This article is the final submission, post-review, version of the following article:

Gagnon, L., Quaranta, G., and Schwaiger, M., « Open Source 3D CFD of a Quadrotor Cyclogyro Aircraft », Selected papers of the 11th Workshop, Edited by Nóbrega J., Jasak H., 2019, https://link.springer.com/chapter/10.1007/978-3-319-60846-4_27

Open Source 3D CFD of a Quadrotor Cyclogyro Aircraft

Louis Gagnon, Giuseppe Quaranta, and Meinhard Schwaiger

Abstract This chapter provides a detailed method for building an unsteady 3D CFD model with multiple embedded and adjacent rotating geometries. This is done relying solely on open source software from the OpenFOAM package. An emphasis is placed on interface meshing and domain decomposition for parallel solutions. The purpose of the model is the aerodynamic analysis of a quadrotor cyclogyro. The challenging features of this aircraft consist of a series of pairwise counter-rotating rotors, each consisting of blades that oscillate by roughly 90° about their own pivot point. The task is complicated by the presence of solid features in the vicinity of the rotating parts. Adequate mesh tuning is required to properly decompose the domain, which has two levels of sliding interfaces. The favored decomposition methods are either to simply divide the domain along the vertical and longitudinal axes or to manually create sets of cell faces that are designated to be held in a single processor domain. The model is validated with wind tunnel data from a past and finished project, for a series of flight velocities. It agrees with the experiment in regard to the magnitude of vertical forces, but only in regard to the trend for longitudinal forces. Comparison of past wind tunnel video footage and CFD field snapshots validates the features of the flow. The model uses the laminar Euler equations and gives a nearly linear speedup on up to 4 processors, requiring one day to attain periodic stability.

Louis Gagnon
DAER, Politecnico di Milano, Italy, e-mail: louis.gagnon@polimi.it

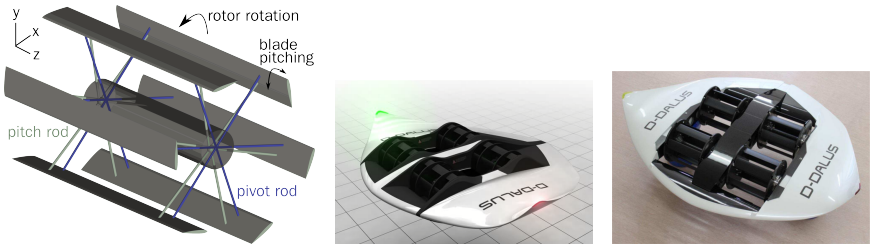
Giuseppe Quaranta
DAER, Politecnico di Milano, Italy, e-mail: giuseppe.quaranta@polimi.it

Meinhard Schwaiger
Innovative Aeronautics Technologies GmbH, Austria, e-mail: amxgmbh@gmail.com

1 Background

The cyclogyro is an aircraft that uses cycloidal rotors to generate propulsion. Cycloidal rotors are still largely unexplored by the aeronautic world. As opposed to conventional propellers, they produce forces that can change direction almost instantly on a 360° plane. Various studies have relied on these rotors to propel aircraft [26, 29, 23], micro-aircraft [2, 3, 4, 24] and airships [17, 16, 30, 20]. They are also used commercially to propel boats such as water tractors [1] and drillships [22]. Furthermore, they have been studied for wind [7, 19, 8, 25] and water [19, 25] turbines.

A cycloidal rotor, as illustrated in Fig. 1a, is an arrangement of symmetric blades of constant cross-section that rotate about a central drum. That drum transmits the spinning motion to the blades through a series of pivot rods. Each blade pitches individually about its intersection point with its pivot rod. The blade pitching motion is transmitted through the pitch rods, which are themselves offset by a central mechanism within the drum. Consequently, the total thrust generated by the rotor is the sum of individual blade lift and drag forces. The D-Dalus, shown in Fig. 1b, is a four rotor cyclogyro aircraft prototype developed by IAT21 [27, 28]. It relies solely on cycloidal rotors for thrust generation and is the object of this study. The actual aircraft prototype is shown in Fig. 1c. The rotor blades have 6 cm chords, while



(a) Cycloidal rotor sketch. (b) Aicraft concept render. (c) Actual aircraft.

Fig. 1: Rotor and aircraft for which the CFD model is created.

their span and the rotor diameter are both 24 cm. The pivot rods are attached to the blades slightly in front of the chord midpoint and allow pitching from -37° to 35°. The endplates have a 1 cm thickness and a 29 cm diameter.

The main purpose of the developed 3D CFD model is to observe the aerodynamic rotor-aircraft-rotor interaction. A better understanding of flow interaction arises [13] from the use of this model and a more informed aircraft design process can be conducted. CFD models for this type of aircraft have not been published before.

