European Rotorcraft Forum

LAGRANGIAN PARTICLE TRACKING IN DEFORMING SLIDING MESH FOR ROTORCRAFT ICING APPLICATIONS

M. Morelli[⋆,†], T. Bellosta[⋆], and A. Guardone[⋆]

[⋆]Politecnico di Milano - Dipartimento di Scienze e Tecnologie Aerospaziali
Campus Bovisa, Via La Masa 34, 20156 Milano, Italy
[†]Email address: mylescarlo.morelli@polimi.it

Abstract

Particle tracking techniques are presented for simulating the development of clouds containing super-cooled water droplets entrained in complex flow systems such as those produced by rotorcraft. The physics of this problem and the associated condition is known as in-flight icing, where extreme ice structures can possibly form, causing severe performance degradation and a reduction in handling qualities. Resolving rotorcraft flow fields frequently entails the use of multi-zone and deforming grid systems to allow independently moving components. The approach advocated here allows particle tracking through mesh with arbitrary motion which includes deformation and particle tracking through non-conformal interfaces which are present in multi-zone problems. These techniques are first described and verified for two-dimensional problems before the results of three-dimensional practical engineering applications are shown.

1. INTRODUCTION

Extensive studies to closely understand the phenomena of in-flight icing on aircraft have been in place for the past 40 years now[1], however, due to the high level of uncertainty and unpredictable nature of in-flight icing, the accuracy of current simulation capabilities is still far from being satisfactory[2]. Rotorcraft configurations represent some of the most challenging case studies due to their multiple moving parts and complex aerodynamic flowfields. In conjunction, typical operational flight paths of rotorcraft mean that avoiding known icing conditions is sometimes unfeasible. In this paper, computational modelling of in-flight icing requires first solving complex aerodynamic flowfields, then tracking super-cooled water droplets to determine their impingement locations, and ultimately computing the final ice shapes.

The numerical discretization of highly complex geometries such as rotorcraft often requires the generation of unstructured multi-zone or structured multi-block grids to ensure a high-quality computational domain. This allows the grid to be split into subregions so moving parts such as helicopter rotor blades can move or deform in relative motion to the fuselage. However, this subsequently results in subregions with non-matching interface boundaries where suitable treatment of the artificially introduced boundaries is pivotal.

One of the most widely used schemes for rotorcraft applications is the Chimera interpolation technique on overlapping grids. In its original formulation, the Chimera scheme is considered non-conservative[3] as it suffers from inaccuracy across interfaces during discontinuities such as shocks waves[4]. To account for this efforts have been made to subsequently improve the Chimera scheme over interface discontinuities[5]. With this recent development on the conservation property of Chimera it can now be considered a conservative scheme by replacing the overlapped zones with patched zones. Different implementations of the Chimera scheme can be found in many rotorcraft CFD codes such as the HMB solver from the University of Glasgow[6], the ROSITA solver from Politecnico di Milano[7], FLOWer from DLR[8], and TURNS from NASA Ames research center[9] as well as many more. Most of the codes are largely based on structured grid topologies.

While Chimera schemes are quite common in rotorcraft codes the contrary is true for schemes such as the sliding mesh technique[10]. A method implemented by Rinaldi et al.[11] presented a scheme for flux-conservation treatment of non-conformal interfaces using the concept of a supermesh which acts as a supplementary interface grid that enables sliding mesh connectivity between mesh interfaces. This sliding mesh technique using the supermesh concept has been implemented into the SU2 solver[12] which is an open-source suite for multi-physics simulation and design based on unstructured meshes. Different sliding mesh techniques have also been implemented in other rotocraft codes such as e.g. HMB[13] and a code from the Korea Advanced Institute of Science and Technology by Nam et al.[14].

The next crucial problem for in-flight icing simulations is the computation of the trajectories of the supercooled water droplets within the computational domain. Two possible frameworks can be adopted: the Eulerian frame of reference and the Lagrangian frame of reference. In the Eulerian approach, the particle phase is considered as a continua whose dynamics is described by a system series of partial differential equations. An advantage of this is that it allows the use of standard numerical tools already available for aerodynamics. In the Lagrangian approach, the particle phase is described using a discrete particle method and the balance of forces acting on each discrete particle is resolved. The Lagrangian approach is thus advantageous as it better resembles the particle physics. However, it soon becomes computationally expensive when modelling a large quantity of particles. Within this work, the Lagrangian method is used to deliver the following system of ordinary differential equations,

(1)
$$\begin{cases} \dfrac{d\mathbf{u}_p}{dt} = \dfrac{3}{4}\dfrac{\mu_f Re_p C_D}{\rho_p d_p^2}(\mathbf{u}_f - \mathbf{u}_p) + \mathbf{g}\left(1 - \dfrac{\rho_f}{\rho_p}\right) \\ \dfrac{d\mathbf{x}_p}{dt} = \mathbf{u}_p \end{cases}$$

where the subscripts p and f are representative of the particle and fluid. Only aerodynamic drag, gravity and particle inertia are considered. Whilst Re_p is the particle Reynolds number which describes the relative flow around the particle. The drag force acting on any given particle can then be computed using equation (2).

In order to solve the above equations the position of the particle relative to the mesh is needed. In the case of regular structured grids, the problem can be solved relatively easily, however, for applications where unstructured grids are required the design of efficient point-locating algorithms becomes a challenging task[15]. Many papers addressed in the literature have discussed this matter[16,17,18]. However, to the authors knowledge what is lacking in all of these algorithms are methods for the treatment of non-conformal boundary interfaces such as those present in sliding mesh and Chimera techniques, as well as, methods to account for deforming mesh. For complex rotorcraft configurations, commonly with multiple non-conformal interface boundaries and deforming mesh, Lagrangian particle tracking soon becomes problematic. With no interface treatment particles are unable to move across boundary interfaces from one zone to another, and, with no deforming mesh consideration, particles are easily lost.

Once the a priori stages are complete, icing computations can then finally take place, where, various different models can be used to compute the ice thickness. Two of the most commonly used ice accretion models are the Messinger and Myers models and are in icing codes such as LEWICE, ONICE3D, FENSAP-ICE and PoliMIce[19,20,21,22]. Although fixed-wing aircraft icing codes are in the mature stages of development, rotorcraft icing codes remain in the early stages of development and it is only recently that fully three-dimensional rotorcraft icing simulations have been conducted[23,24]. This is largely down to the complexity of the icing simulation methods and it is for this reason why further studies are required for more reliable icing simulations. Moreover, all of the latest rotorcraft icing simulations are based on Eulerian particle tracking methods and have yet to test the results against Lagrangian based particle tracking methods.

The purpose of this paper is to present new Lagrangian based particle tracking methods for 1) Arbitrary grid motion, such as, through deforming or rigidly moving grids; and 2) Non-conformal interfaces, such as, multi-zone sliding mesh. The method to account for the arbitrary mesh motion will first be discussed in Section 2. The method to account for the non-conformal interfaces will then be discussed in Section 3. In Section 4 three-dimensional tests will then be introduced representative of more realistic engineering problems. These test cases will include; a wing during pitching motion, an isolated rotor in hover, and a tilt-rotor configuration in aeroplane mode.

