Code-to-Code Comparison of CFD/CSD Simulation for a Helicopter Rotor in Forward Flight

Jasim Ahmad*
NASA Ames Research Center, Moffet Field, CA 94035

Robert T. Biedron†
NASA Langley Research Center, Hampton, VA 23681

Abstract

Two unsteady Reynolds-averaged Navier-Stokes solvers are used to compute the rotor airloads on the UH-60A rotorcraft at several flight conditions across the flight envelope. One code, OVERFLOW, solves the flow equations using either structured grids, or a combination of structured and Cartesian grids. The other solver, FUN3D, uses unstructured grids. Both flow solvers are coupled to the same rotorcraft comprehensive code (CAMRAD II) in order to account for trim and aeroelastic deflections, and both utilize the same loose coupling scheme to transfer data between the flow solver and the comprehensive code. In the process of performing the code-to-code comparison, several small but important details are examined that may sometimes be overlooked when comparing results from rotorcraft simulations using different codes. Computed normal force, pitching moment, and chord force are compared between codes, and also with flight data.

Nomenclature

α_s	shaft angle of attack [°]
β	side slip angle (+ nose left) [°]
μ	advance ratio, M/M_{tip}
Ψ	rotor azimuth [°]
ρ	density [slugs/ft^3]
σ	rotor solidity, $nc/\pi R$ (0.0826)
a	freestream speed of sound [ft/s]
c	local chord [ft]
C_T	thrust coefficient [lb]
M	freestream Mach number
M^2C_c	sectional chord force coefficient, F/$(\frac{1}{2}\rho a^2 c)$
M^2C_m	sectional pitch moment coefficient, PM/$(\frac{1}{2}\rho a^2 c^2)$
M^2C_n	sectional normal force coefficient, N/$(\frac{1}{2}\rho a^2 c)$
M_{tip}	tip Mach number
n	number of rotor blades (4)
R	blade radius (26.833) [ft]
r	radial coordinate [ft]
CAMRAD II	Comprehensive Analytical Model of Rotorcraft Aerodynamics and Dynamics II
F	sectional chord force [lb]
N	sectional normal force [lb]
PM	sectional pitch moment, [ft-lb]

Introduction

The analysis of rotorcraft aerodynamics is quite challenging due to the coupling between the structural deformations of the blades and the complex flowfield. The blade motions are required to either trim the vehicle in level flight or to effect a change in a maneuver. To model aerodynamics and structural dynamics in the analysis of rotorcraft systems, a number of 'comprehensive codes' have been developed. In order to perform the analysis of these systems in a timely manner, comprehensive codes usually employ simplified methods or models within each of the various analysis tasks. For example, nonlinear beam models are used to model the structural dynamics of the blades rather than a more complete finite-element model. For aerodynamic analysis, lifting-line models supplemented by two-dimensional airfoil lookup tables and wake models are often used. While a nonlinear beam is usually a good model for a long, slender rotor blade, for many flight conditions the simpler aerodynamics models are found to lack sufficient fidelity. To overcome the aerodynamic model deficiencies, the idea of correcting comprehensive codes with input from Computational Fluid Dynamics (CFD) codes is widely used.[1–4] The idea is to replace, through an iterative procedure, the aerodynamics computed by the comprehensive code with the aerodynamics computed by the CFD code, while retaining all the other analysis capabilities of the comprehensive code. At the same time, the effects of the structural dynamics of the blades, as well as the trim of the rotor, are brought into the CFD simulation. In this context, the primary contribution of the comprehensive code is to provide

*Aerospace Engineer
†Senior Research Scientist, Senior Member AIAA

a Computational Structural Dynamics (CSD) model and a mechanism for trim. The terms comprehensive code and CSD code are used interchangeably in the remainder of the paper.

Under the NASA Subsonic Rotary Wing Project two general-purpose Reynolds Averaged Navier Stokes (RANS) solvers, OVERFLOW[5] and FUN3D[6] are being modified for and applied to a range of rotorcraft problems. The goal is to produce validated, high-fidelity tools for analysis and design. The two codes share a common file interface for coupling with CSD codes. The purpose of this paper is to perform a detailed comparison of the computed rotor airloads from these two CFD solvers, both coupled to the same CSD code, CAMRAD II.[7] The computations will be performed for several cases of interest for the UH-60A rotorcraft. The UH-60A is an excellent configuration for CFD/CSD validation, as an extensive set of flight-test measurements of aerodynamic and structural loads is available in the UH-60A database.[8,9]

The purpose of performing a detailed code-to-code comparison, along with comparison of computed results with flight-test data, is to validate both codes in depth. In addition, we will examine some details of the coupling process that, while important, are often overlooked or go unreported. By utilizing the same CSD solver (and the same inputs to the CSD solver) for both CFD codes, any observed differences are attributed to the CFD side of the coupled simulations.

CFD Codes

OVERFLOW and FUN3D have had a long history of development. The codes have a number of common attributes, though they utilize distinctly different computational algorithms. OVERFLOW is a Reynolds-averaged Navier-Stokes (RANS) code that uses an overset structured/Cartesian grid system while FUN3D is an unstructured RANS solver that can optionally utilize overset (unstructured) grids. These widely-used codes have very general capabilities for accommodating moving and deforming bodies, and as such are very suitable for rotorcraft simulations. Of the two, OVERFLOW has been used more extensively for rotorcraft analysis. Both codes have interfaces with CAMRAD II (and other rotorcraft analysis codes) to provide blade deformation and trim based on CSD computation.

OVERFLOW is a finite-difference, node-based solver. User-supplied, body-fitted structured grids are employed near solid surfaces, with automatically-generated background Cartesian grids filling the remainder of the computational domain. Higher-order spatial differences, up to 5^{th} order, are available for interior points. Physical boundary conditions are implemented with reduced order. The code has several upwind flux schemes to improve the numerical accuracy for high speed applications as well as for vortex dominated flows. The results presented here were obtained using one of the HLL[10] family of upwind approximate-Riemann flux algorithms, namely, the contact-preserving HLLC variant,[11] with nominally 5^{th} order accuracy in space for the inviscid fluxes. Reference 5 provides details of these algorithms. To advance the solution in time, an implicit, second-order backward difference scheme is used in which the nonlinear system of equations is linearized at each time step. In this study, the resulting system of linear equations is solved using symmetric successive overrelaxation (SSOR), which eliminates factorization errors at the expense of more computational work and memory per time step. Within OVERFLOW, several time-stepping schemes are available, including dual-time schemes. This study used Newton iteration scheme. For this simulation, OVERFLOW used anywhere between 15 to 35 subiterations at each time step. A number of turbulence models are available in OVERFLOW. Here, the standard one-equation model of Spalart and Allmaras[12] is utilized. For inter-grid communications between overset meshes, OVERFLOW can use either Pegasus or Domain Connectivity Function (DCF) using the X-Rays method of hole cutting.

FUN3D is a finite volume, node-based solver with 2^{nd} order spatial accuracy. The code utilizes unstructured grids, which may be comprised of tetrahedral, prismatic, hexahedral, or pyramidal cells. Typical grids utilize prisms near solid surfaces and tetrahedra to fill the remainder of the computational domain. FUN3D also has a number of flux functions available. In this study, the standard Roe scheme[13] is employed, with a least-squares approach used to evaluate the gradients needed for second-order reconstruction of the inviscid fluxes. Viscous terms are evaluated with second-order accuracy using gradients computed using the Green-Gauss theorem. For non-tetrahedral meshes, the Green-Gauss gradients are combined with edge-based gradients to avoid odd-even decoupling. In this approach, the full viscous terms are retained, i.e., no thin-layer approximations are made. The Spalart-Allmaras turbulence model is also used for the FUN3D results presented here. To advance the solution in time, an implicit, dual-time stepping scheme is used. The nonlinear equations are linearized at each time step, with subiterations used to iteratively solve the nonlinear system. A point Gauss-Seidel scheme is used to solve the resulting linear system of equations at each subiteration step. Within FUN3D, the iterative or pseudo-time step is based on a constant CFL number rather than the physical time step. The particular scheme used in this study utilizes backward differences in time and although formally second-order accurate, has a smaller leading-order truncation error than the standard second-order backward difference scheme.[14,15] FUN3D can utilize either a fixed number of subiterations per step, or can use an estimate of the temporal error to determine a suitable level of subiteration convergence. The latter method is employed here. Subiterations are terminated when the subiteration residual (i.e., the error in solving the nonlinear system of equations) drops to less than one percent of the temporal error estimate, or a maximum of 40 subiterations are performed. The goal is to ensure that the error in solving the nonlinear system of equations at each time step is held below the error associated with the temporal discretization. This allows the design order of the time-advancement scheme to be maintained. For the overset meshes needed for rotorcraft simulations, the DiRTlib[16] and

SUGGAR++[17] codes are used in conjunction with FUN3D to facilitate communication between disparate zones in the mesh.

