Received: August 3, 2022 | Revised: January 9, 2023 | Accepted: January 10, 2023 | Published on: January 27, 2023

10.1115/1.4056681

JOURNAL OF VERIFICATION, VALIDATION AND UNCERTAINTY QUANTIFICATION

Influence of Interpolation Scheme on the Accuracy of Overset Method for Computing Rudder-Propeller Interaction

Sébastien Lemaire*[1,2] | Guilherme Vaz[3,1] | Menno Deij - van Rijswijk[2] | Stephen R Turnock[1]

[1]Dept of Civil, Maritime and Environmental Eng, University of Southampton, Southampton, United Kingdom
[2]R&D Dept, MARIN, Wageningen, The Netherlands
[3]HPC & Data Science Team, blueOASIS, Ericeira, Portugal

Correspondence
*Sébastien Lemaire. Email: sebastien.lemaire@soton.ac.uk

Abstract

The overset method and associated interpolation schemes are usually thoroughly verified only on synthetic or academic test cases for which conclusions might not directly translate to real engineering problems. In the present work, an overset grid method is used to simulate a rudder-propeller flow, for which a comprehensive verification and validation study is performed. Three overset-related interpolation schemes (first order inverse distance, second order nearest cell gradient and third order least squares) are tested to quantify and qualify numerical errors on integral quantities, mass imbalance, flow features and rudder pressure distributions. The performance overhead is also measured to help make accuracy vs. performance balance decisions. Rigorous solution verification is performed to estimate time and space discretisation, iterative and statistical uncertainties. Validation of the propeller-rudder flow against experimental data is also done.

The results show that, while the choice of interpolation scheme has minimal impact on time-averaged integral quantities (like propeller and rudder forces), they do influence the smoothness of the time signals, with the first order scheme resulting in large intensity high-frequency temporal oscillations. Lower order interpolation methods also produce more interpolation artefacts in fringe cells, which are then convected downstream. Mass imbalance is also affected by the interpolation scheme, with higher order schemes such as the third order least squares approach resulting in an order of magnitude lower flux errors. The limitations of first order schemes do not, however, result in significant lower computational overhead, with the second order nearest cell gradient being even cheaper than the inverse distance scheme in the tested implementation. Lastly, validation shows promising results with rudder forces within 10% of the experiments.

KEYWORDS:
Overset Grid; Interpolation; Propeller; Verification; Validation; Error quantification

1 | INTRODUCTION

The overset method allows the interaction between objects moving relative to each other to be captured in finite volume computational fluid dynamics (CFD) simulations. Each object is placed on its own body fitted mesh and interpolation is used to couple individual meshes to each other as well as the underlying background mesh.

The versatility of the overset grid method in terms of grid design and motion comes, however, at the cost of interpolation errors generated at meshes' interfaces[1]. These errors propagate in the domain and can lead to spurious oscillations, mass imbalance, amplitude and phase errors, etc.[2]. At the root of all of these effects is the interpolation scheme, and although very few studies assess these issues in detail, the interpolation methodology is often omitted from the description of the numerical setup or not detailed enough to be reproduced[3,4,5,6,7,8]. In recent years, however, some studies started tackling interpolation errors more thoroughly by comparing different interpolation schemes. This is what Chandar et al.[2,9,10,11] did as they tested several schemes of various order of accuracy on different flow solvers. Their studies focused primarily on academic test cases like the flow around a moving circular cylinder or a NACA foil. Similarly, Brazell et al.[12] performed a verification study on a Ringleb flow using using polynomial interpolation schemes of different order. Finally, the authors of the present work used the method of manufactured solution to carry a verification exercise at high Reynolds number[1,13], again comparing a wide range of interpolation schemes. From all these studies, the general consensus is that overset errors become negligible compared to other discretisation errors with high order interpolation schemes (i.e. of order equal or higher than the space discretisation order). They also show that the lack of mass conservation leads to spurious high frequencies oscillations that could be mitigated using special treatments at the overset interface[11].

