CFD Analysis of Rotor-Fuselage Aerodynamics based on a Sliding Mesh Algorithm

R. Steijl, G. Barakos and K. Badcock
CFD Laboratory, Department of Engineering
University of Liverpool, Liverpool L69 3GH, United Kingdom

Abstract

Rotor-fuselage interaction is central to the design and performance analysis of helicopters. However, regardless of its significance this problem is not well-studied and few CFD works have so far been published. In this paper, a method is put forward to allow CFD computations of rotor-fuselage problems using a sliding mesh to interface the rotor and fuselage regions. A sliding plane 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 rotor blades. CFD meshes adjacent to a sliding plane do not necessarily have matching nodes or even the same number of cell-faces. This poses a problem of interpolation between CFD meshes and, in addition, the employed algorithms should have small CPU overhead. The properties of this method are assessed and validation results are presented for several flow cases.

1. Introduction and Numerical Method

CFD computations for rotor and rotor-fuselage aerodynamics utilising multi-block structured grids require complex multi-block topologies so that the exact shapes of the rotor blades and the helicopter fuselage are represented. Additional difficulties come from the need to account for the motion of the rotor blades and the relative motion between the rotor and the fuselage. To alleviate this problem and allow multi-block structured grid solvers to cope with rotor-body flow cases, the concept of sliding grids with non-matching cell faces is used. The concept of employing non-matching cell faces is not new in CFD and has so-far been used by many researchers. Rai[1], amongst others, presented the fundamental ideas and details of numerical schemes for general boundary conditions for non-matching cell faces. Such methods are nowadays very popular in turbomachinery CFD 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[2] where sliding surfaces are used with unstructured meshes for full blade-row CFD analysis of gas turbine engines. While sliding planes are routinely used in turbomachinery CFD, work on helicopter rotor blades and rotor-fuselage configurations is mainly carried out using the CHIMERA technique which allows for intersecting and non-matching meshes. Such efforts are detailed in the works of Pahlke and van der Wall, and Renaud *et all.*[3-4], amongst others. CHIMERA methods provide a good framework for the study of rotor-fuselage aerodynamics, however, the complexity of the employed algorithms and the associated problem of hole-cutting in overlapping meshes, result in significant overhead in terms of CPU time.

For the present work, non-matching cell faces are consider 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. Consequently, the interface between two non-matching grids is not arbitrary, like for the CHIMERA method, but forms a plane shared between the rotor and fuselage domains.

The paper begins by presenting the details of the employed CFD 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.

1.2 Outline of the CFD solver

The Helicopter Multi-Block (HMB) code[5-6] 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 spatial discretisation. For steady-state flows, implicit time-integration is employed, and the resulting linear systems of equations are solved using a pre-conditioned 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[6-8] include dynamic stall[8], blade-vortex interaction[7] and rotors in hover and forward flight[6]. Several rotor trimming methods are available in this solver along with a blade actuation algorithm that allows for the near-blade grid quality to be preserved on deforming meshes[6]. The solver has a library of turbulence closures which include several one- and two-equation models and even non-Boussinesq versions of the k-ω model to allow for Reynolds stress tensor anisotropy. Turbulence simulation is also possible using either Large-Eddy or Detached-Eddy simulation. 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

1.3 Sliding mesh method

For compatibility with the second-order scheme employed in HMB, two layers of halo-cells are used to impose boundary conditions and allow communication between adjacent blocks (see Figure 1a for the details of this arrangement). Each block is independent of its neighbours 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 1(b) presents a situation where two adjacent blocks have non-matching cell faces. If the halo cells of each block are populated with values, the solver will have no difficulty in computing the flow. There are three main steps involved in populating the halo cells which include: a) Identification of the neighbouring cells for each halo-cell. b) Interpolation of the solution at the centroids of the halo cells. 3) Exchange of information between blocks associated with different processors. The last step is important for computations on distributed-memory machines only.

