# Recent Advances in Agglomerated Multigrid

Hiroaki Nishikawa*, Boris Diskin†
*National Institute of Aerospace, Hampton, VA 23666*

James L. Thomas‡, Dana P. Hammond§
*NASA Langley Research Center, Hampton VA 23681*

**We report recent advancements of the agglomerated multigrid methodology for complex flow simulations on fully unstructured grids. An agglomerated multigrid solver is applied to a wide range of test problems from simple two-dimensional geometries to realistic three-dimensional configurations. The solver is evaluated against a single-grid solver and, in some cases, against a structured-grid multigrid solver. Grid and solver issues are identified and overcome, leading to significant improvements over single-grid solvers.**

## I. Nomenclature

| | |
|---|---|
| $\mathbf{R}$ | Residual vector |
| $\mathbf{Q}$ | Solution vector |
| $\delta\mathbf{Q}$ | Solution update due to a relaxation |
| $V$ | Dual-control volume |
| $\Delta\tau$ | Pseudo time step |
| $\omega$ | Local under relaxation parameter |
| $\omega_{min}$ | Minimum of $\omega$ |
| $\epsilon_{mz}$ | $O(10^{-15})$ |
| $\epsilon$ | Displacement parameter for the Frechet derivative |
| $\alpha$ | Angle of attack |
| $p$ | Pressure |
| $Re$ | Free stream Reynolds number |

## II. Introduction

Multigrid techniques[1,2] are used to accelerate convergence of current Reynolds-Averaged Navier-Stokes (RANS) solvers for both steady and unsteady flow solutions, particularly for structured-grid applications. Lallemand et al.[3] and Smith[4] introduced Agglomerated Multigrid (AgMG) methods for unstructured-grid applications, and Mavriplis et al.[5–8] developed AgMG methods for large-scale unstructured-grid applications. The AgMG methods have been implemented in several practical unstructured grid codes and applied to configurations used for the Drag and High Lift Prediction Workshops.[9,10] The AgMG methods accelerated convergence in many instances but problems were encountered, such as grid-dependent convergence to smaller residual levels, discrepancies between single-grid and multigrid solutions, and convergence problems for cases at the edges of the flight envelope.[9,10] In fact, analysis and systematic computations with these techniques showed convergence degradation on highly-refined grids even for simple model problems. To overcome the difficulty, we critically studied AgMG techniques[11,12] for isotropic and highly-stretched grids, and developed quantitative analysis methods and computational techniques to achieve fast grid-independent convergence for a model

*Senior Research Scientist (hiro@nianet.org), National Institute of Aerospace, 100 Exploration Way, Hampton, VA 23666 USA, Senior Member AIAA

†Associate Fellow (bdiskin@nianet.org), National Institute of Aerospace (NIA), 100 Exploration Way, Hampton, VA 23666 USA.; Visiting Associate Professor, Mechanical and Aerospace Engineering Department, University of Virginia, Charlottesville, Associate Fellow AIAA

‡Senior Research Scientist (James.L.Thomas@nasa.gov), Computational AeroSciences Branch, Mail Stop 128, Fellow AIAA.

§Senior Research Scientist (dana.p.hammond@nasa.gov), Computational AeroSciences Branch, Mail Stop 128.

equation representing laminar diffusion in the incompressible limit. It was found that consistent coarse-grid discretizations are essential for grid-independent multigrid convergence on isotropic grids,[11] and that prismatic cells, line-agglomeration, and line-implicit subiterations are essential for grid-independent multigrid convergence on highly-stretched grids.[12] Building upon these fundamental studies, we extended and demonstrated these techniques for inviscid, viscous, and turbulent flow equations over complex geometries using a serial code[13] and a parallel code.[14]

The developed AgMG method has been shown to provide a significant speedup over a state-of-the-art single-grid (SG) solver for the complex flows considered in the previous papers. However, several difficulties were encountered when the AgMG method was applied to some other problems. For example, multigrid convergence degraded in the presence of nearly-degenerate stencils on surface grids, grids with a polar singularity, disjoint grids in grid partitioning, etc. In this paper, we probe these and other issues and develop an improved AgMG method.

The following three properties guide our development of an improved AgMG solver:

1. **Automated agglomeration.** The agglomeration procedure automatically constructs coarse grids for a primal grid.

2. **Discrete Solvability.** Multigrid cycles/relaxations converge to 'machine zero' residuals for all equations (including the turbulence model equations) on all grids used in the multigrid solver.

3. **Efficiency.** The multigrid solver provides a lower operation count to convergence and comparable scalability of parallel computations in comparison with state-of-the-art SG solvers. On structured primal grids, the efficiency of an AgMG solver is comparable with the efficiency of structured multigrid (StMG) solvers.

Our goal is to develop an AgMG method possessing such properties that can be employed with high confidence for practical simulations to accelerate convergence of the current state-of-the-art SG solvers. This paper focuses on attaining these properties and evaluates the performance of an AgMG solver for a set of test problems: inviscid/laminar/RANS flows over a hemisphere cylinder in two dimensions (2D) and in three dimensions (3D) and RANS flows around a flat plate, a bump in a channel, RAE and NACA airfoils, and a wing-body-tail configuration.

The multigrid methods considered in this paper are implemented in a general purpose unstructured computational fluid dynamics code, FUN3D,[15,16] developed and supported at NASA Langley. The methods are based on the full-approximation scheme (FAS)[1,2] where a full solution (not just correction) is approximated on the coarse grid. A correction, computed as the difference between the restricted fine-grid solution and an (approximate) coarse-grid solution, is prolonged to the finer grid to update the fine-grid solution. The two-grid FAS is applied recursively through increasingly coarser grids to define a V-cycle. A V-cycle, denoted as $V(\nu_1, \nu_2)$, uses $\nu_1$ relaxations performed on each grid before proceeding to the coarser grid and $\nu_2$ relaxations after coarse-grid correction. On the coarsest grid within a V-cycle, 10 relaxations are performed. For turbulent flows, the mean-flow and turbulence equations are solved on each grid. The full-multigrid (FMG) method[1,2] is employed to start the multigrid computations on each grid level.

The relaxation scheme used in the AgMG and StMG methods is new. It is based on a Jacobian-free Newton-Krylov (JFNK) approach (e.g., see Ref. [17] for a review). Typically, JFNK methods employ a preconditioner, and efficiency of such methods strongly depends on the choice of the preconditioner. The preconditioner systems used with JFNK methods usually employ approximate Jacobian matrices, e.g., an incomplete lower-upper (ILU) decomposition with various fill approximations. For example, Osusky and Zingg[18] use central differencing with second- and fourth-difference artificial dissipation for the inviscid terms of the residual operator. For the inviscid terms in the Jacobian operator, central differences with scaled second-difference artificial dissipation are used. Lucas et al.[17] use divided-difference Jacobians and apply some techniques to reduce bandwidth, including lumping and approximations to the viscous terms. We use a preconditioner based on a defect-correction Jacobian approximation[19,20] with an adaptive time step control. The preconditioner uses the linearization of a first-order approximation for inviscid fluxes; the linearization of viscous fluxes is exact on primal grids and approximated on the agglomerated grids. The preconditioner system is not fully solved but relaxed in multicolor point-implicit and/or line-implicit manner. This preconditioner has scalable parallel performance and used previously as a smoother for multigrid.[13,14] The properties of this defect-correction method as a solver and a smoother for multigrid have been studied extensively[19,20]. As a further development, we envision that linear multigrid would be beneficial within the preconditioner stage of the current approach. MacLean and White[21] use a Krylov-based approach to solve an update equation based on defect correction with an ILU preconditioner to improve reliability over iterative point- and line-relaxation schemes.

The relaxation methodology draws heavily upon the approach taken by Allmaras et al.[22] that uses a direct linear solver and recovers quadratic convergence of a Newton's method. Our algorithm is formulated such that quadratic convergence can be recovered if the pseudo-time step is large enough and the linear system is solved sufficiently. However, because we do not solve the linear system to completion, superlinear convergence is achieved at best. Although not the focus of this paper, as a SG solver, the new scheme provides substantial improvements over the baseline solver in FUN3D. This scheme is well compatible with the hierarchical multigrid methods developed previously.[20,23]

The paper is organized as follows. The AgMG and StMG methods are described, including the finite-volume discretizations, the agglomeration scheme, the restriction, the prolongation, and the relaxation scheme involving generalized conjugate residual (GCR) method and an automated pseudo-time-step schedule. The numerical results are presented for the test cases. The final section contains conclusions. Appendix VI illustrates dependence of StMG convergence rates on the pseudo-time step.

# III. Multigrid Solvers

## III.A. Primal Grids and Discretizations

A grid is defined as primal if it has elements: quadrilaterals and triangles in 2D; hexahedra, prisms, pyramids, and tetrahedra in 3D. The discretization method used here on structured and unstructured primal grids is a finite-volume discretization centered at nodes. It is based on the integral form of governing equations of interest. The governing equations are the Euler equations, the Navier-Stokes equations, and the RANS equations with the Spalart-Allmaras one-equation turbulence model.[24] The primal-grid discretization of the viscous terms of the Navier-Stokes and RANS equations is obtained by the Green-Gauss (GG) scheme.[15,25] For mixed-element cells, an edge-based augmentation is used to increase the $h$-ellipticity of the viscous operator.[15,25] This augmentation is done by the face-tangent construction[12] with the efficient implementation that is independent of the face-tangent vectors (see Appendices of Ref. [26]); thus the resulting scheme is called here the face-tangent Green-Gauss scheme. The inviscid terms are discretized by a standard edge-based method with unweighted least-squares (LSQ) gradient reconstruction and Roe's approximate Riemann solver.[27,28]

The turbulence model is the Spalart-Allmaras model as described at the Turbulence Modeling Resource website,[29] including the provisions to accommodate negative turbulence.[30] The convective term of the turbulence equation is discretized with first-order accuracy while the diffusion terms are usually discretized with second-order accuracy by the face-tangent Green-Gauss scheme. The node-centered mean-flow gradients used in turbulent source evaluation are computed by the edge-based Green-Gauss scheme on dual elements.

For high-aspect-ratio grids typical for high Reynolds number flows, grid nodes are typically grouped to form so-called implicit lines extended from a body surface. This information is used by implicit-relaxation and agglomeration procedures described below.

## III.B. Structured Grid Multigrid Solver

The StMG solver exploits the grid structure and recursively derives a nested coarser grid from a finer grid by full coarsening, i.e., eliminating every other grid line in each direction. The primal-grid discretizations are used on coarse grids. The prolongation operator is a linear interpolation (bi-linear and tri-linear in 2D and 3D, respectively). The residual restriction operator is fully weighted, globally conservative and constructed as a (scaled) transposition to the prolongation operator. The finer grid solution is restricted to the coarser grid by injection. A special treatment is required at singular points of the mapping associated with the structured grids. For example, hexahedral cells are split into two prismatic cells at a parabolic singularity of a mapping.

## III.C. Agglomerated Grids and Discretizations

*Hierarchical Agglomeration Scheme*

As described in the previous papers,[11–13] the grids are agglomerated within a topology-preserving framework, in which hierarchies are assigned based on connections to the computational boundaries: corners, ridges, valleys, and interiors. The agglomerations proceed hierarchically from seeds within the topologies — first corners, then ridges, then valleys, and finally interiors. See figure 1 for illustration. Hierarchies on each agglomerated grid are inherited from the finer grid. A typical boundary agglomeration generated by the agglomeration scheme is shown in figure 1(a).

**Table 1. Summary of discretizations used on primal and agglomerated grids.**

| Grids | Equations | Inviscid | Diffusion | Source |
|---|---|---|---|---|
| Primal | Mean-flow | Second-order | Face-tangent GG | |
| | Turbulence | First-order | Face-tangent GG | Edge-based GG |
| Agglomerated | Mean-flow | First-order | Avg-WLSQ | |
| | Turbulence | First-order | Avg-LSQ | LSQ |

*Full-Coarsening Line-Agglomeration for Viscous Grids*

For viscous grids, we perform the agglomeration in the following sequence:

1. Agglomerate viscous boundaries (bottom of implicit lines, in general).
2. Agglomerate prismatic layers through the implicit lines (line agglomeration).
3. Agglomerate the rest of the boundaries.
4. Agglomerate the interior.

The second step is a line-agglomeration step where volumes are agglomerated along implicit lines starting from the volume directly above the boundary volume, thus preserving the boundary agglomerates. Figure 1(b) illustrates typical line agglomeration near a curved solid body.

*Coarse-Grid Discretization*

On coarse grids, the governing equations are directly discretized. For inviscid coarse-grid discretization, a first-order edge-based scheme is employed. The gradients at the agglomerated cells are computed using either unweighted least-squares scheme (LSQ) or inverse-distance weighted least-squares scheme (WLSQ). For the mean-flow diffusion term, the Average-Weighted-Least-Squares scheme (Avg-WLSQ) with the face-tangent argumentation is employed. For the turbulence diffusion term, the Average-Least-Squares scheme (Avg-LSQ) with the face-tangent argumentation is employed.[12,26,31] Table 1 shows a summary of discretizations.

The specific choice of gradient evaluations used in this study for computing turbulence sources is not optimal. Our previous experience indicates that accuracy of gradients used in the turbulence source is critical for accuracy of the turbulent-flow solutions. The LSQ gradients are not accurate on curved high-aspect-ratio grids. Improvements in solution accuracy and in iterative convergence are expected if accurate gradients (e.g., the WLSQ scheme on agglomerated grids) are used.

*Parallel Implementation*

For parallel implementation, the hierarchical agglomeration algorithm is applied independently in each partition. Consequently, no agglomeration is performed across partition boundaries. No special modification is necessary for the line agglomeration as our partitioning guarantees that all nodes in each grid-line belong to the same partition (i.e., lines are partitioned in the form of condensed-nodes each of which contains all nodes in a line).

### III.D. Prolongation and Restriction Operators in Agglomerated Multigrid Solver

The residuals are restricted to the coarser grid by summing the finer-grid residuals over the control volumes agglomerated into a coarser control volume. The solution is restricted to the coarser grid by taking the volume-weighted average. The correction to the finer grid is prolonged by the linear interpolation from a tetrahedron defined on the coarse grid.[13,14] For volumes having negative coefficients in the linear interpolation, as typically happens near curved viscous boundaries, piecewise constant prolongation is performed.

### III.E. Relaxation Scheme

The relaxation scheme used in the AgMG and StMG methods uses a GCR method to approximately solve a linear update equation. The update equation for each control volume $V$ is composed of a linearization of the

nonlinear residual with a pseudo-time addition,

$$\left(\frac{\partial \mathbf{R}}{\partial \mathbf{Q}} + \frac{V}{\Delta\tau}\right)\delta\mathbf{Q} = -\mathbf{R}(\mathbf{Q}), \tag{1}$$

$$\mathbf{Q}^{\text{new}} = \mathbf{Q} + \omega\delta\mathbf{Q}, \tag{2}$$

Here $\mathbf{R}$ is the vector of nonlinear residuals, $\frac{\partial \mathbf{R}}{\partial \mathbf{Q}}$ is the linearization matrix corresponding to $\mathbf{R}$, $\delta\mathbf{Q}$ is the solution update, $\omega$ is a local under-relaxation parameter, and $\Delta\tau$ is a pseudo-time step, which is set separately for the mean-flow and the turbulence equations. The pseudo time step $\Delta\tau$ is set through a CFL specification by analogy with time marching schemes for inviscid equations. The pseudo-time step is set according to a specified initial variation and then adapted within specified maximum and minimum bounds. At each node, the under-relaxation parameter $\omega$ enforces constraints on allowable changes of solution components. A maximum change on the order of $\pm 20\%$ is allowed for the density, total energy, and pressure. The pressure update is estimated by a small-perturbation linearization, $\frac{\partial p}{\partial \mathbf{Q}}\delta\mathbf{Q}$, but then checked through a full nonlinear update of the conserved variables. This proved beneficial in eliminating seemingly random updates of pressure to negative values. A similar approach is taken for updating turbulence; a maximum of 100% is allowed for increases and 20% for decreases. Larger increases are allowed because turbulence can grow significantly in computations, and at the same time, we would like to avoid negative turbulence. If the update to turbulence is below a specified threshold (0.001), the full update is taken.

The residual of the update equation (1) is approximated with a matrix-free approach, using a first-order Frechet derivative to account for changes in the solution $\mathbf{Q}$,

$$\frac{\partial \mathbf{R}}{\partial \mathbf{Q}}\delta\mathbf{Q} = \frac{\mathbf{R}(\mathbf{Q} + \epsilon\delta\mathbf{Q}) - \mathbf{R}(\mathbf{Q})}{\epsilon}, \tag{3}$$

where $\epsilon$ is a small parameter subject to a realizability constraint. The value of $\epsilon$ in the Frechet derivative computation is set such that

$$\epsilon = \frac{\sqrt{\epsilon_{mz}}}{\max(\omega_{min}, 0.01)||\delta\mathbf{Q}||}, \tag{4}$$

where $\epsilon_{mz}$ is a tolerance, here $O(10^{-15})$, and $\omega_{min}$ is the minimum of $\omega$ over the field. Typically, $\omega_{min}$ is less than unity only in early stages of convergence and $\omega_{min} < 0.01$ occurs rarely. The Frechet derivative can also be constructed using complex arithmetic. This procedure is known to be very accurate and insensitive to the selection of the size of the perturbation. Several cases have been run with both approaches and the differences are quite small, giving confidence the roundoff associated with real-valued Frechet is not an important concern.

For turbulent flows, the relaxation of the update equation (1) can be performed in either a loosely-coupled or tightly-coupled manner.[14] In loosely-coupled relaxations, the updates of the mean-flow solution are completed first, and then the updates of the turbulence solution are performed. The mean-flow Jacobian is composed of 5x5 block entries and the turbulence Jacobian is composed of scalar entries. In tightly-coupled relaxations, solution updates are performed simultaneously for the mean-flow and turbulence solutions. The Jacobian is composed of 6x6 block entries. The Jacobians for the turbulence model are formed in stages. The Jacobians for convection and nonlinear diffusion terms are formed first. If any negative terms appear on the diagonal, molecular viscosity is added to the residual to ensure a non-negative term on the diagonal. This correction typically occurs only on coarse grids or during the initial convergence. The Jacobians for the source term and the pseudo-time term are added at the next stage. If the resulting diagonal terms are negative, an additional pseudo-time term is added to the Jacobians.

The preconditioner for GCR is an implicit formulation based on a linearization of the residual with first-order inviscid terms and a pseudo-time term. The viscous terms are linearized exactly on the primal grids; for agglomerated grids, the linearization of viscous terms includes only contributions from the edge derivatives.[14] Table 2 summarizes the Jacobians used in the preconditioner for inviscid and viscous terms. The pseudo-time step is nominally the same as the one specified in Equation (1), but lags behind this value when Jacobians are frozen.

