

An Overset Dual-Mesh Solver for Computational Fluid Dynamics

Andrew Wissink

U.S. Army Aeroflightdynamics Directorate
Aviation and Missile Research, Development, and Engineering Center (AMRDEC)
Ames Research Center, USA

Abstract: A dual-mesh unstructured/adaptive Cartesian approach is presented to better resolve the wake in traditional time-dependent unstructured calculations about complex geometries. An unstructured RANS solver is applied near the body surface to capture near-wall viscous boundary layer effects. Away from the wall a high-order adaptive Cartesian solver resolves the wake. An overset procedure facilitates data exchange between the two mesh types as well as enables relative motion between the mesh systems - i.e. the near-body unstructured meshes can move and deform inside the stationary adaptive Cartesian off-body grid system. The key advantage of this approach is that it enables use of traditional unstructured solvers to resolve geometrically-complex configurations while enabling the wake to be resolved through high-order Cartesian AMR. The scheme is demonstrated for several problems, including flow shed over a bluff body, tip vortices from a wing at angle of attack, and rotary-wing flows.

Keywords: Adaptive Mesh Refinement, Cartesian, High-Order, Rotorcraft

1 Introduction

Vortical flows can have a profound impact on control, vibration, and structural integrity of aerodynamic vehicles. Tip vortices generated by wings on a large aircraft have been known to affect other aircraft following at a close distance. Tip vortices emanating from the nose and swept wing of high angle-of-attack fighter jets create tail buffet [1] and loss of control during maneuvers. Rotary-wing vehicles experience both tail buffet from vortices shed from the hub as well as blade-tip vortex interaction that greatly effect handling qualities, vibration, and noise [2]. In the worst case, vortex flows can lead to catastrophic flight conditions like vortex ring state where the rotor experiences a sudden loss of lift when it becomes enveloped in its own vortex wake.

Vehicles in use today did not have the benefit of high fidelity computational tools to predict the vortex-dominated flow phenomena they create. Adverse effects are typically discovered during windtunnel tests when redesign is generally difficult and expensive, or in the worst case during flight tests when catastrophic loss of control have historically led to loss of life. Computational tools that can effectively predict these effects at early stages of the vehicle design are an important resource for cost effective and safer vehicle designs in the future.

In rotorcraft design a 1% change in figure of merit of a rotor, a measure of rotor efficiency, can translate to 200-300 lbs. change in useful load. Being able to predict performance to this level of accuracy is highly desired by rotor designers but is generally not possible with computational tools available today for new rotor systems that have not already been windtunnel tested. The introduction of parallel high performance computing (HPC) systems has led to increases in computing power by a factor of 1000X in the past decade (according to the published list of top 500 HPC systems in the world today). With the growing availability

Approved for public release; distribution unlimited. Review completed by the AMRDEC Public Affairs Office 15 Mar 2010; FN4487.

of relatively cheap computing power it may be possible to achieve this desired level of fidelity through computational methods rather than full scale tests.

Reynolds Averaged Navier Stokes (RANS) schemes, once considered too expensive for engineering analysis, are now used routinely for aerodynamic load prediction. A RANS model handles viscous, compressible, and rotational flow implicitly and use meshes generated directly from CAD representations to accurately represent geometrically complex features. The availability of relatively cheap and fast HPC computing resources has prompted efforts to resolve vortex wakes with RANS alone using fine grid systems in the wake. In 2003 Strawn and Djomehri [3] applied a very fine uniform Cartesian mesh to predict wakes from the HART-II rotor tests, running on cutting-edge HPC computer systems with many processors. Pulliam [4] later took a similar approach to predict the wake from the TRAM rotor, using as many as 4 Billion grid-points in calculations running on up to 4000 processors. Both found that their very-fine grid calculations demonstrated improvement over more routine calculations run on coarser wake meshes, but the improvement was marginal at best. The reason is likely that the rotorcraft wake is still significantly under-resolved even with their large mesh systems.

A technology that has proven effective for resolving fine-scale structures in a variety of other science and engineering applications is adaptive mesh refinement (AMR). AMR is a process in which the local mesh resolution is automatically increased to resolve important fine scale features. Many of the ideas were originally developed for shock hydrodynamics problems [5, 6, 7, 8] but the technology has since been extended to a wide range of other applications including astrophysics [9], flow in porous media [10], reactive flows [11], geographic-scale flows [12], and flow in the heart [13]. It seems natural to extend this technology to aerodynamic wakes.

Similar to fixed wing applications, resolving viscous flow in rotorcraft around the rotor and fuselage surfaces is important in order to achieve the correct prediction of lift and drag as well as other important phenomenon like separation. Rotorcraft also tend to be some of the most geometrically complex vehicles to model. For these reasons, an unstructured solver near the wall surface. The use of unstructured grids permits grid generation around the precise geometry imported directly from engineering CAD models, so there is very little geometric error. The flowfield near a wingtip has been shown both numerically and experimentally to be turbulent with regions of local separation [14, 15]. Thus, a high-fidelity RANS solver with appropriate turbulence models is applied in this near-body region, ensuring proper resolution of the physics that govern the initiation of the vortex field. In the wake region, where numerical dissipation becomes more of a problem, we apply a block structured Cartesian solver with high-order spatial difference algorithms and adaptive mesh refinement.

The CFD approach described in this paper is used as the basis for the aerodynamics calculations in the U.S. Army’s Helios software [16, 17, 18], which is the rotary-wing product of the CREATE-AV (air vehicles) program [19]. Helios contains interfaces to structural dynamics and trim which govern the mesh motion and deformations, but in this paper we concentrate solely on the CFD formulation. Further details describing the application of the dual mesh approach in Helios to rigid rotors in hover [20, 21], deforming rotors in forward flight [22, 23], and rotor-fuselage configurations [24, 25, 26, 27] can be found in the noted references.

The remainder of the paper discusses details of the dual-mesh implementation and, in particular, focuses on the use of the AMR off-body solver to resolve vortex wakes. The paper is organized as follows: The next section discusses further details of the hybrid dual-mesh computational approach. The following section shows some results using the method to resolve wakes of bluff bodies, wingtip vortices, and rotary-wing tip vortices. The last section summarizes the main conclusions of the work and suggests directions for future work.

2 Computational Approach

The spatial discretization scheme employs an overset near-body/off-body approach, using unstructured body-fitted grids near the body surface and adaptive Cartesian grids away from the surface (Fig 1). The mixed-element unstructured near-body mesh is cut a certain distance from the wall. The RANS solver NSU3D [28] by Mavriplis is applied to this subsetted unstructured near-body mesh, although the approach is general and could accommodate any general unstructured solver.

Data is transferred from the near-body mesh to the background Cartesian mesh using standard sec-

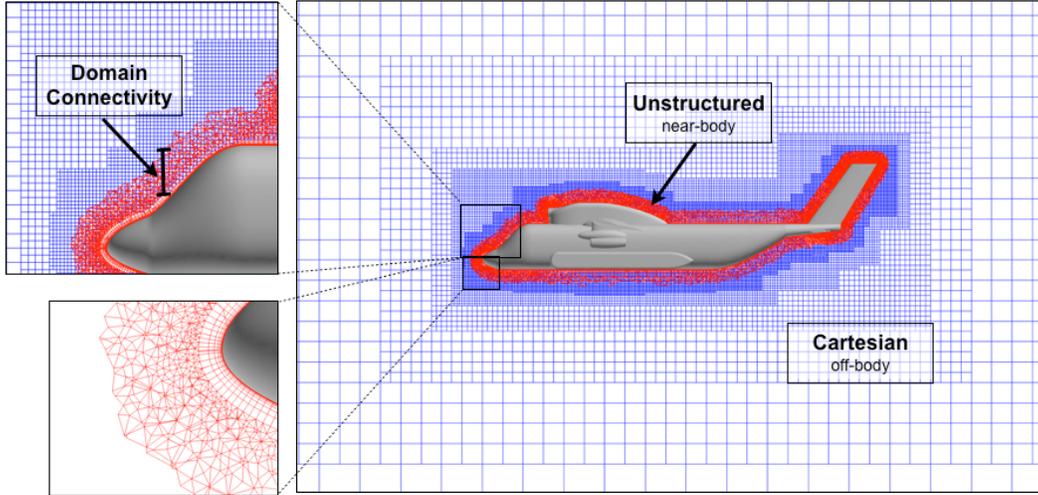


Figure 1: Near-body/off-body overset grid system.

ond order interpolations used commonly for both structured [29, 30] and unstructured [31, 32, 33] overset applications. The PUNDIT code [34] by Sitaraman manages the chimera grid hole cutting and interpolation.

