CFD REQUIREMENTS FOR EFFICIENT SMART-ROTOR ANALYSIS

Rene Steijl, Mark Woodgate and George Barakos

CFD Laboratory
Department of Engineering, University of Liverpool, Liverpool, L69 3GH
United Kingdom

Abstract

A demonstration of the Helicopter Multi-Block CFD solver for the analysis of smart rotors equipped with trailing edge flaps is presented in this paper. The requirement for CFD analysis of smart rotors sets new standards for the efficiency and flexibility of the CFD solvers. Accordingly, the core solver and data structure of HMB had to be revised and upgraded. The mesh deformation and motion algorithm of the solver was also modified to allow for the complex actuation of active blade elements in conjunction with the blade control inputs and aeroelastic deflections. The method has been demonstrated for a four-bladed model rotor equipped with trailing edge flaps and the results showed good convergence behaviour and interesting flow physics. This work highlighted the requirement for experimental data for the validation of such complex CFD tools. With world-wide interest in flapped rotors, a need is now emerging for adequate experiments to complement the development of high-fidelity predictive CFD tools.

1. MOTIVATION

The move towards rotors with active elements is dictated by the need for higher rotor performance and flexibility not available in conventional designs. Active surfaces on rotor blades are to be used to reduce vibration, balance forward flight and hover performances and improve the rotor acoustics. Several recent efforts by helicopter manufacturers and research centres are heading in the direction to develop active rotors and consequently, a need arises for predictive CFD tools than can reliably guide the design process and compute the performance of active rotors. This trend motivates the current research effort. The aim is to enhance the Helicopter Multi-Block (HMB) CFD solver of Liverpool and give it the necessary flexibility to model rotors with flaps. In the past, the HMB solver has been used in conjunction with fixed flaps to enhance helicopter performance [1] as well as to cope with flow control devices [2] on rotor blades. Active flaps were, however, not considered before. The analysis of this new breed of rotors presents CFD with new challenges both in terms of dealing with the complexity of the resulting flow and with the requirements for efficiency, stability and accuracy of the resulting method. In this paper, a modified version of the Helicopter Multi-Block solver of Liverpool [1-3] is presented which enables engineers to simulate flows around smart rotors. There are two main requirements for the CFD method stemming from the need to simulate smart rotors. On one hand, the efficiency of the method must be improved to allow for fine mesh computations able to resolve the flow details introduced by the flow control devices and on the other, the solver must allow for the simulation of rotors with articulation, blade elasticity and deformable blade sections.

2. NUMERICAL METHOD

Building on the HMB framework, the requirement for analysis of smart rotors resulted in several modifications of the solver. A first enhancement includes an improved implicit time-marching method that allows the simultaneous solution of the transport equations for viscous flow (Navier-Stokes) with turbulence (1-equation and 2-equation models) and transition effects (intermittency models) or passive scalars. This efficient algorithm overlaps computation and communication between processors to achieve high efficiency in parallel execution. In addition, a unified algorithm is used which treats all transport equations in a consistent matter. The above has been implemented over a new data structure designed to allow easy expansion of the solver and efficient use of computer memory on distributed parallel computing platforms.

The improvements in the solver were combined with an extension of the handling and representation of the blades within HMB. For this purpose, a hierarchical approach is put forward in combination

with the sliding mesh algorithms of HMB [3], its rotor treatment and aeroelastic model. This technique treats grid movement and deformation due to flaps and control surfaces, combined with the blade actuation and structural deformation and is designed to maintain the original mesh quality.

The combination of the improvements outlined in the previous paragraphs resulted in a sophisticated computational tool able to perform analysis of smart rotors in isolation or as part of a full-helicopter simulation.

2.1. Implicit CFD Method

A unique feature of HMB is that the solver is based on a strongly implicit time-marching technique in contrast to mainstream flow solvers that rely on explicit methods and multi-grid algorithms for convergence acceleration. The semi-discrete form of the governing equations used in HMB is as follows:

$$\left[\frac{V_{i,j,k}}{\Delta t}\mathbf{I}+\frac{\partial \mathbf{R_{i,j,k}}}{\partial \mathbf{W_{i,j,k}}}\right]\Delta \mathbf{W_{i,j,k}} = -\mathbf{R}^{n}_{i,j,k}(\mathbf{W^{n}}) \quad (1)$$

The above equation must be solved for each cell of the CFD mesh and provides the update to the vector of the unknowns ΔW as a solution of a system of algebraic equations formulated on the flux residual (R) of the right hand side and its Jacobian that appears on the left hand side (∂R/∂W).

The global assembly of the discrete equations over the computational domain results in a system of equations of the form Ax=b. The complexity of a direct method to compute a linear system is of the order of N^3, which becomes prohibitive when the total number of equations N becomes large. On the other hand, iterative techniques such as Conjugate Gradient (CG) methods are capable of solving large systems of equations more efficiently in terms of time and memory. CG methods find an approximation to the solution of a linear system by minimising a suitable residual error function in a finite-dimensional space of potential solution vectors. Several algorithms, such as BiCG, CGSTAB, CGS and GMRES, are available in the literature. However, results from this work suggest that the choice of method is not as crucial as the preconditioning. For this reason, the current results use a Generalised Conjugate Gradient method suitable for the non-symmetric matrix A of the resulting system. The preconditioning strategy is based on a Block Incomplete Lower-Upper (BILU) factorisation since it appears to be the most promising and has the same sparsity pattern as the Jacobian matrix - i.e. the sparsity pattern of the Lower and Upper matrices is defined with respect to the sparsity of the un-factored matrix. Furthermore the BILU factorisation is decoupled between blocks on different processors to improve parallel efficiency and this approach does not seem to have a major impact on the solution as the number of blocks increases. The convergence rate, however, changes slightly between different numbers of processors. Implicit schemes require particular treatment during the early stages of the iterative procedure. The usual approach in starting the method is to take a small CFL number and to increase it later on. However, it was found that smoothing out the initial flow by doing some explicit iterations, and then switching to the implicit algorithm was equally efficient. In the present method, a specified number of forward Euler iterations are executed before switching to the implicit scheme.

The Jacobian Matrix has a number of non-zero entries per row. Trying to reduce the number of non-zero entries would have several advantages. First, the memory requirements are lowered. Second, the resolution of the linear system by the GCG method is faster in terms of CPU-time since all the matrix-vector multiplications involved require less operation counts. Finally, the linear system is easier to solve since approximate Jacobian matrices can be used that are more diagonally dominant. All equations are solved simultaneously for the next time level which allows flexibility if the cross-terms are added to the Jacobian matrix. An approximate Jacobian is used for the source term by only taking into account the contribution of the dissipation terms i.e. no account of the production terms is taken on the left hand side of the system.

