A Coupled Unstructured-Adaptive Cartesian CFD Approach for Hover Prediction

Andrew Wissink, Mark Potsdam, Venkateswaran Sankaran

U.S. Army Aeroflightdynamics Directorate (AMRDEC)
Research, Development, and Engineering Command
Ames Research Center, Moffett Field, CA

Jayanarayanan Sitaraman, Zhi Yang, and Dimitri Mavriplis

Department of Mechanical Engineering
University of Wyoming, Laramie, WY

An innovative computational fluid dynamics (CFD) approach is employed to predict the aerodynamic performance of hovering rotors. Two different CFD solvers are applied in different parts of the computational domain: a body-fitted unstructured solver near the blade surface to capture complex geometry and viscous effects, and a high-order block-structured Cartesian solver away from the blade to capture the wake. The Cartesian solver applies Adaptive Mesh Refinement (AMR) to resolve tip vortices. Results are demonstrated on the TRAM isolated rotor. The results show the approach is able to achieve aerodynamic figure of merit performance predictions to within 2% of experiment on a 64-core distributed parallel Linux cluster. Solution driven AMR is effective for resolving the vortex wake at significantly reduced computational cost over fixed-grid calculations with similar resolution.

NOTATION

x	streamwise coordinate
y	spanwise coordinate
z	normal coordinate
u	local flow velocity (in x, y, z)
r	blade radial coordinate
R	blade radius
Ψ	blade azimuth angle
M_{tip}	tip Mach number
C_{tip}	tip chord length
Re_{tip}	tip Reynolds number
θ	blade collective pitch angle
Ψ	blade azimuth angle
C_P	blade pressure coefficient
$M^2 c_n$	blade section normal force coefficient times Mach number squared
C_T	rotor thrust coefficient
C_Q	rotor torque coefficient
FM	rotor figure of merit = $C_T^{3/2}/C_Q\sqrt{2}$
Q_{crit}	Q criterion = $\frac{1}{2}\left(\|\Omega\|^2 - \|S\|^2\right)$ $\nabla u = \Omega + S$, Ω = vorticity, S = strain

American Helicopter Society 66th Annual Forum, Phoenix, AZ, May 11–13, 2010.

INTRODUCTION

The ability to accurately predict the hover performance of a rotor plays an important role in rotary-wing vehicle design. A 1% change in figure of merit of a rotor (a commonly used measure of rotor efficiency) can translate to 200-300 lbs. change in useful payload. Computational tools available today are able to predict to this level of accuracy only with empirically based methods that are tuned through windtunnel and flight tests, making them unreliable for new and novel rotor designs (Ref. 1). Inadequate modeling of the complex flow physics near the blade, poor resolution of the wake, or a combination thereof, are the main reasons for the inaccuracies in these approaches.

Over the past decade high-fidelity simulations based on solutions of the Reynolds-Averaged Navier Stokes (RANS) equations have been shown to accurately compute aerodynamic loads to within a few percent of experimentally measured values (Refs. 2–7). Many of these approaches use body-fitted codes within an overset mesh framework. Specifically, the OVERFLOW (Ref. 8) code utilizes a dual-mesh approach with structured body-fitted meshes applied near the blade with Cartesian meshes in the wake. Moreover, the benefits of high-order schemes have been shown to be important for accurate performance predictions by Hariharan (Ref. 9), Sankar (Ref. 10), Yeshala (Ref. 11), and Duque et. al. (Ref. 12). Despite these advances, routine use

of these tools by design engineers remains limited. Problem setup requirements (such as manual grid generation), the relatively high computational costs, and the considerable CFD knowledge that is required all present significant hurdles to more widespread adoption.

In recent years, unstructured grid generation has become widely adopted in industry because it is considered faster and more automatic than structured grid generation. A number of commercially available unstructured gridding packages are now available that link to standard CAD design software, enabling a more seamless analysis process from CAD to simulation. The flexibility inherent in unstructured meshing more easily allows for use of parametrized surfaces that may be modified during the calculation, facilitating design optimization (Ref. 13). Unstructured solvers can be implemented within an overset framework and used for rotorcraft, as demonstrated recently by Lee-Rausch et al. (Ref. 7). However, unstructured methods in general tend to be less computationally efficient than their structured counterparts, both in terms of memory and CPU time. Also, spatial accuracy is generally limited to second order. This is particularly problematic for rotorcraft wake resolution which, as noted earlier, is important for accurate load predictions. Moreover, high-order algorithms implemented on structured meshes, particularly Cartesian meshes, are currently much more efficient than high-order Discontinuous Galerkin (DG) schemes applied to tetrahedral elements of unstructured meshes.

In this work we introduce an innovative hybrid CFD approach that attempts to combine the benefits of both structured and unstructured approaches, while avoiding their limitations. It maintains the benefits of unstructured methods for modeling complex geometry, while it also utilizes the efficiency and accuracy of high-order schemes available in structured solvers for the wake. In this hybrid-CFD approach, a RANS unstructured flow solver is applied near the blade surface, or the "near-body" region, to resolve geometry and wall-bounded viscous effects while away from the body surface, in the so-called "off-body" region, a high-order Structured Adaptive Mesh Refinement (SAMR) Euler solver is applied. Benefits of the unstructured solver is that it makes it possible to utilize commercially-available meshing tools for the CAD-to-CFD mesh process. Benefits of the block structured solver is that it is very efficient and enables high-order accurate algorithms to be applied at minimal cost.

A key element of this coupled unstructured-Cartesian approach is that the off-body solver also facilitates adaptive mesh refinement (AMR), a process in which local mesh resolution is dynamically adjusted by the solver to resolve important flow features and coarsen the mesh in benign flow regions. It is important to note that the AMR is done automatically as part of the solution process and is controlled by the flow solver. This enables more automation in what is typically a tedious manual process of mesh generation and also enables more targeted refinement of important flow features. The technology is particularly promising for rotorcraft wake flowfields where fine scale tip vortices exist within a relatively large computational domain.

One of the first efforts at using AMR for rotary-wing wakes was by Strawn and Barth (Ref. 14) in 1992 using an unstructured Euler solver. Later, Kang and Kwon (Ref. 15) applied a similar strategy within an unstructured Navier-Stokes solver. Potsdam (Ref. 16) recently used AMR in an unstructured RANS solver for wake resolution of wind turbine predictions. Cartesian-based AMR has been applied within an overset approach by Meakin (Ref. 17) using non-refined near-body structured curvilinear meshes. This approach was further extended recently by Holst and Pulliam (Refs. 18, 19) who successfully applied it to rotorcraft hover simulations. Vasilescu et al. (Ref. 20) presented an alternative approach for structured adaptive gridding for rotor wakes. Dietz et al. (Ref. 21) also introduced an approach which is overset-based but moves around tubular curvilinear grids to align with the tip vortices, instead of using Cartesian meshes.

Most of the aforementioned efforts target steady-state problems which achieve a reasonable solution with a small number of mesh adaptation cycles. Extending the approach to time-dependent problems, such as hover, forward flight, or rotor-fuselage calculations with meshes that move relative to one another requires an AMR approach in which mesh refinement and coarsening are performed continuously during the course of the calculation. The mesh may be adapted tens to hundreds of times over the course of the simulation so a tight integration is required between the mesh refinement scheme and the flow solver in order to make this process efficient. Careful consideration of parallel computing issues like load balancing and re-establishing data communication after adaptation become critical to ensure efficient computational performance. Because of the relative difficulties associated with the application of solution adaptive schemes to rotorcraft problems with moving meshes only a few researchers have successfully demonstrated it. In 2004 Park and Kwon (Ref. 22) used an unstructured code with sliding surfaces to demonstrate unsteady AMR on a trimmed rotor in forward flight. Nam et al. (Ref. 23) later used a similar approach to apply AMR to calculations of a rotor-fuselage configuration.

The goal of the current paper is to present the details of the coupled unstructured-adaptive Cartesian approach and to investigate its performance for steady (non-inertial frame) and unsteady (inertial-frame) rotor hover applications. Specifically, we demonstrate both the predictive performance as well as the computational performance of the proposed approach. The following section discusses details of the computational approach. The Results section presents various comparisons for calculations of the Tilt Rotor Aeroacoustics Model (TRAM) rotor in hover conditions. First, we provide a comparison to standard fully

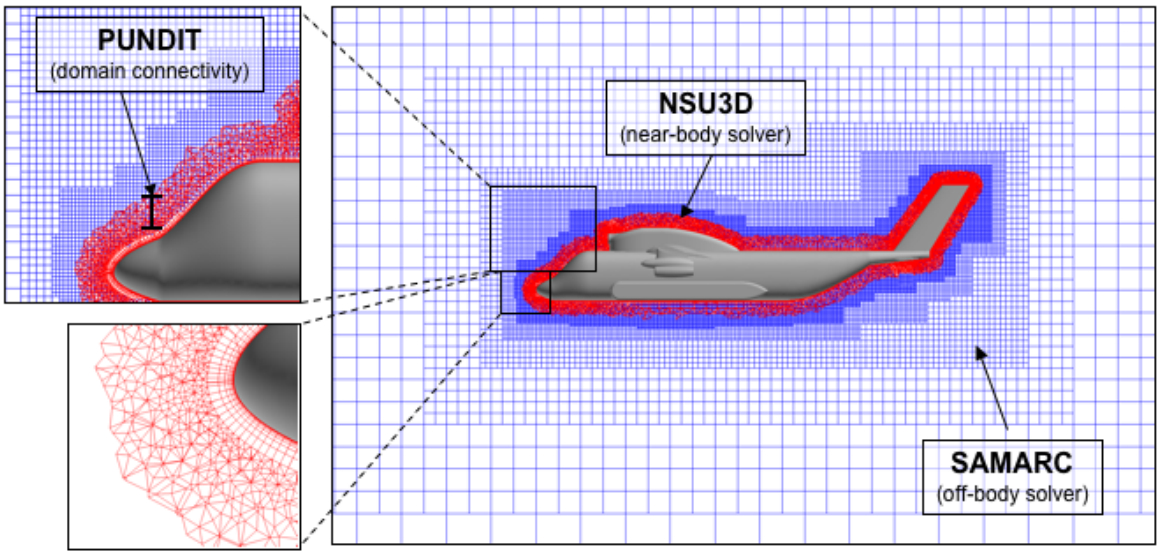


Fig. 1. Near-body/off-body overset grid system.

unstructured calculations for a fixed-collective case. Following this, we present a more detailed study on the effects of grid resolution, steady non-inertial vs. unsteady inertial hover calculations, and considerations in the use of AMR. Finally, we present detailed experimental-computational comparisons for a collective sweep. Conclusions are summarized at the end.