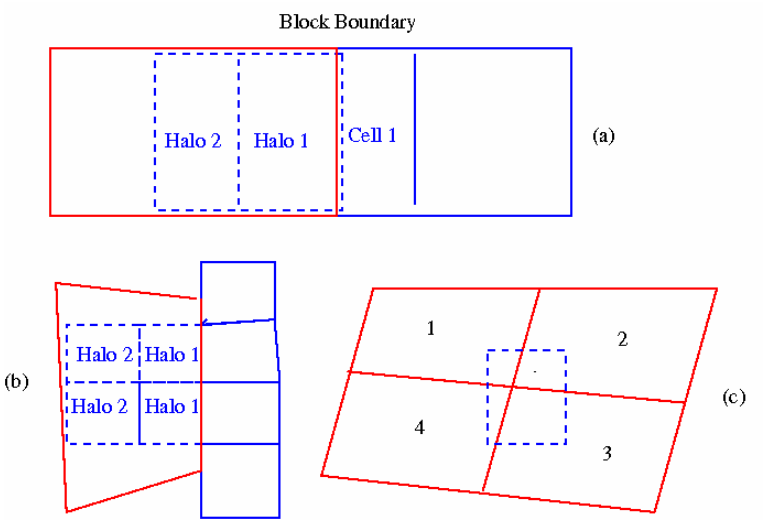


Figure 1: Schematic of halo cells and non-matching cell faces.

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 on 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 the cell at hand. The process of identifying the "neighbours" is difficult with several tests to be made to ensure that all cells contributing to a halo are accounted for. The most trivial case is presented in Figure 1(c) where only four "neighbours" are identified. If the non-matching grid interface is not sliding, the identification process must be performed only once and even an NxN 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 with a significant CPU overhead. In reference 2, a technique was used where a sliding surface was discretised using a regular grid and identification was then taking place in a rapid fashion in small patches of a regular mesh.
The interpolation of the solution across sliding planes can be performed as a weighted sum over the neighbouring cells identified during the process outlined in the previous paragraph. Either distance or overlapping cell area

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

$$\phi_{halo} = \sum_{i}^{neighbours} w_i \phi_i \qquad (1)$$

where w represents the weight and ϕ any primary or conservative variable. 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 obtained

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.

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 place 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.

2. Results and Discussion

A range of test cases have been computed as part of this work ranging from simple flows around aerofoils to rotor-body configurations. The test cases where of increasing complexity and designed to test specific aspects of the sliding mesh method. The first test case attempted was 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 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.

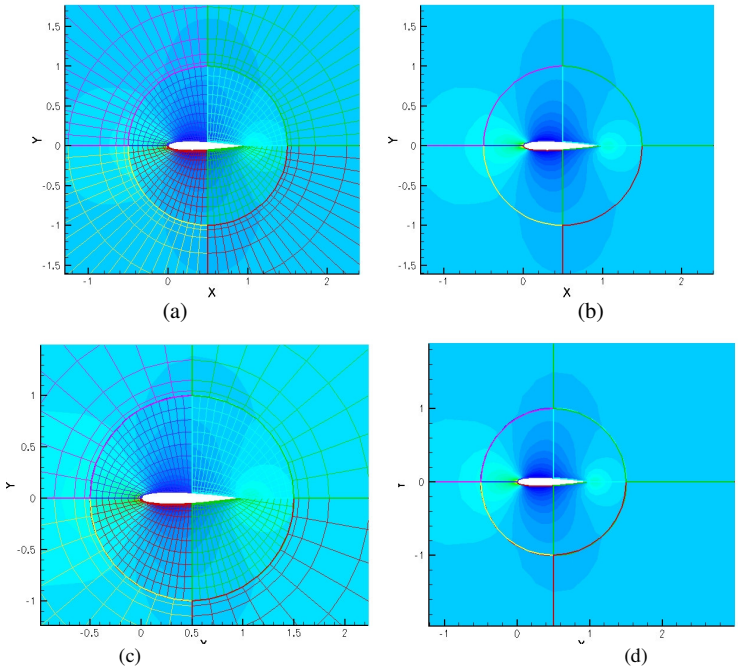


Figure 2: Grid arrangement (a and c) and isobars (b and d) for the inviscid flow around a NACA0012 aerofoil at zero degrees of incidence. Matching grids are used across the sliding plane for (a) and (b) while non-matching cells were employed for (c) and (d). The sliding plane which was placed one chord length away from the aerofoil.

The isobars shown in Figure 2(a,b) are continuous across the sliding plane even one chord away from the profile. In addition, significant flow gradients are present since the sliding plane is very close to the aerofoil. Less than 300 implicit iterations were necessary for convergence to be obtained and a CFL number of 15 was used. To further increase the case complexity, a second version of the same test case was also considered where the outer ring of blocks around the aerofoil had a smaller number of cells on the sliding plane than the inner one. Figure 2(c,d) presents this configuration which almost has a 2:1 correspondence of cells across the sliding plane. This case is more demanding though the proposed method produced a converged solution 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, indicating that the influence of the sliding place was minimal for this case. 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. This was due to the necessary identification process for the neighbours of each cell. Still, 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, iterative linear solver with other operations taking a small fraction of the overall CPU time.
Before attempting the rotor-body problem, computations were undertaken 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-ω turbulence model of Wilcox. Indicative results from this effort can be seen in Figure 3 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.

