

SLIDING MESH ALGORITHM FOR CFD ANALYSIS OF ROTOR-FUSELAGE AERODYNAMICS

René Steijl and George N. Barakos

Department of Engineering
University of Liverpool
Brownlow Hill Road, Liverpool L3 4EX, United Kingdom
e-mail: g.barakos@liverpool.ac.uk

Key words: Rotorcraft CFD, sliding plane, rotor-fuselage

Abstract: The problem of rotor-fuselage interaction is central to the design and performance analysis of helicopters. However, regardless of its significance, rotor-fuselage aerodynamics has so far been addressed by very few authors. This is mainly due to the difficulties associated with both experimental and computational techniques when such complex configurations, rich in flow physics, are considered. In view of the above, the objective of this study is to develop and validate, computational tools suitable for rotor-fuselage engineering analysis using Computational Fluid Dynamics as the main tool with experiments playing a key role in the validation of the proposed methods.

To account for the relative motion between the fuselage and the rotor blades the concept of sliding meshes is introduced. A sliding surface forms a boundary between a CFD mesh around the fuselage and a rotor-fixed CFD mesh which has to be rotated to account for the motion of the blades. The sliding surface allows communication between the sliding meshes. CFD meshes adjacent to sliding surface do not necessarily have matching nodes or even the same number of cell-faces. This poses a problem of interpolation between the meshes which should not introduce numerical artefacts in the solution and should have minimal effects in the overall solution quality. As a secondary objective the employed sliding mesh algorithms should have small CPU overhead. The sliding mesh methods developed for this work are demonstrated for both simple and complex flows and emphasis is placed in the presentation of the inner workings of the developed algorithms

1 INTRODUCTION

The study of rotor and rotor-fuselage aerodynamics with CFD methods based on multi-block structured grids requires complex multi-block topologies so that the exact shape of the rotor blades and fuselage is represented. An additional difficulty comes from the requirement to account for the relative motion between rotor blades and the relative motion between the rotors and the fuselage. In this paper the concept of sliding grids with non-matching cell faces is used in an effort to allow multi-block structured grid solver to cope with rotor-body flow cases. The concept of non-matching cell faces is not new in CFD and has so-far been used by many researchers. Reference 1, amongst others, presents the fundamentals of numerical schemes for general boundary conditions for non-matching cell faces. Such methods are nowadays common in turbomachinery where non-matching and also rotating cell faces are used for the simulation of the flow between adjacent blade-rows of aeroengines. An example of such efforts can be found in reference 2 where multi-blade row configurations are analysed using sliding surfaces with unstructured meshes for full blade-row analysis of gas turbine engines. In the context of helicopter-rotor the CHIMERA technique which allows intersecting and non-matching grids is common (see references 3-4 amongst others). For the present work only non-matching cell faces are considered in an effort to develop a computational technique that allows rotor-body as well as complex rotor cases to be studied with minimal overhead and by re-using existing multi-block grids. The paper begins by presenting the details of the employed solver and, subsequently, the details of the employed sliding mesh technique. Examples demonstrating this method are also given for test cases ranging from simple aerofoils to full rotor-fuselage configurations.

2 NUMERICAL METHOD

2.1 The HMB flow solver.

The Helicopter Multi-Block (HMB) code⁵⁻⁶ was employed for this work. HMB solves the unsteady Reynolds-averaged Navier-Stokes equations on block-structured grids using a cell-centered finite-volume method for the spatial discretisation. An implicit time-integration method is employed, and the resulting linear systems of equations are solved using a preconditioned Generalized Conjugate Gradient method. For unsteady simulations, an implicit dual-time stepping method is used, which is based on Jameson's pseudo-time integration approach. The method has been validated for a wide range of aerospace applications and has demonstrated good accuracy and efficiency for very demanding flows. Examples of work with HMB can be found in references 5-8 and include dynamic stall⁸, blade-vortex interaction⁷ and rotors in hover and forward flight⁶. Several rotor trimming methods are available in HMB along with a blade-actuation algorithm that allows good near-blade grid quality to be maintained on deforming meshes⁶. The solver has a library of turbulence closures which includes several 1 and 2 equation turbulence models and even non-Boussinesq versions of the $k-\omega$ model. Turbulence simulation is also possible using either Large-Eddy or Detached-Eddy simulation. Straight from the beginning the solver was designed with parallel execution in mind. The MPI library along with a load-balancing algorithm are used to this end. Good parallel performance has been demonstrated on Beowulf clusters with up to 150 CPUs and on massively parallel machines like the HPCx facility available to UK Universities. For multi-block grid generation the ICEM-CFD Hexa commercial meshing tool is used and CFD grids with multi-million points and thousands of blocks are commonly used with the HMB solver.

2.2 Sliding and non-matching grids.

Two layers of halo-cells are used by HMB for imposing boundary conditions or to allow communication between adjacent blocks. Figure 1 presents a schematic of this arrangement. Each block is independent of its neighbouring blocks and for as long as the halo cells are populated with values of the primitive variables the solver can compute an update of the flow solution. Figure 1b presents a situation where two adjacent blocks have a non-matching cell faces. If the halo cells of each block could, however, be populated with “correct” values the solver will have no difficulty in computing the flow. There are three main steps involved in populating the halo cells. 1) Identification of the neighbouring cells for each halo-cell 2) interpolation of the solution at the centroids of the halo cells and 3) exchange of information between blocks associated with different processors. The last step is important for computations on distributed-memory machines only.