COMPUTATIONAL APPROACH

The spatial discretization scheme employs a dual-mesh overset paradigm with body-aligned unstructured grids used near the surface, the "near-body" region, and adaptive Cartesian grids in the farfield, the "off-body" region. The near-body grid is a mixed-element unstructured mesh that is "subsetted" a certain distance from the surface. Its role is to transition from the very high aspect ratio viscous boundary layer cells near the surface to isotropic Cartesian cells in the field. This near/off-body overset meshing approach allows the boundary layer to be accurately resolved near the surface while accurately resolving effects away from the body with adaptive Cartesian grids and high-order numerical schemes.

The solver applied to the near-body mesh is the NSU3D code (Ref. 24). The block structured adaptive Cartesian code SAMARC, composed of SAMRAI (Ref. 25) and ARC3DC (Ref. 26), is used for the off-body calculations. The PUNDIT software (Ref. 27) manages the Chimera grid hole cutting and interpolation. These software packages are developed as standalone codes and integrated through a Python-based infrastructure (Refs. 28, 29). Other packages relating to CFD/CSD coupling and trim are also linked in to this infrastructure but are not relevant to the calculations in this paper and are, therefore, not discussed. Use of a high-level Python-based software integration framework enables the ability to plug-in different solvers as needed. The code infrastructure that includes the integration framework and all requisite solvers is called "Helios".

Further details of each of these packages are given in the following sub-sections.

Unstructured Near-Body Solver

The near-body solver, NSU3D, is an unstructured grid unsteady Reynolds-averaged Navier-Stokes solver developed for high-Reynolds number external aerodynamic applications. The NSU3D discretization employs a second-order accurate node-based approach, where the unknown fluid and turbulence variables are stored at the vertices of the mesh, and fluxes are computed on faces delimiting dual control volumes, with each dual face being associated with a mesh edge. This discretization operates on hybrid mixed-element meshes, generally employing prismatic elements in highly stretched boundary layer regions, and tetrahedral elements in isotropic regions of the mesh. A single edge-based data structure is used to compute flux balances across all types of elements. The single-equation Spalart-Allmaras turbulence model, as well as a standard $k-\omega$ two-equation turbulence model are available within the NSU3D solver.

The NSU3D solution scheme was originally developed for optimizing convergence of steady-state problems. The basic approach relies on an explicit multistage scheme which is preconditioned by a local block-Jacobi preconditioner in regions of isotropic grid cells. In boundary layer regions, where the grid is highly stretched, a line preconditioner is employed to relieve the stiffness associated with the mesh anisotropy (Ref. 30). An agglomeration multigrid

algorithm is used to further enhance convergence to steady-state (Refs. 24, 31). The Jacobi and line preconditioners are used to drive the various levels of the multigrid sequence, resulting in a rapidly converging solution technique.

For time-dependent problems, first and second-order implicit backwards difference time discretizations are implemented, and the line-implicit multigrid scheme is used to solve the non-linear problem arising at each implicit time step. NSU3D has been extensively validated in stand-alone mode, both for steady-state fixed-wing cases, as a regular participant in the AIAA Drag Prediction workshop series (Ref. 32), as well as for unsteady aerodynamic and aeroelastic problems (Ref. 33), and has been benchmarked on large parallel computer systems (Ref. 34).

For operation within an overset environment, "iblanking" capability has been added. The "iblanks" specify which nodes the solution variables are to be updated in the near-body solver (iblank = 1) and which nodes are not updated, i.e., fringes and holes (iblank = 0). The fringes correspond to the overset regions of the near-body grid, which are updated by the off-body code, while holes correspond to locations where the grid intersects solid objects.

Cartesian Off-Body Solver

Block-structured Cartesian grids are used to resolve the far-field wake. There are several advantages Cartesian grids present over typical tetrahedral off-body elements used by most unstructured codes. The Cartesian solver can exploit structured data, maximizing cache or vector processor performance. Numerical operations can be optimized because the uniformity of the Cartesian grid eliminates the need to apply grid metrics in the differencing, enabling reduced floating point operations and simplified algorithms. Advanced numerical algorithms, such as implicit solvers, multi-grid, and high-order algorithms are all straightforward on Cartesian grids.

The block-structured Cartesian solver is also efficient in its memory usage. Each Cartesian grid block may be completely defined by the indices of the block diagonal (6 integers), the lower and upper indices, and the level of refinement. In total, only 7 integers are needed to define an entire 3D block. The number of blocks used in a typical calculation is generally in the 100's or 1,000's. A tetrahedral mesh requires storage of the node and/or edge locations and requires *millions* of float or double precision data elements to store its mesh.

The main reason that structured Cartesian meshes have not seen more widespread use in CFD is their inability to accurately represent geometrically complex viscous boundaries. In our approach the near-body solver manages that task. Thus, the Cartesian grids are not used at viscous boundaries. AMR is used to refine the Cartesian grids to the outer boundaries of the near-body unstructured mesh, as well as to desirable features in the wake.

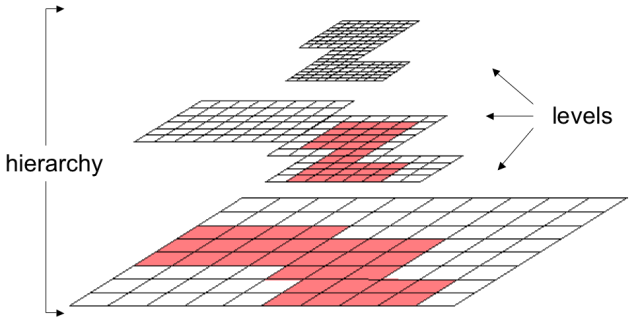


Fig. 2. Off-body block structured AMR mesh composed of a hierarchy of nested levels of refinement. Each level contains uniformly-spaced logically-rectangular regions, defined over a global index space.

The structured adaptive mesh refinement (SAMR) strategy used for the Cartesian off-body solver is based on the ideas of Berger, Colella, and Oliger (Refs. 35, 36). Grid levels composed of a union of 3D Cartesian blocks of like refinement are stored as a composite grid hierarchy. See Fig. 2. Grid levels are constructed from coarsest to finest. The coarsest level defines the physical extent of the computational domain. Each finer level is formed by selecting cells on the coarser level and then clustering the marked cells together to form block regions that will constitute the new finer level. All grid cells on a particular level have the same spacing, and the ratio of spacing between levels is generally a factor of two or four, although it is possible to use other refinement ratios as well.

Computations on the SAMR grid hierarchy are carried out in parallel by distributing the different computational blocks over processors. Each time the grid is adapted it must be repartitioned for load balancing and data communication patterns re-established between processors. It is this process that typically hinders the scalability of unstructured AMR codes. Since the grid is partitioned over processors, significant communication must take place to properly re-partition the mesh and data. This is why most unstructured grid adaptation schemes apply this process as a post-processing step rather than tightly integrating it with the solver. The SAMR paradigm uses such a low-memory mesh description that the block boundaries for the entire 3D composite mesh hierarchy can be known to all processors, minimizing the amount of information that needs to be exchanged during the repartition and making reconstruction of the communication patterns very fast and efficient.

An additional advantage of the SAMR paradigm is that it facilitates a clean separation between grid- and solution-based operations. Grid-based operations – e.g. adaptive grid generation, parallel decomposition, data exchange between blocks, etc. – can be managed by one package while the serial numerical operations on each block are performed by a separate single-block solver. In this work the grid-based op-

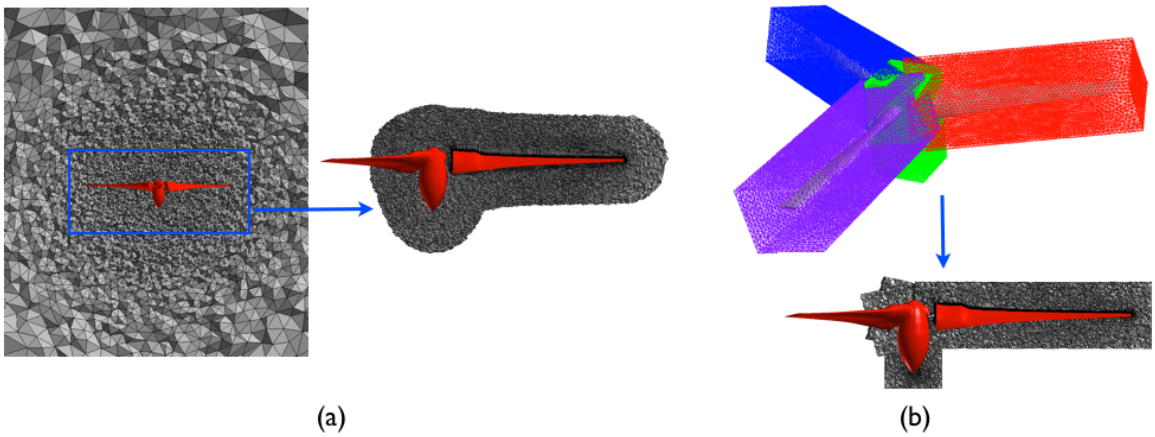


Fig. 3. Near-body unstructured mesh generation options; (a) subsetted full-domain unstructured volume mesh, (b) assembled individual component meshes.