While such studies are essential to the assessment of the overset method's accuracy and to get better insight into error propagation patterns and sources, it is often hard to draw quantitative conclusions for real engineering applications directly as they focus on simplified problems. Translating error analysis for more complex and realistic test cases used in the industry is not trivial. Even though publications applying the overset methods on problems relevant for engineering application exist and are numerous, their angle of interest is not the error analysis due to the overset method but the application itself. Dooley et al.[14] for example studied the various dynamic effects of having a helicopter in the vicinity of a ship in waves without however discussing the overset method itself employed for the computation. Similarly, Gatin et al.[15] focuses on the maneuvering characteristics of a ship without discussing overset related uncertainties and errors. While, once again, such studies are important to understand the dynamics and physics behind this systems, their conclusion would be reinforced and made stronger with an associated verification study made on the overset method and its associated error sources.

For this reason, the present work will focus on the complex and realistic problem of a rudder behind a propeller subjected to a drift angle, replicating experiments done by Molland and Turnock[16,17,18] and Verification and Validation will be performed using a variety of interpolation schemes. In this context, an accurate capture of the flow swirl after the propeller for capturing the spanwise pressure variation over the rudder is essential to assess the maneuverability and characteristics of a propulsion system. Now that CFD has become more ubiquitous by getting more accessible and more accurate, such test cases are replicated computationally. In particular, this set of experiments have been computed by Phillips et al.[19] and Villa et al.[20] both using a RANS-BEM coupling, which requires no moving objects in the domain and uses a single mesh. This coupling, allows the propeller to be modeled by the BEM-method, thereby lowering the CFD grid cell count and making the computation a lot faster, at the cost of poorer accuracy of the propeller flow. In order to simulate the propeller instead of modeling it, Badoe et al.[21] used sliding meshes to rotate the propeller. While this approach is often used for propeller flows, for rudders, due to the lack of clearance to account for two cylindrical domains, sliding grids is a restrictive approach, and often needs rudder remeshing for every angle of attack. Such limitations are lifted with the overset grid method. Instead of relying on pre-determined interfaces of the meshes to slide along one another, this method overlaps the grids and dynamically assigns 'interface cells' (called *fringe* cells) to allow for the transfer of information between the different meshes. For example, replicating another set of experiments, Yilmaz et al.[22] used an overset grid on the rudder while the propeller was simulated with a sliding mesh. Di Mascio et al.[23] also used overset grids, but this time for both the rudder and every single propeller blade, resulting in a total of six overset grids.

Rudder-propeller flows are not only simulated in isolation, but are an inherent part of any full ship computation. Zhirong et al.[24], for example, simulated a complex manoeuvre of a ship with two propellers using six degrees of freedom with six overset grids, one per propeller, one per rudder, one for the rest of the ship and finally a background one. Similarly, Carrica et al.[5] simulated the same vessel but this time combining a total of 37 meshes. Because the CFD solver used, CFDShip-IOWA, is limited to structured grids, even non-moving components had to be meshed independently and the overset grid method used to assemble them. Even though the overset grid method was designed in the early 1980[25], it has seen a broader adoption only in recent years helped by its wider availability in CFD packages (like Simcenter STAR-CCM+[26], FINE™/Marine[27], Ansys Fluent[2], openFOAM[15], etc.).

This goes to show the need for accurate but also predictable computation of rudder-propeller flows using the overset method. The lack of in-depth verification studies using the overset method and comparing different interpolation schemes motivated this study.

