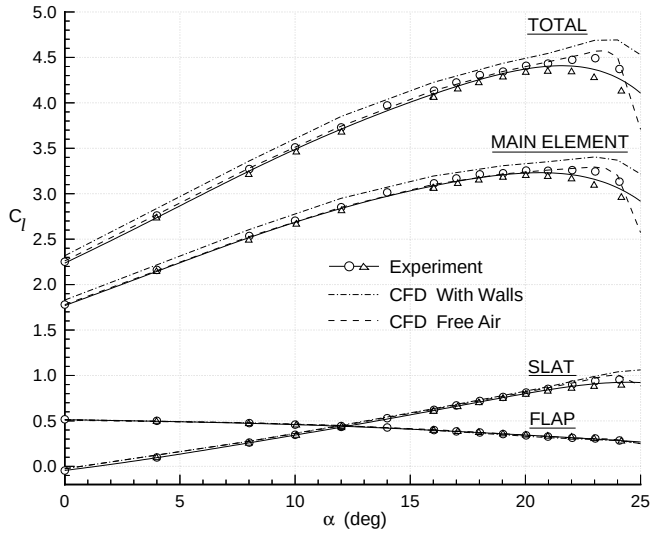

Figure 75: *Comparison of computed and experimental lift coefficients for three-element high-lift airfoil configuration in two-dimensions including effects of wind tunnel walls (Reproduced from Ref [107]).*

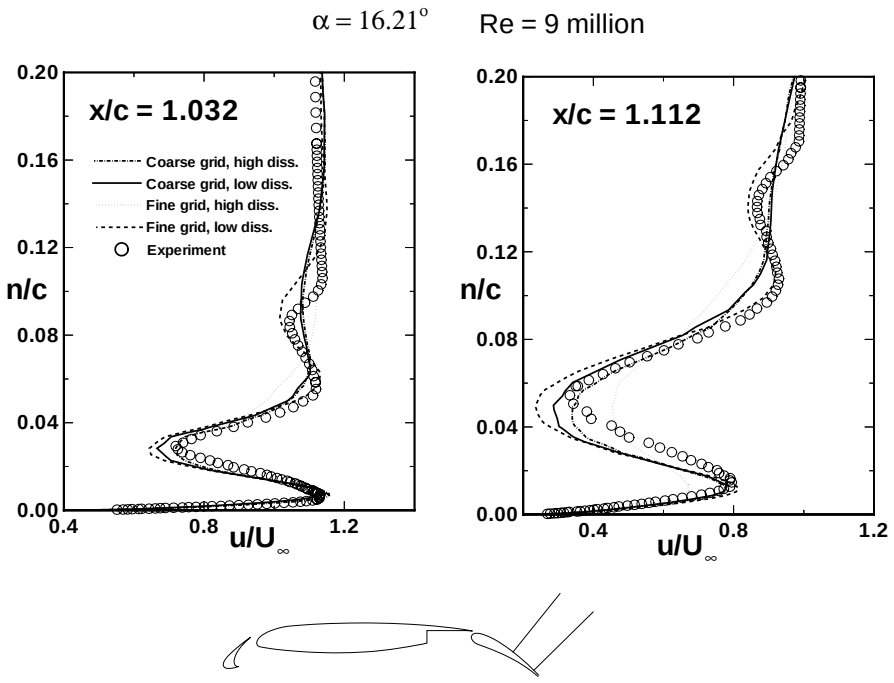

Figure 76: *Comparison of computed off body velocity profiles with experimental values and effect of grid resolution and dissipation levels (Reproduced from [65]).*

The ability of two-dimensional unstructured mesh high-lift simulations in predicting variations with respect to Reynolds number has generally been found to be acceptable [67, 108, 107], in the absence of important transitional effects. Figure 77 illustrates a typical level of agreement obtained between computed and experimentally measured lift increments produced by increasing the Reynolds number from 5 million to 9 million, for a similar three-element airfoil, reproduced from [65]. In these cases, no transition model or specification is employed, and the calculations are performed as fully turbulent. Transitional effects are well known to have the potential for influencing overall high-lift performance. The successful incorporation of various transitional models and specification methods has been reported in the literature [67], and this practice can be expected to become more commonplace as increased simulation fidelity levels are sought.

Sensitivity due to geometry changes, such as flap deflection and gap and overlap studies, has been mixed, with good predictions reported in some studies, while deficiencies have been reported elsewhere [67]. Figure 78 illustrates the change in lift predicted by the NSU2D solver compared with experiment for a small change in gap of 0.25% chord. The comparison shows the trends are well captured at the low and high lift conditions, but the loss of lift in the region in between these two extremes is not captured numerically. In this case, the lift degradation is due to a small separation region which only appears on the flap at these intermediate lift conditions, at the larger gap setting. The discrepancy is due to the failure of the numerical method to predict separation in this range, either due to grid resolution issues, or turbulence modeling.