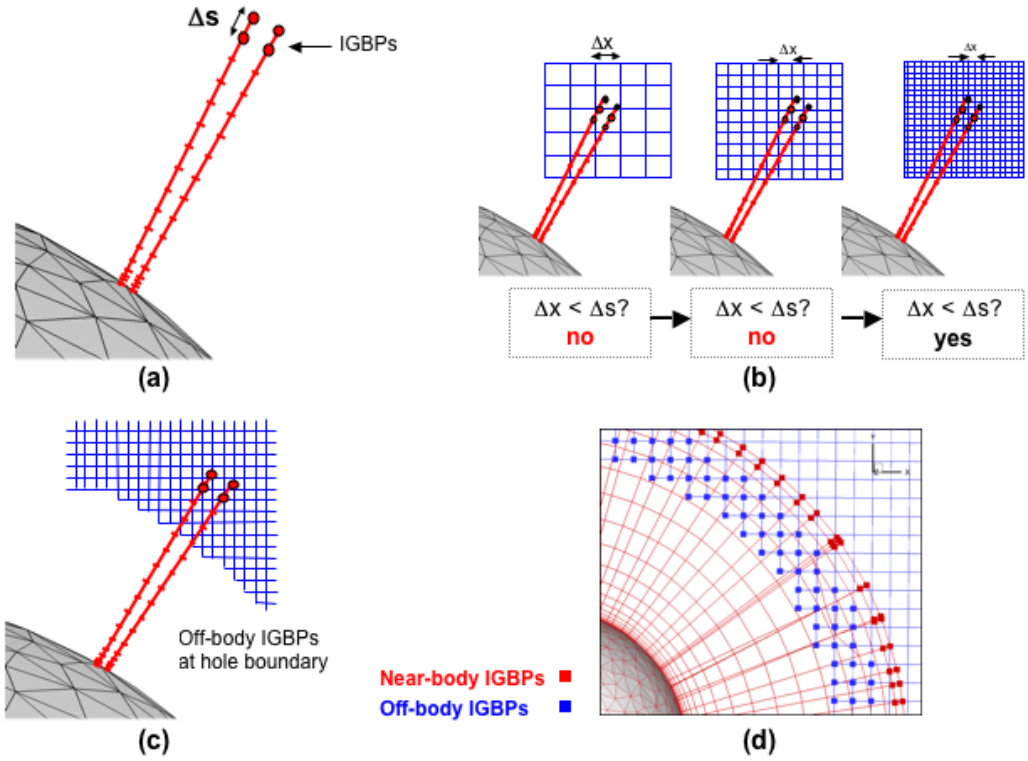


Fig. 4. Cartesian off-body grid generation. (a) determination of near-body inter-grid boundary point (IGBP) spacing, (b) refinement of the Cartesian blocks to match spacing, (c) implicit hole cutting, and (d) determination of off-body IGBPs.

erations are managed by the SAMRAI (Refs. 25, 37, 38) infrastructure from Lawrence Livermore National Lab. SAMRAI manages the construction and adaptation of the AMR grid hierarchy, parallel load balancing, and MPI-based data exchanges between grid blocks. Then, the single block solver applied to each Cartesian block is ARC3DC by Pulliam (Ref. 26). ARC3DC is a version of NASA Ames' ARC3D solver with high-order operations optimized for isotropic Cartesian grids. The high-order finite difference schemes used in ARC3DC are based on central differences with a dissipation term. For instance, a 6th-order central difference scheme uses 5th-order dissipation, making it formally 5th-order accurate spatially.

A 3rd-order accurate 3-stage explicit Runge-Kutta scheme is used for time integration in ARC3DC. All Cartesian grid levels execute the explicit RK scheme with a uniform timestep so the overall timestep is governed by the spacing on the finest level. We currently do not refine in time, although it is possible to do so. At the beginning of each RK sub-step, data on fine patch boundaries are updated either through copying data from a neighboring patch on the same level, if one exists, or through interpolation of data from a coarser level. The number of boundary points required in this exchange depends on the order of the spatial discretization. For example, a 6th-Order central scheme uses a 7-point stencil, requiring three boundary points be exchanged between patches. The numerical operations to advance a single RK sub-step are performed simultaneously across processors on each patch of each level. Data are then injected from fine levels into coarse levels wherever overlap exists. Communication between blocks on the same level, and between blocks on different levels, is managed by SAMRAI.

Domain Connectivity Formulation

The PUNDIT code (Ref. 27) manages all Chimera grid hole cutting and interpolation. Data is transferred between the unstructured near-body mesh and the background Cartesian mesh using standard second order interpolations used commonly for both structured (Refs. 39, 40) and unstructured (Refs. 7, 41, 42) overset applications. PUNDIT uses an implicit hole cutting strategy; it searches all overlapping mesh points and identifies the cell with the best resolution (i.e. smallest volume). It then manages the interpolation of data between the unstructured and Cartesian mesh systems. PUNDIT operates on the partitioned grid data and uses MPI for parallel data exchange.

Grid Generation/Adaptation Procedure

The starting point for the grid generation process is an unstructured near-body mesh. A commercial grid generation package is used to interface with the CAD model and construct a triangulated surface mesh and the AFLR software from Marcum (Ref. 43) is used to construct a mixed prism-tet volume mesh. For bodies that contain multiple components the volume mesh can be constructed in either of two ways, as illustrated in Fig. 3. In the first case, Fig. 3(a), the volume mesh encompasses multiple surface components and extends over the full solution domain. A subsetting operation is then applied to trim off the unstructured elements a certain distance from the bounded surface. In the second case, Fig. 3(b), individual unstructured volume meshes are generated around each component – e.g. blade and centerbody in the figure shown. In this case the volume mesh outer boundary extends only a short distance from the bounded surface. The individual component meshes can be duplicated, translated, and rotated to assemble the near-body mesh system. While option (b) is the most flexible in terms of allowing assembly of complex bodies through simple mesh movement, translation, and rotation, option (b) permits use of unstructured meshes that may have already been generated for use in existing unstructured solvers.

The off-body block-structured Cartesian grid system is automatically generated, targeting refinement to mesh spacing requirements from the near-body mesh and to selected solution quantities. Levels are constructed from coarsest to finest. The coarsest level defines the physical extent of the computational domain. Each finer level is formed by selecting cells on the coarser level and then clustering the marked cells together to form the regions that will constitute the new finer level.

Figure 4 illustrates the mesh refinement process that satisfies near-body resolution requirements and establishes an initial off-body mesh to begin the simulation. Points at the outer boundary of the subsetted near-body mesh receive interpolated data from the off-body Cartesian mesh. Mesh points that receive interpolated data are referred to as intergrid boundary points (IGBPs). They do not compute a solution, they simply receive data interpolated directly from the overlapping background grids. At these points it is desirable that the Cartesian resolution match or be finer than the unstructured mesh resolution to ensure smooth solutions across the overset boundary. The locations (x, y, z coordinates) and resolution Δs, generally the minimum distance to neighboring grid points, of the near-body IGBPs are first computed (Fig. 4(a)). Next, cells on the off-body Cartesian grid system that contain IGBPs are checked to see whether their grid spacing Δx is greater than the spacing Δs at the near-body IGBP. If the resolution is not sufficient ($\Delta x > \Delta s$) the cell is marked for refinement (Fig. 4(b)). All marked cells are clustered to construct a new finer level in the Cartesian hierarchy, and the process is repeated until no Cartesian cells are marked i.e., the resolution of all near-body IGBPs has been satisfied by the off-body Cartesian grid system. At that point implicit hole cutting (Fig. 4(c)) is performed by PUNDIT to establish the set of off-body IGBPs (Fig. 4(c)), the Cartesian mesh points that will receive interpolated data from the near-body unstructured solver. PUNDIT manages

Blade Radius R	57 in
Tip Chord C_{tip}	5.5 in
Tip Mach Number M_{tip}	0.625
Re Number at tip Re_{tip}	2.1×10^6

Table 1. Nominal operating conditions for isolated TRAM rotor studies

the hole cutting and data interpolation between the near-body and off-body solvers.

RESULTS

Results are demonstrated for calculations of the isolated Tilt Rotor Aeroacoustics Model (TRAM) rotor, which is a quarter-scale model of the Bell/Boeing V-22 Osprey tiltrotor aircraft three-bladed rotor. Nominal operating conditions used for the computational model are shown in Table 1. Experimental results were taken from tests conducted in the Duits-Nederlandse Windtunnel Large Low-speed Facility (DNW-LLF). Aerodynamic surface pressures, thrust and power, were measured along with structural loads and aeroacoustics data.

The first sub-section compares results from a fully unstructured calculation and the new dual-mesh approach with a fixed (i.e., without AMR) Cartesian off-body mesh. For these initial studies, we use the 14° collective condition in steady non-inertial hover. The second sub-section investigates the use of solution adaptivity in the Cartesian off-body for the same problem. In addition to comparing solution accuracy and aerodynamic performance, we also compare the relative computational efficiency of the dual-mesh approach with and without AMR. We next investigate the performance of the code for time-dependent inertial hover calculations, comparing with the steady fixed-mesh results. The final sub-section presents results of a collective sweep study and compares the results with experimental data.

Fixed Mesh Results

We start with steady state results obtained using a non-inertial (or rotational) frame of reference for a fixed 14° collective. In the rotational frame, the coordinate frame is attached to the rotor hub and rotates along with the blades so that the resultant flowfield appears steady. Notably, there is no relative motion between the near-body and off-body meshes and the domain connectivity formulation between these mesh systems needs to be performed only once at the commencement of the computations. A variety of different near-body and off-body meshes are tested. In general, we start with a given near-body mesh and then generate appropriate off-body meshes depending upon the resolution of the near-body mesh and whether the off-body mesh is fixed or adapted. In this sub-section, we consider a series of near-body meshes ranging from coarse to intermediate to fine and utilize *fixed* Cartesian off-body meshes. In addition to the coupled unstructured-Cartesian meshes, we also utilize a fully unstructured mesh for comparison purposes. In the following sub-section, we further extend these to include adaptive meshes.

Solver	Near-Body	Off-Body
Fully Unstr. Med	5.0M	—
Dual-Mesh Coarse	1.79M	2.7M
Dual-Mesh Med	2.80M	15.6M
Dual-Mesh Fine	8.34M	15.6M

Table 2. Unstructured and Cartesian mesh sizes used in isolated TRAM $\theta = 14°$ fixed mesh studies.