Identification of neighbouring cells

As shown in Figure 1, for each of the two halo cells of each cell having a face on a sliding plane a search must be carried out over all cells near the sliding plane in order to identify cells within a minimum distance of the cell under consideration. This set of cells is referred to as the “neighbours” and several interpolation schemes can be used in order to “weight” the contribution of each neighbour to the value assigned to cell at hand. If the non-matching grid interface is not sliding, the identification process must be performed only once and even an $N \times N$ search, where each cell is checked against every other is not expensive. For sliding meshes and problems with moving grids the identification process must be repeated for each time-step of the computation which may be a significant overhead. In reference 2 a technique was used where a sliding surface was discretised using a regular grid and identifications was then taking place in a rapid fashion in small patches of this regular surface grid.

Interpolation of the solution

Either distance or overlapping cell area (Figure 1c) can be used as a weight and a summation must be performed over all neighbours. Each of the flow variables is then computed as:

$$\phi_{halo} = \sum_i^{neighbours} w_i \phi_i \quad (1)$$

The weights can easily be obtained if cell distances are used but computation of weights based on the overlapping area between cells can be more time-consuming. Sliding planes can be of arbitrary shape and for this reason the contributing cell surfaces must all be projected on the curvilinear ξ, η, ζ axes used in HMB before the weights can be computed. This step can be combined with a transformation from primitive to conservative variables so that flux-weighted summations can also be computed.

If, in combination with the identification method, a regular surface grid is used to discretise the sliding plane the interpolation method may also benefit. In this way, consistent interpolation on both sides of the sliding plane can be achieved since the sliding plane is discretised in a regular fashion using three-dimensional cells. Using this technique, minimum entropy changes across the sliding plane have been computed even on relatively coarse grids.

Exchange of information between processors

For distributed memory computations and for matching multi-block grids, halo-cell information must be exchanged between processors computing on adjacent blocks. One of the issues with sliding and non-matching cells, however, is that the information to be communicated may change as the computation is carried out. The worst case is when all processors computing blocks with faces on the sliding plane must receive information about the whole sliding surface, including the co-ordinates of the grid points and the flow variables at two layers of cells. This option has been taken for this work since an algorithm to minimize the exchanged information could be complicated and perhaps restricted to certain grid motions.

3 RESULTS AND DISCUSSION

The technique described in the previous paragraph has been implemented in HMB and tested for a variety of cases and results are presented in this section.

3.1 Flows around bumps and aerofoils

The first test case considered was the inviscid flow around a bump. Figure 2 presents results on a grid which had matching cells on both sides of the sliding plane. No grid motion has been applied and consequently this case can only be used to compare the accuracy of the solution interpolation. The same computation has been performed on the same grid but with the inter-block boundary treated in exactly the same way as a normal block boundary of a multi-block grid. As the figure 2 shows, there is no difference between the obtained results and this suggests that the existing interpolation method did not introduce any numerical artefacts. Due to the small size of the employed grid (2000 cells) the computational overhead for the case where the sliding and non-matching grid algorithm is used is very low. Profiling of the solver revealed an increase of 0.4% of CPU time. A small number of iterations, about 120, were needed for convergence and a CFL of 25 was used for the implicit time-marching.

A second test case attempted was for the flow around an aerofoil with a sliding plane placed around the profile at a distance of 1 chord. For this test case an O-grid has been used around a NACA0012 section at zero incidence. The grid was divided in four blocks around the section and a sliding plane was placed 1 chord away from the aerofoil giving a total of 8 blocks and 8,200 cells. Further cases were considered with the sliding grid located at 5 and 10 chords away and the obtained results revealed that the further out the sliding plane the lesser the influence of the employed interpolation method. The isobars shown in Figure 3 are continuous across the sliding plane even 1 chord away from the profile. In comparison to the previous test case the sliding plane is now curved and thus the interpolation of flow variables at halo cells is harder than for the first test case. In addition, significant flow gradients are now present. Less than 300 iterations were necessary for convergence to be obtained and a CFL number of 15 was used.

A second version of the same test case was also considered where the outer ring of blocks around the aerofoil has a smaller number of cells on the sliding plane than the inner one. Figure 4 presents this configuration and one can see an almost 2 to 1 correspondence of cells across the sliding plane. This case is more demanding but again the proposed method managed to cope with a converged solution obtained within 250 implicit steps with a CFL of 15. Comparing the near-field solution in the inner ring of blocks between the last two cases, little deterioration of the solution can be seen as a result of the outer coarse grid employed. This is encouraging and opens new possibilities for reducing the number of cells at the far-field of the

computational domain, something not possible with normal multi-block grids. For the aerofoil cases the second test case was 2% more demanding in terms of CPU time in comparison to the first one due to the necessary identification process for the neighbours of each cell. Still, however, the overhead in terms of CPU time due to the presence of the sliding plane was less than 1%. Profiling of the solver revealed that the computations are dominated by the influence of the implicit linear solver with other operations taking a small fraction of the overall CPU time.

3.2 Rotor-fuselage test cases

For the demonstration of the sliding plane method computations have been undertaken first for the flow around a clean ROBIN fuselage. This test case was needed in order to develop multi-block grid topologies suitable for generic helicopter body shapes [9,10]. Several computations have been undertaken using the Navier-Stokes equations and the $k-\omega$ turbulence model of Wilcox. Indicative results from this effort can be seen in Figure 5 where the surface pressure distribution at a station mid-way along the ROBIN fuselage is compared against experimental data by Freeman and Mineck [9]. A negative incidence of 10 degrees was used for this case and the yaw angle was zero. The flow near the front of the fuselage was affected by transition but its effect was reduced further downstream. Inspection of the flow around the dog-house on top of the fuselage revealed minor separation. The ROBIN body was also used in combination with an actuator disk model in HMB. This model used a uniform source of momentum with a strength computed based on the thrust coefficient specified by the user. On the average the effect of the actuator disk should be similar to the effect of a rotor but, for the employed model, no spatial variation of loading and non unsteadiness is considered. The effect of the actuator disk can be seen in Figure 6 where the clean and actuated cases are compared for the surface pressure coefficient at two stations along the ROBIN fuselage. Flow visualisation with streaks released at the level of the actuator disk reveal significant flow distortion as can be seen in Figure 7.

