50th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition
09 - 12 January 2012, Nashville, Tennessee

AIAA 2012-0713

Capability Enhancements in Version 3 of the Helios High-Fidelity Rotorcraft Simulation Code

Andrew Wissink[1], Buvana Jayaraman[2], Anubhav Datta[2], Jayanarayanan Sitaraman[3], Mark Potsdam[1], Sean Kamkar[2], Dimitri Mavriplis[3], Zhi Yang[3], Rohit Jain[2], Joon Lim[1], and Roger Strawn[1]

[1]*U.S. Army Aeroflightdynamics Directorate (AMRDEC), Moffett Field, CA*
[2]*Science & Tech. Corp, NASA Ames Research Center, Moffett Field, CA*
[3]*Dept. of Mechanical Engineering, University of Wyoming, Laramie, WY*

This paper presents an overview of new capabilities in the Helios v3, or Rainier, high-fidelity rotorcraft simulation software. Key new capabilities include the addition of DES turbulence modeling in the near-body solver and RANS in the off-body solver, introduction of Richardson extrapolation-based error control to automate off-body AMR, and runtime parallel partitioning of near-body grids. We also report on advances made in Helios to support loose-coupling rotor-fuselage and multi-rotor configurations. The paper describes these capability enhancements in detail and provides validation results and computational performance metrics for the model TRAM rotor, HART-II, and UH-60 configurations.

I. Introduction

Rotorcraft computations are challenging due to their dynamic and multidisciplinary nature. Rotor blades experience all the complex aerodynamic flow conditions that make complicate traditional fixed-wing CFD predictions - viscous effects, compressibility, turbulence, and stall - except that rotors may experience all these conditions within a single rotor revolution. Moreover, rotorcraft flowfields demand extremely accurate resolution of the wake vortices over relatively long distances because of the importance of blade-vortex interactions and fuselage effects. On the structural dynamics side, blades experience considerable aero-elastic effects with highly unsteady control loads transmitted through a structurally complex hub connection. The strong coupling between the aerodynamic and structural dynamics simulations must also be coupled with vehicle flight dynamics and controls software achieve a trimmed vehicle state. See the reviews by Strawn et al.[1] and Datta and Johnson[2] for further details on the complexities of rotorcraft modeling.

The Helios software was introduced as a rotary-wing product of the of the CREATE-AV (air vehicles) program,[3] sponsored by the Department of Defense High Performance Computing Modernization Office, specifically to provide a high-fidelity analysis capability to the DoD for the acquisition of new rotary-wing aircraft. The dual-mesh paradigm 1 that is the basis of the CFD aerodynamics solution procedure consists of unstructured meshes in the near-body to capture viscous flow around complex geometry, and block structured Cartesian grids in the off-body to resolve the wake through a combination of high-order algorithms and adaptive mesh refinement (AMR). 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 rotor meshes rotate and deform inside the stationary adaptive Cartesian off-body grids system. Rotor motion, deformation, flight controls and trim operations are provided by an external comprehensive analysis package. Coordination of the different codes is managed through a lightweight and flexible Python-based infrastructure.

The first version of the software, Helios v1 *Whitney*,[4] was released to government and industry users in 2010. It was capable of simulating fuselages and isolated rotors and used a fixed high-order Cartesian background grid. It had interfaces to the U.S. Army's RCAS comprehensive analysis software[5] for structural dynamics and trim calculations. The second version, Helios v2 *Shasta*,[6,7] was released to users in 2011. It added the capability to perform adaptive off-body refinement, support for coupled rotor-fuselage and multiple rotor configurations, and a generalized comprehensive analysis interface. Helios v3 *Rainier* adds

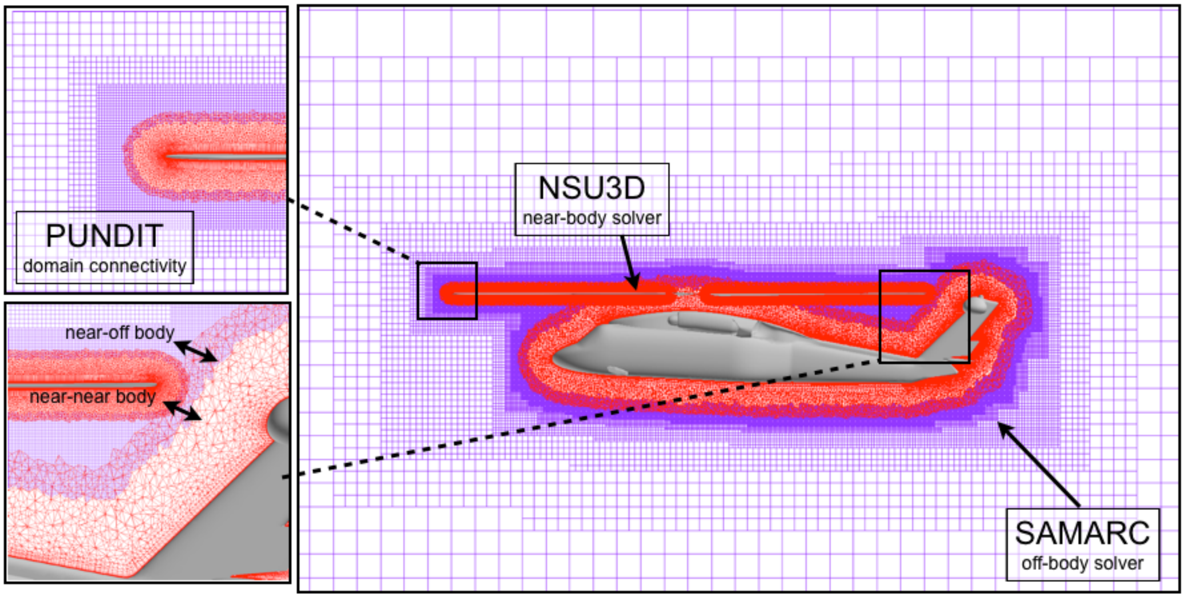


Figure 1: Overset near/off-body gridding paradigm used in `Helios`. Unstructured or curvilinear grids to capture geometric features and boundary layer near body surface, adaptive block-structured Cartesian grids to capture far-field flow features.

support for Detached Eddy Simulation (DES) turbulence modeling capability in the near-body and viscous terms in the off-body. It also includes the introduction of a Richardson-extrapolation based technique to automate the degree of wake resolution with AMR in the off-body. Lastly, it provides parallel near-body unstructured grid partitioning.

This paper describes the Helios v3 capability enhancements in detail and presents validation results for each. Some capabilities described in earlier Helios references, such as loose-coupled rotor-fuselage and multi-rotor cases, had not yet been fully exercised at the time of writing and have since been more thoroughly evaluated and tested. Hence, we devote a portion of the results section in this paper to some recent results obtained using Helios for these cases. The remainder of the paper is organized as follows. The next section II describe the Helios v3 capability enhancements. Section III provides some recent results for rotor-fuselage and multi-rotor configurations and gives performance statistics measured on HPC systems on which Helios users typically run. A brief summary is provided at the end of the paper.

II. Capability Enhancements

The three major new capabilities being released in Helios v3 are DES turbulence modeling, error-driven grid off-body grid refinement, and parallel unstructured grid pre-processing. Each of these capabilities are described in the sub-sections below.

II.A. DES Turbulence Modeling

Earlier versions of Helios utilized the RANS solver in NSU3D for the near-body with the high-order inviscid Euler solver in SAMARC in the off-body. In Rainier the solvers have been augmented in two ways. First, the Detached Eddy Simulation (DES) has been enabled in NSU3D in order to better model the wakes of separated flows. Second, RANS capability has been enabled in the off-body. Further details of these two updates along with validation results are given in the following sub-sections.

II.A.1. Near-body DES/DDES Formulation

Implementing the hybrid DES model involves automatically switching to a subgrid scale Large Edddy Simulation (LES) formulation based on grid density. Regions where the grid cannot support the turbulence length scale, generally near wall boundaries, are solved using traditional RANS. Regions where the grid is length scale is sufficient to resolve the turbulent length scale are solved using LES. Implementation of DES involves changing the length scale used in the Spalart-Allmaras turbulence model. The length scale can be written as:

$$\widetilde{d} = d - f_d \max\left(0, d - C_{DES}\Delta\right) \tag{1}$$

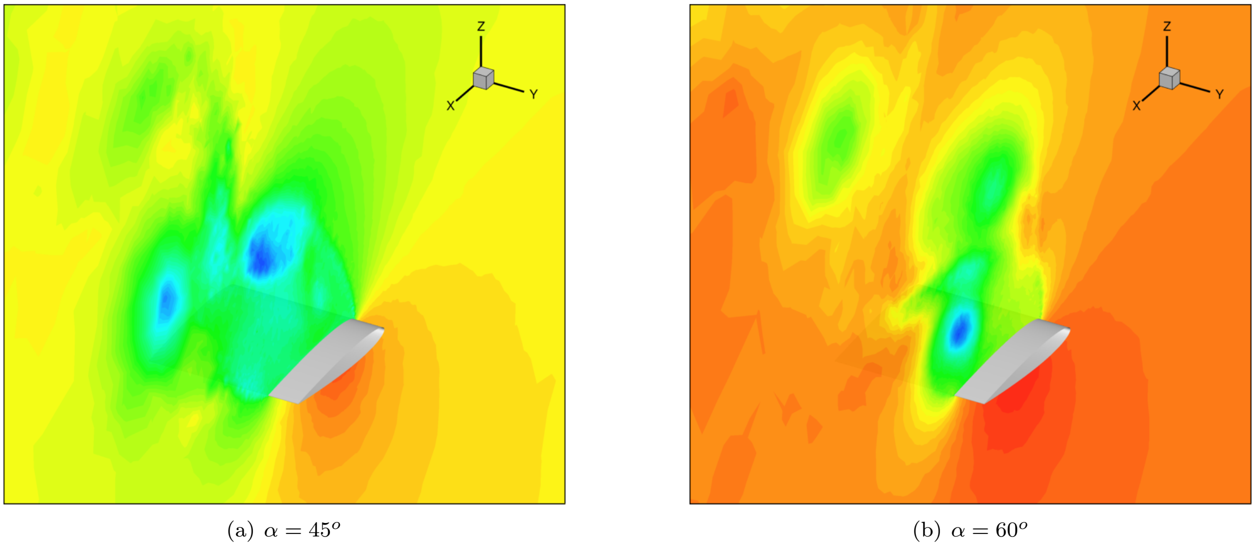
where d is the distance from the wall, $C_{DES} = 0.65$, and Δ a measure of the local grid spacing.[8] In NSU3D Δ is set to the maximum edge length touching a given vertex on unstructured mesh.[9] If f_d is set to 1, then equation (1) becomes:

$$\widetilde{d} = \min\left(d, C_{DES}\Delta\right) \tag{2}$$

The length scale in this original formulation is grid dependent causing incorrect behavior to be observed for some cases with ambiguous grid densities. The Delayed-DES (DDES) model[10] was introduced to be less sensitive to the grid. DDES computes f_d by:

$$f_d = 1 - \tanh\left([8r_d]^3\right) \tag{3}$$

where $r_d = \frac{\nu_t + \nu}{\sqrt{U_{i,j}U_{i,j}}\kappa^2 d^2}$. ν_t and ν are the kinematic eddy viscosity and the molecular viscosity, respectively. $U_{i,j}$ are the velocity gradients. κ is the Karman constant. Either the DES or DDES options can be invoked in NSU3D.