As a result of the need to solve the Navier-Stokes equations augmented with the transport equations of turbulence and transition models as well as passive scalars for transport of tracers, particles or water droplets in the flow, the resulting matrix has a given sparsity pattern that is shown in Figure 1. This pattern is important to the HMB method since its exploitation allows not only for efficient solution of the equations but also to economies in computer memory and message passing for parallel computations. The sparse Jacobian matrix arising from the discretisation of partial differential equations in which several degrees of freedom are associated at each grid point is composed of dense blocks of non zeros. In the past HMB used a Block Compressed Sparse Row (BCSR) format which assumed 7 non zero blocks per row. To overcome this restriction the Variable Block Row (VBR) format is now used to describe the matrix.

HMB is designed for parallel execution on distributed memory machines and further efficiency is gained by exploiting a variable storage shown in Figure 2. As can be seen, for a typical block with some halo, the storage can be optimised to allow for better memory access and efficient overlap of

parallel communication. A suggested scheme stores the external cell content to be received by each processor at the end of an array (elements 26-36 for the example of Figure 2) and the border values which need to be broadcasted data from neighbouring processors in a separate row (17 to 21). The result of this re-numbering is shown in Figure 2b. The solver is written in C and required no external libraries other than MPI, consequently, it is very portable between systems and has so far been compiled and used in several computer clusters in academia and industry as well as several national computing facilities.

Another area of improvement is related to the data structure used for each block. The arrangement is shown n Figure 3. A small amount of data is shared between all processors and belongs to a single C data-structure. Data storage associated with the geometry and flow variables for each block is allocated per processor. Further down the list, each block side and face is allocated separately. According to the required level of coupling and the required storage for the off-diagonal terms of the Jacobian matrix the solver needs between 2308 (Euler solution) and 3292 bytes of RAM (for turbulent and tansitional flows without the cross-terms in the Jacobian) per cell in the grid. This is substantially low for an implicit method.

2.2. Mesh Treatment for Smart Rotors

HMB used multi-block structured grids that allow for complex geometries to be modelled and accurate solutions to be obtained. The main idea behind the rotor mesh system of HMB can be explained using Figure 4. Each blade is surrounded by a high-quality body-fitted mesh that moves rigidly with it according to the control inputs decided by the rotor trimming method [3]. These blocks are shown as grey in Figure 4a. The grid around the rotor is usually embedded in a “drum” to allow for rotor-fuselage cases to be computed [3]. Sliding meshes allow the communication between mesh blocks with non-matching cells [4.5]. The block boundaries on the drum are shown as red lines in Figure 4a. Flow control elements can then be placed on the blade as shown in Figure 4b. For this case, trailing edge flaps are used at two locations highlighted with blue and green colour. To model the blade motion in the solver, mesh velocities are used which are computed using the time-derivatives of the transformation matrices for blade pitch, flap and lead-lag motions.

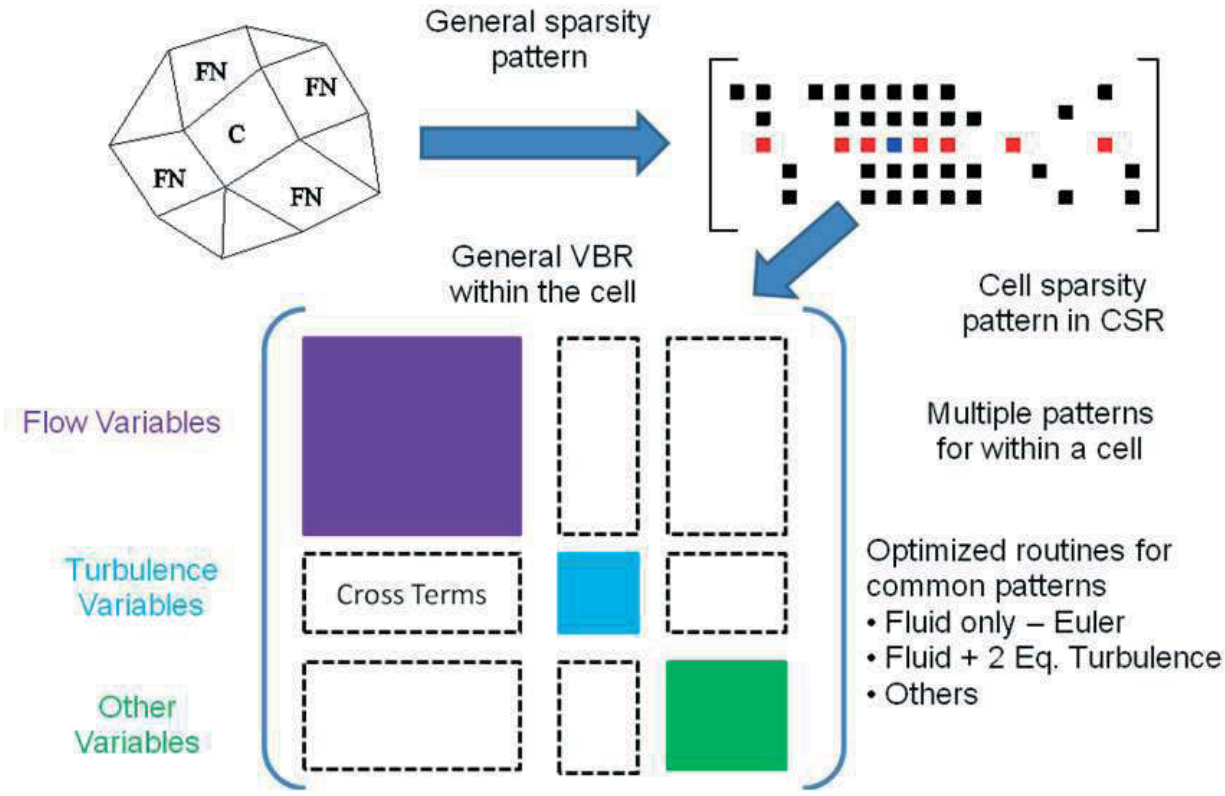


Figure 1: Sparsity pattern of the Jacobian Matrix (stored in Variable Block Row format) and Compressed Sparce Row (CSR) pattern.