Building on the experience gained with the clean and actuated ROBIN cases the rotor-fuselage case with a sliding plane in between has been considered. The overall configuration can be seen in Figure 8. The HIMARCS rotor design [11] has been used due to the availability of CFD grids and its relatively simple planform. The employed multi-block topology around the fuselage can be seen in Figure 9 where the edges of the blocks are shown. These edges are projected on the exact geometry of the body resulting in the configuration shown in Figure 10. For testing the sliding plane method no trimming has been applied with emphasis put on the performance of the interpolation and identification parts of the method.

The employed grid had a total of 1.5 million points with just 0.4 million cells around the fuselage and 1.1 million cells on the four-bladed rotor. Unsteady computations have been carried out for three revolutions of the main rotor using azimuthal steps of 0.25 degrees. A first attempt to visualise the obtained flow can be seen in Figure 11 where streamlines are computed at the two sides of the fuselage. Significant asymmetries have been obtained due to the presence of the rotor and the lack of a trimming method. More details are however revealed by placing streamlines near the sliding plane. This is shown in Figure 12 where snapshots are shown for six values of the azimuth angle between 0 and 360 degrees. The results show clearly the footprint of the rotor on the sliding plane with substantial distortions near the location of the blades. The streamlines tend to bend towards the fuselage and the tips of the rotor disk. In comparison to the actuator disk case discussed in the previous paragraph spatial and temporal flow variations are now present.

This test case was demanding in terms of the sliding plane algorithm since a search for the halo-cells had to be conducted at each time step and, in addition a substantial amount of information had to be communicated between all processors shearing blocks on the sliding plane. About 4% of the overall CPU time was dedicated to the search for cell neighbours with an additional 2% lost for the communication giving a total overhead of 6%. Since this test case represents a worst case scenario for the sliding plane method the obtained results are satisfactory as far as the performance of the method is concerned.

4 SUMMARY AND FUTURE WORK

A CFD technique was presented which allows CFD computations on grids with non-matching cell faces. The main application of the method is the analysis of rotor-fuselage configurations though other configurations are possible. Results have been presented for several flow cases including simple 2-D flows and even complete rotor-fuselage configurations. The obtained results suggest that the proposed method is adequate in terms of both accuracy and efficiency. The overhead of the method in terms of CPU time was found to be almost proportional to the size of the employed grid while little dependence was found between the obtained results and the interpolation algorithm used for reconstructing the solution around the sliding planes. Issues regarding the parallel efficiency of the method still remain since a substantial amount of information is to be communicated between processors shearing blocks with faces on the sliding plane. Further, the search algorithm presently employed takes no advantage of the special rotational motion of the rotors and it is expected that further savings in CPU time can be obtained. At present, further validation of the method and analysis of its performance are being considered for cases where rotor-fuselage experimental data are available.

5 REFERENCES

- [1] M. Rai, A Conservative Treatment of Zonal Boundaries for Euler Equation Calculations, *Journal of Computational Physics*, Vol. 123, Issue 2, 1986, pp. 472-503.
- [2] G. Barakos, M. Vahdati, A.I. Sayma, C. Breard and M. Imregun, A Fully Distributed Unstructured Navier-Stokes Solver for Large-Scale Aeroelasticity Computations, *The Aeronautical Journal*, Vol. 105, Issue 1050, 2001, pp. 419-426.
- [3] Pahlke, K. and van der Wall, B., Calculation of Multibladed Rotors in High-Speed Forward Flight with Weak Fluid-Structure-Coupling, 27th European Rotorcraft Forum, Moscow, Russia, September 2001.
- [4] T. Renaud, C. Benoît, J. -C. Boniface, P. Gardarein, Navier-Stokes computations of a complete helicopter configuration accounting for main and tail rotors effects, 29th European Rotorcraft Forum, Friedrichshafen, Germany, September 2003.
- [5] G. Barakos, R. Steijl, K. Badcock and A. Brocklehurst, Development of CFD Capability for Full Helicopter Engineering Analysis. 31st European Rotorcraft Forum, 13-15 September 2005, Florence, Italy.
- [6] R. Steijl, G. Barakos and K. Badcock, A Framework for CFD Analysis of Rotors in Hover and forward Flight, *Int. J. for Num. Meth. in Fluids*, Vol. 51, 2006, pp. 819-847.
- [7] R. Morvant K. Badcock, G. Barakos and B. E. Richards, Aerofoil-Vortex Interaction Using the Compressible Vorticity Confinement Method, *AIAA J.*, Vol.43, Issue 1, 2004, pp. 63-75.
- [8] A. Spentzos, G. Barakos, K. Badcock, B. Richards, P. Wernert, S. Schreck and M. Raf-

fel, CFD Investigation of 2D and 3D Dynamic Stall, *AIAA J.*, Vol. 34, Issue 5, 2005, 1023-1033.

- [9] C.E. Freeman, R.E. Mineck, Fuselage Surface Pressure Measurements of a Helicopter Wind-Tunnel model with a 3.15-Meter Diameter Single Rotor, NASA TM-80051, 1979.
- [10] M.S. Chaffin and J. D. Berry, Navier-Stokes and Potential Theory Solutions for a Helicopter Fuselage and Comparison with Experiment, NASA TM-4566, June 1994.
- [11] Noonan, K.W., Yeager, W.T., Singleton, J.D., Wilbur, M.L., and Mirick, P.H., "Wind Tunnel Evaluation of a Helicopter Main-Rotor Blade with Slotted Airfoils at the Tips", NASA-TP-2001-211260, December 2001.