In this work the flow field will be computed using the SU2 solver[12]. The particle tracking will then be performed using an in-house particle tracking code[25].

$$(2)\quad C_D = \begin{cases} \dfrac{24}{Re_p} + 2.6\dfrac{\frac{Re_p}{5}}{1+\left(\frac{Re_p}{5}\right)^{1.52}} + 0.411\dfrac{\left(\frac{Re_p}{263000}\right)^{-7.94}}{1+\left(\frac{Re_p}{263000}\right)^{-8}} + 0.25\dfrac{\frac{Re_p}{10^6}}{1+\frac{Re_p}{10^6}} & Re_p \leq 10^6 \\ 0.19 - \dfrac{8\cdot 10^4}{Re_p} + \delta & Re_p > 10^6 \end{cases}$$

2. ARBITRARY MESH MOTION

In rotorcraft applications, it is frequently a requirement that grids are able to independently move and deform to account for the movement of the rotating blades and flexible structures. The main rotor of a conventional helicopter, for instance, moves in relative motion to the fuselage, whilst in forward flight, the blades are also flapping and pitching to eliminate rolling moments and generate lift. The SU2 solver allows for rigid grid rotations as well as surface and volumetric mesh deformation which makes simulating challenging geometries possible.

In our in-house Lagrangian particle tracking software for an unsteady flow field in a stationary mesh, the particles equations of motion are integrated between two flow solution time-steps and tracked using the known-vicinity algorithm described in Bellosta[25]. The solution remains constant until the particle time reaches the next flow time-step. Within this work, we extend this procedure to unsteady moving mesh problems, which requires the mesh to be updated with the solution. After the update, the particles must be assigned to new elements as the cell containing them before the mesh update may differ. Moreover, during the mesh displacement, some particles may finish outside the computational domain or intersect a boundary more than once and finish back inside the domain. Each of these issues does not allow for the accurate computation of the particle impingement on the boundary, which is crucial for accurately modelling ice accretion.

In order to avoid the aforementioned issues, we account for the mesh displacement as we track the particles. This is computed through combining the particle displacement and the grid displacement to give the displacement of the particle relative to the grid as shown in Fig. 1. This particles relative motion, $\mathbf{x}_{\text{relative}}$ can be expressed as,

$$(3)\quad \mathbf{x}_{\text{relative}} = \mathbf{x}_p - \mathbf{x}_{\text{grid}}$$

The new relative particle position can then be obtained using a forward Euler scheme as,

$$(4)\quad \begin{cases} \mathbf{u}_p^{n+1} = \mathbf{u}_p^n + f(\mathbf{u}_p^n, \mathbf{x}_p^n)\Delta t \\ \mathbf{x}_p^{n+1} = \mathbf{x}_p^n + \mathbf{u}_p^{n+1}\Delta t \; - \underbrace{\mathbf{u}_{\text{grid}}\Delta t}_{\text{grid motion}} \end{cases}$$

where the function f is the right hand side of the velocity equation reported in equation 1 and $\mathbf{u}_{\text{grid}}$ is the grid velocity at the particle position. Using this approach arbitrary mesh motion can be accounted, while the only information needed is the grid velocity in equation (4).

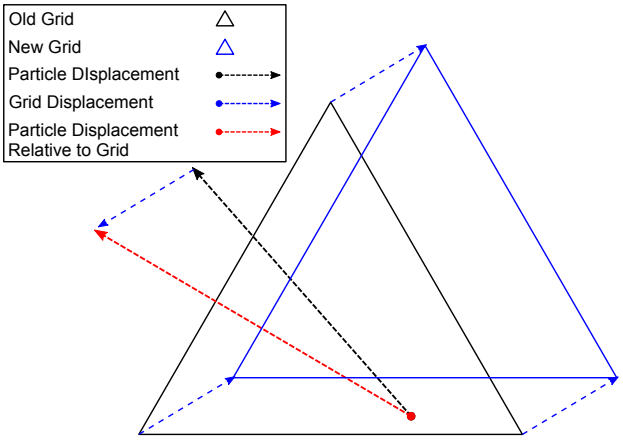


Figure 1: Schematic showing the particles relative displacement within a 2-dimensional triangle.

When the next flow time step is reached the new mesh and solution can be loaded and the new grid velocity obtained. Since the particles were tracked within a relative frame of reference they already occupy the correct relative position and hence cell within the new mesh. Accordingly, the particle position is then updated in order to conserve its relative position within the same cell in the new mesh. This is done through inverting the shape functions of the elements of the old mesh to evaluate the positions in the new mesh as described below,

$$(5)\quad \boldsymbol{\xi} = I_o^{-1}(\mathbf{x}_o) \quad \rightarrow \quad \mathbf{x}_n = I_n(\boldsymbol{\xi})$$

where the subscripts o and n refer to the old and new mesh respectively and I is the function that

maps the local element coordinate $\boldsymbol{\xi}$ to the global position $\mathbf{x}$. The overall process described can be summarized in Fig. 2.

Deforming and rigid grid movement types are compared in Fig. 3 and show the differences in the mesh which are required to achieve identical motion of a pitching airfoil. A two-dimensional verification simulation confirming the particle trajectories accounting for the deforming grid movement is shown in Fig. 4. It compares the particle trajectories in a deforming mesh against a mesh with known rigid motion. It shows that the two results are close enough so that only the top layer of particles are visible which are contained within the deforming mesh.

3. NON-CONFORMAL INTERFACES

The support of the supermesh feature within SU2 allows multiple independently moving grids to slide within one another whilst maintaining connectivity between them. Each zone is considered as a separate entity within SU2 where the flow field is subsequently solved and associated with the corresponding zones solution. Different techniques can be used for the interpolation between zone interfaces such as the nearest-neighbour and weight-average approaches. One requirement of the supermesh implementation within SU2 is that there must not be any continuous voids between grids and the interfaces must be complementary.

Once the solution of the flow field is computed the multi-zone particle tracking can start. The process used to implement the multi-zone particle tracking based on the sliding mesh scheme is outlined in Fig. 5 and will now be discussed. First, each zone's mesh and its corresponding solution is loaded. A cloud is then defined for each of the zones. Within each zone's domain, a cloud of supercooled water droplets is initialized and the fluid data from the zone's solution are applied to the particles. The cloud is then updated as the particles are tracked using a forward Euler time integration algorithm until the final simulation time is reached. To ensure a particle travelling along a trajectory crossing a boundary interface can communicate from one zone to another and does not simply stop at the intersection, the following method is proposed.

With the known-vicinity algorithm mentioned above the boundaries can easily be identified during intersections. The final position, $\mathbf{x}_p$ of a particle can then be described dependent upon its trajectory and intersections,

$$\mathbf{x}_p = \begin{cases} \mathbf{x}_{\text{intersection}} & \text{boundary intersection} \\ \mathbf{x}_{\text{destination}} & \text{interface intersection} \\ \mathbf{x}_{\text{destination}} & \text{no intersection} \end{cases} \tag{6}$$

When a particle impinges on an interface boundary a transfer routine is called to pass the particle onto the neighbouring zone. Inside the interface routine, the particle being transferred over the interface from the donor cloud to the target cloud is assigned within the new target mesh. The donor cloud then consequently loses a particle and the target cloud gains a particle.

The particle is then located in the target mesh where either a brute force or nearest neighbour algorithm is used to find the cell with the closest centroid to speed up the search. Once the cell containing the particle is found, the fluid properties are then assigned to the particle using an inverse distance weighted interpolation. The particle then continues to be tracked in the target mesh for the following time-steps until the simulation time either finishes or another boundary interface intersection occurs in which case the particle is transferred across the next interface into the next target zone.