Table 2 gives a summary of the near- and off-body mesh sizes used for the fixed-mesh studies. The fully unstructured mesh contains approximately 5M nodes comprised of prismatic and tetrahedral elements (see Fig. 3(a)). The medium resolution Helios near-body mesh is constructed by trimming this 5M point fully unstructured mesh to a distance of approximately two chord lengths from the blade surface (also shown in Fig. 3(a)), resulting in a subsetted mesh size of 2.80M points. This is referred to herein as the "medium" resolution near-body mesh. Two other near-body meshes are also utilized, both constructed by combining individual component meshes for each of the blades and the centerbody (shown in Fig. 3(b)). The first mesh is coarser, comprised of 1.79M points (517K points per blade, 80K points center-body), and the second mesh is finer, comprised of 8.34M points (2.75M points per blade). These two meshes are referred to as the "coarse" and "fine" near-body meshes in Table 2. The coarse mesh case uses an off-body grid with six levels of refinement, corresponding to a fine mesh Cartesian spacing of 20% C_{tip}. Both the medium and fine dual-mesh cases apply an off-body mesh with seven levels of refinement with fine mesh spacing of 10% C_{tip}. The medium and fine near-body meshes have similar spacing at their subset boundaries where they interface the Cartesian grid. In other words, main difference between the medium and fine mesh systems is the resolution near the blade.

Representative mesh and flowfield solutions for the fully-unstructured and medium dual-mesh cases are shown in Fig. 5. Iso-surfaces of the Q-criterion at Q=0.00001 are shown, colored by vorticity magnitude. Fig. 5(a) shows that the wake is dissipated quickly in the unstructured mesh while the wake in the dual-mesh solution Fig. 5(b) is well preserved. This improvement is as expected because the Cartesian off-body mesh utilizes considerably larger number of grid points than the unstructured mesh system. Moreover, the off-body solution is spatially fifth-order accurate while the unstructured solver is second-order throughout. As will be discussed shortly, the dual-mesh formulation enables these improvements at a computational cost that is comparable to the fully unstructured calculation.

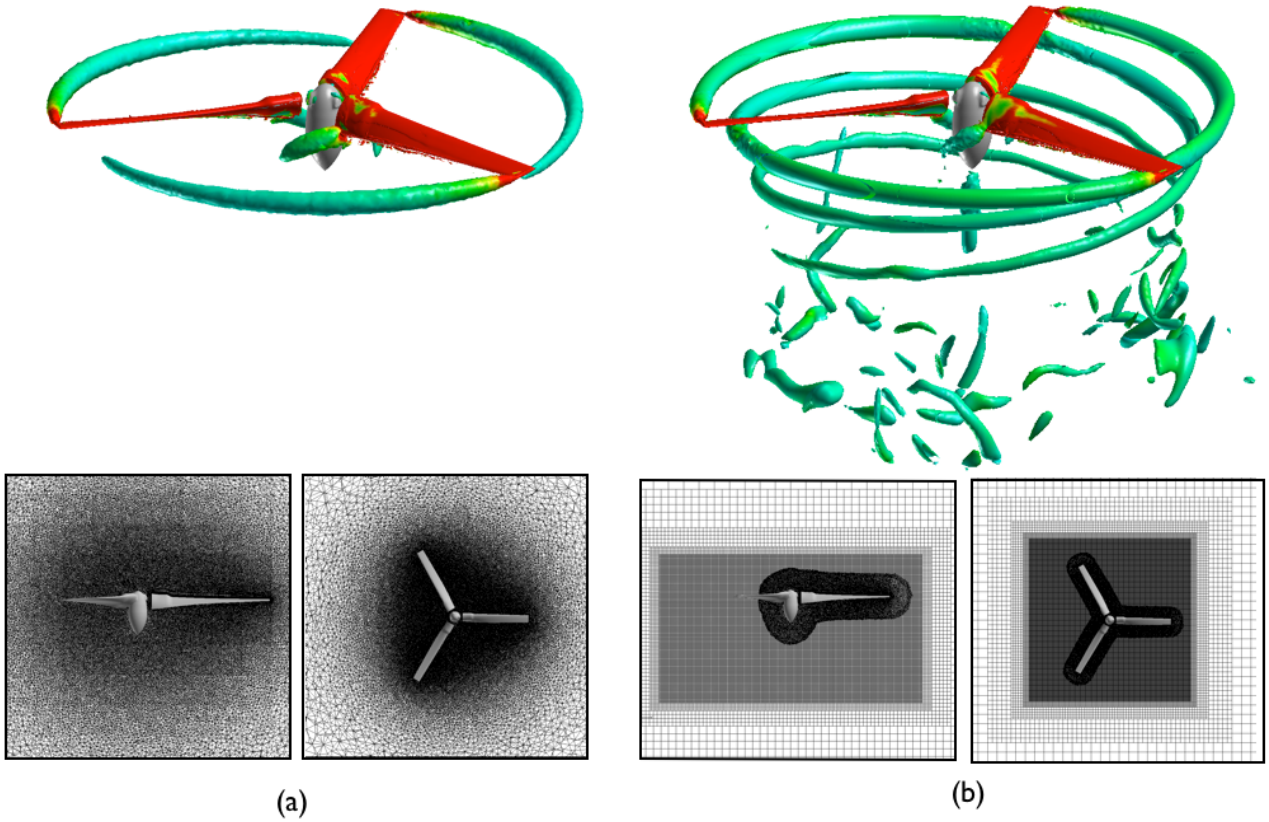


Fig. 5. Wake solution and mesh, $\theta = 14^\circ$ isolated TRAM calculation. Solution shows iso-surfaces of Q_{crit}. (a) Fully unstructured solution with NSU3D, (b) Helios medium resolution subsetted unstructured near-body mesh with fixed Cartesian off-body mesh.

	C_T	C_Q	FM
Experiment	0.0149	0.00165	0.779
Fully Unstr. Med	0.0146	0.00179	0.694
Dual-Mesh Coarse	0.0148	0.00180	0.709
Dual-Mesh Med	0.0150	0.00178	0.734
Dual-Mesh Fine	0.0151	0.00171	0.764

Table 3. Performance of isolated TRAM rotor at $\theta = 14^\circ$ using fixed meshes. Fully unstructured solution compared to fixed dual-mesh with coarse, medium, and fine resolution.

Table 3 shows the computed thrust (C_T) and power coefficients (C_Q) and the resulting Figure of Merit (FM) for the three mesh systems. All the dual-mesh cases utilize the fifth-order accurate spatial discretization scheme in the fixed-off-body mesh. Two trends are apparent from these results. First, the dual-mesh results show a progressive improvement in the predictions going from the coarse to medium to fine meshes. This may be primarily attributed to improved near-body mesh resolution which is progressing from 1.8M to 2.8M to 8.3M nodes for the three cases. On the fine-mesh, the results are within 2-3% of the target experimental values and in close agreement with results obtained with other state-of-the-art codes (Ref. 19). The second trend that is evident from these results is that, for comparable near-body mesh resolution, the dual-mesh method significantly outperforms the fully unstructured calculations. Specifically, the 5M-node unstructured prediction of FM is about 0.694, which is more than 10% below the experimental value. In contrast, the intermediate dual-mesh case which utilizes precisely the same near-body mesh system (trimmed to a wall-distance of two chord-lengths) computes a FM of 0.734, which is about 5% below the experimental value.

Table 4 compares the computational cost for the fixed mesh calculations. The cases were run on the Army Research Lab MJM computer system, a distributed parallel Linux cluster with nodes that each contain two 3.0 GHz Intel Woodcrest processors with two cores each (i.e. 4 cores per node), and 8GB of memory. The number of computational cores applied, the number of points points in the near and off-body meshes, and the measured time per step are shown. For the medium mesh case, the time with 2nd-order and 5th-order spatial resolution in the off-body are both reported. Although the off-body introduces significantly more gridpoints, the calculations on the Cartesian mesh system are significantly faster and, as a result, the extra gridpoints are still affordable. Referring to the medium mesh case specifically, the cost per step of the dual mesh Helios calculation is about 17% higer than the fully unstructured NSU3D calculation even though the total number of grid points (considering both near- and off-body meshes) is about four times that of NSU3D. Moreover, we also observe that the extra cost incurred from high-order methods is minimal, with the fifth-order accurate scheme applied to the off-body solver being only about 6% more expensive than the second-order scheme.

Adaptive Mesh Refinement

We next investigate the effects of off-body AMR on solution quality and computational performance. The dual-mesh results shown in the previous section applied off-body Cartesian meshes with a pre-defined fixed refinement region. In the results shown here, the off-body mesh is adapted during the solution to satisfy resolution requirements at the near-body mesh interface and to regions of high vorticity in order to resolve tip vortices.

At the start of the calculation, refinement of the off-body grid targets locations of the near-body inter-grid boundary points as shown in Fig. 4. During these initial stages of the solution we deliberately turn off refinement that targets specific flow quantities since there tend to be many non-physical startup effects. In other words, at the start of the solution the AMR process targets refinement specifically to satisfy mesh resolution requirements at the near-body/off-body interface, and not on solution-based quantities. As the solution progresses, the non-physical startup effects are damped out by the coarse surrounding mesh system. Beyond this point the solution-based refinement is turned on.

Solution-based AMR in the off-body solver targets regions of high vorticity. A threshold vorticity is specified and any cell in the off-body mesh with vorticity larger than this specified value is marked for refinement. A conservative refinement approach is adopted in which all marked cells are guaranteed to be refined. However, due to the block-based nature of the scheme it also means that some cells that are not marked may also be refined. In practice, unmarked refined cells generally occur in close proximity to marked cells whose vorticity exceeds the threshold value so it is manifested as a buffer region of fine cells around the vortices.

The Cartesian grid system is constructed as a series of levels where the coarsest is level one and the finest is level n. In these studies, a six-level Cartesian grid system has finest level n mesh spacing of 20% C_{tip}, a seven-level grid has spacing of 10% C_{tip}, and an eight-level grid has a spacing of 5% C_{tip}. This multi-level structure is identical in both fixed and AMR off-body grids, and the main difference is that the fixed grid is generated by explicitly forcing refinement in pre-defined regions whereas the AMR grid is generated through the solver detecting regions where refinement is needed.

