20th AIAA Computational Fluid Dynamics Conference
27 - 30 June 2011, Honolulu, Hawaii

AIAA 2011-3693

Cartesian Off-Body Grid Adaption for Viscous Time-Accurate Flow Simulation

Pieter G. Buning*
NASA Langley Research Center, Hampton, Virginia, 23681

and

Thomas H. Pulliam†
NASA Ames Research Center, Moffett Field, California 94035

An improved solution adaption capability has been implemented in the OVERFLOW overset grid CFD code. Building on the Cartesian off-body approach inherent in OVERFLOW and the original adaptive refinement method developed by Meakin, the new scheme provides for automated creation of multiple levels of finer Cartesian grids. Refinement can be based on the undivided second-difference of the flow solution variables, or on a specific flow quantity such as vorticity. Coupled with load-balancing and an in-memory solution interpolation procedure, the adaption process provides very good performance for time-accurate simulations on parallel compute platforms. A method of using refined, thin body-fitted grids combined with adaption in the off-body grids is presented, which maximizes the part of the domain subject to adaption. Two- and three-dimensional examples are used to illustrate the effectiveness and performance of the adaption scheme.

I. Introduction

SOLUTION adaption, or adaptive mesh refinement (AMR) is a powerful tool for improving the accuracy of computational fluid dynamics (CFD) simulations.[1] AMR has been developed and applied to structured and unstructured grid methods, Cartesian mesh solvers, and mixed-mesh software frameworks (c.f. Refs. 2-4). There are several challenges to implementing a robust AMR capability for complex unsteady problems, however. These include: (1) finding a sensor function which is appropriate for viscous flows with a range of flow features; (2) parallelization of the adaption process while maintaining a load-balanced grid system; and (3) efficiency suitable for adaption every few steps. While the scheme presented here only covers the off-body part of the computational domain, it does address each of these challenges.

The proposed adaption scheme is developed as an integral part of the OVERFLOW CFD code,[5] a structured, overset grid framework where "near-body" grids are body-fitted curvilinear grids, and "off-body" grids are Cartesian grids which fill the computational space between the near-body grids and the far-field boundary. The off-body grids have finer and coarser levels, each level differing by a factor of two in spatial resolution in all three coordinate directions. This terminology and framework was established by Meakin.[6] The simulation procedure is parallelized using the Message Passing Interface (MPI), with each MPI process responsible for a collection of grids. Processes are load-balanced by subdividing original near-body and off-body grids until each process can be assigned a collection of grid partitions which provides a balanced workload. Each MPI process can be further parallelized using OpenMP, a shared-memory paradigm. For most flow solver tasks, OpenMP parallelization is accomplished by processing planes of a computational grid in parallel.

This scheme builds on the work of Meakin,[6] specifically the Cartesian off-body grid approach and mechanics, and the grid adaption process and sensor function. New aspects include the ability to create refinement levels finer than the original grid system and the solution interpolation procedure, while implementation and scaling of the sensor function have been improved. The scheme also follows closely the capabilities of the Helios software

* Aerospace Engineer, Computational AeroSciences Branch, MS 128, AIAA Associate Fellow.
† Aerospace Engineer, Fundamental Modeling and Simulation Branch, MS 258-2, AIAA Associate Fellow.

framework for rotorcraft flows,[4] though here we emphasize that the adaption capability is an integral part of the flow solver, and that it is applicable to a wide variety of flows. There are also a number of similarities to the parallel AMR scheme for reacting flows on overset grids presented by Henshaw and Schwendeman.[7] The scheme of Henshaw and Schwendeman employs adaption in both the near-body and off-body grids, and is applied to inviscid, high-speed flows.

II. Sensor Function and Grid Marking

Calculation of a sensor function for grid adaption is performed in a modular fashion, so that adaption can be performed using one of several different functions. These could include flow quantities, solution error estimates, or adjoint variables. Adaption to any particular flow quantity runs the risk of being appropriate for only certain classes of flows. Adjoint-driven grid adaption is an extremely powerful approach, but requires a significant infrastructure to compute the adjoint flow solution. Here we focus on a relatively simple error estimation quantity derived from the flow solution, the undivided second-difference of the primary flow variables (density, momentum, and stagnation energy per unit volume); we will refer to these flow variables collectively as Q, and individual elements as q. Note that this function has been studied by Warren et al.,[2] and was used for the adaptive grid approach of Meakin[6] (but with some scaling difficulties). It is (almost) a subset of the function proposed by Henshaw and Schwendeman,[8] who use a weighted combination of undivided first- and second-differences, summed over coordinate directions. The term "undivided" refers to these differences not being divided by the appropriate spatial difference term, Δx for the first-difference, Δx^2 for the second-difference.

The undivided second-difference is represented at each point as $S_i = q_{i-1} - 2q_i + q_{i+1}$. It can be thought of in several ways, each providing some insight into the usefulness of the function. Per the name, it represents the second-difference of q times Δx^2. It is also (proportional to) the difference between q_i and the average of its neighbors q_{i+1} and q_{i-1}, or the difference between q as represented on a grid and the linear interpolation of q from neighboring points on a 2x coarser grid.

In the context of computing S as a general function for adaption, we (a) normalize by a reference quantity q_{ref}, (b) square it to create a non-negative value, (c) take the maximum over all components of Q, and (d) take the maximum over all coordinate directions:

$$S = \max_{i=j,k,l} \left\{ \max_{Q \text{ components}} \left[\left(\frac{q_{i-1} - 2q_i + q_{i+1}}{2q_{\text{ref}}} \right)^2 \right] \right\}$$

This function is non-dimensional, independent of grid units, and becomes smaller as the grid is refined (where Q is smooth), all desirable properties for a sensor function for solution adaption. Further, it is simple to compute and the computation can be fully parallelized. An example of this function is shown in Fig. 1(a) for supersonic flow about an airfoil. Here it is plotted on an adapted grid system, and contouring is shown on a logarithmic scale.

In contrast to the undivided second-difference (at least in smooth regions of the flow), we note that a flow quantity sensor function such as vorticity will not get smaller as the grid is refined. As such, a grid adapting to this sensor will continue to be refined until limited by the number of refinement levels allowed. In some cases this can be desirable (such as for the two-dimensional vortex convection problem below), and give the finest level of refinement to any feature with function value above the refinement threshold.

The sensor function is used to mark the current grid for refinement or coarsening as follows. If the function value at a grid point exceeds a specified threshold for refinement, that point is marked for refinement; if the value is below a coarsening threshold, the point is marked for coarsening. The marker function for the airfoil case is shown in Fig. 1(b).