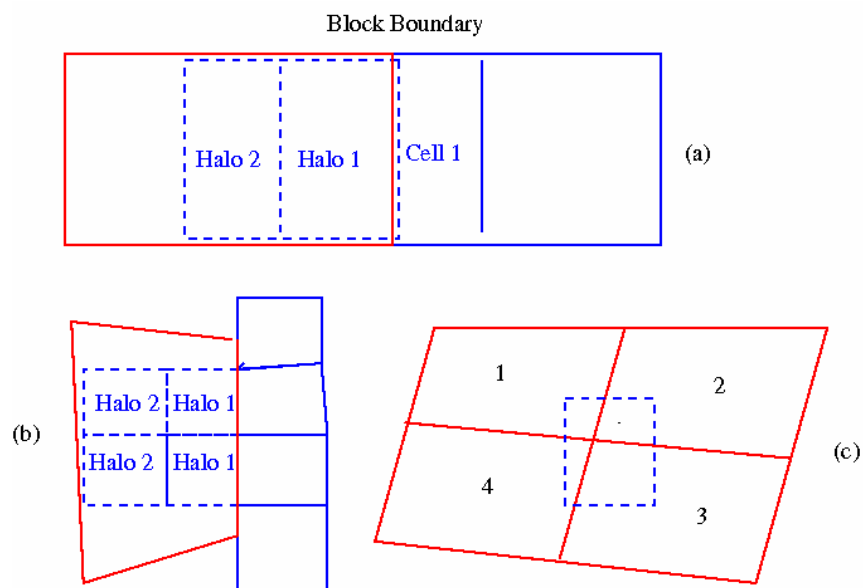


Figure 1: Schematic of halo cells and non-matching cell faces.

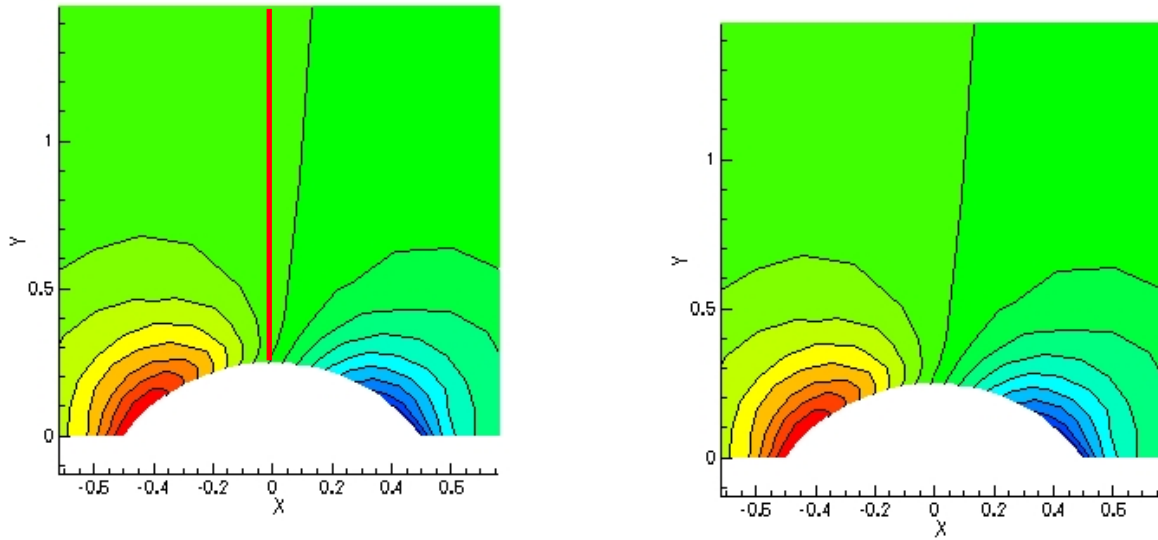


Figure 2: Comparison of the pressure field around a circular bump for the case where interpolation is used across a matching sliding plane (left) and the full multi-block grid (right). For each case 15 isobars are shown between the minimum and maximum pressure values. The inviscid solution was obtained on a coarse grid of 2000 cells and the Mach number was 0.4.