Figure 6 shows wake solutions for the adaptive cases with (a) seven and (b) eight levels of refinement, corresponding to a finest level resolution of 10% and 5% C_{tip}, respectively. Iso-surfaces of the Q-criterion at Q=0.00001 are shown colored by vorticity magnitude. It is evident that

	ncores	Unstruct		Cartesian		Total	
NSU3D Med	32	5.0M	5.34s	–	–	**5.0M**	**5.34s**
Helios Coarse	16	1.8M	3.65s	2.7M	1.17s	**4.5M**	5.10s
Helios Med 2ndO	32	2.8M	2.95s	15.6M	3.00s	**18.4M**	**6.11s**
Helios Med 5thO	32	2.8M	2.95s	15.6M	3.17s	**18.4M**	**6.27s**
Helios Fine	64	8.3M	4.40s	15.6M	1.65	**23.9M**	**6.65s**

Table 4. Computational statistics for isolated TRAM $\theta = 14^\circ$ fixed mesh study. NSU3D run on fully unstructured medium resolution mesh, Helios run on dual-mesh mesh with coarse, medium, and fine resolution. Linux Woodcrest cluster, 4 cores per node.

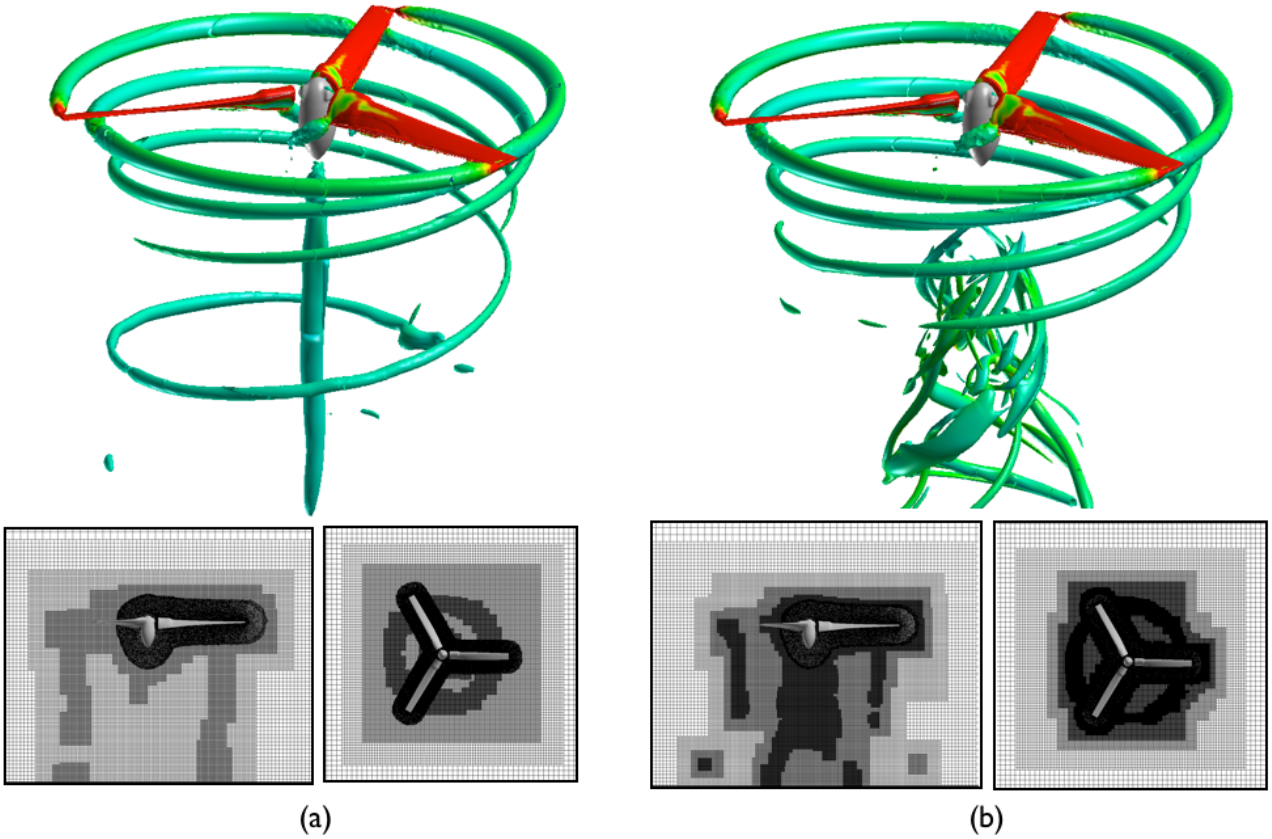


Fig. 6. Wake solution and mesh, $\theta = 14^\circ$ isolated TRAM calculation. Solution shows iso-surfaces of Q_{crit}. Helios off-body adaptive solution with (a) seven levels and (b) eight levels of adaptive refinement.

	C_T	C_Q	FM
Experiment	0.0149	0.00165	0.779
Coarse-Fixed-6lev	0.0148	0.00180	0.709
Coarse-AMR-7lev	0.0149	0.00181	0.711
Med-Fixed-7lev	0.0150	0.00178	0.734
Med-AMR-7lev	0.0151	0.00176	0.0747
Med-AMR-8lev	0.0152	0.00179	0.738
Fine-Fixed-7lev	0.0151	0.00171	0.764
Fine-AMR-7lev	0.0151	0.00171	0.768

Table 5. Performance of isolated TRAM rotor at $\theta = 14^\circ$ using fixed and adaptive (AMR) off-body meshes.

the adaptive scheme is properly targeting the wake effects. Tip vortex structures, as well as vortices shed from the hub, are maintained at high resolution well downstream from the blade plane.

A plot of the rotor wake vorticity pattern at the $\Psi = 0^\circ$ azimuth blade at the quarter-chord plane is shown in Fig. 7. Figure 7(a) shows the result from the fully unstructured NSU3D calculation, discussed in the previous section. Figure 7(b) shows the fixed-mesh Helios result with seven levels of refinement, and Fig. 7(c) shows the adaptive result with eight levels of refinement. The first vortex observed in this plane below the blade is shed from the $\Psi = 120^\circ$ blade, the second is shed from the $\Psi = 240^\circ$ azimuth blade, and the third is from the shown blade at a $\Psi = 360^\circ$ wake age. The fully-unstructured solution (Fig. 7(a)) shows significant dissipation in the first vortex, and also predicts the vortex location to be far below the blade. The second and third vortices are also highly dissipated and incorrectly located. Helios with fixed refinement (Fig. 7(b)) shows a significant improvement, capturing the first blade vortex more compactly and the second vortex reasonably well. However, by the third vortex the solution is quite dissipated. Helios with adaptive refinement (Fig. 7(c)) shows a significantly improved result, with all three vortices being captured nicely.

Table 5 shows the computed performance using adaptive meshes compared to the baseline fixed-mesh results shown in the previous section. The adaptive results give comparable performance to the fixed results of the same resolution, which is expected. Moreover, adding additional AMR levels do not appreciably improve the performance predictions. This suggests that the dominant factor in the accuracy of the performance calculation is the resolution of the near-body mesh, and not the off-body mesh. Thus, for the cases tested here the advantage in using AMR in the off-body mesh lies in the improved wake and tip vortex resolution and not in performance predictions. However, as shown below, AMR does enable reduced off-body mesh sizes.

Table 6 shows a comparison of the computational resources required for the adaptive cases. It should be noted that the off-body mesh size is continually changing in the adaptive calculation, the mesh size shown in the table is that recorded at the end of the solution. The time per step,

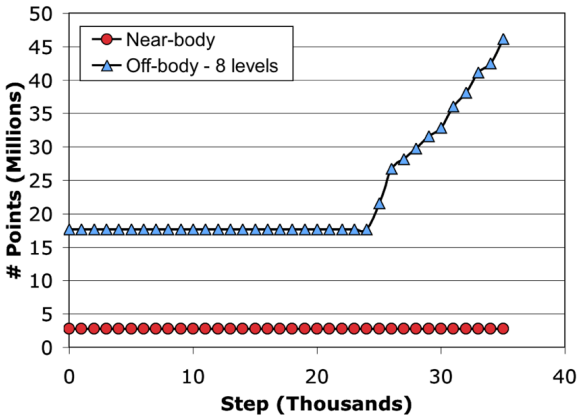


Fig. 8. Number of near and off-body points for Helios adaptive eight-level off-body calculation.

on the other hand, is taken as an average over the simulation. This is why, in some circumstances, the total points in the AMR mesh are greater than the comparable fixed mesh but the time per step is actually less. AMR is effective at reducing the problem size. Specifically, note the medium fixed eight-level case has 150.2M points in the off-body, which was too large to run on 32 cores of the parallel system. With AMR the eight-level case contains approximately 41M points, which is about a factor of four reduction, and which is now capable of fitting on the 32 core system.

It is interesting to note the problem size variance for the adaptive mesh calculation. Figure 8 shows a plot of the number of gridpoints vs. step for the 8-level medium adaptive calculation. An off-body mesh with 5% C_{tip} resolution is initially generated, targeting the near-body inter-grid boundary points and resulting in an initial adapted mesh of 17.7M points. The solution is run on this mesh without any solution-based adaptation for a number of steps to dissipate any non-physical startup effects. Then solution-based adaptivity is applied every 250 steps, approximately 40 adapt cycles total, to refine the mesh to regions of high vorticity, which correspond to tip vortices and vorticity shed from the hub (Fig. 6(c)). In practice we find similar trends for other problems. As the adaptive scheme detects regions of high vorticity and refines the grid around them, new regions are subsequently detected and refined. Typically, this process may be terminated by specifying a maximum number of off-body mesh points based on memory availability on the system. It should be noted, however, that there are likely applications, such as detecting ground effects, where it may be desirable to let the process continue to avoid premature termination (after insuring the availability of system resources).