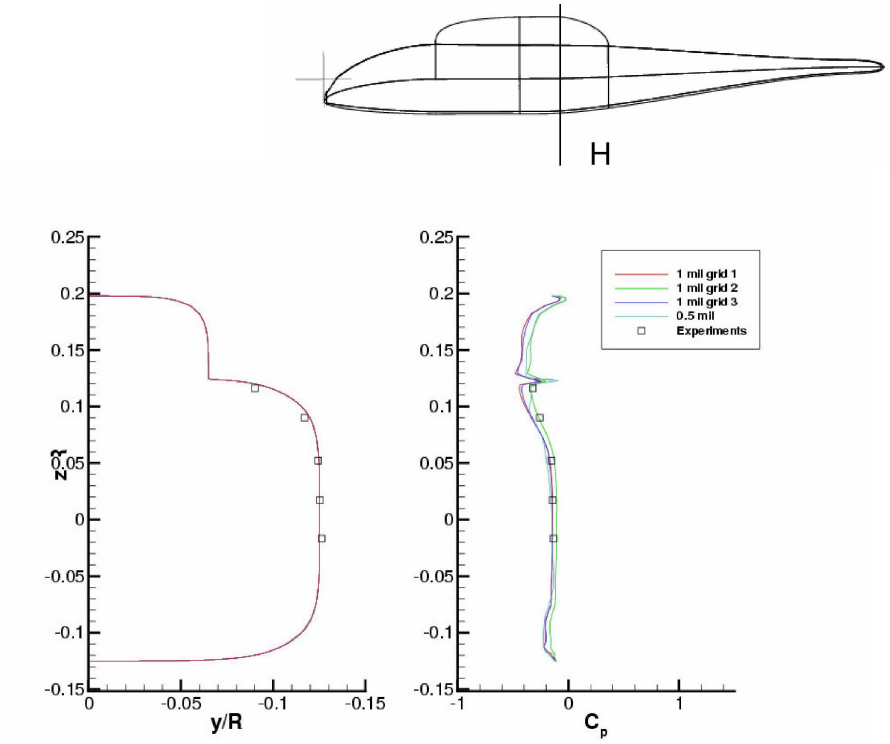
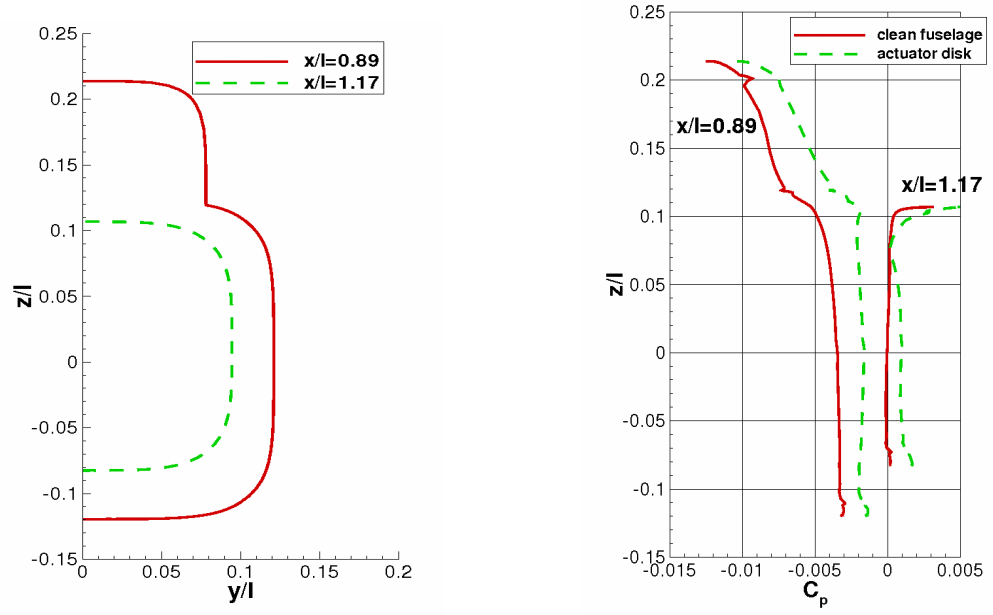


Figure 3: Comparisons between experiments[9] and CFD computations for the surface pressure coefficient distribution along a station on the ROBIN fuselage case. CFD results on several grids are shown each with a different multi-block topology. The locations of the pressure taps on the fuselage are also presented. The computations correspond to -10 degrees of incidence angle and zero sideslip.