If however, during the particle transfer from the donor to the target mesh, the particle end position is in a void as it has left the donor mesh but not entered the target mesh due to the non-conformal boundary interfaces not overlapping, an additional check is required. In this case, the particle maintains the previous time-steps fluid properties and a semi-adaptive integration time-step is used to ensure that during the following time-step it enters the target mesh. The particle can then inherit the fluid properties of the target mesh cell-Id and again continue to be tracked in the target mesh. The donor or target host, of any particle, $\mathcal{P}$, can thus be described by the following conditions,

$$\mathcal{P} = \begin{cases} \mathcal{P}_{\text{donor}} & \text{boundary intersection} \\ \mathcal{P}_{\text{target}} & \text{interface intersection} \\ \mathcal{P}_{\text{donor}} & \text{in a void} \end{cases} \tag{7}$$

This process can be shown in the schematic from Fig. 6 where it shows the effect of tracking a particle across multiple zones. For clarity, the stationary donor outer mesh is shown in blue whilst the moving target inner mesh is shown in red. To distinguish which zone each particle is in the particles within the stationary donor outer zone are shown in pink whilst the particles within the inner target moving zone are shown in green. The schematic shows

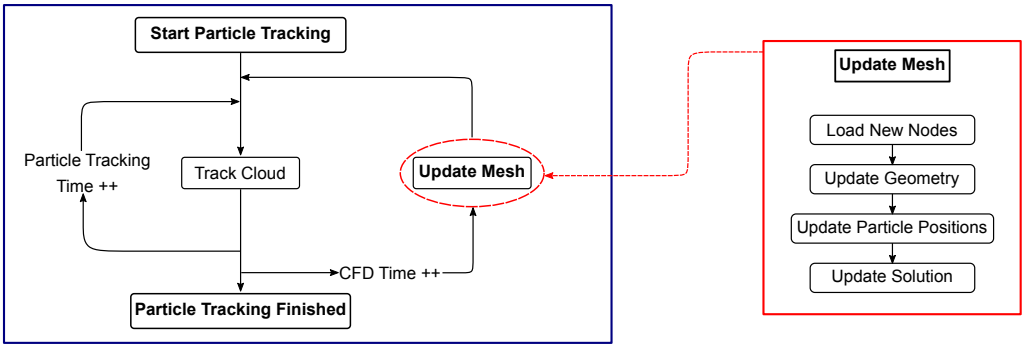


Figure 2: Flowchart displaying the process for particle tracking simulations in moving mesh.

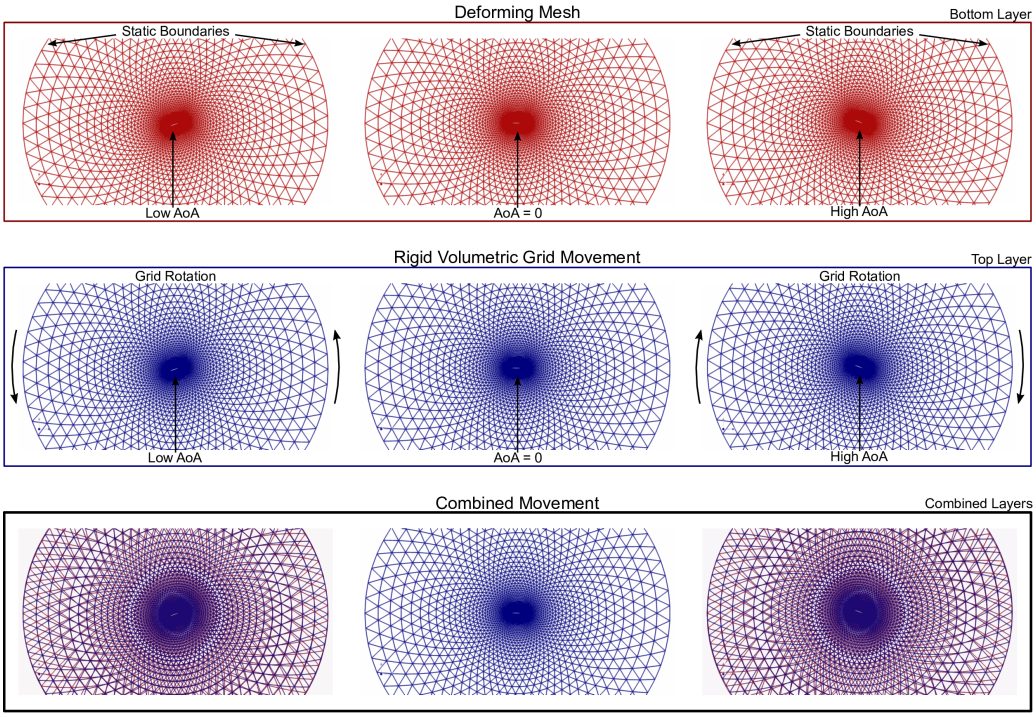


Figure 3: Different kinds of grid movement used to apply motion to an arbitrary object.

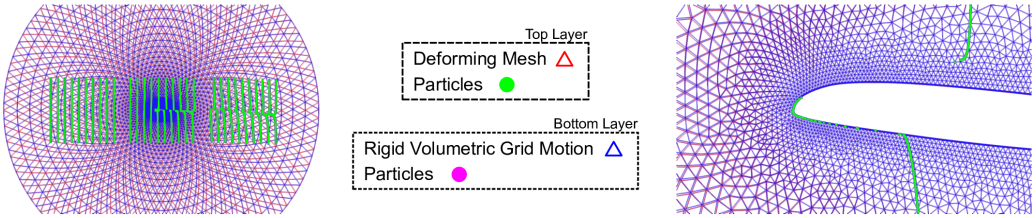


Figure 4: Particle tracking simulation comparing the kind of grid movement. On the left, the particle front is shown at different time steps. On the right, a single time snapshot is shown close to the airfoil.

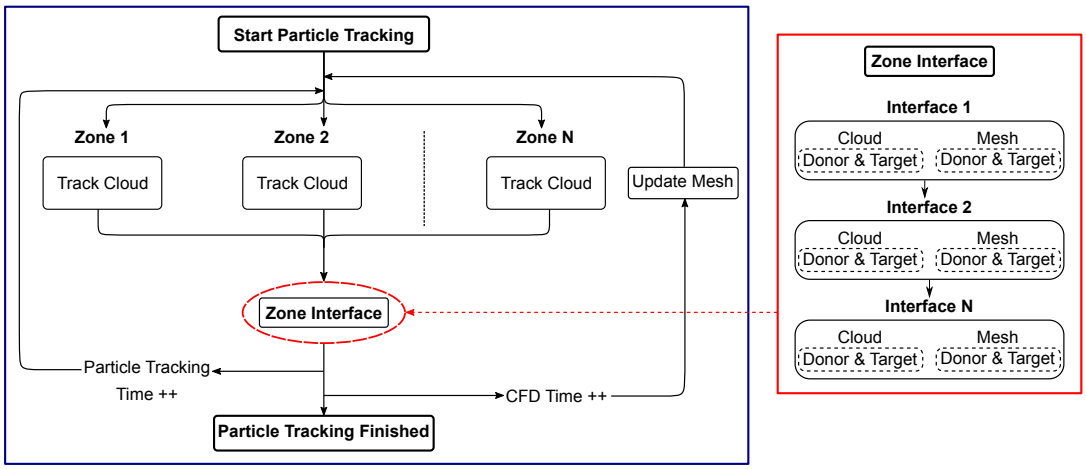


Figure 5: Flowchart displaying the process for a multi-zone particle tracking simulation. Including the implementation of the boundary interface cloud transfer from donor to target meshes.