2.1 Block Structured Cartesian

Block-structured Cartesian grids are used for domain coverage in the off-body region. There are several advantages Cartesian grids present over typical tetrahedral elements used for the wake 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 FLOP counts 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 INTs), the lower and upper indices, and the level of refinement. In total, only 7 INTs 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 vertex and/or edge locations and requires *millions* of REALS 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 boundaries of the near-body unstructured mesh, as well as to desirable features in the wake.

2.2 Cartesian AMR

Unstructured AMR has been used extensively in aerospace applications for local refinement of unstructured tetrahedral-based meshes. Mavriplis [37] and Park [38] developed locally-adaptive schemes for CFD solutions on unstructured general element meshes. Potsdam [39] applied the same technology for wake resolution of wind turbine predictions. Unstructured Cartesian AMR is also commonly used. Aftosmis et al. [40] have shown impressive results using unstructured Cartesian AMR for very complex geometries and flowfields. Buning and Pulliam [41] developed an AMR scheme for the off-body block-structured Cartesian portion of the Overflow code.

Many of the aforementioned efforts target steady problems in which adaptivity is applied as a form of mesh post-processing. That is, a solution is computed on an initial mesh, the mesh is adapted according to

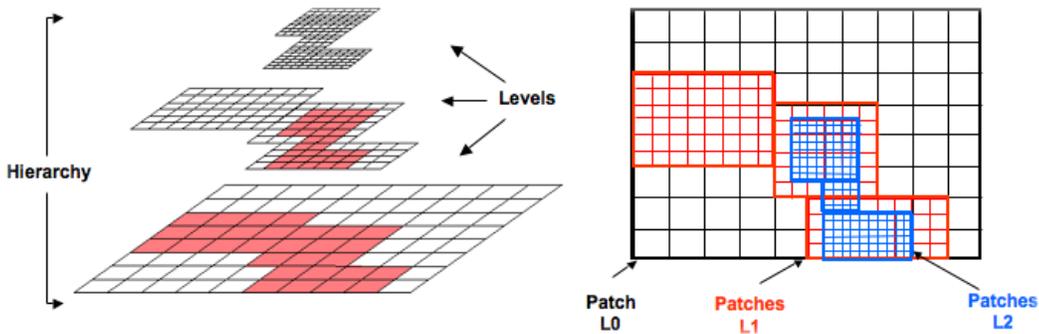


Figure 2: Block structured AMR grid composed of a hierarchy of nested levels of refinement. Each level contains uniformly-spaced logically-rectangular regions, defined over a global index space.

features in the solution, the flow solver is restarted using the new mesh to compute a solution. The process is repeated as many times as necessary, typically through several adaptation cycles. A notable exception is the scheme used in Overflow [41] which adapts in a time-accurate fashion.

In this work we are targeting use of AMR for unsteady wake flowfields. This requires a somewhat different approach than AMR for steady problems because the scheme must be able to adapt in a time dependent fashion. For steady problems it is generally satisfactory to refine the mesh a few times. For unsteady applications, mesh refinement and coarsening are performed continuously throughout the solution, meaning the mesh is adapted tens to hundreds of times over the course of the simulation. A tight integration is therefore required between the mesh refinement scheme and the flow solver in order to make this process efficient. Unsteady AMR also requires careful consideration of parallel computing issues like re-load balancing and re-establishing data communication after adaptation.

The structured adaptive mesh refinement (SAMR) strategy used for the Cartesian off-body solver is based on the ideas of Berger, Colella, and Olinger [5, 42]. 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. Tests of time-dependent adaptive structured AMR calculations in which the grid is adapted frequently (every other time step) have shown parallel scaling to over 1000 processors [43, 44].

An advantage of the SAMR paradigm is that it facilitates a clean separation between grid- and solution-based operations. Several infrastructures, for example Chombo [47], GrACE [45], and PARAMESH [46], have been developed to support parallel SAMR solutions. The way each of these is formulated is the infrastructure manages grid-based operations – e.g. adaptive grid generation, parallel decomposition, data exchange between blocks, etc. – while the user constructs a 3D single block solver that runs serially on each block. The particular infrastructure we have adopted for the off-body solution is SAMRAI [48, 49, 50] 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. The single block solver applied on each block is a high-order Cartesian version of NASA’s ARC3D code, as discussed next.

2.3 High Order Algorithms

High-order algorithms have proven effective for CFD modeling of rotorcraft wakes, see Hariharan and Sankar [51, 52]. Implementing SAMR versions of high-order schemes is straightforward. Once each grid block is generated, it is passed to a solver that has been optimized for high-order operations on a 3D structured isotropic Cartesian grids. The particular solver we use is ARC3DC, a version of the ARC3D [54, 53] of Pulliam at NASA Ames with high-order algorithms 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 explicit Runge-Kutta scheme is used for time integration scheme in ARC3D. 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. All parallel communication for these operations is managed by SAMRAI.

High-order algorithms are very efficient on structured Cartesian meshes. The 5th-Order scheme discussed above is only about 20%-30% more expensive than a standard second order scheme and requires little additional memory. Contrast this with high order Discontinuous Galerkin (DG) schemes implemented on tetrahedral elements, which tend to be about an order of magnitude more expensive than standard second order schemes.

3 Results

Results with the coupled RANS-Cartesian AMR approach are demonstrated to resolve shed wakes for three problems. This geometrically simple problem demonstrates the ability of the scheme to resolve unsteady wakes from bluff bodies. The second case involves tip vortices shed from a NACA0015 wing at angle of attack, demonstrating the ability of the scheme to capture and maintain the tip vortices. Lastly, we resolve the wake of a quarter-scale V-22 rotor blade in hover conditions.

3.1 Sphere

The physical characteristics of unsteady flow over a sphere, such as onset of instabilities and shedding frequency at different Reynolds numbers, are well known and documented both experimentally and computationally. The wealth of validation data available makes this problem useful to evaluate the accuracy of RANS-Cartesian approach.

A near-body sphere mesh used for the RANS-Cartesian solution is constructed by trimming a standard unstructured prism-tet mesh. The trim distance is one half of the sphere diameter and the RANS solver is applied to this near-body mesh. The off-body Cartesian off-body mesh extends into the far field, using a maximum of 7 levels of refinement. It adapts to match the mesh spacing at the interface to the near-body mesh and adapts time-dependently to regions of high vorticity throughout the simulation.