An approximate solution to the preconditioner linear system is computed by a certain number of subiterations. Each subiteration consists of a set of two multicolor Gauss-Seidel iterations: a point-implicit iteration over the entire grid and a subsequent line-implicit iteration over the region where lines are available (generally in stretched grid regions). The strategy used in SG and AgMG is to use 10 subiterations for both mean-flow and turbulence equations. In general, 10 subiterations may not be sufficiently general (e.g., Ref. [21]). An alternative strategy pursued in the StMG solver is to specify targets for reduction in the $L_2$-norm of the residual

**Table 2. Accuracy of preconditioner Jacobians.**

| Grids | Inviscid | Viscous |
|---|---|---|
| Primal, mean-flow | Approximate (first order) | Exact |
| Primal, turbulence | Exact | Exact |
| Agglomerated, mean-flow and turbulence | Exact | Approximate (face-tangent, edge-term only) |

of the linear system. The targeted reduction is 0.5. Subiterations are discontinued if the target is met after a specified minimal number of subiterations. If a minimal reduction of 0.98 is not attained within a specified maximum number of subiterations, the pseudo-time step is scheduled for reduction. For the loosely-coupled approach, typical minimum subiterations are 10 and 5 and typical maximum subiterations are 100 and 50 for the mean-flow and turbulence equations, respectively. For the tightly-coupled approach, minimum and maximum numbers are 10 and 100, respectively. The target of 0.5 for the preconditioner residual reduction is usually reached within the minimum number of subiterations. However, in the later stages of convergence at higher CFL numbers, it is not uncommon to require 2-3 times more subiterations. It is quite rare for the maximum number of subiterations to be used. As mentioned above, a linear multigrid scheme (as in Ref. [20]) would be beneficial in reducing this variability and efficiently providing convergence well beyond this threshold.

The GCR tolerance for reductions of the $L_2$-norm of the linear update residuals (1) is the same as the preconditioner tolerance; for loosely coupled relaxations, the targets are 0.9 for the mean-flow and 0.5 for the turbulence equations, respectively. A restarting procedure[17] is implemented, but generally, no restarts and a maximum of 4 projections are allowed. The GCR procedure is terminated when the GCR targets are reached. If the GCR targets have not been reached with 4 projections and are above a threshold (0.9), the pseudo-time step is scheduled for reduction; if the GCR residual reduction is below the threshold, the pseudo-time step is retained.

Generally, the role of GCR in the present approach is more to control instabilities of the preconditioner solution procedure rather than accelerate convergence. This stabilizing effect was noted to be particularly important on the coarsest grids in a multigrid cycle. Usually only one or two GCR projections are used. When the maximum number of projections are used for many cycles, generally the solver is not efficient.

The Jacobians used in the preconditioner are updated infrequently. On each new (currently finest) grid within the FMG algorithm, the finest-grid Jacobians are updated after each finest-grid relaxation within the first V-cycle. For the first 10 V-cycles, the Jacobians on all grids are updated at the beginning of each V-cycle. Thereafter, the Jacobian updates are performed at the beginning of a V-cycle; the update schedule is a function of the finest-grid residual reduction. The schedule is tuned to update Jacobians every 10 cycles whenever the finest-grid residual is reduced by more than 4 orders of magnitude from the initial level.

A user-specified variation of CFL versus FMG cycles is prescribed, after which the exponential progression with under-relaxation CFL update strategy[22,32] is followed. The V-cycle is truncated if the CFL parameter is scheduled for reduction. In other words, if a CFL reduction is called on any level within a V-cycle, further actions at and below this level are terminated, i.e., the V-cycle continues on finer levels without coarse-grid correction from the current level. All CFL updates happen at the beginning (top) of the next V-cycle. In general, multigrid exhibits better asymptotic convergence with higher maximum CFL specifications (see Appendix VI for examples). In this paper, a relatively small maximum CFL= 200 is chosen as the value that provided convergence for all applications tested here.

## IV. Discovery and Resolution of Grid and Solver Issues

In this section, we consider various grid and solver issues that have been found detrimental for performance of multigrid solvers.

### IV.A. Polar Singularity

Convergence degradation was encountered in laminar computations for a 3D hemisphere cylinder. The cause of the problem was found to be the umbrella-like stencils on the polar grid line having hundreds of neighbors in a spindle arrangement and only two neighbors along the polar grid line (see figure 2(a)). In such a stencil,

the unweighted LSQ gradient construction over-emphasizes those hundreds of spindle neighbors, resulting in a very inaccurate gradient along the polar grid line. Multigrid converges satisfactorily for relatively coarse grids with fewer neighbors, but fails on fine grids. We found that the inverse-distance weighted LSQ gradient construction applied locally at the polar grid line recovers convergence. However, we do not recommend this as a remedy for three reasons: 1) with the fixed inverse-distance weights, the same problem may occur on a sufficiently fine grid where the effective-weight due to the number of neighbors exceeds the inverse-distance weight; 2) on high-aspect-ratio grids, the accuracy of the weighted LSQ method is compromised due to a minor perturbation (wiggling) of the polar grid line; 3) the preconditioner subiterations may become unstable when unlimited weighted LSQ gradients are applied, even locally.

A general solution for the polar-singularity issue would be the use of the approximate mapping technique.[33] Yet another remedy is to generate grids without such stencils. An example is shown in figure 2(b), where the umbrella-like stencils always have six neighbors in the spindle arrangement on grids of any size. In this paper, we show results for a 3D hemisphere cylinder geometry obtained on grids with no polar singularity.

### IV.B. Disjoint Grids

A partitioner does not guarantee that each partition consists of a contiguous block of nodes, where a block is defined as a collection of connected nodes. In fact, we often encounter disjoint grids, i.e., partitions that have more than one block. This occurs as generalized partitioners (e.g., ParMetis, PT-scotch) focus more on load balancing and communication constraints than on keeping partitions contiguous. A partitioned triangular grid for the 2D hemisphere cylinder (2135 nodes) is shown in figure 3. Since the agglomeration is performed in the advancing-front manner, it will not cover all grid blocks unless the initial seed is chosen from all disjoint grid blocks. In order to guarantee that all grid blocks are agglomerated, we first count the number of disjoint grids in each partition (by a neighbor-to-neighbor search), and then make sure we pick an initial seed from each grid block. A problem is encountered when a partition has grid blocks of unbalanced sizes, especially in the boundary nodes. An extreme case is that a partition has a grid with only one boundary node. The boundary node has no boundary-neighbors to be agglomerated within the partition, and can be easily contained by a single volume arising from the agglomeration in the neighboring partition. It causes various problems: vanishing face normal vector, invalid LSQ stencil, etc. Far and near field views of disjoint partitions (non-extreme cases) can be seen in figures 3(a) and 3(b), respectively. To avoid unbalanced and disjoint grids, a post-partitioning fix is implemented where small grid blocks are merged with a neighboring partition. In 3D, we also set a lower bound for the size of partitions, e.g., each partition is required to have at least 60,000 nodes, and typically opt for a three-level AgMG algorithm.

### IV.C. Flat Least-Squares Stencil at a Viscous Surface

In early attempts to solve viscous problems with an AgMG solver, we observed negative pressures on coarse agglomerated grids. The problem was encountered only for the full Navier-Stokes equations, not for the inviscid/diffusion equations. The problem was traced to boundary volumes on a viscous surface having only two neighbors on the surface. See figure 4(a) for illustration. On advancing-layer prismatic grids, such a boundary volume may lead to a four-point stencil for the LSQ gradient reconstruction with an obtuse-angle triangle at the surface. Such a stencil results in a very inaccurate LSQ gradient in the direction tangential to the surface and perpendicular to the edge across the surface-neighbors. This inaccurate gradient component does not greatly affect the inviscid scheme because only the edge gradient component is used in the reconstruction step. Also, it does not affect the diffusion scheme because only the face-normal component (which is nearly the same as the edge component) is used in the diffusive flux. However, for the full Navier-Stokes equations, the face-tangent gradient component is used for the shear stresses, and that is where the problem occurred. To avoid this problem, we perform a search for surface volumes having only two neighbors and merge them with one of their neighbors before the line agglomeration. See figure 4(b) for illustration. The process successfully removed the problematic stencils.

### IV.D. Highly Curved Stencil on Agglomerated Grids

In the hierarchical agglomeration, the boundary control volumes are agglomerated with boundary neighbors only and never agglomerated with the interior neighbors. Such an agglomeration scheme can create very thin control volumes especially near the boundary. For curved boundaries, it results in highly-curved stencils as shown in figure 5. Such stencils cause difficulties in gradient computations and solution reconstructions for

inviscid fluxes. We expect that the approximate mapping technique[33] will solve this problem. In this paper, we use the first-order inviscid scheme on all agglomerated grids. The scheme is very robust, but it can slow down the AgMG convergence.

### IV.E. Implicit Lines in Wakes

Line-agglomeration can be used to agglomerate the entire domain in structured grids. In such cases, nodes of different hierarchies can be contained in a line. This problem was not encountered in previous tests. An example is the line that originates from a part of the outer boundary, passes through a corner (e.g., the trailing edge of an airfoil), and then extends to another part of the outer boundary. Figure 6 illustrates the line passing through the trailing edge where the trailing edge node is indicated by the blue circle. To deal with such a situation, we first assign the highest nodal-hierarchy to each line, and perform the bottom-of-line surface agglomeration within the same hierarchy. Then, the line-agglomeration progresses from the surface, but the agglomeration is performed only for nodes in the same hierarchy. In this way, the line passing through the trailing edge will not be agglomerated with neighbor lines and the trailing-edge node will not be agglomerated with any neighbor. The obtained agglomerated grids may lose some symmetry property characteristic of the primal grid and also introduce large skewing and high-aspect ratio not present on the primal grid. However, as will be shown later in the RAE airfoil test case, numerical tests indicate that the performance of the AgMG is not strongly affected by the quality of the agglomerated grids.

## V. Numerical Results

Unless otherwise stated, numerical results are presented for the three-level AgMG V(3,3) cycle with the loosely-coupled GCR relaxation. Mean-flow and turbulence equations are relaxed on primal and agglomerated grids. The CFL number is ramped from 10 to 200 over the first 50 relaxations on each grid level. The FMG scheme is employed with residual convergence to a specified low tolerance on every level. This low tolerance is case dependent, but chosen to be comparable with 'machine zero' residuals. The above parameters are selected to provide convergence for all cases tested in this paper. The AgMG convergence is compared with the SG convergence to demonstrate the benefit of the multigrid coarse-grid correction. For consistency in comparisons, the SG iteration is performed by the 2-level AgMG V(3,3) cycle with no coarse-grid corrections. In this way, we can ensure that the set of parameters, e.g., the CFL ramping strategy, Jacobian updates, etc., are the same. This increases the elapsed time by less than 10 percent over that of computation with only 1 grid. Tables 3 and 5 summarize the total number of cycles required to converge on the primal grids for 2D and 3D cases, respectively.

### V.A. 2D Hemisphere Cylinder

We consider 2D inviscid, laminar, and RANS flows over a hemisphere cylinder with the radius 0.5 and the length 20 in the grid units. The geometry is taken as a 2D section of the experimental model studied in Ref. [34]. In all cases, the Mach number is 0.4, and the angle of attack is 1.0 degree. The outer boundary is located approximately at the average distance of 60 radii from the body. The primal computational grids are triangular for the inviscid case and mixed element for the laminar cases with approximately 0.24 million nodes. For the RANS case, three grids are prepared with approximately 0.24, 0.48, and 0.96 million nodes. The number of processors used is 4, 8, 16 for the 0.24, 0.48, and 0.96 million-node grids, respectively, so that each partition has approximately 60,000 nodes. The tolerance is set to be $10^{-14}$ in the residual $L_2$ norm. In all cases, both the AgMG and SG solvers converged to the specified tolerance with no issues. Convergence histories are given in figure 7.

For the inviscid flow, a triangular grid was used. Although the grid has a regular structure (i.e., generated from a structured quadrilateral grid by diagonal insertion), it is treated as an unstructured grid and agglomerated without the line-agglomeration technique. We see from table 3 that AgMG accelerates the convergence over SG by a factor of 5. See figure 7(a) for convergence histories.

For the laminar flow, we set $Re = 4.0 \times 10^2/(\text{unit grid length})$. The grid is a mixed grid with quadrilaterals in the boundary layer region and triangles everywhere else. Line agglomeration and line-implicit subiterations are performed in the quadrilateral part. For this case, AgMG achieves a factor of 8 speed-up over SG as can be seen in table 3. See figure 7(b) for convergence histories.

Finally, we consider a turbulent flow with $Re = 3.5 \times 10^5/(\text{unit grid length})$. The grid is strongly stretched near the body to keep $y^+ < 1$. The grid is a mixed grid with quadrilaterals in the boundary layer region and

**Table 3. Two dimensional test cases (RANS unless otherwise stated): the number of cycles required for convergence on the primal grid is shown for SG, 3-level AgMG, and 3-level StMG. Additional AgMG tests for the RAE airfoil: line 1 agglomeration results are left from the colon; line 2 agglomeration results are right from the colon; results from 5-level AgMG cycle are in parenthesis.**

| Case (stopping tolerance) | Grid (nodes/processors) | SG | AgMG | StMG |
|---|---|---|---|---|
| Hemisphere-Cylinder - Inviscid ($10^{-14}$) | 241,542/4 | 379 | 75 | |
| Hemisphere-Cylinder - Laminar ($10^{-14}$) | 239,625/4 | 512 | 66 | |
| Hemisphere-Cylinder - RANS ($10^{-14}$) | Grid 1 (960,939/16) | 1,116 | 71 | |
| | Grid 2 (480,225/8) | 764 | 46 | |
| | Grid 3 (240,381/4) | 514 | 45 | |
| Flat Plate ($10^{-14}$) | 209,825/4 | 2,297 | 247 | |
| Bump ($10^{-13}$) | 3,649/1 | 335 | 135 | |
| RAE Airfoil ($10^{-13}$) | Grid 1 (98,944/1) | 238 | 96(60) : 65(62) | 57 |
| | Grid 2 (24,896/1) | 120 | 40(40) : 40(40) | 33 |
| | Grid 3 (6,304/1) | 77 | 45(43) : 45(44) | 33 |
| | Grid 4 (1,616/1) | 58 | 33(33) : 44(43) | 67 |
| NACA0012 $\alpha = 0^\circ$ ($10^{-10}$) | Grid 1 (919,424/48) | >10,000 | 5,118 | 3072 |
| | Grid 2 (230,336/8) | >10,000 | 3,926 | 2873 |
| | Grid 3 (57,824/2) | >10,000 | 2,982 | 2476 |
| NACA0012 $\alpha = 10^\circ$ ($10^{-10}$) | Grid 1 (919,424/48) | 1,589 | 191 | 112 |
| | Grid 2 (230,336/8) | 861 | 131 | 71 |
| | Grid 3 (57,824/2) | 388 | 86 | 73 |
| NACA0012 $\alpha = 15^\circ$ ($10^{-10}$) | Grid 1 (919,424/48) | 2,136 | 276 | 174 |
| | Grid 2 (230,336/8) | 1,290 | 241 | 109 |
| | Grid 3 (57,824/2) | 583 | 116 | 121 |

triangles everywhere else. Again, the line agglomeration and line-implicit subiterations are performed in the quadrilateral region. See figures 7(c) and 7(d) for convergence histories.

Qualitatively, the convergence results are similar for flows of all types. As can be seen in figure 7, on the coarsest grids, the AgMG and SG solvers are identical. On all other grids, the AgMG solver converges significantly faster than the SG solver.

### V.B. 2D Flat Plate

We consider a turbulent flow with Mach number 0.5 and $Re = 5.0 \times 10^6$/(unit grid length). The grid is a 545×385 quadrilateral grid (209,825 nodes) taken from the Turbulence Modeling Resource website.[29] Implicit-lines are formed from the bottom to the top, covering the entire domain. The line-agglomeration is performed over the entire domain. The case was run with four processors, leading to approximately 50,000 nodes per partition. Both the AgMG and SG solvers converged to the specified tolerance, $10^{-14}$, with no major issues. See figure 8 for convergence histories. From table 3, we see that the AgMG solver converges nearly 10 times faster than the SG solver.

### V.C. 2D Bump

We consider a turbulent flow with Mach number 0.75 and $Re = 3.0 \times 10^6$/(unit grid length). The grid is a 89×41 quadrilateral grid (3,649 nodes) taken from the Turbulence Modeling Resource website.[29] This case was run with one processor. The tolerance is set to $10^{-13}$ in the residual $L_2$ norm. The coarse grids are generated by the line-agglomeration with implicit-lines defined similarly to the flat plate case. Both the AgMG and SG solvers converged to the specified tolerance with no major issues. AgMG achieves a speed-up of factor 3 over SG as seen in table 3. See figure 9 for convergence histories.

Table 4. Improvement in cycles to converge for finest grids of 2D RANS test cases.

| Case | AgMG Improvement | StMG Improvement |
|---|---|---|
| Hemisphere-Cylinder | 15.7 | |
| Flat Plate | 9.3 | |
| Bump | 2.5 | |
| RAE Airfoil | 2.5 | 4.2 |
| NACA0012 $\alpha = 0°$ | >2.0 | >3.3 |
| NACA0012 $\alpha = 10°$ | 8.3 | 14.2 |
| NACA0012 $\alpha = 15°$ | 7.7 | 12.3 |

### V.D. 2D RAE Airfoil

We consider a turbulent flow over the RAE airfoil with Mach number 0.676, $Re = 5.7 \times 10^6$/(unit grid length), and $\alpha = 1.93$ degrees. There are four primal grids: Grid 1 with 98,944 nodes, Grid 2 with 24,896 nodes, Grid 3 with 6,304 nodes, and Grid 4 with 1,616 nodes. The primal grids are all structured quadrilateral grids. We apply the 3-level AgMG to each grid independently. The tolerance is set to be $10^{-13}$ in the residual norm. This test case involves a line with mixed hierarchies: the line passing through the trailing edge has a corner (the trailing edge node) in the middle. This line is treated exactly as described in Section IV.E. Two line agglomeration strategies are defined. The strategies are identical for the lines that start at the airfoil surface: the agglomeration starts at the surface and proceeds toward the outer boundary according to the algorithm described in Section III.C. The strategies are different in the wake. The line 1 strategy starts from the bottom outer boundary and proceeds to the top outer boundary. The line 2 strategy splits into two procedures: one procedure starts from the wake center-line and proceeds to the top boundary and another procedure starts from the same wake center-line and proceeds to the bottom boundary. Each agglomeration strategy covers the entire domain, and thus the line-agglomeration alone generates an entire coarse grid. Examples of agglomerated grids generated by these two line-agglomeration strategies are shown in figures 10 and 11. As can be seen in the figures, the agglomerated grid generated by the line 1 strategy has a gap in the grid line as indicated by the arrows while the agglomerated grid generated by the line 2 strategy has no such gaps.

One can see from table 3 that the AgMG solver always converges faster than the SG solver. The results for different line-agglomeration strategies are separated by a colon; the line 1 results are on the left side. The numbers in the parenthesis are obtained by the 5-level multigrid. On the finest primal grid, the three-level line-1 AgMG slows down compared with the three-level line-2 AgMG. Convergence histories are shown in figures 12 and 13. The line-1 AgMG experiences a difficulty in the early convergence stage at the primal grid level, but eventually converges rapidly. Although the line-2 strategy seems to give an advantage on Grid 1, the line-1 AgMG converges faster on Grid 4. On Grid 1 and for the five-level AgMG solvers on all grids, convergence of line-1 and line-2 solvers is quite comparable. See figures 14 and 15 for convergence histories of 5-level AgMG on Grid 1. In table 3, the AgMG convergence is also compared with convergence of an StMG solver with the same multigrid parameters: three-level V(3,3) cycle with 10 relaxations on the coarsest grid and maximum CFL = 200 ramped over 50 cycles. The initial solutions on the target grids are obtained by the FMG method and, thus, differ between AgMG and StMG solvers. Nevertheless, the number of target-grid cycles to converge to the prescribed residual tolerance is similar. To illustrate convergence starting from the same initial solution, residual and lift convergence of a solution at $\alpha = 2.5°$ starting from the converged solution at $\alpha = 1.93°$ is compared in figures 16 and 17.