Fluid/Structural Coupling

The rotorcraft CSD code used for this study is CAMRAD II.[7] The aerodynamics modules within CAMRAD II are based on lifting-line models utilizing airfoil tables, coupled with wake models. Within CAMRAD II, each blade is structurally modeled as a set of nonlinear beam elements. In addition to the structural dynamics modeling, CAMRAD II offers a trim capability. For the UH-60A simulations in this paper, a three-degree-of-freedom trim is utilized, with the solidity-weighted thrust coefficient, pitching moment, and rolling moment specified as trim targets within CAMRAD II. In turn, CAMRAD II provides the collective pitch, longitudinal cyclic, and lateral cyclic pitch angles.

The goal of the loose coupling approach is to correct the low-order lifting-line aerodynamics of the CSD code with the higher-fidelity Navier-Stokes aerodynamics of the CFD code. In the loose coupling approach, shown schematically in Figure 1, quarter-chord (c/4) blade motion data from the CSD solver (upper red box in the figure) and quarter-chord aerodynamic force and moment (F/M) data from the CFD solver (lower red box) are exchanged at periodic intervals, for example, once per revolution or more generally once per integer multiple of the blade passage. In a typical coupled simulation, the initial execution of the CFD code is carried out for one or two complete rotor revolutions, using blade deformations from a trimmed CAMRAD II solution with unmodified lifting-line (superscript LL Fig. 1) in aerodynamics. In subsequent coupling cycles, the flow solver is run for one-half revolution between coupling cycles (full revolution for some cases). At each coupling step, rotor loads from CFD are provided to CAMRAD II and in turn, CAMRAD II uses this CFD load to make a correction to its own lifting-line aerodynamics to retrim the rotor. At convergence, the CFD airloads fully replace comprehensive analysis airloads. The coupled FUN3D/CAMRAD and OVERFLOW/CAMRAD systems are tied together by separate interface codes that perform data translation between CFD and CSD. C-Shell scripts are used to manage the execution of the various codes, and to provide restart, post-processing, and archiving functions. FUN3D and OVERFLOW utilize their own shell scripts, although the fundamental input and output are common to both CFD codes.

Geometrical Considerations

Reference Blade Geometry

For the computed results presented below, both CFD solvers have utilized independently-created grids that define the rotor blade. For each solver, one reference blade grid is created, and that reference grid is replicated and positioned to define the four-bladed UH-60A rotor. The question then arises, how different are these independently-generated reference-blade geometries? Differences in reference-blade geometry will obviously lead to differences in solutions between the two codes. Although we may not be able to assess the sensitivity of the solution to the geometry for both solvers, we can at least quantify the differences in the geometries themselves. The differences (δ) in the y and z coordinates of the two reference-blade geometries are shown in Fig. 2. It is seen that maximum differences in the reference surfaces are generally very small, except in the region near the blade root. The actual blade surface was not defined in this region of the original blade model provided by Sikorsky and so the required closure in this region was done differently in each case. These differences in the root region are believed not to have a significant impact on the computed results as the root loading is generally very small.

Blade Motion/Deformation Verification

When utilizing a coupled CFD/CSD approach there are relatively complex geometry manipulations, for both input and output, that must be done for each time step within the CFD solver. Blade deformations from CAMRAD II are given as (x,y,z) deflections of the quarter chord along with Euler-angle rotations, as functions of span and rotor azimuth angle. These quarter-chord data, encapsulated in the upper red box of Fig. 1, are used by the CFD code to define the three-dimensional blade surface at each time step. The deformed blades are then rotated through the proper shaft azimuth angle to position each blade at the current instant of time. The deformations and Euler angles extracted from the CAMRAD II solution are contained in a 'motion file', which is in a common format utilized by both codes. To assess the differences in the way both flow solvers process this deformation data, a verification test was performed in which both solvers were given the same grid and motion file, and then the blades were run through each solver, accessing only the blade motion routines, without actually performing a flow solution. At selected points along the rotor disk, the deformed blade surfaces were output from each solver and compared. Fig. 3 shows the comparison, at a rotor azimuth of 180°, of the differences in y and z, normalized by the rotor radius. Comparisons at other azimuthal angles are similar. It can be seen that given the same grid and motion file, the two solvers produce virtually identical blade deformations. Maximum differences are on the order of 10^{-5}R - a few thousandths of an inch on the full scale UH-60A rotor. Thus, geometrical differences arising from how the blades are moved and deformed in each code will be negligible compared to differences that are inherent in the independently-generated reference blade grids.

Twist Distribution

As implemented, coupling with CAMRAD II requires that forces and moments from CFD be defined in a local, section-aligned coordinate system for multiple radial stations along the blade. The CFD codes must therefore take the computed forces and moments, defined at discrete surface points, and resolve them into this local coordinate system. This process is encapsulated in the lower red box of Fig. 1. From a geometrical standpoint, this requires the flow solver to determine the local quarter-chord location and local geometric twist, which in turn requires determination of the local leading and trailing edges.

With a structured grid, this is a more straightforward operation than with an unstructured grid since the trailing edge is always identified with a constant grid index and thus is explicitly set. With the trailing edge known, the leading edge can then be defined by finding the point having the largest distance from the trailing edge (at the fixed radial location in question). In structured grids used for rotorcraft simulation, the chordwise grid lines are either collocated with, or parallel to, the radial stations at which sectional force and moment data are gathered for CAMRAD II. In an unstructured grid, there is often no grid point that lies at the trailing edge, and there is no convenient means (such as a constant grid index) to identify it a priori. Furthermore, grid points do not follow constant radial lines in an unstructured grid. Thus FUN3D must perform an intersection with the blade surface at each predetermined radial station. Within that intersected section, a search must be performed to find either the aft-most point in the blade section (for an idealized sharp trailing edge), or identify two corner points and compute the midpoint between them (for the squared-off trailing edge typically found on production rotorcraft blades). Once the local trailing edge is found, the same process used by OVERFLOW to find the leading edge may be utilized.

To assess how closely the two CFD codes determine this local coordinate system, the resulting twist distributions in the reference blade from both codes are shown in Fig. 4, together with the twist distribution specified in the CAMRAD II input. It is seen that the extracted twist from both codes is very nearly identical. The twist extracted from either grid differs slightly from the twist in CAMRAD II in the region $0.2 < r/R < 0.4$; it is uncertain at this time which twist distribution is more faithful to the 'as-built' twist. Although the CFD-computed twist distributions are shown only for the undeformed reference blade, the twist distribution must be recomputed for the current deformed blade shape at every time step to determine the local section-aligned coordinate system in which loads are transferred back to CAMRAD II.