Within a cube or "box" of points (e.g., 8 x 8 x 8 points), if any point is marked for refinement, the box is flagged for refinement. Alternatively, if all points are marked for coarsening, the box is flagged for coarsening. If the box is not flagged for refinement or coarsening, it is left at the current refinement level. In the current implementation, grid regions can only coarsen or refine by one grid level at a time. These coarsening and refinement boxes are then used in the off-body Cartesian grid generation process to create a new adapted grid system.

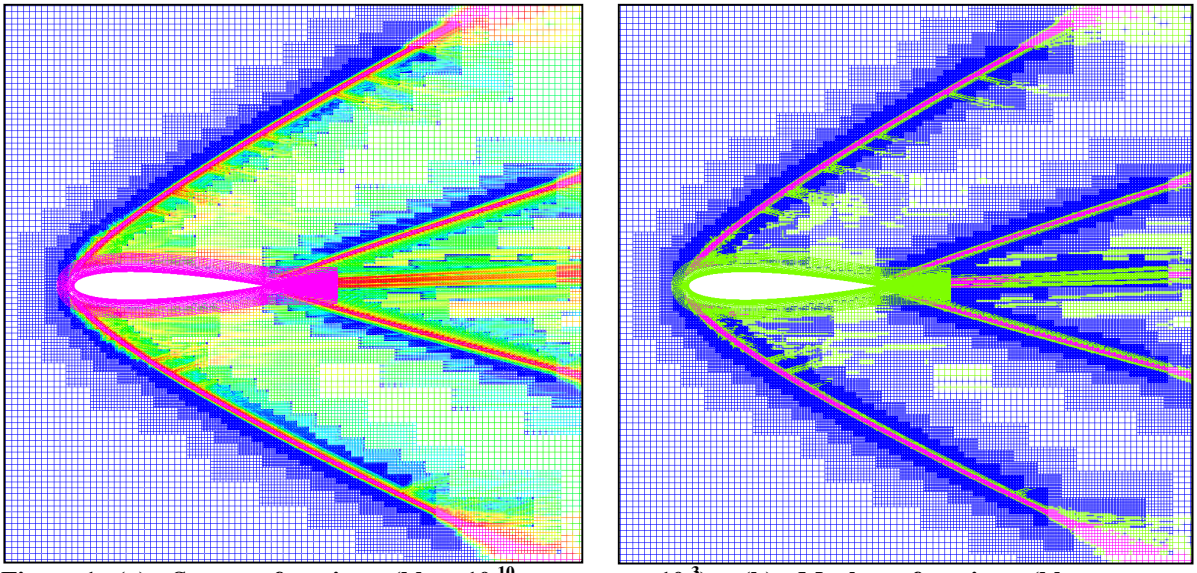

Figure 1. (a) Sensor function (blue=10^{-10}, magenta=10^{-3}); (b) Marker function (blue=coarsen, green=maintain, magenta=refine).

III. Grid Generation and Connectivity

As mentioned before, the purpose of the off-body grid system is to cover the computational domain between the near-body grids and the far-field boundary using Cartesian grids of different refinement levels. There are reasons to choose refinement levels in different regions: to maintain consistency with the near-body grid spacing, and to coarsen as we approach the far-field boundary. With solution adaption, there are additional requirements on grid spacing from the adaption boxes generated from the sensor function and grid marking process. We refer to the original (non-adapted) finest off-body grids as "level 1" grids, and coarser grid levels as level 2, 3, etc. The level 1 grid spacing is provided by the user, and level 1 grids automatically surround all near-body grids. Level 2 grids surround level 1 grids, etc., out to the far-field boundary.

Finer grids generated by the adaption process are referred to as level -1, -2, etc., each finer than the previous level by a factor of two. A maximum number of refinement levels is supplied by the user as part of the grid generation process. The number of refinement levels can additionally be limited inside or outside of (x,y,z) regions of the computational domain, for example to limit adaption away from areas of interest, or to allow more refinement near one component than another. This is done by modifying the adaption boxes generated from the sensor function and grid marking process. Note that solution adaption can be performed using no additional refinement levels, in which case level 1 grids will track features in the flow.

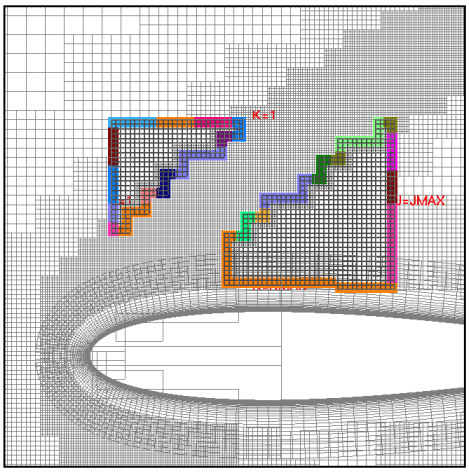

Figure 2. Sample refinement grid showing blanked-out region and Chimera boundary points.

In this implementation, refined grid patches (level -1 and finer) overlay level 1 grids, blanking out regions of the next-coarser grid level. This allows each grid level to only communicate with neighboring levels, and/or near-body grids. Blanking and communication between levels is done through the Chimera overset grid process.[9] Specifically, we use the X-ray method of Meakin for hole-cutting,[10] coupled with the Domain Connectivity Function (DCF) for establishing communication between overset grids.[11,12] Each X-ray provides a digitized map of the surface geometry, and is used to cut holes in other grids. The spacing of the rays determines how accurately the geometry is represented, and depends on how far the near-body grids extend from the geometry surface and the grid spacing of the neighboring grids. As in previous versions of OVERFLOW, all near-body grids are surrounded by level 1 off-body grids, and X-rays are used to cut holes in the level 1 grids. With off-body solution adaption, X-rays

now cut holes in refinement grids as well. An example of this type of grid system is shown in Fig. 2, again for the supersonic airfoil case. One grid is highlighted, showing a jagged blanked-out region through the center where finer grids overlap this grid. Chimera boundary points for this grid are also highlighted, showing outer boundaries and hole boundaries.

IV. Solution Interpolation

Once a new off-body grid system has been generated, the flow solution has to be interpolated from the old grids to the new ones. This process is done (for each new grid) by interpolating flow quantities from any overlapping old off-body grids. Old grids are checked coarse-to-fine, so that the final interpolated solution represents the finest available solution in each part of the grid. Interpolation weightings are easy to determine, since both old and new grids are Cartesian. The current process uses trilinear interpolation.