### V.E. NACA 0012 airfoil

We consider a turbulent flow over the NACA 0012 airfoil. The finest mesh is the structured C-type mesh from the Turbulence Modeling Resource website[29] of dimensions 1793×513 around and normal to the body; the free-stream Mach number and Reynolds number are taken from the website (the Mach number is 0.15 and $Re = 6.0 \times 10^6$/(unit grid length)). There are three levels of primal grids: Grid 1 with 919,424 nodes, Grid 2 with 230,336 nodes, and Grid 3 with 57,824 nodes. They are all structured quadrilateral grids. We apply the three-level AgMG to each grid independently with 48 processors for Grid 1, 8 processors for Grid 2, and 2

processors for Grid 3. Tolerance is set to be $10^{-10}$ in the residual norm. Agglomeration was performed by the line-1 agglomeration strategy over the entire domain. The line passing through the trailing edge was treated as described in Section IV.E.

Three angles of attack, $\alpha = 0°$, $\alpha = 10°$, and $\alpha = 15°$, are considered. For $\alpha = 0°$, as indicated in table 3, the SG solver is extremely slow to converge while AgMG converged significantly faster; and StMG is even faster than AgMG. For $\alpha = 10°$, the number of SG cycles required for convergence significantly increases for finer grids. Both AgMG and StMG methods exhibit convergence that is less sensitive to the grid size. Similar results are obtained for $\alpha = 15°$. For detailed convergence histories, see figures 18 to 20 for $\alpha = 0°$, figures 21 to 23 for $\alpha = 10°$, and figures 24 to 26 for $\alpha = 15°$. To illustrate convergence starting from the same initial solution, residual and lift convergence of a solution at $\alpha = 15°$ starting from the converged solution at $\alpha = 10°$ is compared in figures 27 and 28.

### V.F. 3D Hemisphere Cylinder

We consider 3D inviscid, laminar, and turbulent flows over a hemisphere cylinder with the radius 0.5 and the length 20.0 in the grid units. The geometry is taken from the experimental model studied in Ref.[34]. In all cases, Mach number is 0.6, and the angle of attack is 5.0 degrees. The outer boundary is located approximately at the average distance of 60 radii from the body. The computational grid is tetrahedral for the inviscid case and mixed element for the viscous cases. The tolerance is set to be $10^{-14}$ in the residual $L_2$ norm. Primal grids with no polar singularity as shown in figure 2(b) were employed.

For the inviscid flow, a tetrahedral grid with 4,088,874 nodes was used. The number of processors is 64 so that each partition has about 60,000 nodes. The coarse grids are generated to the third level: two agglomerated grids and the target grid, see figure 29. Implicit-lines are not formed for this grid, and thus the agglomeration was performed without the line-agglomeration. As can be seen in table 5, the AgMG solver is about 5 times faster than the SG solver. See figure 30(a) for convergence histories.

For the laminar flow, we set $Re = 4.0 \times 10^2$/(unit grid length). The grid has a prismatic layer off the body and tetrahedra elsewhere with the total number of nodes 4,262,115; 64 processors are used. Implicit-lines are formed in the prismatic layer, and the line-agglomeration is performed within the layer. In this case, the AgMG solver is about 8 times faster than the SG solver as can be seen in table 5. Figure 30(b) shows convergence histories.

For a turbulent flow with $Re = 3.5 \times 10^5$/(unit grid length), five grids are generated to study the grid dependence: 1 million-node grid to 15 million-node grid. Each grid is a mixed element grid similar to the one employed for the laminar case, but the prismatic layer has a stronger stretching to keep $y^+ < 1$. Implicit-lines are formed in the prismatic layer, and the line-agglomeration is performed within the layer. As indicated in table 5, the number of processors are chosen to keep approximately 60,000 nodes per partition. The results in the table indicate that the AgMG solver significantly accelerates convergence over the SG solver, and the speed-up factor increases on finer grids. The speed-up factor grows because AgMG convergence is almost grid-independent whereas SG convergence deteriorates on finer grids. Moreover, since each partition has the same number of nodes for all grids, the grid-independent convergence is achieved not only in the number of cycles but also in the CPU time as can be seen in figure 31. Typical convergence histories are also shown in figures 30(c) and 30(d) for Grid 2.

For inviscid, laminar, and turbulent flows, the AgMG solver is significantly faster than the SG solver. The acceleration is demonstrated also in convergence of the drag coefficient for turbulent-flow computations on the target grid as shown in figure 32 for Grid 2. Numerical solution obtained by the AgMG solver for Grid 5 is compared with the experimental data in figure 33. Good agreement demonstrates that the AgMG method produces a correct solution with a significant speed-up.

### V.G. Wing-Body-Tail Configuration

We consider a turbulent flow over a wing-body-tail configuration with Mach number 0.85, $\alpha = 2.5$ degrees, and $Re = 5.0 \times 10^6$/(unit grid length), taken from Drag Prediction Workshop 4 (DPW4).[35] Results are obtained for two mixed element grids: a coarse grid of approximately 3.5 million nodes and a medium grid of approximately 10 million nodes. The tolerance is set to be $10^{-8}$ in the residual $L_2$ norm. Again, the AgMG has 3 levels; coarser grids are shown in figure 34. For this case, 50 cycles are performed at each coarsest-grid level within the FMG. Also, the turbulence diffusion is discretized by the edge-terms-only scheme on all grids. Results are given in table 5. On both grids, AgMG accelerates the convergence over SG by a factor of 7 on the primal grid level. See figure 35 for complete convergence histories.

**Table 5. Three dimensional test cases (RANS unless otherwise stated): the number of cycles required for convergence on the primal grid by SG and 3-level AgMG.**

| Case (stopping tolerance) | Grid (nodes/processors) | SG | AgMG |
|---|---|---|---|
| Hemisphere-Cylinder - Inviscid ($10^{-14}$) | 4,088,874/64 | 418 | 78 |
| Hemisphere-Cylinder - Laminar ($10^{-14}$) | 4,262,115/64 | 338 | 39 |
| Hemisphere-Cylinder ($10^{-14}$) | Grid 1 (1,017,289/16) | 270 | 74 |
| | Grid 2 (4,040,409/64) | 331 | 81 |
| | Grid 3 (7,087,899/112) | 513 | 85 |
| | Grid 4 (10,486,091/168) | 652 | 85 |
| | Grid 5 (15,285,776/246) | 1652 | 86 |
| Wing-Body-Tail (DPW4) ($10^{-8}$) | Grid 1 (3,675,916/56) | 946 | 130 |
| | Grid 2 (10,252,669/144) | 725 | 102 |

**Table 6. Improvement in cycles to converge for finest grids of 3D RANS test cases.**

| Case | AgMG Improvement |
|---|---|
| Hemisphere-Cylinder | 19.2 |
| Wing-Body-Tail (DPW4) | 7.1 |

Table 4 shows the improvement in cycles to convergence with the AgMG and StMG solvers relative to the single-grid solver for the finest grid 2D RANS results reported above. Table 6 shows the corresponding improvement with the AgMG solver for the finest grid 3D results. In 2S, the improvement ranges from 2 to 16 for the AgMG solver and is less than but comparable with the StMG solver for the two airfoil cases. The improvement in 3D is between 7 and 20.

## VI. Conclusions

The research presented in this paper is work in progress. An agglomeration multigrid (AgMG) solver developed and reported in previous papers has been systematically applied to a series of test cases ranging from simple 2D flows to some realistic 3D configurations. The goal of this systematic study is to advance the AgMG solver toward automated agglomeration, discrete solvability, and superior efficiency for a wide range of practical turbulent flows. The approach we follow is to identify, isolate, and fix issues encountered as we progress from simpler to more complicated tests. Convergence of the developed AgMG solver has been compared with convergence of an advanced single-grid (SG) solver and, in some cases, with convergence of a structured multigrid (StMG) solver. The SG solver used in this study as a convergence benchmark and as a relaxation scheme for both AgMG and StMG solvers is itself a new scheme based on a Jacobian-free Newton-Krylov approach with a preconditioner using defect correction and an adaptive pseudo-time step. Although not shown here, this SG solver represents a significant improvement over the baseline scheme used in the state-of-the-art flow solver, FUN3D. The convergence of the StMG solver on structured primal grids provided the performance targets for the AgMG solver on the same grids.

For all reported cases, the AgMG solver has significantly accelerated convergence compared to the SG solver convergence and showed performance comparable with the StMG solver performance with similar multigrid cycle parameters. Further improvements are expected if better gradient reconstruction schemes are used for turbulence source computations on agglomerated grids. Although most comparisons are done in terms of cycles, similar gains are expected in terms of computer times. Note the SG method uses grid sequencing that relies on agglomeration and without this, the SG method would require more iterations to converge.

In spite of demonstrations of significant speed-up over the SG method, the current state of AgMG development is far from completion. The systematic advancement from simpler to more complicated cases required algorithmic adjustments at every step in the process. For example, a relatively low limit on CFL number in the

pseudo time has been imposed to provide convergence for a wide range of test cases, even if a higher CFL would improve convergence rates in many cases. A limitation on the minimum number of nodes per partition has been enforced to ensure scalability in parallel computations. In 3D wing-body-tail computations, the viscous terms of the turbulent equations have been discretized with an inconsistent, but positive edge-terms-only scheme; otherwise convergence problems have been encountered. With a high degree of confidence based on our experiences to date, we expect our AgMG solver with the current set of parameters to work well for general inviscid and laminar flows and for 2D turbulent flows. While the current AgMG solver shows encouraging performance for some 3D turbulent flows, the search for a robust set of parameters and for algorithmic improvements in applications to general 3D RANS equations is still ongoing. Specifically, the current AgMG solver does not provide reliable performance for turbulent flows on structured grids around the DPW5 configuration. The SG solver converges for all test cases and the StMG solver has shown improvements over the SG solver in many of those.

## Acknowledgments

The first and second authors acknowledge the support from the NASA contract #NNL12AB00T. This work was supported by the NASA Fundamental Aeronautics (FA) Program. All results presented were computed within the FUN3D suite of codes at NASA Langley Research Center.

## References

[1]Trottenberg, U., Oosterlee, C. W., and Schüller, A., *Multigrid*, Academic Press, 2000.

[2]Briggs, W. L., Henson, V. E., and McCormick, S. F., *A Multigrid Tutorial*, SIAM, 2nd ed., 2000.

[3]Lallemand, M.-H., Steve, H., and Dervieux, A., "Unstructured Multigridding by Volume Agglomeration: Current status," *Computers and Fluids*, Vol. 21, No. 3, 1992, pp. 397–433.

[4]Smith, W. A., "Multigrid Solution of Transonic Flow on Unstructured Grids," *Recent Advances and Applications in Computational Fluid Dynamics, Proceedings of the ASME Winter Annual Meeting*, edited by O. Baysal, November 1990.

[5]Mavriplis, D. J., "Multigrid Techniques for Unstructured Meshes," *VKI Lecture Series VKI-LS 1995-02*, edited by H. Deconinck and M. Ricchiuto, Von Karman Institute for Fluid Dynamics, Rhode-Saint-Genese, Belgium, 1995.

[6]Mavriplis, D. J., "Unstructured Grid Techniques," *Annual Review of Fluid Mechanics*, Vol. 29, 1997, pp. 473–514.

[7]Mavriplis, D. J. and Pirzadeh, S., "Large-Scale Parallel Unstructured Mesh Computations for 3D High-Lift Analysis," *Journal of Aircraft*, Vol. 36, No. 6, 1999, pp. 987–998.

[8]Mavriplis, D. J., "An Assessment of Linear versus Non-Linear Multigrid Methods for Unstructured Mesh Solvers," *Journal of Computational Physics*, Vol. 275, 2002, pp. 302–325.

[9]Mavriplis, D. J. and Long, M., "NSU3D Results for the Fourth AIAA Drag Prediction Workshop," AIAA Paper 2010-4363, 2010.

[10]Long, M. and Mavriplis, D. J., "NSU3D Results for the First AIAA High Lift Prediction Workshop," AIAA Paper 2011-863, 2011.

[11]Nishikawa, H., Diskin, B., and Thomas, J. L., "Critical Study of Agglomerated Multigrid Methods for Diffusion," *AIAA Journal*, Vol. 48, No. 4, April 2010, pp. 839–847.

[12]Thomas, J. L., Diskin, B., and Nishikawa, H., "A Critical Study of Agglomerated Multigrid Methods for Diffusion on Highly-Stretched Grids," *Computers and Fluids*, Vol. 41, No. 1, February 2011, pp. 82–93.

[13]Nishikawa, H., Diskin, B., and Thomas, J. L., "Development and Application of Agglomerated Multigrid Methods for Complex Geometries," *40th AIAA Fluid Dynamics Conference and Exhibit*, AIAA Paper 2010-4731, Chicago, 2010.

[14]Nishikawa, H. and Diskin, B., "Development and Application of Parallel Agglomerated Multigrid Methods for Complex Geometries," *20th AIAA Computational Fluid Dynamics Conference*, AIAA Paper 2011-3232, Hawaii, 2011.

[15]Anderson, W. K. and Bonhaus, D. L., "An implicit upwind algorithm for computing turbulent flows on unstructured grids," *Computers and Fluids*, Vol. 23, 1994, pp. 1–21.

[16]Lee-Rausch, E. M., Hammond, D. P., Nielsen, E. J., Pirzadeh, S. Z., and Rumsey, C. L., "Application of the FUN3D Unstructured-Grid Navier-Stokes Solver to the 4th AIAA Drag PredictionWorkshop Cases," *28th AIAA Applied Aerodynamics Conference*, AIAA Paper 2010-4551, Chicago, 2010.

[17]Lucas, P., van Zuijlen, A. H., and Bijl, H., "Fast unsteady flow computations with a Jacobian-free NewtonKrylov algorithm," *Journal of Computational Physics*, Vol. 229, No. 24, 2010, pp. 9201–9215.

[18]Osusky, M. and Zingg, D. W., "A parallel Newton-Krylov-Shur flow solver for the Reynolds-Averaged Navier-Stokes equations," *50th AIAA Aerospace Sciences Meeting*, AIAA Paper 2012-0442, Nashville, 2012.

[19]Diskin, B., Thomas, J. L., and Brandt, A., "Distributed Relaxation Multigrid and Defect Correction Applied to the Compressible Navier-Stokes Equations," *14th AIAA Computational Fluid Dynamics Conference*, AIAA Paper 1999-3334, Norfolk, 1999.

[20]Diskin, B. and Thomas, J. L., "Convergence of defect-correction and multigrid iterations for inviscid flows," *20th AIAA Computational Fluid Dynamics Conference*, AIAA Paper 2011-3235, Hawaii, 2011.

[21]MacLean, M. and White, T., "Implementation of Generalized Minimum Residual Krylov Subspace Method for Chemically Reacting Flows," *50th AIAA Aerospace Sciences Meeting*, AIAA Paper 2012-0441, Nashville, 2012.

[22]Allmaras, S. R., Bussoletti, J. E., Hilmes, C. L., Johnson, F. T., Melvin, R. G., Tinoco, E. N., Venkatakrishnan, V., Wigton, L. B., and Young, D. P., "Algorithm Issues and Challenges Associated with the Development of Robust CFD Codes," *Variational Analysis and Aerospace Engineering*, Vol. 33, Springer-Verlag, 2009, pp. 1–19.

[23]Edwards, J. R. and Thomas, J. L., "Development of $O(Nm^2)$ Preconditioned Multigrid Solvers for Euler and Navier-Stokes Equations," *AIAA Journal*, Vol. 38, No. 4, 2000, pp. 717–720.

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

[25]Diskin, B., Thomas, J. L., Nielsen, E. J., Nishikawa, H., and White, J. A., "Comparison of Node-Centered and Cell-Centered Unstructured Finite-Volume Discretizations: Viscous Fluxes," *AIAA Journal*, Vol. 48, No. 7, July 2010, pp. 1326–1338.

[26]Nishikawa, H., "Beyond Interface Gradient: A General Principle for Constructing Diffusion Schemes," *40th AIAA Fluid Dynamics Conference and Exhibit*, AIAA Paper 2010-5093, Chicago, 2010.

[27]Roe, P. L., "Approximate Riemann Solvers, Parameter Vectors, and Difference Schemes," *Journal of Computational Physics*, Vol. 43, 1981, pp. 357–372.

[28]Diskin, B. and Thomas, J. L., "Accuracy Analysis for Mixed-Element Finite-Volume Discretization Schemes," *NIA Report No. 2007-08*, 2007.

[29]"Turbulence Modeling Resource Website," `http://turbmodels.larc.nasa.gov`, December, 2012.

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

[31]Nishikawa, H., "Robust and Accurate Viscous Discretization via Upwind Scheme - I: Basic Principle," *Computers and Fluids*, Vol. 49, No. 1, 2011, pp. 62–86.

[32]Ceze, M. and Fidkowski, K., "A Robust Adaptive Solution Strategy for High-Order Implicit CFD solvers," *20th AIAA Computational Fluid Dynamics Conference*, AIAA Paper 2011-3696, Hawaii, 2011.

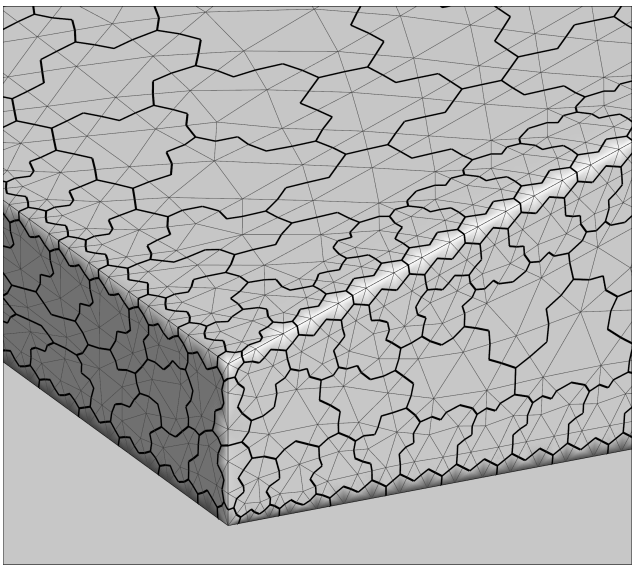
[33]Diskin, B. and Thomas, J. L., "Comparison of Node-Centered and Cell-Centered Unstructured Finite-Volume Discretizations: Inviscid Fluxes," *AIAA Journal*, Vol. 49, No. 4, 2011, pp. 836–854.

[34]Tsieh, T., "An Investigation of Separated Flow About a Hemisphere-Cylinder at 0- to 19-Deg Incidence in the Mach Number Range from 0.6 to 1.5," AEDC-TR-76-112, 1976.

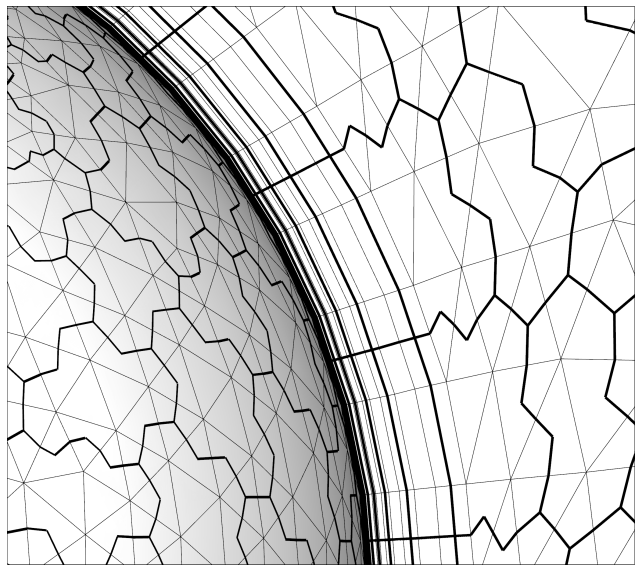
[35]"4th AIAA CFD Drag Prediction Workshop Website," `http://aaac.larc.nasa.gov/tsab/cfdlarc/aiaa-dpw/Workshop4`, December, 2012.