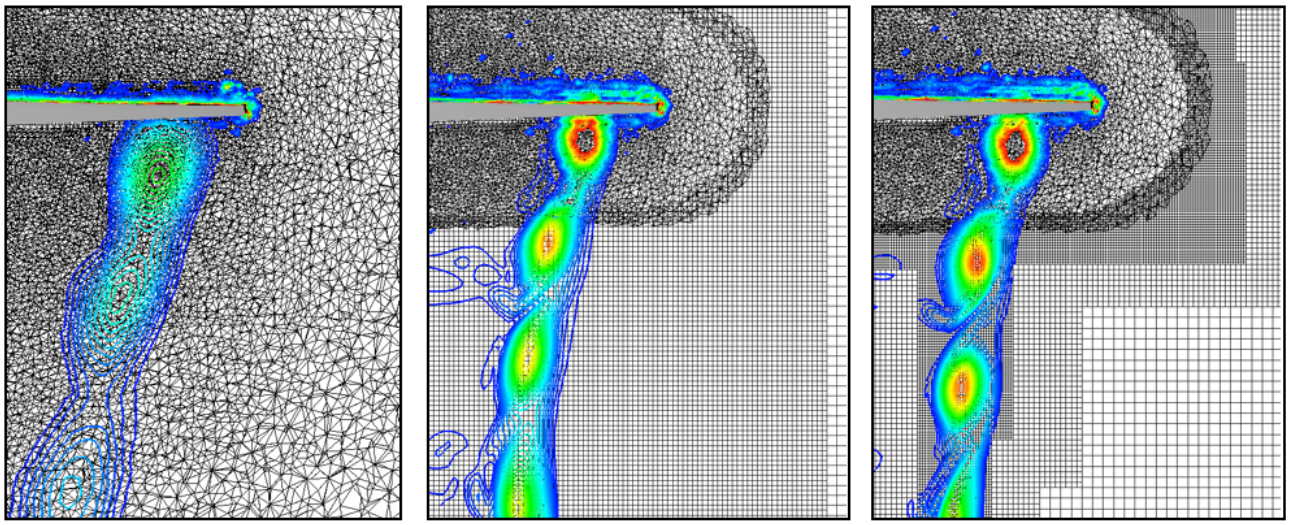


Fig. 7. Wake vorticity overlaid on mesh (a) NSU3D, (b) Helios with fixed seven-level off-body mesh, (c) Helios with adaptive eight-level off-body mesh.

	ncores	Unstruct		Cartesian		Total	
Coarse-Fixed-6lev	16	1.8M	3.65s	2.7M	1.17s	**4.5M**	**5.10s**
Coarse-AMR-7lev	16	1.8M	3.65s	19.3M	8.40s	**21.1M**	**12.31s**
Med-Fixed-7lev	32	2.8M	2.95s	15.6M	3.17s	**18.4M**	**6.27s**
Med-AMR-7lev	32	2.8M	2.95s	9.4M	2.13s	**12.2M**	**5.21s**
†Med-Fixed-8lev	32	2.8M	–	150.2M	–	**153.0M**	–
Med-AMR-8lev	32	2.8M	2.95s	22.7M	4.62s	**25.5M**	**7.89s**
Fine-Fixed-7lev	64	8.3M	4.40s	15.6M	1.65s	**23.9M**	**6.65s**
Fine-AMR-7lev	64	2.8M	4.40s	22.8M	1.63s	**25.6M**	**6.13s**

Table 6. Computational statistics for isolated TRAM $\theta = 14^\circ$ adaptive (AMR) mesh study. Helios dual-mesh with coarse, medium, and fine near-body meshes. Fixed mesh off-body times shown for reference. Coarse, medium, and fine near-body meshes applied with different levels of Cartesian refinement. (6 levels corresponds to 20%C_{tip} finest Cartesian resolution, 7 levels to 10%C_{tip}, and 8 levels to 5%C_{tip} resolution). Linux Woodcrest cluster, 4 cores per node.

	C_T	C_Q	FM
experiment	0.0149	0.00165	0.779
steady-fixed-6lev	0.01484	0.001802	0.709
iner-fixed-6lev	0.01485	0.001805	0.709
steady-AMR-7lev	0.01491	0.001811	0.711
iner-AMR-7lev	0.01489	0.001808	0.711

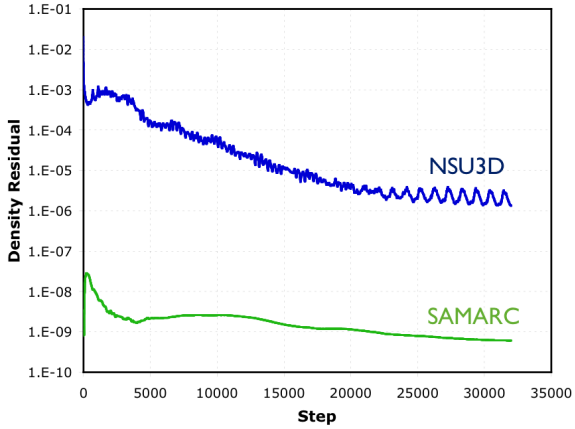
Table 7. Comparison of steady non-inertial ("steady") and unsteady inertial ("iner") computed loads for $\theta = 14^\circ$ collective. Fixed and adaptive off-body grids.

Inertial Hover

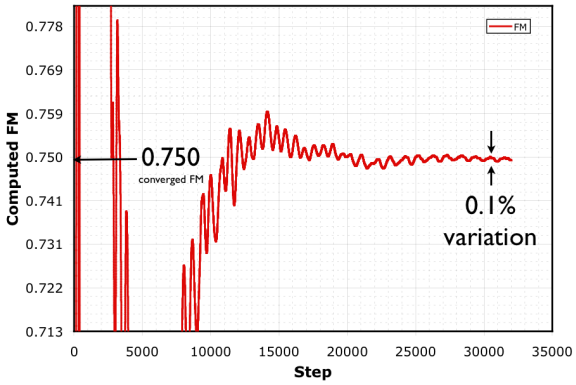
The steady hover assumption, which uses a fixed grid with rotational source terms applied to the equations solved on the grid, is accepted as a good approximation for isolated hover predictions. However, problems involving rotor-fuselage interactions, multiple rotors, or helicopters in forward flight which are ultimately of interest to helicopter engineers require an inertial formulation with moving grids capable of simulating bodies in relative motion. A useful validation is an investigation into whether the two formulations – steady hover vs. inertial hover – give the same answers. Further, it is of interest to investigate the behavior and performance of the AMR scheme for time-dependent moving grid problems.

The coarse near-body grid 1.8M nodes, discussed in previous sections, is used for this investigation. Two off-body Cartesian mesh systems are tested. The first is a six-level mesh with fixed refinement (20% C_{tip} at finest level) and the second is adaptive with seven levels (10% C_{tip} at finest level). Steady non-inertial calculations were typically run until forces oscillated by less than one-tenth of a percent. Inertial hover cases used a step size of 0.25 degrees azimuth and were run out to 6 revolutions. Computed loads are shown in Table 7. Steady and inertial results are shown on the two off-body grid systems.

A few interesting observations can be drawn from the results in Table 7. First, there is no noticeable difference between calculations using steady vs. inertial hover. This confirms that the steady approximation is appropriate for this problem and that the implementation of both the steady hover terms and the inertial formulation in Helios code is correct. Second, enhancing resolution in the off-body grid through mesh refinement does not significantly improve the load predictions in this case. The likely reason is that the primary source of the error is the coarse near-body grid. Similar behavior was observed in the earlier adaptive mesh results which used a finer near-body mesh (2.8M nodes). Third, the adaptive scheme works successfully for both steady and inertial cases, which is encouraging for its application to rotor-fuselage and forward flight applications.



(a) Residual convergence



(b) FM convergence

Fig. 9. Solution convergence characteristics for steady non-inertial fine dual-mesh calculation of $\theta = 10^\circ$ isolated TRAM rotor. (a) Residual convergence in near-body (NSU3D) and off-body (SAMARC) solvers, (b) FM force convergence history.

Collective Sweep

This section investigates the performance of the dual-mesh approach for a range of collective pitch θ settings. Earlier studies conducted on the fixed $\theta = 14^\circ$ condition found that calculations on the fine mesh, 8.34M nodes near-body with 7-level 15.6M node off-body, computed the most accurate performance numbers. Moreover, the results from the previous section indicated the steady non-inertial formulation gives essentially the same load predictions as the unsteady inertial formulation but at much lower cost. As a result, the collective sweep is run using the fine mesh with the steady hover formulation. The the $M_{tip} = 0.625$ condition is used with collective pitch θ settings of 6°, 8°, 10°, 12°, 14°, and 16° are tested.

All cases are converged until the equation residuals have dropped several orders of magnitude and the forces vary by a small degree. For example, Fig. 9 shows a sample convergence history for the $\theta = 10^\circ$ case. Solver residual convergence is shown in Fig. 9(a) while figure of merit FM

Collective	C_T	C_Q	FM
6°	0.00668	0.000591	0.654
9°	0.0086	0.000787	0.718
10°	0.0107	0.00104	0.750
12°	0.0129	0.00136	0.762
14°	0.0151	0.00171	0.764
16°	0.0173	0.00212	0.757

Table 8. Computed values of C_T, C_Q and FM as a function of collective pitch.

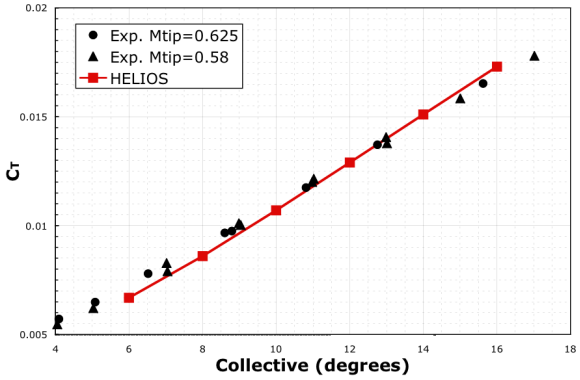


Fig. 10. Isolated TRAM collective sweep, thrust C_T vs. collective.

convergence is shown in Fig. 9(b). Although the residuals show some oscillatory behavior at the end of the solution, likely due to unsteadiness in the vortex flowfield, the forces converge to within a tenth of a percent of the final computed value.