This procedure is parallelized by structuring the interpolation as an outer loop through old off-body grids and an inner loop through new off-body grids. The new grid system is load-balanced in the same way as the old system, but without any consideration for communication between the two systems. The MPI process that owns the old grid extracts appropriate subsets and sends them to processes that own overlapping new grids, using non-blocking sends. The receiving processes use blocking receives, and interpolate the grid subset immediately. If the owner of the old grid also owns an overlapping new grid, interpolation is done in-place. In this way, much of the processing of each old grid is performed in parallel, while only sending the required subsets of the grid to those processes that need them.

It is noted that while a solution adaption capability existed in the OVERFLOW-D code,[13] this approach stored the old flow solution on disk before interpolating onto the new grid system. This process avoided the need for having the old and new solutions in memory at the same time, at the expense of using disk I/O. This was generally not a limitation when performing steady-state simulations where the solution adaption process was performed a limited number of times. Further, for many problems the ratio of off-body to the total grid size is small, reducing the burden of the (off-body) adaption process. However, we wish to use solution adaption for unsteady and moving-body problems which require adaption every 5-10 time steps. A primary application area is rotorcraft flows, where the accurate representation of blade tip vortices is the first step to providing accurate detailed blade loads and acoustic signature. In this environment, the off-body grid points account for roughly 90% of the total grid size. The use of disk I/O in this situation results in severe performance degradation: for a small rotorcraft problem, 50% of the total wall-clock time was devoted to the adaption process when using disk, while the same case using the parallelized in-memory process spent less than 3% of its time in adaption.

When using Newton subiteration or dual time-stepping for time-accuracy, two time levels of the flow solution are interpolated, allowing the second-order time advance scheme to continue through adaption steps.

V. Demonstration of Spatial Accuracy

As a first step to validating the solution adaption technique, it is desirable to show that a grid system adapted to a flow feature demonstrates a quantifiable improvement in the solution with each finer level of grid. Given that adaption is performed only in the off-body region of the domain, a two-dimensional, inviscid vortex convection problem is chosen, with characteristic far-field boundary conditions applied far from the vortex path. A small Cartesian grid is used to cover the initial vortex position, with extent of ±5 vortex core radii and spacing of 0.1 radii (20 points across the vortex core). This serves as the near-body grid for this case, and off-body grids are generated around this grid, extending out to the far field boundary. As the vortex convects downstream, the off-body grids adapt to the vortex. Off-body level 1 spacing is set to 0.4 (4 times larger than the near-body grid), and runs are made with 0, 1, and 2 levels of refinement. These result in adapted grids with 5, 10, and 20 points across the vortex core, respectively. To guarantee that the adaption scheme uses the finest selected grid spacing for the vortex, vorticity magnitude is chosen as the sensor function. A convection speed of Mach 0.5 is used, with 10 Newton subiterations and a time step based on convection speed of 0.02, 0.01, or 0.005 (corresponding to 0, 1, or 2 refinement levels). These time step values were chosen to give a temporal error that is small compared to the spatial error, and result in 50, 100, or 200 steps per core radius traveled.

As a measure of accuracy we track the minimum pressure in the field, normalized by the free stream pressure. The error is taken as the difference between the minimum pressure and the initial value, and this is plotted vs. the (ideal) distance the vortex has traveled. Our goal here is to show spatial accuracy, so the time step is adjusted until there is no effect on the result beyond the initial startup. Using a 5th-order WENOM flux scheme,[14] Fig. 3 shows pressure contours on the initial grid system, and on the adapted grid system after the vortex has traveled 1000 core radii (using 2 levels of refinement). The minimum pressure error vs. convection distance is plotted in Fig. 4. “Ideal”

