Advances in Parallelization and High-Fidelity Simulation of Helicopter Phenomena

Patrick P. Kranzinger, Ulrich Kowarsch, Matthias Schuff, Manuel Keßler, and Ewald Krämer

Abstract A weak and strong scaling study is presented which shows substantial improvements to the scalability of the CFD solver FLOWer by introducing a node-to-node MPI communication strategy. Furthermore, an overview of an extremely flexible and reusable CFD-CSD coupling interface is giving. It is able to handle unstructured, structured, and overset meshes without topology limitations and performance drawbacks. Finally, using these new capabilities a full helicopter configuration is investigated with regard to its aeroacoustic noise emission.

1 Introduction

The recent increase of computational power available at HLRS allows increasing the case size of aerodynamic helicopter investigations accordingly. In combination with newly implemented higher order methods, the numerical exploration of helicopter specific phenomena like noise emissions, tail shake or dynamic stall, as well as in-ground effect flights becomes possible [1, 2].

However, the performance increase of computing clusters mainly resulted from increasing the number of available computing cores, instead of significantly increasing the per-core performance itself. This burdens great demands on scalability on the used CFD codes. Substantial improvements of the scalability of the CFD code FLOWer by using intra-node shared memory for communication purposes will be shown.

In addition, a new CFD-CSD coupling library implementation will be presented, which enables the usage of the increased computational power for a large variety of future studies, requiring mesh topology independent loads and moments evaluation, e.g. for applying automated optimizers, mesh-to-mesh interpolation and parameter and time dependent mesh deformation. It is completely reusable within any structured and unstructured CFD code, and offers a new level of grid and application flexibility.

P.P. Kranzinger (✉) • U. Kowarsch • M. Schuff • M. Keßler • E. Krämer
Institut für Aerodynamik und Gasdynamik, Universität Stuttgart, Pfaffenwaldring 21, 70569 Stuttgart, Germany
e-mail: kranzinger@iag.uni-stuttgart.de

W.E. Nagel et al. (eds.), *High Performance Computing in Science and Engineering '15*, DOI 10.1007/978-3-319-24633-8_31

With these new options available, an earlier H145 isolated rotor investigation was extended to a full helicopter configuration. The influence of the fuselage to the aeroacoustic noise emission, in terms of e. g. fuselage shading, reflection and diffraction effects, will be investigated. Furthermore, for acoustic investigation the anti-torque Fenestron system is taken into account as well.

2 Numerical Methods

2.1 Computational Fluid Dynamics (CFD)

For high-fidelity aerodynamic investigations, the block structured finite volume Reynolds-averaged Navier-Stokes (RANS) CFD code FLOWer, initially developed by the German Aerospace Center (DLR) [3], is used. For the closure of the RANS equations various types of turbulence models are available. For helicopter and contra rotating open rotor investigations, the Wilcox k-ω turbulence model [4] with a fully turbulent far field flow state has shown best robustness and accuracy. The time discretization is achieved by integrating the governing differential equations with the implicit dual time-stepping approach according to Jameson [5]. The consideration of grid motions using an Arbitrary Lagrangian Eulerian approach enables the code for helicopter flow simulation. In addition, the Chimera technique for over-set grids simplifies the meshing of complex helicopter geometries like rotor-fuselage configurations including relative grid movements. To consider the effects of fluid-structure interaction on the rotor blade, the mesh is deformed to a given structural deformation of the blade in each time step. The efficiency of the computation is achieved by a multi-block structure of the grid to enable parallel computing.

Within the last years, the CFD solver was extended by IAG [6] with different methods of fifth order spatial WENO schemes to guarantee a detailed conservation of the flow field and especially vortices. Besides the basic WENO scheme according to Jiang [7], an improved order preserving WENO scheme near discontinuities denoted as WENO-Z according to Borges [8] is available. The compact reconstruction WENO (CRWENO) scheme [9] also available in FLOWer improves the spectral resolution with a higher efficiency than the basic WENO scheme. Irrespectively of the WENO scheme in use, the base reconstruction at the cell boundaries results in a Riemann problem, which is solved using the upwind HLLC scheme of Toro [10]. The viscous fluxes are solved with conventional central differences of second order accuracy.

2.2 Flight Mechanics (FM) and Structural Dynamics (CSD)

For helicopter applications, a proper reproduction of the flight state including the aero-elasticity of the rotor blades is mandatory. Especially in forward flight, blade elasticity influences the aerodynamic behavior and force generation substantially.

At IAG, the structural deformation of the rotor blades is modeled using the Comprehensive Analytical Model of Rotorcraft Aerodynamics and Dynamics (CAMRAD II) code [11] as part of a weak coupling scheme: The CAMRAD II code provides solutions for the blade deformation and flight kinematics modeling the rotor blades as Euler-Bernoulli beams with isotropic material and elastic axis. For aerodynamics load estimations a low-fidelity aerodynamics model based on lifting line theory and two dimensional steady airfoil data tables is used. The initial deformation and trim-angle values provided are used for performing a CFD based aerodynamics simulation, providing load results of high-fidelity. By correcting the internal low-fidelity loads evaluation of CAMRAD II with the CFD results, the CSD internal aerodynamics are successively replaced by the CFD solution, leading to blade dynamics based on CFD loads with deformation and deflection calculated with CSD.