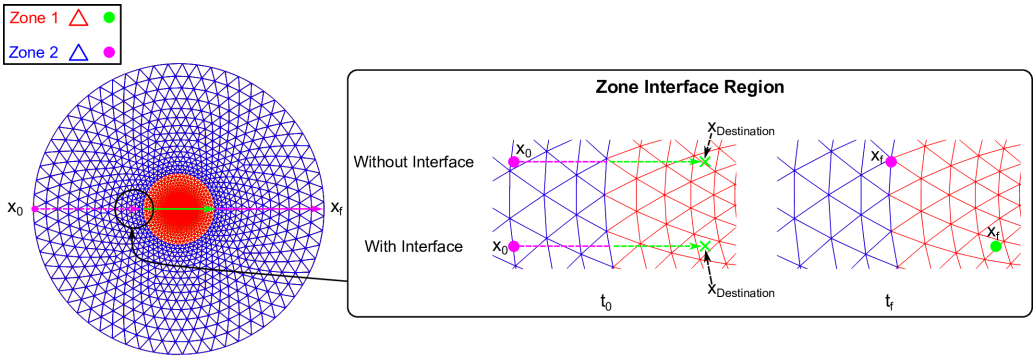


Figure 6: Schematic showing particle tracking across a boundary interface with and without zone interface treatment.

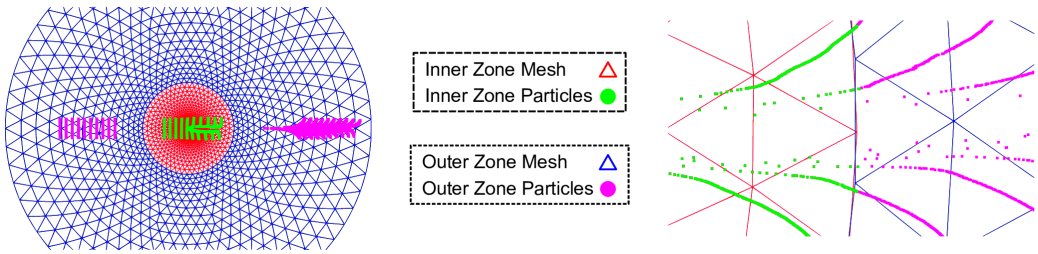


Figure 7: Results of a 2-dimensional pitching airfoil particle tracking simulation showing the multi-zone implementation and passing of particles between donor and target zones. On the left, the particle front is shown at different time steps. On the right, a single snapshot is shown close to the interface.

that when accounting for the zone interface a particle can arrive at its final destination whereas when a particle is tracked without accounting for the zone interface remains bound to the edge of the zone interface as if the zone interface represents a boundary wall.

A two-dimensional verification simulation ensuring the conservation of the particle trajectories from one zone to another is shown in Fig. 7. It shows the particles being successfully transferred between donor and target zones and highlights the partition of the cloud at the interface where the two mesh are non-conformal. It features particles in voids which maintain the donor cloud properties without being lost.

4. RESULTS

The compressible Euler equations within SU2 are used to resolve the flow for the verification of the proposed particle tracking methods in arbitrary mesh motion and across non-conformal interfaces. Differently from the CFD model which does not account for viscosity, the particle equation of motion accounts for viscous effects. In this work, the viscosity is computed locally based on the temperature field available from the CFD solution using Sutherland's law[26]. Three exemplary test cases provide numerical verification. A single-zone grid undergoing mesh deformation is consider in Section 4.1, while, two- and three-zone grids of varying complexity experiencing rotation are considered in Section 4.2 and 4.3. All tests are conducted under unsteady conditions in subsonic flow regimes. In each of the test cases, particulates are initialized upstream in the far-field and traced while entrained in the flow.

4.1. Pitching Wing

The first set of computations are performed on a straight untwisted NACA 0012 wing configuration. The wing has a nominal aspect ratio of, $AR = 4$ and has sharp rectangular edges at the wing tips. A low free-stream Mach number of, $M_\infty = 0.3$ is specified. The wing is pitching about its quarter-chord axis where the following conditions describe its motion; the mean angle of attack is $\alpha_0 = 8^\circ$, and it experiences 5° of oscillation so that the minimum angle of incidence is $\alpha_{min} = 3^\circ$ and its maximum angle of incidence is $\alpha = 13^\circ$. The simulation lasts for six full periods of oscillations.

A fully unstructured mesh is used containing a total of $1\,506\,549$ elements and $269\,383$ nodes. Along each section of the wing surface in the chord-wise direction 300 nodes congregate towards the leading and trailing edge. There are a total of $43\,090$ quad elements on the wing and far-field surface meaning there are $43\,090$ associated pyramid elements in the volume mesh. These pyramid elements are then connected to the remaining tetrahedral volume elements. The far-field is placed 8 chord lengths from the tips of the wing and 10 chord lengths from the leading and trailing edges. The coordinate system is defined so that the free-stream velocity is in the positive x-direction and the pitching motion is along the y-axis whilst z is positive in the upward direction. The single-zone mesh described is shown in Fig. 8.

The SU2 software suite[12] is able to solve implicitly unsteady problems using a dual-time stepping strategy, leading to second-order accuracy in time. While a Monotonic Upwind Scheme for Conservation Laws coupled with the Venkatakrishnan limiter is used in the flow equations for second order accuracy. The computation prescribes a non-dimensional time-step $\Delta t U_\infty / c = 0.12$ to provide sufficient temporal resolution. This value of Δt corresponds to 100 time-steps per pitching cycle which means 600 flow fields in total are saved for the unsteady particle tracking computations. To account for the motion of the pitching wing the volume mesh is deforming at each unsteady time-step. The technique for the volume mesh deformation is based on the linear elasticity analogy. The element stiffness imposed for the mesh deformation is based on the inverse volume so that the mesh deformation mainly occurs in the far-field away from the wing surface. During each deformation, there are 500 internal smoothing iterations and 3 nonlinear deformation iterations to maintain the deformed grid quality. The resultant flow field of the pitching wing is shown in Fig. 9 where it clearly shows the three-dimensional effects caused by the wing tips as strong vortical structures are present. It also shows that at the center of the wing the flow can also be classified almost entirely as two-dimensional.

Using the method described during Section 2 to account for arbitrary mesh motion the unsteady particle tracking of a wing during pitching motion supported via mesh deformation is shown in Fig. 10. The particle tracking simulation uses $50\,000$ uniformly distributed supercooled water droplets which are positioned as two cloud fronts upstream in the farfield. The particles have a mean volume diameter of $20\ \mu\mathrm{m}$ and a constant liquid water density of $1\,000\ \mathrm{kg/m^3}$. The particles are tracked with an integration time-step of $10^{-5}\ \mathrm{s}$ for a simulation time of $0.2\ \mathrm{s}$ whilst the mesh is updated 100 times per oscillation. It shows that during the pitching cycle at different time instances the quantity of particles impacted is highly influenced by the angle of

attack. At instances of higher angles of attack, the aerodynamic forces are not great enough to exceed the inertial forces of the particles and so more particles impact on the wing. At high angles of attack, there is also slight flow separation causing a small omega-type vortex, which was first described experimentally by Lorber et al.[27] and can be visualized in the trajectories of the particles. The tip vortices are also clearly visible in the particle trajectories as the particles roll up and disperse from the core of the vortex.