A calculation at three time instances at Re 800 is shown in Fig. 3. The calculation was performed on 32 processors of an SGI-based Linux cluster and ran in about a day. The flow at Re=800 is unsteady but laminar, so no turbulence model is used in the near-body solver. Experimental results have demonstrated transition from large scale to small scale structures at Re 800. The computational results pick up this

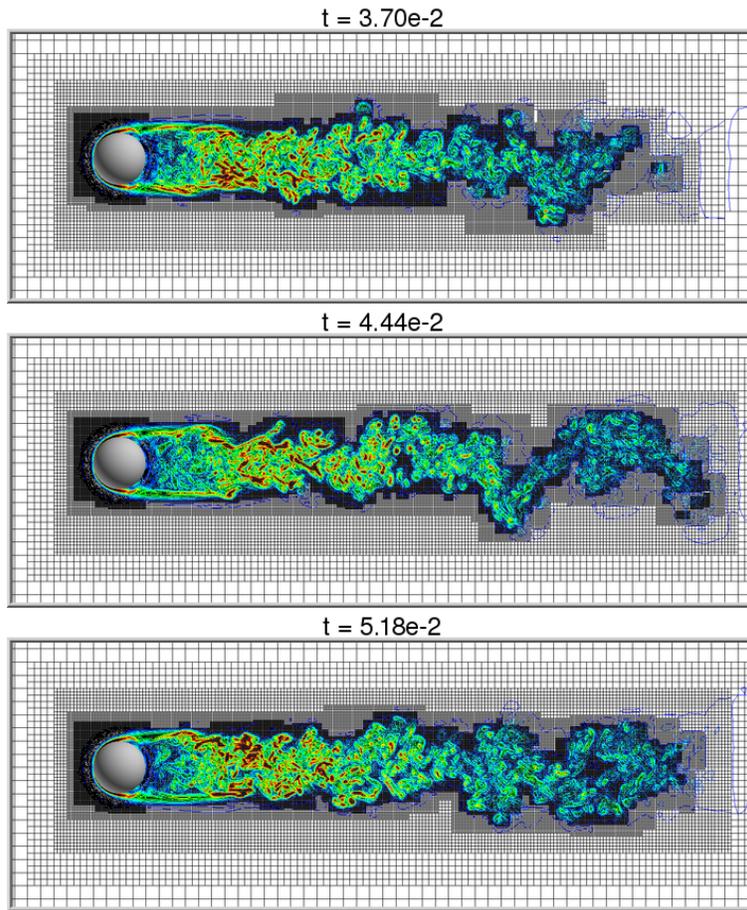


Figure 3: Unsteady shedding over sphere at Re 800 using RANS-Cartesian. Vorticity contours at different solution times. Cartesian meshes adapt to regions of high vorticity magnitude.

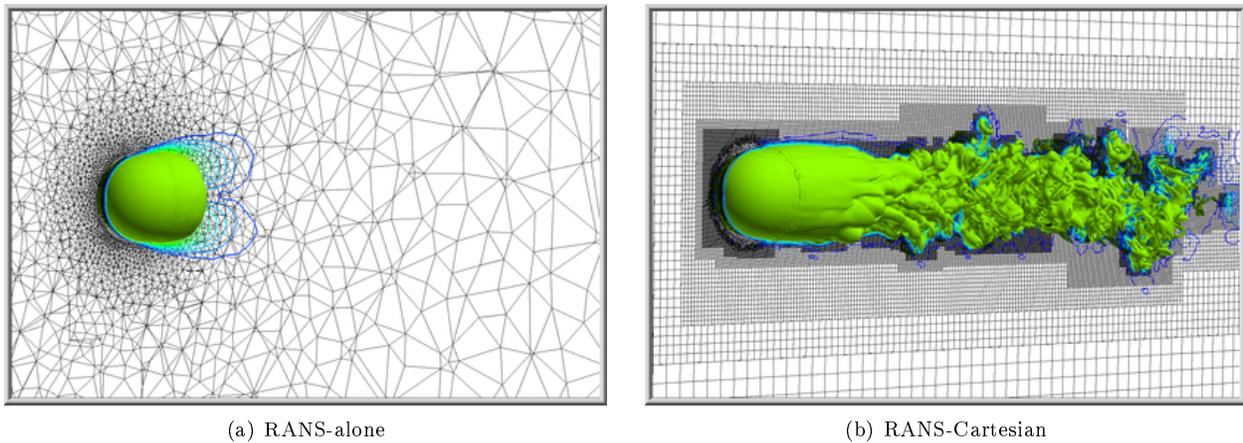


Figure 4: Wake resolution comparison, flow over sphere at Re 800. Iso-surface of vorticity overlaid on mesh.

	RANS alone	RANS-Cartesian
Gridpoints	4.79M	14.41M 4.51M Unstructured 9.90M Cartesian
Computation Time/Step	6.53 sec	7.39 sec 6.15s RANS (83%) 1.24s Cartesian (17%)

Table 1: NACA 0015 problem size and computational performance, 32 processor SGI Altix system.

phenomenon. Studies of this case presented elsewhere [36] with this solution approach show proper capture of the separation point in the steady regime at $Re < 160$.

For comparison purposes, Figure 4 shows an iso-surface of vorticity of the wake computed using RANS-alone on a purely unstructured mesh compared to the RANS-Cartesian approach. The RANS-alone solution captures the near-field effects well but dissipates the vorticity quickly, causing the wake solution to appear steady. The RANS-Cartesian approach, on the other hand, properly captures the unsteady shedding behavior.

The unstructured mesh in Fig. 4(b) is not refined to capture the wake and the solution quality could be likely be improved if it were. However, there are two drawbacks to static grid refinement performed during grid generation. First, adding refinement manually requires knowledge by the person generating the grid to know where to add points to resolve the wake. Clearly, it is advantageous from both an automation and an accuracy standpoint for the solver to make this decision as the solution evolves rather than putting the burden on the grid generator. Second, as is demonstrated in the next section, there is a significant computational cost savings in using structured Cartesian grids over tets in the wake region.

3.2 NACA 0015 Wing

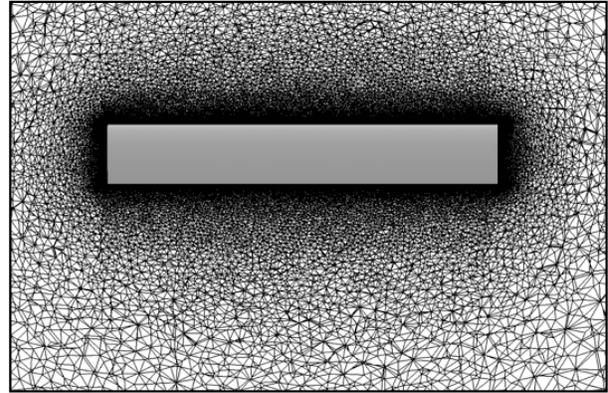
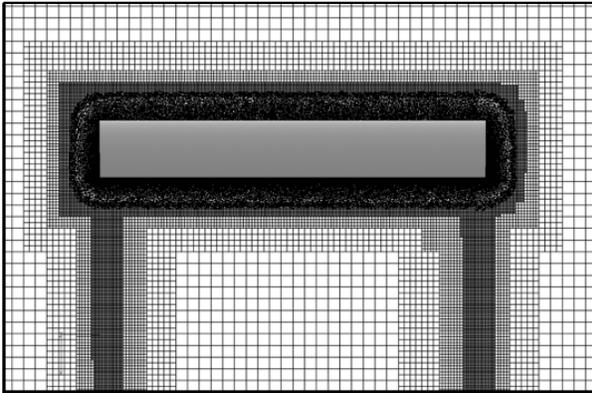
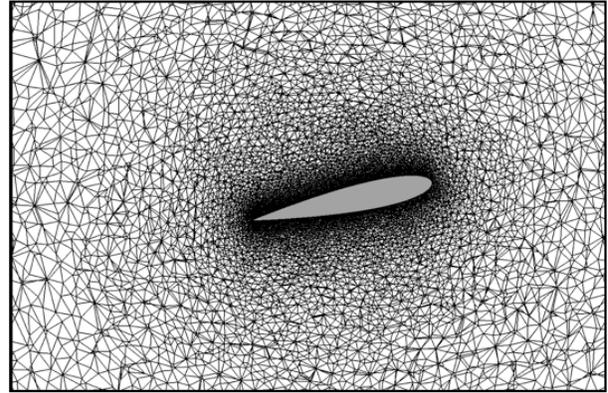
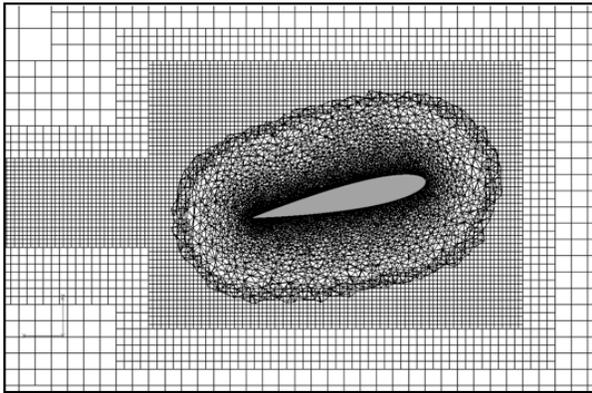
The next test case involves steady flow around a flat-tipped NACA0015 wing at 12° angle of attack at Mach number 0.1235 with Reynolds number 1.5 Million. This case was studied experimentally by McAlister and Takahashi [56] in the 7'x10' windtunnel at NASA Ames in 1991. Computational results have been presented by Sitaraman and Baeder [57] and by Hariharan and Sankar [58]