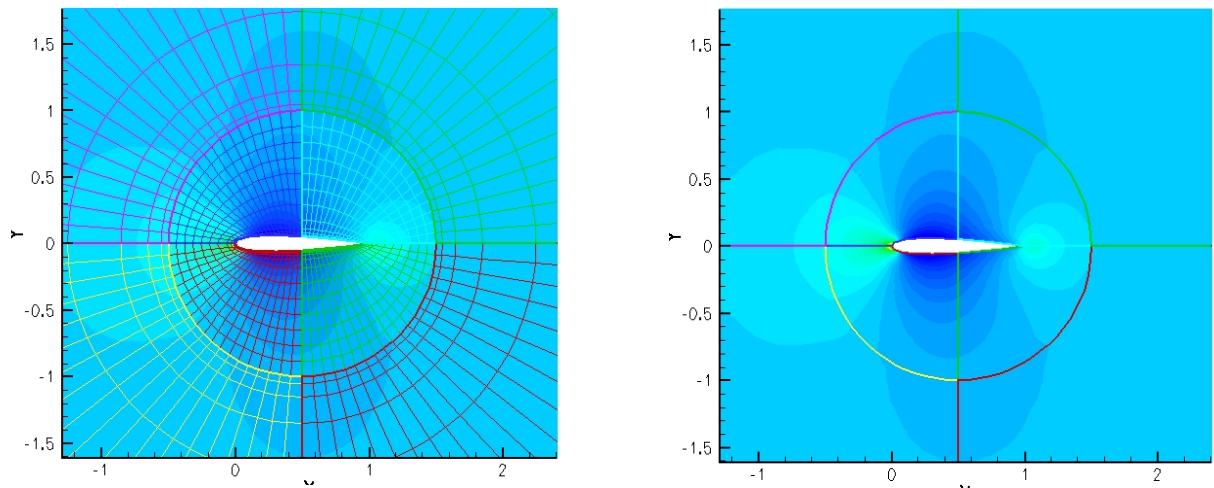


Figure 3: Grid arrangement (left) and isobars (right) for the flow around an inviscid NACA0012 section at zero degrees of incidence. Matching grids are used across the sliding plane.

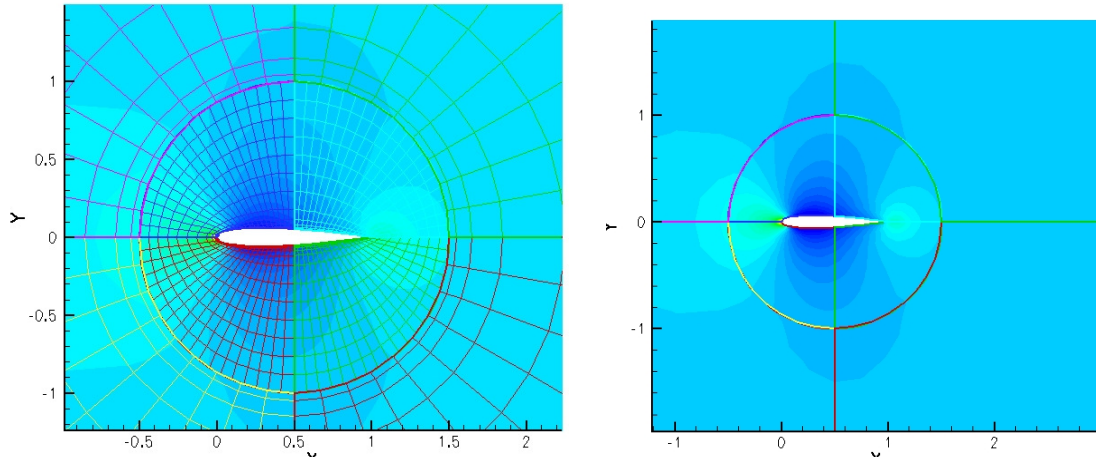


Figure 4: Grid arrangement (left) and isobars (right) for the flow around an inviscid NACA0012 section at zero degrees of incidence. Non-matching grids were used across the sliding plane.

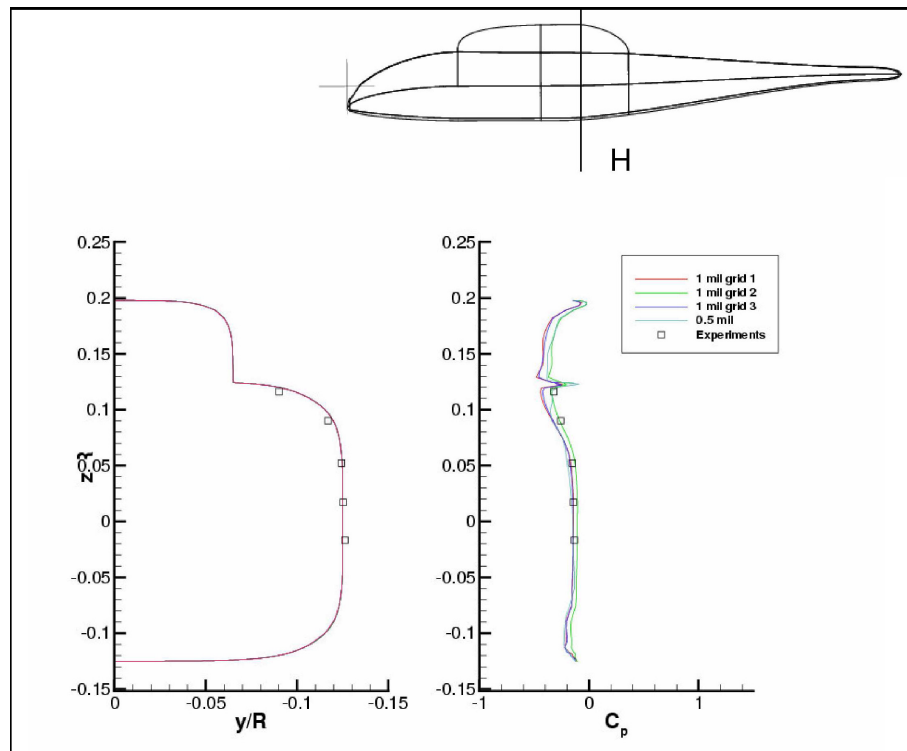


Figure 5: Comparisons between experiments [9] and CFD computations for the surface pressure coefficient distribution along station H for the ROBIN fuselage case. CFD results on 4 grids are shown each with a different multi-block topology. The locations of the pressure taps on the fuselage are also presented. The computations correspond to -10 degrees of incidence angle.