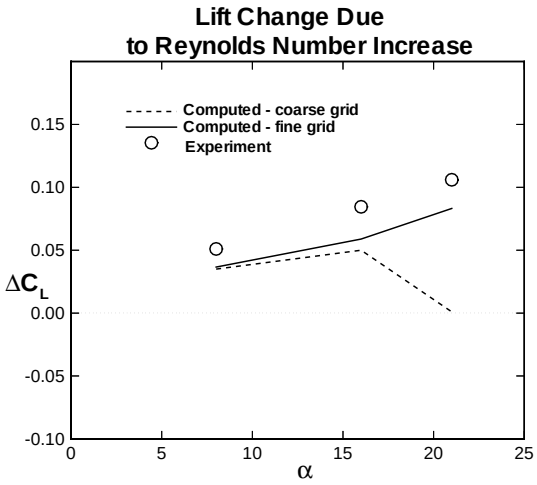

Figure 77: *Comparison of computed and experimental changes in lift due to Reynolds number variation for three-element high-lift airfoil configuration (Reproduced from Ref [65]).*

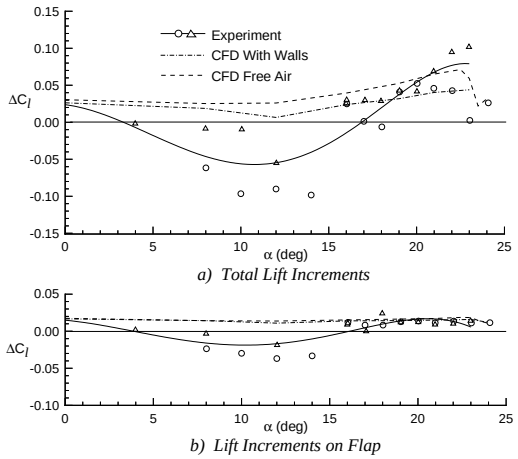

Figure 78: *Comparison of computed and experimental changes in lift due to change in flap gap for three-element high-lift airfoil configuration in two-dimensions including effects of wind tunnel walls (Reproduced from Ref [107]).*

The experience from two-dimensional studies indicates that with current unstructured mesh solvers, when adequate grid resolution is employed, and the basic flow physics phenomena are captured by the transition and turbulence modeling of the simulation approach, good absolute and incremental predictive ability can be achieved. Effort must now be devoted to transferring the experience gained in two-dimensional studies into the three-dimensional setting, as well as continued investment in basic flow physics modeling efforts. Three-dimensional numerical results of the type shown in this section (c.f. Figures 71 through 73) must be considered as a feasibility study for unstructured meshes in three-dimensional high-lift, and as a first step towards a comprehensive development and validation effort required for the establishment of a reliable high-lift numerical predictive ability, drawing on the two-dimensional experience base.

6 Conclusions and Future Work

Throughout the course of these notes, three areas have been emphasized as pacing items in the development and improvement of drag prediction capabilities: grid resolution, flow physics, and validation. While the issues surrounding flow physics and validation requirements apply generally to all numerical methodologies, particular aspects of grid resolution issues are specific to unstructured mesh approaches.

As stated earlier, grid resolution issues are perhaps the most important consideration in achieving an accurate drag prediction capability. However, grid converged solutions are almost never achieved, and considerable user expertise is required in tailoring meshes with adequate resolution in all important regions of the domain. Unstructured mesh approaches are well suited for the use of adaptive meshing techniques, which can ensure adequate resolution in all regions of the domain, by refining and coarsening the mesh locally as the solution evolves. However, successful application of adaptive meshing requires techniques for estimating local discretization error, in order to guide the refinement process. While the mechanics of refining and coarsening unstructured meshes are reasonably well understood today, it is the lack of reliable refinement criteria or error estimators which has held back the potential of adaptive meshing. Most error estimates assume that the solution is asymptotically close to the converged result, an assumption which may not be valid for non-linear problems such as fluid flow, where entire flow patterns(i.e. a separation region) may not be present until sufficient grid resolution is achieved. Because of these difficulties, adaptive meshing strategies have seldom been exploited for demanding problems such as drag predictions. Notable exceptions include the TRANAIR code developed at Boeing [9], and the DLR tau unstructured mesh solver [35, 52] in which an adaptive meshing capability has been incorporated as part of the production process. However, even these cases rely largely on heuristic refinement criteria, which are supported by extensive validation and calibration for the specific types of problems at hand.

The field of drag prediction could benefit enormously from the development of a robust adaptive meshing capability, based on precise discretization error estimates capable of automatically delivering grid converged solutions to a prescribed error tolerance. Rather than attempting to quantify the discretization error at every point in the mesh for adaptive refinement criteria, an alternative approach involving adjoint-based error prediction methods has been gaining popularity recently [109, 110, 111]. In this approach, solution of the adjoint of the governing flow equations is used to quantify the sensitivity of an objective function with respect to local grid resolution throughout the domain. Adjoint sensitivity techniques have been well developed in the context of design-optimization problems [112, 113]. This approach to mesh refinement is non-local, in the respect that it provides for the global effect of local resolution changes. Objective functions representing engineering quantities of interest such as lift, drag, of L/D can be constructed, thus enabling the mesh refinement process to target quantities of interest. Since vastly different grid resolution distributions are required for different enginnering applications of the same problem (i.e. drag prediction versus sonic boom prediction), this approach should enable accurate and efficient drag prediction results by avoiding excessive resolution in areas of little influence on overall drag.