Grid Systems

The grid systems used to date by the two solvers are considerably different in size and type. The grid used for the FUN3D simulations consists of approximately 15 million nodes and 80 million cells. The number of unknowns in a FUN3D solution is proportional to the number of nodes. The grid is comprised of prismatic cells in the boundary-layer region of the rotor blades, tetrahedral cells outside the boundary layer, and a small number of pyramidal cells in the transition region between prismatic cells and tetrahedral cells. The background grid into which the rotor-blade grids are overset is comprised entirely of tetrahedral cells. Near the outer boundary of the blade grids, the average spacing between nodes is approximately ten percent of the tip chord, though, due to the nature of the unstructured grid, this is only an estimate. As can be seen in Fig. 5, the spacing in the (blue) background grid is more highly clustered in a region surrounding the rotor than away from it to better resolve the vortices shed by the blades. The unstructured grids used in this study were generated using the advancing-layer and advancing-front grid generation software package.[18]

The grid used for the OVERFLOW simulations consists of approximately 71 million nodes with a nearly identical number of cells. The number of unknowns in an OVERFLOW is also proportional to the number of nodes. The large number of grid points used in this simulation is in anticipation of running a future Detached Eddy Simulation (DES) that would also include fuselage. Although not shown, results from OVERFLOW were also obtained on a grid with approximately 35 million nodes, and were virtually identical to those obtained using the larger mesh. The grid is comprised of curvilinear, 'O-mesh' grids surrounding each blade, with the blade grids overset into automatically-generated Cartesian grids. Each blade grid is comprised of three separate grids, one for the main portion of the blade, and separate cap grids for the tip and the root. The off-body Cartesian meshes are generated at various levels of resolution starting from 'Level 1' near the blades, to 'Level n' in the farfield. For this study, the 'Level 1' grid spacing is set to ten percent of the tip chord, similar to that used in the unstructured FUN3D mesh. A slice through the structured/Cartesian grid system is shown in Fig. 6, where background grids through Level 5 are visible. The overset mesh used in OVERFLOW simulations are generated using Chimera Grid Tools (CGT).[19] The CGT contains very efficient and modular grid scripting libraries used for grid manipulation, generation, reorganization, overset hole cutting surfaces, and producing flow solver input files. This scripting procedure allows production of high-quality surface meshes and subsequent volume meshes with a set of input parameters. These scripts and input parameters enable modifications to the existing geometry and grid system to be made automatically and quickly.

Computational Results

In this section we compare the computational results obtained from OVERFLOW/CAMRAD and FUN3D/CAMRAD with flight data. The focus is on rotor airloads. The UH-60A helicopter flight tests were part of the joint Army/NASA UH-60A Airloads Program and are widely used for validation of rotorcraft CFD simulation. In the flight test, the helicopter was extensively instrumented with 242 pressure transducers at the nine radial locations shown in Fig. 7. Flight data ranging from high-torsional load maneuvers to very low-speed level flight were obtained. The details of all these flight conditions can be found in Ref. 9. For this paper, two level flight cases were simulated.

When performing the computations, "best practices" were utilized for each code. In particular, the OVERFLOW simulations were performed using a time step corresponding to 0.25° azimuth change, with a fixed number of subiterations at each time step. FUN3D simulations were performed using a time step corresponding to 1.0° azimuth change, with a variable number of subiterations at each time step to attempt to drive the nonlinear residual to a small fraction (0.01) of the temporal error.

In terms of computational resources, FUN3D simulations on a grid of 15 million nodes required 6.8 hours per half revolution (a typical interval between data exchanges with CAMRAD-II), using 289 Westmere cores of NASA's Pleiades supercomputer. For a grid of 71 million nodes, a half revolution in OVERFLOW required 5.5 hours using 984 Westmere cores on Pleiades. Converting the disparate grid sizes and CPUs utilized to a common basis, FUN3D required 131 CPU hours per million grid points for each half revolution, while OVERFLOW required 76 CPU hours per million grid points for each half revolution.

The first comparison of results is shown for a flight condition denoted as Counter 9017 (C9017) in the UH-60A Airloads Database. This condition is one of the most challenging level-flight conditions in the database, as it represents a high-thrust condition ($C_T/\sigma = 0.129$) at a moderate advance ratio ($\mu = 0.245$). The high-thrust level is very close to the stall boundary and leads to dynamic stall at several locations on the retreating side of the rotor disk. The high loading for this level flight condition was achieved by flying at a high altitude.

Results are presented for normal force, pitching moment, chord force, and torsion moments. Fig. 8 shows the variation with rotor azimuth angle of the flight data and computed results for a subset of these radial stations. The results are presented in a triptych format which makes it easy to see the simultaneous effects of an "event", such as dynamic stall, on all three quantities. In these plots, all harmonics are shown and means are removed from the data. The mean removal from the data for comparison is an accepted practice in order to compare oscillatory behavior of the load. This also removes known uncertainty in the measured data, especially, the pressure transducers near the trailing edge. The pitching moment is difficult to measure due to sensitivity of pressure transducer performance. The dynamic stall cycle in a rotor flow can be better examined from various plots when we look at the stall cycle in unconventional azimuthal scale of the rotor disk. This follows from the convention used by Bousman in his study of dynamic stall,[20] where the starting azimuth is taken as 135° instead of 0°. This azimuth of 135° is midway through the second quadrant of the disk. This shift allows one to look at the stall cycle in its entirety as one does not need to look back and forth at the fourth quadrant and the first quadrant as in the case of plots starting from 0°.

In general the variations of rotor airloads with azimuthal angle are reasonably well predicted by both codes. At station $r/R = 0.865$, near azimuth angles of 255° and 315°, successive dynamic stall events are observed, wherein the normal force drops sharply, chord force increases, and the pitching moment becomes much more negative. On the advancing side at this radial station, a much more mild drop in normal and chord force, together with a slightly more nose-down pitching moment is observed near $\psi = 55^\circ$. These are indicative of a light stall, one that does not extend to the trailing edge.[20] It should be noted that the chord force is taken as positive in the local "y" direction of the reference-blade coordinate system, which progresses from the trailing edge to the leading edge, as indicated in Fig. 2. Both simulations capture the double stall event on the retreating side well, and both exhibit the light lift-stall on the advancing side. The stall events are similar in the two simulations, with some minor differences in the stall location and the magnitude of the forces and moments.

In Fig. 9, the computed and measured loads at all nine radial stations are averaged over 360° of rotor rotation to define the mean sectional loads. The mean of the measured data for normal force and chord force at $r/R = 0.400$ clearly does not follow the trend of the mean of the measured data at nearby stations. Similarly, the mean pitching moment in the measured data at $r/R = 0.865$ seems unexpectedly different from neighboring radial stations. Except for a small difference in the mean pitching moment at $r/R = 0.865$, the mean sectional loads from the two simulations are very nearly identical. However, the agreement between computation and flight for the mean normal force, and especially for mean pitching moment, is not satisfactory.

Some of the offsets in airload between flight and computation may be due to inconsistencies in the measurements, as discussed above. Therefore, to better assess how well the computational results capture the variation with azimuthal position, Figures 10-12 qualitatively compare the normal force, pitching moment, and chord force, each with the respective mean values removed, across the entire rotor disk. The two CFD/CSD simulations agree well with each other, and generally agree well with the flight data. Both simulations miss the peak negative loading near the tip in the second quadrant of the rotor disc. Both codes predict too large a chord force in the second quadrant, two thirds of the way out the span.

Some of the torsional moments are compared in Fig. 13 for this case. The torsional loads from the two simulations are very similar to each other and generally follow the trend of the flight data. The OVERFLOW results tend to produce somewhat larger peak torsional moments, in slightly better agreement with the data. The flight data show successive peaks in the torsional moment in the first quadrant, particularly evident at $r/R = 0.0466$ and $r/R = 0.500$. Both computations only show a single peak. Likewise, successive troughs are seen in the flight data near $\psi = 270°$, whereas both computations exhibit only a single local minimum in that region.

The second comparison of results from the two codes is for the low-speed condition of flight counter C8513. This level-flight conditions is not as severe as the high thrust case of C9017. It represents a Blade Vortex Interaction (BVI) condition with trim target of $C_T/\sigma = 0.087$ at a moderate advance ratio ($\mu = 0.15$). The flight data for this condition were taken at a relatively large side-slip angle $\beta = 7.71°$. This side-slip angle was used in the OVERFLOW simulations, but not in the FUN3D simulations. To approximately compensate for this difference between the two simulations, the FUN3D results are shifted in azimuth (to the left) by $7.71°$. Since neither simulation included fuselage, this shift should yield the same end result as if the correct side-slip had been used. The OVERFLOW and FUN3D results are close to each other, indicating that the rotor is trimmed to appropriate aerodynamic loads, despite the difference in side slip angles. Results are again presented for normal force, pitching moment, and chord force. Fig. 14 shows the variation with rotor azimuth angle of the flight data and computed results. The BVI events at 90 degrees, and 270 degrees are reasonably captured, especially, large variation of the blade loading and phase on the advancing side and retreating side. The normal force and pitching moment comparison is noticeably better near the root region. The pitching moments are underpredicted near the tip. The chord forces show different trend than the normal force and pitching moments which both simulations predict too high a value near the root and too low a value near the tip.