As with the sphere case presented previously, we show comparisons of the wake using the RANS-Cartesian approach vs. RANS-alone. Figure 5(a) shows the fully unstructured RANS-alone mesh while Fig. 5(b) shows the RANS-Cartesian mesh. The near-body mesh is a subsetted version of the RANS-alone mesh, trimmed a distance of 0.5 chords from the surface. This case experiences turbulent flow near the surface so the Spalart-Allmaras one-equation turbulence model is applied (on the near-body mesh only for the RANS-Cartesian calculation). The RANS-Cartesian solution applies refinement to regions of high vorticity to capture wingtip vortices.

Figure 6 shows an iso-surface of vorticity at $\omega = 0.25$ for the RANS-Cartesian and RANS alone solutions. In the near-vicinity of the wing surface the solutions for the two approaches are essentially the same, which is to be expected since the same unstructured mesh and solver are applied in both cases. Downstream of the wing, however, there is a major difference in the ability of the two schemes to capture the tip vortices emanating from the wingtips. The wing tip vortices dissipate very quickly in the RANS alone solution but the RANS-Cartesian solver is able to resolve the tip vortices well back from the trailing edge. It captures the breakdown into sub-structures happening approximately 15 chords behind the wing.

Experimental measurements by McAlister measured the vertical velocity components of the wingtip vortices downstream at 2, 4 and 6 chords. Figure 7 shows plots of computed vs. experimental V_Z at these locations and, additionally, at 12 chords where there were no measurements so only computed results are available. The adapted Cartesian grid used in the RANS-Cartesian solution does a much better job at preserving the vortex strength than the RANS-alone solution.

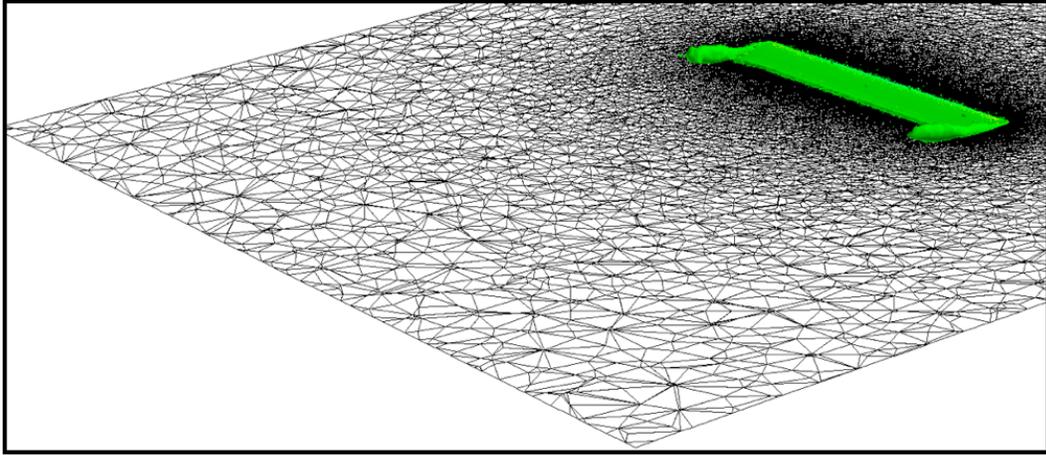
The computational cost incurred by the refined Cartesian meshes is minimal relative to the cost of the unstructured solver. Consider the problem size and performance statistics for this calculation shown in Table 1. Although the RANS-Cartesian calculation uses almost triple the number of gridpoints of the RANS



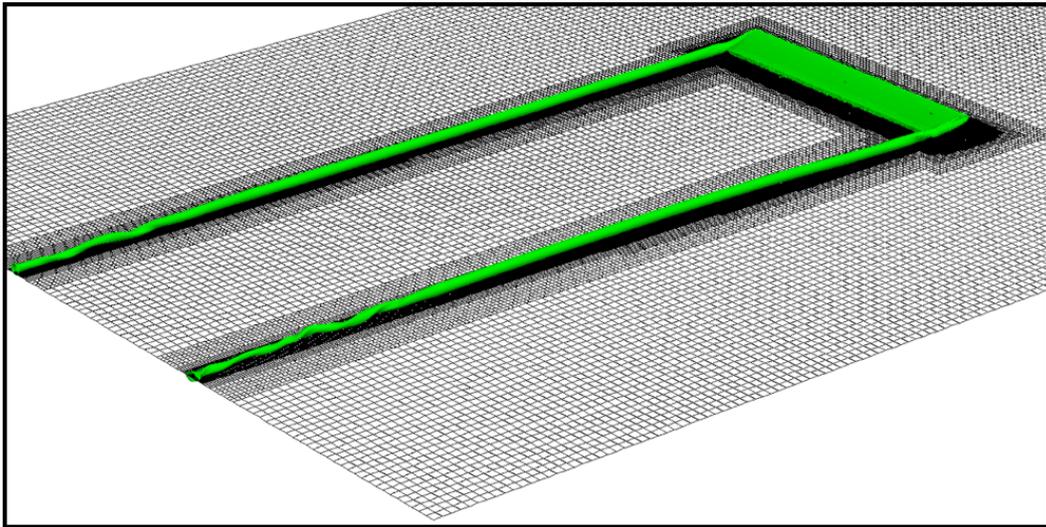
(a) RANS-Cartesian

(b) RANS Alone

Figure 5: Mesh systems for NACA 0015 wing calculations (a) unstructured everywhere; (b) near-body unstructured with off-body adaptive Cartesian



(a) RANS-alone



(b) RANS-Cartesian

Figure 6: NACA 0015 calculation, iso-surface of vorticity showing downstream tip vortices. (a) unstructured grid with no refinement; (b) adaptive Cartesian grid refined to regions of high vorticity.