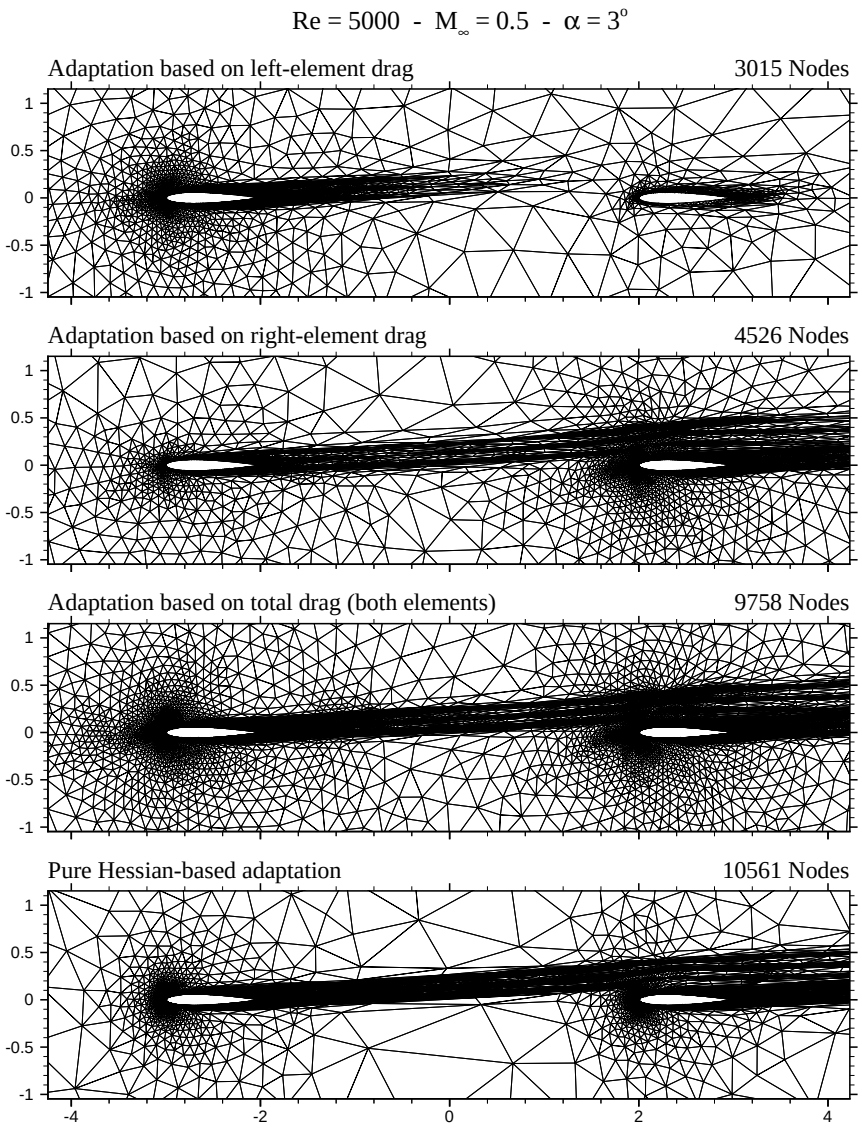

Figure 79: *Illustration of adjoint-based mesh adaptation criterion using different objective functions. (Reproduced with permission from Ref [114])*