2 CFD Model

The CFD model is tridimensional and uses the finite volume method to solve the PIMPLE algorithm, which consists of a merger of the PISO and SIMPLE algorithms. In OpenFOAM 2.4.x, which is the version used for this project, this is achieved by using the *pimpleDyMFoam* solver. One pressure correction step is used and the pressure-momentum coupling is calculated twice. A bounded first order implicit discretization scheme is used on the time derivative. A Gaussian integration with linear interpolation is used for the derivative terms of pressure and velocity, with bounding on velocity. A second order upwind interpolation scheme is used for the advection of velocity. Linear interpolation is used for the Laplacian terms, with an under-relaxed face gradient corrected for mesh non-orthogonality. The convergence tolerance on the residual is 10^{-6} for both pressure and velocity. Prior to the end of the iteration loop, fields are also considered converged if the pressure and velocity residuals become 1% and 10% of their initial residuals, respectively. Although the solver is designed for Navier-Stokes equations, the Euler laminar equations are instead solved by setting the viscosity to zero. Air density is 1.204 kg/m^3. The main motivator for ignoring the effects of viscosity is to reduce the required computer time. The omission of viscosity is justified because the dynamics of rotors are dominated by pressure contributions and dynamic effects. This is also demonstrated by a study [21] that showed marginally small differences between experimental, Euler, and Navier-Stokes results for a helicopter rotor. A total of 14 moving meshes use sliding interfaces of interpolation. They are solved by the Arbitrary Mesh Interface (AMI) algorithm [9] and are shown in Fig. 2. Each rotor blade is inserted into a dou-

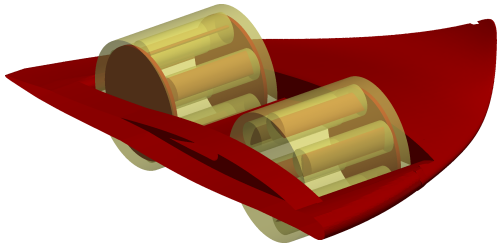


Fig. 2 The 14 AMI cylinders used for the cyclogyro aircraft, of which 12 are embedded.

ble AMI. The outer AMIs rotate and the inner ones strongly oscillate. The model relies on an embedded moving mesh algorithm [14] and an accompanying moving wall slip boundary condition [15] that were previously created [10] and publicly released. The embedded moving mesh is based on a regular oscillating mesh method, called the *oscillatingRotatingMotion* class in OpenFOAM. It incorporates a new origin vector $\mathbf{o_c}$,

$$\mathbf{o_c} = \mathbf{o_o} + r_o\{\text{sign}(\omega_o)\cos(|\omega_o|t + \phi_o\pi), \sin(|\omega_o t| + \phi_o\pi), 0\}, \tag{1}$$

where $\mathbf{o_o}$, r_o, ω_o, and ϕ_o are the outer AMI's origin, radius, angular velocity, and initial offset, respectively, and t is the time. The $\mathbf{o_c}$ vector is applied as the new origin of the transformation septernion.[1] The moving wall slip is based on a moving wall boundary condition, called the *movingWallVelocityFvPatchVectorField* class in OpenFOAM, and imposes the normal velocity vectors of the field as

$$\mathbf{n}\left(\mathbf{n}\cdot\left(\mathbf{u_p}+\mathbf{n}(u_n-(\mathbf{n}\cdot\mathbf{u_p}))\right)\right), \tag{2}$$

where $\mathbf{u_p}$ is the wall velocity, $\mathbf{n}$ is the unit normal to the wall and u_n is the wall mesh flux per area. The tangent velocity is taken as the planar vector component of the velocity adjacent to the wall.

The single rotor meshes with and without endplates have roughly 1 million and 300,000 cells, respectively. The endplates make the flow more two-dimensional, both in experiment and simulation. A first harmonic sinusoidal pitching schedule is imposed on the blades of the rotor. The blade angle, θ, with respect to a line perpendicular to the pivot rod, is

$$\theta = \theta_o + \theta_s \sin(\omega t + \phi), \tag{3}$$

where θ_o is the fixed pitch angle offset, θ_s is the magnitude of the pitch angle oscillations, ϕ is the imposed phase angle, and ω is the constant rotor angular velocity. The purpose of θ_o is to increase the pitch angle on the bottom part of the rotation cycle to counter the stronger inflow. The position of maximum pitch is anticipated by ϕ with respect to the bottommost angular position in order to counter the aerodynamic delay.

The aircraft is fixed in space, and thus the model disregards the inertial effects of gravity and aerodynamic forces. Careful tuning of the mesh interfaces allows us to keep the actual geometry of the aircraft. The only change is that the spanwise distance between the endplates and the rotor blades is slightly increased, to roughly one tenth of the chord length. A gap of this size has the same effect as if the endplate were attached to the foil [5], and is thus negligible. The space available in the physical aircraft between the rotor blades' pivot points and the airframe allows us to have an AMI cylinder radius at least equal to the maximum distance between the pivot point and any edge of the blade. The blades can thus pitch at any angle.