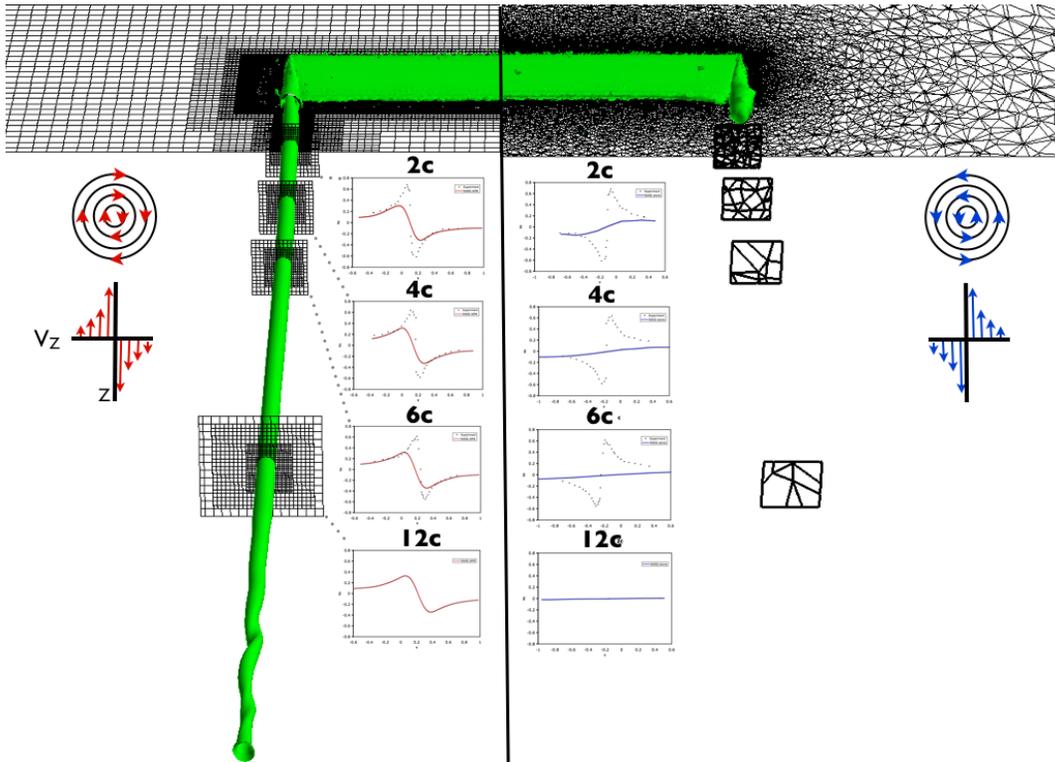


Figure 7: Computed vorticity iso-surface around NACA 0015 wing with RANS near-body with high-order adaptive Cartesian off-body grids (left) versus RANS only (right). Plots of computed and experimentally-measured vertical velocity at 2, 4, and 6 chords downstream. Computed solution only at 12 chords (no experimental data at this location).

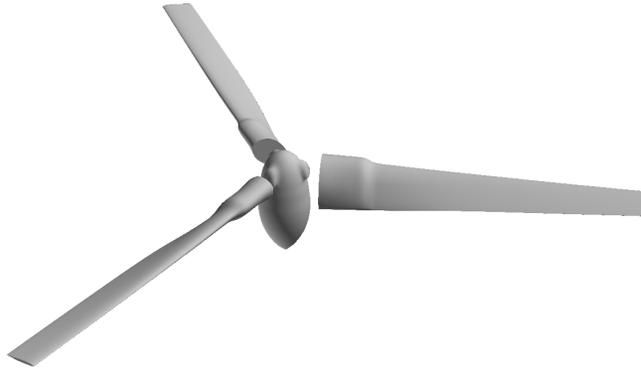
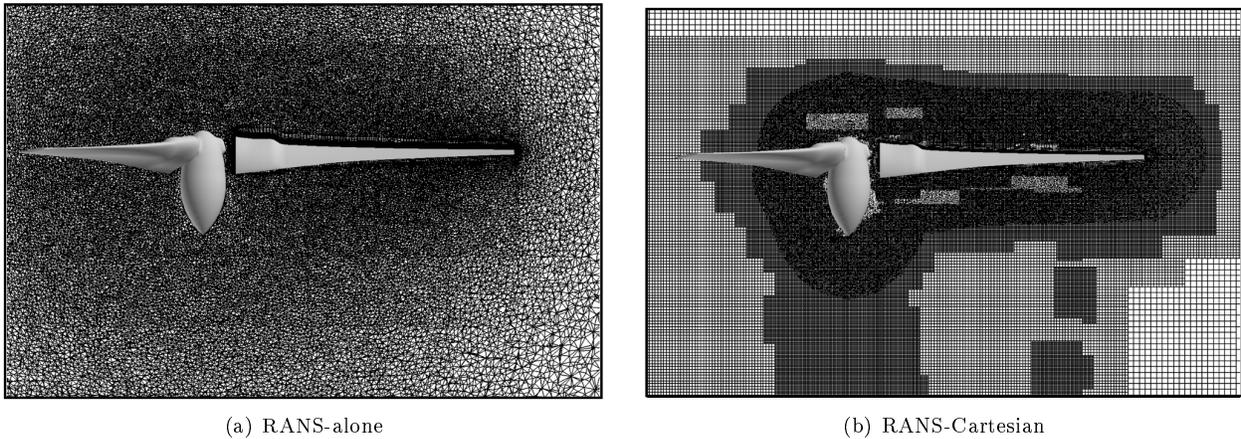


Figure 8: Three-bladed TRAM rotor.



(a) RANS-alone

(b) RANS-Cartesian

Figure 9: Mesh systems for TRAM calculations (a) fully unstructured; (b) unstructured near-body with adaptive Cartesian off-body.

alone calculation, the computational cost per step is only 13% larger. The reason is that the finite-difference numerics on structured Cartesian grids are found to be more than an order of magnitude faster than the finite-volume numerics on unstructured grids, even though the Cartesian solver is 5th-Order accurate while the unstructured solver is standard 2nd-order. While it is true that clustering gridpoints to the tip vortex region in the unstructured grid would improve the results of the RANS alone solution, doing so would have been significantly more computationally expensive.

3.3 Isolated V-22 (TRAM) Rotor in Hover

The Tilt Rotor Acoustic Model (TRAM) is a 0.25 scale model of the Bell/Boeing V-22 Osprey tiltrotor aircraft right-hand 3-bladed rotor. The isolated TRAM rotor was tested in the Duits-Nederlandse Windtunnel Large Low-speed Facility (DNW-LLF) in the spring of 1998. Aerodynamic pressures, thrust and power, were measured along with structural loads and aeroacoustics data. Wake geometry, in particular the locations of tip vortices, was not part of the data collected. Further details on the TRAM experiment and extensive CFD validations can be found in the work of Potsdam and Strawn [59].

The TRAM geometry contains multiple components, the 3 blades and a center-body (Fig. 8). Com-

	C_T		C_Q		FM	
Experiment	0.0149		0.00165		0.7794	
RANS-alone	0.0151	1% high	0.00183	11% high	0.7120	8.6% low
RANS-Cartesian	0.0155	4% high	0.00176	7% high	0.7707	1.1% low

Table 2: Comparison of computed and experimental loads. Thrust coefficient C_T , Torque coefficient C_Q , and Figure of Merit FM.

putations are performed for the $M_{tip} = 0.625$, 14° collective experimental condition with a tip Reynolds number of 2.1M. As in the previous cases, results are compared for the RANS-Cartesian and RANS-alone solutions. The grids for these two cases are shown in Fig. 9. The RANS-alone mesh, shown in Fig. 9(a), uses prismatic elements near the surface and tetrahedral elements elsewhere. It contains a total of 4.9M nodes. The RANS-Cartesian mesh system, shown in Fig. 9(b), uses a subsetting version of this same mesh in the near-body, trimmed a distance of approximately two blade chords. The off-body Cartesian mesh contains 7 levels of refinement with finest level spacing of 5% of the tip chord (C_{tip}). The off-body grid system is initially refined at the beginning of the simulation to the outer surface of the unstructured near-body mesh. As the solution evolves, it adapts to regions of high vorticity. The RANS-Cartesian mesh system shown in Fig. 9(a) contains 2.9M near-body nodes and 44M off-body nodes.