Table 8 summarizes the loads and power predictions for all collective angles. Figure 10 shows the computed thrust coefficient C_T vs. collective angle. Figure 11 shows the variation of power C_Q with $C_T^{3/2}$. Experimental results from the $M_{tip} = 0.625$ and $M_{tip} = 0.58$ conditions are included in both plots for reference. The computed thrust closely matches experimental results while the computed torque slightly over-predicts experiment.

The variation in Figure of Merit FM with respect to thrust is plotted in Fig. 12. As a result of the over prediction in computed torque the figure of merit is under-predicted by 1% - 3%. The adaptive $\theta = 14°$ result obtained earlier with the same near-body mesh is included in the plot for reference. The differences between the fixed and adaptive solution are negligible. The figure also shows data predictions for a $\theta = 14°$ result obtained by Holst and Pulliam (Ref. 19) using the Overflow code. The data point labeled "med" corresponds to the two-level adapted mesh in that study, and the "fine" corresponds to the three-level mesh. Overall, the results are showing good agreement with both experiment as well as results obtained by other state-of-the-art CFD codes.

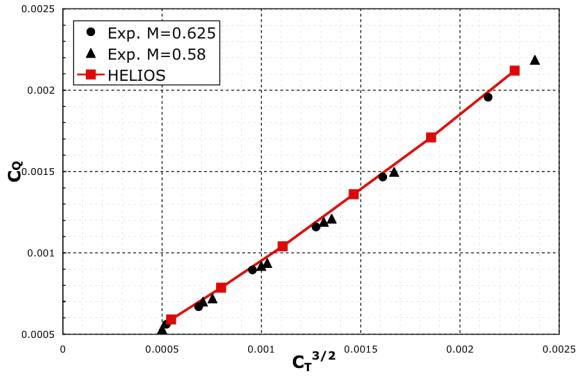


Fig. 11. Isolated TRAM collective sweep, power C_Q vs. $C_T^{3/2}$ for different collective angles.

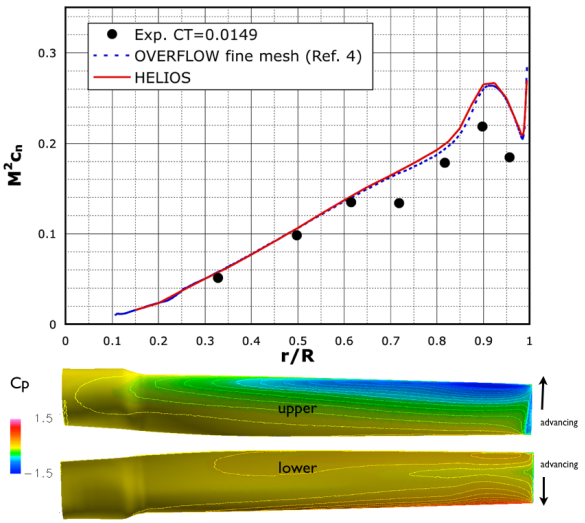


Fig. 13. Plot of blade section normal force at spanwise locations, with C_P on upper and lower surfaces of blade. $\theta = 14°$, $M_{tip} = 0.625$ condition, isolated TRAM rotor.

Figure 13 shows the integrated normal force at different spanwise locations for the $M_{tip} = 0.625$, $\theta = 14°$ condition. Results from Overflow taken from Potsdam and Strawn (Ref. 4) are included for comparison. The computed Helios results over-predict experimentally measured values but closely match the fine-mesh results obtained with Overflow. The flattening of the normal forces at the $\sim r/R = 0.7$ spanwise location is caused by the tip vortex passage under the blade. The fact that the computations do not show the significant bend in the curve in this region indicates that the tip vortex passing under the blade is excessively diffused. This agrees with findings in other works (Refs. 4,44). Further studies are needed to elucidate the underlying reasons for this discrepancy, in particular whether near-body mesh refinement, or off-body mesh refinement, or both, are needed to improve the predictions.

Despite the discrepancies, the Helios results are encour-

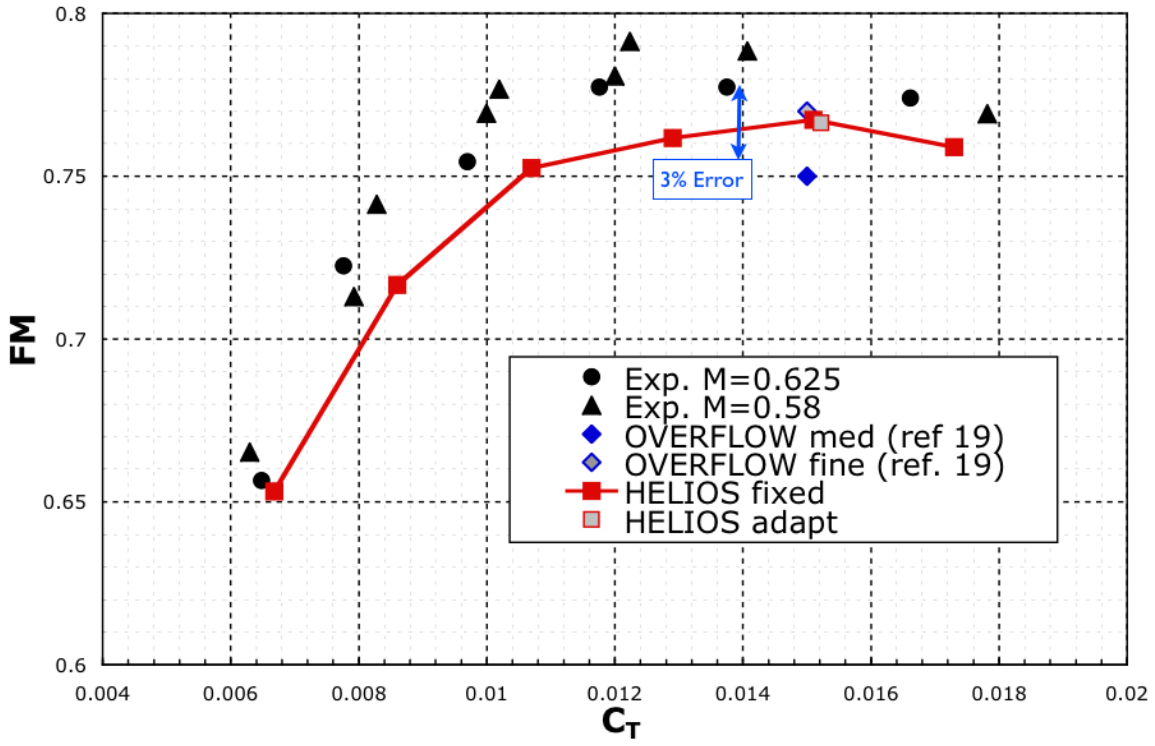


Fig. 12. Isolated TRAM collective sweep, figure of merit FM vs. thrust C_T. Arrow showing 3% variation from experimental results shown for reference.

aging on two fronts. First, they show it is possible to consistently predict FM to within 2-3% accuracy on a relatively modest 64 core computer system. Second, there is good evidence that the prediction accuracy is a function of the mesh resolution (see Table 3) and that further refinements will likely yield closer agreement. Holst and Pulliam (Ref. 19) report similar improvements in prediction accuracy as mesh resolution is enhanced.

CONCLUSIONS

An innovative dual-mesh CFD paradigm is developed and validated for hovering rotor computations. An unstructured RANS solver is applied to a near-body mesh in order to capture complex geometry and wall-bounded viscous effects, while a structured adaptive Cartesian Euler solver is applied to the off-body to resolve the rotor wake. The motivation for approach is to create a CFD solution methodology which exercises the best features of both solvers — i.e., body-conforming unstructured grids to capture the geometry and boundary layer effects, and efficient adaptive Cartesian grids with a high-order flow solver to resolve the wake. The code that employs this strategy is called "Helios".

In comparison to calculations with a fully unstructured mesh, the dual-mesh approach in Helios is able to achieve significantly better aerodynamic performance predictions for about the same cost. The cost savings arise because of the inherent efficiency of the Cartesian off-body solver, which means that significantly larger grids can be solved with only nominal increases in CPU time. Moreover, high-order solutions in the Cartesian off-body solver incurs negligible cost; e.g. the 5th-order algorithm is only about 6% more expensive than the standard 2nd-order scheme. Further savings arise because of the use of adaptive mesh refinement in the off-body, which is able to resolve the wake with one-fifth the number of grid points as a fixed-refined case with the same resolution.

Calculations with the steady hover assumption – fixed grid with rotational source terms applied to the equations solved on the grid – gives nearly identical results to a moving-grid (inertial) hover calculations, in which the near-body grid rotates while the background Cartesian grid remains fixed. Further, we have shown that the adaptive scheme was found to be robust for the moving-grid calculations and also gives comparable results. Using a fine 8.3M point near-body mesh, the dual-mesh scheme provides figure of merit predictions to within 1%–3% of experimental data for a range of collective angles. It is noteworthy that these results were obtained on a relatively modest cluster employing 64 cores.

Future work will investigate the dual-mesh approach for

fuselage-rotor and forward flight calculations. The AMR scheme currently targets refinement to regions of high vorticity. While this has been shown to work, it requires considerable experimentation to determine the correct threshold value of vorticity to trigger refinement. More automated schemes to target the location of tip vortices are currently being investigated.

Acknowledgments

Material presented in this paper is a product of the CREATE-AV Element of the Computational Research and Engineering for Acquisition Tools and Environments (CREATE) Program sponsored by the U.S. Department of Defense HPC Modernization Program Office. This work was conducted at the High Performance Computing Institute for Advanced Rotorcraft Modeling and Simulation (HIARMS). The ARC3DC high-order Cartesian solver was developed by Dr. Thomas Pulliam. The authors gratefully acknowledge additional contributions by Dr. Buvana Jayaraman, Dr. Anubhav Datta, and Dr. Roger Strawn.

References

[1]Datta, A. and Johnson, W., "An Assessment of the State-of-the-art in Multidisciplinary Aeromechanical Analyses," AHS International Specialists Conference on Aeromechanics, San Francisco, CA, January 2008.

[2]Wake, B. and Baeder, J., "Evaluation of a Navier-Stokes Analysis Method for Hover Performance Prediction," *Journal of the American Helicopter Society*, Vol. 41, (7), 1996.