Fig. 15 shows the computed and measured azimuthal average loads of all nine radial stations. Both FUN3D and OVERFLOW results are almost identical. However, the agreement between computation and flight for the mean chord force, and the mean pitching moment is not satisfactory.

Fig. 16 shows the torsional moment variations for this low-speed case. The FUN3D results are again shifted to the left in azimuth by $7.71°$ to account for $\beta = 0°$ in the FUN3D simulation. The torsional loads computed by the two codes are virtually indistinguishable. Comparison with flight data is not entirely satisfactory.

Finally we present the convergence of the loose coupling steps in Fig. 17 for C9017 and in Fig. 18 for C8513. The convergence is represented here with control angles measured at the pitch link, where θ_0 is the collective angle, θ_{1c}, and θ_{1s} are the lateral and longitudinal cyclic pitch respectively. The dynamic stall case of C9017 needed more than 25 cycles to reach a converged state. This is true for both codes although FUN3D was run only for 25 cycles. With FUN3D, the convergence of all three control angles exhibited a lightly damped oscillation, and by 25 trim cycles, the peak-to-peak oscillations control angles were less than $0.25°$. The convergence of OVERFLOW was seemingly more rapid, although between trim cycles 20 and 30, oscillations on the order of $0.25°$ appeared and were then damped out. The large number of coupling steps required for this high thrust condition makes this case computationally expensive. It would be interesting to see how this compares when one tries to simulate a maneuvering flight. In contrast, the simulation of C8513 trimmed in less than 10 cycles. For both flight counters, the OVERFLOW/CAMRAD-II and FUN3D/CAMRAD-II coupling yield trimmed control angles that differ by less than $0.50°$.

Conclusions

The unstructured Navier-Stokes CFD code FUN3D and structured Navier-Stokes code OVERFLOW used the same comprehensive code CAMRAD II in a loosely coupled manner that satisfactorily compared with the flight test data. The FUN3D uses second-order spatial and time accuracy, while OVERFLOW uses mixed-order spatial accuracy with second-order time accuracy. The blade motion and grid deformation for both cases presented were properly verified to make sure that both of the codes solved the same problem. The challenging flight test case with multiple dynamic stall events is reasonably represented, although some obvious discrepancies of forces and moments were seen near the stall location, especially, for chord forces. However, the stall location seems to be well captured. The low-speed case of blade vortex interaction compares well for both codes. For both codes, the loose coupling convergence to a trim state is expensive for the dynamic stall case. The number of coupling steps needed for the BVI case is reasonably small.

Acknowledgments

The authors would like to thank Ms. Elizabeth Lee-Rausch of the Computational AeroSciences Branch at NASA Langley Research Center for her many contributions to this work. The rotor-blade definitions were graciously provided by Mr. T. Alan Egolf of the Sikorsky Aircraft Corporation. The authors would like to thank Dr. Pieter Buning for helpful discussions regarding OVERFLOW. Finally, we would like thank Dr. Hyeonsoo Yeo of the U.S. Army Aeroflightdynamics Directorate for providing the CAMRAD II input files.

References

[1]Potsdam, M., Yeo, H., and Johnson, W., "Rotor Airloads Prediction Using Aerodynamic/Structural Coupling," *American Helicopter Society 60th Annual Forum Proceedings*, 2004.

[2]Pahlke, K. and van der Wall, B., "Chimera simulations of multibladed rotors in high-speed forward flight with weak fluid-structure-coupling," *Aerospace Science and Technology*, Vol. 9, No. 5, 2005, pp. 379–389.

[3]Lim, J. W. and Strawn, R. C., "Prediction of HART II Rotor BVI Loading and Wake System Using CFD/CSD Loose Coupling," AIAA Paper 2007-1281, Jan. 2007.

[4]Sankaran, V., Sitaraman, J., Wissink, A., Datta, A., Jayaraman, B., Potsdam, M., Mavriplis, D., Yang, Z., O'Brien, D., Saberi, H., Cheng, R., Hariharan, N., and Strawn, R., "Application of the Helios Computational Platform to Rotorcraft Flowfields," AIAA Paper 2010-1230, Jan. 2010.

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

[6]Anderson, W. K. and Bonhaus, D. L., "An Implicit Upwind Algorithm for Computing Turbulent Flows on Unstructured Grids," *Computers and Fluids*, Vol. 23, No. 1, 1994, pp. 1–22.

[7]Johnson, W., "Rotorcraft Aerodynamics Models for a Comprehensive Analysis," *American Helicopter Society 54th Annual Forum Proceedings*, 1998.

[8]Bondi, M. J. and Bjorkman, W. S., "Trends User's Guide and Reference Manual," NASA TM 108806, June 1994.

[9]Bousman, W. G. and Kufeld, R. M., "UH-60A Airloads Catalog," NASA TM 2005-212827, Aug. 2005.

[10]Harten, A., Lax, P. D., and van Leer, B., "On Upstream Differencing and Godunov-type Schemes for Hyperbolic Conservation Laws," *SIAM Review*, Vol. 25, No. 1, 1983, pp. 35–61.

[11]Toro, E. F., Spruce, M., and Speares, W., "Restoration of the Contact Surface in the HLL Riemann Solver," *Shock Waves*, Vol. 2, 1994, pp. 25–34.

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

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

[14]Vatsa, V. and Carpenter, M. H., "Higher-Order Temporal Schemes with Error Controllers for Unsteady Navier-Stokes Equations," AIAA Paper 2005-5245, June 2005.

[15]Vatsa, V. N., Carpenter, M. H., and Lockard, D. P., "Re-evaluation of an Optimized Second Order Backward Difference (BDF2OPT) Scheme for Unsteady Flow Applications," AIAA Paper 2010-0122, Jan. 2010.

[16]Noack, R. W., "DiRTlib: A Library to Add an Overset Capability to Your Flow Solver," AIAA Paper 2005-5116, June 2005.

[17]Noack, R. W., Bogar, D. A., Kunz, R. F., and Carrica, P. M., "SUGGAR++: An Improved General Overset Grid Assembly Capability," AIAA Paper 2009-3992, June 2009.

[18]Prizadeh, S. Z., "Advanced Unstructured Grid Generation for Complex Aerodynamic Applications," *AIAA Journal*, Vol. 48, No. 5, May 2010, pp. 904–915.

[19]Chan, W. M., "Enhancements to the Grid Generation Script Library and Post-Processing Utilities in Chimera Grid Tools," Proceedings 9th Symposium on Overset Composite Grid and Solution Technology, Oct. 2008.

[20]Bousman, W. G., "A Qualitative Examination of Dynamic Stall from Flight Test Data," *American Helicopter Society 53th Annual Forum Proceedings*, 1997.

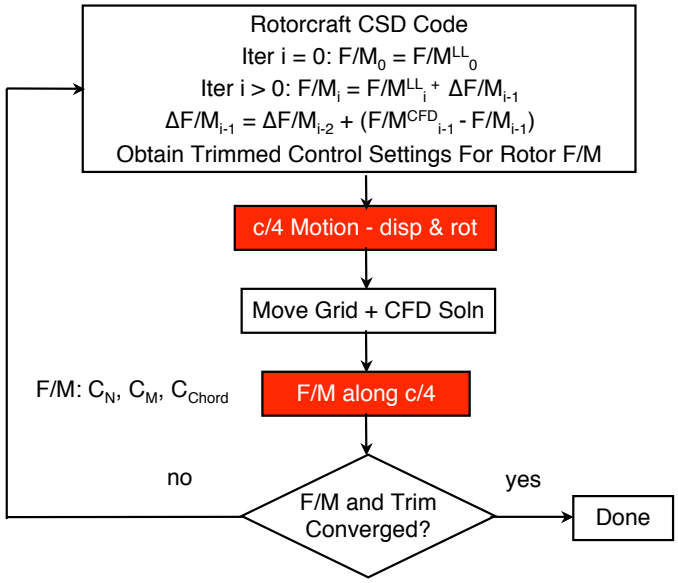

Figure 1: Flow diagram for loose coupling between CFD and CSD codes; adapted from Reference 1.