In this work, three interpolation schemes are tested: *Inverse distance*, a first order weighted average scheme, available in many overset implementations (Simcenter STAR-CCM+[26], Ansys Fluent[2], OpenFOAM[15] and SUGGAR++[28]); *Nearest cell gradient*, a scheme using gradients of a single *donor* cell to reach second order accuracy, and finally a *Least squares* scheme using a degree two polynomial function, a third order scheme. Besides analysing the differences of these schemes on integral quantities, mass imbalance, flow features and pressure distribution, a detailed solution verification study is performed using the methodologies from Eça et al.[29] and Brouwer et al.[30] to estimate the time and space discretisation uncertainties, and the iterative and statistical uncertainties. Performance considerations are also given to better understand the accuracy/performance trade-offs of any given interpolation scheme in an overset grid context. Finally, validation of the rudder flow is performed against a set of experiments done in similar conditions by Molland and Turnock[16]. The main objectives of this work are to inform decisions regarding the choice of interpolation schemes as well as qualifying and quantifying their impact on the accuracy of the simulation and quantities of interest.

This paper is structured as follows: after this introduction, the experimental and numerical setup are explained in section 2; preliminary solution verification is done in section 3; the impact of the different interpolation schemes is given in section 4, section 5 shows validation of the rudder flow results, and, finally, section 6 highlights the major conclusions of the work and some aspects to be further investigated.

2 | PROBLEM SETUP

2.1 | Experiments presentation

Molland and Turnock[31,16] conducted a series of experiments to study propeller - rudder interactions in various realistic ship manoeuvring conditions. Done in Southampton R.J. Mitchell's wind tunnel, the experiments included a centre board upstream of the propeller, the propeller itself and a rudder. Besides varying the rudder angle of attack, the entire assembly could be rotated to replicate different ship drift angles. Overall, air inlet speed, propeller rotation speed, rudder and drift angles were varied, and several centre board and rudder shapes were tested. The propeller used was a four bladed modified B4.40 Wageningen propeller, an 800 mm diameter, 0 degree rake and 0.95 mean pitch ratio propeller for which design details are found in Turnock et al.[32]. The rudder used in the present work is named rudder n°2 in the experiments description: a NACA0020 profile straight rudder with a chord of 667 mm and a span of 1000 mm. Finally, the long centre board, the one used in this work, is 2690 mm long and 1018 mm high. Load cells on the rudder assembly and propeller allowed the recording, amongst other quantities, of lift, drag and moment coefficients for the rudder and torque and moment coefficients for the propeller. Furthermore, for some runs, pressure was recorded on the rudder and centre board surfaces.

The present work focuses on replicating a subset of the input conditions done experimentally: a drift angle of -7.5 degrees, a propeller advance ratio of $J = 0.51$, an inlet velocity of 10 m/s with rudder angles of 10 and 20 degrees. This set of conditions was picked for their flow behaviour complexity and the larger amount of experimentally recorded quantities.

2.2 | Numerical setup

2.2.1 | ReFRESCO

The CFD computations are performed with ReFRESCO (REliable & Fast Rans Equations (solver for) Ships (and) Constructions Offshore)[33], a research and commercial CFD code optimised for maritime applications. It is being developed, verified and validated at MARIN (Maritime Research Institute Netherlands) in collaboration with several other organisations like IST (Instituto Superior Técnico in Lisbon, Portugal), the University of Southampton (Southampton, the UK) and blueOASIS (Lisbon, Portugal).

ReFRESCO is a viscous-flow CFD code that solves multi-phase (unsteady) incompressible flows using the Navier-Stokes equations, complemented with turbulence models, cavitation models and volume-fraction transport equations for different phases. The equations are discretised using a finite volume approach with cell-centered collocated variables, in strong-conservation form, and a pressure-correction equation based on the SIMPLE algorithm is used to ensure mass conservation[34]. Time integration is performed implicitly with first or second-order backward schemes. The implementation is face-based, which permits grids with elements consisting of an arbitrary number of faces (polyhedral, hexahedral, tetrahedral, prisms, pyramids, etc.), and if needed h-refinement (hanging nodes).