In order to fit the specified global forces and moments, in case of an isolated rotor trim the collective and two cyclic control angles are determined with a fixed rotor orientation, known as the wind-tunnel trim. For a complete helicopter, three additional degrees of freedom are taken into account for the spatial fuselage orientation and the tail rotor thrust. This approach is designated as a free-flight trim.

3 Tool Chain Improvements

3.1 Advanced CFD-CSD Coupling Implementation

For more than a decade now, the performance of a single computing core has been nearly stagnating. Present day's high performance computing clusters offer 100,000 cores and more instead. Thus, efficient parallelization is a key skill for being able to use the power of current and future computation clusters.

Due to restrictions on the mesh topology when using the former CFD-CSD coupling implementation, the preparation of rotor blade meshes in consideration of efficient parallelization was strictly limited [12]. After completing the block splitting process, some blocks remained, whose geometric dimensions could not be further shrunk. Increasing the spatial discretization would lead to a higher number of grid nodes per block. Hence, the power of next generations of supercomputers could not be used adequately.

In addition, up to now resulting line loads and integral sectional loads were also evaluated based on the same mesh topology depending scheme. Geometric changes e.g. of the twist angle distribution or due to parameter variations describing the base shapes of rotor blades, cause a variation of the load evaluation points, which entails uncertainties when comparing results of parameter studies. Common loads for overset grids could not be evaluated considering the overlapping area appropriately, which severely constrained, the usage of Chimera setups for modeling rotor blades, a promising method for increasing the mesh density at the blade tip locally.

To overcome these limitations and because of the ongoing development of the unstructured Discontinuous Galerkin code SUNWinT [13] and the free wake method FIRST, a mesh deformation algorithm and a load integration algorithm were developed. Both algorithms are implemented as separate object oriented C++ libraries. The key design goals were reusability on current and future codes, a software design which allows easy extensions for future applications, no limitations concerning mesh topologies, and attaining the performance of the former FLOWer internal implementation.

For achieving these aims, radial basis functions (RBF) have been selected as backbone for the deformation algorithm. The load integration algorithm uses a three-dimensional spatial search tree (octree) for evaluating overlapping regions and computing sectional and line loads.

Both algorithms are able to handle overlapping Chimera structures and structured as well as unstructured meshes, respecting relative motions (e.g. if slotted leading and trailing edge flaps are modeled) [14].

3.1.1 Deformation

Radial Basis Function Interpolation

RBF can be used for interpolation of discrete data in an n-dimensional space. The deformation of a three-dimensional CFD mesh can thus be interpreted as an interpolation of the known discrete deformation of the surface in the surrounding area.

RBFs are real-valued functions whose value only depend on a certain distance [15], so that

$$\phi(\mathbf{x}, \mathbf{d}) = \phi(\|\mathbf{x} - \mathbf{d}\|)$$

where $\mathbf{x}$ is the evaluation point and $\mathbf{d}$ the point of reference. These functions can be used for interpolating data by defining a function for each data point and summing them up. When deforming volume meshes, the data point is a point on a surface and the data to be interpolated is its displacement in the deformed state. By adding a constant and linear polynomial to this sum, offsets and linear parts of the data are represented directly analytically. When handling CFD meshes, translation and rotation in space can be interpreted as x, y, z-offset and a linear function of x, y, z.

The displacement of an arbitrary point $\mathbf{p}$ may be described as

$$s(\mathbf{p})_{x,y,z} = \sum_{i=1}^{n} \Big(\alpha_{i\ x,y,z} \phi \left(\|\mathbf{p} - \mathbf{p}_{surf_i}\| \right) \Big) + q(\mathbf{p})_{x,y,z}$$

where α_i is a weighting factor for each RBF and $q(\mathbf{p})$ is an arbitrary linear polynomial [16]:

$$q(\mathbf{p})_{x,y,z} = [\beta_0\ \beta_1\ \beta_2\ \beta_3]_{x,y,z} \cdot \begin{bmatrix} 1 \\ \mathbf{p}_x \\ \mathbf{p}_y \\ \mathbf{p}_z \end{bmatrix}$$

The requirement is that there is a number of n reference points—or *surface points*—$\mathbf{p}_{surf}$, that have a known deformation.

Thus, the displacement of these surface points must exactly map with the $s(\mathbf{p}_{surf})$ such that

$$s(\mathbf{p}_{surf_i})_{x,y,z} = \mathbf{d}_{b_i\ x,y,z}$$

with $\mathbf{d}_{b_i}$ being the known displacement of the deformed surface points.