The case is run in hover conditions ($M_\infty = 0$ in the far-field) with $M_{tip} = 0.625$ and $Re=2.1M$. A non-inertial reference frame is used, such that the rotor stays fixed within a rotational freestream set through moving grid source terms. Although the freestream Mach number is low, the speed of the flow relative to the blade is high due to the rotational terms, so low-Mach preconditioning is not applied. The Spalart-Allmaras turbulence model is used.

The computed wake for the RANS-Cartesian and RANS alone calculations are shown in Fig. 10. An iso-surface of the Q-criterion at $Q=0.0001$ is shown, colored by vorticity magnitude. The Q-criterion of Hunt et al. [60] is often used to identify vortices in wake structures. It decomposes the velocity gradient into the vorticity tensor Ω and strain rate tensor S and defines the quantity Q to be the difference in their respective magnitudes:

$$\begin{aligned}\nabla u &= \Omega + S \\ Q &\equiv \frac{1}{2} (u_{i,i}^2 - u_{i,j}u_{j,i}) = \frac{1}{2} (\|\Omega\|^2 - \|S\|^2)\end{aligned}$$

In regions where $Q > 0$, vorticity magnitude prevails over the strain-rate magnitude, indicating large vortical structures such as tip vortices. Where $Q < 0$ the strain-rate magnitude is larger which indicates regions of high-vorticity but little structure, such as in boundary layers. Plotting $Q = 0$ (or slightly above) gives a nice representation of where large scale structures in the wake occur. The Q iso-surface in Fig. 10 is colored by vorticity magnitude.

The vortex wake computed using the RANS-Cartesian solution in Fig. 10(a) is clearly much better resolved than the RANS alone solution shown in Fig. 10(b). The RANS-Cartesian solution benefits from the use of high-order algorithms and refinement of the mesh to vortex structures. Vortex structures are maintained at nearly full-strength for four rotor revolutions. On the other hand, the RANS alone solution causes the wake structures to dissipate quickly because the solution is lower (2nd) order and the grid is not clustered to capture the vortices.

Table 2 compares the experimental and calculated thrust (C_T), torque (C_Q), and Figure of Merit (FM) for the RANS-alone and RANS-Cartesian solutions. (Figure of Merit is a measure the relative efficiency of the rotor, the ratio of the ideal power required to hover to the actual power required, and is computed as $FM = \frac{C_T^{1.5}}{\sqrt{2C_Q}}$). The calculated quantity that most rotor designers target is FM, the thrust and torque are often slightly over or under due to minor differences in the experimental vs. computed collective pitch angle (this is analogous to fixed wing calculations in which the computational angle of attack is adjusted until the computed lift coefficient matches the measured experimental lift coefficient). The FM computed using the RANS-Cartesian approach differs from experiment by only about 1%, compared to the RANS alone which differs by almost 9%. Better off-body wake resolution is the reason for this better performance prediction.

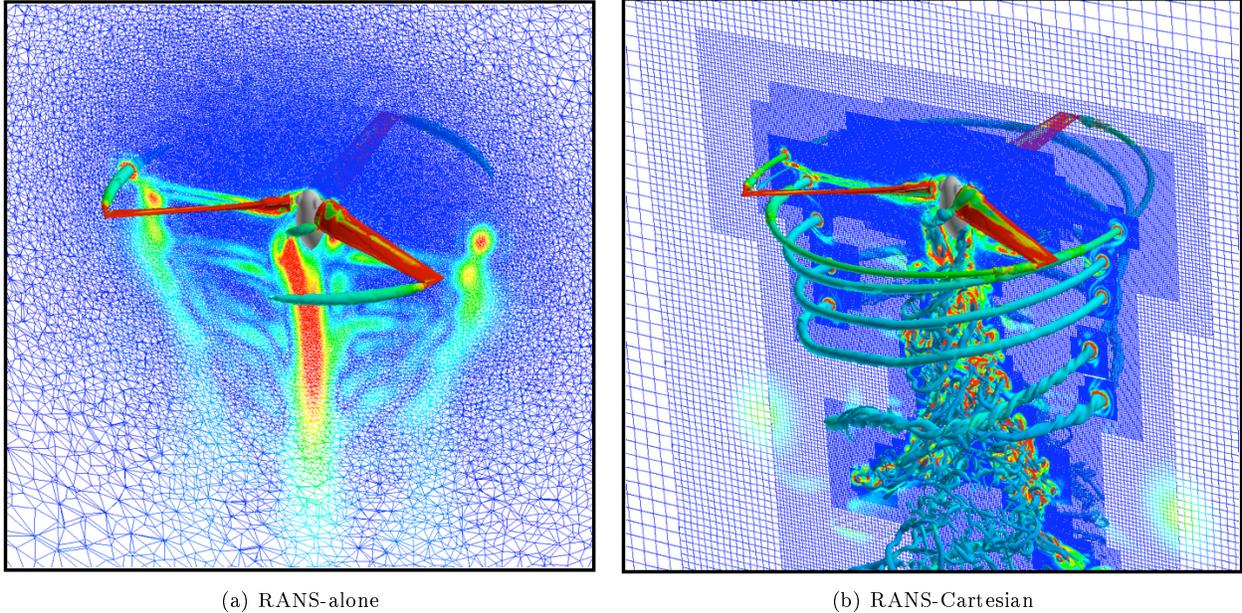


Figure 10: Isolated TRAM rotor calculations. Iso-surface of Q -criterion shown, colored by vorticity magnitude (red indicates high vorticity, blue low). (a) fully unstructured; (b) unstructured near-body with adaptive Cartesian off-body.

These results were run on a 64-processor x86 64-bit linux system which can be purchased today for modest price of \$50K-\$80K and which, in a decade or less, is very likely to be on a standard desktop. The final adapted Cartesian off-body mesh contained 44M gridpoints. While this may be considered large for an unstructured grid, it is actually quite reasonable for structured grids. The rule-of-thumb typically used for structured Cartesian meshes is 2M nodes per processor, so there is still room to increase the size of the off-body mesh further on this computer system.

The use of AMR plays an important role in preserving the ability to run such calculations on relatively modest computer resources. If a uniformly refined Cartesian grid system were used, whereby finely-spaced Cartesian grids were applied in the entire region downstream of the rotor plane, the number of Cartesian points would exceed 500M. While not unheard of – it would require a system with 250 or more processors – it nevertheless would require a high-end HPC system housed at a centralized HPC computing center. AMR makes it possible to run the calculation on a significantly smaller computer system, one that might be used commonly for CFD calculations in an industrial department or university research group.

4 Concluding Remarks

The use of a high-order adaptive Cartesian solver to enhance wake prediction of traditional unstructured RANS solvers is presented. The approach applies the native RANS solver in the near-body region then transitions to a high-order structured Cartesian Euler solver in the off-body or wake region. Results are shown using the Cartesian solver to refine to shed wakes from three applications; unsteady flow over a sphere, steady flow over a NACA0015 wing in which the Cartesian grids refine to wingtip vortices, and flow about a three-bladed helicopter rotor in which Cartesian grids refine to the rotor wake.

In all three cases tested the adaptive Cartesian grids are able to resolve the wake to a much better degree than a typical RANS mesh, and at little extra cost. A promising aspect of the approach is that it can be combined with any existing unstructured RANS solvers which are commonly used for engineering analysis today.

An important future direction for this work is to gain a better understanding of where to apply grid refinement and introduce measures to determine the appropriate level of mesh resolution. The results shown

here were achieved by dialing-in a desired vorticity threshold used to guide refinement. However, this is not be practical in engineering applications where such quantities are unknown beforehand and may vary considerably between problems.

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. Rotary-wing capability was added to NSU3D by Prof. Dimitri Mavriplis, under sponsorship of the institute. The author gratefully acknowledges additional contributions by Prof. Jayanarayanan Sitaraman, Dr. Venkateswaran Sankaran, Dr. Buvana Jayaraman, Mr. Mark Potsdam, Dr. Aaron Katz, Dr. Sean Kamkar, and Dr. Roger Strawn.