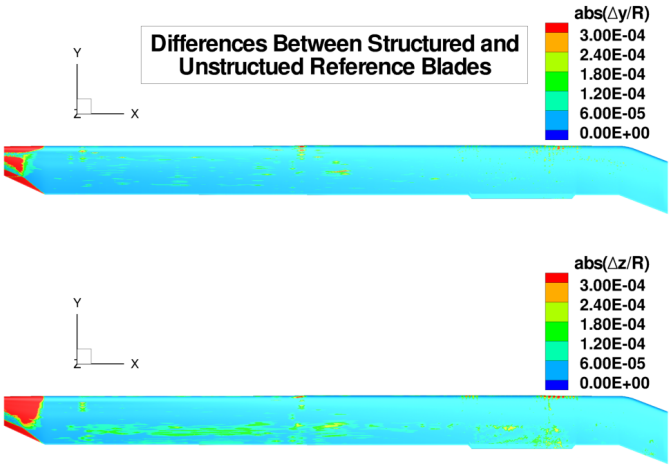

Figure 2: Differences in independently-generated reference-blade grids used in each solver.

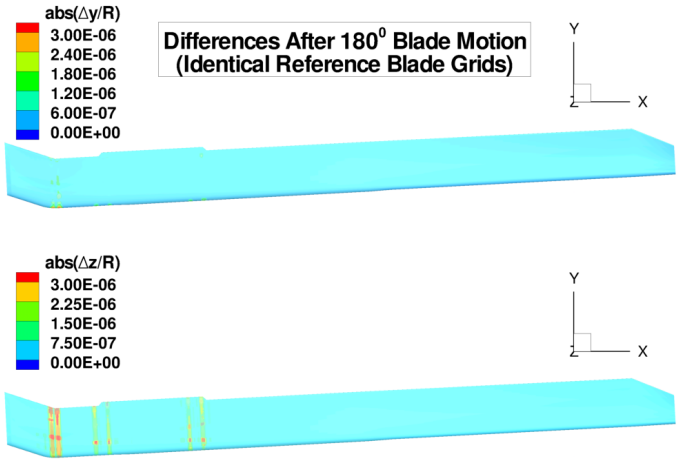

Figure 3: Differences in blade deformation after motion through 180° azimuth, using the identical structured/hexahedral reference-blade grid in each solver; identical motion files.

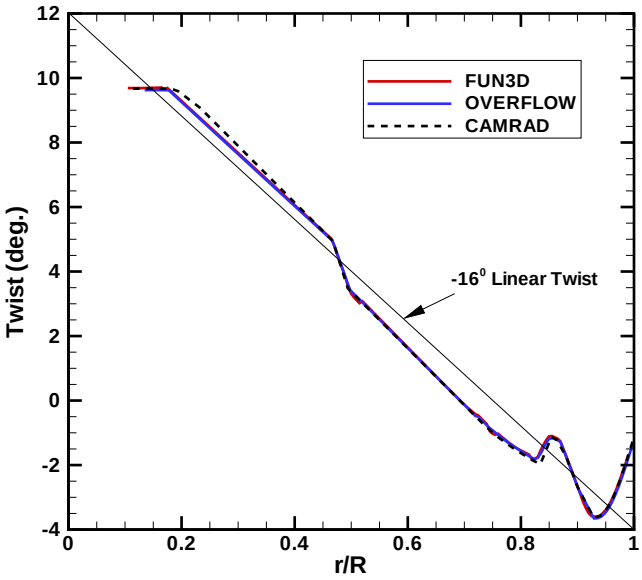

Figure 4: Computed twist distributions for independently-generated reference blades, compared with twist distribution used in CAMRAD II.

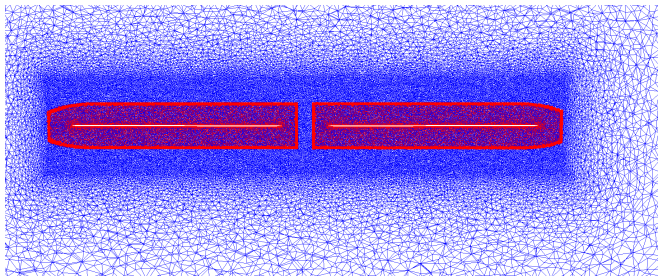

Figure 5: Slice through FUN3D volume grid system.

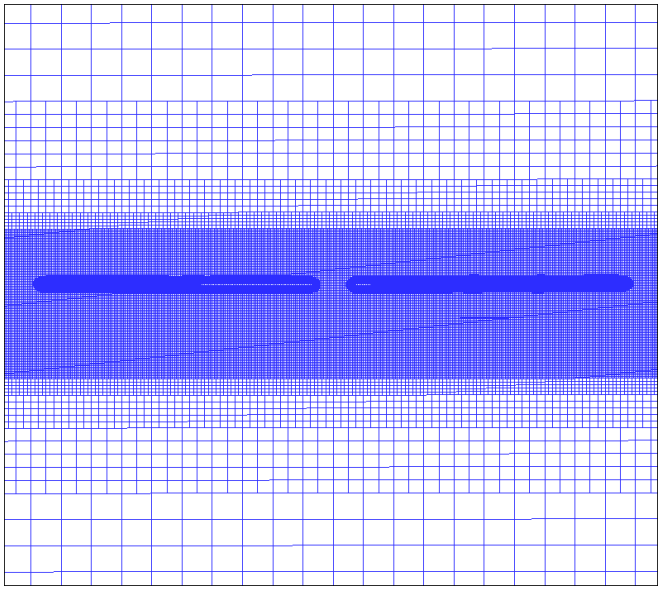

Figure 6: Slice through OVERFLOW volume grid system.

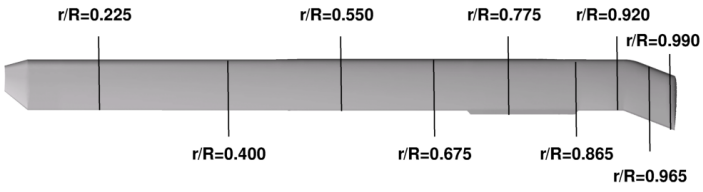

Figure 7: Blade radial stations for which flight airloads are available.