As averaged rigid body motions should be solely covered by the polynomial part $q(\mathbf{p})$, the requirement

$$\sum_{i=1}^{n} \alpha_i \mathbf{P}_{b_i} = 0$$

with

$$\mathbf{P}_{b_i} = \left[1\ \mathbf{p}_{surf_i\ x}\ \mathbf{p}_{surf_i\ y}\ \mathbf{p}_{surf_i\ z} \right] \tag{1}$$

has to be fulfilled [17].

Eventually, an equation system can be built and the coefficients for the basis functions (α) and the rigid body motions (β) can be computed by inverting the resulting dense matrix $\mathbf{M}_{complete}$:

$$\begin{bmatrix} \mathbf{d}_{b\ x,y,z} \\ 0 \end{bmatrix} = \underbrace{\begin{bmatrix} \mathbf{M}_{bb} & \mathbf{P}_b \\ \mathbf{P}_b^T & 0 \end{bmatrix}}_{M_{complete}} \begin{bmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\beta} \end{bmatrix}$$

where $\mathbf{M}_{bb}$ with row i and column j is the evaluation of the basis functions for the reference points between each other $\phi(\mathbf{p}_{surf_i}, \mathbf{p}_{surf_j})$ and $\mathbf{P}_b$ is an $n \times 4$ matrix with the row i given as in (1).

As $\mathbf{M}_{complete}$ contains no information of the actual displacement, but only information of the distances of each node to every other in non-deformed state, its inversion needs only to be executed once as long as the non-deformed surfaces will not change.

Detailed description of RBFs can be found in [15]. A detailed study of different RBFs for deforming volume meshes has been done by [17].

Performance

The performance of a RBF based deformation algorithm depends on two things. Firstly, the inversion of a densely populated $(n_{surf} + 4) \times (n_{surf} + 4)$ matrix. As a rule, this is a $\mathscr{O}(n_{surf})^3$ problem and can be solved by LU decomposition or Gaussian elimination. Secondly, applying each RBF coefficient to each volume grid node, where the number of radial basis function coefficients is equal to the number of selected surface mesh points. Hence, the complexity for each time step is $(n_{surf} \cdot n_{volumemesh})$. As n_{surf} can be kept mostly constant with increasing case size (as shown in [14]) the resulting deformation algorithm complexity is $\mathscr{O}(n_{volumemesh})$.

For computational setups with about 35,000 mesh points per MPI process, representing an average per process workload for the midterm future, the wall time required for applying the deformation is < 1 s, which is comparable to the performance of the original FLOWer internal algorithm. Increasing the case size will not affect the wall time required for deforming the meshes, as the number of mesh points per MPI process is kept constant.

Validation

A trim validation with an isolated five bladed rotor at a flight speed of 125 kts has been conducted. As reference, the former Hermite interpolation deformation was used [12].

In Fig. 1 the pitch angles are plotted against the iteration steps for both trim runs. The convergence of these angles is almost the same, with some neglectable discrepancy which is due to a slightly different twist of the elastic rotor blade

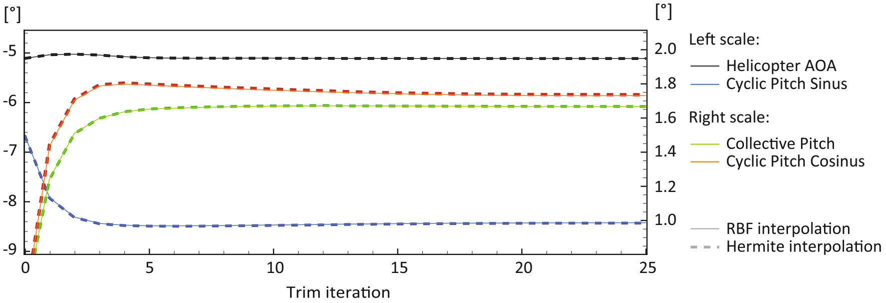

Fig. 1 Angles of pitch and helicopter's angle of attack (AOA) during trim iterations

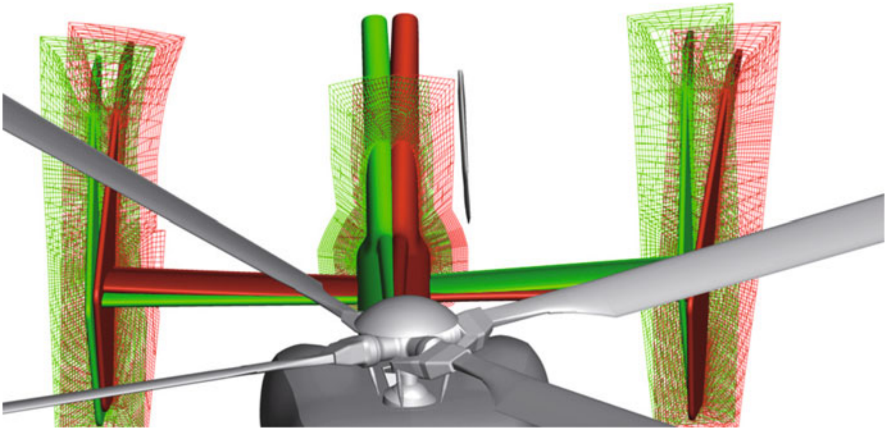

Fig. 2 Deformed near surface meshes of strongly coupled tail boom in conjunction with a weakly coupled main rotor

because of the misinterpretation of the beam angles in the former algorithm (as mentioned in [14]).