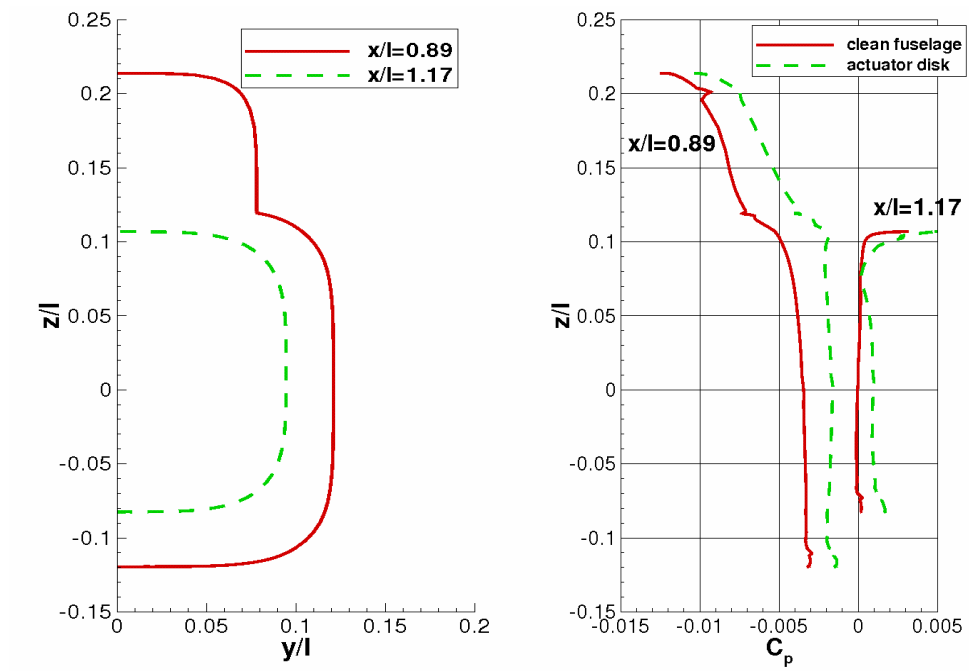


Figure 6: Comparisons between clean and actuated cases for the surface pressure coefficient distribution at two stations on the ROBIN fuselage. For the actuated case an advance ratio of 0.3 was used and C_T/σ of 0.064. The rotor disk was tilted by -3 degrees.

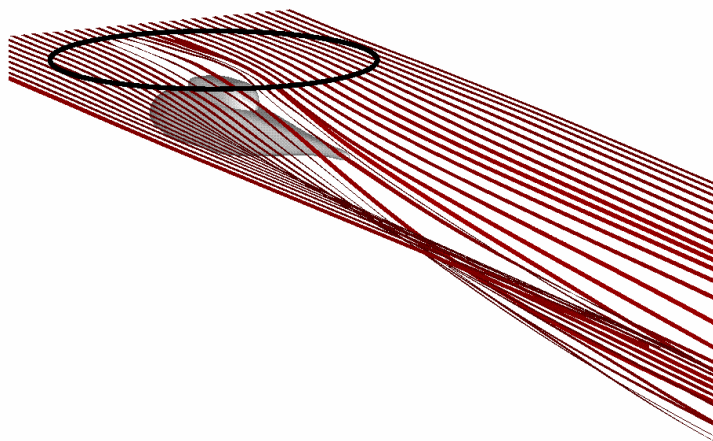


Figure 7: Flow visualization of the flow through the actuator disk plane for the ROBIN fuselage, $\mu=0.3$, 3deg nose down configuration, $C_T/\sigma=0.064$.

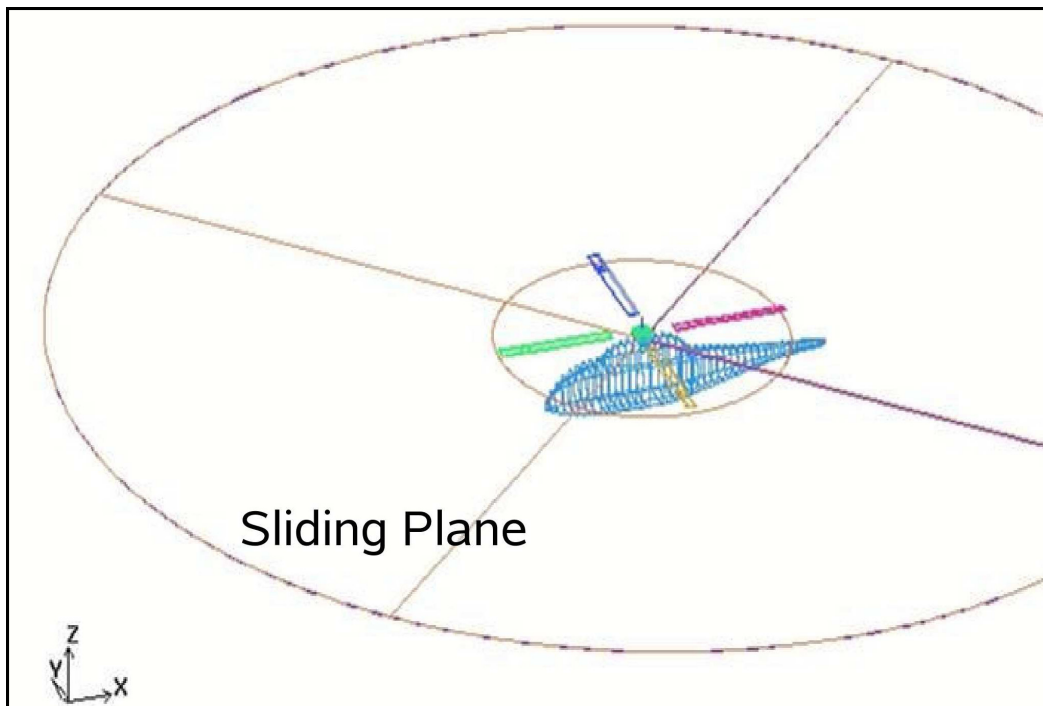


Figure 8: Sliding plane arrangement for the rotor-body configuration. The sliding plane is placed half-way between the doghouse of the ROBIN body and the tip path plane. The far-field of the domain was of cylindrical shape and was placed at 4 rotor radii from the rotation centre.