References

- [1] Morton, S., R. Cummings, and D. Kholodar, "High Resolution Turbulence Treatment of F/A-18 Tail Buffet," AIAA-2004-1676, 45th AIAA Structures, Structural Dynamics and Materials Conference, Palm Springs CA, 2004.
- [2] Caradonna, F. X., R.C. Strawn, and J.O. Bridgeman, "An Experimental and Computational Study of Rotor-Vortex Interactions," *Vertica*, Vol. 12, No. 4, 1988, pp. 315–327.
- [3] Strawn, R., and M. Djomehri, "Computational modeling of hovering rotor and wake aerodynamics," *AIAA Journal of Aircraft*, 39(5): 786-793, 2002.
- [4] Pulliam, T., and D. Jespersen, "Large Scale Aerodynamic Calculation on Pleiades," 2009 Parallel Computational Fluid Dynamics Conference, Moffett Field, CA, May 2009.
- [5] Berger, M.J., and J. Olinger, "Adaptive mesh refinement for hyperbolic partial differential equations," *Journal of Computational Physics*, 53, pp. 484-512, 1984.
- [6] Berger, M.J., and A. Jameson, "Automatic adaptive grid refinement for the Euler equations, *AIAA Journal*, Vol. 23, No. 4, pp.561–568, 1985.
- [7] Berger, M.J., "Data structures for adaptive grid generation," *SIAM Journal on Scientific and Statistical Computing*, 7(3):904-916, July 1986.
- [8] Bell, J., P. Colella, J. Trangenstein, and M. Welcome, "Adaptive Methods for High Mach Number Reacting Flow", AIAA 8th Computational Fluid Dynamics Conference, Honolulu, Hawaii, 1987.
- [9] Bryan, G., T. Abel, and M.L. Norman, "Achieving extreme resolution in numerical cosmology using adaptive mesh refinement: resolving primordial star formation," *Scientific Programming* 10(4): 291-302, 2002.
- [10] Hornung, R., and J. Trangenstein, "Adaptive Mesh Refinement and Multilevel Iteration for Flow in Porous Media," *Journal of Computational Physics*, 136, pp.522-545, 1998.
- [11] Bell, J.B., M. S. Day, J. F. Grear, M. J. Lijewski, J. F. Driscoll and S. F. Filatyev, "Numerical Simulation of a Laboratory-Scale Turbulent Slot Flame," *Proc. Combust. Inst.*, 31, pp.1299-1307, 2007.
- [12] Bono, G.C., and P. Colella, "An anelastic allspeed projection method for gravitationally stratified flows," *Journal of Computational Physics*, Vol. 216, pp. 589-615, 2006.
- [13] Griffith, B.E., R.D. Hornung, D.M. McQueen, and C.S. Peskin, "An adaptive, formally second order accurate version of the immersed boundary method," *Journal of Computational Physics*, 223:10-49, 2007.
- [14] Dacles-Mariani, J., S. Rogers, D. Kwak, G.G. Zilliac, and J. Chow, "A Computational Study of a Wingtip Vortex Flow Field," AIAA 93-3010, 1993.
- [15] Dacles-Mariani, J., D. Kwak, G.G. Zilliac, J. Chow, and P. Bradshaw, "A Numerical/Experimental Study of a Wingtip Vortex in the Near Field," *AIAA Journal*, Vol. 33, pp. 1561-1568, 1995.
- [16] Wissink, A.M., B. Jayaraman, A. Datta, J. Sitaraman, M. Potsdam, S. Kamkar, D. Mavriplis, Z. Yang, R. Jain, J. Lim, R. Strawn, "Capability Enhancements in Version 3 of the Helios High-Fidelity Rotorcraft Simulation Code," AIAA-2012-0713, 50th AIAA Aerospace Sciences Meeting, Nashville TN, Jan 2012.

- [17] Sankaran, V., A. Wissink, A. Datta, J. Sitaraman, B. Jayaraman, M. Potsdam, A. Katz, S. Kamkar, B. Roget, H. Saberi, W. Chen, W. Johnson, and R. Strawn, "Overview of the Helios Version 2.0 Computational Platform for Rotorcraft Simulations," AIAA-2011-1105, 49th AIAA Aerospace Sciences Meeting, Orlando FL, Jan 2011.
- [18] Sankaran, V., J. Sitaraman, A. Wissink, A. Datta, B. Jayaraman, M. Potsdam, D. Mavriplis, Z. Yang, D. O'Brien, H. Saberi, R. Cheng, N. Hariharan, and R. Strawn, "Application of the Helios Computational Platform to Rotorcraft Flowfields," AIAA-2010-1230, 48th AIAA Aerospace Sciences Meeting, Orlando FL, Jan 2010.
- [19] Post, D.E., "A new DoD initiative: the Computational Research and Engineering Acquisition Tools and Environments (CREATE) program," *Journal of Physics*, Conference Series 125, 2008.
- [20] Wissink, A.M., M. Potsdam, V. Sankaran, J. Sitaraman, Z. Yang, and D. Mavriplis, "A Coupled Unstructured-Adaptive Cartesian CFD Approach for Hover Prediction," *American Helicopter Society 66th Annual Forum*, Phoenix AZ, May 11-16, 2010.
- [21] Hariharan, N., M. Potsdam, A. Wissink, "Helicopter Rotor Aerodynamic Modeling in Hover Based on First-Principles: State-of-the-Art and Remaining Challenges," AIAA-2012-1016, 50th AIAA Aerospace Sciences Meeting, Nashville TN, Jan 2012.
- [22] Sitaraman, J., Potsdam, M., Jayaraman, B., Datta, A., Wissink, A., Mavriplis D., and Saberi, H., "Rotor Loads Prediction Using Helios: A Multi-Solver Framework for Rotorcraft CFD/CSD Analysis," AIAA-2011-1123, 49th AIAA Aerospace Sciences Meeting, Orlando FL, Jan 2011.
- [23] Lim, J. W., A.M. Wissink, B. Jayaraman, and A. Dimanlig, "Application of Adaptive Mesh Refinement Technique in Helios to Blade-Vortex Interaction Loading and Wakes," *American Helicopter Society 67th Annual Forum*, Virginia Beach, VA, May 2011.
- [24] Sankaran, V., M. Potsdam, A. Wissink, A. Datta, B. Jayaraman, and J. Sitaraman, "Rotor Loads Prediction in Level and Maneuvering Flight Using Unstructured-Adaptive Cartesian CFD," *American Helicopter Society 67th Annual Forum*, Virginia Beach, VA, May 3-5, 2011.
- [25] Potsdam, M., A. Wissink, S. Kamkar, B. Jayaraman, "CFD Adaptive Mesh Refinement for Rotorcraft Wake Simulations," Paper 148, 37th European Rotorcraft Forum, Milano Italy, Sept 13-15, 2011.
- [26] Jayaraman, B., A.M. Wissink, J. Lim, M. Potsdam, A. Dimanlig, "Helios Prediction of Blade-Vortex Interaction and Wake of the HART II Rotor," AIAA-2012-0714, AIAA 50th Aerospace Sciences Meeting, Nashville TN, January 2012.
- [27] Lim, J., A. Wissink, B. Jayaraman, and A. Dimanlig, "Helios Adaptive Mesh Refinement for HART II Rotor Wake Simulations," *Proceedings of the 68th Annual Forum of the American Helicopter Society*, Fort Worth TX, May 2012.
- [28] Mavriplis, D. J., and V. Venkatakrishnan, "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.
- [29] Benek, J., P. Buning, and J. Steger, "A 3-D chimera grid embedding technique," AIAA Paper 85-1523, 1985.
- [30] Steger, J.L., and J. A. Benek, "On the use of composite grid schemes in computational aerodynamics," *Computer Methods in Applied Mechanics and Engineering*, v.64 n.1-3, p.301-320, 1986.
- [31] Noack, R. W., "DiRTlib: A Library to Add an Overset Capability to Your Flow Solver," AIAA Paper 2005-5116, June 2005.
- [32] Noack, R. W., "SUGGAR: a General Capability for Moving Body Overset Grid Assembly," AIAA Paper 2005-5117, June 2005.
- [33] Lee-Rausch, E.M., and R.T. Biedron, "Simulation of an Isolated Tiltrotor in Hover with an Unstructured Overset-Grid RANS Solver," *American Helicopter Society 65th Annual Forum*, Grapevine, TX, May 27-29, 2009.
- [34] J. Sitaraman, M. Floros, A. Wissink, and M. Potsdam, "Parallel domain connectivity algorithm for unsteady flow computations using overlapping and adaptive grids," *Journal of Computational Physics*, 229(12):4703-4723, 2010.
- [35] Wissink, A. M., J. Sitaraman, V. Sankaran, D. J. Mavriplis, and T. H. Pulliam, "A Multi-Code Python-Based Infrastructure for Overset CFD with Adaptive Cartesian Grids", AIAA-2008-0927, 46th AIAA Aeroscience Conference, Reno NV, Jan 2008.
- [36] Sitaraman, J., A. Katz, B. Jayaraman, A. Wissink, V. Sankaran, "Evaluation of a Multi-Solver Paradigm