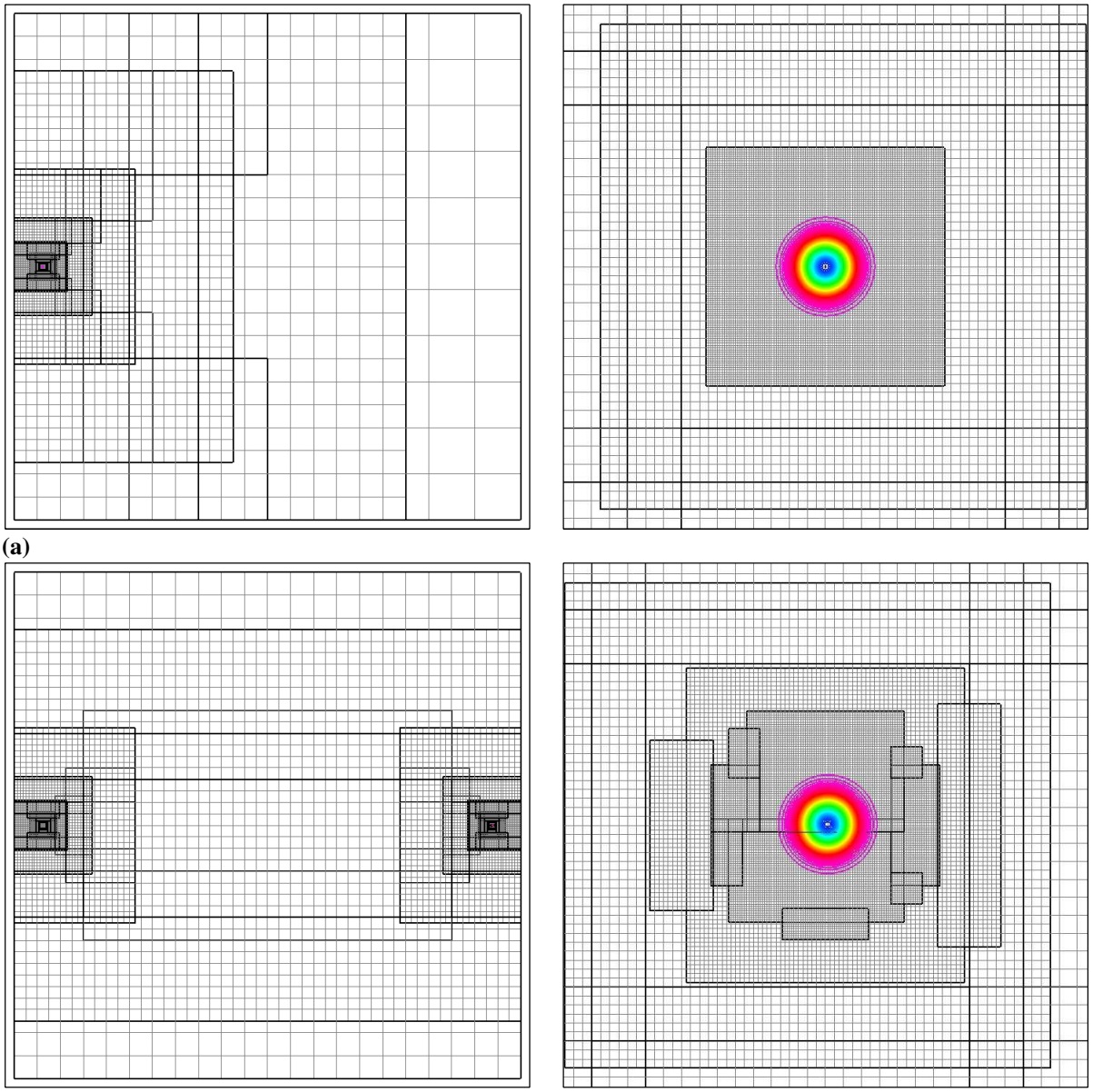

Figure 3. Grid system and pressure contours for the vortex convection problem (full domain (left) and closeup (right)), on (a) the initial grid system, and (b) the adapted system after 1000 core radii travel, using two levels of refinement.

curves for 0 and 2 refinement levels are created by multiplying and dividing the 1 refinement level errors by 32 (or 2^5). The fact that these curves closely follow the actual error curves for 0 and 2 refinement levels illustrates that 5^{th}-order spatial accuracy is maintained over these three grid levels.

VI. Thin Near-Body Grid Approach

Solution adaption in off-body regions can be useful for resolving flow features outside of the parts of the domain covered by the near-body grids. In this way helicopter rotor tip vortices can be accurately convected downstream, and vortex interactions can be resolved. However, to resolve the effect of these features as they impact back on the helicopter fuselage or rotor, similarly refined near-body grids are needed. While development of a near-body adaption capability for OVERFLOW is underway, another approach is to use high resolution near-body grids. Making these grids as thin as possible in the surface-normal direction avoids wasting near-body grid points where

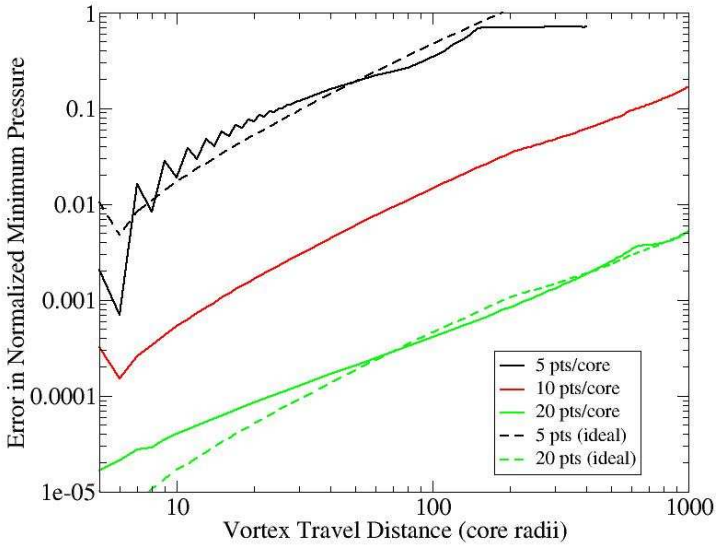

Figure 4. Error in minimum pressure with vortex convection for three levels of grid refinement.

they may not be needed, and allows the off-body grid adaption to be used for more of the flow domain. This can result in better resolution of features such as shock-boundary layer interactions or vortex generation. To illustrate, we use a simulation of flow about a NACA 0012 airfoil at Mach 0.55, 8.34 deg. angle-of-attack, and a Reynolds number of 9 million based on chord. Previous computations and experimental data for this case were published as part of the Viscous Transonic Airfoil Workshop.[15] These flow conditions lead to a small shock near the leading edge of the airfoil. While a moderately-resolved single block grid captures the shock wave and the resulting pressure jump, a fine near-body grid with solution adapted off-body grids results in a sharper shock and captures the lambda shock that forms within the boundary layer. Mach contours on the single-block "standard" grid are compared to those on a thin near-body grid system with adapted off-body grids in Fig. 5. The resulting bulge in the pressure coefficient after the shock is visible in Fig. 6, consistent with experimental data. The single-block grid is 253 x 73, while the thin near-body grid is 485 x 40, and extends only 0.01 chords from the airfoil. Both use an O-grid topology. The adapted off-body grid system uses a level 1 spacing of 0.005 chords, with 3 levels of refinement driven by the undivided second-difference as the sensor function. Final grid size was 545,000 points in a 2D slice of the grid. These calculations used 4th-order central differencing with 2nd- and 4th-difference matrix dissipation for the inviscid fluxes, and 2nd-order central differencing