2.1 Mesh Generation

The *snappyHexMesh* hexahedral meshing tool is used to generate the mesh. A description of this mesher is given to introduce concepts that clarify the generation of embedded AMI interfaces in a very narrow space. The mesher inserts imported CAD geometries into a structured volume mesh. It then refines the volume mesh

[1] The septernion is a seven component array used in OpenFOAM composed of a translation vector and a rotation quaternion

near and on surfaces; in portions of surfaces that are close to other surfaces; and inside user-defined regions. Refinement is applied as a user-specified number of subdivisions to the original structured mesh. Cells are refined either inside or within a specified distance of a given region or when intersecting a given surface. Once the mesh is refined, the cell faces are moved so as to smoothly adhere to the boundaries, which can be wall boundaries or simple reference geometries. This last option allows us to create the sliding interfaces of the blade oscillating zones and the rotor spinning zones. The size of the mesh and the time it takes to generate are controlled by quality and iteration options. While this chapter focuses on the aspects critical for the cycloidal rotor aircraft application, a detailed guide to the mesher is available online [18].

2.2 Isolated Airframe Mesh

Before initiating the actual modeling of the cyclogyro, a separate mesh quality evaluation campaign is conducted for the airframe taken alone without the rotors. The impact of mesh refinement on the airframe alone is studied in order to obtain the smallest possible grid while having a mostly mesh-independent solution with low discretization error. Eight different meshes are generated and evaluated. The tests are all done at a 15° airframe angle of attack and a 30 m/s flow velocity. Different meshing techniques are also studied and the influence of different mesh parameters on the force results is examined. These parameters are the value of the surface feature angles that trigger mesh refinement, the refinement level of the mesh in the wake zone, and the increase of the overall mesh density. Table 1 shows the mesh attempts, along with the parameters of interest and the number of cells, which go from 293,000 to 1.7 million. The first three cases, *baseline*, *halfSize*, and *thirdSize*,

Table 1: Airframe mesh refinement tests.

Test	Surface	Distance	Wake zone	kCells	100Fx	Fy	T	M
baseline	5 8	6 3 1	no	293	-3.93	3.31	3.31	-1.47
halfSize	5 8	6 3 1	no	747	-6.22	3.15	3.15	-1.36
thirdSize	5 8	6 3 1	no	1729	5.50	2.75	2.75	-1.20
angles[a]	5 9	5 3 1	no	507	-4.53	3.22	3.22	-1.39
surf	6 9	6 3 1	no	524	-4.80	3.26	3.26	-1.40
surf2	7 8	6 3 1	no	423	-1.98	3.19	3.19	-1.41
wake	4 8	5[b]	yes[c]	507	-5.61	3.12	3.12	-1.40
noWake	4 8	5[b]	no	492	-4.56	3.14	3.14	-1.41

[a] attempt at changing the *featureAngle* value (surface feature angles that trigger refinement)
[b] at a 25 cm distance
[c] 3 levels of refinement inside and 2 levels within 1 m of the wake zone

are meshes created with identical parameters. Their only difference is that the ini-

tial structured volume mesh cells of *halfSize* and *thirdSize* are 1/2 and 1/3 the sizes of those of *baseline*, respectively. The table also shows the normalized mean longitudinal, vertical, thrust, and moment forces obtained for each mesh over a period equivalent to one rotation at 3750 rpm. One period takes 1,300 to 7,800 timesteps, depending on the refinement level of the mesh. The numbers in the surface column of Table 1 are the minimum and maximum number of divisions to apply to the structured mesh cells that encounter the airframe surface. The numbers in the *Distance* column are the respective number of divisions to apply to the cells that are located at 1 cm, 20 cm, and 50 cm from the airframe.

The forces obtained using any of these meshes reach a fairly constant value after 10 periods. At this point, the longitudinal and vertical forces oscillate by less than 2% and 9% of the thrust, respectively. These oscillations are caused by the vortex shedding that occurs on the airframe at a 15° angle of attack. The magnitudes of these oscillations are not linked to the refinement level of the mesh, but the most refined mesh does take longer to stabilize. Table 1 also indicates that for the case studied, the level of refinement on the airframe influences the thrust by 8% of its maximum value. That value drops to 6% when the average thrust is measured over more periods. Finally, the coarsest case is run for 3 seconds, which is equivalent to 200 periods, in order to see the long term tendency of the flow. It is shown that the average forces remain almost constant over time, with an oscillation in the mean lift generated of roughly 1%.

2.3 Rotor Model

The rotor model is initiated by enhancing a simulation from a previous project [11, 12], which had been validated against experimental data [30] for a larger rotor without endplates. That prior CFD simulation had been shown to yield more accurate quantitative results than its analogous 2D version. It had also shown that the size and velocity of the inlet and outlet boundary conditions have little influence on the rotor flow features and forces. The difference between simulation and experiment was below 20% for the power, with a much better agreement at a low pitch angle and at low angular velocities. For thrust, the errors were contained at low angular velocities, but reached problematic magnitudes at higher angular velocities. That existing simulation had been used for proof of concept simulations and had not been tested for stability. It is thus reconfigured to match the new geometry, which is roughly proportional, 3 times smaller, and has different pivot points for the blades. The mesh is tweaked to allow for locating an oscillating sliding interface between the rotor and its blades. Mesh tweaking also ensures validity over a range of rotor angular velocities, which reach roughly 7 times the maximum angular velocity of the previous model. The presence of endplates considerably increases the model's complexity. This is due to the very small space between the oscillating blades, the rotating endplates, and, eventually, the airframe. The spacing between the blades and the endplates is only 3% of the blade span. Thus, the mesher is forced to move faces

from cells lying in a very narrow zone and make them adhere to the airframe, to the rotor and blade interfaces, and to the blades. This zone therefore requires a carefully refined mesh. The mesh is completely parametrized to allow automatic generation for different geometries and to have a stable and repeatable meshing procedure.