- for CFD using Unstructured and Structured Adaptive Cartesian Grids,” AIAA-2008-0660, 46th AIAA Aerosciences Conference, Reno NV, Jan 2008.
- [37] Mavriplis, D. J., “Adaptive meshing techniques for viscous flow calculations on mixed element unstructured meshes.” *International Journal for Numerical Methods in Fluids*, Vol 34, Issue 2, September 2000, Pages: 93-111.
- [38] Park, M. A., “Adjoint-Based, Three-Dimensional Error Prediction and Grid Adaptation,” AIAA-2002-3286, 32nd AIAA Fluid Dynamics Conference, St. Louis MO, June 2002.
- [39] Potsdam, M. A., and D. J. Mavriplis, “Unstructured Mesh CFD Aerodynamic Analysis of the NREL Phase VI Rotor,” AIAA-2009-1221, 47th AIAA Aerosciences Meeting, Orlando FL, Jan 5–8, 2009.
- [40] Aftosmis, M.J., Berger, M.J., Alonso, J.J., "Applications of a Cartesian mesh boundary-layer approach for complex configurations", AIAA 2006-0652, 44th AIAA Aerospace Sciences Meeting and Exhibit, Reno NV, Jan 2006.
- [41] Buning, P.G., and T. H. Pulliam, “Cartesian Off-Body Grid Adaption for Viscous Time Accurate Flow Simulation,” AIAA-2011-3693, 20th AIAA Computational Fluid Dynamics Conference, Honolulu HI, June 27–30, 2011.
- [42] Berger, M. J., and P. Colella, “Local Adaptive Mesh Refinement for Shock Hydrodynamics,” *J. Comp. Phys.*, **82**, 1989, pp. 65–84.
- [43] Wissink, A. M., D. A. Hysom, and R. D. Hornung, “Enhancing Scalability of Parallel Structured AMR Calculations”, Proceedings of the 17th ACM International Conference on Supercomputing (ICS03), San Francisco CA, June 2003, pp. 336–347.
- [44] Gunney, B. T. N., A. M. Wissink, and D. A. Hysom, “Parallel Clustering Algorithms for Structured AMR,” *J. Parallel. Dist. Computing*, **66**, 2006, pp. 1419–1430.
- [45] Parashar, M., and J.C. Browne, “A Common Data Management Infrastructure for Parallel Adaptive Algorithms for PDE Solutions,” Proceedings of Supercomputing 97, 1997.
- [46] Olson, K., "PARAMESH: A Parallel Adaptive Grid Tool", *Parallel Computational Fluid Dynamics 2005: Theory and Applications*, College Park, MD, U.S.A., Elsevier pub., eds. A. Deane, A. Ecer, G. Brenner, D. Emerson, J. McDonough, J. Periaux, N. Satofuka, 2006.
- [47] Colella, P., D.T. Graves, T.J. Ligocki, D.F. Martin, D. Modiano, D.B. Serafini, B. Van Straalen, “Chombo Software Package for AMR Applications,” Applied Numerical Algorithms Group, NERSC Division, September 2003.
- [48] Hornung, R. D., A. M. Wissink, and S. R. Kohn, “Managing Complex Data and Geometry in Parallel Structured AMR Applications,” *Engineering with Computers*, Vol. 22, No. 3-4, Dec. 2006, pp. 181-195. Also see www.llnl.gov/casc/samrai.
- [49] Hornung, R. D., and S. R. Kohn, “Managing Application Complexity in the SAMRAI object-oriented framework,” *Concurrency and Computation: Practise and Experience*, Vol. 14, 2002, pp. 347-368.
- [50] Wissink, A. M., R. D. Hornung, S. Kohn, S. Smith, and N. Elliott, “Large-Scale Parallel Structured AMR Calculations using the SAMRAI Framework”, Proceedings of Supercomputing 2001 (SC01), Denver CO, Nov 2001.
- [51] Hariharan, N., and L. Sankar, “A Review of Computational techniques for Rotor Wake Modeling,” AIAA paper 2000-0014, 38th AIAA Aerospace Sciences Meeting, Reno, NV, Jan 2000.
- [52] Hariharan, N., and Sankar, L. N., “High-Order Essentially Non-oscillatory Schemes for Rotary-Wing Wake Computations,” *Journal of Aircraft*, Vol. 41, No. 2, pp. 258-267, March-April 2004.
- [53] Pulliam, T. H., “Euler and Thin-Layer Navier-Stokes Codes: ARC2D, and ARC3D” Computational Fluid Dynamics Users Workshop, The University of Tennessee Space Institute, Tullahoma TN, March 1984.
- [54] Pulliam, T. H., “Solution Methods in Computational Fluid Dynamics,” *von Karman Institute for Fluid Mechanics Lecture Series, Numerical Techniques for Viscous Flow Computations in Turbomachinery*, Rhode-St-Genese, Belgium, Jan 1986. See http://people.nas.nasa.gov/pulliam/mypapers/vki_notes/vki_notes.html.
- [55] Mavriplis, D. J., C. Nastase, L. Wang, and K. Shahbazi, “Progress in High- Order Discontinuous Galerkin Methods for Aerospace Applications,” AIAA-2009-601, 47th AIAA Aerospace Sciences Meeting, Orlando, FL, January 2009.
- [56] McAlister, K. W., and R. K. Takahashi, “NACA 0015 Wing Pressure and Trailing Vortex Measurements,” NASA Technical Paper 3151, AVSCOM Technical Report 91-A-003, Nov 1991.

- [57] Sitaraman, J., and J. D. Baeder, "Evaluation of the Wake Prediction Methodologies used in CFD Based Rotor Airload Computations," AIAA-2006-3472, 24th AIAA Applied Aerodynamics Conference, San Francisco CA, June 2006.
- [58] Hariharan, N., "Rotary-Wing Wake Capturing: High Order Schemes Towards Minimizing Numerical Vortex Dissipation," *AIAA Journal of Aircraft*, Vol. 39, No. 5, pp. 822-830, 2002.
- [59] Potsdam, M. A., and R. C. Strawn, "CFD Simulations of Tiltrotor Configurations in Hover," *Journal of the American Helicopter Society*, Vol. 50, No. 1, pp. 82-94, 2005.
- [60] Hunt, J. C. R., A. A. Wray, and P. Moin, "Eddies, Stream, and Convergence Zones in Turbulent Flows," Center for Turbulence Research Report CTR-S88, pp. 193-208.