(a) $\alpha = 45^o$

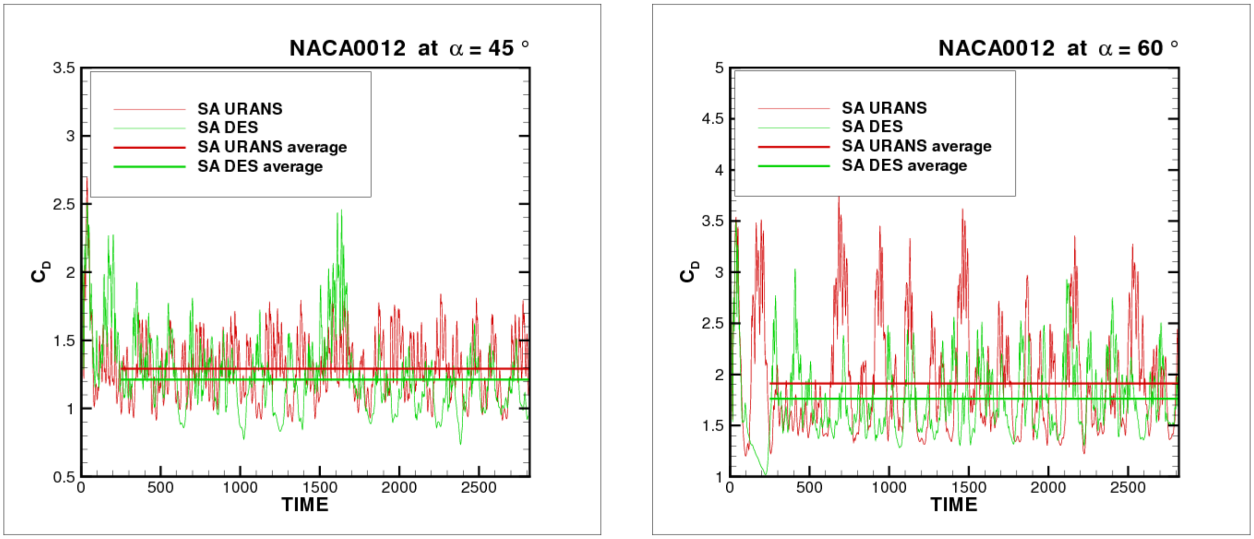
(b) $\alpha = 60^o$

Figure 2: Stalled NACA 0012 wing at $M_\infty = 0.15$, computed density contours with DES turbulence model.

The DES capability is evaluated for calculations about a NACA 0012 airfoil at $M_\infty = 0.15, Re = 10^5$ at 45^o and 60^o angle of attack. The airfoil is fully stalled at these high angles of attack and experiences time-dependent shedding behavior. Figure 2(a) and (b) shows the density contour solution at the $y = 0.8s$ spanwise location for the 45 and 60 degrees AOA cases. Figure 3 shows the computed drag coefficient, both the time dependent shedding behavior and the average over the simulation. Figure 4 shows the computed average C_L and C_D with DES and traditional RANS compared to the experimentally measured quantities.[11] With DES turned on the computed results more closely approximate the measured experimental values.

II.A.2. RANS Off-Body

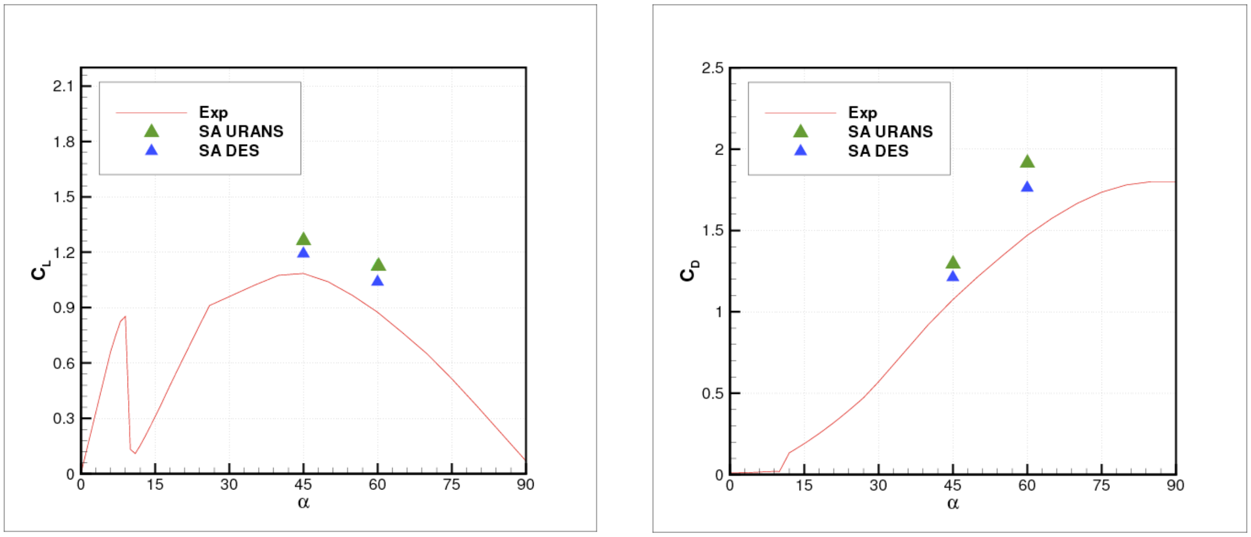
The version of the SAMARC solver released in Helios v1 and v2 used high-order spatial discretizations to solve the inviscid Euler equations.[12] In Helios v3 viscous terms have been enabled to make SAMARC a RANS solver. The viscous terms are discretized with a second order scheme but this is not expected to reduce the overall accuracy since diffusion terms are generally localized and sufficiently resolved with lower order methods.



(a) $\alpha = 45^o$

(b) $\alpha = 60^o$

Figure 3: Time dependent and average computed drag coefficients, RANS vs. DES.



(a) C_L

(b) C_D

Figure 4: Measured and computed average lift and drag coefficients using RANS and DES.

We have not yet enabled turbulence modeling in the off-body so the eddy viscosity remains zero. The general consensus in the aerospace community was that turbulence models are primarily needed in the wall-bounded regions, although recent results from Chaderjian[14] show that LES may also be needed for very fine wake meshes. Addition of LES to SAMARC is currently on-going.

The RANS off-body capability is investigated for hover calculations of the isolated Tilt Rotor Acoustic Model (TRAM) rotor, a quarter-scale model of the Bell/Boeing V-22 Osprey tiltrotor aircraft right-hand 3-bladed rotor shown in Fig. 5. The TRAM geometry contains four components—3 blades and a center-body. Computations are performed for the $M_{tip} = 0.625$, 14^o collective experimental condition with a tip Reynolds number of 2.1M. The near-body volume mesh was generated using AFLR[13] and contained a total of 9.3M nodes (111K surface faces, 3.0M nodes per blade). It extends approximately 1.5 rotor radii from blade (Fig. 5(a)).

The off-body Cartesian domain extends 10 rotor radii in the lateral and upward directions, and 20 radii below the rotor plane. A total of 8 levels of grid refinement are applied with finest level spacing corresponding to 5% of the tip chord. AMR adapts to regions of high vorticity. If the grid is allowed to refine everywhere in these hover calculations the scheme tends to add an excessive number of points in the wake region well below the rotor plane, where the tip vortices break down into smaller-scale structures. Flow in this region generally has minimal impact on the force calculations and is considered unnecessary to resolve with many gridpoints so solution-based refinement is cut off one rotor radii below the blade plane. Figure 5(b) shows the resulting adaptive off-body grid system.