$$\begin{aligned}\frac{d\{C_1C_2C_3C_4\}}{dt} &= \{C_1C_2C_3\}\frac{dC_4}{dt} + \frac{d\{C_1C_2C_3\}}{dt}C_4 \\ &= \{C_1C_2C_3\}\frac{dC_4}{dt} + \left[\{C_1C_2\}\frac{dC_3}{dt} + \frac{d\{C_1C_2\}}{dt}C_3\right]C_4 \\ &= \{C_1C_2C_3\}\frac{dC_4}{dt} + \left[\{C_1C_2\}\frac{dC_3}{dt} + \left(C_1\frac{dC_2}{dt} + \frac{dC_1}{dt}C_2\right)C_3\right]C_4\end{aligned} \quad (2)$$

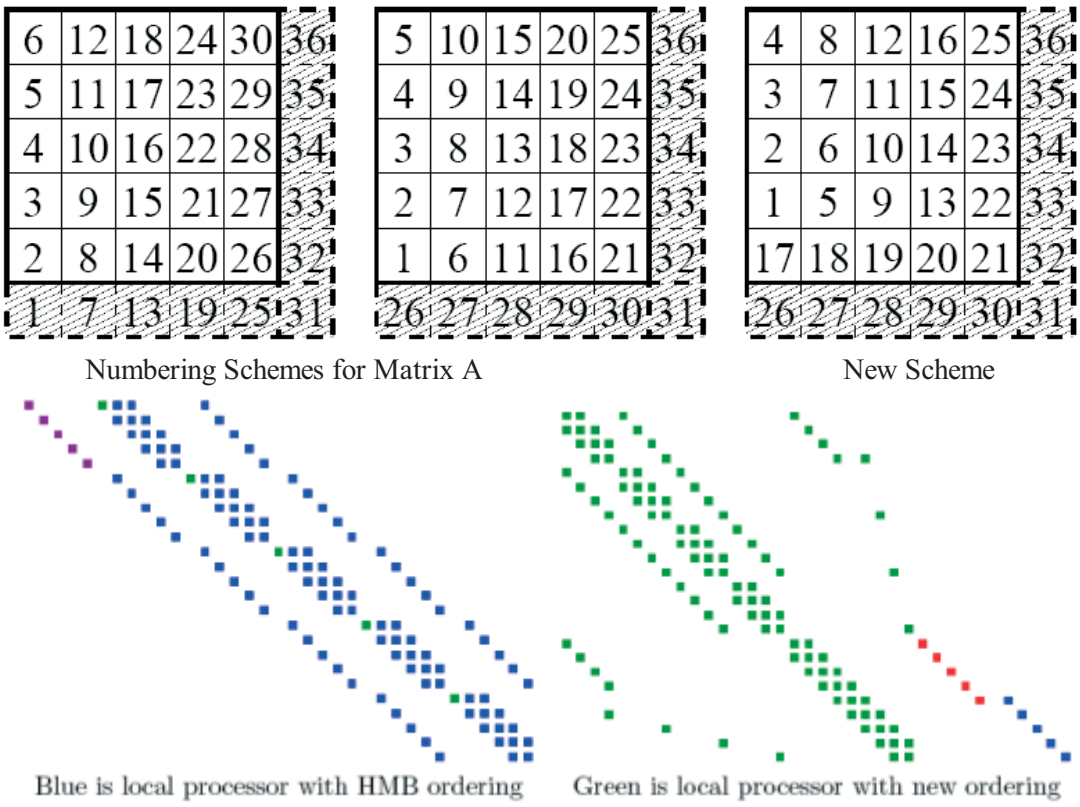


Figure 2: Efficient cell numbering in HMB.

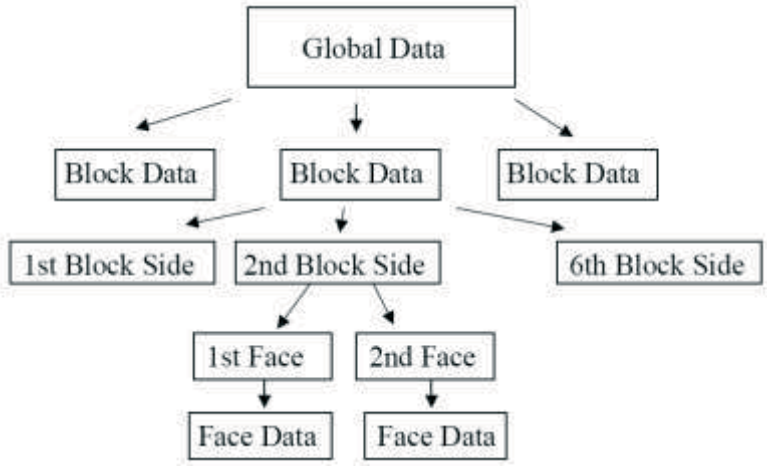


Figure 3: Top-level datastructure of the HMB solver.

$$C_{rot} = \begin{pmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad ; \quad \frac{dC_{rot}}{dt} = \omega \begin{pmatrix} -\sin\psi & -\cos\psi & 0 \\ \cos\psi & -\sin\omega & 0 \\ 0 & 0 & 0 \end{pmatrix} \tag{3}$$

$$C_{pitch} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{pmatrix} \quad ; \quad \frac{dC_{pitch}}{dt} = \dot{\theta} \begin{pmatrix} 0 & 0 & 0 \\ 0 & -\sin\theta & -\cos\theta \\ 0 & \cos\theta & -\sin\theta \end{pmatrix} \tag{4}$$

$$C_{lag} = \begin{pmatrix} \cos\delta & -\sin\delta & 0 \\ \sin\delta & \cos\delta & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad ; \quad \frac{dC_{lag}}{dt} = \dot{\delta} \begin{pmatrix} -\sin\delta & -\cos\delta & 0 \\ \cos\delta & -\sin\delta & 0 \\ 0 & 0 & 0 \end{pmatrix} \tag{5}$$

$$C_{flap} = \begin{pmatrix} \cos\beta & 0 & -\sin\beta \\ 0 & 1 & 0 \\ \sin\beta & 0 & \cos\beta \end{pmatrix} \quad ; \quad \frac{dC_{flap}}{dt} = \dot{\beta} \begin{pmatrix} -\sin\beta & 0 & -\cos\beta \\ 0 & 0 & 0 \\ \cos\beta & 0 & -\sin\beta \end{pmatrix} \tag{6}$$

$$\left[\frac{d\mathbf{x}_o}{dt}\right]_{artic.} = C_1 \left[\frac{d\{C_2C_3C_4\}}{dt}(\mathbf{x}_o - \vec{x}_3) + \frac{d\{C_2C_3\}}{dt}(\vec{x}_3 - \vec{x}_2) + \frac{d\{C_2\}}{dt}(\vec{x}_2 - \vec{x}_1) \right. \tag{7}$$

$$\frac{d\mathbf{x}_o}{dt} = \left[\frac{d\mathbf{x}_o}{dt}\right]_{activeT.E.} + \left[\frac{d\mathbf{x}_o}{dt}\right]_{artic.} \tag{8}$$