The thrusts obtained for the rotor alone match up in order of magnitude with the experimental data from the same rotor installed on a fixed apparatus in calm air. This confirms that the model is properly set up and the preliminary model development is deemed complete. Fig. 3 shows qualitative results from that preliminary single rotor model. The simulation is set up with a rotational velocity of 3970 RPM, a null

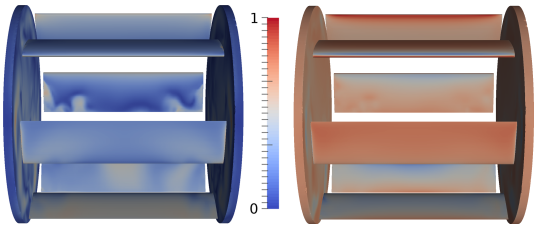


Fig. 3: Visualization of the surface pressure and velocity on the preliminary rotor simulation.

incoming wind velocity, and a mesh size of 1.3 million cells.

2.4 Entire Aircraft Mesh

The rotor model is combined with a second rotor and the half D-Dalus L1 airframe to create the full aircraft model, using a symmetry about the central plane. The mesh separating the various AMIs, the endplates, and the airframe is very delicate, and thus several iterations of the parametrized mesh generation are undertaken. The important parameters for generating a cyclogyro mesh are described in Table 2.

Table 2: Mesh parameters for *snappyHexMesh* and brief explanations.

Parameter	Value	Explanation
addLayers	false	layers are not needed in a non-viscous simulation
maxGlobalCells	8×10^6	this number limits the mesh size when the level refinement required would otherwise surpass maxGlobalCells
faceType	baffle	this creates duplicate patches at the AMIs
implicitFeatureSnap	true	uses the implicit method for finding refinement surfaces
detectNearSurfacesSnap	true	prevents cell faces from adhering to a nearby surface by mistake

A proof of concept model is then created with a preliminary mesh. Its purpose is to develop into a working model that converges for the most unstable flow condition before refining the mesh until satisfactory validation results are reached. It represents the most unstable case that can be expected to be encountered during the simulations and its purpose is to verify the robustness of the model. The airframe angle of attack is 15° and the horizontal incoming wind velocity is 30 m/s. The mesh has 2.7 million cells and is shown in Fig. 4. The boundary conditions are null

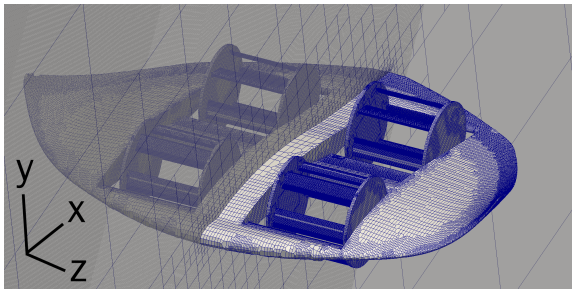


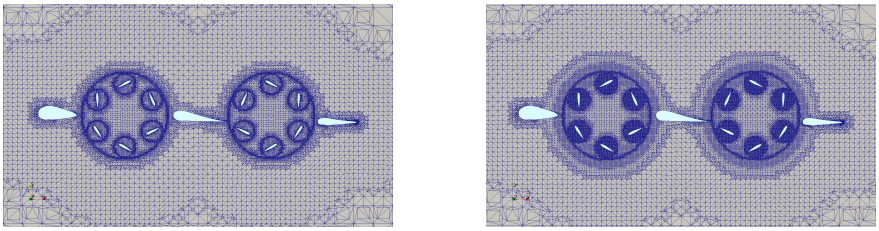
Fig. 4 Mesh of the proof of concept simulation shown along with its symmetry plane.

normal gradient for pressure and fixed velocity at the inlet. At the outlet, they are ambient pressure and null normal velocity gradient, which becomes a null velocity when the flow attempts to re-enter the domain. The latter velocity condition is referred to as *inletOutlet* in OpenFOAM jargon. On the outside walls and on the aircraft body, a slip velocity and null normal pressure gradient are used. Finally, on the rotor blades and endplates, the conditions are null normal pressure gradient and the developed moving slip velocity condition. The modeled flow domain is 5.3, 20.5, and 4.3 times the half-aircraft's corresponding lengths in the longitudinal, vertical, and span directions, respectively.