Moving, sliding and deforming grids, as well as automatic grid adaptation are available features. For turbulence modelling, both RANS/URANS and Scale-Resolving Simulations (SRS) models such as SAS, DDES/IDDES, XLES, PANS and LES approaches can be used.

The newly developed Overset grid method makes use of IBLANK information coming from the Suggar++ library [35,36] (defining the location of *in*, *fringe* and *hole* cells). The *donor* search as well as the interpolation itself are done by ReFRESCO with a wide set of interpolation schemes available, from 1st order *Inverse distance* to 4th order polynomial based *Least squares* methods. A detailed explanation of the overset and interpolation implementations can be found in Lemaire et al. [1].

2.2.2 | Grid and Overset setup

As stated in section 2.1, the numerical setup replicates the long centre board and rudder n°2 from Molland et al. To accommodate each component, three sets of meshes were generated, a first one containing the tunnel and centre board, then a cylindrical domain containing the propeller geometry and, finally, a body-fitted rudder mesh. With the overset method, the propeller can rotate and rudder can be set to any angle of attack without the need for re-meshing. Figure 1 shows the domain geometry and dimensions and Figure 2 defines the different angles used (angle of attack and drift angle) and shows the outline of the propeller and rudder domains. It should be noted that, compared to Molland et al.'s experiments, the propeller is rotating in the opposite direction, with corresponding change in drift and rudder angles sign conventions when comparing CFD with Experimental data.

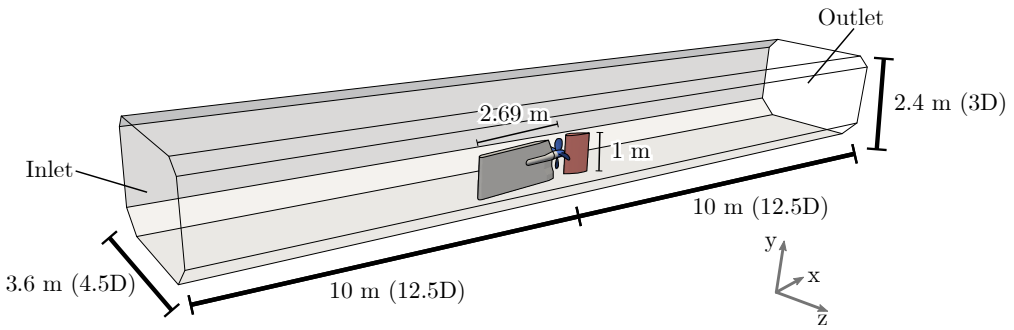


FIGURE 1 Computational domain dimensions replicating the cross section of the R.J Mitchel wind tunnel.

The tunnel mesh is fully structured, made with Pointwise [37]. Its cross section is at the R.J Mitchel wind tunnel's dimension, 3.6 m wide, 2.4 m high. It is also 20 m long with the rudder leading edge in the center of the domain. The centre board's y^+ was kept below 30, and refinements were made where the propeller and rudder would be positioned to have enough cells in the overlap region.

The propeller mesh, also fully structured, was generated using GridPro [38] with some custom preprocessing tools made by MARIN [39] using Rhino [40]. To ease the meshing, especially at the blade root, the geometry of the propeller itself does not exactly replicate the shape used in the experiments. Instead of the modified B series propeller, the CFD analysis was done with a conventional Wageningen B4.40 [41] with a constant pitch ratio of $P/D = 0.95$. Figure 3 compares both geometries. Because of this difference, validation is performed for K_T-equivalent computations to produce a similar flow wake on the rudder. The detailed procedure is explained in section 5. y^+ is kept below three on the blades with a mean y^+ below 1, and below 100 on the hub. The mesh diameter of 1 m or $1.25D$ allows for sufficient clearance at the blade tip for overset *fringe* cells.

Finally, the rudder mesh, also fully structured, was generated with Pointwise. It is an O-grid that extends above the rudder tip. Refinements at the leading and trailing edges were made and a y^+ below 1 was maintained.