4.2. Rotor in Hover

The Caradonna and Tung rotor[28] is used for the assessment of the non-conformal interface particle treatment. It is a two-bladed untwisted rotor with a NACA 0012 profile. The experimental rotor has a 1.143 meter radius an aspect ratio of, $AR = 6$. The test conditions used are based on Table 18 from the experiment which consists of a rotor with a collective pitch of $\Theta_c = 8°$, a rotational velocity of $\Omega = 1\,500$ rpm, and consequently a tip Mach number of $M_{tip} = 0.526$. These test conditions were chosen so that the rotor induced velocity would visibly affect the particle trajectories.

A fully unstructured multi-zone mesh is used containing two-zones. The first zone is the stationary background mesh comprising of $237\,015$ elements and $41\,850$ nodes. The external interface – the location at which properties of flow are exchanged between zones – is positioned 0.43 radii from the tips of the rotor blades and has $8\,668$ elements. The far-field is positioned just over 6 radii from the rotor blade tips. The second zone is the moving inner mesh with the embedded rotor blades and comprises of $1\,598\,613$ elements and $283\,434$ nodes. The internal interface is positioned complementary to the external interface and likewise has $8\,668$ elements on the surface. Each of the rotor blades then has $23\,024$ elements on the surface. The fluid region in both zones contains mainly tetrahedral volume elements with pyramid elements connecting the quadrilateral surface elements to the tetrahedral elements. The total number of elements and nodes in the multi-zone grid is thus $1\,835\,628$ and $325\,284$ respectively. The multi-zone mesh described is shown in Fig. 11.

The supermesh method described by Rinaldi[11] is used to account for the conservation of fluid properties between the non-conformal boundary interfaces which are present in multi-zone problems. A weight average interpolation between the concentric mesh is used. The resultant flow field of the rotor blades is shown in Fig. 12. Here the z-component of the velocity represents the rotors induced velocity and the connectivity between the zones solutions is apparent. The results from Fig. 12a display the velocity field in a diametric plane near and below the rotor operating in hover and shows the fluid velocity increases as it passes into and through the rotor disk plane. As the velocity is greatest at the rotor tips, the strength of the downward velocity is greatest. Due to there being no rotor hub modelled, there is an upward velocity present at the root cutout. Fig. 12b shows the disk plane of the rotor and shows how the flow field is azimuthally axisymmetric.

Using the method outlined in Section 3 for the treatment of non-conformal interfaces the results of particle tracking through the flow field of a rotor in hover via sliding mesh are shown in Fig. 13. The particle tracking simulation initializes a singular uniformly distributed cloud just above the rotor disk and interface which contains $40\,000$ supercooled water droplets. The duration of the particle tracking simulation is long enough for 10 full revolutions of the main rotor. It shows that the induced velocity produced from the rotor is significant enough to entrain the particles into and through the rotor disk plane. The particles then trail behind and below each blade and are convected through the wake. The particles remain within the wake boundary which separates the quiescent flow from the rotor wake. As the trajectories of the particles progress there is a contraction which represents an increase in the particle velocity. As the particles then move further into the free-stream the contrary occurs as the flow begins to numerically dissipate where the grid coarsens. Also as the downwash through the rotor ages, the particles begin to suppress as the induced velocity decreases. Finally, it is possible to visualize the blade passage frequency in the perturbations of the particles as they convect downwards.

4.3. Full Tilt-Rotor Configuration

The purpose of this test case is to show the effectiveness of the particle tracking in sliding mesh on a problem of engineering interest and so the Tiltrotor by Leonardo Helicopters in forward flight was chosen due to it being designed for flight in known icing conditions. Due to much of the data for this aircraft not being in the public domain, the majority of the geometry has been approximated. The tilt-rotors main characteristics taken from Malpica[29] can be summarized. Each rotor has 3 blades which have a nominal rotor speed of 569 rpm. The rotor radius is $3.96\ \mathrm{m}$ resulting in a tip speed of Mach 0.695. The blades are non-linearly taped and twisted. The blade twist from root to tip is $47.5\ \mathrm{deg}$ and the rotor solidity is 0.09. The rotor airfoil ge-

ometry is based on the Narramore airfoil design[30]. The fuselage geometry is taken from GrabCAD[31] and simplified to allow for unstructured mesh generation. The period of time for this simulation is 6 full blade rotations, where one set of blades rotates in the clockwise direction and the other the anticlockwise direction to eliminate the moments produced from each of the rotors. That said, the incentive here is to demonstrate highly complex and challenging particle tracking methods and not for the verification of in-flight icing trails.

A fully unstructured multi-zone mesh is used containing three-zones. The first zone is the stationary background mesh comprising of 6 410 004 elements and 1 110 729 nodes. The fuselage contains 427 329 elements. The clockwise and anticlockwise external interface cutouts contains 32 012 elements and the far-field contains 14 340 elements. The second zone is the clockwise set of rotor blades comprising of 5 504 883 elements and 980 020 nodes. The rotor blades, nacelle and internal interface contain 104 782, 13 586, and 14 054 elements respectively. The third zone is the anticlockwise set of rotor blades and is identical to the clockwise mesh except for the direction of which the blades rotate. The total size of the three-zone mesh is 17 419 770 elements and 3 070 769 nodes. The multi-zone mesh described is shown in Fig. 14.

Once again the supermesh technique is used to account for the non-conformal interfaces due to the multiple zones in the flow field. The resultant flow field computations of the tilt-rotor configuration in aeroplane mode are shown in Fig. 15. The velocity magnitude is shown here to display the combined contributions of the two rotors and the fuselage on the flow field. It portrays how the rotor wake directly influences and disturbs the flow over the wing and nacelle. The rotor blades highly twisted geometric nature allows them to efficiently operate in both hover and forward flight, however, the flow physics associated with them is very complex. It appears due to the highly twisted root and slightly negatively twisted tip that the largest forward propelling force is produced around 70% along the radius of the blades.

Using the method discussed within Section 3 for the treatment of non-conformal interfaces the results of the particle tracking through the flow field of the tilt-rotor in aeroplane mode are shown in Fig. 16. The particle tracking simulation initializes 40 000 uniformly distributed supercooled water droplet particles upstream of the aircraft. The particle tracking simulation lasts for 6 full revolutions of the rotor blades to ensure the particles are entrained past the aircraft. It shows that the deformation of the particle trajectories is quite large. As the particles transfer from the outer stationary zone into the moving internal zones they interact with the rotor blades due to their relative motion being significantly greater. These particles then enter the wake of the rotor blades which increases the velocity and causes the particles to be ahead of the main front of particles. The particles trajectories which are influenced by the fuselage, wing, nacelle, and tail of the aircraft however slow and appear slight aft of the main front of particles. The location with the highest collection efficiency, by a large margin, is the rotor blades which for flight in icing conditions is concerning.