2.5 Final Mesh Tuning

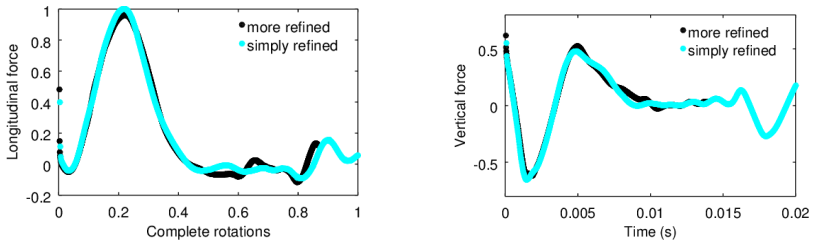
Once the full aircraft model is ready, a final mesh refinement is performed. The *simply refined* and the *more refined* meshes are created with 3.7 million and 5.7 million cells, respectively.

The *more refined* mesh has a greater refinement zone around the rotors and a finer grid within each inner AMI cylinder, as shown in Fig. 5. It also has a wider wake zone that extends up to the front of the aircraft to fit with both hover and forward flight conditions. Nonetheless, both meshes yield very similar force results right from the start of the simulation. That match between both cases is shown for a foil of the rear rotor, being the most perturbed rotor, in Fig. 6. There are still small differences in values which indicates that a completely mesh-independent solution has not been fully reached. Nevertheless, the smaller, less refined mesh is kept, because both solutions are very close. This avoids doubling the solution time, as



(a) Simply refined. (b) More refined.

Fig. 5: Comparison between the simply and more refined meshes.



(a) Drag. (b) Lift.

Fig. 6: Comparison of drag and lift on one foil of the rear rotor for both refinement levels over the first simulation cycle.

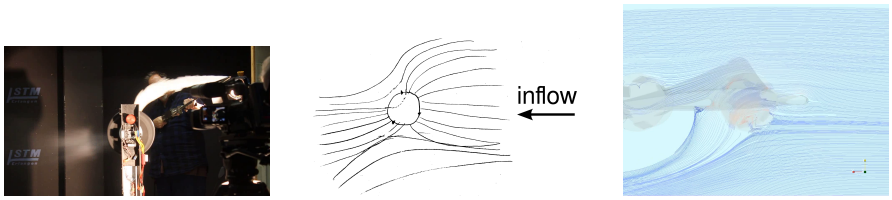
required by the *more refined* case, and allows us to run the number of required analyses within the fixed project timeframe. This constraint is further reinforced by the limitations of the sliding interface domain decomposition, which is covered in Section 3. However, widening the wake zone has very little time cost and only adds a small number of cells. Thus, a final mesh relying on refinement far from wall and consisting of a slight improvement of the *simply refined* case is used for the definitive model. It has 4.5 million cells, and the solution is periodically stable after 6 rotations, because the rotors have a dominant effect on the flow and cause stability to be reached faster than for the airframe alone.

2.6 Validation

The main challenge of the validation is that no wind tunnel data had been gathered while both the front and rear rotors were in use. The experimental operation having been completed and resigned to the past, no more data can be obtained. It follows that the experimental data available is for the quadrotor cyclogyro propelled by the two front rotors alone. No velocity information is available for the rear rotor.

The values of drag and lift obtained by CFD are nevertheless compared to the experimental data from the wind tunnel. The highest wind tunnel velocity is chosen

as a basis for validation and to approximate the angular velocity for the unpowered rear rotors during the wind tunnel tests. Attempts with various rear rotor angular velocities lead to the conclusion that the fixed rear rotor most adequately reproduces the experiment. Rotor flow visualization from past experiments matches the CFD streamlines of the powered rotor, as shown in Fig. 7. The trend of thrust matches



(a) Snapshot from wind tunnel. (b) Wind tunnel streamlines. (c) Powered rotor.

Fig. 7: Streamlines of the rotor in wind tunnel compared to the powered aircraft rotor.

that of the wind tunnel at 10 m/s, 15 m/s, 20 m/s, and 25 m/s, as shown in Fig. 8. The agreement in vertical forces is the main objective of the project, and this justifies

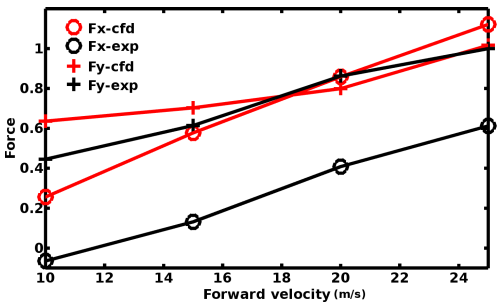


Fig. 8 Trend match between CFD (red) and wind tunnel (black).

neglecting the contribution of the viscous forces on the airframe. The remainder of the validation process is reported in the article that focuses on the aerodynamics of the aircraft [13].

3 Domain-decomposition Parallelization

This section presents the method developed in order to fully solve the aircraft in parallel. This starts with parallel meshing, which is followed by a parallel CFD solution, and finally by a parallel visualization. This last one does not require any tuning

and is done using a recent version of ParaView. The whole process is achieved locally on a 12 core machine. For the solution phase, the most efficient parallelization strategy is to divide the domain along the vertical and longitudinal axes, leaving an equal number of cells on each side. This decomposition method is called the *simple* method in *snappyHexMesh* jargon. That method reduces the communication across processors for sliding AMI interface pairs to a minimum. Fig. 9a shows the 4 processor submeshes obtained with the *simple* decomposition algorithm. Solving this case in parallel takes roughly one day, instead of 4 days, to reach a periodically stable solution.