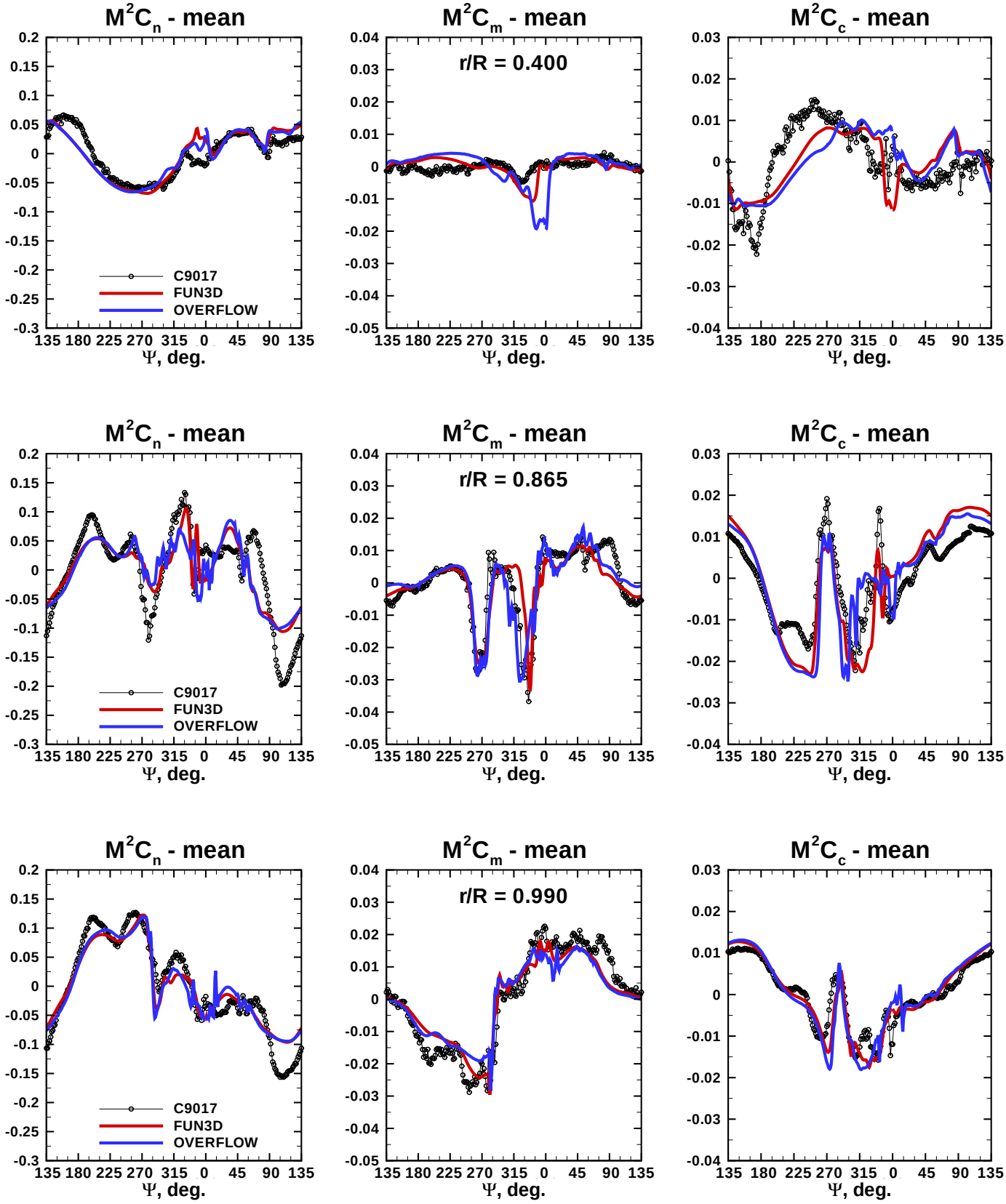

Figure 8: Comparison of computed and measured sectional airloads at r/R = 0.400, r/R = 0.865 and r/R=0.990, Counter 9017.

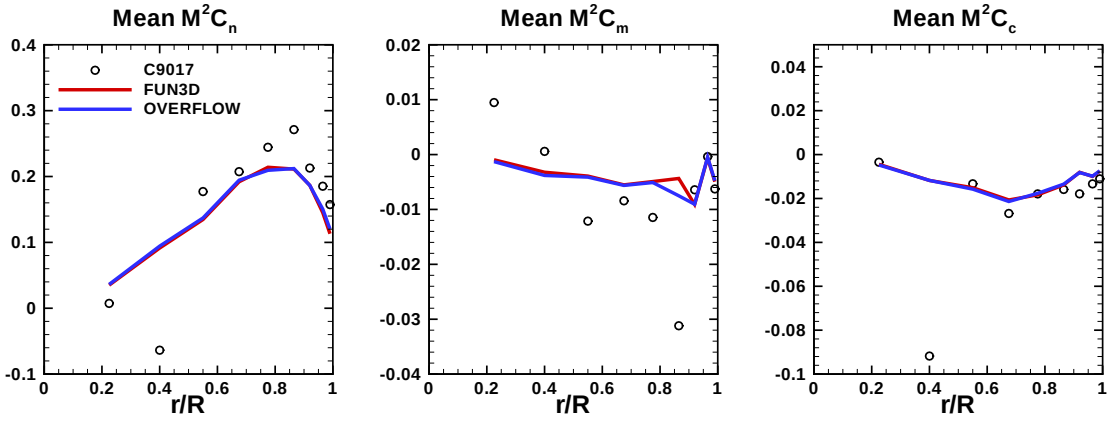

Figure 9: Comparison of computed and measured mean (azimuthally averaged) sectional airloads, Counter 9017.

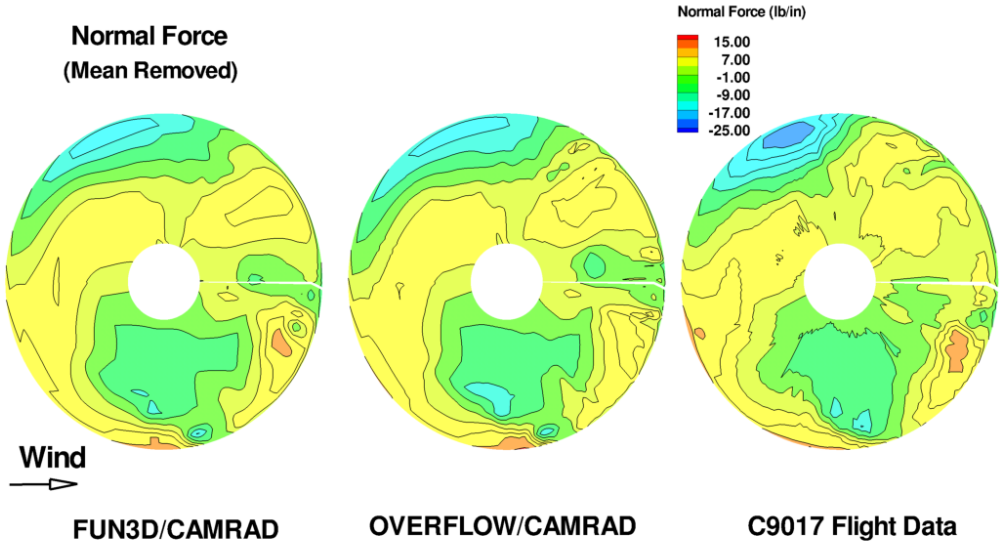

Figure 10: Comparison of computed and measured section normal force (mean removed), Counter 9017.

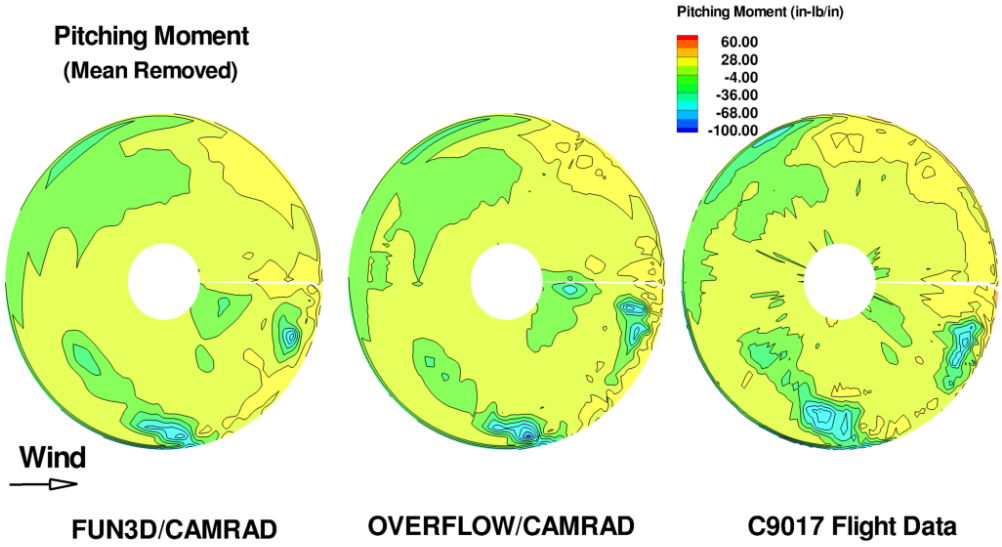

Figure 11: Comparison of computed and measured section pitching (mean removed), Counter 9017.