The original tunnel mesh was designed without drift angle with the centre board aligned with the tunnel inlet-outlet. To prevent remeshing meshes with drift angles were generated using deforming grids by rotating the centre board part of the tunnel mesh.

For each component, a set of geometrically similar meshes with different refinements was generated, Table 1 summarises the different cell counts. As seen in Figure 4 and 5, the overset cell status generated by Suggar++ places two layers of *fringe* cells

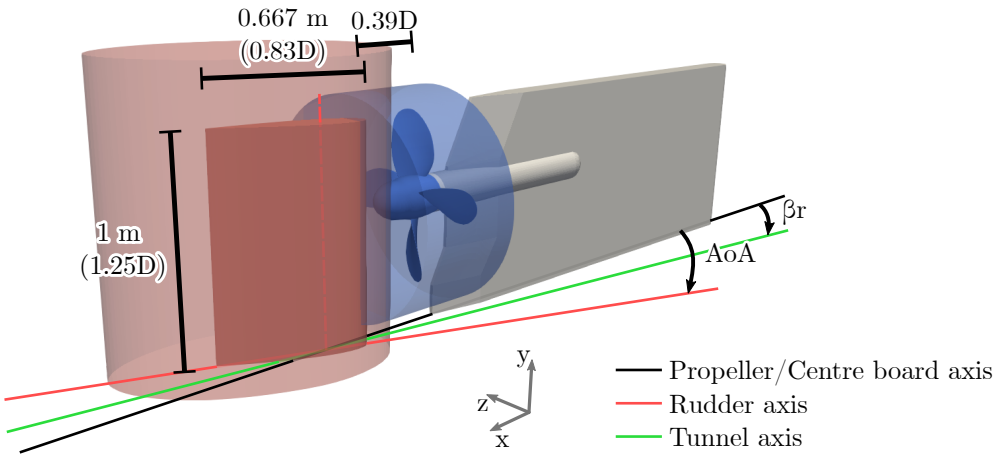


FIGURE 2 Definition of the different coordinate and angle systems. AoA is the rudder angle of attack while βr defines the drift angle of the assembly.

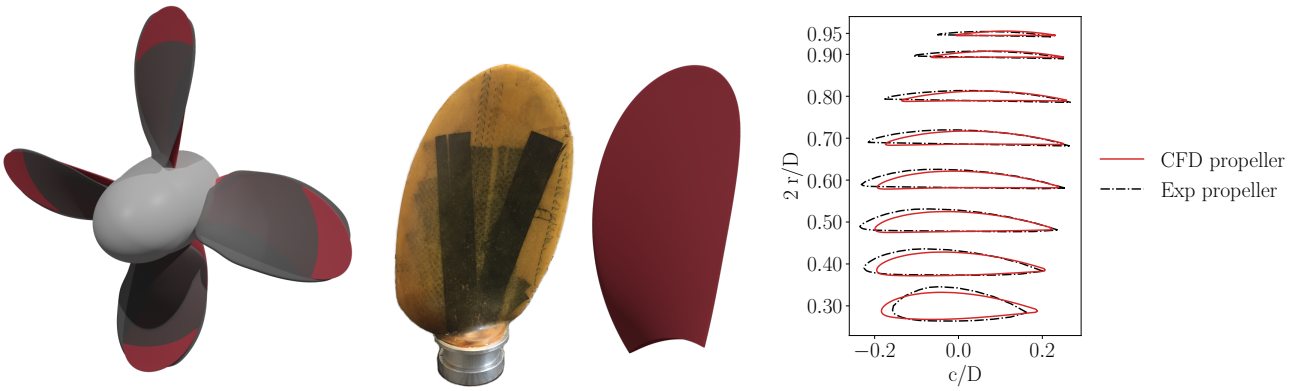


FIGURE 3 Comparison of the propeller blade shapes used during the experimental campaign[16] (black) and in this CFD study (red).