The part of the process that benefits the most from default parallelization is the meshing process of *snappyHexMesh*, for which the method can be found in the OpenFOAM tutorials. Diversely, using the default options for AMI interface decomposition with the Scotch [6] algorithm, the solving phase of the simulation has an increase in speed that ranges between 45% and 95% on two processors and 100% on 10 processors. An equivalent simulation without the AMIs yields a 350% speedup on 10 processors. The cause is that the AMIs are distributed over different processors, and thus communication is slowed down. This decomposition of the AMIs can be seen for a 10 processor mesh in Fig. 9b, where each color represents one processor domain. Coincident sliding interface boundaries, referred to as the master and

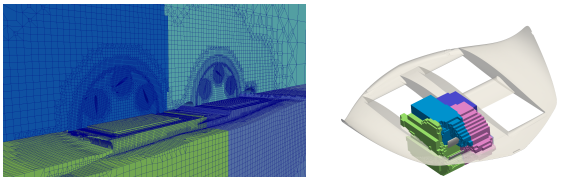


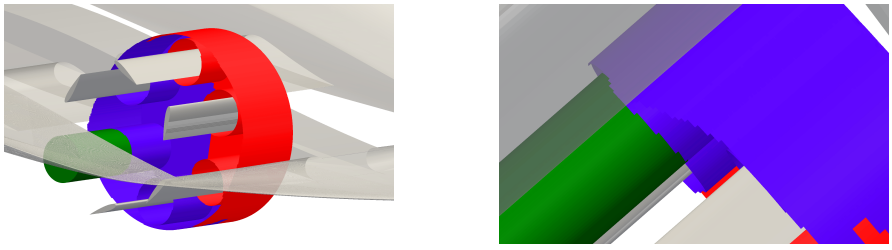
Fig. 9 Decomposition methods with distinct processor colors.

(a) 4 processors *simple* decomposition.

(b) Part of a 10 processor mesh.

slave AMI patches in OpenFOAM, maintain their matching cell faces on the same processor, but the patches themselves are split into two or more portions. This is visible in Fig. 10a and in the close-up in Fig. 10b. The interface irregularities force the AMI cells of one processor to communicate with those of another processor as soon as they start rotating. The extra communication step between processors at the AMI slows down the simulation.

Thus, the goal is to maintain the whole AMI, with its master and slave patches, on a single processor. A summary of the available methods and their observed behaviors is given in Table 3. The Scotch method that uses these options does not automatically yield an efficient AMI decomposition. Also, if not carefully controlled, the decomposition creates more than one cell block for a single processor in an attempt to respect the given interface constraints. The resulting mesh thus has an increased computation time, due to each processor zone being split over parts of the domain that are not physically connected. An example of such a decomposition is shown in Fig. 11, where the blue surface is the AMI and the pale gray zones represent the submesh of a single processor. In that case, although the AMIs are preserved on



(a) Rotor view. (b) Close-up on AMI.

Fig. 10: AMI cells distribution over different processors.

Table 3: Available options to preserve mesh zones on a specific processor.

Method	Description	Result
preserveFaceZones	Preserves face zones on a single processor	Is not effective in doing so
preservePatches	Ensures the the patches[a] given are meshed on a single processor	The coincident sliding interface boundaries are, however, not always meshed on the same processor
singleProcessorFaceSets	Ensures that the given face set is meshed on a single processor	Using a trick, it is possible to define a whole volume as a face set, and thus obtain that the meshing algorithm maintains that volume on a single processor

[a] A patch in OpenFOAM consists of a wall, an interface, or any continuous set of cell faces that represents a surface

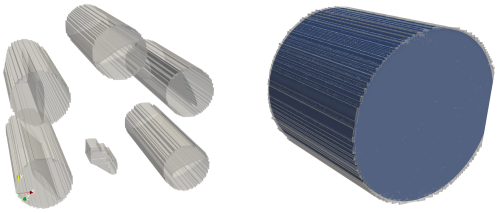


Fig. 11 Decomposed mesh showing a processor divided into multiple zones; the processor zone is in pale gray and the outer AMI interface in blue.

a single processor, the number of different blocks for one processor cause latency. The *simple* decomposition method, with a domain division in two boxes, remains the most effective one in preserving the whole AMIs of the front and rear rotors on two different processors. Using the *simple* method with four processors, 12 of the 14 sliding interfaces are preserved on a single processor, the yields of which are a nearly linear speedup.