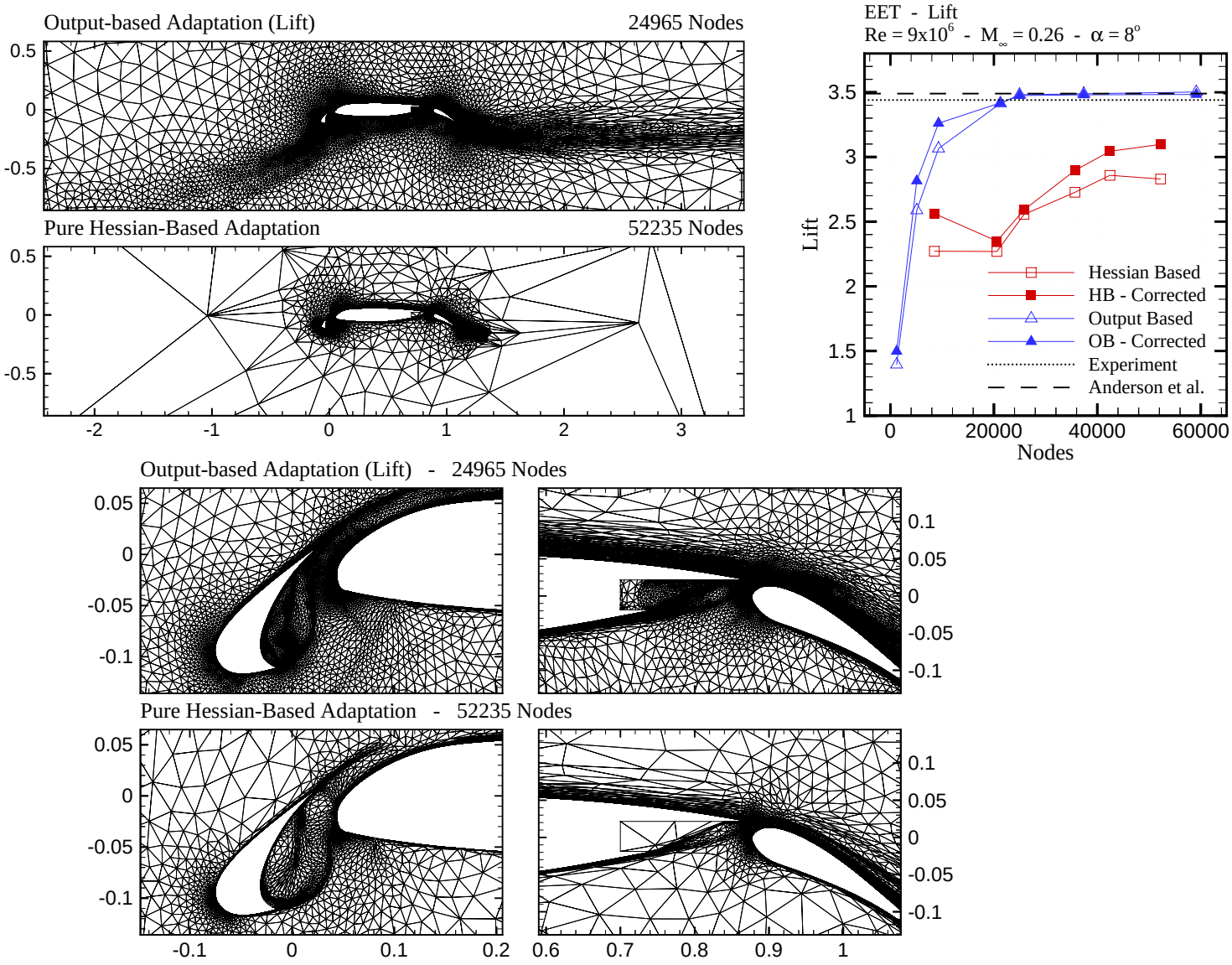

Figure 80: *Illustration of grid convergence properties of adjoint-based mesh adaptation for two-dimensional high-lift problem. (Reproduced with permission from Ref [114])*

An example of adjoint-based error prediction is given in Figures 79 and 80, reproduced from reference [114]. The first part of Figure 79 depicts a mesh produced by prescribing an objective function defined as the left element airfoil drag alone, while the second part depicts the mesh produced for an objective function based on the right airfoil element drag. The third grid was obtained considering the drag on both airfoils in the adjoint-based approach and the fourth grid was obtained using Hessian-based adaptation which looks at local gradients in the solution. In the first case, the wake of the upstream element is not fully resolved, since it has little influence on the drag of the upstream element, while in the second case, it must be well captured due to its influence on the drag of the downstream element. In all cases, the computed drag is more accurate and achieved at lower cost than in the Hessian-based adaptation strategy. Figure 80 illustrates the resulting mesh for a high-lift configuration, and convergence of the lift as the grid is refined both using the adjoint-based approach, and the Hessian-based approach. Using a globally refined mesh solution as the accuracy baseline, the adjoint-based refinement approach is seen to approach the grid converged value much more rapidly than the Hessian-based approach, which may converge to an incorrect solution due to inadequate grid resolution in critical areas undetected by the Hessian refinement criterion. The potential of this type of approach for drag prediction studies is evident, and their implementation and validation in three-dimensional Reynolds-averaged Navier-Stokes unstructured mesh solvers is currently under development [111].

While increasing grid resolution represents one approach to improving solution accuracy, the implementation and use of higher-order discretizations represent an alternative strategy for improving accuracy. Higher-order accuracy discretizations provide more rapid asymptotic convergence of the discrete solution with increasing levels of grid refinement. This is particularly important for precise three-dimensional simulations, where increasing grid resolution can become prohibitively expensive. For the current situation involving second-order accurate discretizations in three-dimensions, a doubling of the grid resolution in all coordinate directions results in an eight-fold increase in overall work, but only delivers a four-fold increase in solution accuracy. By contrast, a fourth-order accurate discretization would result in a 16-fold increase in solution accuracy when doubling the grid resolution. The drawbacks of higher-order discretizations are that they are considerably more expensive to evaluate than lower order discretizations on equivalent grids, and result in more dense discrete equation systems which are more challenging to solve efficiently. Hence, for moderate accuracy levels, high-order accurate discretizations have often been found to be non-competitive. However, due to their superior asymptotic behavior, as illustrated in Figure 81, as the accuracy requirements are increased, these methods become increasingly competitive. The high accuracy requirements of drag prediction studies, coupled with the high level of grid resolution required for suitable accuracy, should be seen as an indication that higher-order discretizations may have a role to play in increasing the efficiency and accuracy of such simulations in the future. The development of Streamwise Upwind Petrov Galerkin (SUPG) methods [42, 115] and Discontinuous Galerkin methods [116, 117] has demonstrated the potential of these techniques for fluid dynamics problems, although applications to aerodynamics has been generally lacking.