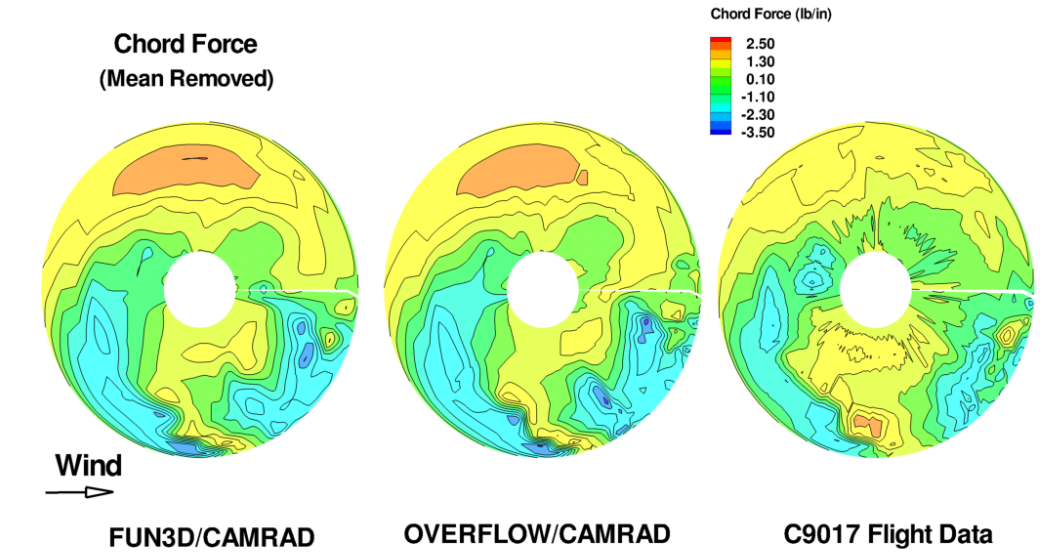

Figure 12: Comparison of computed and measured section chord force (mean removed), Counter 9017.

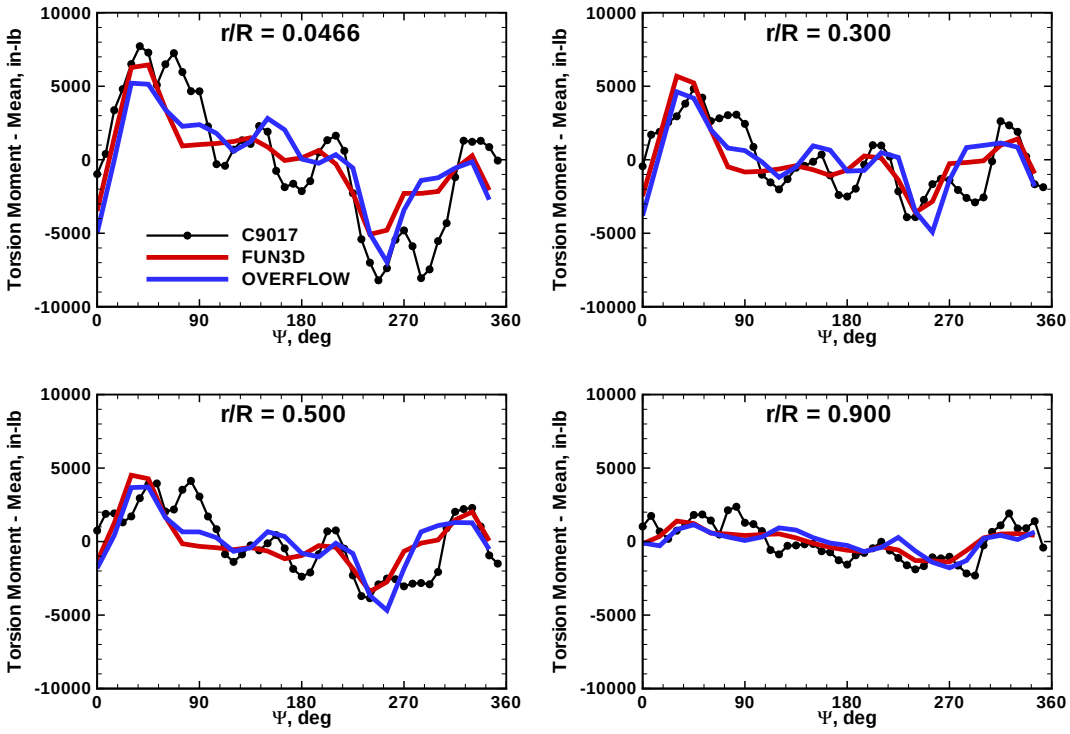

Figure 13: Comparison of computed and measured torsional moments (mean removed), Counter 9017.

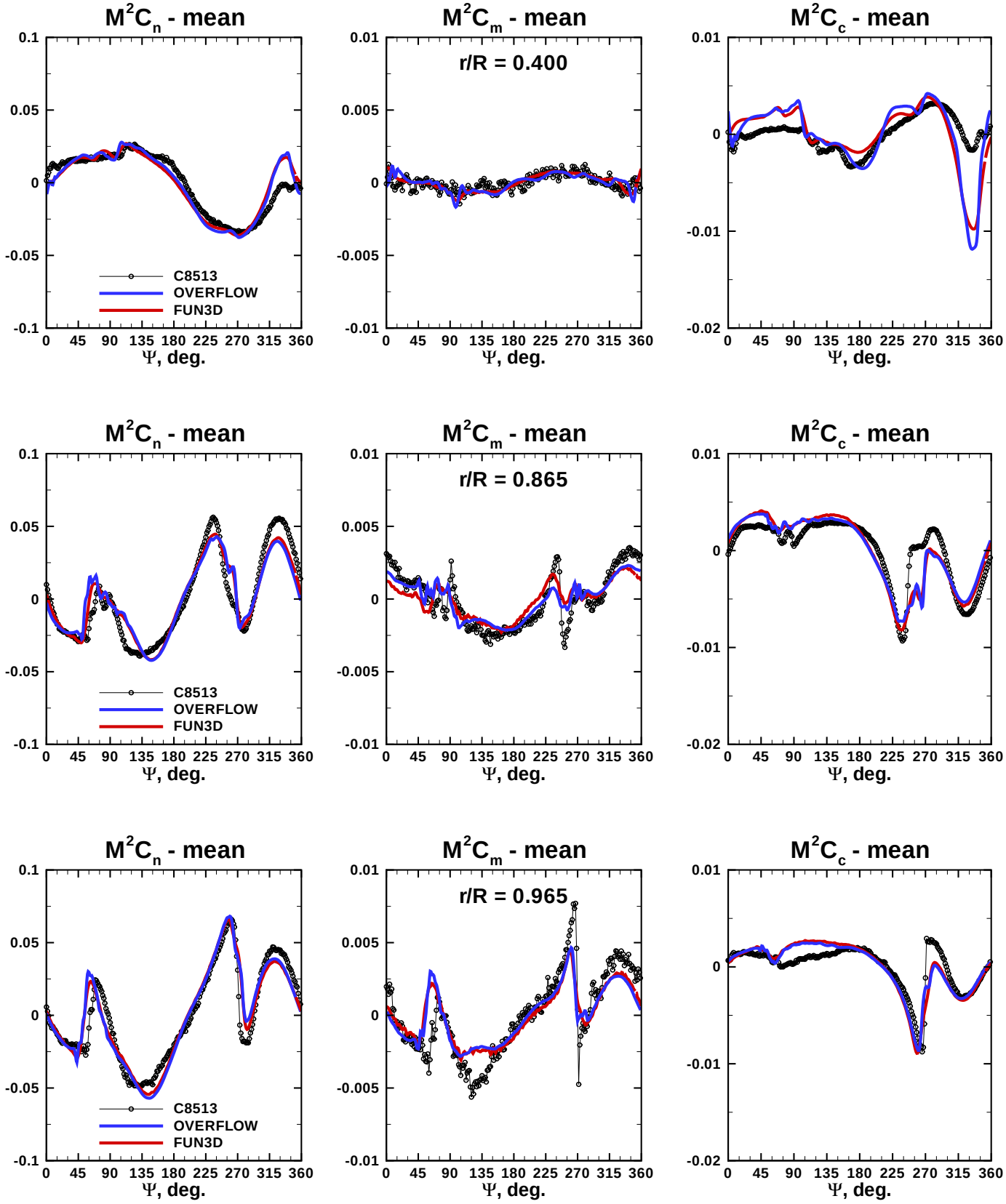

Figure 14: Comparison of computed and measured sectional airloads at r/R = 0.400, r/R = 0.865 and r/R=0.965, Counter 8513.