[36]"5th AIAA CFD Drag Prediction Workshop Website," `http://aaac.larc.nasa.gov/tsab/cfdlarc/aiaa-dpw`, December, 2012.

[37]Park, M. A., Laflin, K. R., Chaffin, M. S., Powell, N., and Levy, D. W., "CFL3D, FUN3D, and NSU3D Contributions to the Fifth Drag Prediction Workshop," 51st Aerospace Sciences Meeting, Grapevine, Texas, American Institute of Aeronautics and Astronautics, Reston, VA (submitted for publication).

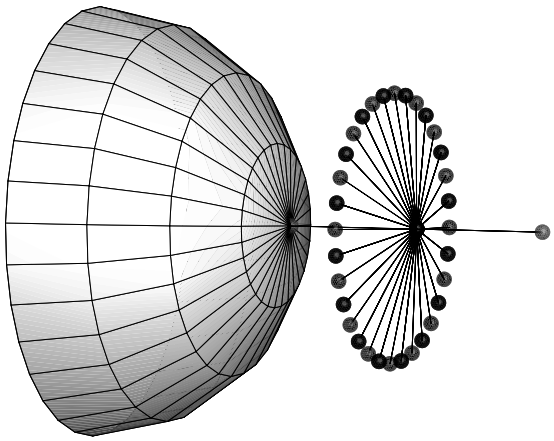


(a) Agglomerated trailing-edge area of a 3D wing. Primal grid is shown by thin lines; agglomerated grid is shown by thick lines.

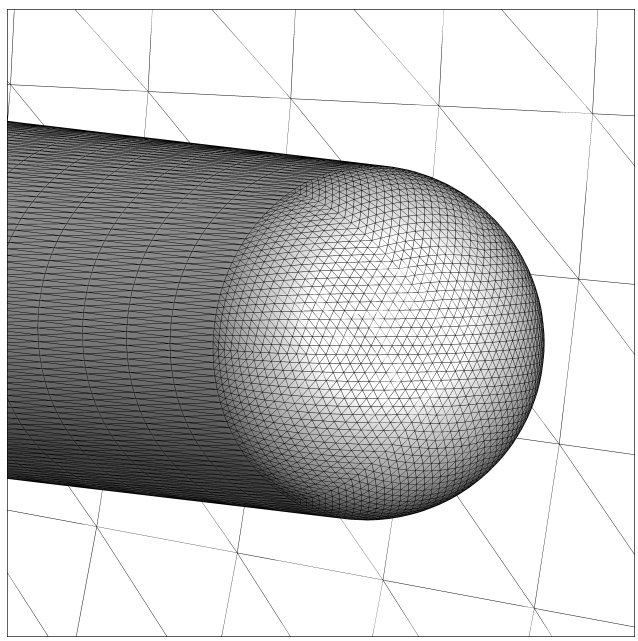


(b) Typical implicit line-agglomeration showing a curved solid body surface on the left and a symmetry plane on the right. The projection of the line-agglomerations can be seen on the symmetry plane.

**Figure 1. Hierarchical agglomeration scheme.**

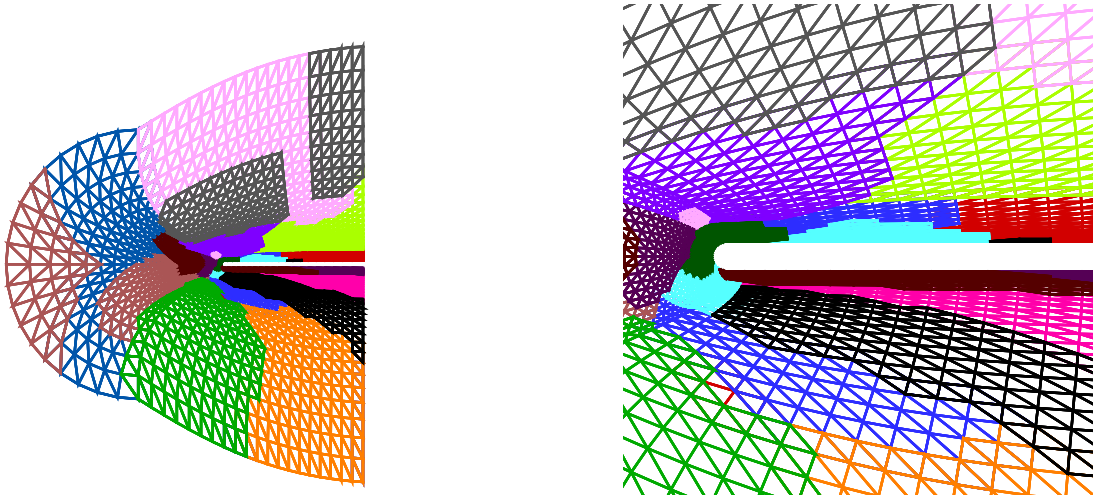


(a) Illustration of the least-squares stencil (circles) on a grid with polar singularity.



(b) A grid with no polar singularity. The apex node always has six neighbors for grids of any size.

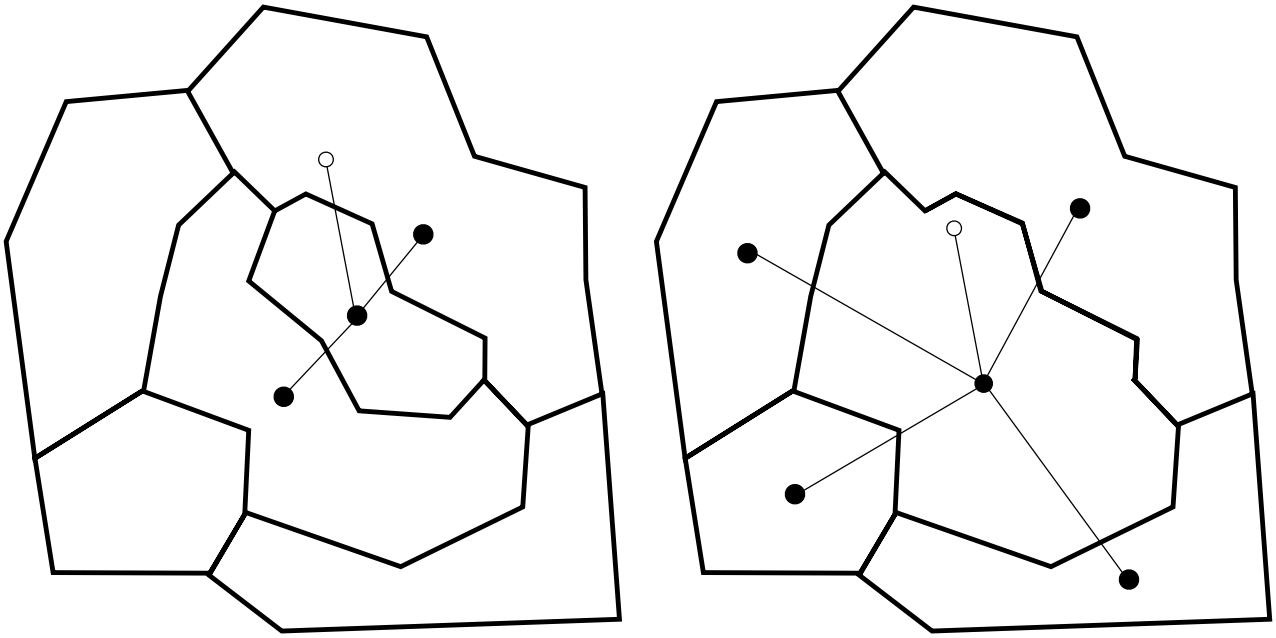
**Figure 2. 3D hemisphere cylinder grid.**



(a) Far field view.

(b) Near field view.

**Figure 3. Example of disjoint grids: a partitioned triangular grid for the 2D hemisphere cylinder grid (2135 nodes). Partitions are indicated by colors.**



(a) Boundary volume having two neighbors on surface. Filled circles indicate the volume centers of the boundary volumes. Unfilled circle indicates the interior volume center above the implicit line from the two-neighbor volume. The four centers form a stencil for the linear interpolation.

(b) Two-neighbor volume merged with a neighbor. Any two neighbors can be selected to form a valid stencil.

**Figure 4. Boundary volume with two neighbors.**

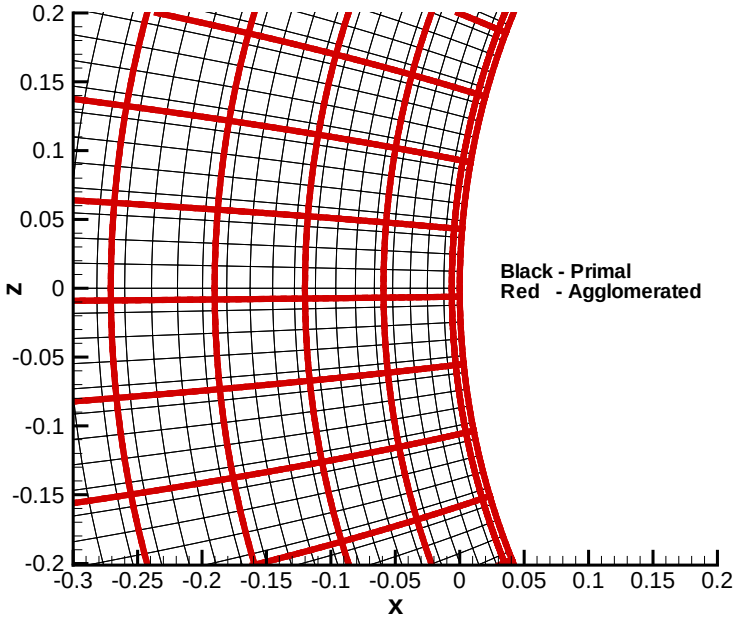



**Figure 5. 2D hemisphere cylinder inviscid grid and the agglomerated grid.**

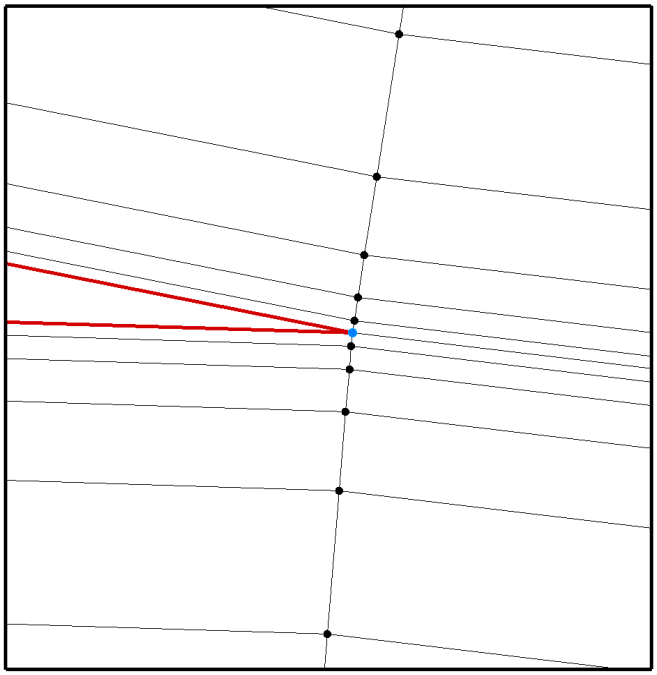
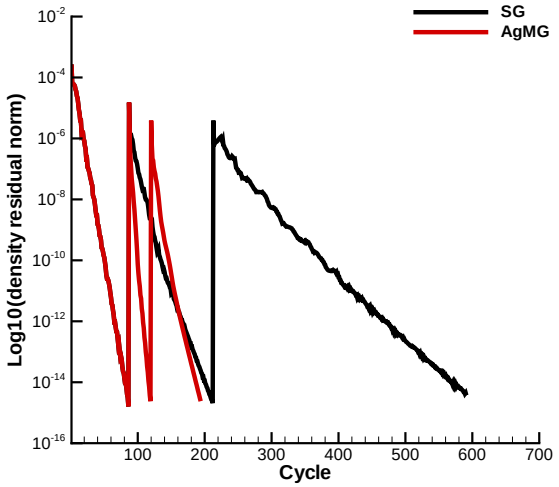
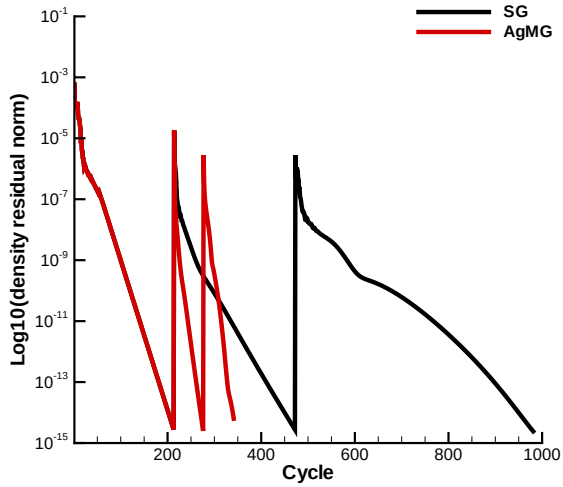


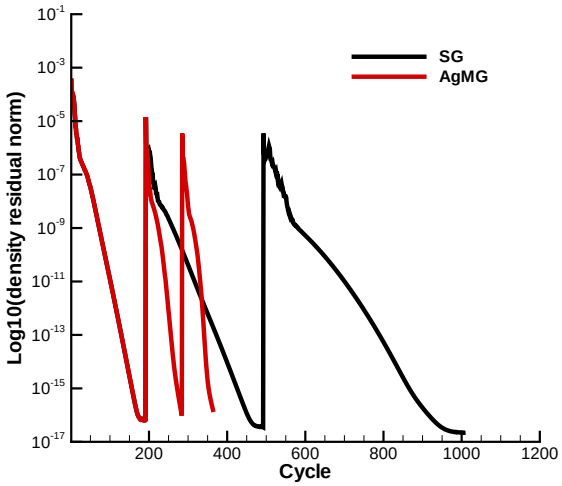
**Figure 6. A line with mixed hierarchies.**



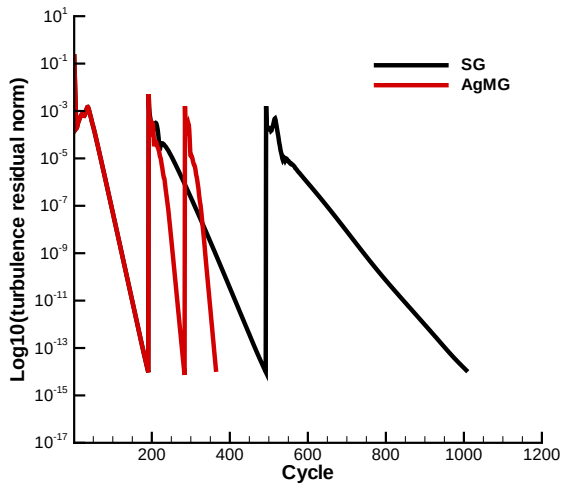

(a) Inviscid flow: Density residual convergence.




(b) Laminar flow: Density residual convergence.

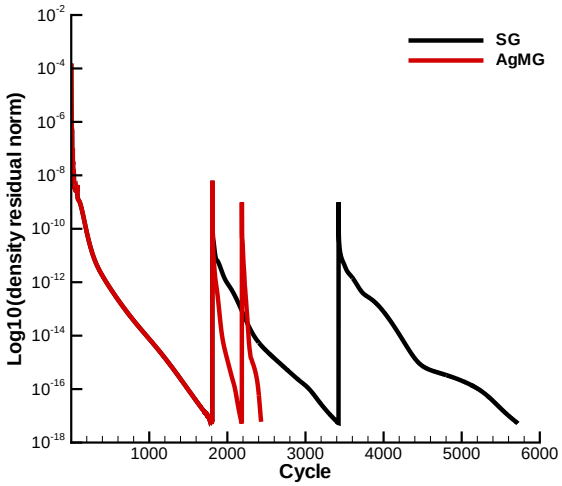



(c) Turbulent flow: Density residual convergence.

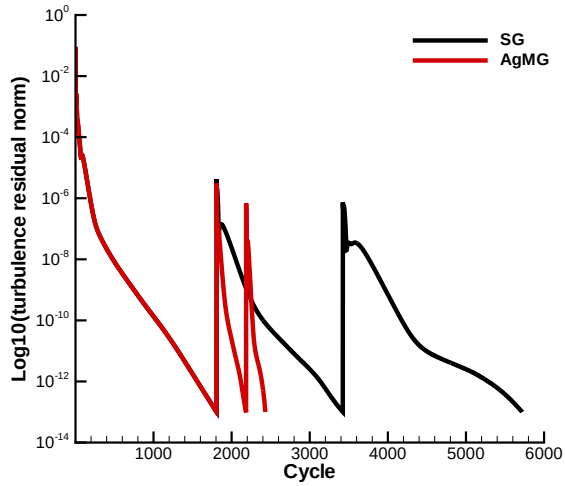



(d) Turbulent flow: Turbulence residual convergence.

**Figure 7. 2D hemisphere cylinder.**

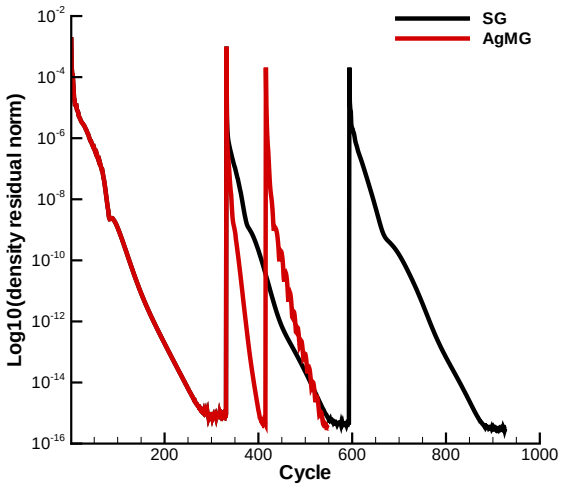



(a) Density residual convergence.

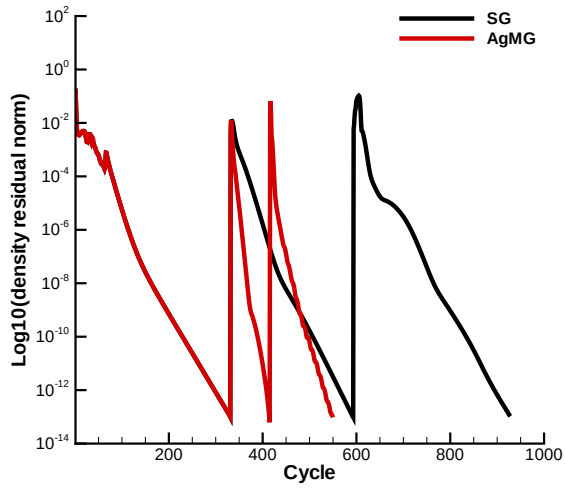



(b) Turbulence residual convergence.