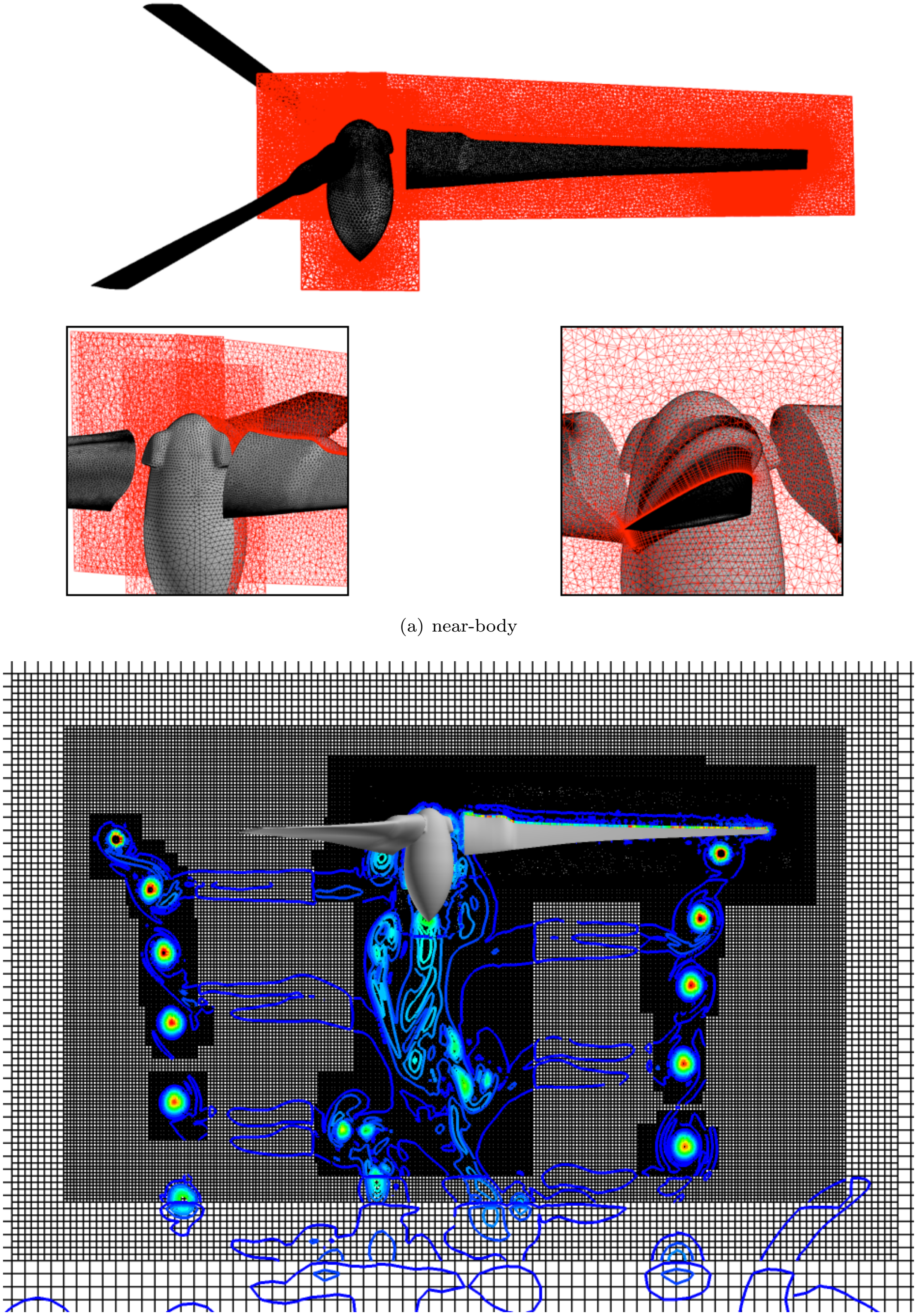
The case is run two ways. First, as a steady calculation in a non-inertial reference frame, where the rotor remains fixed and rotational terms are to the solved equations. Second, in a time-dependent inertial reference frame where the near-body blade mesh spins within the off-body Cartesian mesh that remains fixed. The steady case is run for a total of 35K steps; 10K steps initially applying only geometry refinement to dissipate non-physical startup vortices, then for 25K additional steps with solution-based refinement. Off-body refinement is applied every 100 steps. The unsteady case is run using a timestep of $\Delta\Psi = 0.25°$ for 8 rotor revolutions; 2 revs with geometry refinement only to dissipate startup vortices then an additional 6 revs with solution-based refinement. The near-body solver uses an implicit BDF2 time-stepping scheme with a user-prescribed 20 sub-iterations per timestep. The explicit off-body solver automatically determines the number of steps based on CFL restrictions of the finest level grid, in this case applying 5 sub-iterations per timestep. Off-body refinement is applied every 10 steps or 2.5^o azimuth.

RANS and Euler off-body solutions for the steady non-inertial and unsteady inertial calculations are shown in Figure 6. Figures 6(a) and (b) show the converged steady RANS and Euler solutions while (c) and (d) show the unsteady RANS and Euler solutions after 8 revs. Table 1 compares the accuracy of the computed figure of merit (FM) for these four cases. FM is a measure of the relative efficiency of the rotor, the ratio of the ideal power required to hover to the actual power required, and is computed as $FM = \frac{C_T^{1.5}}{\sqrt{2}C_Q}$. The percentage error from experiment is included for reference. Both the steady and unsteady calculations show some small variation in the final computed loads, which is included in the table as error bounds on the average computed solution.

	#OB Nodes (avg)	**FM**	**%Error**
Experiment	–	0.774	–
Steady Euler	31.5M	0.772	-0.25% ± 0.2%
Steady RANS	31.5M	0.772	-0.25% ± 0.2%
Unsteady Euler	34.4M	0.774	+0.01% ± 0.1%
Unsteady RANS	33.8M	0.774	+0.02% ± 0.1%

Table 1: TRAM calculation accuracy vs. number of gridpoints for different levels of RE error control. FM is rotor figure of merit.

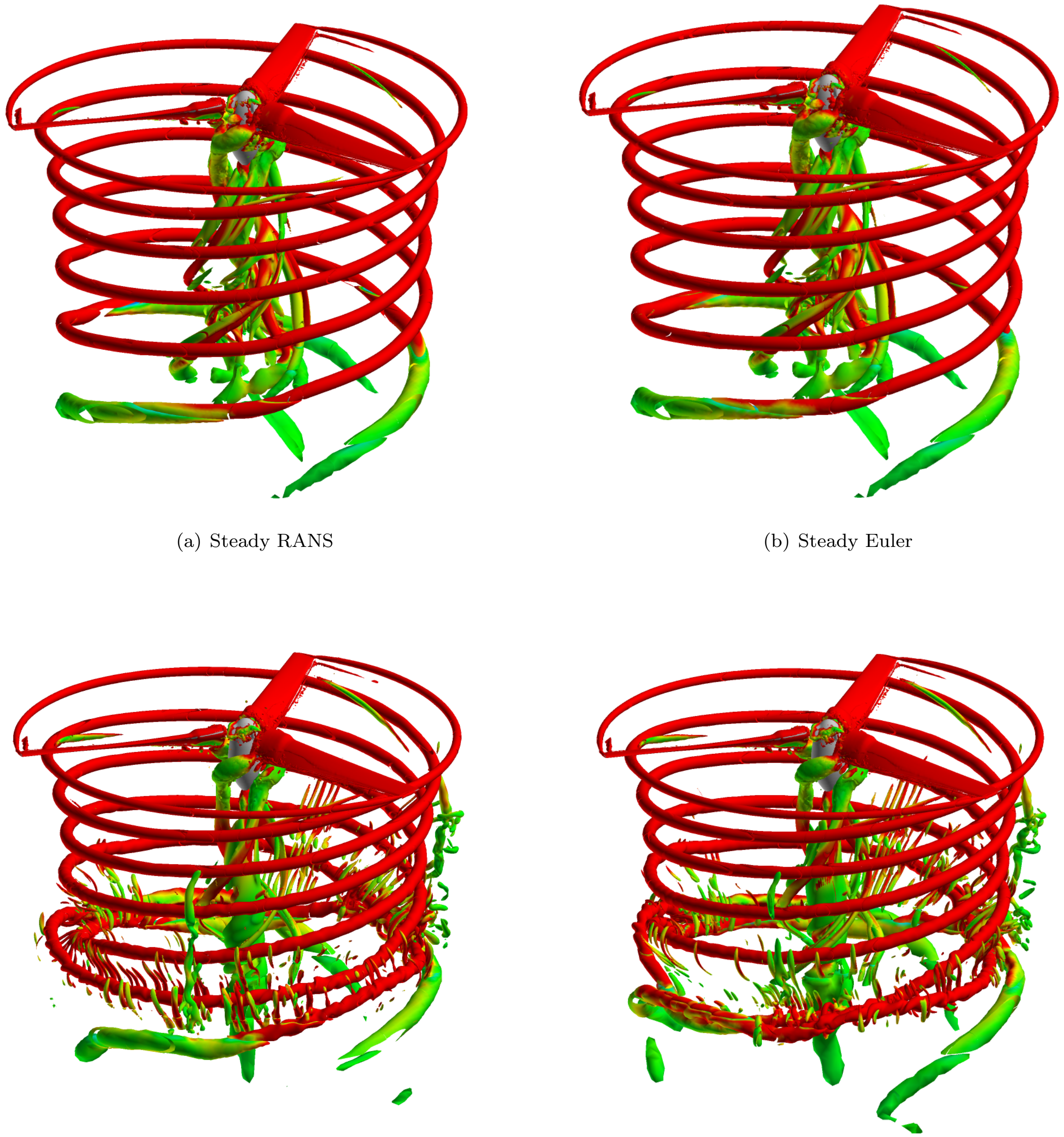
There is in general very little difference between the different solutions. Both the steady and unsteady formulations closely match the experimental result with deviation that is likely within the experimental measurement error. Inclusion of RANS in the off-body makes very little difference to the overall solution, both in terms of computed loads and wake structure. This implies that the inviscid off-body assumption, used in earlier versions of Helios, is probably reasonable for problems of this type. However, the investigation



(a) near-body

(b) off-body

Figure 5: TRAM near-body (a) and off-body (b) grid systems.



(a) Steady RANS

(b) Steady Euler

(c) Unsteady RANS

(d) Unsteady Euler

Figure 6: TRAM wake solutions using steady and unsteady RANS and Euler off-body solutions.

should be repeated once LES turbulence modeling has been added to the off-body solution.

II.B. Error-Driven Off-body AMR

Helios v2 introduced the use of off-body AMR to resolve the wakes of rotors in both steady hover and unsteady forward flight. In v3 we enhance this capability with the addition of a feature-error automated off-body refinement scheme developed by Kamkar.[15,16] A brief overview of the approach is presented here, further details are in an accompanying paper at this meeting.[17]

The original approach for wake refinement used in earlier versions of Helios is to target regions of high vorticity or Q-criterion and refine the mesh wherever this quantity exceeds some specified threshold. This approach has been successfully employed for a number of applications but has the drawback that it requires problem-specific tuning. The user must decide apriori the specific quantity they wish to refine to and the threshold value of that quantity, as well as the level of grid resolution that should be applied. This can be difficult to judge beforehand so it usually involves some intervention and tuning on the part of the user before a well-resolved wake solution can be obtained. Another drawback is the lack of an easily computable error function, which can be used to observe improvements in the solution accuracy.

The refinement strategy employed in Helios v3 utilizes a mix of non-dimensional feature detection schemes[15] together with Richardson extrapolation error with the objective of providing a single error threshold to control the degree of wake resolution. The feature detection scheme identifies regions of swirling flow independent of magnitude and the error-adaptive control, guided by Richardson extrapolation, adjusts the mesh resolution. Put more simply, the feature part of the algorithm identifies where to apply grid refinement and the Richardson part identifies how much resolution to apply.