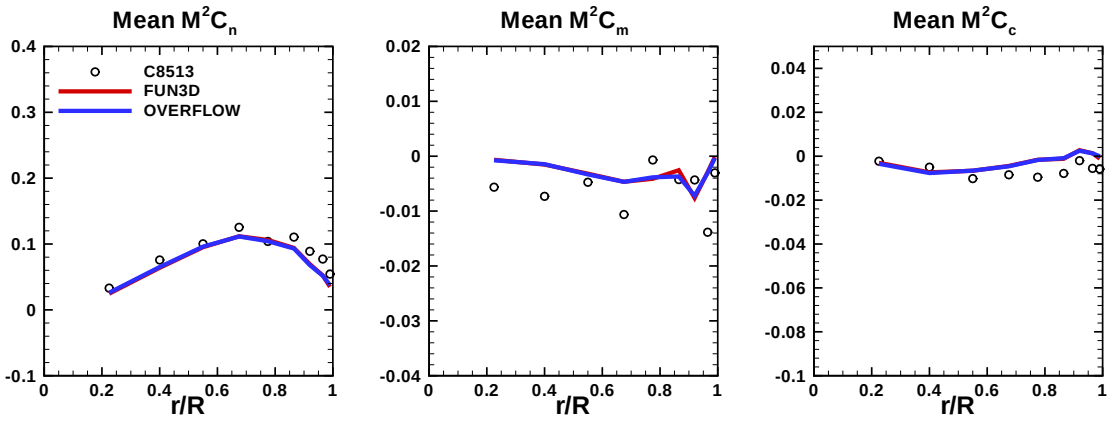

Figure 15: Comparison of computed and measured mean (azimuthally averaged) sectional airloads, Counter 8513.

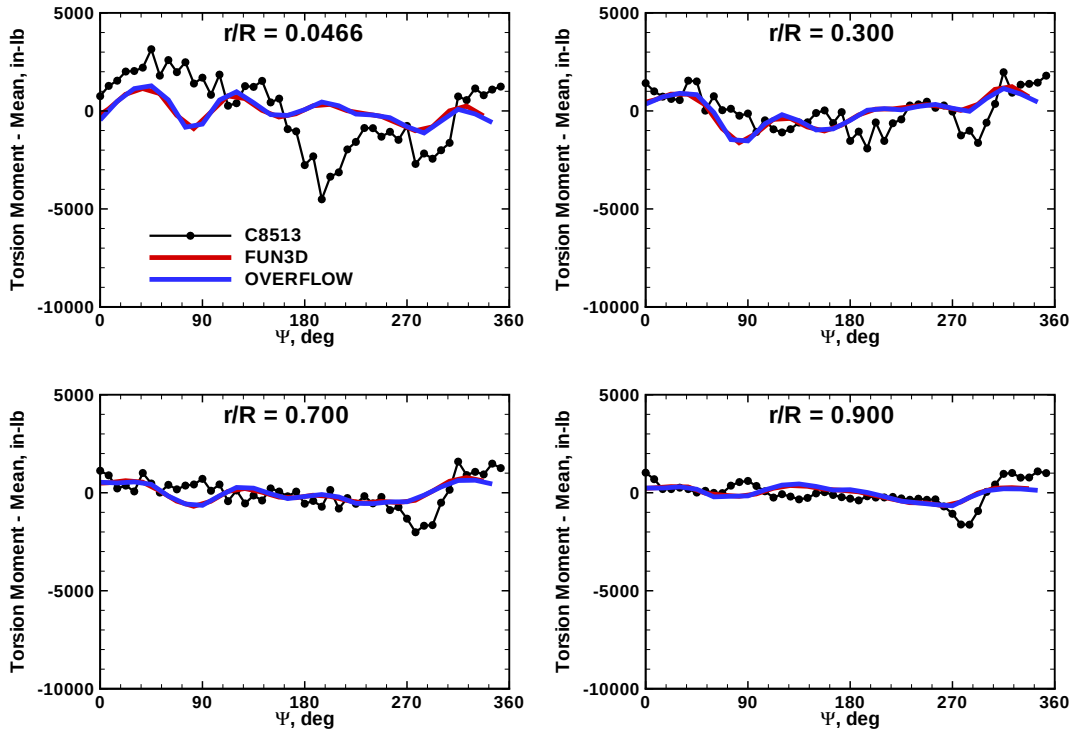

Figure 16: Comparison of computed and measured torsional moments (mean removed), Counter 8513.

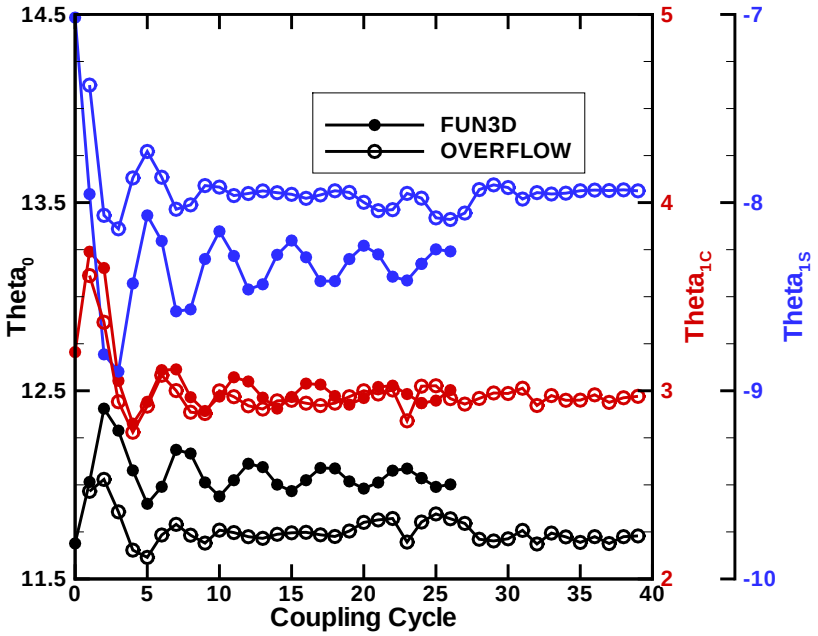

Figure 17: Comparison of coupling convergence, Counter 9017.

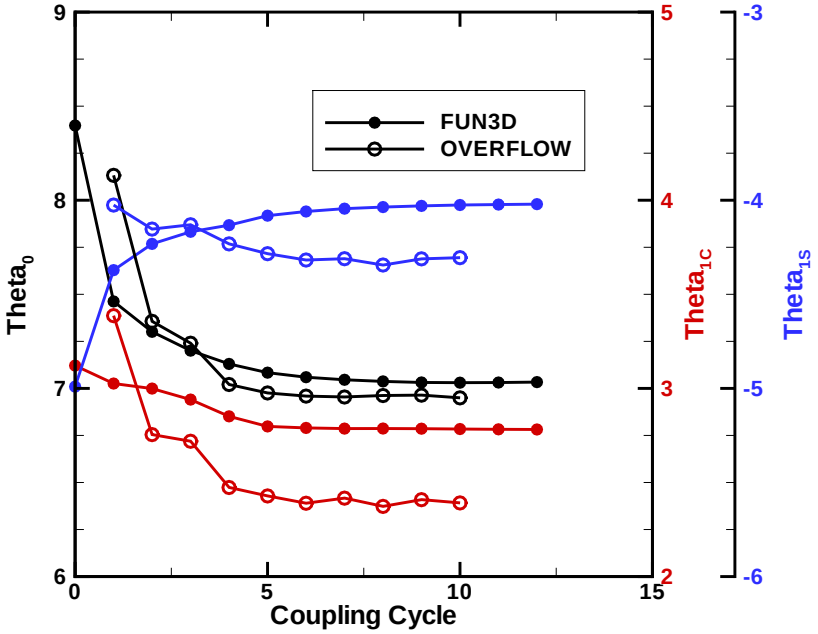

Figure 18: Comparison of coupling convergence, Counter 8513.