**Figure 8. 2D flat plate 545 × 385 grid.**




(a) Density residual convergence.




(b) Turbulence residual convergence.

**Figure 9. 2D bump 89×41 quadrilateral grid.**

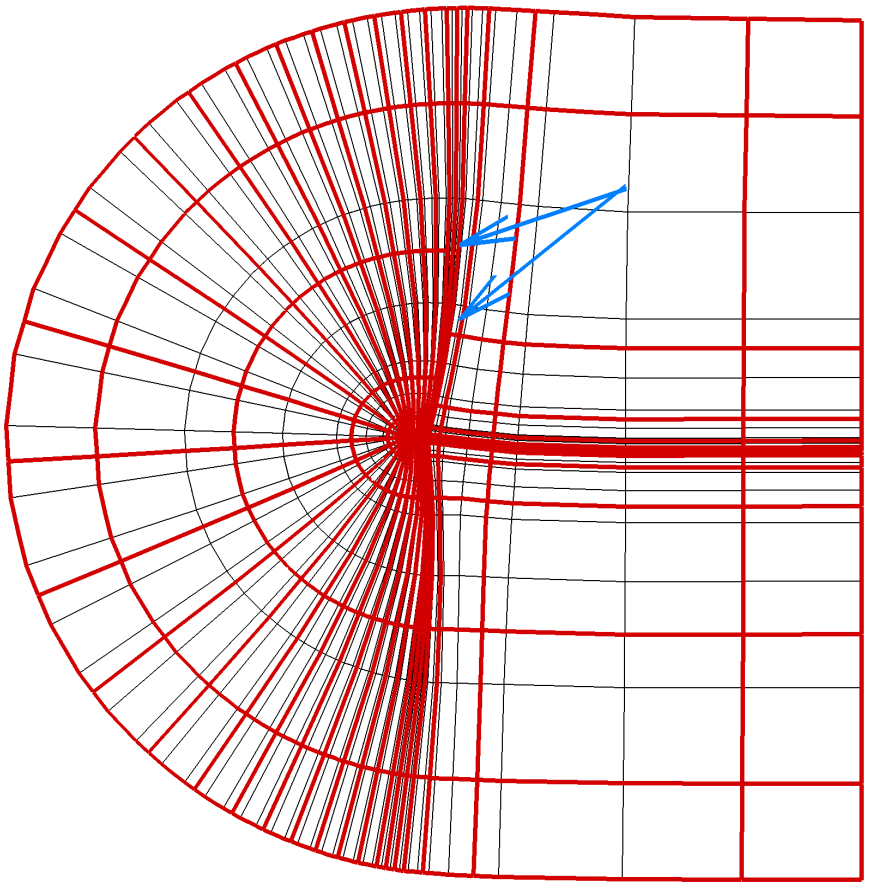


**Figure 10. RAE airfoil, Grid 4 (1616 nodes): primal grid (black) and agglomerated grid (red) generated with line 1 strategy. Arrows point to the gap between lines.**

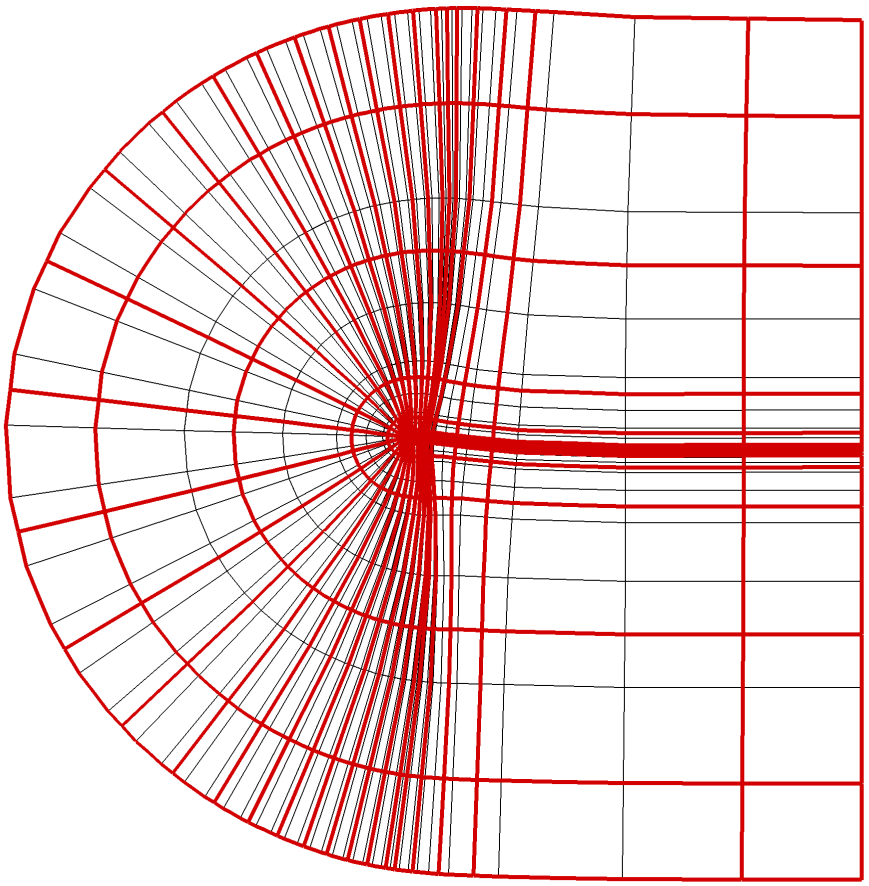


**Figure 11. RAE airfoil, Grid 4 (1616 nodes): primal grid (black) and agglomerated grid (red) generated with line 2 strategy.**

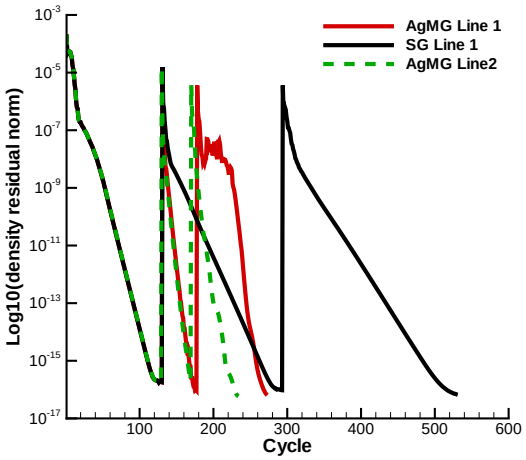



**Figure 12. RAE airfoil, Grid 1 (98,944 nodes): Density residual convergence.**

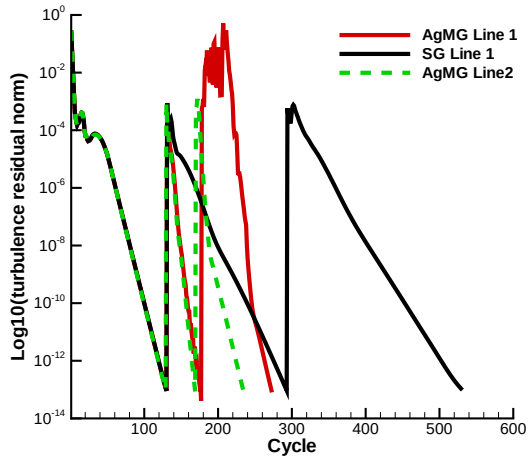



**Figure 13. RAE airfoil, Grid 1 (98,944 nodes): Turbulence residual convergence.**

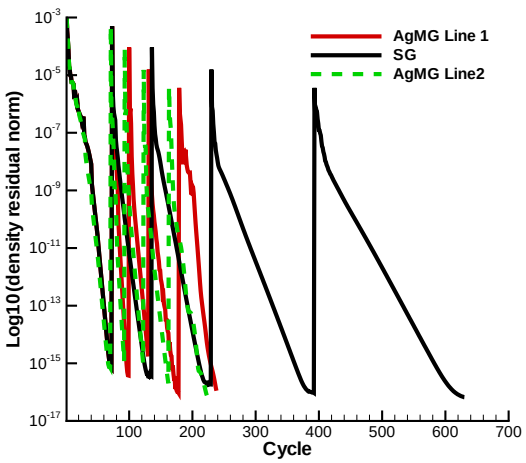



**Figure 14. 5-level multigrid results for RAE airfoil, Grid 1 (98,944 nodes): Density residual convergence.**

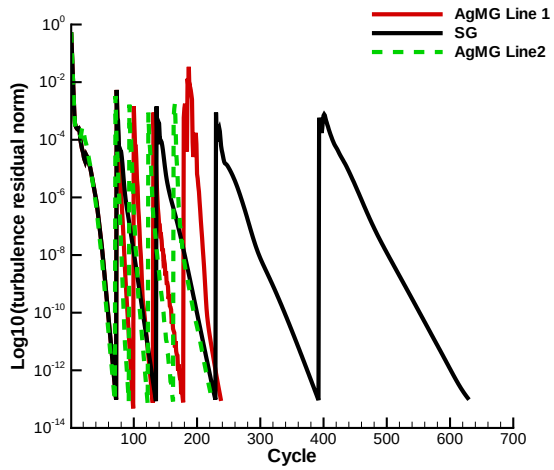



**Figure 15. 5-level multigrid results for RAE airfoil, Grid 1 (98,944 nodes): Turbulence residual convergence.**

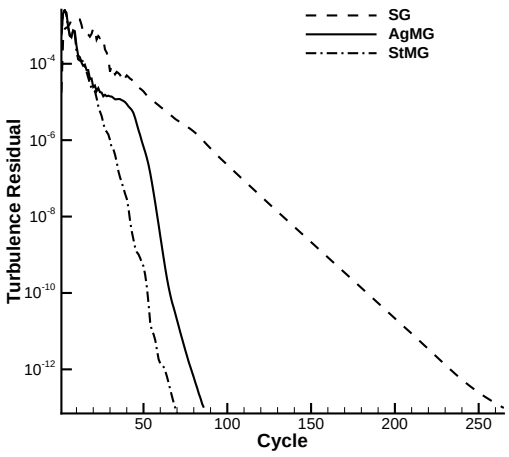



**Figure 16. Residual convergence comparison for SG, AgMG, and StMG, RAE airfoil at $\alpha = 2.5^\circ$, Grid 1.**

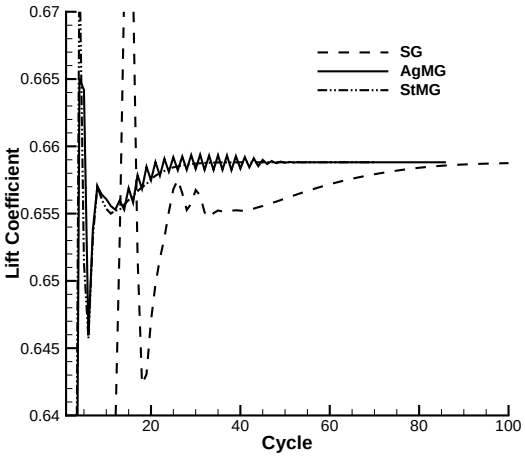
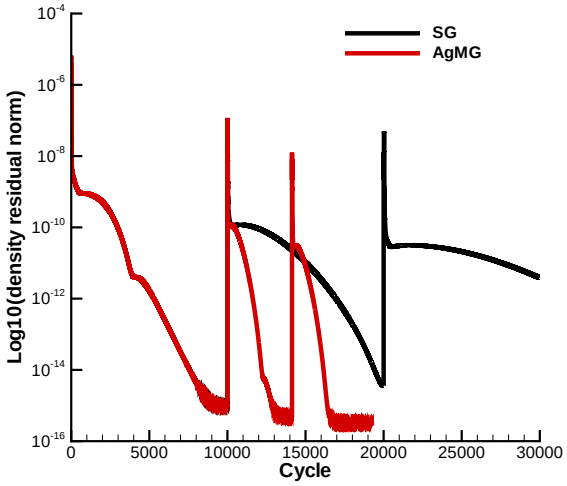
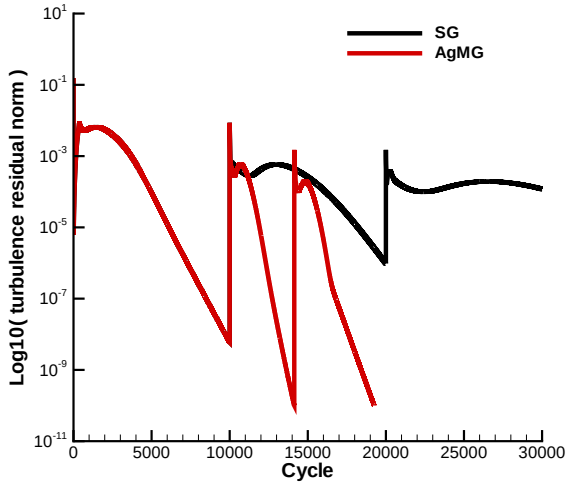



**Figure 17. Lift convergence comparison for SG, AgMG, and StMG, RAE airfoil at $\alpha = 2.5^\circ$, Grid 1.**

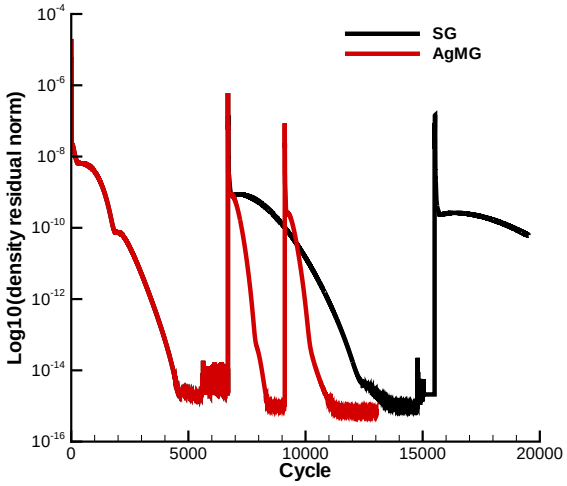



(a) Density residual convergence.

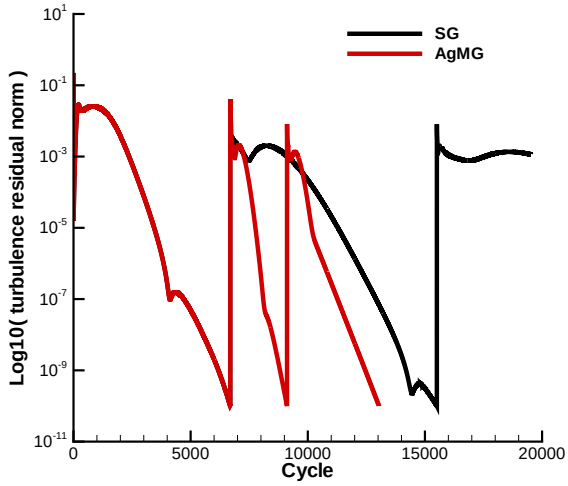



(b) Turbulence residual convergence.

**Figure 18. AgMG: Turbulence residual and density residuals convergence. NACA 0012 airfoil, Grid 1; $\alpha$=0 degree.**

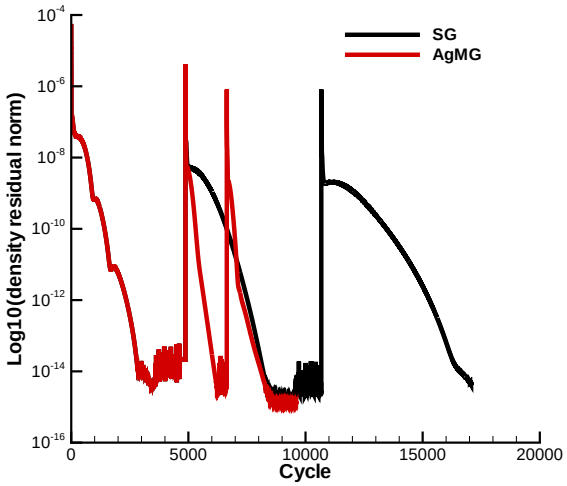



(a) Density residual convergence.

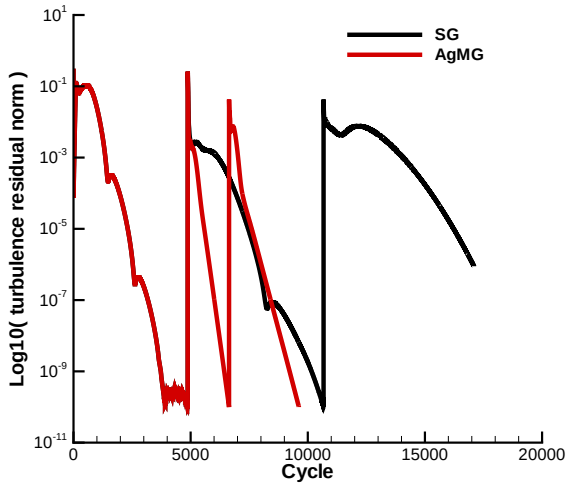



(b) Turbulence residual convergence.

**Figure 19. AgMG: Turbulence residual and density residuals convergence. NACA 0012 airfoil, Grid 2; $\alpha$=0 degree.**

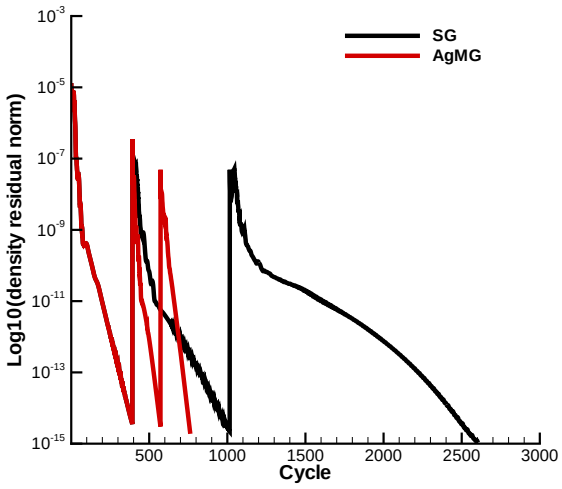



(a) Density residual convergence.

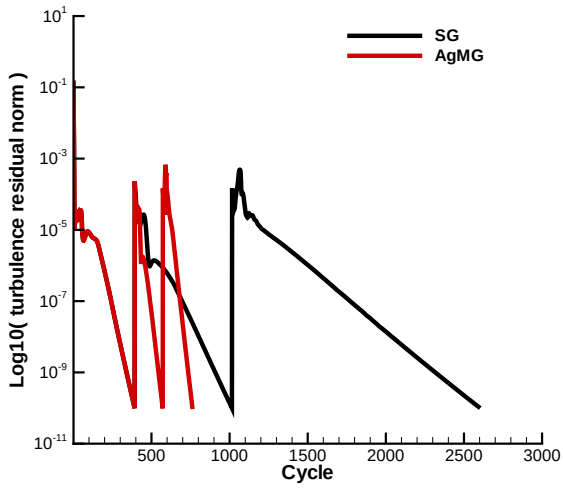



(b) Turbulence residual convergence.