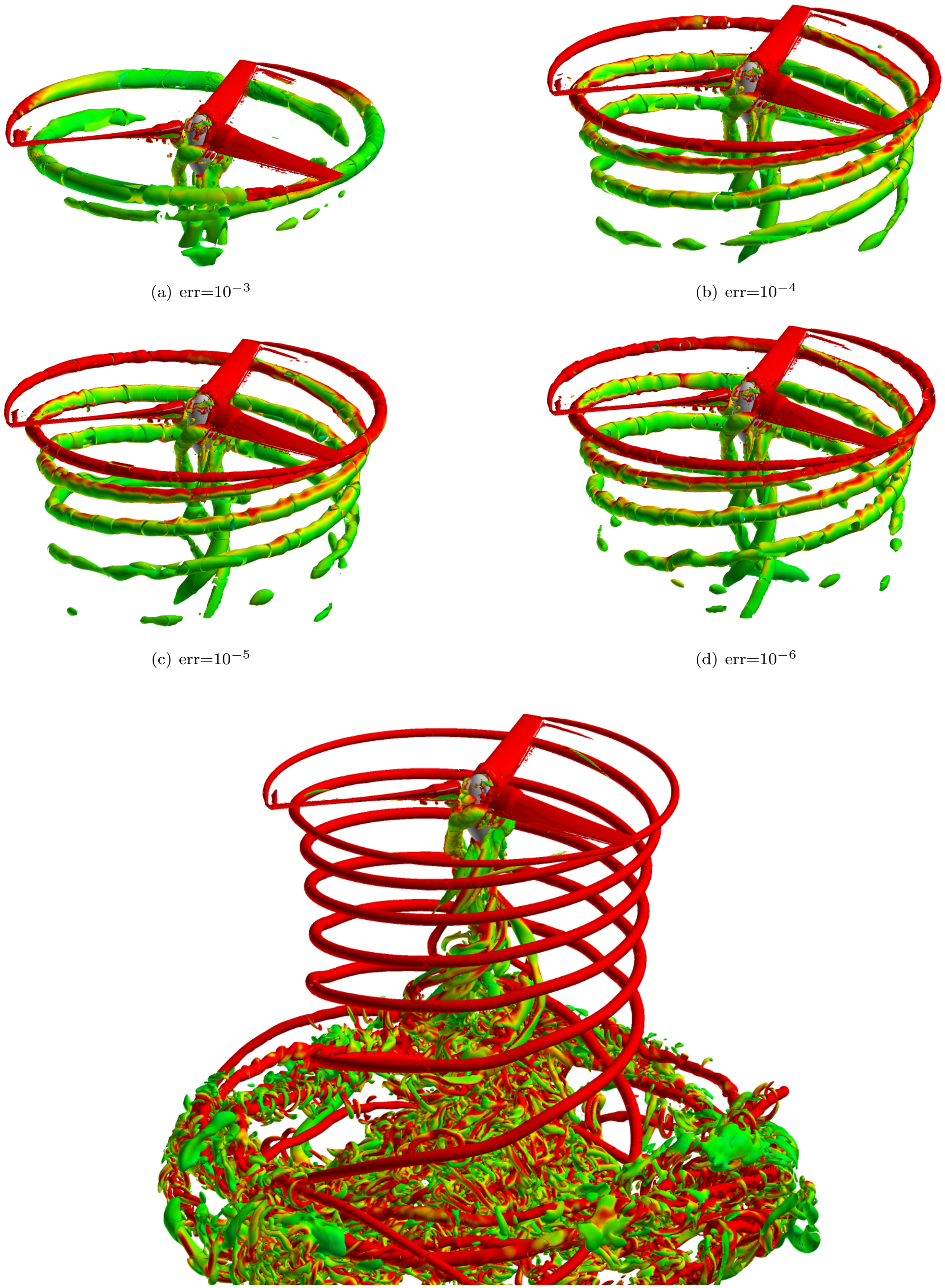
This new capability is evaluated for the same TRAM hover case described in Section II.A.2. For the tests here only the unsteady inertial formulation is run, the off-body is Euler, and no geometric refinement cutoff is applied. The Richardson error threshold is the only quantity used to guide refinement. Figure 7 shows the wake solutions, iso-surfaces of Q criterion colored by vorticity magnitude, achieved using different levels of Richardson error thresholds. Figure 8 shows the resulting off-body mesh system. As the error tolerance is tightened from 10^{-3} to 10^{-6} ((a)-(d), respectively) the wake resolution improves. With error control turned off completely (e) the wake is very well resolved, but many gridpoints are used to resolve the small-scale features well downstream of the rotor.

In practice we found that further tightening of the tolerance beyond 10^{-6} did not appreciably enhance the wake solution. We suspect this may be due to the noise threshold setting and intend to pursue it further in future work. Nevertheless, the error threshold appears achieve its intended goal of providing a means to control the degree of unsteady wake resolution.

	#OB Nodes (avg)	**FM**	**%Error**
Experiment	–	0.774	–
err = 10^{-3}	9.6M	0.760	-1.8% ± 0.4%
err = 10^{-4}	13.2M	0.767	-0.9% ± 0.4%
err = 10^{-5}	14.2M	0.768	-0.8% ± 0.2%
err = 10^{-6}	14.3M	0.769	-0.6% ± 0.4%
no error control	85.9M	0.773	-0.1% ± 0.2%

Table 2: TRAM calculation accuracy vs. number of gridpoints for different levels of RE error control. FM is rotor figure of merit.

Table 2 compares the accuracy of the computed figure of merit (FM) and the number of off-body Cartesian gridpoints for different levels of RE error control. The percentage error from experiment is included for reference. Because this is an unsteady calculation the final computed loads have a small but not insignificant time-dependent variation, shown in the table with error bounds. The best result is obtained with no error control (effectively, err = 0), albeit with a very large number of gridpoints. With error tolerances of 10^{-4} to 10^{-6} the computed FM differs by about 1% of experiment with 6X fewer gridpoints.



(a) err=10^{-3}

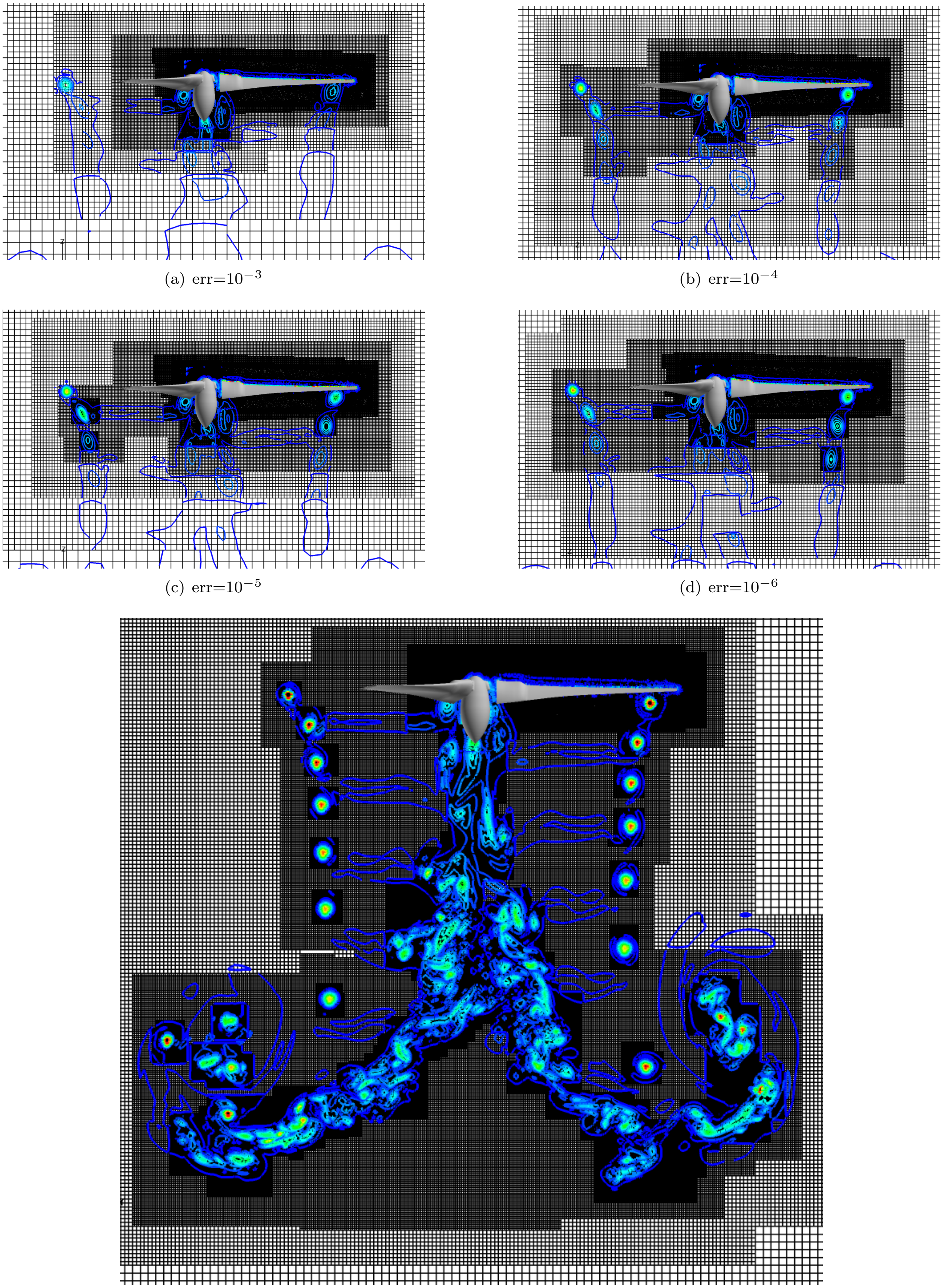
(b) err=10^{-4}

(c) err=10^{-5}

(d) err=10^{-6}

(e) no error control

Figure 7: Resolution of TRAM wake with different levels of RE error control.



(a) err=10^{-3}

(b) err=10^{-4}

(c) err=10^{-5}

(d) err=10^{-6}

(e) no error control

Figure 8: TRAM AMR wake meshes overlaid with vorticity contours using different levels of RE error control.