TABLE 1 Refinements ratio (h_i) and cell counts for the different meshes generated.

	h_i	Total	Tunnel	Propeller	Rudder
G1	1.51	$10.8 \cdot 10^6$	$6.6 \cdot 10^6$	$3.2 \cdot 10^6$	$1.0 \cdot 10^6$
G2	1.36	$14.8 \cdot 10^6$	$9.8 \cdot 10^6$	$3.4 \cdot 10^6$	$1.6 \cdot 10^6$
G3	1.21	$21.4 \cdot 10^6$	$13.1 \cdot 10^6$	$6.2 \cdot 10^6$	$2.1 \cdot 10^6$
G4	1.0	$37.4 \cdot 10^6$	$23.1 \cdot 10^6$	$10.8 \cdot 10^6$	$3.6 \cdot 10^6$

distributed on each side of the interface. Moreover, *hole* cells are placed in the propeller mesh (due to the rudder overlap) and in the tunnel mesh where the propeller and rudder meshes are positioned.

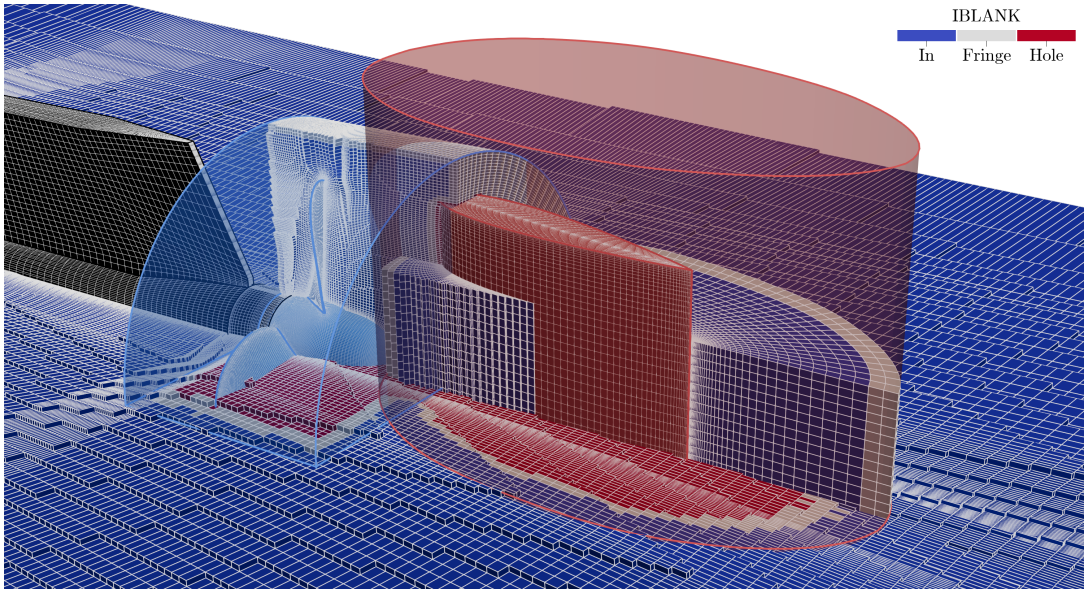


FIGURE 4 3D view of the different meshes colored with IBLANK information for the coarsest assembly G1. The wireframe of each mesh is also colored differently with the tunnel in white, the propeller in blue and the rudder in red.