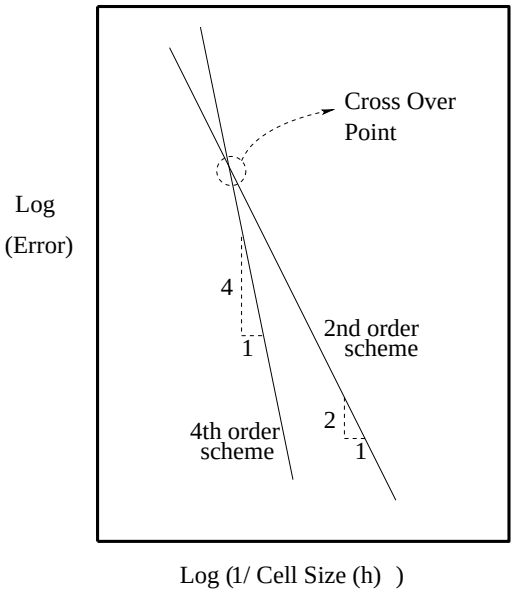

Figure 81: *Illustration of Accuracy Properties of a 2nd and 4th Order Discretization as a Function of Grid Resolution*

Higher-order methods are best suited for problems were the solutions are smooth, and the treatment of discontinuities, either as flow phenomena (i.e. shocks) or geometrical characteristics such as sharp trailing edges represent additional difficulties for such methods [118]. These difficulties may be overcome by reducing the order of the discretization in the vicinity of discontinuous features, and resorting to traditional mesh refinement for greater local accuracy in these regions. This leads to the h-p refinement concept, where higher-order methods are combined with mesh adaptivity in a complementary fashion, choosing to increase mesh resolution (h refinement) near non-smooth features, while resorting to higher-order discretizations (p-refinement) in smooth regions of the solution. The ultimate success of these approaches rests on the development of reliable h and p refinement criteria, which must include an assessment of solution smoothness.

The capturing of important flow physics has been shown to be essential for accurate drag prediction. For cruise conditions with minimal regions of flow separation, gradual improvements will likely be delivered through the development of more sophisticated turbulence and transition modeling. However, for high-lift and off-design cases involving potentially large regions of separated flow, there is a growing consensus that large eddy-simulation techniques may offer the best hope for improved predictive capabilities. Large-eddy simulations are currently several orders of magnitude more expensive than Reynolds-averaged Navier-Stokes calculations, making them impractical for aerodynamic applications. Nevertheless, accurate LES simulations of aerodynamic flows such as an airfoil near stall have recently been demonstrated [23]. The large amount of grid resolution required by these calculations is also an indication that higher-order methods may be used for effectively reducing the cost of LES calculations in future work [115]. Hybrid methods, such as the Detached Eddy Simulation (DES) technique, which employ RANS methods in boundary layer regions and LES methods in off-body regions, have also been proposed, in order to reduce the overall numerical costs of these problems. These methods have shown superior fidelity for simulating flows with large

amounts of separation [119, 25], although they are not currently mature enough to be used for accurate drag prediction tasks. However, these methods may benefit as well from the inclusion of higher-order discretization techniques, h-p refinement strategies, and improved physics modeling in the transition region between the boundary layer and outer regions.

These notes have provided an overview of the current capabilities of aerodynamic drag prediction using unstructured mesh solvers. The pacing items of grid resolution, discretization accuracy, solution efficiency, and flow physics have been identified, and strategies for dealing with these items in their current state, as well as strategies for improving these items through future research have been discussed. However, while improvements will be enabled through continued research in these areas, the ultimate delivery of more accurate, efficient, and reliable drag prediction tools for the aircraft designer will only occur with continued and sustained effort in the incorporation and validation of these techniques into the existing design framework as they become available. Continued long term investment in the validation and verification of these methods, both computationally and experimentally, is essential for the benefits of advances in research to be exploited by the aerodynamics community.

References

[1] M. Hemsch. Statistical analysis of CFD solutions from the Drag Prediction Workshop. AIAA Paper 2002-0842, January 2002.

[2] E. C. Carter and K. C. Pallister. Development of testing techniques in a large transonic wind tunnel to achieve a required drag accuracy and flow standards for modern civil transports. In *Aerodynamic Data Accuracy and Quality: Requirements and Capabilities in Wind-Tunnel Testing*. AGARD-CP-429, July 1988.

[3] J. B. Vos, A. Rizzi, D. Darracq, and E. H. Hirschel. Navier-Stokes solvers in European aircraft design. *Progress in Aerospace Sciences*, 38:601–697, 2002.

[4] J. C. Vassberg, P. G. Buning, and C. L. Rumsey. Drag prediction for the DLR-F4 wing/body using OVERFLOW and CFL3D on an overset mesh. AIAA Paper 2002-0840, January 2002.