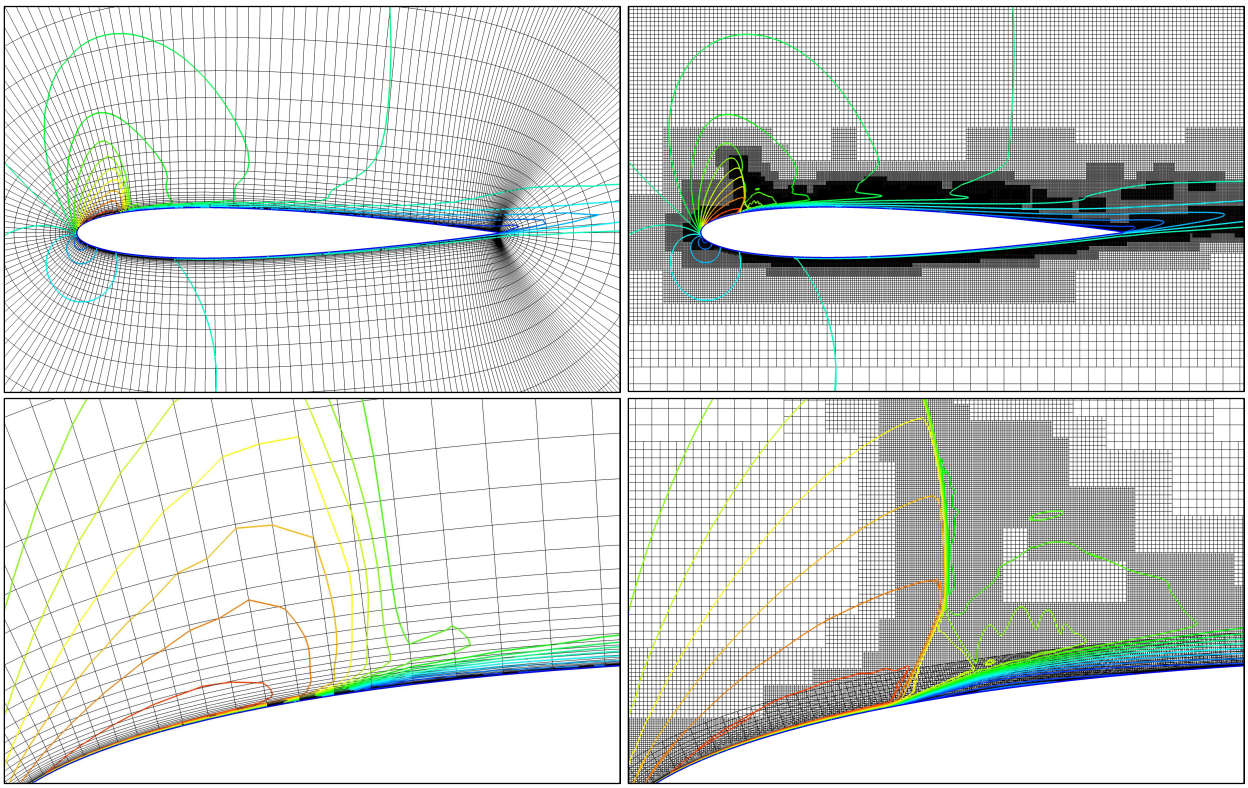

Figure 5. NACA 0012 grid and Mach contours, standard grid (left) and adapted grid (right).

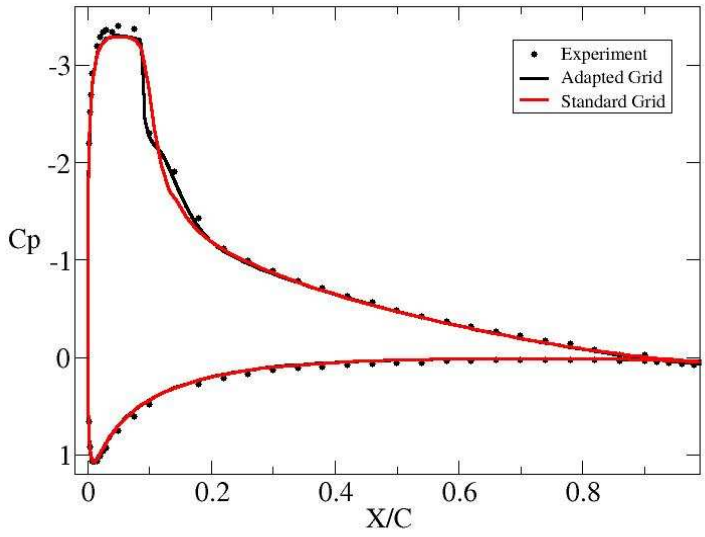

Figure 6. Comparison of pressure coefficient for the standard and adapted grid.

for viscous terms. The Spalart-Allmaras turbulence model was used for these simulations, using an exact wall distance function valid in all grid blocks.

One characteristic of this approach must be treated carefully. With a thin near-body grid, it is difficult to establish good overset grid communication between the near-body and off-body grids. One goal is to have enough overlap between the grids so that both have enough interpolated boundary points (fringe points) to maintain the differencing stencil of the interior flux scheme. For 2nd- or 3rd-order flux differencing, two fringe points are needed; for a 4th- or 5th-order scheme, three points are needed. This must be balanced with the goal of keeping the coarser off-body grid away from the high flow gradients in the boundary layer. In order to accomplish both goals, the off-body grid spacing must be fairly fine, on the order of 1/10th the extent of the near-body grid in the viscous direction. The spacing of the X-rays used to cut holes in the overset grids must also be fine, of the same order as the off-body spacing. Without such fine off-body grid spacing, reduced-order flux difference stencils must be used, resulting in a loss of accuracy in the neighborhood of the near-body/off-body interface. Note that the creation of refined off-body grids from the solution adaption process can improve overset grid communication, at least locally.

A three-dimensional example of this approach is presented in Ref. 16 for the NASA Trap Wing, a configuration used for the AIAA High Lift Prediction Workshop.[17] Use of thin near-body grids greatly simplifies the communication of various grids about the components of the geometry, and off-body solution adaption is shown to improve wake and tip vortex resolution and the correlation of wing tip pressures with experiment. An illustration of the grid in the wake is shown in Fig. 7.

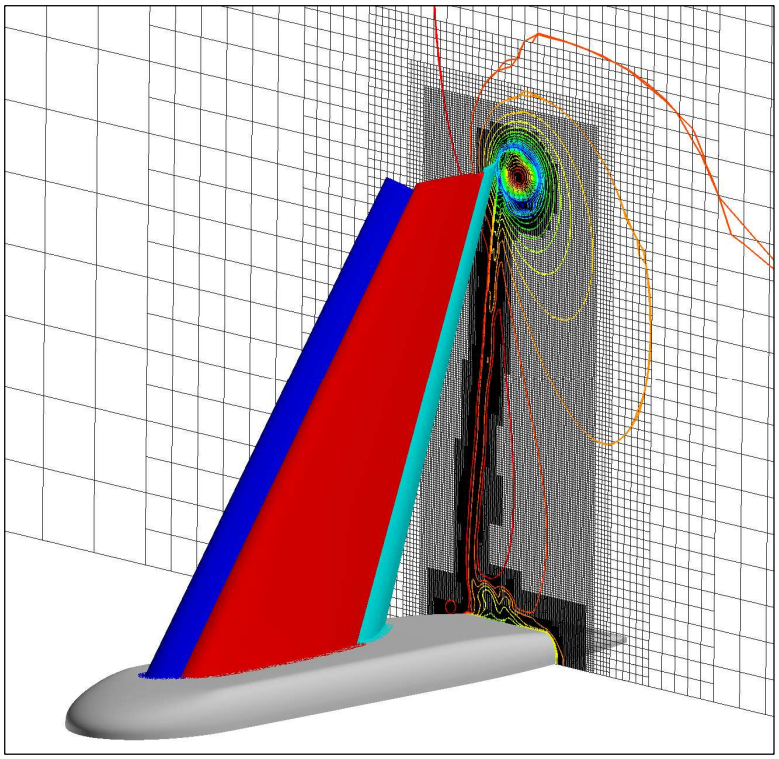

Figure 7. Grid and Mach contours in the wake of the NASA Trap Wing (from Ref. 16).