5. CONCLUSION

Two new Lagrangian particle tracking techniques to account for arbitrary mesh motion and non-conformal interfaces are introduced. The simulation of clouds containing supercooled water droplets on complex problems of engineering interest are shown. The first test case presented exhibits particles moving throughout continuously deforming mesh without being lost or misplaced. The second and third test cases feature particles moving across non-conformal interfaces in sliding mesh. Most computational rotorcraft icing codes use the Eulerian frame of reference for particle tracking, whereas, the most obvious description of the dynamics of individual particles is in a Lagrangian frame of reference. An important feature of the Lagrangian framework is that it allows for a better description of the underlying physics of supercooled large water droplets (SLD). An example of this is at high velocities close to the rotor blade tips where particles impact causing splashing. Another example is at high Weber numbers when SLD go through instabilities and break up into smaller droplets.

ACKNOWLEDGMENTS

The NITROS (Network for Innovative Training on ROtorcraft Safety) project has received funding from the European Union's Horizon 2020 research and innovation program under the Marie Skłodowska-Curie grant agreement No. 721920.

REFERENCES

[1] RW Gent, NP Dart, and JT Cansdale. Aircraft icing. *Philosophical Transactions of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, 358(1776):2873–2911, 2000.

[2] TP Ratvasky, BP Barnhart, and S Lee. Current methods modeling and simulating icing effects on air-

craft performance, stability, control. *Journal of Aircraft*, 47(1):201–211, 2010.

[3] JA Benek, JL Steger, FC Dougherty, and PG Buning. Chimera. A grid-embedding technique. Technical report, Arnold Engineering Development Center Arnold AFB TN, 1986.

[4] ZJ Wang. A fully conservative interface algorithm for overlapped grids. *Journal of Computational Physics*, 122(1):96–106, 1995.

[5] ZJ Wang, N Hariharan, and R Chen. Recent development on the conservation property of chimera. *International Journal of Computational Fluid Dynamics*, 15(4):265–278, 2001.

[6] R Steijl, G Barakos, and K Badcock. A framework for CFD analysis of helicopter rotors in hover and forward flight. *International Journal for Numerical Methods in Fluids*, 51(8):819–847, 2006.

[7] M Biava. Rans computations of rotor/fuselage unsteady interactional aerodynamics. *Doctor of Philosophy Thesis, Politecnico di Milano, Milano, Italia*, 2007.

[8] N Kroll, CC Rossow, K Becker, and F Thiele. The megaflow project. *Aerospace Science and Technology*, 4(4):223–237, 2000.

[9] GR Srinivasan and JD Baedert. Turns: A Free-wake Eule/Navier-Stokes numerical method for helicopter rotors. *AIAA Journal*, 31(5):959–962, 1993.

[10] PE Farrell, MD Piggott, CC Pain, GJ Gorman, and CR Wilson. Conservative interpolation between unstructured meshes via supermesh construction. *Computer Methods in Applied Mechanics and Engineering*, 198(33-36):2632–2642, 2009.

[11] E Rinaldi, P Colonna, and R Pecnik. Flux-conserving treatment of non-conformal interfaces for finite-volume discretization of conservation laws. *Computers & Fluids*, 120:126–139, 2015.

[12] TD Economon, F Palacios, SR Copeland, TW Lukaczyk, and JJ Alonso. SU2: An open-source suite for multiphysics simulation and design. *AIAA Journal*, 54(3):828–846, December 2015.

[13] R Steijl and G Barakos. Sliding mesh algorithm for CFD analysis of helicopter rotor–fuselage aerodynamics. *International Journal for Numerical Methods in Fluids*, 58(5):527–549, 2008.

[14] HJ Nam, YM Park, and OJ Kwon. Simulation of unsteady rotor-fuselage aerodynamic interaction using unstructured adaptive meshes. *Journal of the American Helicopter Society*, 51(2):141–149, 2006.

[15] Rainald Löhner and John Ambrosiano. A vectorized particle tracer for unstructured grids. *Journal of Computational Physics*, 91(1):22–31, 1990.

[16] A Haselbacher, FM Najjar, and JP Ferry. An efficient and robust particle-localization algorithm for unstructured grids. *Journal of Computational Physics*, 225(2):2198–2213, 2007.

[17] R Chorda, JA Blasco, and N Fueyo. An efficient particle-locating algorithm for application in arbitrary 2D and 3D grids. *International Journal of Multiphase Flow*, 28(9):1565–1580, 2002.

[18] Q Zhou and MA Leschziner. An improved particle-locating algorithm for Eulerian-Lagrangian computations of two-phase flows in general coordinates. *International Journal of Multiphase Flow*, 25(5):813–825, 1999.

[19] GA Ruff and BM Berkowitz. Users manual for the nasa Lewis ice accretion prediction code (LEWICE). 1990.

[20] T Hedde and D Guffond. ONERA three-dimensional icing model. *AIAA journal*, 33(6):1038–1045, 1995.

[21] H Beaugendre, F Morency, and WG Habashi. FENSAP-ICE's three-dimensional in-flight ice accretion module: ICE3D. *Journal of Aircraft*, 40(2):239–247, 2003.

[22] G Gori, M Zocca, M Garabelli, A Guardone, and G Quaranta. PoliMIce: A simulation framework for three-dimensional ice accretion. *Applied Mathematics and Computation*, 267:96–107, 2015.

[23] Z Wang, N Zhao, and C Zhu. Numerical simulation for three-dimensional rotor icing in forward flight. *Advances in Mechanical Engineering*, 10(4):1687814018772404, 2018.

[24] X Chen, Q Zhao, and G Barakos. Numerical Analysis of Aerodynamic Characteristics of Iced Rotor in Forward Flight. *AIAA Journal*, pages 1–15, 2018.

[25] T Bellosta, G Parma, and A Guardone. A robust 3D particle tracking solver for in-fight ice accretion using arbitrary precision arithmetic. In *VIII International Conference on Computational Methods for Coupled Problems in Science and Engineering*, 2019.

[26] W Sutherland. Lii. the viscosity of gases and molecular force. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 36(223):507–531, 1893.

[27] P Lorber, A Covino, and F Carta. Dynamic stall experiments on a swept three-dimensional wing in compressible flow. In *22nd Fluid Dynamics, Plasma Dynamics and Lasers Conference*, page 1795, 1991.

[28] FX Caradonna and C Tung. Experimental and analytical studies of a model helicopter rotor in hover. NASA Technical Memorandum 81232, September 1981.

[29] C Malpica. Parametric investigation on the use of lateral and logitudinal rotor trim flapping for tiltrotor noise reduction. In *AHS International 73rd Annual Forum and Technology Display, Fort Worth, Texas, USA*, 2017.

[30] JC Narramore. Airfoil design, test, and evaluation for the v-22 tilt rotor vehicle. In *43rd Annual Forum of the American Helicopter Society, St. Louis, Missouri*, 1987.

[31] GrabCAD Inc. GrabCAD: Design community, cad library, 3d printing software. `https://grabcad.com`, 2009.