II.C. Parallel Unstructured Grid Preprocessing

Helios v1 and v2 users are familiar with the unstructured mesh preprocessing phase required during problem setup. Given a single unstructured mesh or a series of overset component unstructured meshes, preprocessing was required to move or transform meshes, apply component and body tags, subset the meshes for the dual-mesh paradigm (Fig. 9), and apply mesh partitioning for parallel processing. In earlier versions of Helios this procedure required the entire near-body mesh system to be loaded into core memory and this became problematic if the combined unstructured mesh became large, generally if it exceeded 10M grid points.

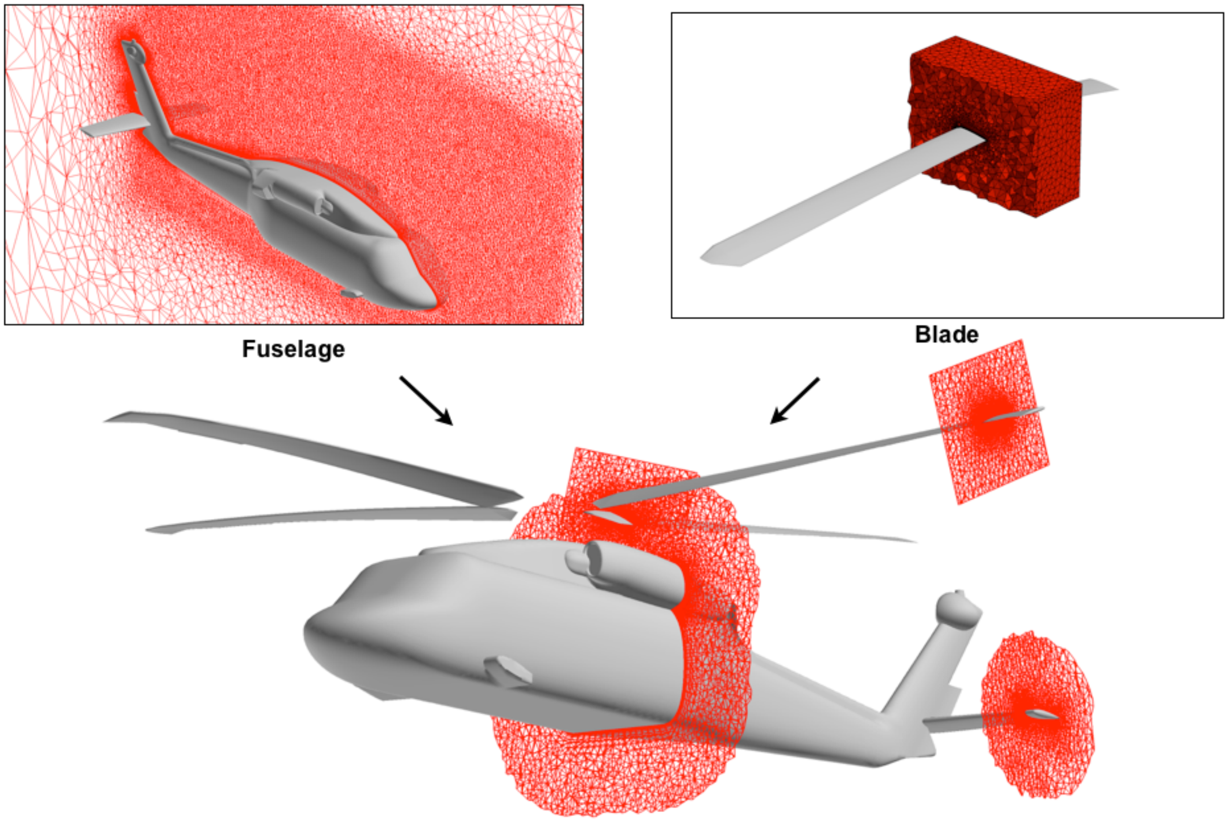


Figure 9: Unstructured mesh preprocessing; component meshes copied, moved, combined, and subsetted.

Helios v3 introduces parallel mesh preprocessing to avoid this limitation. The process works as follows. A user generates component meshes (e.g. rotor blade, hub, fuselage) using a standard unstructured mesh generator and applies the Helios GUI to specify the initial orientation. Combining and subsetting are done as before but all other preprocessing operations are done in parallel. This includes constructing the implicit lines in mesh boundary layer regions for the flow solver, extracting all other data-structures required by the solver, partitioning the unstructured meshes and computing the distance function.

The parallel preprocessor first reads the subset unstructured mesh system from a single file and distributes the mesh across the available processors. After the implicit line data structures are constructed, the edge-based graph is extracted from the unstructured mesh and contracted and weighted based on the implicit line data structures in order to ensure lines do not get split across processors in the final partitioning process. The weighted graph is then equidistributed across the available processors before calling the partitioner in order to improve the quality of the final partitions. The parallel graph partitioner is then called as a library, and the returned partition information is applied to the original mesh configuration by decontracting the graph. The mesh is then redistributed according to the newly determined partition information. Two different parallel graph partitioners are supported: ParMetis developed at the University of Minnesota[18] and Scotch, developed at INRIA in France.[19] The performance of these partitioners is illustrated in Table 3 for an 18 million point unstructured mesh about a UH-60 rotor configuration. Version 3 of ParMetis was initially found to produce poorly load balanced partitions and occasionally generated one or more empty partitions. Subsequently, Version 4 of ParMetis was found to produce much higher quality partitions. Version 5.11 of Scotch was generally found to produce equal or better quality partitions compared to ParMetis V4. It should be noted that the behavior of these commonly used partitioners must be investigated carefully since their performance is likely different in our implementation due to the use of weighted contracted graphs that is required due to the presence of a local line solver.

		Parmetis v3.1.1		Parmetis v4.0.2		Scotch v5.1.11	
np		min	max	min	max	min	max
32	nnode	283000	699973	46375	600548	569148	574196
	nedge	1278581	3353778	2135924	2766060	2473157	2674488
	ncell	736053	2113173	1291246	1703765	1406279	1637812
64	nnode	76027	357654	218570	300268	284029	287556
	nedge	322923	1806429	940934	1457050	1214642	1430894
	ncell	165127	1172754	511533	925679	677269	9331162
96	nnode	0	244368	152303	200141	189268	191874
	nedge	0	1388703	686124	1057579	775824	1067411
	ncell	0	979804	373601	742850	422769	768931
128	nnode	0	188554	113337	150087	141942	143942
	nedge	0	1039837	512343	799276	587506	784186
	ncell	0	719849	285527	575254	323154	546264
128	nnode	20685	121117	73275	98033	92042	94782
	nedge	89580	748420	318453	613857	379159	571423
	ncell	44252	588292	166667	478881	201867	435925

Table 3: Performance of different unstructured mesh partitioning packages; comparison of the min/max nodes, edges, and cells across partitions.

Once the mesh has been partitioned and redistributed, the distance function must be computed for turbulent flow cases. Brute force and octree distance function approaches were developed and tested, but the favored implementation is based on the solution of the Eikonal equation.[20, 21] The overall partitioning process and Eikonal equation-based distance function calculation scales reasonably well, as illustrated in Fig. 10, for the 18 million point UH-60 mesh using up to 196 processors.

The Eikonal distance function solution can consume a significant amount of time, sometimes up to 75% of the total preprocessing time. Therefore, the process is automated by saving the distance function values to a file once they have been computed. When the same mesh is to be re-used in a computation, possibly on different numbers of processors, if a valid distance function file is available, the file is read in and distributed to the new mesh partitions, and the distance function calculation is bypassed. In this mode, the preprocessor can run in about one-quarter the time. For example, the total time to partition a 13.8M node mesh on 256 processors of a standard linux cluster with fast ethernet connectivity was about 5 minutes. The parallel partitioner required approximately 30 seconds while the distance function required about 90 seconds, the rest of the time was spent in IO. When invoked a second time, where the saved distance function file is used the actual partitioning time requires only about 30 seconds.

The parallel preprocessing overhead is trivial when compared to typical Helios run times, which are typically measured in hours to days. The distance function need only be computed once for a given mesh configuration, it does not need to be recomputed when changing processor counts. Once a grid is partitioned for a certain number of processors it is written to disk and may be reused upon restart or for another run on the same number of processors. Work is currently underway to investigate partitioning strategies for much larger core counts, and to further accelerate the distance function calculation.

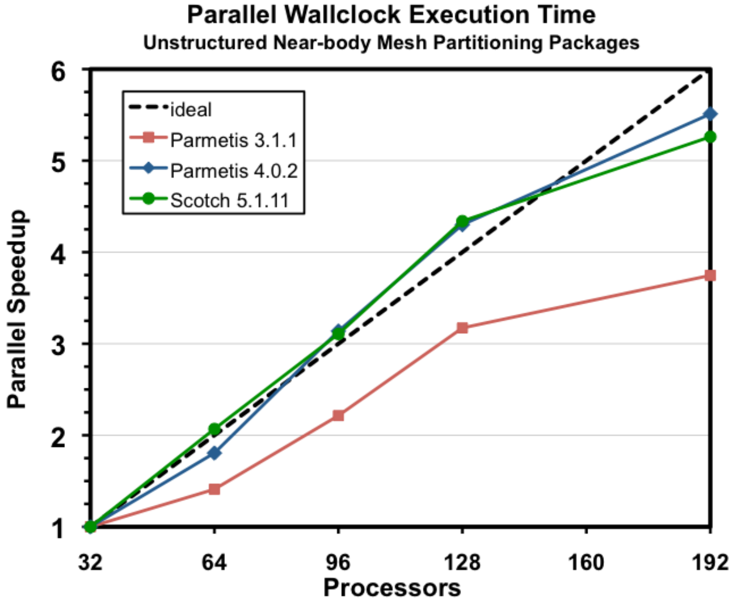


Figure 10: Wallclock scalability of Eikonal distance function calculation.

III. Results

In this section we show results for some complex configurations that have been run recently with Helios. In particular, multi-rotor configurations, loosely-coupled CSD/CFD rotor-fuselage computations, and evaluation of the common comprehensive analysis interface. Measured computational performance on state-of-the-art HPC systems is given in the final sub-section.

III.A. Multi-Rotor

Many problems in rotorcraft require analysis of multi-rotor configurations. Examples include modern main-rotor tail-rotor interactions, older tandem configurations like the Boeing CH-47, or newer co-axial configurations like the Sikorsky X2. Hence, the ability to handle multiple rotors is a key requirement for Helios.