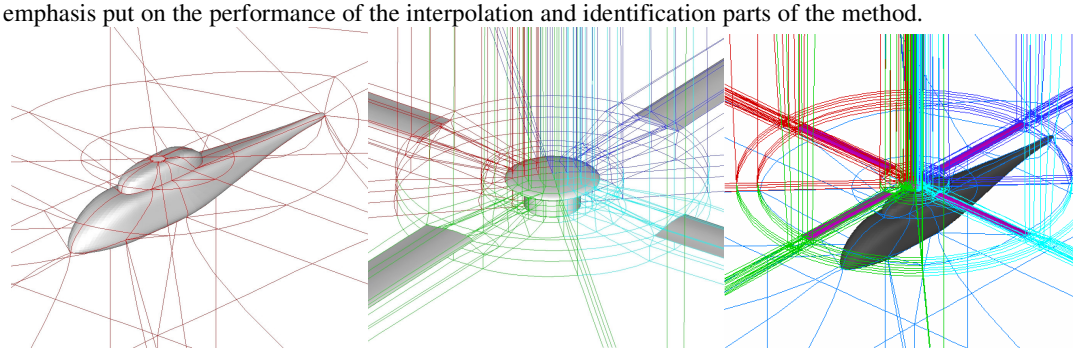
The ROBIN body was also used in combination with an actuator disk model available in HMB. This model used a uniform source of momentum with a strength computed based on the thrust coefficient specified by the user. For this case a thrust coefficient (C_T) to solidity (σ) ratio (C_T/σ) of 0.064 was used. On the average, the effect of the actuator disk should be similar to the effect of a rotor but, for the employed actuator disk model, no spatial variation of loading and non unsteadiness is considered. Figure 4 presents a comparison between experiments and CFD for the case where the actuator disk model is active. Results from the actuator disk model are compared later on with the solution for the rotor-body configuration.



(a) (b)

Figure 4: 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. (a) ROBIN shape at two stations and (b) surface pressure coefficient distributions at the stations shown in (a).

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 5. 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 5 where the edges of the blocks are shown. These edges are projected on the exact geometry of the body. 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.



(a) (b) (c)

Figure 5: Multi-block topologies used around the fuselage (a) and the rotor (b) for the rotor/body configuration (c). 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 fuselage and the rotor blades.

The employed grid had a total of 1.5 million points with just 0.4 million cells around the fuselage and about 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 6 where streamlines are computed at the two sides of the fuselage.