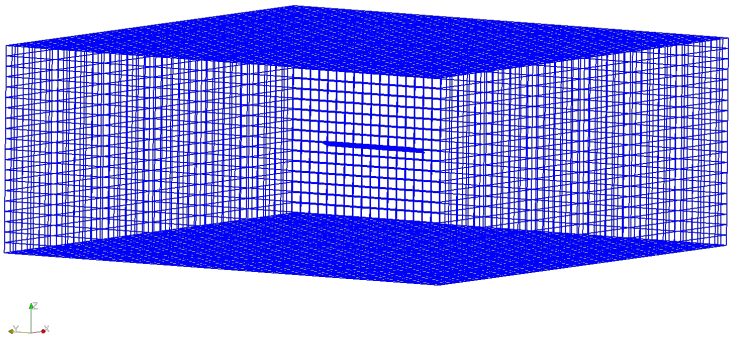
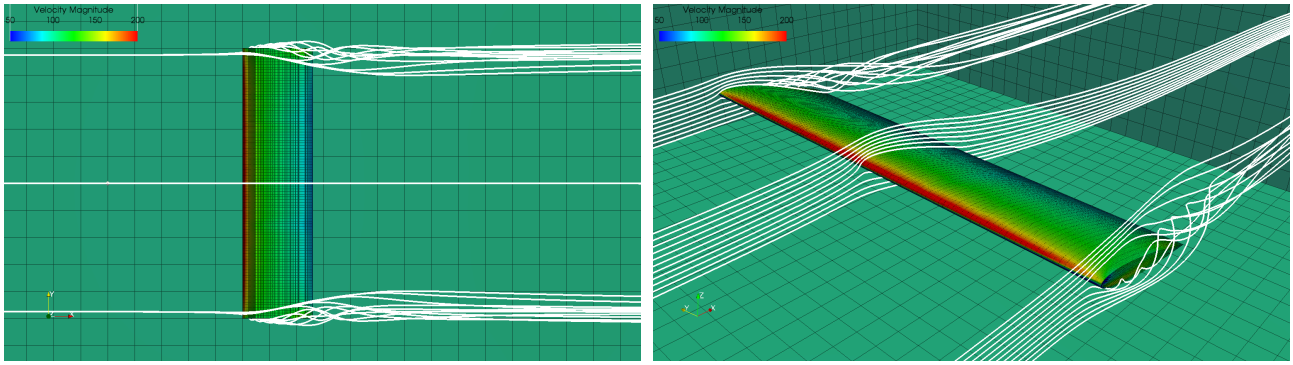
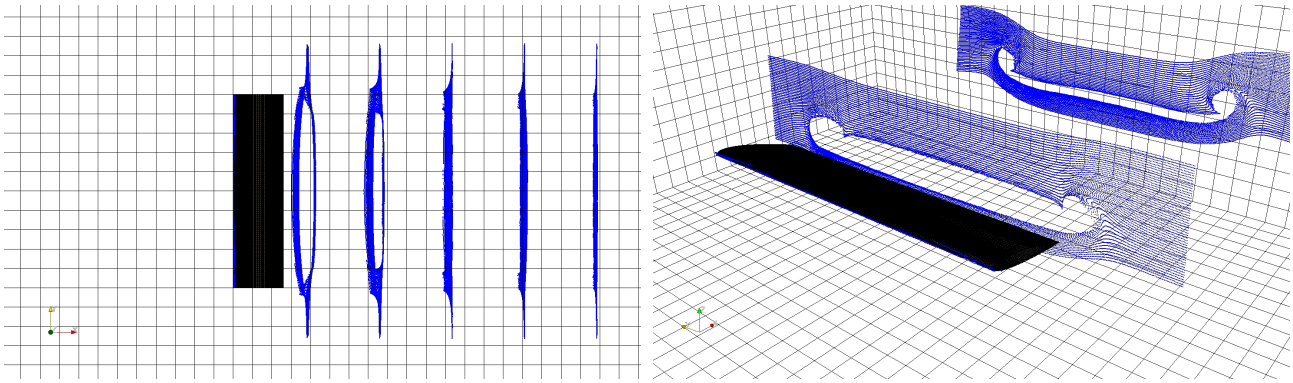


Figure 8: Single-zone mesh containing a wing undergoing pitching motion via mesh deformation.



(a) View from above (b) Isometric view

Figure 9: Flow field of a wing during pitching motion displaying streamlines at the wing tips and center of the wing. Exhibiting how complex tip vortices roll up in opposite directions.



(a) View from above (b) Isometric view

Figure 10: Visualization of particle trajectories over a pitching wing in a deforming mesh. Displaying different time instances at which the particles pass over the wing. The initial front is a planar rectangle.

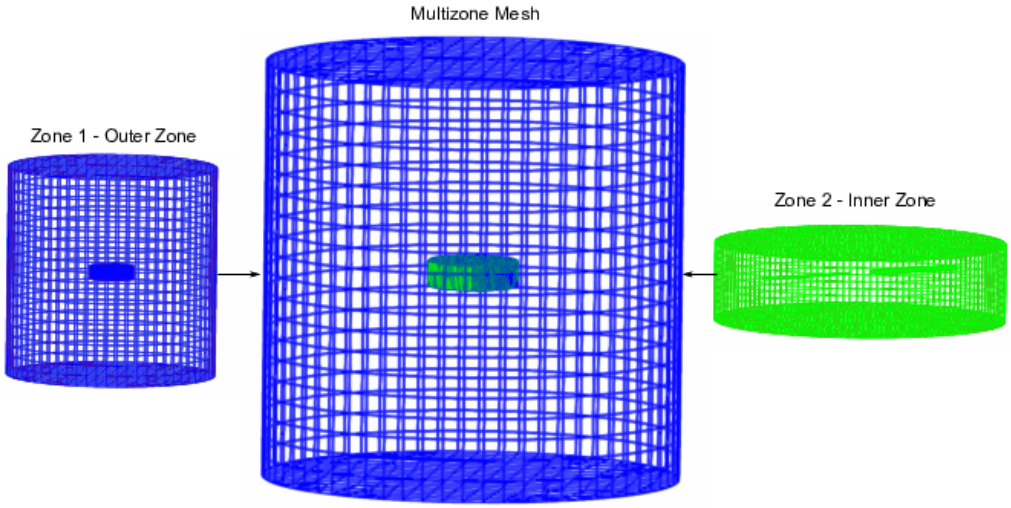
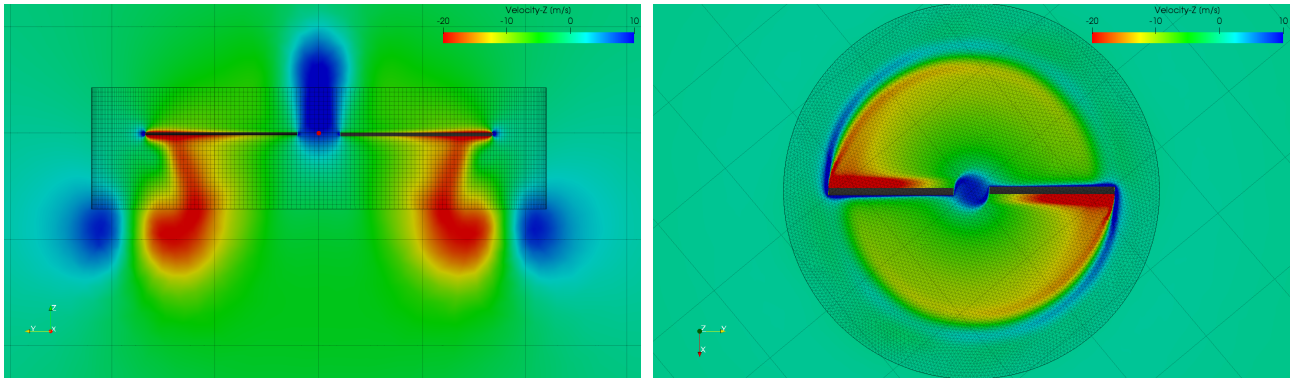