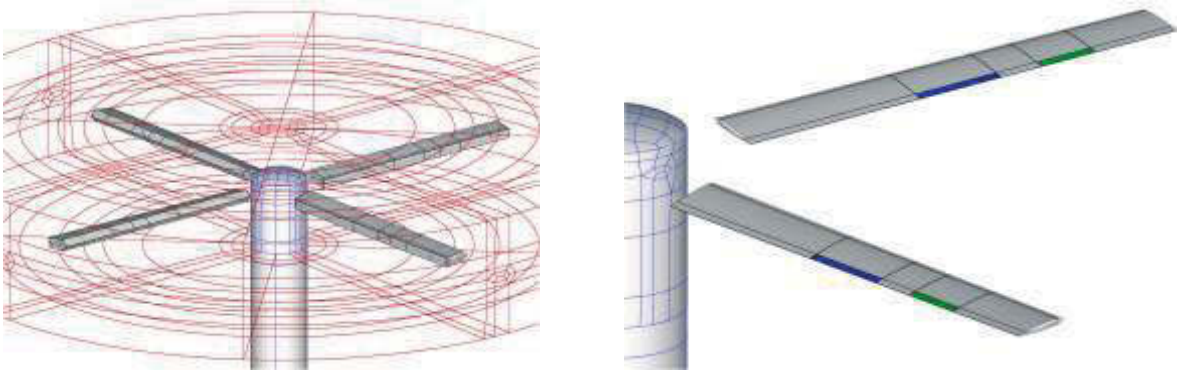


Figure 4: (a) Rigid blocks and rotor "drum" and (b) active blade elements for a 4-bladed model rotor.

The derivative involves a chain calculation using several transformation matrices as shown in equation (2). $C_{1\text{-}4}$ represent the required transformation matrices which are given in equations (3-6). The above transformations, can be performed in any order e.g. Flap-Lag-Pitch etc. The blade rotation results in no grid deformation and the total mesh velocity due to blade articulation is given by equation (7). In addition to the above, mesh deformation due to blade aeroelasticity or active elements that move (live flaps) parts of the blade require an additional grid velocity as given by equation (8).

The mesh outside the rigid blocks is deformed using trans-finite interpolation. The combination of rigid and deforming grids results in high mesh quality as can be seen in Figure 5 where the mesh near the active trailing edges of Figure 4b is show.

The combined effect of the mesh deformation due to blade articulation and the actuation of the trailing edge flap can be seen in Figure 5b.
A near-field view of the mesh is shown in Figure 6 after 3 degrees of flap deflection combined with blade flapping. The mesh deformation did not seem to affect the quality of the flow solution as shown in Figure 6.

2.3. Demonstration Cases

The efficiency of the employed implicit method has been demonstrated for several cases. Indicative results from a simple computation of the flow around an aerofoil are presented in Figure 7. In this figure the effect of the Courant Fredrichs Levy (CFL) number on the obtained solution can be clearly seen. The current method can achieve high CFL numbers (Figure 7a). Figure 7b suggests that the accuracy specified for the solution of the linear system is important though tolerances less than 0.001 resulted in no faster convergence rates. A second observation is related to the recovery of quadratic convergence rate. Figure 7c shows quadratic convergence when a first-order Jacobian discretisation is combined with first-order flux discretisation. The Jacobian is only approximately accurate when the second-order discretizaton is used and for this reason the rate of convergence is less than quadratic.
As can be seen, high CFL numbers are possible and high rate of convergence can be achieved. The implicit solver proved to be robust over a range of cases and for this reason it is used in HMB. The down-side of the implicit scheme is related to the memory needed to store the extra matrix for the jacobian. In practice this means that with the current implementation, about 300 thousand cells can be processed per GByte of RAM in a computer.

To demonstrate the ability of the solver to deal with active elements on rotor blades, a model rotor is considered made out of four untwisted blades using the NACA0012 section. Two flaps are considered inboard and outboard with a length of about 10% of the rotor radius and chords equal to 15% of the local blade chord at each station. The flaps were allowed to move in or out-of-phase and could have the same or different actuation amplitudes. Table 1 presents a summary of the conditions considered in this work. The tests included inviscid and viscous flow at an advance ratio of 0.15. A simple cyclic variation is considered and on top of it the flap was actuated at a five degree amplitude.
Figure 8 presents the normal force coefficients around the azimuth for at two radial stations located at 60 and 80 percent of the radius.
This case did not consider blade flapping or aeroelastic deformation of the blades. The 5/rev input is clearly visible in the sectional loads and appears to be in almost perfect phase with the trailing edge actuation. The effect of the flap is local and for the 60% station the effect of the outboard flap is small. Both flaps seem to have some influence at the 80% station.

Figure 9 shows similar results for case 3 of Table 1 where a viscous flow model is considered. The near-wall resolution was not affected by the grid motion and deformation and again the sectional loads appear to have the 5/rev input of the trailing edge flap. On the other hand, the employed combination of the blade and flap actuation, results in negative flap hinge moments for the whole blade rotation. Figure 10 shows the sectional surface pressure coefficient distributions for the same case for three azimuth angles. The effect of the blended trailing edge flap is evident near x/c of 0.9. For this case, an additional blade deformation in torsion was considered. The case was designed to test the mesh deformation algorithm. The surface pressure distribution is shown in Figure 11 and the local change of the isobars due to the flap is clearly visible. Looking more into the obtained results, Figure 12 shows the higher harmonic content of the obtained solutions. At first, the 5/rev and higher frequencies are retained and the azimuthal variation of the loads is given in Figure 12a along with the sectional loads. The amplitude of the 5/rev and higher actuations is much less as can be seen from the green and blue curves on the Figure. Interestingly, filtering out the 5/rev content showed that 6/rev and higher harmonics were also present. These are evident in Figure 12b.

The current set of results provided evidence that the new solver in HMB and the grid deformation can in principle be used for the analysis of flapped rotors. However, no experimental data were available in the open literature for a quantitative assessment of the predictions. This problem should be addressed before the method is used for the analysis and design of flapped rotors.