VII. Rotorcraft Applications and Parallel Performance

In this section we present results from two rotorcraft applications, and highlight the performance of the solution adaptive scheme in a parallel computing environment. These applications demonstrate improved resolution of off-body flow physics, important for predicting the unsteady flow fields associated with rotorcraft. They also illustrate the applicability and performance of the grid adaption scheme to unsteady flows where a large percentage of the grid points are in the off-body part of the domain. We note that these cases do not explicitly use the thin near-body grid approach proposed in the previous section. The first case looks at the effect of simply applying grid adaption to an existing rotorcraft grid system, while the second uses grid adaption to examine detailed flow physics in a hovering rotor wake.

A. UH-60 4-Bladed Rotor

For the first application, we compare a coarse-grid simulation of a 4-bladed rotor using 4.5 million grid points and a simulation using the same near-body grids and two finer levels of grid adaption. The coarse grid system and calculations at similar flow conditions are described in Ref. 18. For adaption we use the undivided second-difference

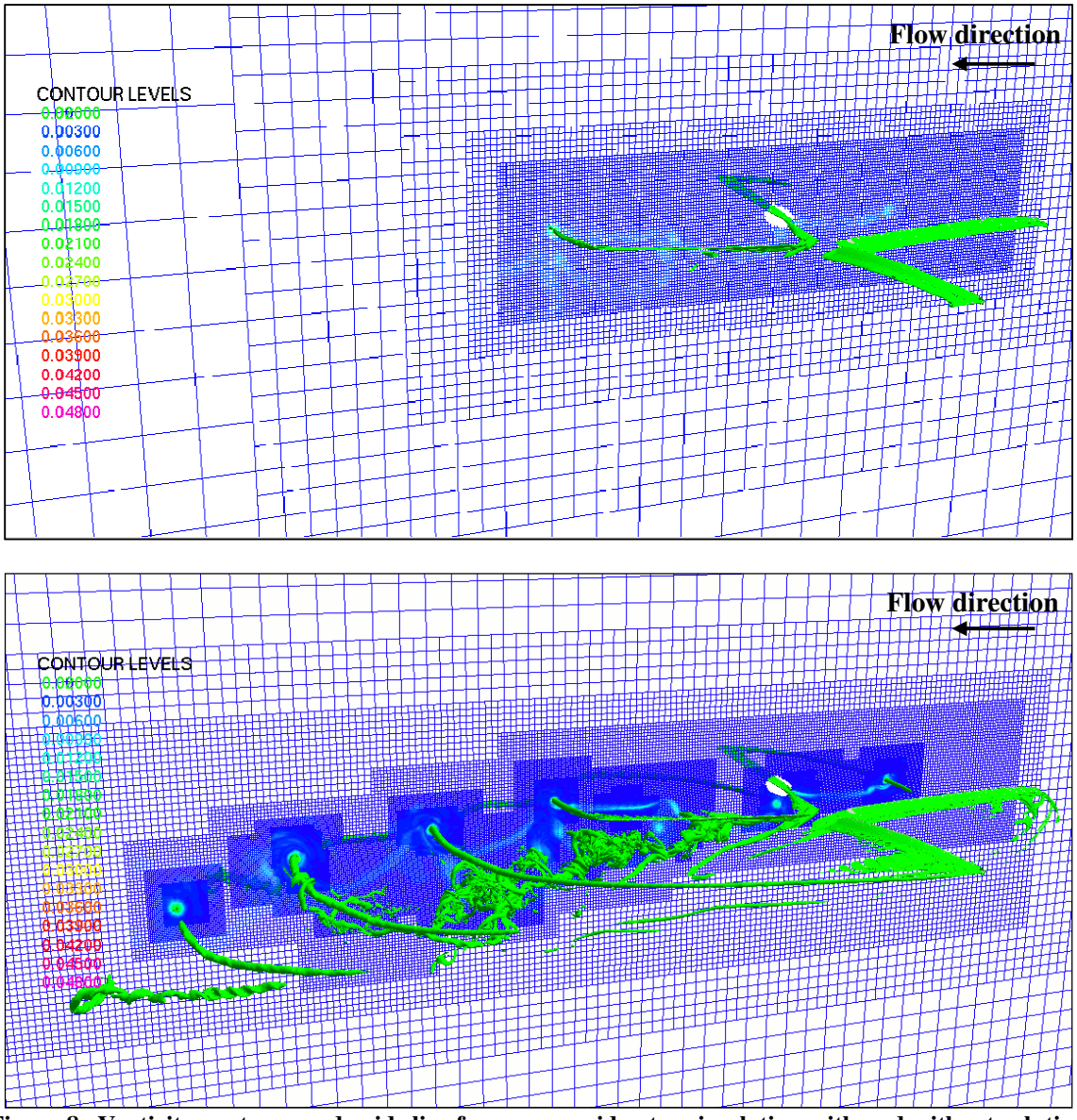

Figure 8. Vorticity contours and grid slice for coarse-grid rotor simulation with and without solution adaption (flow is right-to-left).

of Q as the sensor function. The grid refinement threshold is $(1/8)^5$, and the coarsen threshold is $(1/8)^7$. After one full revolution, the adapted solution contains 66.5 million grid points. In this case, grid adaption was not limited to the domain near the rotor, allowing the tip vortices to be resolved as they are convected downstream. This contributes to the large number of grid points in the adapted grid system.

As in previous examples, the adapted grid system provides an improved representation of the flow features in the off-body region of the domain. Figure 8 compares a vorticity contour surface and a representative slice of the computational grid for the two simulations. The contour surface shows that the tip vortices are preserved much longer with the adapted grid system than without. The grid slice illustrates that the solution adaption scheme has applied the maximum allowed two levels of refinement to the vortices, and refinement to the blade wakes as well. Though not a quantitative assessment, clearly the undivided second-difference sensor function has identified the rotor wake and tip vortices in this flow. Both simulations use a 5th-order upwind WENO scheme for the inviscid fluxes.