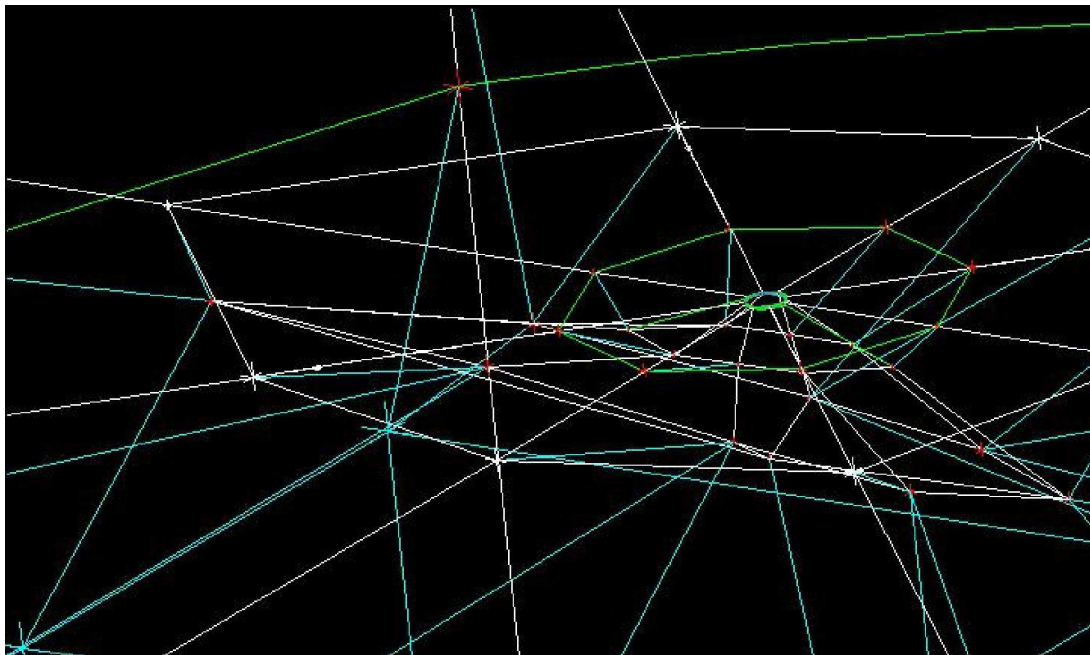


Figure 9: Multi-block topology used around the ROBIN fuselage for the rotor/body configuration of Figure 8. The employed topology allowed for a sliding surface to be placed above the fuselage and around the hub. The straight edges of the topology were mapped to the actual shape of the ROBIN body.

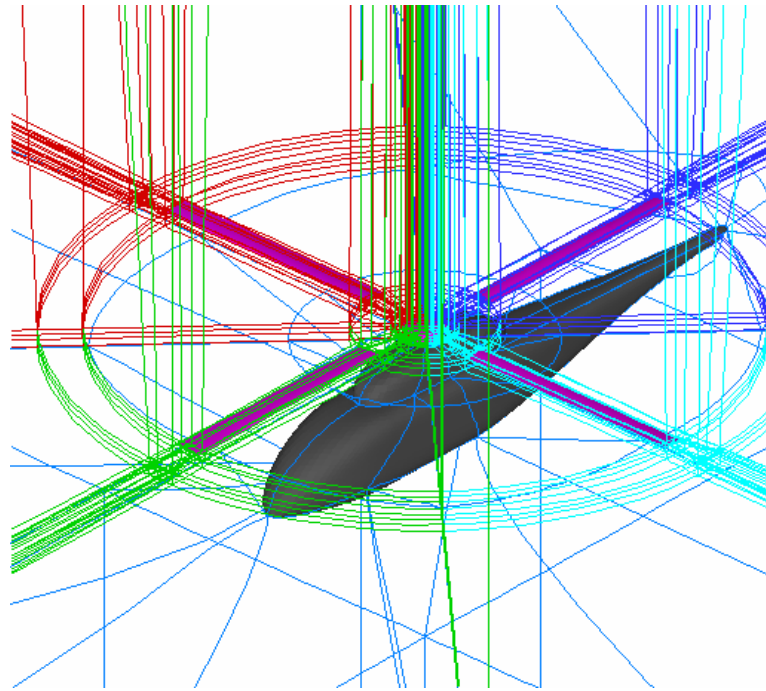


Figure 10: Multi-block grids around the sliding plane for the case of Figure 8. The block edges shown around the fuselage in Figure 9 are now mapped to the shape of the ROBIN body. The multi-block topology around the rotor is also shown.