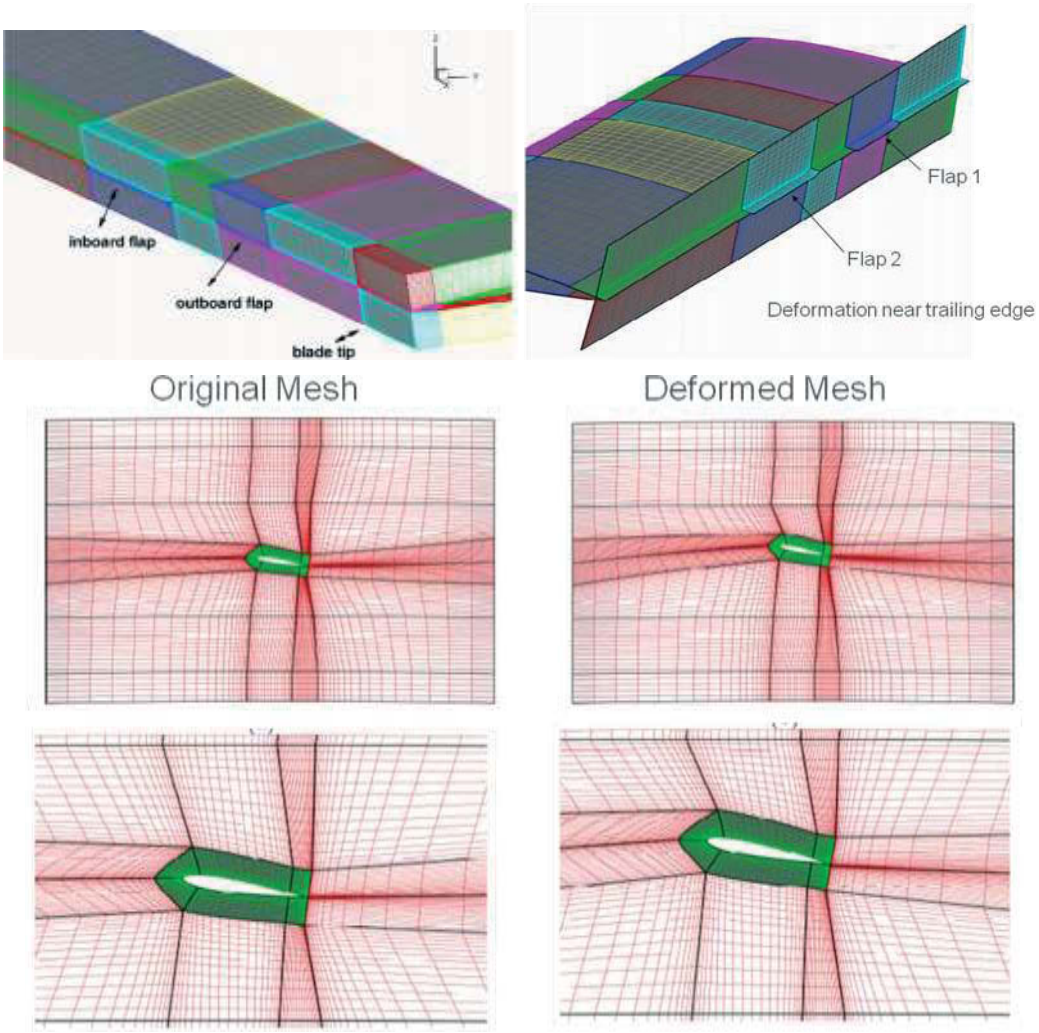


Figure 5: Original and deformed, far- and near-field meshes for an articulated rotor with active blade trailing edge.

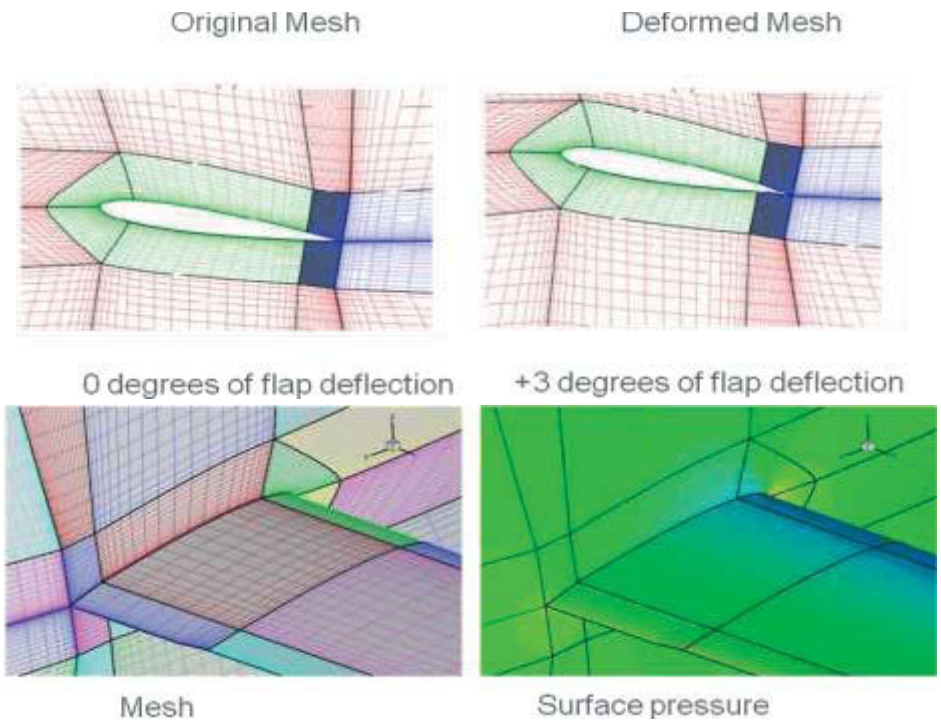
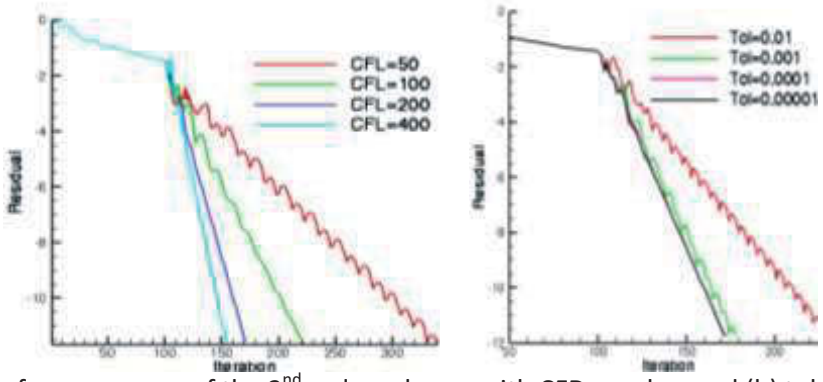
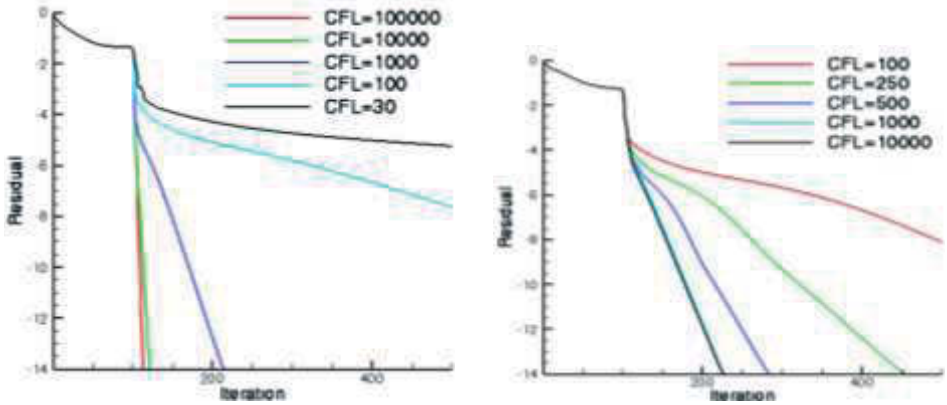