Further Applications

The implementation allows the usage of different information sources to deform various components of the complete simulated CFD environment separately. Exemplarily, the deformation of a ground boundary layer to represent ground roughness for in-ground-effect forward flight simulations, or strongly coupled tailboom deformation in combination with weakly coupled structural rotor dynamics (cf. Fig. 2), have become possible.

3.1.2 Load Integration

The load integration algorithm collects the complete surfaces of all simulation components as specified from the CFD solver, disassembles all cell faces to triangles, and sorts the resulting triangles by their center of area into an octree. For a helicopter rotor blade, the resulting octree is exemplarily shown in Fig. 3.

For respecting overset areas of surfaces, an algorithm was implemented based on the octree data structure, which allows finding all overset objects with a complexity of $n\log(n)$. The actual overlap is computed by projecting two possible overlap candidates to a common plane. The overlapping area is identified by a geometric comparison and summed up for each triangle separately. The actual overlap is later on respected by defining a specific overlap factor f_o for each element, which is

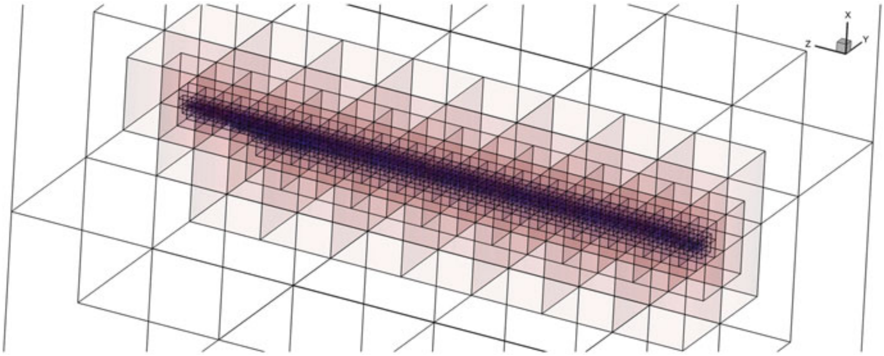

Fig. 3 Visualization of resulting octree of a helicopter rotor blade

always taken into account when computing loads and moments:

$$f_o = 1 - 0.5 \cdot \frac{A_{overlapping}}{A_{complete}}$$

Hence, for a complete overlap the factor results to 0.5; for no overlap it is 1.0. Table 1 shows validation results for overlapping rotor blade surfaces.

Line loads are now mesh topology independently evaluated at defining planes, cutting all components selected for load integration. Using the octree, all triangular faces cut by the plane are identified. All segments resulting from cutting the faces by the plane build up together a two-dimensional outline of the cut structure on the cutting plane. Line loads are now calculated by summing up the length of the resulting segments l_i multiplied with the specific overlap factor $f_{o,\ i}$, and the three-

Table 1 Global loads and moments for partly, and completely overlapping rotor blade surfaces in relation to stand-alone blade surface

	No overlap	Partly overlapping (‰)	Completely overlapping (‰)
ΔF_x	Reference	1.19	0.68
ΔF_y	Reference	1.03	0.73
ΔF_z	Reference	1.46	0.73
ΔM_x	Reference	0.90	0.63
ΔM_y	Reference	1.02	0.63
ΔM_z	Reference	1.10	0.55

dimensional force vector of all cut triangles normalized by their surface areas $\mathbf{f}_i/A_i$:

$$\mathbf{f} = \sum_{i=1}^{n_{cut}} l_i \cdot f_{o,\,i} \cdot \frac{\mathbf{f}_i}{A_i}$$

Resulting moments are evaluated the same way relating to a global or sectional reference point. Therefore, the distance vector from the reference point to each segment center $\mathbf{d_i}$ is calculated and taken into account

$$\mathbf{m} = \sum_{i=1}^{n_{cut}} l_i \cdot f_{o,\,i} \cdot \frac{\mathbf{f}_i}{A_i} \times \mathbf{d_i}.$$

Integral sectional loads, demanded e.g. by CAMRAD II, are also evaluated by defining cutting planes. However, all triangles between two cutting planes are taken into account with their complete three-dimensional surfaces. The cut triangles are split into a part, which lies in the specific section, and a part outside of the specific section.

In all cases, the resulting algorithm has a complexity of $\mathcal{O}\left(n_{surfpts} \cdot log(n_{surfpts})\right)$. For current computation setups with up to 400 million volume mesh cells and an appropriate surface discretization, the wall time required for computing all loads required for post-processing and CFD-CSD coupling is below 100 ms (Fig. 4).