**Figure 20. AgMG: Turbulence residual and denisty residuals convergence. NACA 0012 airfoil, Grid 3; $\alpha$=0 degree.**

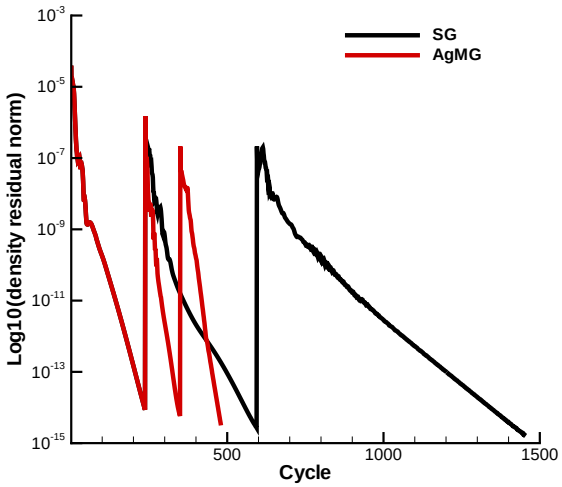



(a) Density residual convergence.

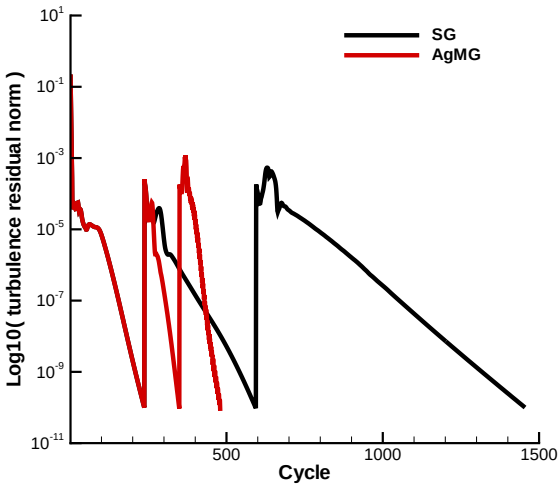



(b) Turbulence residual convergence.

**Figure 21. AgMG: Turbulence residual and density residuals convergence. NACA 0012 airfoil, Grid 1; $\alpha$=10 degrees.**

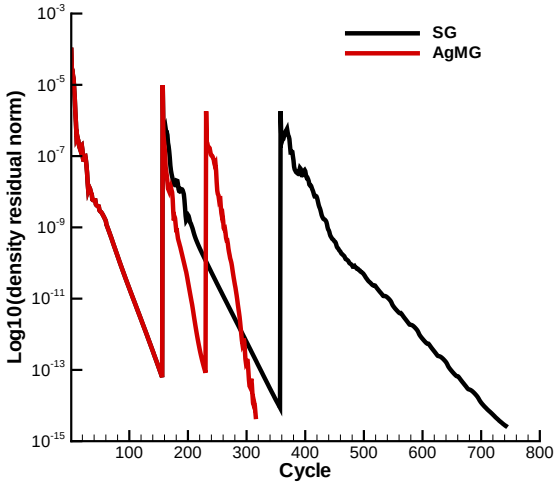



(a) Density residual convergence.

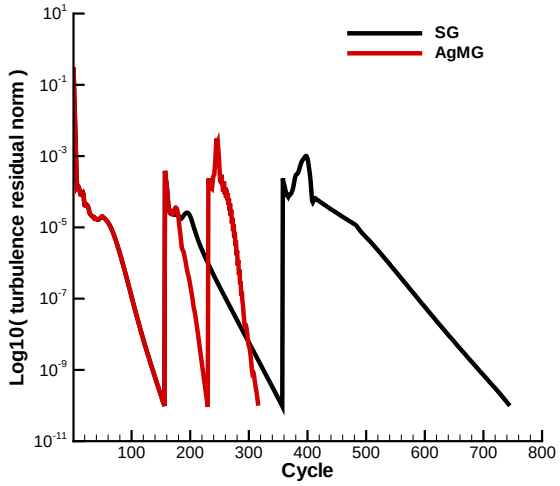



(b) Turbulence residual convergence.

**Figure 22. AgMG: Turbulence residual and density residuals convergence. NACA 0012 airfoil, Grid 2; $\alpha$=10 degrees.**

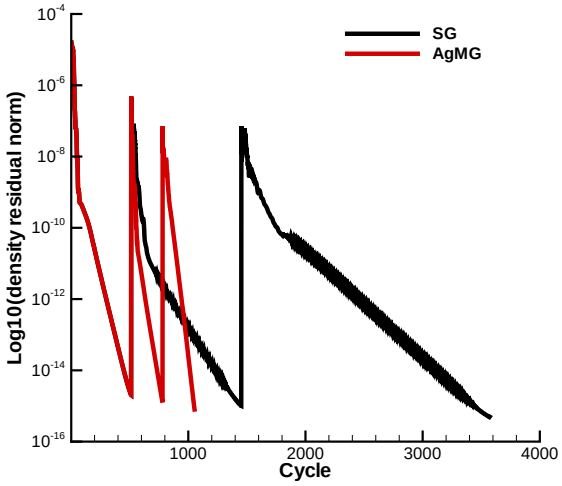



(a) Density residual convergence.

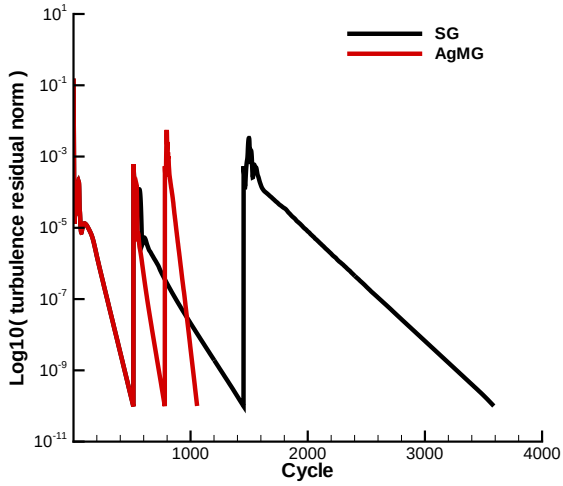



(b) Turbulence residual convergence.

**Figure 23. AgMG: Turbulence residual and density residuals convergence. NACA 0012 airfoil, Grid 3; $\alpha$=10 degrees.**

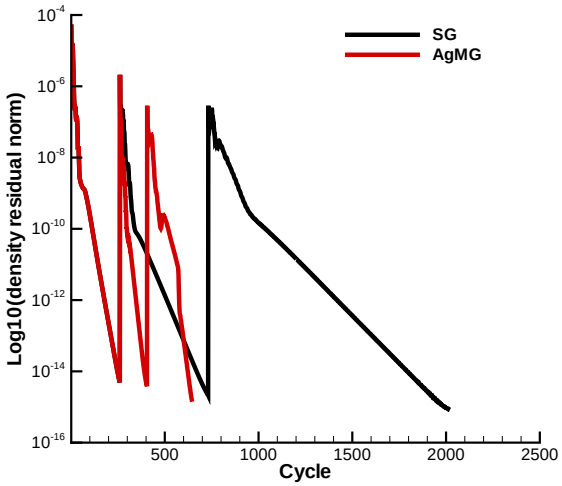



(a) Density residual convergence.

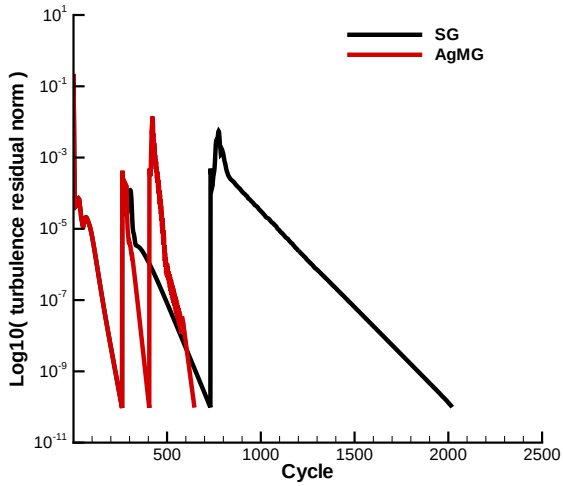



(b) Turbulence residual convergence.

**Figure 24. AgMG: Turbulence residual and density residuals convergence. NACA 0012 airfoil, Grid 1; $\alpha$=15 degrees.**

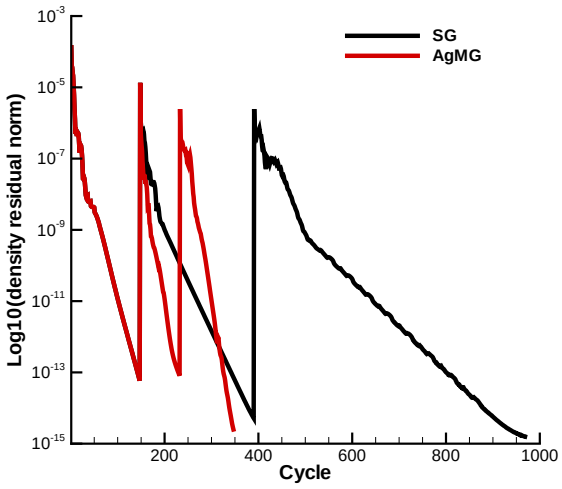



(a) Density residual convergence.

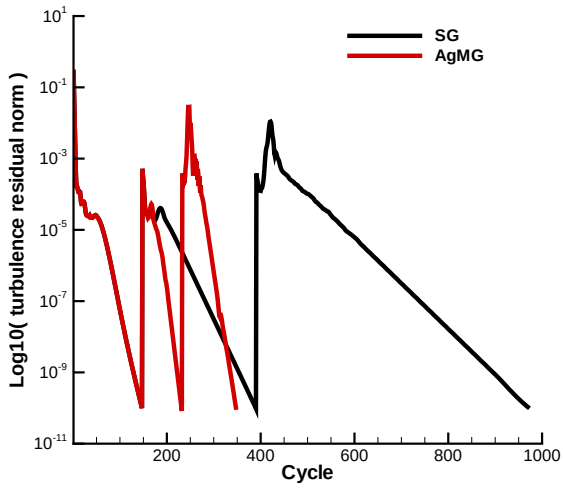



(b) Turbulence residual convergence.

**Figure 25. AgMG: Turbulence residual and density residuals convergence. NACA 0012 airfoil, Grid 2; $\alpha$=15 degrees.**




(a) Density residual convergence.




(b) Turbulence residual convergence.

**Figure 26. AgMG: Turbulence residual and density residuals convergence. NACA 0012 airfoil, Grid 3; $\alpha$=15 degrees.**

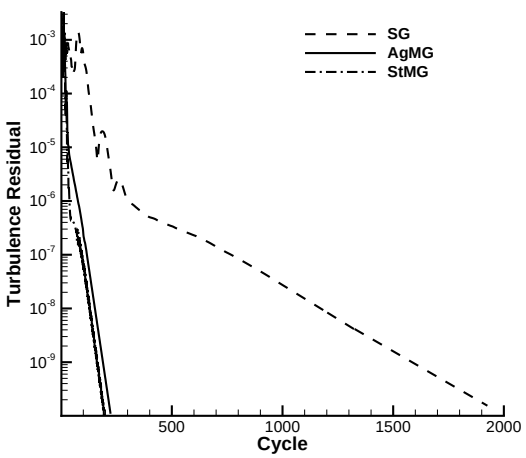



**Figure 27. Residual convergence comparison for SG, AgMG, and StMG for NACA 0012 airfoil at $\alpha$=15 degrees, Grid 1.**

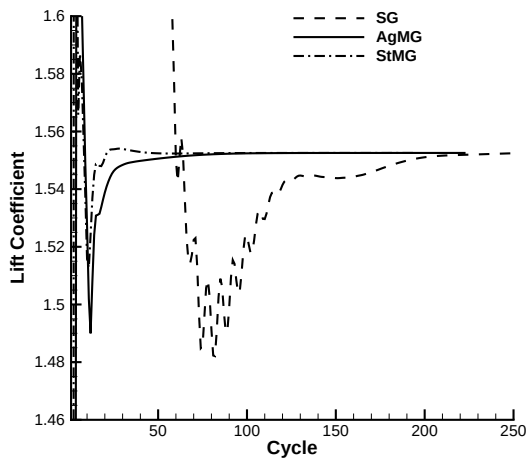
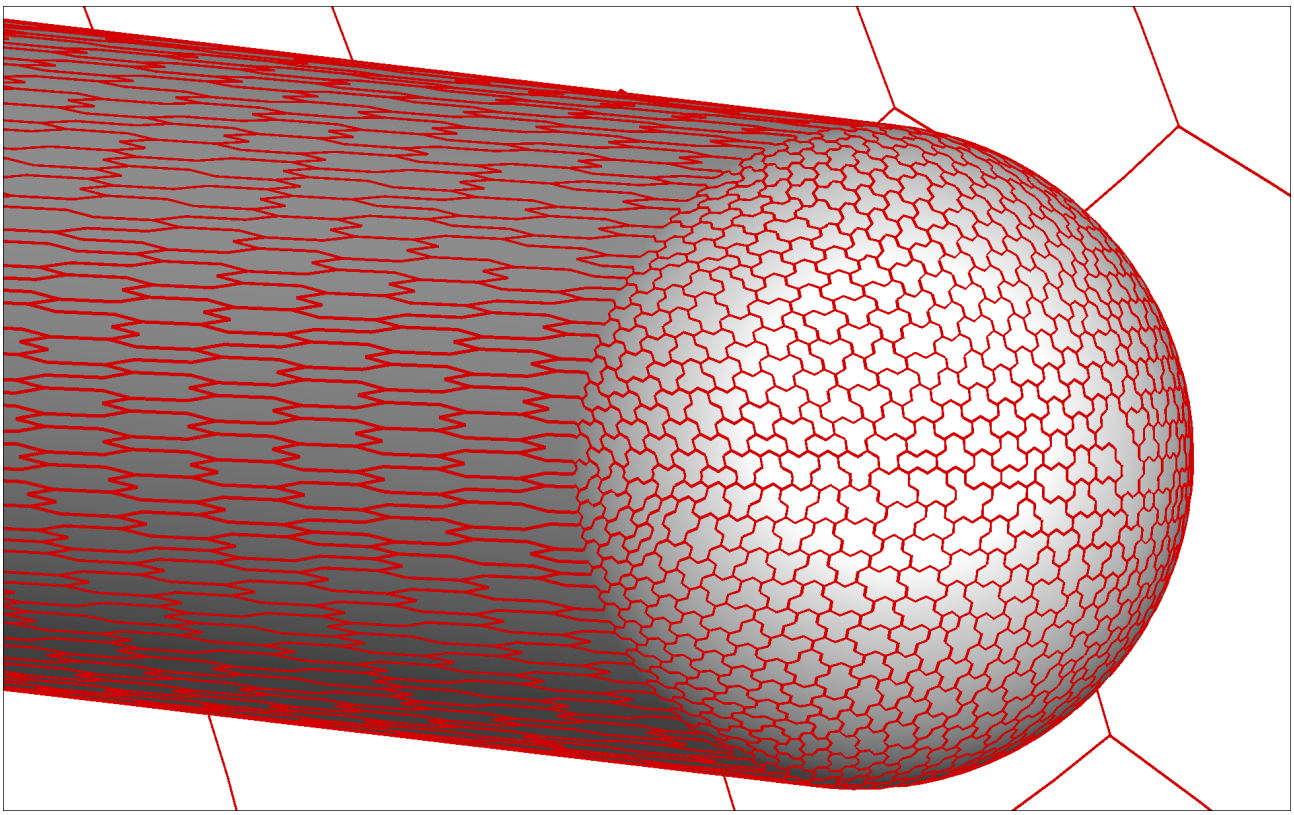
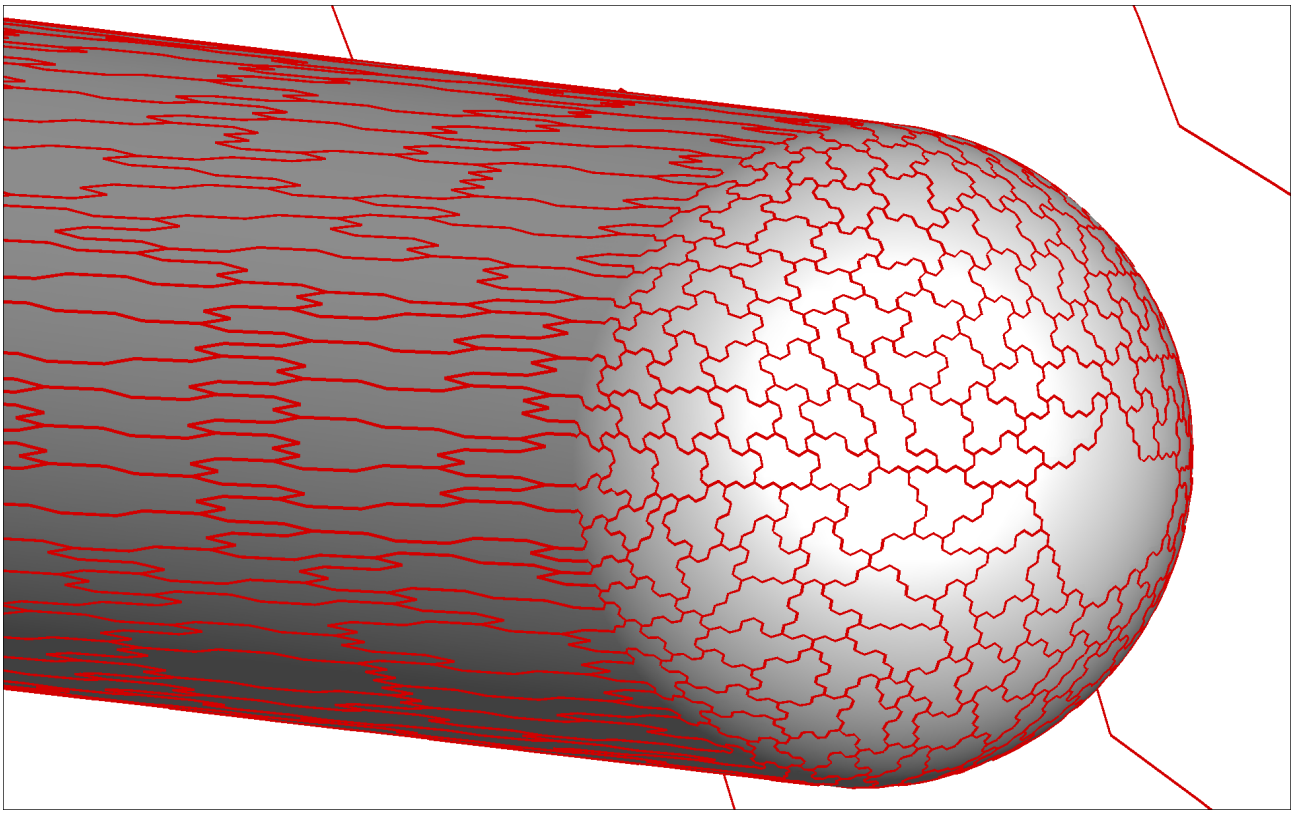



**Figure 28. Lift convergence comparison for SG, AgMG, and StMG for NACA 0012 airfoil at $\alpha$=15 degrees, Grid 1.**

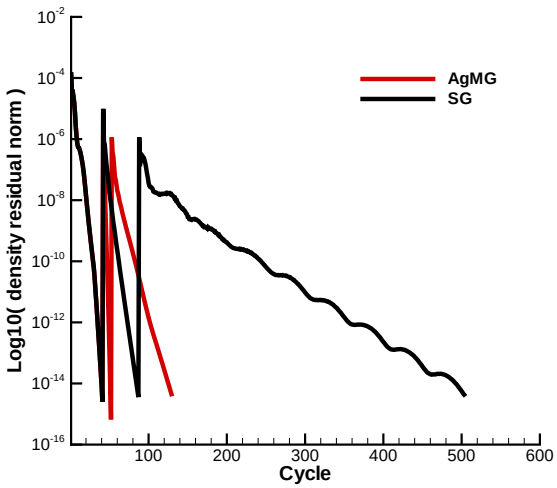


(a) First agglomerated grid.