Figure 6: Mesh deformation resulting from combined blade flapping an trailing edge actuation.



(a) Rate of convergence of the 2nd order scheme with CFD number and (b) tolerance of the linear solver.

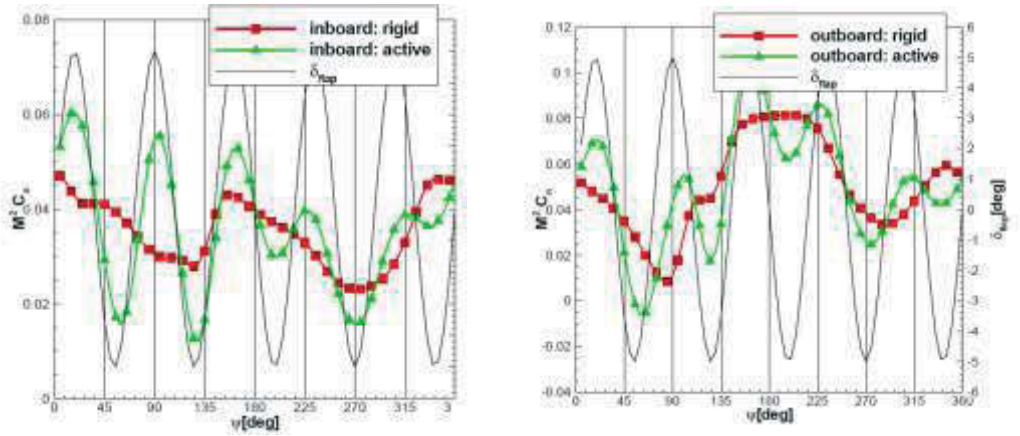


(c) Rate of convergence of the first- and (d) second-order discretization with varying CFL number.

Figure 7: Convergence characteristics of the HMB flow solver.

Table1: Summary of conditions for the demonstration of active elements in HMB.

model	M_{tip}	μ	$\theta(\psi)$	$\delta_f(\psi)$	gridsize	blocks
Euler (rigid)	0.50	0.15	$8^o - 2^o\sin(\psi) + 2^o\cos(\psi)$	$5^o\sin(5\cdot\psi)$	$9.6\cdot10^6$	1602
Euler (active)	0.50	0.15	$8^o - 2^o\sin(\psi) + 2^o\cos(\psi)$	$5^o\sin(5\cdot\psi)$	$9.6\cdot10^6$	1602
$k-\omega$ (active)	0.50	0.15	$8^o - 2^o\sin(\psi) + 2^o\cos(\psi)$	$5^o\sin(5\cdot\psi)$	$13.6\cdot10^6$	1602