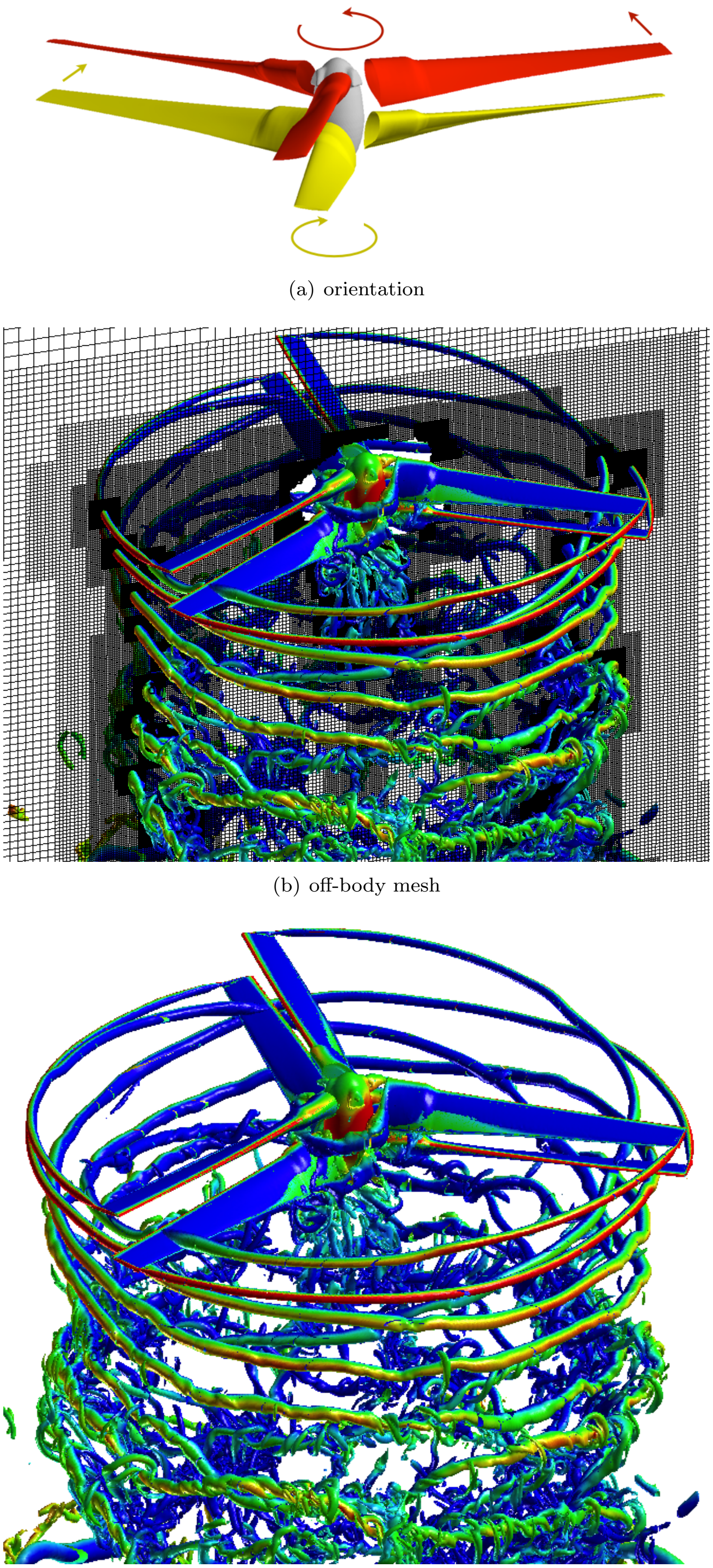
Helios v2 introduced the ability to handle multiple rigid rotors. As an example, Fig. 11 shows a calculation using a co-axial configuration involving two TRAM rotors shown in Fig. 11 (see further discussion about TRAM in Sec. II.A.2). The spin orientation is illustrated in (a). The off-body mesh is set to adapt to Q-criterion every 10 timesteps or 2.5^o azimuth, the resulting wake mesh is shown in (b), and the resulting complex wake solution is shown in (c).

Helios v3 extends this capability to handle pitching blades, providing the ability to prescribe a collective control input. It is also able to handle elastic deforming blades from comprehensive analysis, as long as the package supports multiple rotors.

III.B. Rotor-Fuselage Loose Coupling

Loose coupling between CFD and comprehensive analysis codes is necessary to accurately predict the periodic characteristics of a rotorcraft in steady forward flight. The comprehensive code provides the time-dependent collective pitch and flapping motion as well as the structural dynamics solution while CFD provides a solution to the complex aerodynamic flowfield.

Isolated rotor loose-coupling calculations were demonstrated in Helios v1.[4] Helios v2 introduced rotor-fuselage capability,[7] although only prescribed motion results were presented as validation. A number of refinements and improvements to the loose coupling have been introduced in Helios v3. Here, we present a summary of results for the HART-II test.



(a) orientation

(b) off-body mesh

(c) solution

Figure 11: TRAM coaxial solution.

The HART II test was conducted in the German-Dutch Wind tunnel (DNW) using a 40% Mach-scaled Bo105 model rotor in descending flight. It is a widely-used test database for rotary-wing forward flight analysis because it provides high-quality measured data of blade pressures, wake Particle Image Velocimetry (PIV), Stereo Pattern Recognition (SPR) blade deflection measurement, and noise measurements. Further details about the test and loose-coupling isolated rotor computations with Helios are described by Lim et al.[23] We present a summary of the HART-II rotor-fuselage analysis recently carried out with Helios v3. Further details are provided in the article by Jayaraman[24] at this meeting.

The rotor-fuselage mesh system is shown in Fig. 12. The near-body (a) fuselage mesh contains 3.6M nodes and the rotor contains 3.3M nodes. The adaptive off-body (b) and (c) mesh contains 8 levels of grid resolution with finest level spacing of 2.5% chord. Since we are not interested in the flowfield well downstream of the rotor, refinement is cutoff shortly behind the fuselage.

The case is run using a timestep of $\Delta\Psi = 0.1°$. The near-body BDF2 time-stepping scheme applies 25 sub-iterations per timestep, the off-body solver applies 6 sub-iterations (determined automatically based on CFL conditions of the finest level grid). Refinement is applied every 2.5 degrees azimuth, or every 25 timesteps, and the AMR scheme adapts to a non-scaled Q-criterion value of $q = 0.0005$. Because the off-body grid is so fine we run the case as a sequence, building up to the final 8L off-body grid system. First, it is run for 4 revolutions using a 6L fixed off-body grid, converging a loose-coupled result with a relatively coarse wake of spacing Δx=10%c. Next, it is run for an additional revolution using a 7L AMR off-body grid, providing wake resolution of Δx=5%c. Last, the 6th and final revolution is run using the 8L AMR grid with spacing Δx=2.5%c. This entire sequence is run on 256 cores of the MHPCC mana system.

The off-body grid size and computational requirements grow commensurately over the course of the run sequence. For the first 4 revs the 6L fixed off-body grid system contains 1138 blocks and 31.8M points, the off-body solution constitutes only 11% of the total time. For the 5th rev with the 7L AMR grid, the size grows to 3350 blocks and 55.8M points and the off-body solution makes up 23% of the total solution time. For the 6th rev with the fine 8L AMR grid, the size again grows to 11800 blocks and 313.7M points and the off-body solution constitutes 74% of the total solution time. At more than 300M points the 8L AMR grid may seem excessively large, but a comparable fixed 8L off-body mesh was found to contain 14000 blocks and 1665M points. So use of AMR provides a computational savings of 5.3X.

Inclusion of the fuselage in the CFD model improves the computational prediction. Figure 14 compares the computed normal force $C_N M^2$ at the $r/R = 0.87$ spanwise station to measured data over the full azimuth range. Compared to the isolated rotor results[23] the rotor-fuselage model better predicts the blade-vortex interactions on the advancing side ($\Psi = 0° - 90°$) and avoids the phase lag experienced by the isolated rotor case on the retreating side ($\Psi = 150° - 180°$). Additional details and results for the loose-coupling HART-II rotor fuselage configuration are described in an accompanying paper[24] presented at this meeting.

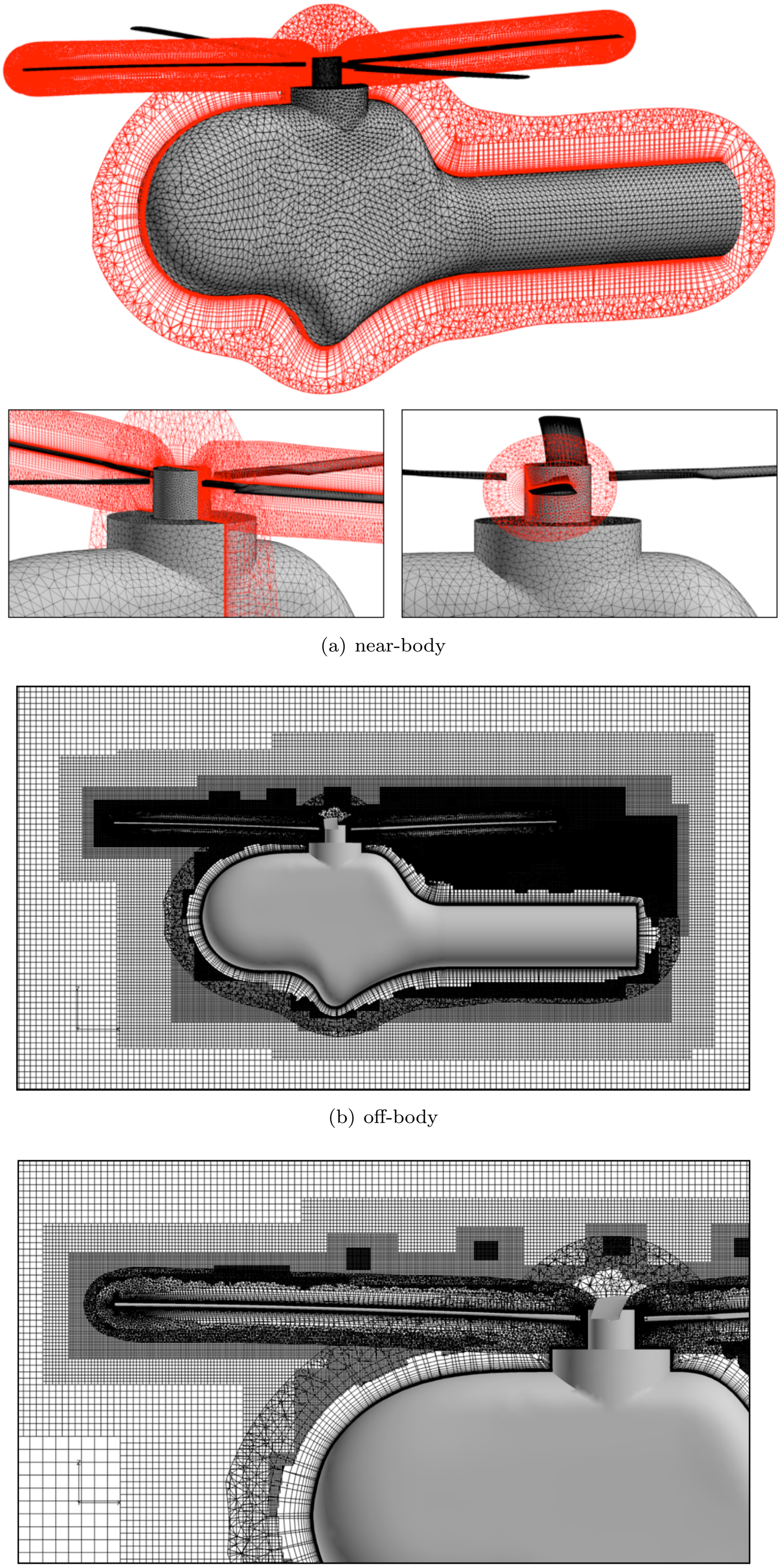
III.C. Common Comprehensive Analysis Interface