When the *simple* algorithm is no longer efficient due to a large number of processors, the manual creation of sets of cell faces that are designated to be held in a single processor domain can be done. This is called the *singleProcessorFaceSets* method in *snappyHexMesh* jargon. It can be explicitly defined for each processor

and for each set. This Scotch method option allows for preserving all the faces of a rotor's AMIs on a single processor. The *singleProcessorFaceSets* method may be applied using the *topoSet* tool to create a set of cell faces, *faceSet*, from a set of cells, *cellSet*, using the *cellToFace* option. After running *topoSet* with these indications, decomposition must be run with the *singleProcessorFaceSets* option. Including the AMIs in the *faceSets* to be assigned to a single processor can, however, confuse the algorithm and create an unbalanced mesh. By considering zones as small as one inner AMI cylinder as single processor zones, meshes with more than 13 processors and a reasonable speedup can be obtained. In that case, an annulus processor zone may be created to ensure that the outer rotor AMIs are also meshed on a single processor. However, the effects of divided outer AMIs on the parallel speedup are less harmful than those of a multitude of divided inner embedded AMIs.

A considerable amount of time is required to implement the *singleProcessorFaceSets* method, thus the favored method remains the use the *simple* division with two or four processors. The retained procedure for running the case is to first run the mesher on any desired number of processors, then reconstruct the case as a single-processor mesh, and finally redivide it into 4 processors using the *simple* method.

4 Closing Remarks

This chapter presented a methodology for modeling rotating and strongly oscillating components of a rotor using open source CFD software. These rotor components can be embedded one inside another, and parallelization is fairly straightforward up to 4 processors through the simple division of the domain along the vertical and longitudinal axes. This is called the *simple* method. A more efficient and refined parallelization could be obtained by manually creating sets of cell faces that are designated to be held in a single processor. This is called the *singleProcessorFaceSets* decomposition method and can eventually be parametrized to allow for large scale parallel solutions. However, such a process requires a significant set-up time that may be rewarding only if a large number of analyses is foreseen. Special care is necessary when generating the mesh near the rotating interfaces and when choosing the decomposition methods. A study of the impact of refinement on the airframe was conducted to grasp the impacts of surface- and region-based refinement levels. The final mesh is small enough to allow the CFD to be solved in one day on four processors, yet refined enough to grasp the important flow features and forces. The best meshes were generated by allowing large cells on nearly flat surfaces and refining near the sliding interfaces. The CFD was done using the laminar Euler equations of the *pimpleDyMFoam* solver. A brief validation section, based on prior experiments both on the rotor alone and on the full aircraft inside a wind tunnel, was presented. The methods from this article, in combination with the available OpenFOAM tutorials, can be used as a starting point for modeling similar rotating machines.

Acknowledgements The research presented in this paper was supported by the Austrian Research Promotion Agency (FFG) *Basis programe* research grant #849514: *Entwicklung des Fluggertes D-Dalus L2 als eigenstabil flugfähigen Prototypen*.

References

[1] Barrass C (2004) Chapter 22 - Improvements in propeller performance. In: Barrass C (ed) Ship Design and Performance for Masters and Mates, Butterworth-Heinemann, Oxford, pp 218–227, DOI 10.1016/B978-075066000-6/50024-6, URL http://www.sciencedirect.com/science/article/pii/B9780750660006500246

[2] Benedict M, Ramasamy M, Chopra I, Leishman JG (2009) Experiments on the Optimization of MAV-Scale Cycloidal Rotor Characteristics Towards Improving Their Aerodynamic Performance. In: American Helicopter SocietyInternational Specialist Meeting on Unmanned Rotorcraft, Phoenix, Arizona

[3] Benedict M, Ramasamy M, Chopra I, Leishman JG (2010) Performance of a Cycloidal Rotor Concept for Micro Air Vehicle Applications. Journal of the American Helicopter Society 55(2):022,002–1–14, doi:10.4050/JAHS.55.022002

[4] Benedict M, Mattaboni M, Chopra I, Masarati P (2011) Aeroelastic Analysis of a Micro-Air-Vehicle-Scale Cycloidal Rotor. AIAA Journal 49(11):2430–2443, doi:10.2514/1.J050756

[5] Calderon DE, Cleaver D, Wang Z, Gursul I (2013) Wake Structure of Plunging Finite Wings. In: 43rd AIAA Fluid Dynamics Conference

[6] Chevalier C, Pellegrini F (2008) Pt-scotch: A tool for efficient parallel graph ordering. Parallel Computing 34(68):318 – 331, DOI https://doi.org/10.1016/j.parco.2007.12.001, URL http://www.sciencedirect.com/science/article/pii/S0167819107001342, parallel Matrix Algorithms and Applications

[7] Darrieus GJM (1931) Turbine having its rotating shaft transverse to the flow of the current. URL https://encrypted.google.com/patents/US1835018, uS Patent 1,835,018

[8] El-Samanoudy M, Ghorab AAE, Youssef SZ (2010) Effect of some design parameters on the performance of a Giromill vertical axis wind turbine. Ain Shams Engineering Journal 1(1):85–95

[9] Farrell P, Maddison J (2011) Conservative interpolation between volume meshes by local Galerkin projection. Computer Methods in Applied Mechanics and Engineering 200(1–4):89–100, DOI 10.1016/j.cma.2010.07.015, URL http://www.sciencedirect.com/science/article/pii/S0045782510002276