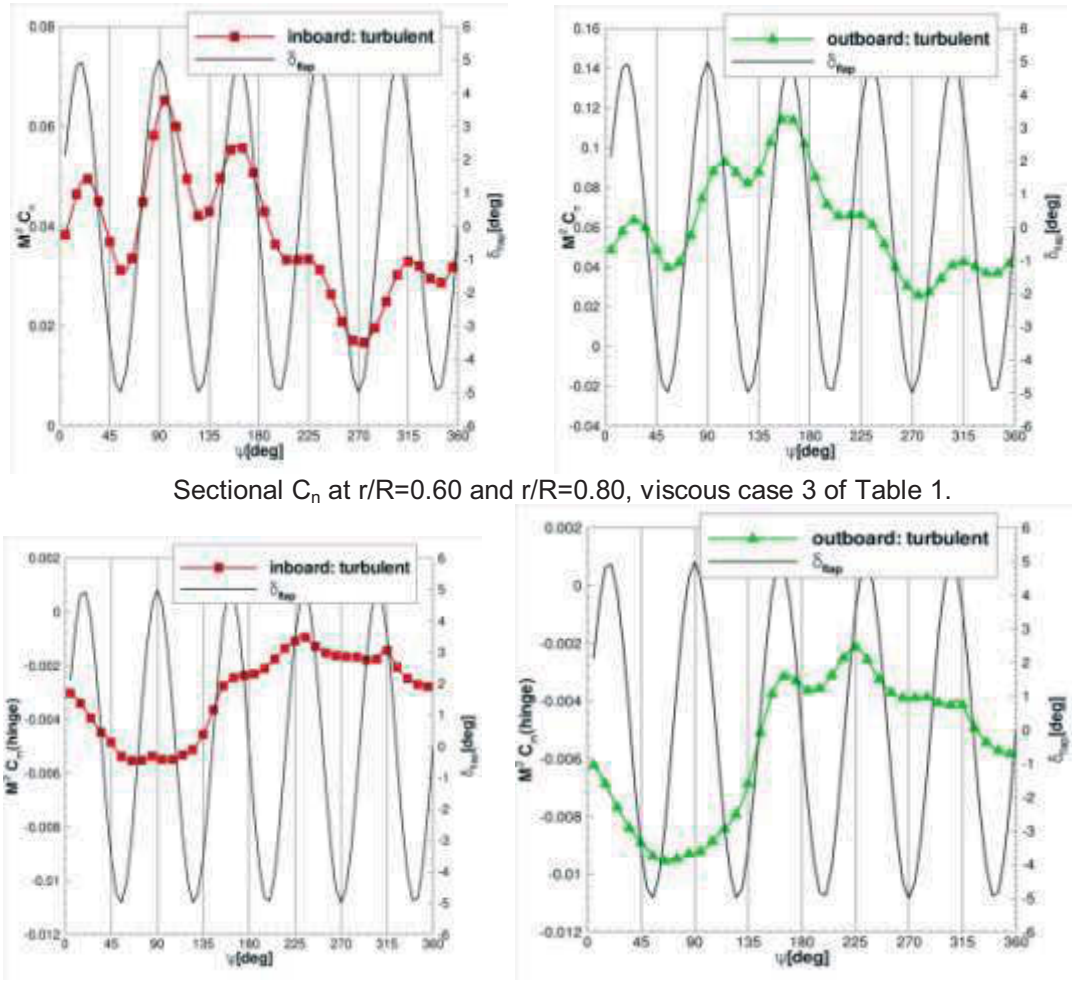


r/R=0.6 r/R=0.8

Figure 8: Effect of active-trailing edge actuation on sectional loads. Inviscid flow simulation, μ = 0.15, θ(ψ) = 8.0°−2.0° sin(ψ)+2.0° cos(ψ), 5° trailing-edge deflection. Blade flapping and elastic twist deformation were not considered.

3. CONCLUSIONS AND FUTURE WORK

Emerging smart-rotor concepts set new challenges for predictive CFD tools. In this paper, the necessary improvements and modifications of the Helicopter Multi-Block solver were discussed and the key features of its CFD algorithm were evaluated. The HMB framework had to be extended to allow for active flow control elements on the blades and it is now possible to use HMB for the analysis of smart rotors. A simple test case was put forward demonstrating the ability of the solver to actuate two trailing edge flaps during forward-flight simulation. Results have been presented for the integrated and sectional loads for a 5/rev flap actuation. The flap actuation was combined with cyclic input, blade flapping, and twist changes. The results show that higher harmonics were present in the flow above the 5/rev input. Unfortunately, no experimental data for a flapped rotor are available and little is known about the structural properties of blades with flaps. Validation against flapped wing data is possible and this is already underway for HMB. The flow physics of the rotor is, however, different and the flap actuation must be validated with respect to unsteady aerodynamic influences. This is most of the times not the case for wing experiments. Based on the current development and the obtained set of results, a need is emerging for experiments in this area to provide a database for CFD validation. With world-wide interest in this research area, an experiment for flapped rotors could be part of an international research effort aiming to build a solid database for this very interesting flow case.



Sectional C_n at r/R=0.60 and r/R=0.80, viscous case 3 of Table 1.

Hinge moment coefficients for the inboard and outboard flaps.

Figure 9: Sectional loads and flap hinge moment coefficient for case 3 of Table 1.

Acknowledgements

The financial support through the REACT programme of AgustaWestland and the DBERR of UK is gratefully acknowledged.

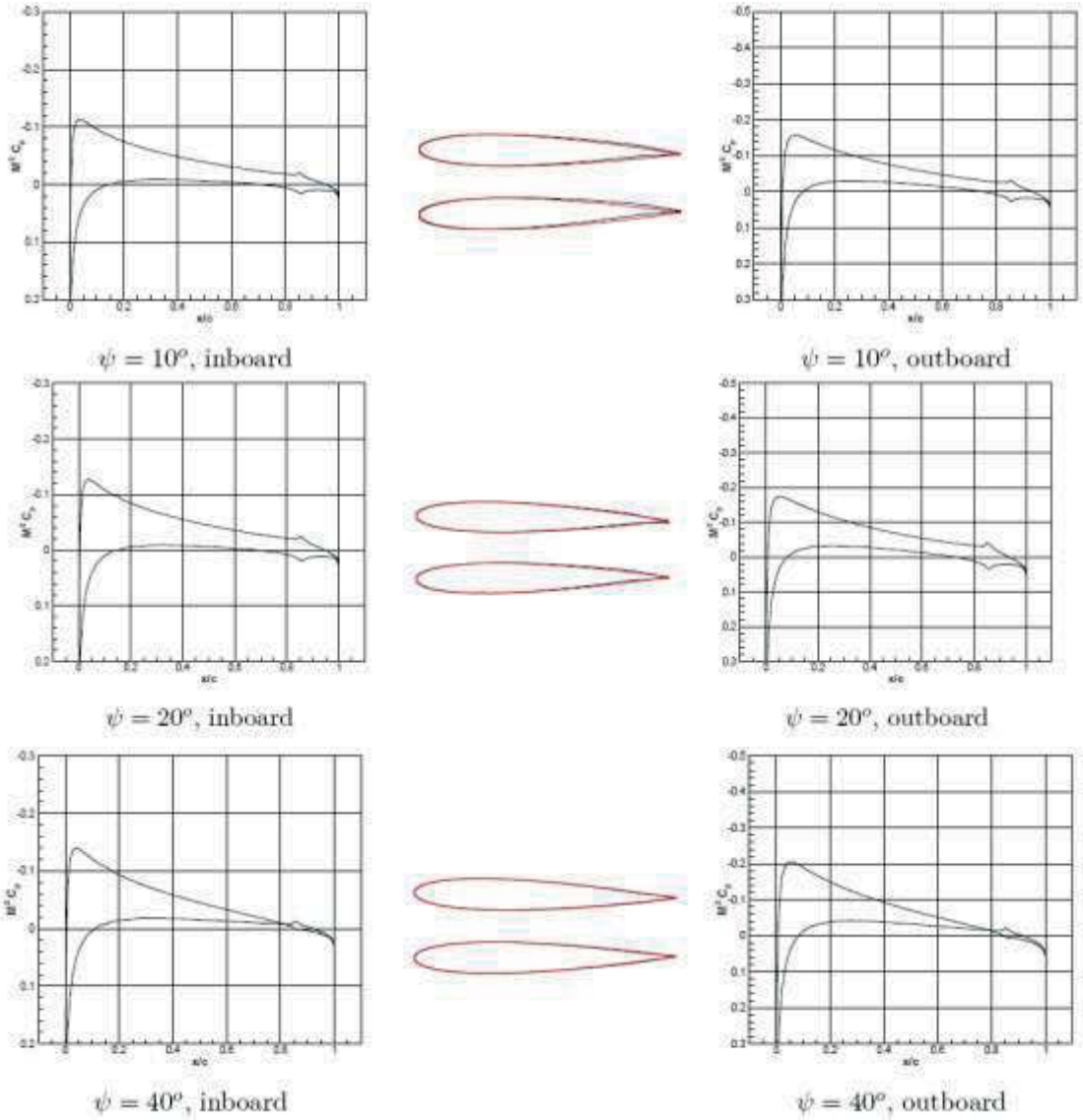


Figure 10: Figure 33: Effect of active-trailing edge actuation on sectional surface pressure distribution. Turbulent flow simulation, μ = 0.15, θ(ψ) = 8.0° − 2.0° sin(ψ) + 2.0° cos(ψ), 5° trailing-edge deflection, elastic deformation. Blade flapping was not considered.