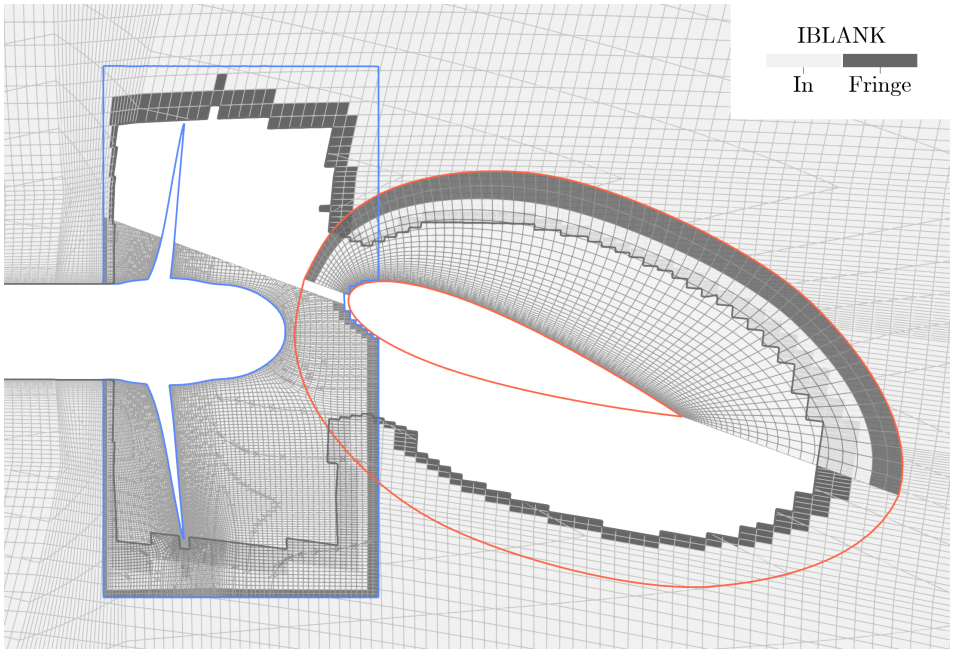


FIGURE 5 Top view of the different meshes showing IBLANK information for the coarsest grid assembly G1. Only the lower two thirds of the propeller (blue) and top half of the rudder (red) meshes are displayed to reveal the tunnel mesh in the background.

2.2.3 | Computational setup

In this study, turbulence modeling is done with the $k - \sqrt{k}L$ two equations turbulence model[42], chosen for its better robustness compared to $k - \omega$ SST models. Momentum and pressure correction equations are coupled using the SIMPLE methods and all equations are solved in a segregated way. Convective fluxes for the momentum and turbulence equations are discretised by the second order QUICK scheme (with limiters), and a three-time level scheme is used for second order in time discretisation.

The side walls of the tunnel are modeled by a slip boundary condition while rudder, propeller and centre board surfaces are modeled using non-slip wall boundary condition. Automatic wall functions are used, where for $y^+ < 5$ (viscous-layer) a full-resolved boundary layer treatment is employed, for $y^+ > 30$ (log-layer) wall-functions are used, and in-between (in the buffer layer) a blending between the two approaches is used. At the outlet a pressure boundary condition is used and the inflow velocity ($U_\infty = 10\ m/s$) is set at the inlet of the tunnel (aligned with the tunnel itself) together with an eddy viscosity ratio of 0.01.

Three interpolation schemes are used for the overset grid method interpolation. First order *Inverse distance*, second order *Nearest cell gradient*, and a third order polynomial-based *Least squares* schemes. The *Inverse distance* scheme is a weighted average based on the distance between *donor* and *fringe* cells that uses, in this work, seven *donor* cells per *fringe* cell as it is common to use a cell and all its direct neighbours. The *Nearest cell gradient* uses a single *donor* cell but is second order accurate due to the use of the gradient at the *donor* cell center. Finally, the *Least squares* scheme employed in this work is third order accurate. It uses 25 *donor* cells per *fringe* cell to over-determine the system of linear equations and help robustness. Detailed information about each scheme is given in Lemaire et al.[1].