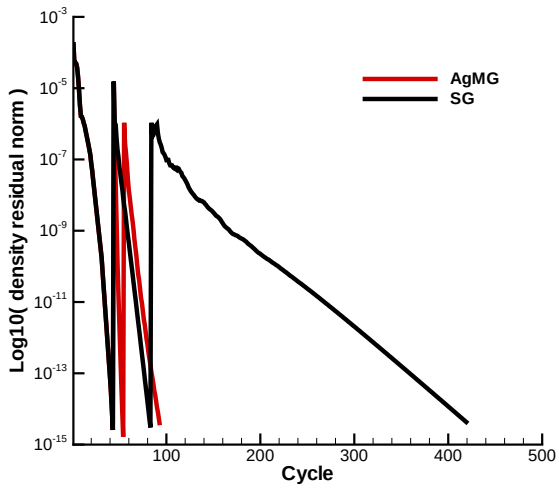


(b) Second agglomerated grid.

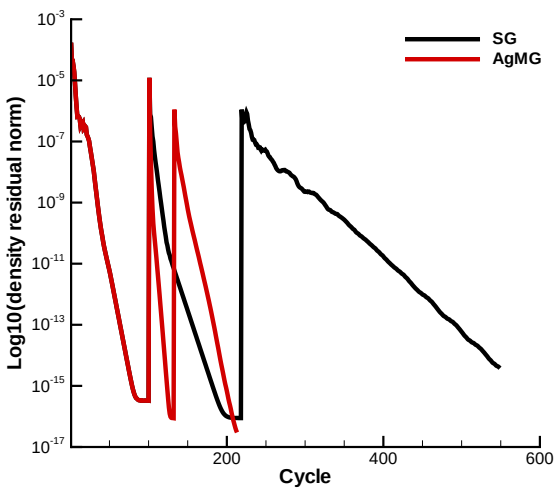
**Figure 29. Agglomerated grids for the 3D hemisphere-cylinder RANS grid (Grid 1).**



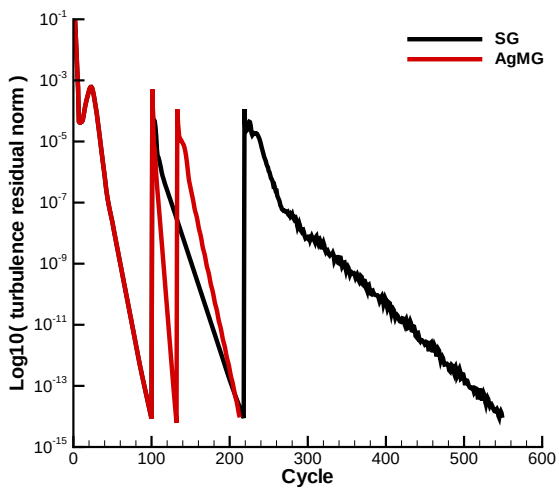

(a) Inviscid flow: Density residual convergence.




(b) Laminar flow: Density residual convergence.

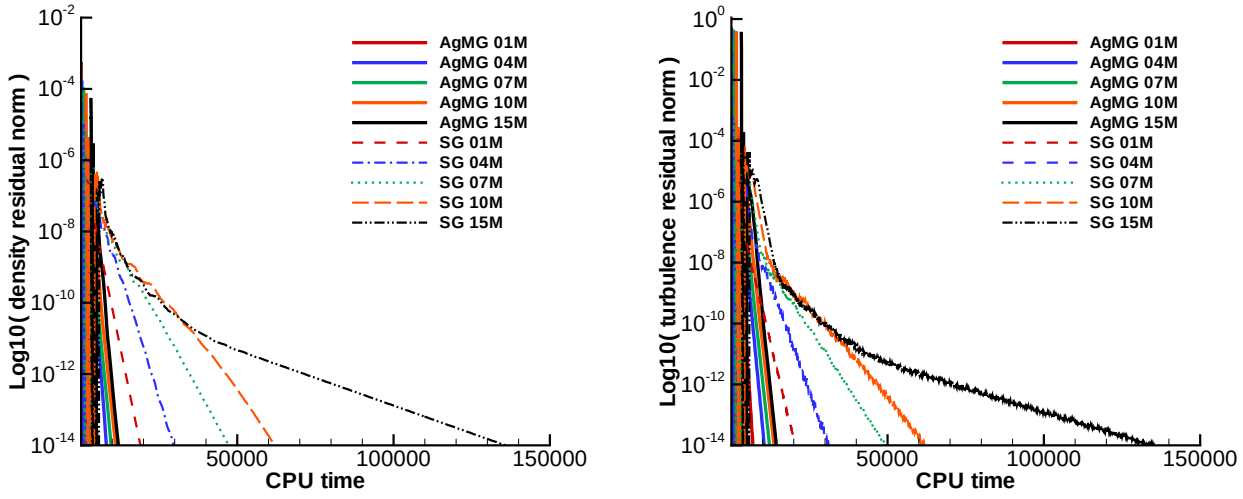



(c) Turbulent flow: Density residual convergence.




(d) Turbulent flow: Turbulence residual convergence.

**Figure 30. 3D hemisphere cylinder: residual convergence**




(a) Density residual convergence.

(b) Turbulence residual convergence.

**Figure 31. 3D hemisphere cylinder RANS: CPU time comparison for 1, 4, 7, 10, and 15 million-node grids. AgMG is almost grid-independent in CPU time with the number of processors chosen to keep 60,000 nodes per processor.**

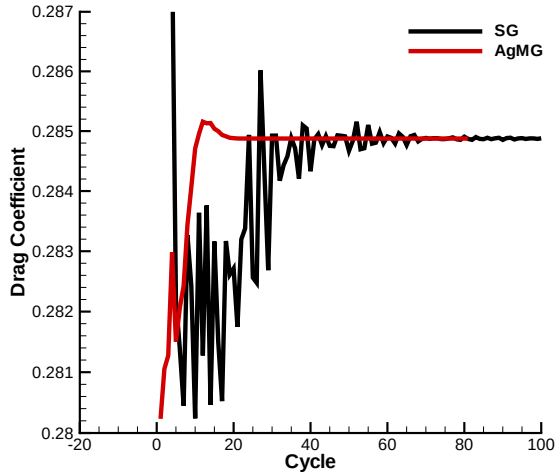



**Figure 32. 3D hemisphere cylinder RANS: Drag coefficient**

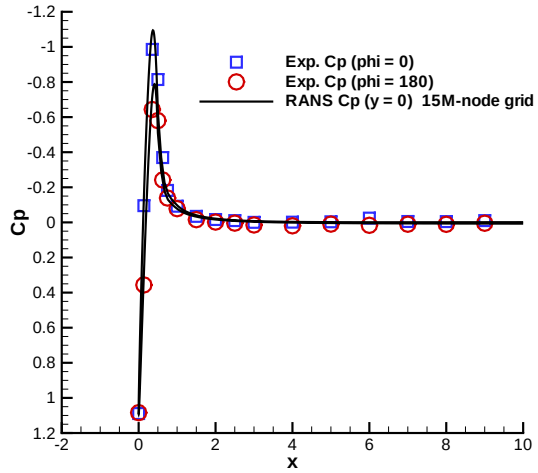
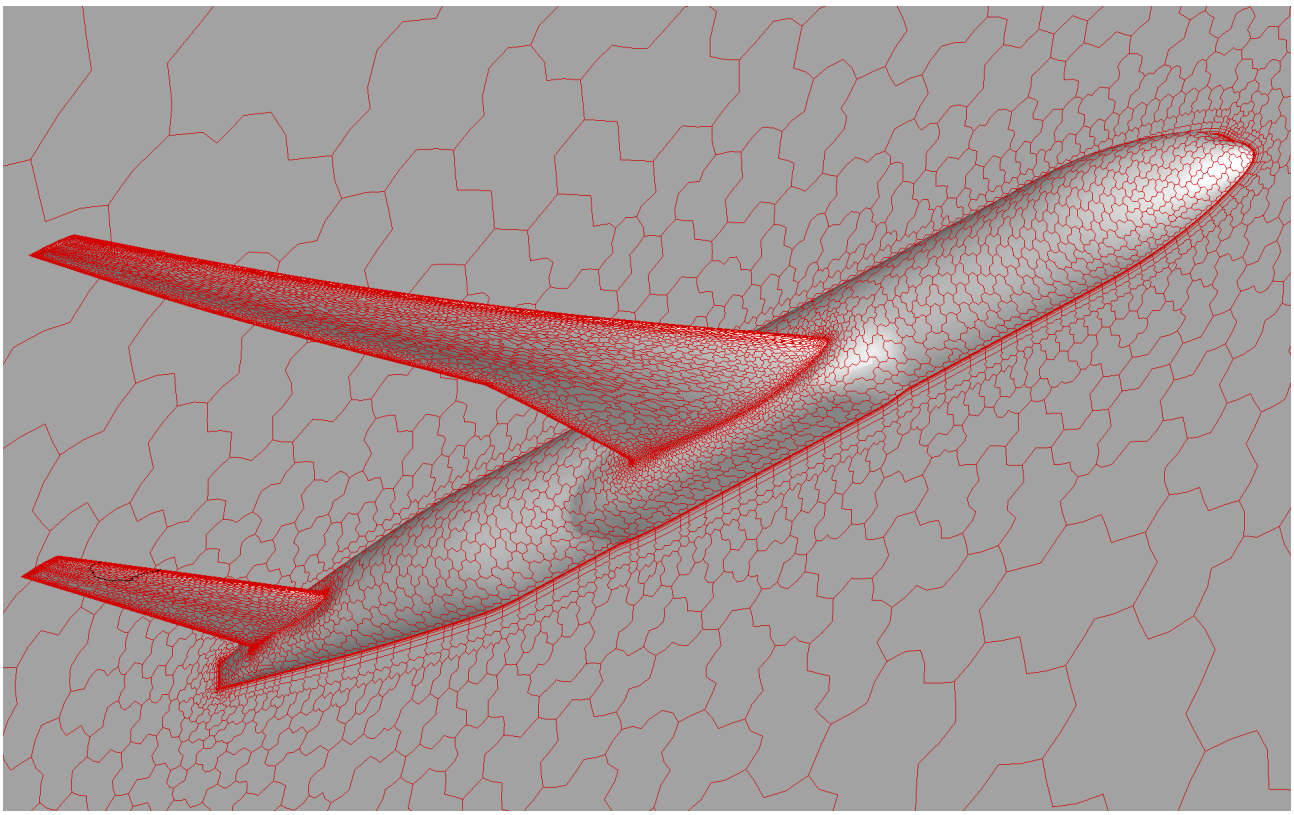
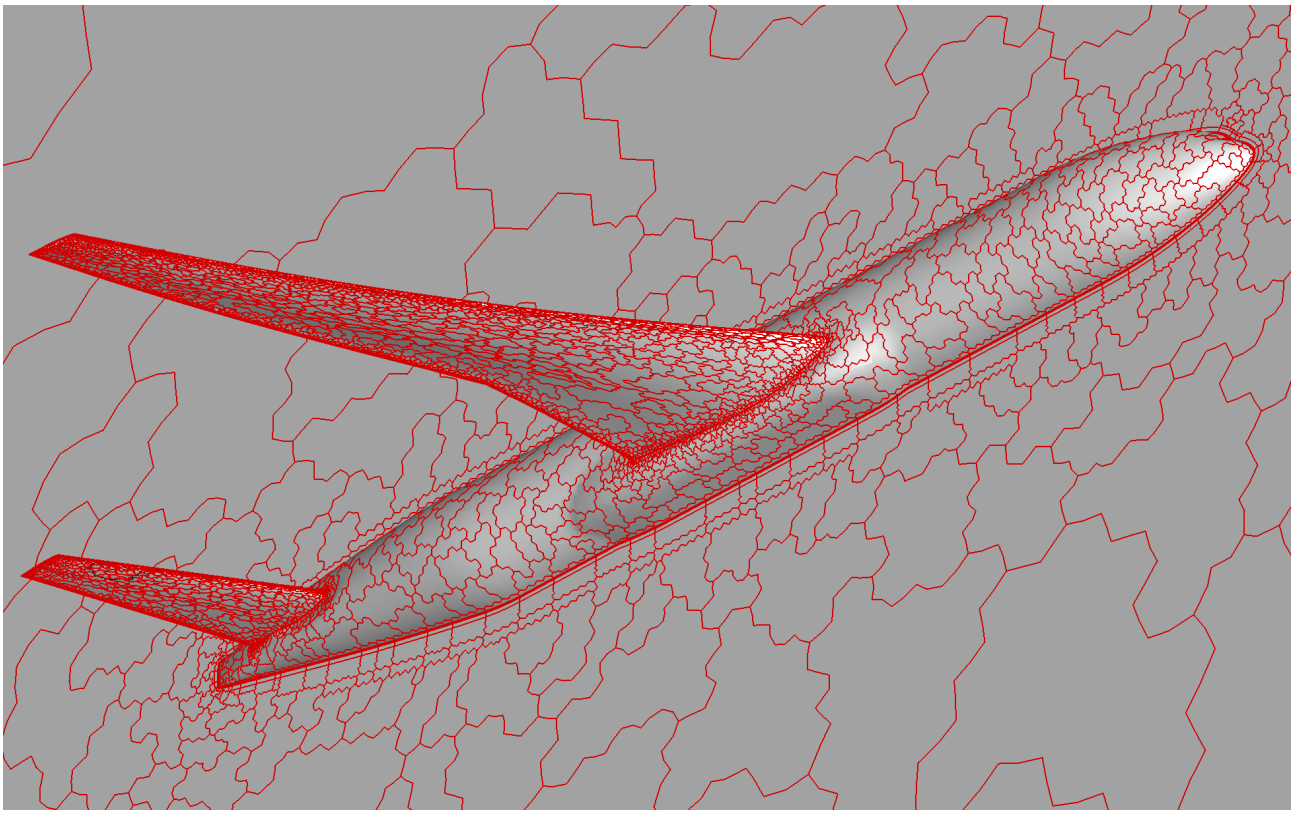



**Figure 33. Pressure coefficient comparison between RANS simulation and experiment.**

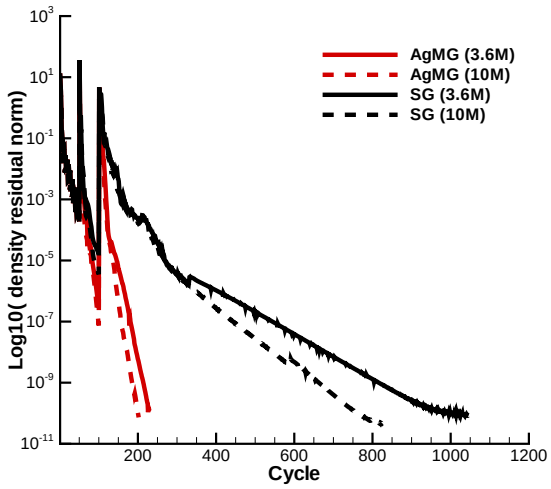


(a) First agglomerated grid.

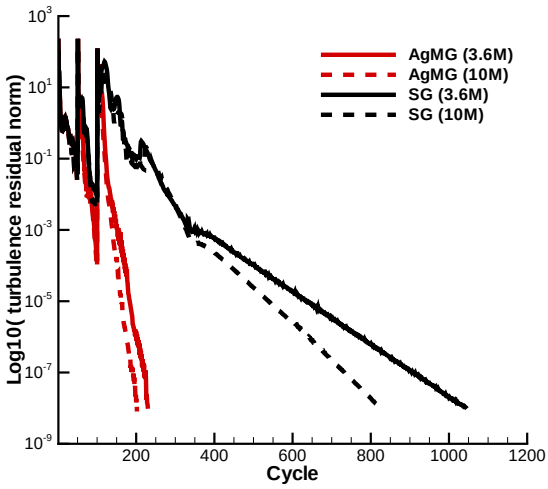


(b) Second agglomerated grid.

**Figure 34. Agglomerated grids for the DPW4 coarse grid.**




(a) DPW4: Density residual convergence.




(b) DPW4: Turbulence residual convergence.

**Figure 35. DPW4: Residual convergence.**

## Appendix A: Effect of Increasing Maximum CFL with StMG Solver

Improved convergence of the StMG solver can be obtained by increasing the maximum CFL beyond the nominal maximum of 200 used for the results in the main body of the paper. The effectiveness of higher CFL specification is case dependent. Generally, on coarse grids and during the initial transient stages of convergence in three dimensions, effective CFL specifications are $O(100)$ for the mean-flow and $O(1000)$ for the turbulence equation. Loosely-coupled or tightly-coupled relaxation of the mean-flow and turbulence equations has not been a discriminating factor in enabling faster convergence through higher CFL specification.

Using a maximum CFL of 10,000, the cycles to converge within an FMG StMG solver are collected in tables A.1 and A.2 for the RAE and NACA 0012 airfoils, respectively. As per the convention used in this paper, the SG cycles that are shown correspond to the same FMG method except that during the FAS cycle, only two levels are used and no coarse grid correction is taken. For the RAE airfoil, convergence is reached within 30 cycles of V(2,2) StMG solver for the 3 finer meshes for either loosely-coupled or tightly-coupled relaxations. For the NACA 0012 airfoil, convergence is also reached within 30 cycles of V(2,2) StMG solver for each of the 3 meshes for the nonzero angles of attack. At zero angle of attack, more than four times more cycles are required. For all of the tests in these two tables, the convergence is uniformly better than the convergence corresponding to a maximum CFL of 200 shown in table 3 in the main body of the paper.

In the tightly-coupled approach, the GCR is also tightly coupled. The GCR minimization is over all residuals and because the turbulence residuals are several orders of magnitude higher than the mean-flow residuals, the minimization focuses on the turbulence residuals. A scaling factor for the turbulence residuals within the GCR approach similar to that used by Lucas et al.[17] was introduced but generally had little influence on the results. The effect of scaling within the GCR when using tightly-coupled relaxation is shown in table A.2 and indicates no positive benefit.

Table A.2 also shows convergence using the second-order order scheme only on Grid 1. The discretization of inviscid terms on Grids 2 through 5 is first-order accurate, and the corresponding FAS convergence on those grids is improved, as expected. FAS convergence on Grid 1 is reduced somewhat. For lower CFL, the convergence of the StMG solver with first-order coarse grid discretization is expected to be more comparable with the convergence of the AgMG solver shown in table 3 in the main body of the paper.

**Table A.1. Number of V(2,2) cycles to converge within an FMG 5-level StMG cycle for RAE airfoil; stopping tolerance $10^{-13}$; maximum CFL 10,000; 30 processors.**

| Grid | Nodes | SG Cycles | FAS Cycles | Relaxation Coupling |
|---|---|---|---|---|
| 1 | 98,944 | | 27 | Loose |
| 2 | 24,896 | | 27 | |
| 3 | 6,304 | | 30 | |
| 4 | 1,616 | | 39 | |
| 1 | 98,944 | 34 | 29 | Tight |
| 2 | 24,896 | 26 | 29 | |
| 3 | 6,304 | 25 | 29 | |
| 4 | 1,616 | 44 | 44 | |

Table A.3 shows the improvement in cycles to convergence with the FAS solver relative to the single-grid solver for the finest grid results reported above with tightly-coupled relaxation. Cycles to convergence were measured within an FMG method as above. The improvement in convergence is slight for the RAE airfoil, perhaps because that grid is coarser by a factor of ten than the finest NACA 0012 grid. The improvement is more pronounced for the NACA 0012, varying from 2 at $\alpha = 10°$ to 9 at $\alpha = 0°$.