[3]Ahmad, J. and Strawn, R., "Hovering Rotor and Wake Calculations with an Overset-Grid Navier-Stokes Solver," Proceedings of the American Helicopter Society 55th Annual Forum, Montreal, Canada, May 1999.

[4]Potsdam, M. and Strawn, R., "CFD Simulations of Tiltrotor Configurations in Hover," *Journal of the American Helicopter Society*, Vol. 50, (1), 2005, pp. 82–94.

[5]Bhagwat, M., Dimanlig, A., Saberi, H., Meadowcroft, E., Panda, B., and Strawn, R., "CFD/CSD Coupled Trim Solution for the Dual-Rotor CH-47 Helicopter Including Fuselage Modeling," , January 2008.

[6]Lakshminararyan, V. and Baeder, J., "High Resolution Computational Investigation of Trimmed Coaxial Rotor Aerodynamics in Hover," AHS International Specialists Conference on Aeromechanics, San Francisco, CA, January 2008.

[7]Lee-Rausch, E. M. and Biedron, R. T., "Simulation of an Isolated Tiltrotor in Hover with an Unstructured Overset-Grid RANS Solver," American Helicopter Society 65th Annual Forum, Grapevine, TX, May 2009.

[8]Buning, P. G., Jespersen, D. C., ad William H. Chan, T. H. P., Slotnick, J. P., Krist, S. E., and Renze, K. J., "OVERFLOW User's Manual, Version 1.8," Nasa langley research center, 1998.

[9]Hariharan, N. and Sankar, L., "High-Order Essentially Non-oscillatory Schemes for Rotary-Wing Wake Computations," *Journal of Aircraft*, Vol. 41, (2).

[10]Sankar, L., Yeshala, N., and Hariharan, N., "Application of Spatially High Order Adaptive Methods for Unsteady Flow over Rotary Wing Configurations," International Forum on Rotorcraft Multidisciplinary Technology, American Helicopter Society Specialists Meeting, paper No. 21-1, Seoul, Korea, October 2007.

[11]Yeshala, N., Egolf, A. T., Vasilescu, R., and Sankar, L., "Application of Higher Order Spatially Accurate Schemes to Rotors in Hover," AIAA Paper No. 2006-2818, 24th AIAA Applied Aerodynamics Conference, San Francisco, CA, June 2006.

[12]Duque, E., Sankar, L., Menon, S., Bauchau, L., Ruffin, S., Smith, M., Ahuja, K., Brentner, K., Long, L., Morris, P., and Gandhi, F., "Revolutionary Physics-based Design Tools for Quiet Helicopters," AIAA Paper 2006-1068, 44th AIAA Aerospace Sciences Meeting.

[13]Nielsen, E., Diskin, B., and Yamaleev, N., "Discrete Adjoint- Based Design Optimization of Unsteady Turbulent Flows on Dynamic Unstructured Grids," AIAA-2009-3802, 19th AIAA Computational Fluid Dynamics Conference, San Antonio, TX.

[14]Strawn, R. C. and Barth, T. J., "A finite-volume Euler solver for computing rotary-wing aerodynamics on unstructured meshes," *Journal of the American Helicopter Society*, Vol. 38, 1993, pp. 61–67.

[15]Kang, H. J. and Kwon, O. J., "Unstructured Mesh Navier-Stokes Calculations of the Flow Field of a Helicopter Rotor in Hover," *Journal of the American Helicopter Society*, April 2002, pp. 90–99.

[16]Potsdam, M. and Mavriplis, D. J., "Unstructured Mesh CFD Aerodynamic Analysis of the NREL Phase VI Rotor," AIAA-2009-1221, 47th AIAA Aerosciences Meeting, Orlando FL, Jan 2009.

[17]Meakin, R., "Automatic Off-body Grid Generation for Domains of Arbitrary Size," AIAA-2001-2536, 15th AIAA Computational Fluid Dynamics Conference, Anaheim CA, June 2001.

[18]Holst, T. and Pulliam, T., "Overset Solution Adaptive Grid Approach Applied to Hovering Rotorcraft Flows," AIAA-2009-3519, 27th AIAA Applied Aerodynamics Conference, San Antonio, TX.

[19]Holst, T. and Pulliam, T., "Optimization of Overset Solution Adaptive Grids for Hovering Rotorcraft Flows," 2010 AHS Specialists Meeting on Aeromechanics, San Francisco CA.

[20]Vasilescu, R., Yeshala, N., Sankar, L., and Egolf, A., "Structured Adaptive Mesh Refinment (SAMR) Algorithms Applied to Rotor Wake Capturing," American Helicopter Society 63rd Annual Forum, Virginia Beach, VA, May 2007.

[21]Dietz, M., Karmer, E., and Wagner, S., "Tip Vortex Conservation on a Main Rotor in Slow Descent Flight Using Vortex-Adapted Chimera Grids," AIAA-2006-3478, 24th AIAA Applied Aerodynamics Conference, San Francisco, CA, June 2006.

[22]Park, Y. M. and Kwon, O. J., "Simulation of Unsteady Rotor Flow Field Using Unstructured Adaptive Sliding Meshes," *Journal of the American Helicopter Society*, October 2004, pp. 391–400.

[23]Nam, H., Park, Y. M., and Kwon, O. J., "Simulation of Unsteady Rotor-Fuselage Aerodynamics Interaction Using Unstructured Adaptive Meshes," *Journal of the American Helicopter Society*, April 2006, pp. 141–149.

[24]Mavriplis, D. and Venkatakrishnan, V., "A Unified Multigrid Solver for the Navier-Stokes Equations on Mixed Element Meshes," *International Journal for Computational Fluid Dynamics*, Vol. 8, 1997, pp. 247–263.

[25]Hornung, R. D., Wissink, A. M., and Kohn, S. R., "Managing Complex Data and Geometry in Parallel Structured AMR Applications," *Engineering with Computers*, Vol. 22, (3-4), 2006, pp. 181–195.

[26]Pulliam, T. H., "Study to Eliminate Ground Resonance Using Active," Utsi computational fluid dynamics users workshop, March 1984.

[27]Sitaraman, J., Floros, M., Wissink, A., and Potsdam, M., "Parallel Unsteady Overset Mesh Methodology for a Multi-Solver Paradigm with Adaptive Cartesian Grids," AIAA-2008-7117, 26th AIAA Applied Aerodynamics Conference, Honolulu, HI, June 2008.

[28]Wissink, A., Sitaraman, J., Sankaran, V., Mavriplis, D., and Pulliam, T., "A Multi-Code Python-Based Infrastucture for Overset CFD with Adaptive Cartesian Grids," AIAA-2008-0927, 46th AIAA Aerosciences Conference, Reno, NV, Jan 2008.

[29]Sitaraman, J., Katz, A., Jayaraman, B., Wissink, A., and Sankaran, V., "Parallel Unsteady Overset Mesh Methodology for a Multi-Solver Paradigm with Adaptive Cartesian Grids," AIAA-2008-0660, 46th AIAA Aerosciences Conference, Reno, NV, January 2008.

[30]Mavriplis, D. J., "Multigrid Strategies for Viscous Flow Solvers on Anisotropic Unstructured Meshes," *Journal of Computational Physics*, Vol. 145, (1), September 1998, pp. 141–165.

[31]Mavriplis, D. J. and Pirzadeh, S., "Large-Scale Parallel Unstructured Mesh Computations for 3D High-Lift Analysis," *AIAA Journal of Aircraft*, Vol. 36, (6), 1999, pp. 987–998.

[32]Mavriplis, D. J., "Results from the Third Drag Prediction Workshop Using the NSU3D Unstructured Mesh Solver," AIAA-2007-0256, 45th AIAA Aerosciences Conference, Reno, NV, January 2007.

[33]Yang, Z. and Mavriplis, D. J., "Higher-order Time Integration Schemes for Aeroelastic Applications on Unstructured Meshes," AIAA-2006-0441, 44th AIAA Aerosciences Conference, Reno, NV, January 2006.

[34]Mavriplis, D. J., Aftosmis, M., and Berger, M., "High-Resolution Aerospace Applications using the NASA Columbia Supercomputer," Proceedings of Supercomputing 2005, Seattle, WA, November 2005.

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

[36]Berger, M. and Colella, P., "Local Adaptive Mesh Refinement for Shock Hydrodynamics," *Journal of Computational Physics*, Vol. 82, 1989, pp. 65–84.

[37]Hornung, R. D., Kohn, S. R., and Wissink, A. M., "Managing Application Complexity in the SAMRAI object-oriented framework," *Concurrency and Computation: Practice and Experience*, Vol. 14, 2002, pp. 347–368.

[38]Wissink, A. M., Hornung, R. D., Kohn, S. R., Smith, S., and Elliott, N., "Large-Scale Parallel Structured AMR Calculations using the SAMRAI Framework," Proceedings of Supercomputing 2001 (SC01), Denver, CO, November 2001.

[39]Benek, J., Buning, P., and Steger, J., "A 3-D chimera grid embedding technique," AIAA Paper 85-1523, 1985.

[40]Steger, J. L. and Benek, J. A., "On the use of composite grid schemes in computational aerodynamics," *Computer Methods in Applied Mechanics and Engineering*, Vol. 64, (1–3), 1986, pp. 301–320.

[41]Noack, R., "DiRTlib: A Library to Add an Overset Capability to Your Flow Solver," AIAA Paper 2005-5116, 17th AIAA Computational Fluid Dynamics Conference, Toronto, Canada, June 2005.

[42]Noack, R., "SUGGAR: a General Capability for Moving Body Overset Grid Assembly," AIAA Paper 2005-5117, 17th AIAA Computational Fluid Dynamics Conference, Toronto, Canada, June 2005.

[43]Marcum, D. L., "Advancing-Front/Local-Reconnection (AFLR) Unstructured Grid Generation," Computational Fluid Dynamics Review, World Scientific-Singapore, 1998.

[44]Strawn, R. and Djomehri, M. J., "Computational Modeling of Hovering Rotor and Wake Aerodynamics," American Helicopter Society 57th Annual Forum, Washington, DC, May 2001.