The comprehensive analysis (CA) software interface used in Helios v3 is compatible with both the RCAS[5] and CAMRADII[25] software packages. As an example, Figure 15 shows a comparison of the computed normal force, pitching moment, and chord force at three stations for a CFD/CSD coupled Helios calculation of the UH60 airloads program high speed C8534 flight test.[26] Details about the CFD grid and setup are provided in Ref.[22]

Different organizations often prefer particular CA analysis packages based on history and knowledge base for a particular vehicle. Hence, having the ability to readily plug-in different CA software packages enables greater versatility. It is also helpful from a validation perspective to compare CFD/CSD results using alternative CA software.

III.D. Computational Performance

Helios v3 performance is reported for CFD/CSD coupled calculations on three DoD DSRC systems. First, *mana* is a 1,152 node Dell PowerEdge system located at MHPCC. Each node has two 2.8 GHz quad-core Intel Nehalem processors and 24 GB RAM, providing a total of 9,216 compute cores with 3 GB/core memory. Second, *garnet* is a 1,260 node Cray XE6 system located at ERDC. Each node has 2.4 GHz AMD opteron processors with 16 cores and 32GB RAM per node, providing a total of 20,160 cores with 2 GB/core memory. Third, *raptor* is a 2732 node Cray XE6 system located at AFRL with the same per core processor and memory profile as garnet, but with 43,712 total cores.

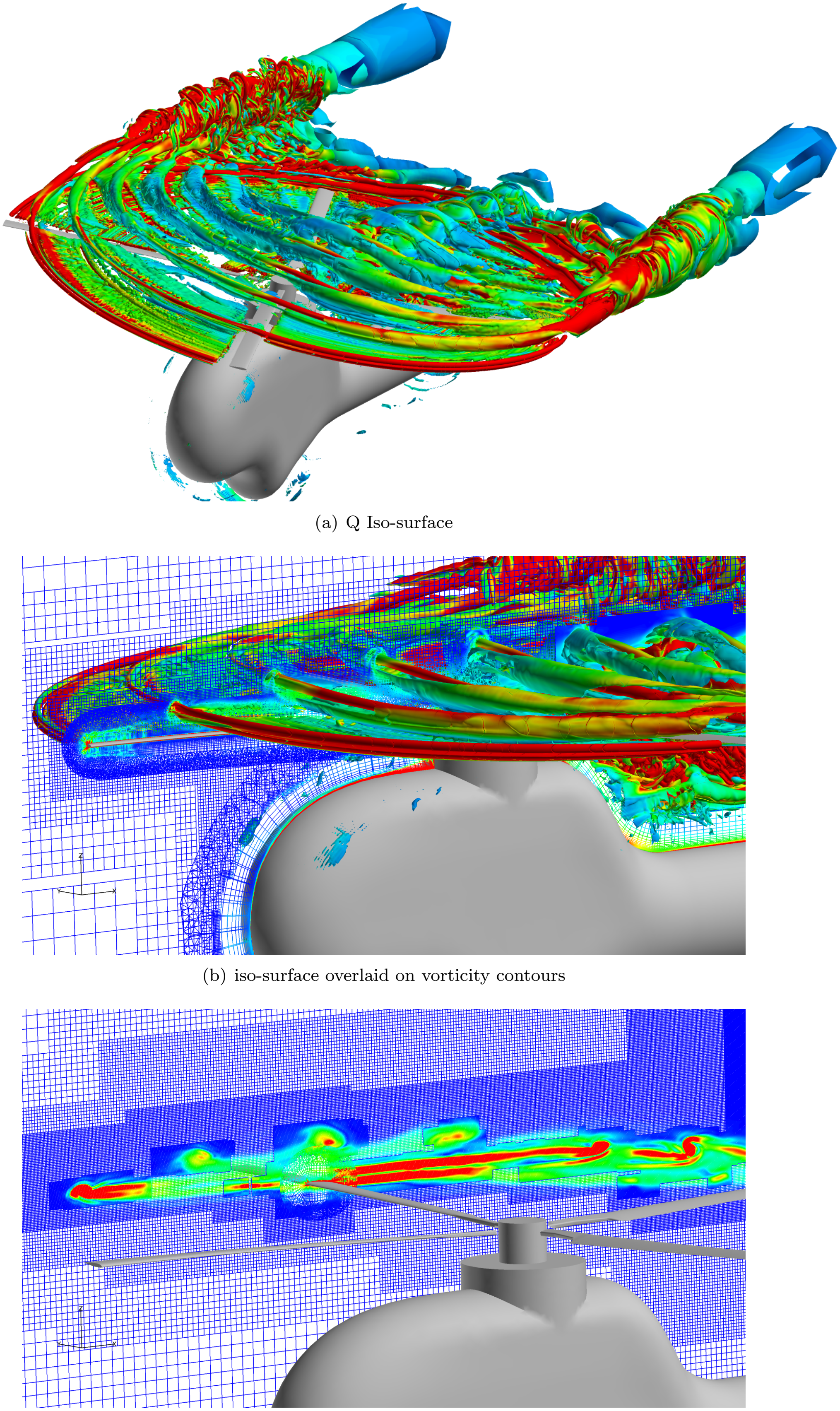


(a) near-body

(b) off-body

(c)

Figure 12: HART-II rotor-fuselage near-body and off-body grid systems.



(a) Q Iso-surface

(b) iso-surface overlaid on vorticity contours

(c) vorticity contours at $y = 0.7$ spanwise location

Figure 13: HART-II solution with AMR off-body grid using finest spacing $\Delta x = 2.5\%$c.

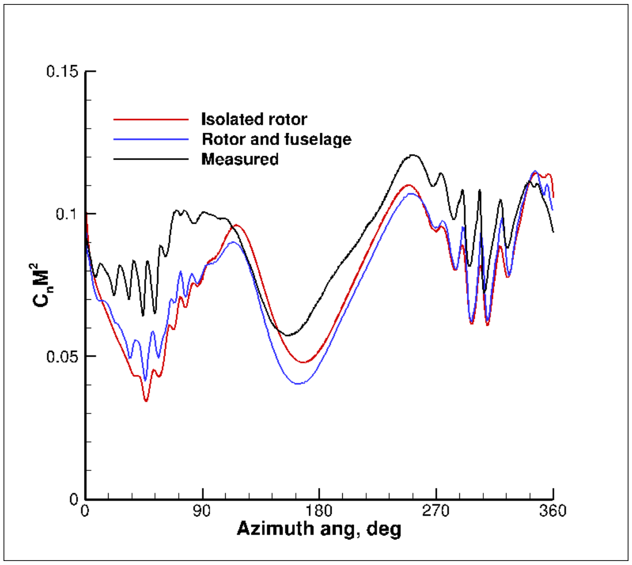


Figure 14: HART-II measured and computed normal force $C_N M^2$ at spanwise station r/R=0.87. CFD/CSD loose coupling isolated rotor vs. rotor with fuselage.

The test case is a UH60 CFD/CSD loose-coupling calculation. The near-body unstructured rotor mesh contains 15.4M nodes, the fixed off-body mesh has spacing of 5%c and contains 146M gridpoints. AMR was not applied for the off-body solution. All calculations were run on 512 cores for 1 rev (3600 steps) using a timestep of $\Delta\Psi = 0.1°$. The near-body solution used 25 sub-iterations per timestep while the off-body solver applied 7 sub-iterations (determined automatically based on CFL conditions of the finest level grid). Loose coupling iterations with CSD was applied every 90° azimuth. Further details about the case can be found in Ref.[22]

	mana	*garnet*	*raptor*
Component	Dell Cluster	Cray XE6	Cray XE6
Near-body unstructured	18.5hr (65%)	25.9hr (65%)	29.7hr (67%)
Off-body Cartesian	8.1hr (28%)	12.2hr (29%)	12.2hr (27%)
Domain connectivity	1.5hr (5%)	1.4hr (4%)	1.7hr (4%)
Struct dyn & trim	0.04hr (0.1%)	0.1hr (0.2%)	0.1hr (0.2%)
Other	0.4hr (1%)	0.9hr (2%)	1.0hr (2%)
Total	28.5hr	40.1hr	44.6hr

Table 4: Helios per-rev computation time on different HPC architectures for CFD/CSD fine mesh UH60 loose-coupling calculation. 512 cores used on each machine.

Table 4 shows the wallclock run time for this one rev calculation. Computational cost of the main computational kernels are broken down; near-body unstructured solution, off-body Cartesian solution, domain connectivity, and the comprehensive analysis solution for structural dynamics and trim. The largest contributor to the computation time is the near-body unstructured solver. Its per-point cost is higher than the structured Cartesian code and it additionally applies about four times the number of sub-iterations at each timestep. Domain connectivity costs are quite reasonable at around 4%–5%, and the cost of the structural dynamics and trim computation performed by comprehensive analysis is trivial relative to the CFD costs.