In this particular case, integrated blade loads are not significantly affected by adaption. This is due to the trajectory of the tip vortices, and to the coarseness of the near-body blade grids. The effect of the latter (and the need for adaption in the near-body region) has been nicely illustrated by Wissink et al.[4]

In order to evaluate the efficiency of the adaption scheme, we look at the amount of time spent in the adaption process compared to tasks that are performed every time step. This simulation was performed using 128 MPI processes, starting with 43 grids and 4.5 million grid points, and ending with 1871 grids and 66.5 million points. 1440 time steps were used to advance one revolution, using 20 dual time-steps (subiterations) per physical time step. Solution adaption was performed every 10 time steps. The simulation took a total of 19.2 hours, of which 81% was used directly by the flow solver, 7.3% idle (synchronizing), 6.5% exchanging Chimera boundary data, 2.4% computing overset grid connectivity, and 1.6% performing grid adaption. This grid adaption time includes computing the sensor function and identifying coarsen and refinement boxes, generating and load-balancing the new off-body grid system, transferring near-body grids and solutions to new processes, and interpolating old off-body grid solutions onto the new off-body grids.

Even though a large number of component grids are often created by the adaption process, the mechanics of the solution procedure is not significantly affected. Each additional grid does introduce explicit boundaries in the computational domain, which can affect convergence of the time-advancement process. This is ameliorated by the use of dual time-stepping or Newton subiteration, which serves to iterate the Chimera boundary communication (besides reducing implicit factorization error and providing 2nd-order time accuracy). We also note that the MPI parallelization relies on splitting component grids until a load-balanced grid system is obtained, so having a large number of grids can make this process easier.

B. TRAM Tiltrotor in Hover

The second rotorcraft application is taken from Ref. 19, which uses high resolution Navier-Stokes simulation to examine the wake of a rotor in hover. The geometry is the Tilt Rotor Aeroacoustics Model (TRAM), a ¼-scale wind tunnel model similar to a rotor of the V-22 Osprey. Wind tunnel testing gathered aeroacoustics, aerodynamic loads, and structural loads data for validation of rotorcraft analysis methods.[20] The case presented here has a blade tip Mach number of 0.625, corresponding Reynolds number of 2.1 million per tip chord, and collective angle of 14 deg.

Computational grids include O-grids about the three blades and center hub, and cap grids to resolve the blade root and tip. Level 1 off-body grid spacing is 1/10th of the tip chord, with the extent of the level 1 grid fixed at 0.4 and 2 rotor radii above and below the rotor, respectively, and 2.5 radii in the horizontal plane. The far-field boundaries are 17 radii from the rotor. Without adaption, this results in a total grid size of 34.6 million points, of which 62% are in the off-body grids. Computations considered here use 6th-order central differencing with 5th-order artificial dissipation for the inviscid fluxes. Details of this high-order central differencing are presented by Pulliam.[21] Viscous terms and near-body grid metric terms are evaluated to 2nd-order accuracy. A 2nd-order time-advance scheme is used with 20 dual time-step subiterations per physical time step. One time step corresponds to ¼ deg. rotation of the rotor, yielding 1440 steps per revolution.

We compare performance of calculations on the non-adapted grid system with those using solution adaption based on vorticity, with adaption performed every 20 time steps. With two levels of refinement, total grid size after 10 revolutions stabilized at 671 million grid points in 14,000 grids. Figure 9 shows a sample of vorticity magnitude on a centerline cut of the adapted grid solution, showing in great detail the tip vortices and entrainment of the shear layers behind the blades. From runs with 1536 Intel Nehalem cores (each an MPI process), the non-adapted grid system took approximately one hour per revolution, while the adapted case took 18 hours for 19 times the number of points.

A sample run of 1/3 revolution on 1536 cores, averaging 671 million grid points, took 6.0 hours. Of this time, the flow solver used 59.2%, 11.7% was idle time (which includes synchronizing the MPI processes), 10.1% was spent exchanging Chimera boundary information, 8.2% was used to compute overset grid connectivity, and grid adaption took 8.7%. Adaption was performed 24 times, and 98% of the grid points belong to off-body grids. The amount of time spent performing solution adaption is reasonable, given the large number of grids and percentage of off-body points. A comparable calculation of 10 revolutions on 576 cores without adaption took 31.6 hours, with 71.7% flow solver, 15.9% idle, 10.0% Chimera boundary exchange, and 1.8% overset grid connectivity time. With adaption we see an increase in the amount of time computing overset grid connectivity besides the time spent doing the adaption. This is due to the increased complexity of the adapted overset grid system. The use of solution adaption has not increased the percentage of idle time over the non-adapted case, indicating that load-balancing of the computation has been maintained through the redistribution of grids associated with each adaption step.

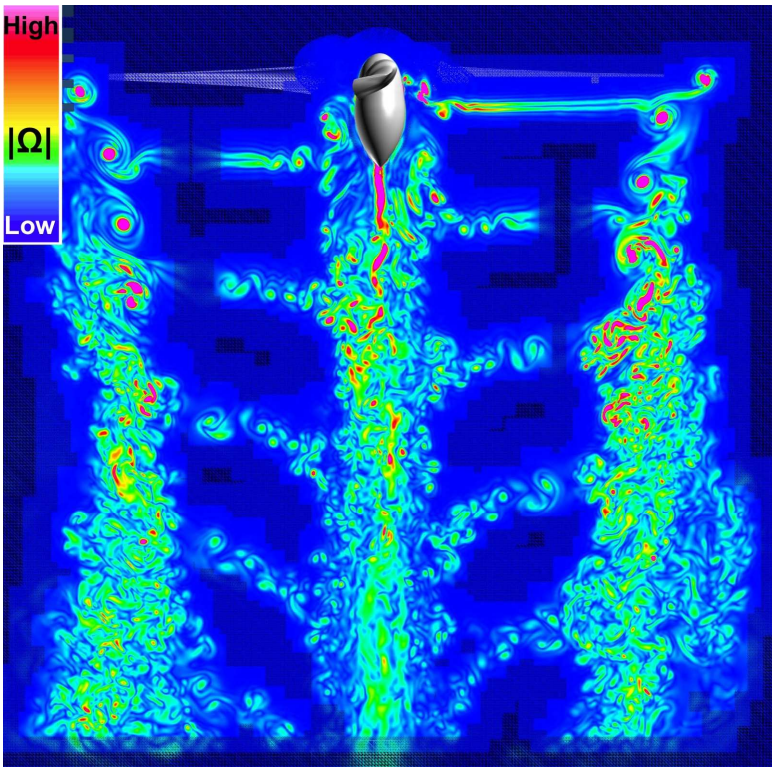

Figure 9. Vorticity magnitude on a centerline cut showing solution adaption for the TRAM rotor with 14 deg. collective (from Ref. 19).