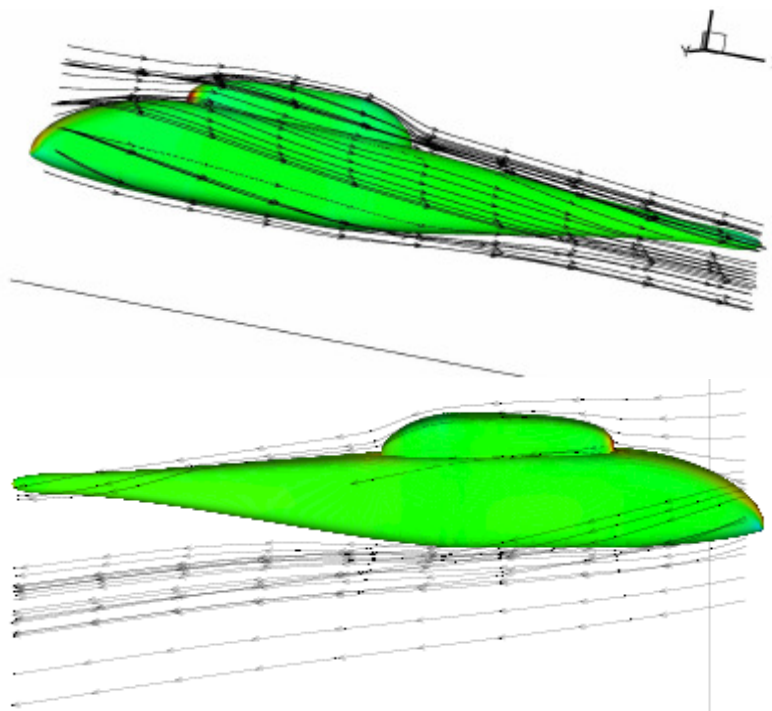


Figure 11: Flow visualization around the ROBIN body with the HIMARCS rotor above. The rotor was not trimmed and an advance ratio of 0.25 was applied with a fixed cyclic. A total of 1.5 million points were used with 0.4 million cells on the fuselage side and 1.1 million cells on the rotor.

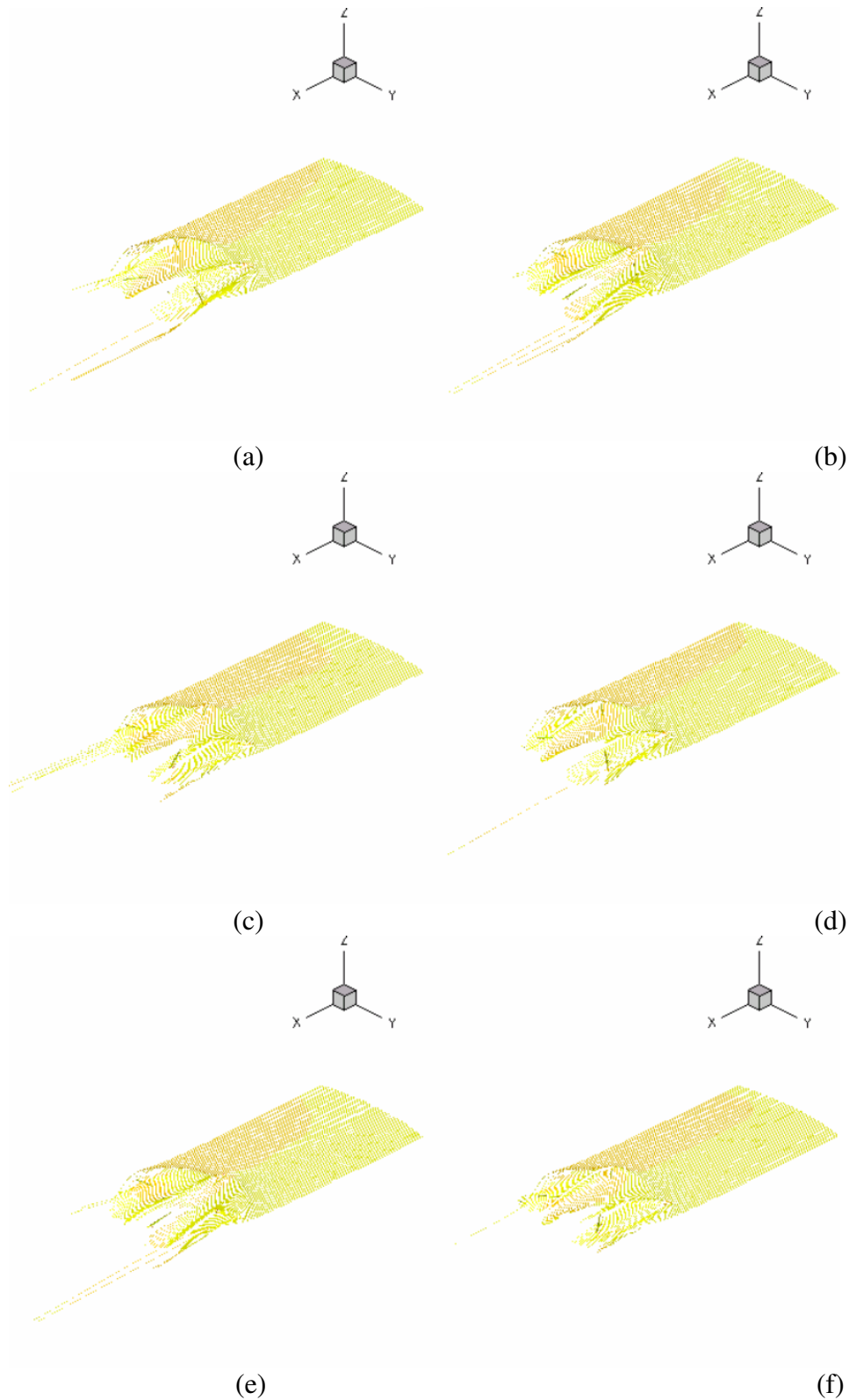


Figure 12: Flow visualization near the sliding plane area between the ROBIN body and the rotor at (a) 345, (b) 355, (c) 25, (d) 45, (e) 90 and (f) 200 degrees of azimuth. The footprint of the rotor blades can be seen as distortions on streaks placed at the sliding plane.