3.2 Improvements in Code Scaling

The suitability of the FLOWer code has been demonstrated in the past in several successfully performed highly parallel simulations, which have been presented at the HLRS annual user workshop [6]. The CFD code is parallelized by the distribution of the overall grid cells into work packages for each MPI rank defined by mesh blocks. Each MPI rank runs on a physical CPU core, whereby all available cores per node are used. At each block boundary, data exchange in terms of an MPI message has to be performed to proceed the global solution. This leads to a large number of MPI messages. When increasing the number of MPI processes, the wall time required for exchanging the blocks' boundary data became dominant. It was found, that the performance limitations were not caused by the total size of data exchanged during the rank-to-rank communication, but the number of MPI messages itself was limiting the performance.

This issue was overcome by introducing intra-node shared memory with one MPI process managing all MPI-ranks' boundary exchanges on the node. Thereby, a transition from core-to-core to node-to-node communication was achieved. Intra-node boundary communication is directly performed over the MPI-decoupled shared memory. In addition, OpenMP directives were implemented for a hybrid

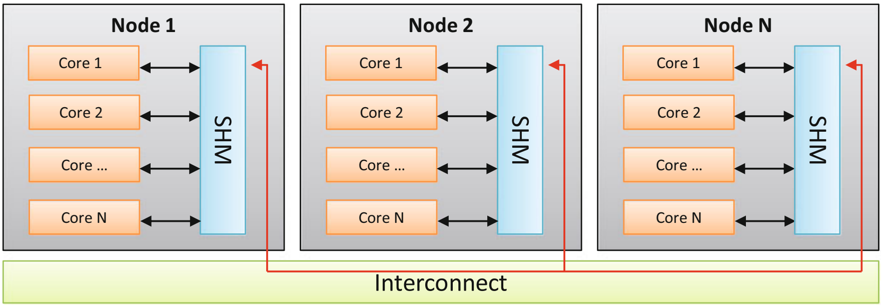

Fig. 4 Methodology of advanced MPI communication on a SMP system

parallelization approach, where memory demands prohibit the utilization of all cores in a node for MPI ranks.

A pure MPI scaling study was performed on the Hermit, and on the new Hornet system to evaluate the current efficiency of the CFD code FLOWer. A reference simulation with 135 million grid cells served as basis, which represents an average simulation size for the forthcoming research topics. The FLOWer code was compiled using the ifort respectively the crayftn compiler available on the Hornet, respectively on the Hermit user environment.

As basis served the numerical setup used for complete helicopter simulations featuring i.e. full viscous flux computation incl. turbulence modeling, and convergence acceleration based on the multigrid method. Running the reference simulation on the Hermit using 4.096 cores, computation of one subiteration takes 53.58 μs/grid point. Using the updated MPI inter-process communication this value could be reduced to 44.72 μs/grid point. In typical helicopter simulations 30–60 subiterations lead to numerical convergence within a time step.

3.2.1 Strong Scaling

The usual core count for the strong scaling simulation size is in the range between 2,000 and 6,000 cores depending on the available computational resources and time constraints (cf. Fig. 5a). The strong scaling shows highly satisfying results with only minor deviations from ideal scaling.

3.2.2 Weak Scaling

In case of the weak scaling the test run using 4,008 cores equals the strong scaling simulation case with 135 million grid cells. Slight unexpected variations are observed with a speed-up over 1.0 with increasing core amount (cf. Fig. 5b). However, this variation is within the usual scattering of node performance and intranode loading, which is observed to vary in the range of ±10 %. Even the 270 million

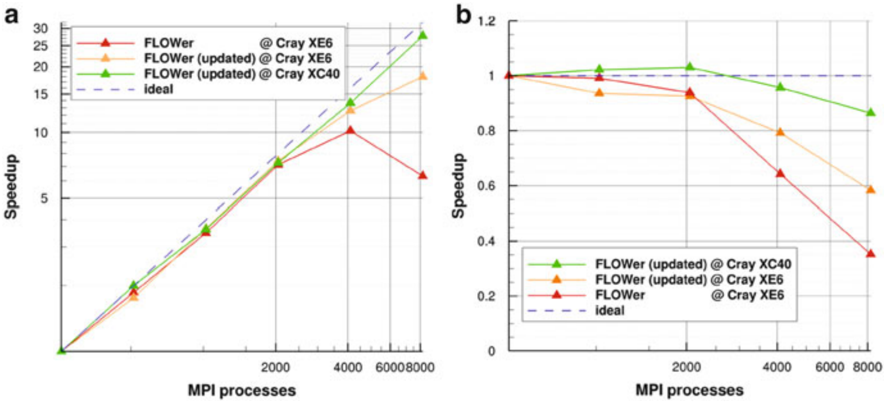

Fig. 5 Pure MPI FLOWer scaling study on HLRS systems. (**a**) Strong scaling (**b**)Weak scaling

cells test case using 8016 cores shows reasonable performance with ∼85 % of the reference computational speed with a total memory consumption of 4.06 TB.

Studies not listed here have shown efficient scaling using OpenMP in addition to the node-to-node based communication up to 32,000 cores.