From the timing breakdowns of the UH-60 and TRAM cases we see that (1) the adaption process has a large effect on the size of the computational grid, (2) the flow solver is effectively parallelized and load-balanced even with repeated grid adaption steps, and (3) the adaption process takes a reasonably small fraction of the total time. This demonstrates that the solution adaption scheme is well suited for time-accurate simulations from the point-of-view of computational expense.

VIII. Conclusions

We have presented a scheme for solution adaption of the Cartesian off-body grids in a structured, overset grid flow solver. The grid adaption and solution interpolation (as well as the off-body grid generation) are an integral part of the flow solver, providing a high level of efficiency. The scheme is implemented in a parallel fashion using MPI, in particular the interpolation of the flow solution onto the new grid system. Load-balancing of the flow solution process is maintained over many adaption cycles, and the adaption scheme is demonstrated to take a small fraction of the total solution time, even for significantly large problems and large numbers of grids. The adaption scheme has been shown to provide the expected improvement in solution accuracy based on grid refinement, provided that errors due to time advancement are comparatively small.

A sensor function based on the undivided second-difference of the conservative flow variables is shown to be effective in identifying both shock waves and vortices in examples presented. In other cases vorticity magnitude is used, which guarantees full grid refinement of identified flow features. While resolution of off-body flow features is clearly improved in the examples presented here, the more difficult goal of providing more accurate aerodynamic forces and moments requires similar refinement of near-body grids. The use of refined, thin near-body grids is presented as an alternative to solution adaption of the near-body grids, but does impose limits on off-body grid spacing and resolution of overset grid hole-cutting.

Future work includes extending the adaption scheme to the curvilinear near-body grids, improving methods to control the number of grid points produced by the adaption, and investigating use of an adjoint formulation as a sensor function.

Acknowledgments

This work is sponsored by the NASA Subsonic Rotary Wing Project. Rotorcraft computations were performed on the NAS Pleiades supercomputer at NASA Ames Research Center.

References

[1]Berger, M.J., and Oliger, J., "Adaptive Mesh Refinement for Hyperbolic Partial Differential Equations," *Journal of Computational Physics*, Vol. 53, 1984, pp. 484-512.

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

[3]Aftosmis, M.J., and Berger, M.J., "Multilevel Error Estimation and Adaptive h-Refinement for Cartesian Meshes with Embedded Boundaries," AIAA 2002-0863, Jan. 2002.

[4]Wissink, A.M., Kamkar, S., Pulliam, T.H., Sitaraman, J., and Sankaran, V., "Cartesian Adaptive Mesh Refinement for Rotorcraft Wake Resolution," AIAA 2010-4554, June 2010.

[5]Nichols, R.H., Tramel, R.W., and Buning, P.G., "Solver and Turbulence Model Upgrades to OVERFLOW 2 for Unsteady and High-Speed Applications," AIAA 2006-2824, June 2006.

[6]Meakin, R.L., "An Efficient Means of Adaptive Refinement Within Systems of Overset Grids," AIAA 95-1722, June 1995.

[7]Henshaw, W.D., and Schwendeman, D.W., "Parallel Computation of Three-Dimensional Flows using Overlapping Grids with Adaptive Mesh Refinement," *Journal of Computational Physics*, Vol. 227, 2008, pp. 7469-7502.

[8]Henshaw, W.D., and Schwendeman, D.W., "An Adaptive Numerical Scheme for High-Speed Reactive Flow on Overlapping Grids," *Journal of Computational Physics*, Vol. 191, 2003, pp. 420-447.

[9]Benek, J.A., Buning, P.G., and Steger, J.L., "A 3-D Chimera Grid Embedding Technique," AIAA 85-1523, July 1985.

[10]Meakin, R.L., "Object X-Rays for Cutting Holes in Composite Overset Structured Meshes," AIAA 2001-2537, June 2001.

[11]Meakin, R.L., "A New Method for Establishing Intergrid Communication Among Systems of Overset Grids," AIAA 91-1586, June 1991.

[12]Chiu, I.T., and Meakin, R.L., "On Automating Domain Connectivity for Overset Grids," AIAA 95-0854, Jan. 1995.

[13]Chan, W.M., Meakin, R.L., and Potsdam, M.A., "CHSSI Software for Geometrically Complex Unsteady Aerodynamic Applications," AIAA 2001-0593, Jan. 2001.

[14]Nichols, R.H., Tramel, R.W., and Buning, P.G., "Evaluation of Two High-Order Weighted Essentially Nonoscillatory Schemes," *AIAA Journal*, Vol. 46, No. 12, 2008, pp.3090-3102.

[15]Holst, T.L., "Viscous Transonic Airfoil Workshop Compendium of Results," AIAA 87-1460, June 1987.

[16]Lee, H.C., and Pulliam, T.H., "Studying the Effects of Grid Generation Strategies and Grid Adaption on Airplane Geometries Using OVERFLOW," AIAA 20th Computational Fluid Dynamics Conference, Honolulu, HI, June 2011 (to be published).

[17]Sclafani, A.J., Slotnick, J.P., Vassberg, J.C., Pulliam, T.H., and Lee, H.C., "OVERFLOW Analysis of the NASA Trap Wing Model from the First High Lift Prediction Workshop," AIAA 2011-0866, Jan. 2011.

[18]Potsdam, M., Yeo, H., and Johnson, W., "Rotor Airloads Prediction Using Loose Aerodynamic/Structural Coupling," American Helicopter Society (AHS) 60th Annual Forum, Baltimore, MD, June 2004.

[19]Chaderjian, N.M., and Buning, P.G., "High Resolution Navier-Stokes Simulation of Rotor Wakes," American Helicopter Society (AHS) 67th Annual Forum, Virginia Beach, VA, May 2011.

[20]Johnson, J.L., and Young, L.A., "Tilt Rotor Aeroacoustics Model Project," Confederation of European Aerospace Societies (CEAS), Forum on Aeroacoustics of Rotorcraft and Propellers, Rome, Italy, June 1999.

[21]Pulliam, T.H., "High Order Accurate Finite-Difference Methods: As Seen in OVERFLOW," AIAA 20th Computational Fluid Dynamics Conference, Honolulu, HI, June 2011 (to be published).