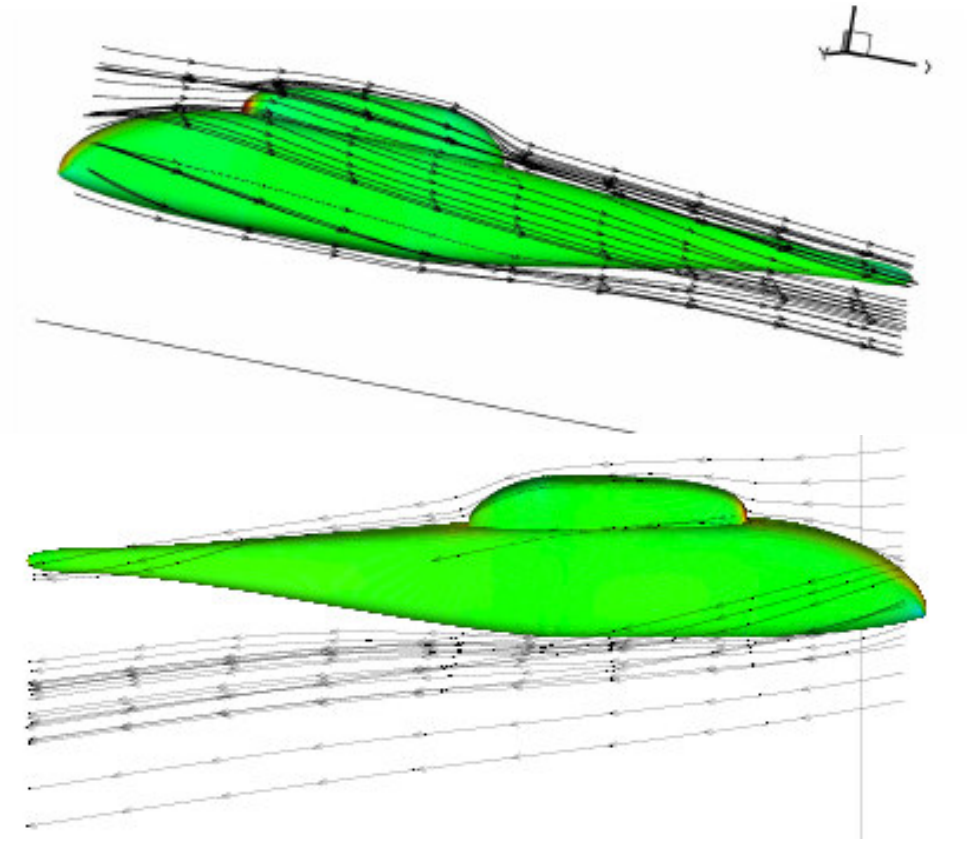


Figure 6: Flow visualization around the ROBIN body combined with the HIMARCS rotor. The rotor was not trimmed and an advance ratio of 0.25 was applied with a fixed cyclic. A total of 1.5 million points were used with 0.4 million cells on the fuselage side and 1.1 million cells on the rotor. The flow show substantial asymmetries between 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 7 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. Clearly, there is a lot more detail in the obtained solution in comparison to the actuator disk. Flow visualisation shown in Figure 7(e), with the actuator disk present, produced an almost symmetric pattern for streaks computed just below the disk plane. In addition, there was no temporal variation of the flow.

This test case was a challenge for the employed 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%. Regardless of this overhead, the method allowed us to extract a significant amount of flow detail and made the computation of the rotor-body configuration possible with our multi-block structured solver.

3. Conclusions and further 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. Three key steps in the sliding mesh algorithm have been identified and discussed which are: (i) the identification of neighbouring cells on the two sides of a sliding mesh, (ii) an interpolation technique for populating halo cells and (iii) the exchange of information regarding sliding planes for parallel computations.
Results have been presented for several flow cases starting from simple 2-D flows and progressing to 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. Further, 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 sharing 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 if this extra information is utilised.

References

[1] Rai, M., "A Conservative Treatment of Zonal Boundaries for Euler Equation Calculations", *Journal of Computational Physics*, Vol. 123, No. 2, 1986, pp. 472-503.
[2] Barakos, G., Vahdati, M., Sayma, A.I., Breard, C. and Imregun, M., "A Fully Distributed Unstructured Navier-Stokes Solver for Large-Scale Aeroelasticity Computations", *The Aeronautical Journal*, Vol. 105, No. 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] Renaud, T., Benoît, C., Boniface, J.-C., Gardarein, P., "Navier-Stokes Computations of a Complete Helicopter Configuration Accounting for Main and Tail Rotors Effects", *Proceedings of the 29th European Rotorcraft Forum*, Friedrichshafen, Germany, September 2003.
[5] Barakos, G., Steijl, R., Badcock, K. and Brocklehurst, A., "Development of CFD Capability for Full Helicopter Engineering Analysis", *Proceedings of the 31st European Rotorcraft Forum*, Florence, Italy, September 2005.
[6] Steijl, R., Barakos, G. and Badcock, K., "A Framework for CFD Analysis of Rotors in Hover and forward Flight", *International. Journal for Numerical Methods. in Fluids*, Vol. 51, 2006, pp. 819-847.
[7] Morvant, R., Badcock, K., Barakos, G. and Richards, B.E., "Aerofoil-Vortex Interaction Using the Compressible Vorticity Confinement Method", *AIAA J.*, Vol.43, No. 1, 2004, pp. 63-75.
[8] Spentzos, A., Barakos, G., Badcock, K., Richards, B.E., Wernert, P., Schreck, S. and Raffel, M., "CFD Investigation of 2D and 3D Dynamic Stall", *AIAA J.*, Vol. 34, No. 5, 2005, pp. 1023-1033.
[9] Freeman, C.E. and Mineck, R.E., "Fuselage Surface Pressure Measurements of a Helicopter Wind-Tunnel Model with a 3.15-Meter Diameter Single Rotor", NASA TM-80051, 1979.
[10] Chaffin, M.S. and Berry, J.D., "Navier-Stokes and Potential Theory Solutions for a Helicopter Fuselage and Comparison with Experiment", *NASA Technical Memorandum*, 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 Technical Paper*, TP-2001-211260, December 2001.