[10] Gagnon L, Morandini M, Quaranta G, Muscarello V, Bindolino G, Masarati P (2014) Cyclogyro Thrust Vectoring for Anti-Torque and Control of Helicopters. In: AHS 70th Annual Forum, Montréal, Canada

[11] Gagnon L, Quaranta G, Morandini M, Masarati P, Lanz M, Xisto CM, Páscoa JC (2014) Aerodynamic and Aeroelastic Analysis of a Cycloidal Rotor. In: AIAA Modeling and Simulation Conference, Atlanta, Georgia

[12] Gagnon L, Morandini M, Quaranta G, Muscarello V, Masarati P (2016) Aerodynamic models for cycloidal rotor analysis. J of Aircraft Engineering and Aerospace Technology 88(2):215–231

[13] Gagnon L, Quaranta G, Schwaiger M, Wills D (2017) Aerodynamic analysis of an unmanned cyclogiro aircraft. submitted to SAE Tech Papers

[14] Gagnon, L (2014) Interface within an interface c++ code. http://www.cfd-online.com/Forums/openfoam-solving/124586-dynamic-mesh-within-dynamic-mesh.html#post476869, last accessed Feb. 2016

[15] Gagnon, L (2014) Slip moving wall boundary condition c++ code. http://www.cfd-online.com/Forums/openfoam-solving/105274-free-slip-moving-wall-bc.html#post509989, last accessed Feb. 2016

[16] Gibbens R (2003) Improvements in Airship Control Using Vertical Axis Propellers. In: Proceedings of AIAA's 3rd Annual Aviation Technology, Integration, and Operations (ATIO) Forum, DOI 10.2514/6.2003-6853

[17] Gibbens R, Boschma J, Sullivan C (1999) Construction and testing of a new aircraft cycloidal propeller. In: Proceedings of 13th Lighter-Than-Air Systems Technology Conference., DOI 10.2514/6.1999-3906

[18] Greenshields, C (2016) OpenFOAM User Guide: 5.4 Mesh generation, snappyHexMesh. http://cfd.direct/openfoam/user-guide/snappyhexmesh/, last accessed Sept. 2016

[19] Hwang IS, Lee HY, Kim SJ (2009) Optimization of cycloidal water turbine and the performance improvement by individual blade control. Applied Energy 86(9):1532–1540

[20] Ilieva G, Páscoa JC, Dumas A, Trancossi M (2012) A critical review of propulsion concepts for modern airships. Central European Journal of Engineering 2(2):189–200, doi:10.2478/s13531-011-0070-1

[21] Kim JW, Park SH, Yu YH (2009) Euler and Navier-Stokes Simulations of Helicopter Rotor Blade in Forward Flight Using an Overlapped Grid Solver. In: 19th AIAA Computational Fluid Dynamics Conference Proceedings, paper AIAA 2009-4268

[22] Koschorrek P, Siebert C, Haghani A, Jeinsch T (2015) Dynamic Positioning with Active Roll Reduction using Voith Schneider Propeller. IFAC-PapersOnLine 48(16):178–183, DOI 10.1016/j.ifacol.2015.10.277, URL http://www.sciencedirect.com/science/article/pii/S2405896315021680, 10th {IFAC} Conference on Manoeuvring and Control of Marine Craft {MCMC} 2015Copenhagen, 24–26 August 2015

[23] Leger JA, Páscoa JC, Xisto CM (2016) Aerodynamic Optimization of Cyclorotors. Accepted by J of Aircraft Engineering and Aerospace Technology

[24] Lind A, Jarugumilli T, Benedict M, Lakshminarayan V, Jones A, Chopra I (2014) Flow field studies on a micro-air-vehicle-scale cycloidal rotor in forward flight. Experiments in Fluids 55(12):1826, DOI 10.1007/s00348-014-1826-1, URL http://dx.doi.org/10.1007/s00348-014-1826-1

[25] Maître T, Amet E, Pellone C (2013) Modeling of the flow in a Darrieus water turbine: Wall grid refinement analysis and comparison with experiments. Renewable Energy 51:497–512

[26] McNabb ML (2001) Development of a Cycloidal Propulsion Computer Model and Comparison with Experiment. Master's thesis, Mississippi State University

[27] Schwaiger M (2010) Aircraft: US Patent 7735773 B2. http://www.freepatentsonline.com/7735773.html, last accessed Feb. 2016

[28] Schwaiger M (2014) Aeroplane: US Patent USD709430 S1. http://patents.google.com/patent/USD709430S1, last accessed Feb. 2016

[29] Xisto CM, Páscoa JC, Leger JA, Masarati P, Quaranta G, Morandini M, Gagnon L, Wills D, Schwaiger M (2014) Numerical modelling of geometrical effects in the performance of a cycloidal rotor. In: 6th European Conference on Computational Fluid Dynamics, Barcelona, Spain

[30] Yun CY, Park IK, Lee HY, Jung JS, H IS (2007) Design of a New Unmanned Aerial Vehicle Cyclocopter. Journal of the American Helicopter Society 52(1)