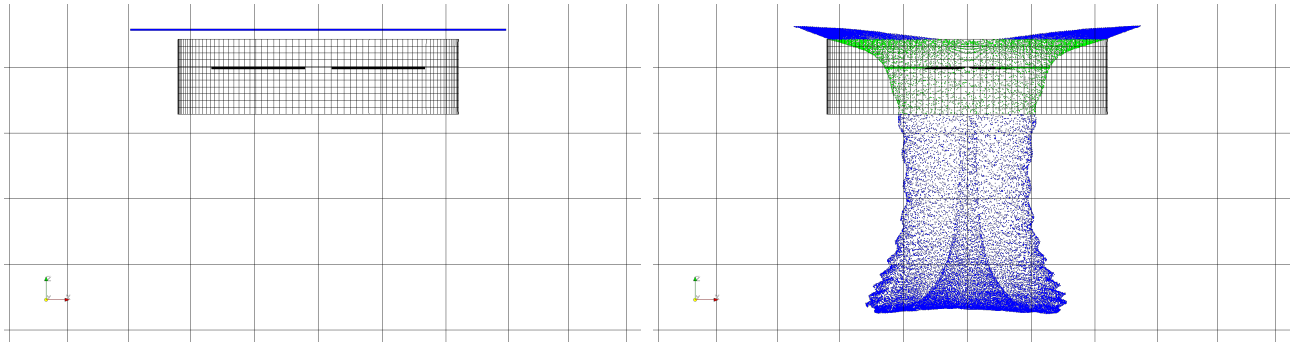
Figure 11: Multi-zone mesh containing a rotor in hover via sliding mesh.



(a) Diametric plane

(b) Disk plane

Figure 12: Results of the flow field of a rotor in hover. Displaying the connectivity present between the multi-zone solutions.



(a) Initialization of particles above the rotor

(b) End position of particles

Figure 13: Results of the particle tracking of a rotor in hover. Displaying the connectivity between zones where the particles in the outer zone are and inner zone are represented by blue and green respectively.

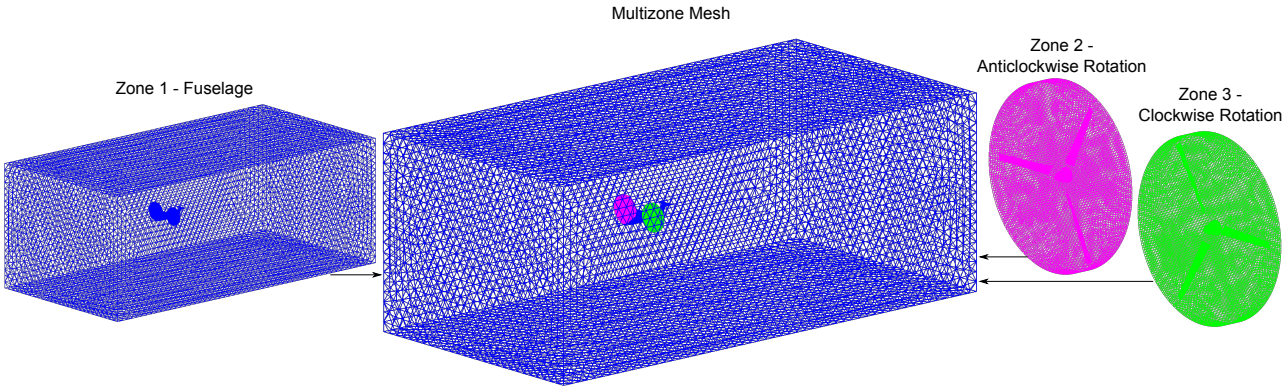
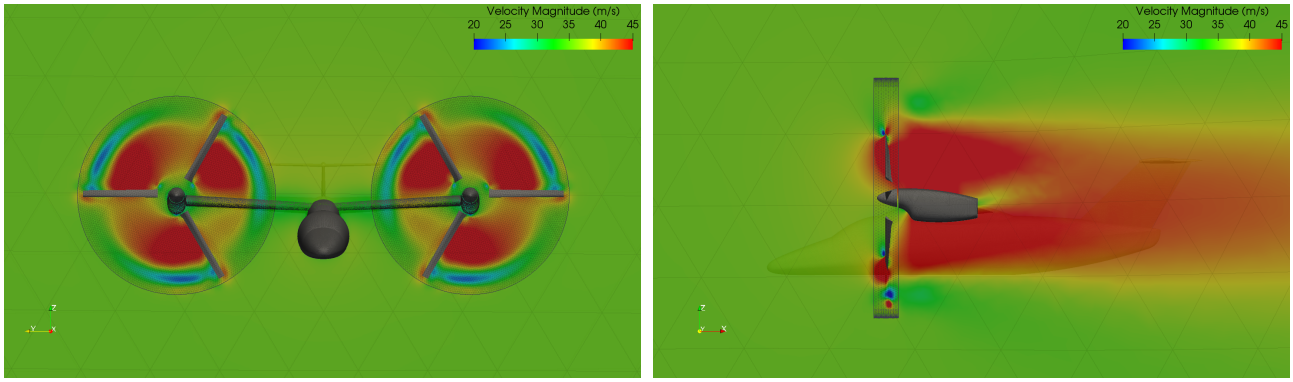


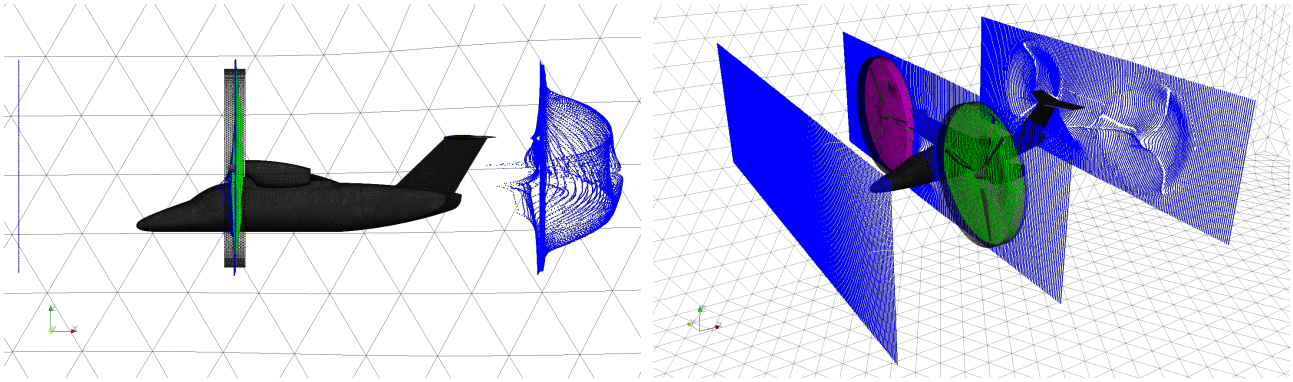
Figure 14: Multi-zone mesh containing a tilt-rotor configuration in forward flight via sliding mesh.



(a) Cross section through the disk plane

(b) Cross section through the rotor, wing and nacelle

Figure 15: Results of the flow field of a tilt-rotor configuration in aeroplane mode. Displaying the connectivity between the rotating zones and stationary zone.



(a) View from above

(b) Isometric view

Figure 16: Results of the particle tracking over a tilt-rotor in aeroplane mode. Displaying the connectivity between zones where the particles in the 1, 2 and 3 zones are represented by blue, green and pink respectively.