[5] L. M. Gea, N. D. Hasley, G. A. Intemann, and P. G. Buning. Applications of the 3D Navier-Stokes code for analyzing propulsion airframe integration related issues on subsonic transports. In *Proc. of the 19th Congress of the International Council of the Aeronautical Sciences (ICAS'94), Anaheim, CA*, pages 2420–2435, September 1994. Paper ICAS-94-3.7.4.

[6] D. Om, M. Curtin, D. Bogue, D. Witkowski, and D. Ball. Reynolds number effects on a subsonic transport at transonic conditions. AIAA Paper 2001-0909, January 2001.

[7] S. Rolston and E. Elsholz. Initial achievements of the European high Reynolds number aerodynamic research project "HiReTT". AIAA Paper 2002-0421, January 2002.

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

[9] M. F. Smith. User friendly CFD - Application to TRANAIR for analysis of transport aircraft. AIAA Paper 98-5574, SAE Paper 985574, September 1998.

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

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

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

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

[14] 11th international meshing roundtable. In *Proc. of the 11th International Meshing Roundtable*, pages 29–43, September 2002. Prepared by Sandia National Laboratory.

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

[16] M. Potsdam. An unstrucutred mesh Euler and interactive boundary layer method for complex geometries. In *Proceedings of the 12th AIAA Aerodynamics Conference, Colorado Springs, CO*, June 1994. AIAA Paper 94-1844.

[17] J. Szmelter and A. Pagano. Viscous inviscid interaction techniques using unstructured meshes. In *Numerical Methods for Fluid Dynamics V*, pages 599–604, Oxford, UK, 1995. Oxford Science Publications. eds. K. W. Morton and M. J. Baines.

[18] A. G. McDonald and J. F. Nash. A turbulent skin-friction law for use at subsonic and transonic speeds. Aeronatical Research Council Report No. ARC-CP-948, January 1967.

[19] T. Cebeci and J. Cousteaux. *Modeling and Computation of Boundary Layer Flows.* Springer, Horizon Publishing Inc., Long Beach, CA, 2001.

[20] P. R. Spalart, W-H. Jou, M. Strelets, and S. R. Allmaras. Comments on the feasibility of LES for wings and on a hybrid RANS/LES approach. Paper presented at the First AFOSR International Conference on DNS and LES, Louisiana Tech University, Ruston, Louisiana, August 1997.

[21] K. Jansen. Unstructured grid large eddy simulation of flow over an airfoil. Annual Research Briefs, Center for Turbulence Research, Stanford University, 1994.

[22] C. P. Mellen, J. Frohlich, and W. Rodi. Lessons from the European LESFOIL project on LES of flow around an airfoil. AIAA Paper 2002-0111, January 2002.

[23] I. Mary and P. Sagaut. Large eddy simulation of flow around an airfoil. AIAA Paper 2001-2559, June 2001.

[24] G. S. Constantinescu and K. D. Squires. LES and DES investigations of turbulent flow over a sphere. AIAA Paper 2000-0540, January 2000.

[25] J. Forsythe, K. Squires, K. Wurtzler, and P. Spalart. Detached eddy simulation of fighter aircraft at high alpha. AIAA Paper 2002-0591, January 2002.

[26] D. J. Mavriplis, J. Pelaez, and O. Kandil. Large eddy and detached eddy simulations using an unstructured multigrid solver. In *DNS/LES - Progress and Challenges. Proc. of the Third AFOSR Int. Conf. on DNS/LES, eds. C.L. Liu, L. Sakell, T. Beutner*, pages 461–470, October 2001.

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

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

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

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

[31] N. T. Frink. Recent progress toward a three-dimensional unstructured flow solver. AIAA Paper 94-0061, January 1994.

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

[33] D. J. Mavriplis and D. W. Levy. Transonic drag prediction using an unstructured multigrid solver. AIAA-Paper 2002-838, January 2002.

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

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

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

[37] A. Haselbacher, J. McGuirk, and G. Page. Finite-volume discretization aspects for viscous flows on unstructured meshes. In *Proceedings of the 13th AIAA CFD Conference, Snowmass, CO*, pages 599–609, June 1997. AIAA Paper 97-1946-CP.

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

[39] W. A. Wood and W. L. Kleb. Diffusion characteristics of upwind schemes on unstructured triangulations. AIAA Paper 98-2443, June 1998.

[40] C. Johnson, A. Szepessy, and P. Hansbo. On the convergence of shock-capturing streamline-diffusion finite element methods for hyperbolic conservation laws. *Math. Comp.*, 54(189):107–130, 1990.

[41] T. J. R. Hughes, L. P. Franca, and G. M. Hulbert. A new finite element formulation for computational fluid dynamics: VIII The Galerkin/least-squares method for advective-diffusive systems. *Comput. Methods Appl. Mech. Engrg*, 73:173–189, 1989.

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

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

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

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

[46] A. Jameson, W. Schmidt, and E. Turkel. Numerical solution of the Euler equations by finite volume methods using Runge-Kutta time stepping schemes. AIAA Paper 81-1259, 1981.