The results of multigrid computations over an angle of attack sweep for the NACA 0012 are shown in figures A.1 and A.2 for an increment between angles of attack of 1° The angle of attack sweep begins at $\alpha = 10°$, increases to the first angle of attack beyond maximum lift ($\alpha = 19°$), decreases to $\alpha = -5°$ and then increases back to $\alpha = -3°$. The stopping tolerance is taken as $10^{-5}$ which is sufficient to converge forces to well below plotting accuracy. Overall, the computed lift is slightly higher at any positive angle of attck than the experimental data of Ladson taken from the Turbulence Modeling Resource website.[29] The maximum lift of the computation occurs at $\alpha = 18°$ and is greater than the maximum lift of the experiment at $\alpha = 17°$. The

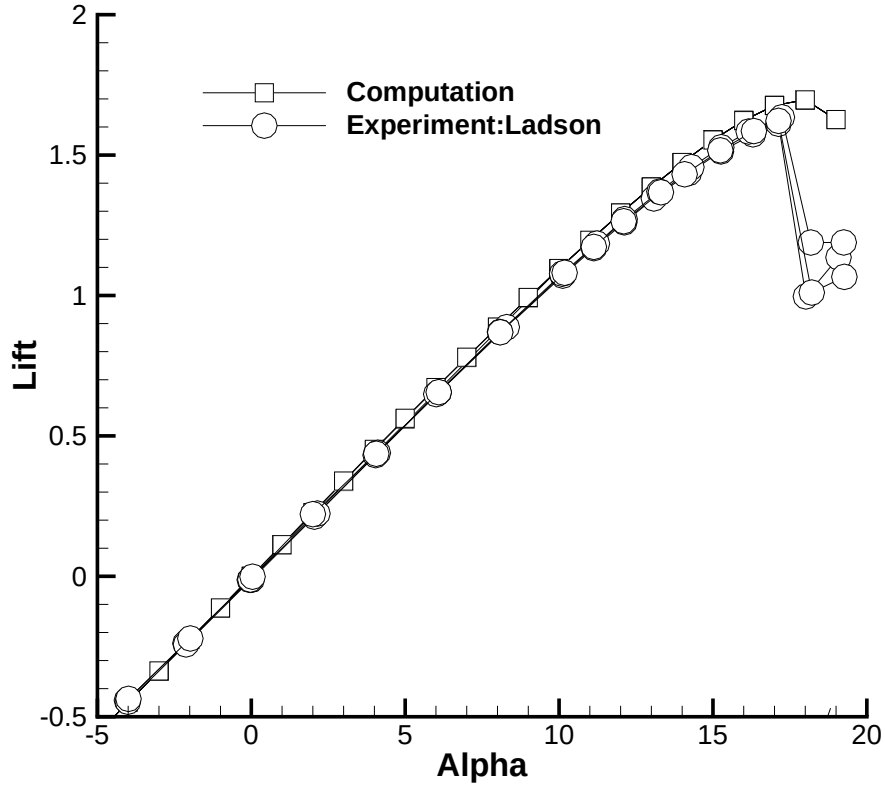



**Figure A.1. Comparison of lift versus angle of attack for NACA 0012 airfoil; computation is on finest grid (Grid 1).**

computed lift at $\alpha = 18°$ is the same whether the starting condition is a lower lift condition or a higher lift condition and, thus, there is no evidence of lift hysteresis. The cycles to converge show an increase near the maximum lift and a more pronounced increase near the zero lift condition. The cycles to converge at a given angle of attack vary slightly depending on the initial condition. Also, for reference, the SG cycles to converge are shown; convergence is accelerated by an order of magnitude with multigrid across most of the angle of attack range.

## Appendix B: DPW5 Wing-Body Convergence Using StMG Solver

Results are shown below for the wing-body configuration of the Drag Prediction Workshop 5 (DPW5)[36] using only the SG and StMG solvers. Families of structured grids are available and pre-processed into hybrid grids; prismatic grids extend from the surface outward a fixed percentage of the points in the normal direction and tetrahedra fill the remainder of the domain. The conditions are taken as those of the workshop (Mach number 0.85, $Re = 5.0 \times 10^6$/(unit grid length)) and $\alpha$ is varied. Advantage is taken of the advancing-layer nature of the grid in that the inviscid reconstructions along the edges outward from the surface are taken as those corresponding to a structured grid. The implicit lines extend from the surface to the far field outer boundary.

Results are shown for the three coarsest workshop grids denoted as L1, L2, and L3, corresponding to 0.7, 2.2, and 5.2 million grid points. A solution at $\alpha = 1°$ was obtained from freesteam initial conditions and then subsequent solutions were obtained by increasing $\alpha$ with an increment of 0.5°. The lift and StMG cycles to converge are shown in figure B.1. The stopping tolerance is taken as $10^{-5}$ which is sufficient to converge forces to well below plotting accuracy. The lift increases with grid refinement but the number of cycles to converge remains fairly constant. The number of V(2,2) cycles to converge at an intermediate angle of attack is shown in Table B.1 for both the SG and StMG solvers. The improvement with the StMG solver over the SG solver

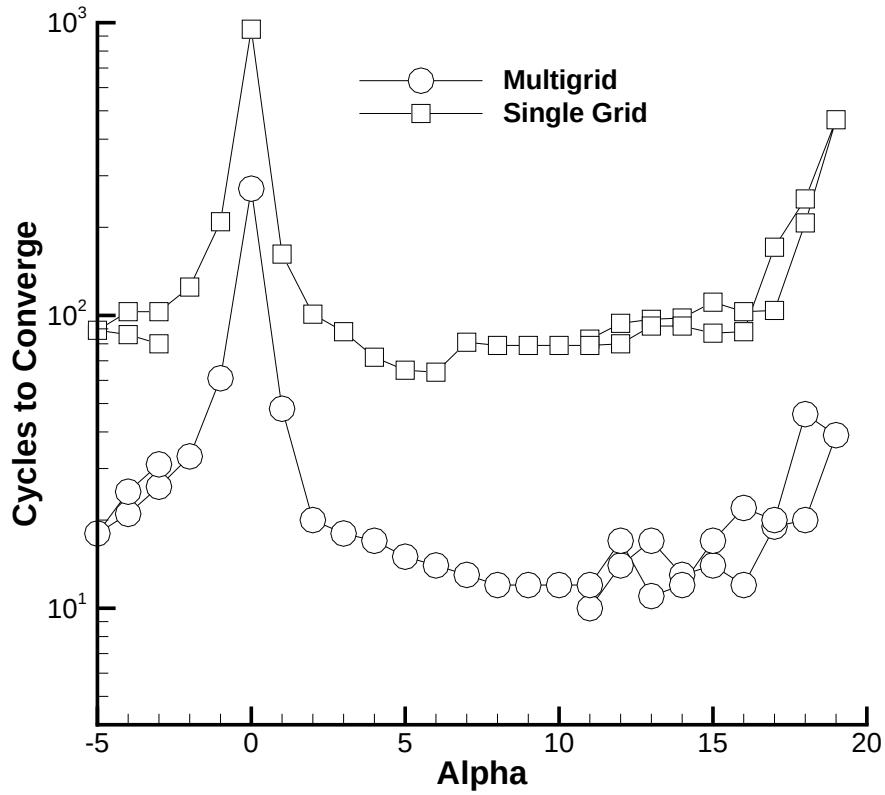



**Figure A.2. Cycles to converge for NACA 0012 airfoil (Grid 1) over an angle of attack sweep starting at $\alpha = 10°$; stopping tolerance $10^{-5}$.**

increases as the grid is refined.

Beyond $\alpha = 3°$, the lift reaches a maximum before $\alpha = 4°$ as the grid is refined. The reduction in lift before $\alpha = 4°$ is not in agreement with the experimental results and is associated with a massive inboard separation pattern, as noted by others.[37] As an example, the computed lift is shown in figure B.2 for the L1-L3 grids over an angle of attack sweep beginning at $\alpha = 3°$, continuing to $\alpha = 4°$, and then decreasing. The L1 computation shows no sign of either a lift decrease or a hysteresis in angle of atttack. The L2 computation exhibits a maximum lift at $\alpha = 2.75°$ mesh and a large hysteresis as the angle of attack is reduced from the maximum lift condition. The same effect is seen with computations on the L3 grid.

Part of the angle of attack continuation process using SG cycles is shown in figure B.3 for the L3 grid. The lift and residuals for the mean flow and turbulence equations are shown for the angle of attack increasing from 3.0 to 4.0 and then decreasing to 3.25; the solution at $\alpha = 3.0°$ was obtained starting from free stream initial conditions. Each solution was converged to the same residual tolerance before changing angle of attack. The hysteresis is indicated by the difference in lift at $\alpha = 3.25°$. The number of cycles is noticeable larger for several of the angles of attack, namely $\alpha = 3.5°$ and $\alpha = 3.75°$, corresponding to the first angle of attack beyond maximum lift and the first decreasing angle of attack.

**Table A.2. Number of V(2,2) cycles to converge within an FMG 5-level StMG cycle for NACA 0012 airfoil; stopping tolerance $10^{-10}$; maximum CFL 10,000; 40 processors; tightly-coupled relaxation of mean-flow and turbulence.**

| $\alpha$ | Grid | Nodes | SG Cycles | FAS Cycles | FAS Cycles (GCR scaling) | FAS Cycles ($2^{nd}$ order inviscid on only Grid 1) |
|---|---|---|---|---|---|---|
| 0° | 1 | 919,424 | 1007 | 118 | 139 | 139 |
| | 2 | 230,336 | 534 | 109 | 138 | 113 |
| | 3 | 57,824 | 295 | 95 | 109 | 92 |
| | 4 | 14,576 | 190 | 82 | 86 | 78 |
| | 5 | 3,704 | 144 | 144 | 186 | 142 |
| 10° | 1 | 919,424 | 52 | 24 | 27 | 36 |
| | 2 | 230,336 | 55 | 20 | 49 | 20 |
| | 3 | 57,824 | 44 | 21 | 34 | 18 |
| | 4 | 14,576 | 36 | 21 | 47 | 13 |
| | 5 | 3,704 | 54 | 54 | $> 300$ | 42 |
| 15° | 1 | 919,424 | 89 | 24 | 38 | 46 |
| | 2 | 230,336 | 80 | 23 | 35 | 23 |
| | 3 | 57,824 | 52 | 24 | 44 | 18 |
| | 4 | 14,576 | 83 | 38 | 102 | 14 |
| | 5 | 3,704 | 80 | 80 | $>300$ | 42 |

**Table A.3. Improvement in cycles to converge for finest airfoil grids; maximum CFL 10,000; tight coupling.**

| Case | StMG Improvement |
|---|---|
| RAE | 1.2 |
| NACA0012 $\alpha = 0°$ | 8.5 |
| NACA0012 $\alpha = 10°$ | 2.2 |
| NACA0012 $\alpha = 15°$ | 3.7 |

**Table B.1. Number of V(2,2) cycles to converge within an FAS 4-level StMG cycle for DPW5 Wing-Body configuration grids; maximum CFL 10,000; tight coupling; stopping tolerance $10^{-5}$; $\alpha = 1.5°$.**

| Grid | SG Cyles | FAS Cycles | ( Sg Cycles) / ( FAS Cycles) |
|---|---|---|---|
| L1 | 72 | 52 | 1.4 |
| L2 | 93 | 40 | 2.3 |
| L3 | 218 | 43 | 5.1 |

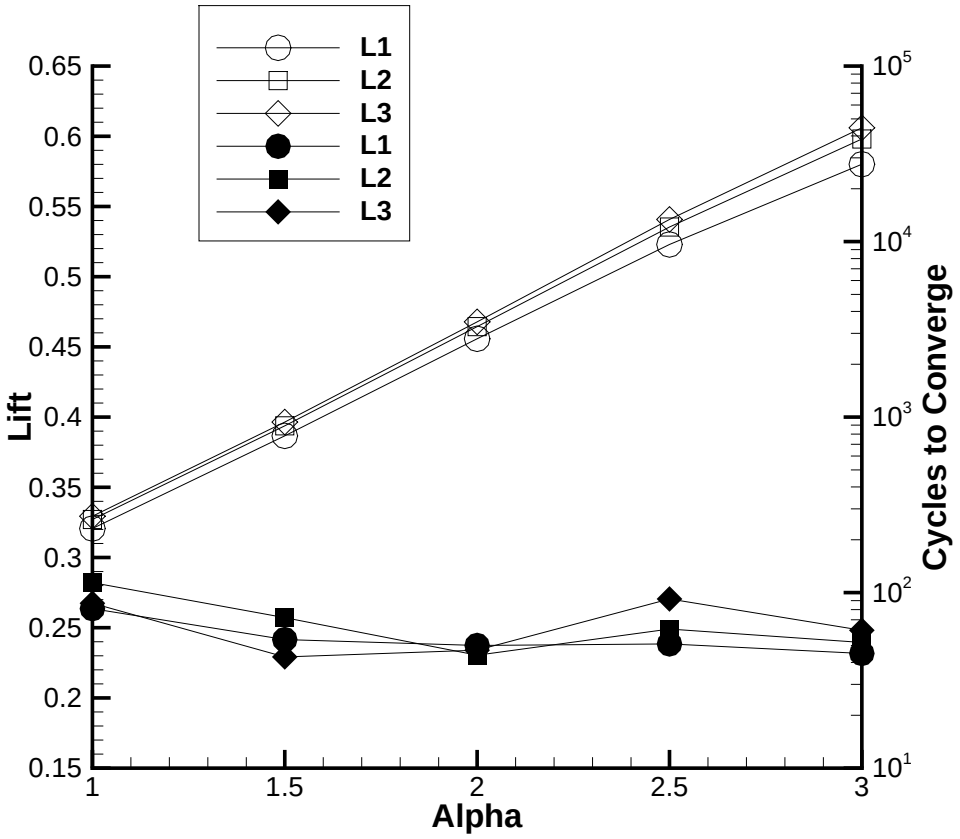



**Figure B.1. Lift (open symbols) and StMG cycles (solid symbols) to converge for 3 grids of the DPW5 workshop; stopping tolerance $10^{-5}$.**

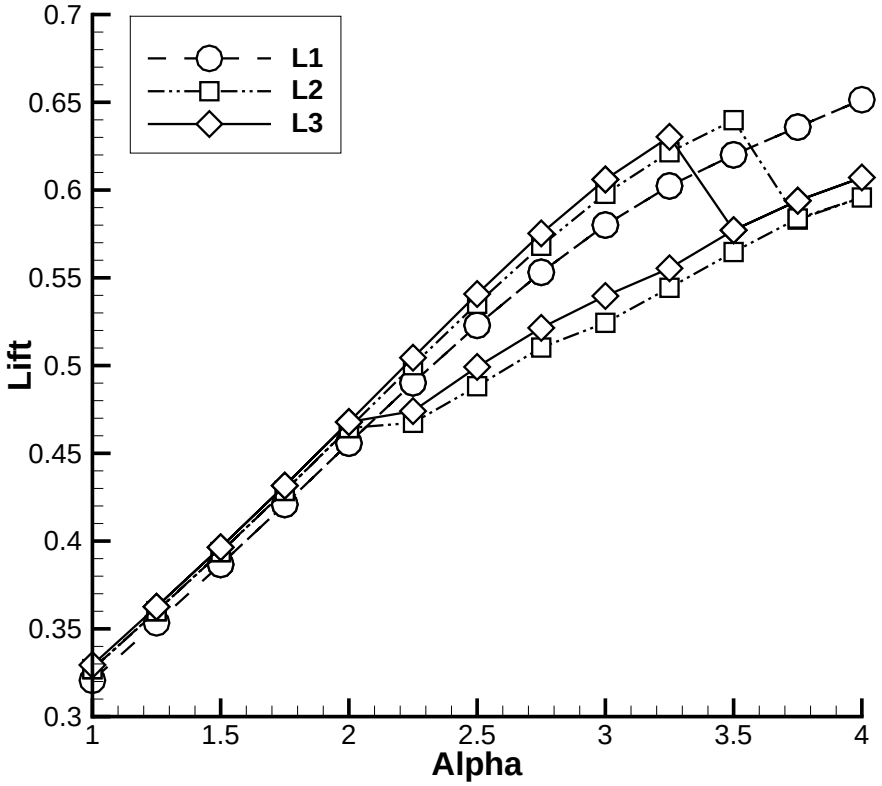



**Figure B.2. Lift over an angle of attack sweep with the L1-L3 grids of the DPW5 workshop; stopping tolerance $10^{-5}$.**

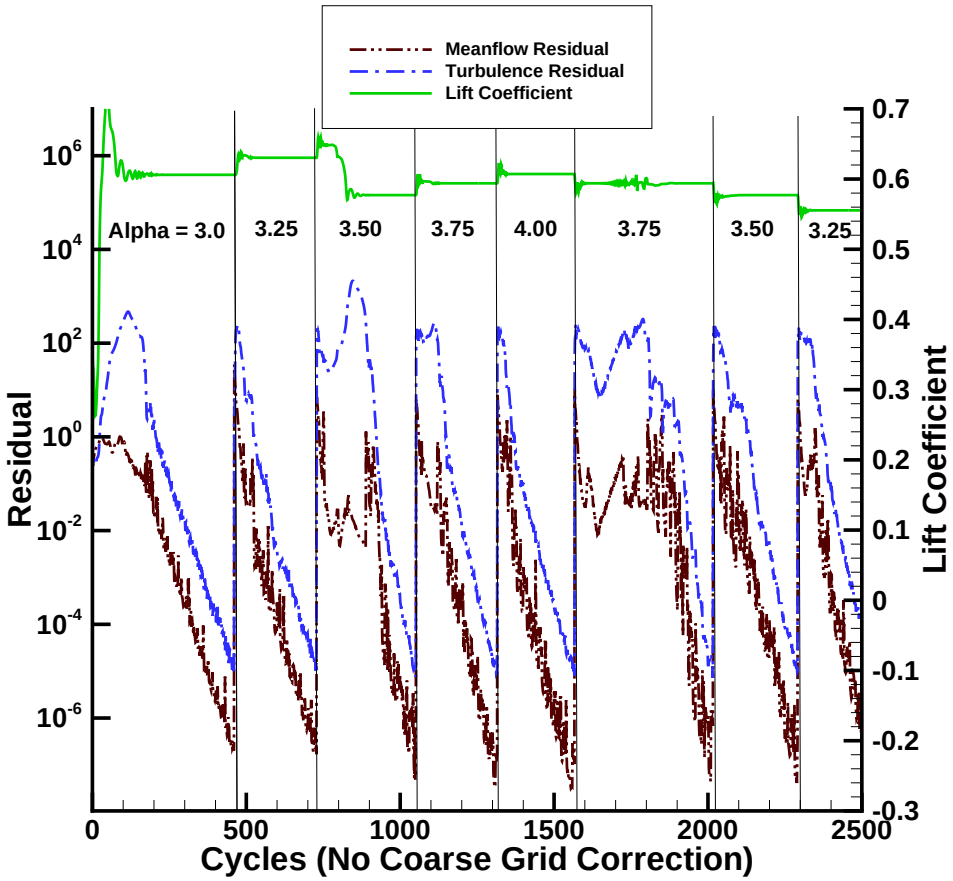



**Figure B.3. Residuals and lift for angle of attack continuation with L3 grid.**