4 Complete Helicopter Simulation

With the higher order methods and the improved parallelization, a further step in the investigation of fluid-structure interaction phenomena can be launched. Especially simulations of aeroacoustic noise benefit from a detailed preservation of pressure disturbances to map interaction effects.

For an engineering-oriented investigation an H145 helicopter configuration of Airbus Helicopters is investigated. To evaluate possible further improvements, the demand for a highly detailed investigation of the aerodynamics is certainly challenging.

4.1 Setup and Simulation

Figure 6 shows the helicopter surface and the mesh components highlighted with different colors. Overall 59 separate meshes, partly generated with an automated mesh generator [18], are required to reproduce the helicopter surface including their different relative movements, e.g. the rotor-blades, rotor-head and Fenestron tail-rotor system. Edges mark the block splitting within the individual meshes, to enable an efficient parallelization of the simulation on high performance systems. A maximum of 38,000 cells in each grid unit is not exceeded, which defines

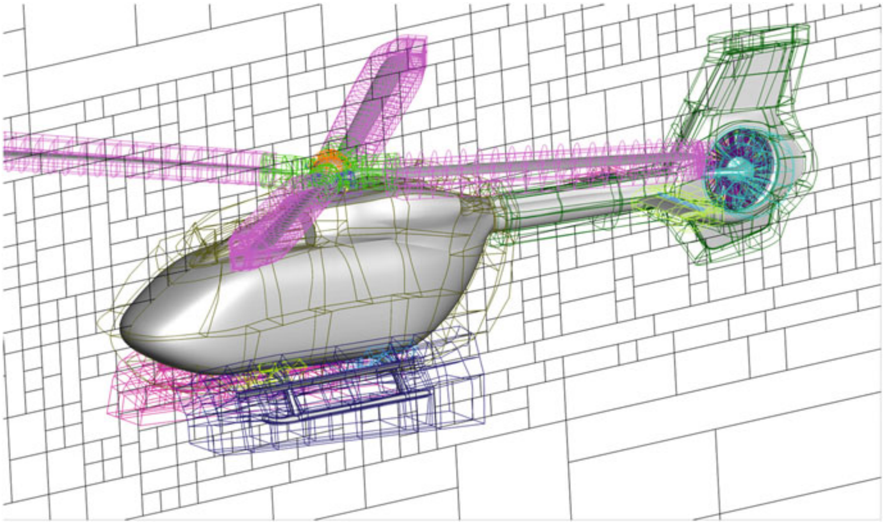

Fig. 6 Mesh blocking of the body grids

Table 2 Grid components of complete helicopter CFD setup

Component	No. of blocks	No. of cells (mio.)
Background	6563	106.3
Main rotor blade	4× 230	4× 6.9
Blade root	4× 56	4×1.0
Rotor hub system	723	9.2
Fuselage	1365	26.7
Skid system	375	7.1
Fenestron® stator + rotor	670	11.9
Total	10876	191.7

the parallelization limits of the setup. The overall mesh size is 192 million grid cells, which represents a high-fidelity simulation in the field of helicopter CFD investigations. The different grid components are listed in Table 2. Focus is set on vortex convection and mapping of vortex-structure interactions using the higher order methods. Since the method works best for Cartesian meshes, the body meshes are embedded in a Cartesian background mesh with refinements in the vicinity of the helicopter's surface. The usage of hanging grid nodes enables a coarsening towards the far field to ensure a reasonable grid size. With this approach more than 50 % of the overall cells are located in the Cartesian background mesh to guarantee best higher order results.

Different numerical methods to improve the physical behavior of the helicopter are taken into account. Besides aeroelasticity of the rotor blades, the mass flow through the engines is considered by a prescribed pressure at the inlet and exhaust.

The exhaust flux is furthermore prescribed with the average exhaust temperature of this helicopter type. Especially the mass flux influences the fuselage wake significantly in terms of occurrence and extent of separation areas.

As presented in previous HLRS annual reports of the HELISIM project, a free-flight trim technique has been established, which has been applied for this investigation [19].

The convergence is mainly dependent on physical processes in the flow field. As numerical convergence is already achieved after a few time steps, physical convergence is the limiting factor. To achieve physical convergence of the flow field starting vortices must have convected out of vicinity of the helicopter. For this case, three complete main rotor revolutions were required until all aerodynamic structures caused by startup effects have left the area of concern. Physical convergence of the complete CSD-CFD-coupled model has been achieved after eight trim iterations. For each trim iteration one complete rotor revolution has been computed with a time step of 0.5°. For better acoustic analysis, three further revolutions with a shorter time step of 0.25° has been computed after achieving convergence of the complete model.