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

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

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

[50] D. J. Mavriplis. Multigrid solution of the two-dimensional Euler equations on unstructured triangular meshes. *AIAA Journal*, 26(7):824–831, July 1988.

[51] D. J. Mavriplis, A. Jameson, and L. Martinelli. Multigrid solution of the Navier-Stokes equations on triangular meshes. AIAA Paper 89-0120, January 1989.

[52] O. Brodersen and A. Sturmer. Drag prediction of engine-airframe interference effects using unstructured Navier-Stokes calculations. AIAA Paper 2001-2414, June 2001.

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

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

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

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

[57] D. J. Mavriplis. Adaptive meshing techniques for viscous flow calculations on mixed-element unstructured meshes. AIAA paper 97-0857, January 1997.

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

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

[60] J. R. Shewchuk. What is a good linear element ?, Interpolation, conditioning, and quality measures. In *Proc. of the 11th Int. Meshing Roundtable*. Sandia National Laboratories, September 2002.

[61] D. J. Mavriplis and S. Pirzadeh. Large-scale parallel unstructured mesh computations for 3D high-lift analysis. *AIAA Journal of Aircraft*, 36(6):987–998, December 1999.

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

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

[64] D. J. Mavriplis. Multigrid strategies for viscous flow solvers on anisotrpic unstructured meshes. In *Proceedings of the 13th AIAA CFD Conference, Snowmass, CO*, pages 659–675, June 1997. AIAA Paper 97-1952-CP.

[65] W. O. Valarezo and D. J. Mavriplis. Navier-Stokes applications to high-lift airfoil analysis. *AIAA J. of Aircraft*, 23(3):457–688, 1995.

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

[67] C. L. Rumsey and S. X. Ying. Prediction of high-lift: Review of present CFD capability. *Progress in Aerospace Sciences*, 38:145–180, 2002.

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

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

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

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

[72] P. J. Roache. Verification of codes and calculations. *AIAA Journal*, 36(5):696–702, 1998.

[73] I. G. Currie. *Fundamentals of Fluid Mechanics*. McGraw-Hill, New York, NY, 1974.

[74] R. D. Richtmyer and K. W. Morton. *Difference Methods for Initial-Value Problems*. Interscience Tracts in Pure and Applied Mathematics. John Wiley and Sons, New York, NY, 1967.

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

[76] W. K. Anderson R. Rausch and D. Bonhaus. Implicit multigrid algorithms for incompressible turbulent flows on unstructured grids. In *Proceedings of the 12th AIAA CFD Conference, San Diego CA*, June 1995. AIAA Paper 95-1740-CP.

[77] V. Venkatakrishnan and D. J. Mavriplis. Implicit solvers for unstructured meshes. *Journal of Computational Physics*, 105(1):83–91, June 1993.

[78] D. Zingg and A. Pueyo. An efficient Newton-GMRES solver for aerodynamic computations. In *Proceedings of the 13th AIAA CFD Conference, Snowmass, CO*, pages 712–721, June 1997. AIAA Paper 97-1955-CP.

[79] D. J. Mavriplis. Multigrid techniques for unstructured meshes. In *VKI Lecture Series VKI-LS 1995-02*, March 1995.

[80] M. P. Leclercq. *Resolution des Equations d'Euler par des Methods Multigrilles Conditions aux Limites en Regime Hypersonique*. PhD thesis, Univerite de Saint-Etienne, Applied Math, April 1990.

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

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

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

[84] D. J. Mavriplis and V. Venkatakrishnan. A 3D agglomeration multigrid solver for the Reynolds-averaged Navier-Stokes equations on unstructured meshes. *International Journal for Numerical Methods in Fluids*, 23(6):527–544, 1996.

[85] J. W. Ruge and K. Stüben. Algebraic multigrid. In S. F. McCormick, editor, *Multigrid Methods*, SIAM Frontiers in Applied Mathematics, pages 73–131, Philadelphia, 1987. SIAM.

[86] O. Hassan, K. Morgan, and J. Peraire. An adaptive implicit/explicit finite element scheme for compressible high speed flows. AIAA Paper 89-0363, January 1989.

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

[88] D. J. Mavriplis. Directional agglomeration multigrid techniques for high-Reynolds number viscous flows. AIAA paper 98-0612, January 1998.

[89] D. J. Mavriplis. On convergence acceleration techniques for unstructured meshes. AIAA paper 98-2966, presented at the 29th AIAA Fluid Dynamics Conference, Albuquerque, NM, June 1998.

[90] E. Isaacson and H. B. Keller. *Analysis of Numerical Methods*. John Wiley and Sons, New York, NY, 1966.

[91] D. J. Mavriplis. Directional coarsening and smoothing multigrid strategies for anisotropic Navier-Stokes problems. In *8th Copper Mountain Conf. on Multigrid Methods*, 1997.

[92] D. E. Keyes, D. K. Kaushik, and B. F. Smith. Prospects for CFD on petaflops systems. In *CFD Review M. Hafez, et. al., eds., Wiley, New York*, 1997.