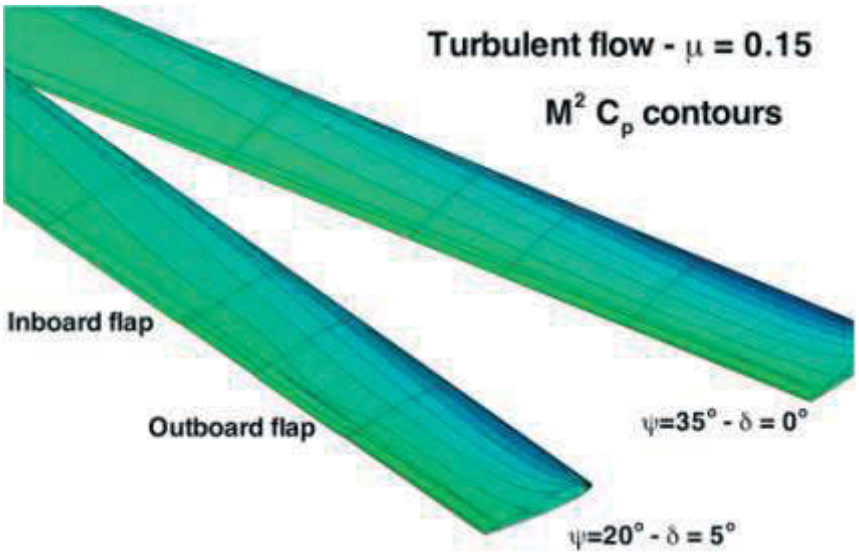
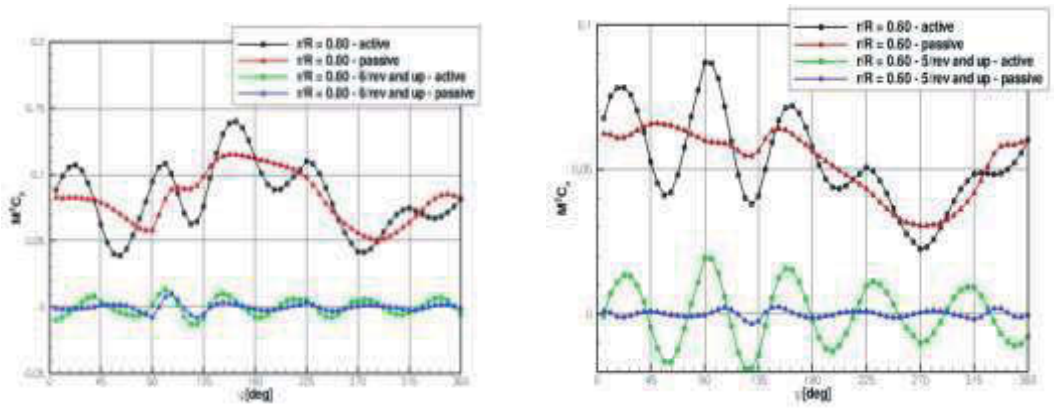
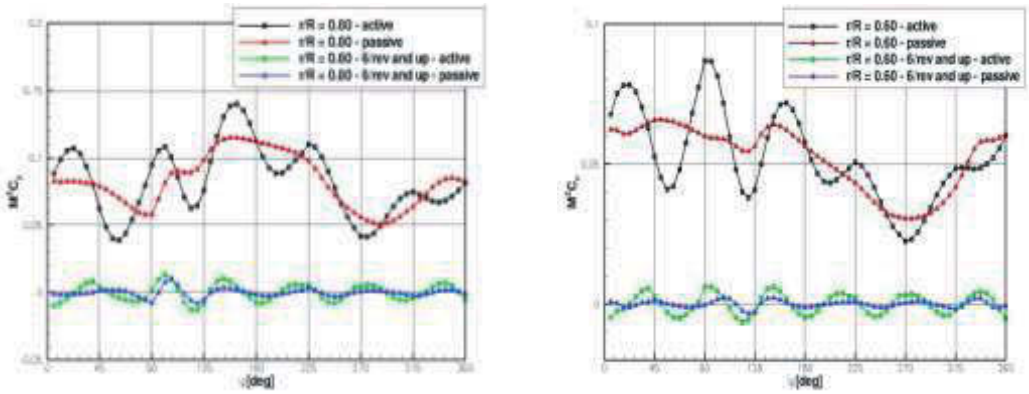


Figure 11: Surface pressure coefficient distribution for deflected and retracted flaps around the azimuth.



(a) 5/rev and higher (in- and outboard sections)



(b) 6/rev and higher (in- and outboard sections)

Figure 12: Higher harmonic content of the sectional loads for case 3 of Table 1.

References

[1] Gagliardi, A. and Barakos, G.N., "Analysis and Design of a Flap-Equipped Low-Twist Rotor for Hover, AIAA J. of Aircraft, Volume 46, Issue 1, pp. 74-84, 2009.

[2] Barakos, G.N., "CFD Simulation of Flow Control Devices for Helicopter Rotors", CEAS / KATnet Conference on Key Aerodynamic Technologies, Bremen, Germany, 20-22 June 2005.

[3] Steijl, R., Barakos, G.N. and Badcock, K., "A Framework for CFD Analysis of Helicopter Rotors in Hover and Forward Flight", International Journal for Numerical Methods in Fluids, Volume 51, Issue 8, pp. 819-847, 2006. DOI: 10.1002 / fld.1086.

[4] Steijl, R. and Barakos, G.N., "Sliding Mesh Algorithm for CFD Analysis of Helicopter Rotor-Fuselage Aerodynamics", International Journal for Numerical Methods in Fluids, Volume 58, Issue 5, pp. 527-549, October 2008, DOI: 10.1002 / fld.1757.

[5] Steijl, R. and Barakos, G.N., "Computational Study of Helicopter Rotor-Fuselage Aerodynamic Interactions", AIAA Journal, Volume 47, Issue 9, pp. 2143-2157, 2009. DOI: 10.2514/1.41287.