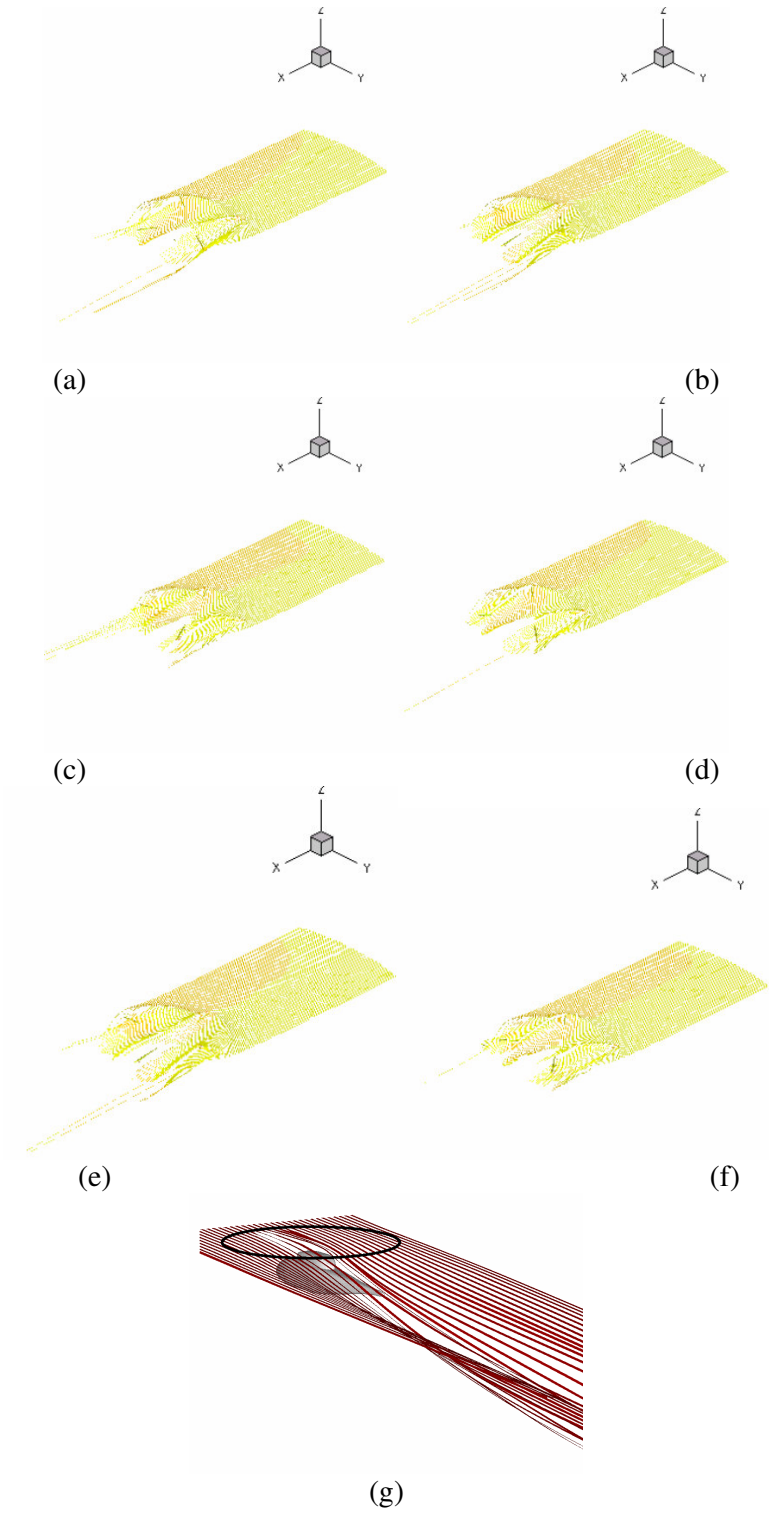


Figure 7: Flow visualization near the sliding plane area between the ROBIN body and the rotor at (a) 345, (b) 355, (c) 25, (d) 45, (e) 90 and (f) 200 degrees of azimuth. The footprint of the rotor blades can be seen as distortions on streaks placed at the sliding plane. (g) Flow visualization of the flow through the actuator disk plane, μ=0.3, 3deg nose down configuration, C_T/σ=0.064.

This page has been purposedly left blank