[93] D. J. Mavriplis, R. Das, J. Saltz, and R. E. Vermeland. Implementation of a parallel unstructured Euler solver on shared and distributed memory machines. *The J. of Supercomputing*, 8(4):329–344, 1995.

[94] W. Gropp, E. Lusk, and A. Skjellum. *Using MPI: Portable Parallel Programming with the Message Passing Interface*. MIT Press, Cambridge, MA, 1994.

[95] OpenMP: Simple, portable,scalable SMP programming. http://www.openmp.org, 1999.

[96] D. J. Mavriplis. Parallel performance investigations of an unstructured mesh Navier-Stokes solver. ICASE Report No. 2000-13, NASA CR 2000-210088, March 2000.

[97] G. Karypis and V. Kumar. A fast and high quality multilevel scheme for partitioning irregular graphs. Technical Report 95-035, University of Minnesota, 1995. A short version appears in Intl. Conf. on Parallel Processing 1995.

[98] B. Hendrickson and R. Leland. A multilevel algorithm for partitioning graphs. In *Proceedings Supercomputing '95, ACM*, December 1995.

[99] E. N. Tinoco. An assessment of CFD prediction of drag and other longitudinal characteristics. AIAA Paper 2001-1002, January 2001.

[100] T. Gerhold and J. Evans. Efficient computation of 3D flows for complex configurations with the DLR-Tau code using automatic adaptation. In *Notes on Numerical Fluid Mechanics, Ed. W. Nitsche et, al.*, volume 72, pages 178–185. Vieweg Braunschweig, 1998.

[101] N. Kroll, C. C. Rossow, D. Schwamborn, K. Becker, and G. Heller. MEGAFLOW - A numerical flow simulation tool for transport aircraft design. In *Proceedings of the 23rd ICAS Congress, Toronto, Canada*, September 2002. Paper ICAS-2002-1.10.5.

[102] D. J. Mavriplis. Algebraic turbulence modeling for unstructured and adaptive meshes. *AIAA J.*, 29(12):2086–2093, 1991.

[103] G. M. Gatlin and R. J. McGhee. Experimental investigation of semispan model testing techniques. *AIAA Journal of Aircraft*, 34(4):500–505, 1997.

[104] C. L. Rumsey and T. B. Gatski. Recent turbulence model advances applied to multi-element airfoil computations. *AIAA J. of Aircraft*, 38(5):904–910, 2001.

[105] C. L. Rumsey, E. M. Lee-Rausch, and R. D. Watson. Three-dimensional effects on multi-element high-lift computations. AIAA Paper 2002-0845, January 2002.

[106] I. Fejtek. Summary of code validation results for a multiple element airfoil test case. AIAA Paper 97-1932, June 1997.

[107] F. T. Lynch R. C. Potter and F. W. Spaid. Requirements for effective high lift CFD. In *Proceedings of the 20th ICAS Congress, Sorrento, Italy*, pages 1479–1492, September 1996. paper ICAS-96-2.7.1.

[108] A. Moitra. Automated CFD analysis of two-dimensional high-lift flows. *AIAA J. of Aircraft*, 39(4):662–667, 2002.

[109] J. D. Mueller and M. B. Giles. Solution adaptive mesh refinement using adjoint error analysis. AIAA Paper 2001-2550, June 2001.

[110] D. A. Venditti and D. L. Darmofal. A grid adaptive methodology for functional outputs of compressible flow simulations. AIAA Paper 2001-2659, June 2001.

[111] M. Park. Adjoint-based, three-dimensional error prediction and grid adaptation. AIAA Paper 2002-3286, June 2002.

[112] A. Jameson, J. J. Alonso, J. J. Reuther, L. Martinelli, and J. C. Vassberg. Aerodynamic shape optimization techniques based on control theory. AIAA Paper 98-2538, June 1998.

[113] W. K. Anderson and V. Venkatakrishnan. Aerodynamic design optimization on unstructured grids with a continuous adjoint formulation. AIAA paper 97-0643, January 1997.

[114] D. A. Venditti and D. L. Darmofal. Anisotropic grid adaptation for functional outputs: Application to two-dimensional viscous flows. submitted to J. of Comp. Physics, September 2002.

[115] C. H. Whiting and K. E. Jansen. A stabilized finite-element method for the incompressible Navier-Stokes equations using a hierarchical basis. *International Journal of Numerical Methods in Fluids*, 35(1):93–116, 2001.

[116] B. Cockburn and C. W. Shu. The Runge-Kutta discontinuous Galerkin method for conservation laws V: Multidimensional systems. *Journal of Computational Physics*, 141(2):199–2241, 1998.

[117] F. Bassi and S. Rebay. A high-order accurate discontinuous finite element method for the numerical solution of the compressible Navier-Stokes equations. *Journal of Computational Physics*, 131:267–279, 1997.

[118] J. H. Casper and M. H. Carpenter. Computational considerations for the simulation of shock-induced sound. *SIAM J. Sci. Comp.*, 19(3):813–828, 1998.

[119] M. Strelets. Detached eddy simulation of massively separated flows. AIAA Paper 2001-0879, January 2001.