4.2 Aerodynamic Results

Using the higher order methods ensures a highly preserved vortex structure around the helicopter (cf. Fig. 7). Besides the preservation of the blade vortex interaction (BVI) relevant main rotor blade tip vortices, the turbulent wakes of the fuselage and

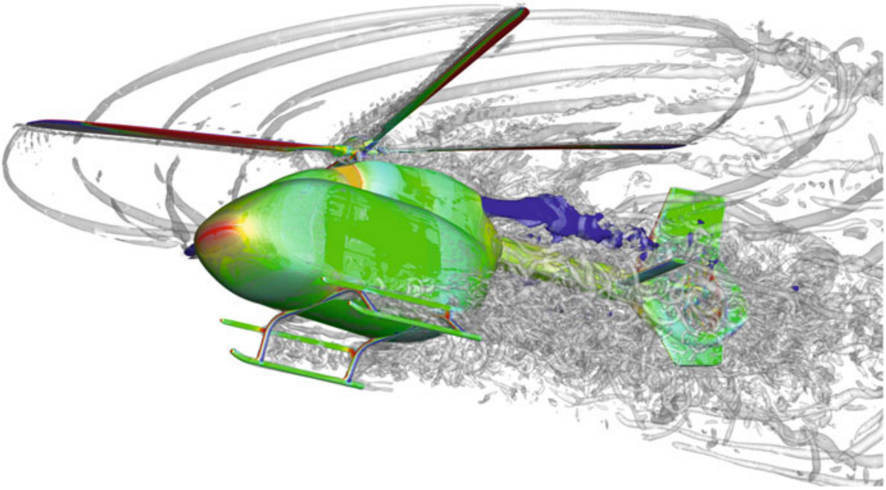

Fig. 7 λ_2-visualization of the flow field around the H145 in a BVI relevant descent flight with pressure contours on the surface. Hot areas caused by the engine exhaust are marked in blue

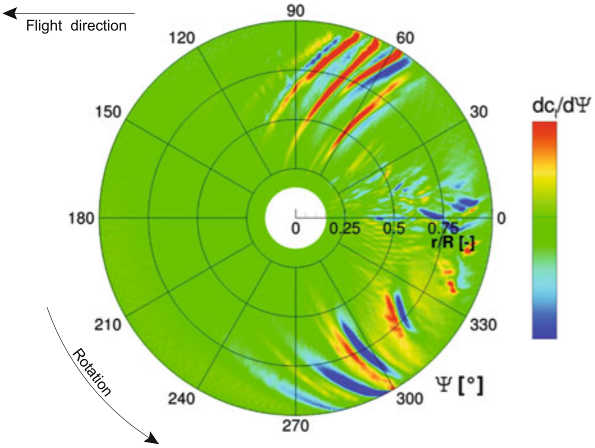

Fig. 8 Azimuthal (time) derivative of the sectional lift coefficient ($c_l/d\Psi$)

especially of the skids are visible. The area of high temperatures are marked in blue, which result from the engine exhaust. The detailed preservation of the wake ensures a realistic behavior of the Fenestron tail rotor operation in terms of provided thrust and therefore required control angles.

One of the primary drivers causing aeroacoustic impulsive noise emission by the rotor blade is the time rate of load change. Therefore, the azimuth angle was taken as the independent variable and the azimuthal derivative was formed. Figure 8 shows this azimuthal derivative of the sectional lift coefficient c_l in polar coordinates for one rotor revolution. The BVI events are clearly visible on the advancing blade side as well as on the retreating blade side in terms of high lift fluctuations. In case of the advancing side, four strong interaction events are seen in the area between Ψ=40° and 90°. All interactions show a strong gradient with short interactions periods. Taking the timing of the events into account, the first and third event show a nearly coherent interaction over the blade span at the same time. These are first indications of events with high noise emission due to a coherent interference of the caused pressure disturbances.

Focusing on the retreating side, high load fluctuations are found in the azimuth range between Ψ=270° and 330°, mostly symmetric to the advancing side. The interaction events show a stretched temporal influence compared to the advancing side, where the events are limited to short periods. A higher variation of the level of influence between BVI events is present compared to the advancing side, where all events have comparable magnitudes.

A more detailed investigation of a previous isolated rotor simulation is found in [2]. The analysis of the aeroacoustic noise emission of the full helicopter configuration is published in [20].

5 Conclusion

Generalized, high performance mesh deformation and load evaluation libraries have been developed and validated, that are suitable for structured, unstructured and overlapping meshes. A major achievement is the flexibility, re-usability and performance in highly distributed environments. Validation results show excellent agreement with former implementations. Further applications, e.g. deforming ground boundary layers for simulating in-ground-effect forward flights above rough terrain, or for analyzing ground roughness influence to wind turbines, and deformation of slotted and elastic flaps and slats, are in development.

The scaling investigation shows the high efficiency of the current FLOWer version operating on the HLRS Hornet system. These recent upgrades of the code's architecture in terms of MPI process data exchange allow the fulfillment of all demands of todays' high performance computing clusters in terms of parallelization. From our point of view, currently no additional updates are required to increase the code performance.