A typical loose-coupling calculation with a grid at this resolution requires between five and ten revolutions to reach a converged periodic steady flight condition. An unsteady maneuvering flight simulation requires around forty revolutions. Thus, at a per-rev cost of 29hr − 45hr, as observed for the case above, the computational cost of a full rotorcraft simulation is measured in weeks rather than hours or days. Clearly, CFD/CSD coupled rotary-wing simulations of this type need to be able to take advantage of continued

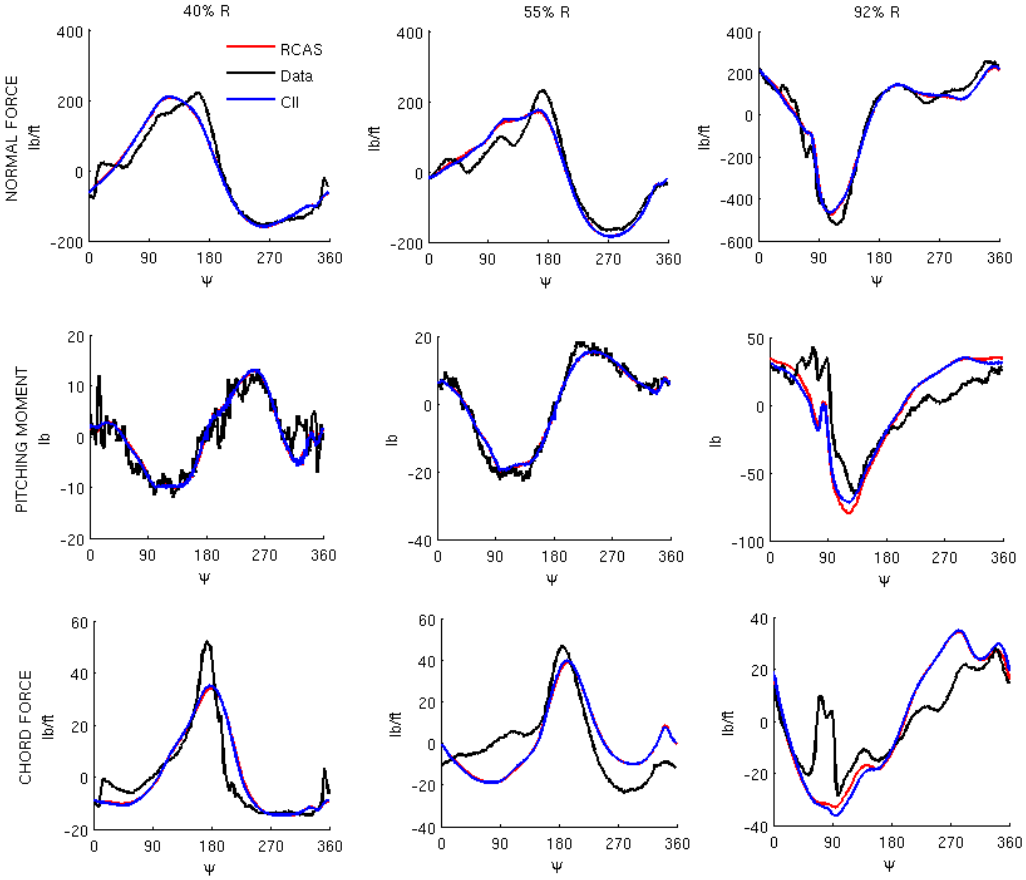


Figure 15: UH60 8534 high speed case. Helios with RCAS and CAMRAD II comprehensive analysis packages.

advances in HPC hardware and software technology in order to bring down the cost in order to be routinely used for analysis and design.

IV. Summary

This paper presents a summary of the Helios v3 capabilities. The three major new capabilities released in this version are DES turbulence modeling, error-driven grid off-body grid refinement, and parallel unstructured grid pre-processing. DES turbulence modeling should improve the accuracy of computations for non-attached shed wakes, such as rotors experiencing dynamic stall. Error-driven grid refinement will improve automation of AMR in the off-body solution by avoiding the tuning typically required when refining to vorticity or Q-criterion. Parallel grid preprocessing will eliminate the need for partitioning unstructured grids as a pre-process which will enable more routine use of larger near-body meshes and reduce the number of problem setup tasks required of the user.

V. Acknowledgments

Material presented in this paper is a product of the CREATE-AV (Air Vehicles) 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 authors gratefully acknowledge the contributions by Dr. Venkateswaran Sankaran of the U.S. Air Force Research Lab (AFRL) at Edwards Air Force Base, and by Dr. Hossein Saberi and Dr. Wei-Bin Chen of Advanced Rotorcraft Technology, Inc. Compute resources were provided by the DoD High Performance Computing and Modernization Office (HPCMO) DOD Shared Resource Centers (DSRC).

References

[1] Strawn, R.C., F.X. Caradonna, E.P.N. Duque, "30 years of Rotorcraft Computational Fluid Dynamics Research and Development," *Journal of the American Helicopter Society*, Vol. 51, (1), Jan 2006, pp. 5–21.

[2] Datta, A., and W. Johnson, "An Assessment of the State-of-the-art in Multidisciplinary Aero-mechanical Analysis," American Helicopter Society Technical Specialists Meeting, San Francisco, January 22-24, 2008.

[3] 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.

[4] 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.

[5] Saberi, H., Khoshlahjeh, M., Ormiston, R., and Rutkowski, M. J., "Overview of RCAS and Application to Advanced Rotorcraft Problems," 4th AHS Decennial Specialist's Conference on Aeromechanics, San Francisco, CA, January 21-23, 2004.

[6] 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.

[7] 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.

[8] S Spalart, P. R., W.H. Jou, M. Stretlets, and S.R. Allmaras, "Comments on the Feasibility of LES for Wings and on the Hybrid RANS/LES Approach", Proceedings of the First AFOSR International Conference on DNS/LES Aug 4-8, 1997, Ruston, LA. In: Advances in DNS/LES, C. Liu and Z. Liu Eds., Greyden Press, Columbus, OH, USA (1997).

[9] D. J. Mavriplis, J. Pelaez and O. Kandil, "Large Eddy and Detached Eddy Simulations using an Unstructured Multigrid Solver," Proc. of the 3rd AFOSR Int. Conf. on DNS/LES, Arlington, TX, Aug 2001, C. L. Liu, L. Sakell, and T. Beutner Eds (2001).

[10] Spalart, P. R., S. Deck, M.L. Shur, K.D. Squires, M. Strelets, and A. Travin, "A New Version of Detached-Eddy Simulation, Resistant to Ambiguous Grid Densities," Theoretical and Computational Fluid Dynamics, Vol. 20, No. 3, pp.181-195., 2006.

[11] Sheldahl, R. E., and P.C. Klimas,"Aerodynamic Characteristics of Seven Airfoil Sections Through 180 Degrees Angle of Attack for Use in Aerodynamic Analysis of Vertical Axis Wind Turbines," SAND80-2114, Sandia National Laboratories, Albuquerque NM, Mar 1981.

[12] A. Wissink, S. Kamkar, T. Pulliam, J. Sitaraman, and V. Sankaran, "Cartesian Adaptive Mesh Refinement for Rotorcraft Wake Resolution," AIAA-2010-4554, AIAA 28th Applied Aerodynamics Conference, Chicago IL, June 2010.

[13] Marcum, D. L., "Efficient Generation of High-Quality Unstructured Surface and Volume Grids," Engineering with Computers, Vol. 17, No. 3, October, 2001.

[14] Chaderjian, N. M., and P. G. Buning, "High Resolution Navier-Stokes Simulation of Rotor Wakes," American Helicopter Society 67th Annual Forum, Virginia Beach, VA, May 3–5, 2011.

[15] Kamkar, S. J., A.M. Wissink, V. Sankaran, A. Jameson, "Automated Off-Body Cartesian Mesh Adaption for Rotorcraft Simulations," AIAA-2011-1269, AIAA 49th Aerospace Sciences Meeting, January 2011, Orlando FL.

[16] Kamkar, S. J., A.M. Wissink, V. Sankaran, A. Jameson, "Feature-Driven Cartesian Adaptive Mesh Refinement for Vortex-Dominated Flows," *Journal of Computational Physics*, Vol. 230, No. 16, July 2011, pp. 6271-6298.

[17] Kamkar, S. J., and A.M. Wissink, "An Automated Adaptive Mesh Refinement Scheme for Unsteady Aerodynamic Wakes," AIAA 50th Aerospace Sciences Meeting, January 2012, Nashville TN.

[18] Karypis, G., and V. Kumar, "A Fast and High Quality Multilevel Scheme for Partitioning Irregular Graphs," *SIAM Journal on Scientific Computing*, Vol. 20, No. 1, pp. 359–392, 1999.

[19] Chevalier, C., and F. Pellegrini, "PT-SCOTCH: a tool for efficient parallel graph ordering," *Parallel Computing*, 34(6-8), pp 318-331, 2008.

[20] Xia H., and P.G. Tucker, "Finite volume distance field and its application to medial axis transforms," *Int. J. Num. Meths in Eng.* Vol. 8, No. 6., 2010, pp. 535-554.

[21] Tucker, P. G., "Differential Equation Based Wall Distance Computation for DES and RANS," *Journal of Computational Physics*, Vol. 190, No. 1, Sept. 2003, pp. 229-248.

[22] Sitaraman, J., Potsdam, M., Jayaraman, B., Wissink, A., Datta, A., Mavriplis, D., and Saberi, H., "Rotor Loads Prediction Using Helios: A Multi-Solver Framework for Rotorcraft CFD/CSD Analysis," AIAA-2011-1123, 49th Aerospace Sciences Meeting, January 2011, Orlando, FL.

[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] 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 50th Aerospace Sciences Meeting, January 2012, Nashville TN.

[25] Johnson, W., "Rotorcraft aerodynamic models for a comprehensive analysis," American Helicopter Society 54th Annual Forum, Washington D.C., May 1998.

[26] R. M. Kufeld, D. L. Balough, J. L. Cross, K. F. Studebaker, C. D. Jennison, and W. G. Bousman, "Flight Testing of the UH-60A Airloads Aircraft," Presented at the 50th Forum of the American Helicopter Society, Washington, DC, May 1994.