Furthermore, the present paper shows significant advantages using the higher order WENO reconstruction method newly implemented by IAG into the CFD code FLOWer. Especially in combination with the HLLC Riemann solver a significantly higher vortex conservation is achieved. Even if a higher computational effort is required, a computation with as much as 70 % less grid cells compared to a $2nd$ order scheme still shows a more precise solution. The presented application to a helicopter forward flight situation shows obvious benefits of the higher accuracy of the fifth order scheme. Remarkable results are achieved in the computation of the acoustic noise emission being the outcome of a more detailed vortex topology and a good preservation of vortex strength.

Acknowledgements We greatly acknowledge the provision of supercomputing time and technical support by the High Performance Computing Center Stuttgart (HLRS) for our project HELISIM.

References

1. Kutz, B.M., Günther, T., Rumpf, A., Kuhn, A.: Numerical examination of a model rotor in brownout conditions. In: Proceedings of the American Helicopter Society, 70th Annual Forum, Montreal, May 2014
2. Kowarsch, U., Lippert, D., Schneider, S., Keßler, M., Krämer, E.: Aeroacoustic Simulation of an EC145T2 Rotor in descent flight. In: 71th American Helicopter Society Annual Forum, Virginia, May 2015
3. Kroll, N., Eisfeld, B., and Bleeke, H.: The Navier-Stokes code FLOWer. Notes Numer. Fluid Mech. **71**, 58–71 (1999)
4. Wilcox, D.: Multiscale model for turbulent flows. AIAA J. **26**(11), 1311–1320 (1988)
5. Jameson, A.: Time dependent calculations using multigrid, with applications to unsteady flows past airfoils and wings. In: AIAA 10th Computational Fluid Dynamics Conference, Honolulu (1991)

6. Kowarsch, U., Oehrle, C., Hollands, M., Keßler, M., Krämer, E.: Computation of helicopter phenomena using a higher order method. In: High Performance Computing in Science and Engineering '13. pp. 423–438. Springer, Berlin (2013)
7. Jiang, G.-S., Shu, C.-W.: Efficient Implementation of Weighted ENO Schemes. J. Comput. Phys. **126**, 202–228 (1996)
8. Borges, R., Carmona, M., Costa, B., Don, W.S.: An improved weighted essentially non-oscillatory scheme for hyperbolic conservation laws. J. Comput. Phys. **227**(6), 3191–3211 (2008)
9. Ghosh, D., Baeder, J.: Weighted non-linear compact schemes for the direct numerical simulation of compressible, turbulent flows. J. Sci. Comput. **61**(1), 61–89 (2014)
10. Toro, E.F.: Riemann Solvers and Numerical Methods for Fluid Dynamics. Springer, Berlin (1997)
11. Johnson, W.: CAMRAD II Comprehensive analytical model of rotorcraft aerodynamics and dynamics, 4th edn. Johnson Aeronautics, Palo Alto (2009)
12. Dietz, M.: Simulation der Umströmung von Hubschrauberkonfigurationen unter Berücksichtigung von Strömungs-Struktur-Kopplung und Trimmung, Ph.D. thesis, Institut für Aerodynamik und Gasdynamik, Universität Stuttgart (2009)
13. Wurst, M., Keßler, M., Krämer, E.: Detached eddy simulation using the discontinuous galerkin method. In: Dillmann, A., Heller, G., Krämer, E., Kreplin, H.-P., Nitsche, W., Rist, U. (eds.) New Results in Numerical and Experimental Fluid Mechanics IX. Notes on Numerical Fluid Mechanics and Multidisciplinary Design, vol. 124, pp. 435–442. Springer International Publishing, New York (2014)
14. Schuff, M., Kranzinger, P.P., Keßler, M., Krämer, E.: Advanced CFD-CSD Coupling: Generalized, High Performant, Radial Basis Function Based Volume Mesh Deformation Algorithm for Structured, Unstructured and Overlapping Meshes, 40th European Rotorcraft Forum, Southampton (2014)
15. Buhmann, M.D.: Radial Basis Functions: Theory and Implementations, 8th edn. Cambridge University Press, Cambridge (2008)
16. Beckert, A., Wendland, H.: Multivariate interpolation for fluid-structure-interaction problems using radial basis functions. Aerosp. Sci. Technol. **5**(2), 125–134 (2001)
17. De Boer, A., Van der Schoot, M., and Bijl, H.: Mesh deformation based on radial basis function interpolation. Comput. Struct. **85**(11), 784–795 (2007)
18. Kranzinger, P.P., Hollands, M., Keßler, M., Wagner, S., Krämer, E.: Generation and Verification of Meshes Used in Automated Process Chains to Optimize Rotor Blades, Paper AIAA-2012-1260, 50th AIAA Aerospace Sciences Conference, Nashville, TN (2012)
19. Embacher, M., Keßler, M., Dietz, M., and Krämer, E.: Coupled CFD-simulation of a helicopter in free flight trim. In: Proceedings of the American Helicopter Society 66th Annual Forum, Phoenix, 11–13 May 2010
20. Kowarsch, C., U. Öhrle, Schneider, S., Keßler, M., Krämer, E.: Aeroacoustic Simulation of a Complete EC145T2 Helicopter in Descent Flight, 41th European Rotorcraft Forum, Munich (2015)