2.2.4 | Analysed quantities

In this study, integral quantities related to both the propeller and rudder are assessed, which are defined in the following equations,

$$J = \frac{U_\infty}{nD}, \quad K_T = \frac{T}{\rho n^2 D^4}, \quad K_Q = \frac{Q}{\rho n^2 D^5}, \quad C_L = \frac{L}{\frac{1}{2}\rho U_\infty^2 c s}, \tag{1}$$

$$C_D = \frac{D}{\frac{1}{2}\rho U_\infty^2 c s}, \quad C_m = \frac{M_y}{\frac{1}{2}\rho U_\infty^2 c^2 s}, \quad C_{pc} = 100 \times \left(\frac{C_m}{\sqrt{C_L^2 + C_D^2}} + \frac{0.2}{c} \right). \tag{2}$$

J is the propeller's advance ratio, K_T its thrust coefficient and K_Q its torque coefficient. Concerning the rudder, C_L is the lift coefficient, C_D the drag coefficient, C_m its moment coefficient, here calculated at 30% chord and C_{pc} the location of the center of pressure along the chord. Also, U_∞ is the inlet velocity ($U_\infty = 10\ m/s$), n the propeller rotation speed in revolution per second, D the propeller diameter ($D = 0.8\ m$), ρ the air density ($\rho = 1.225\ kg/m^3$), c the chord length ($c = 0.667\ m$) and s the rudder span ($s = 1\ m$).

For each quantity, time averaging is done over an integer number of blade passing periods and statistical uncertainty is quantified in section 3.3 using Brouwer et al.[43,30] methodology. Time-averaged velocity and pressure fields are also recorded.

Finally, given that the overset grid method implementation used in this study does not guarantee total mass flux conservation, monitoring the mass imbalance is of particular importance. To this end, signed mass fluxes between *fringe* cells and *in* cells are summed and recorded over time for each mesh individually, thereby effectively measuring the mass imbalance on each mesh. The cells where the summation is performed are shown in Figure 6. In total four quantities are analysed: total mass fluxes going in and out of the domain via the inlet and outlet (Q_{total}), and fluxes going through each of the overset 'interfaces' on the tunnel (Q_{tunnel}), propeller (Q_{prop}) and rudder (Q_{rudder}) meshes. One should note that, beside overset grid interpolation, mass fluxes are governed by the pressure correction equation, hence a non-zero value is a marker of overset grid interpolation errors and/or iterative errors for that particular equation.

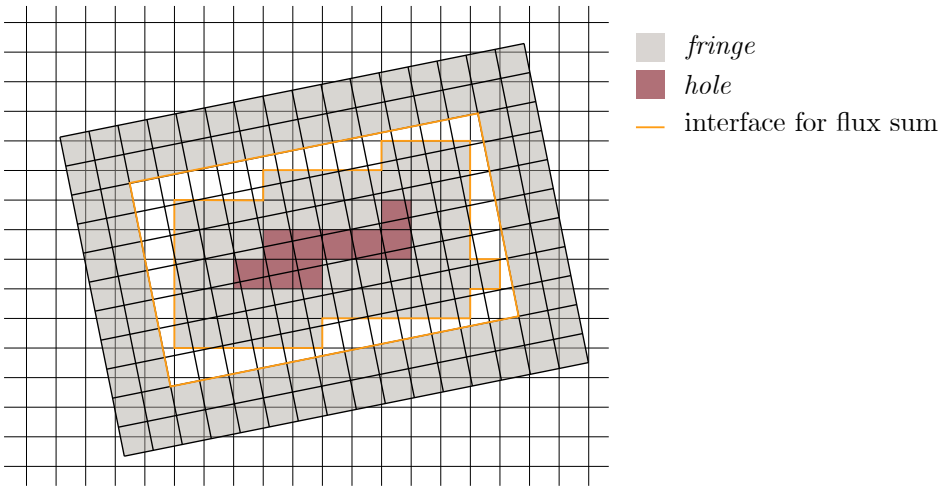


FIGURE 6 Overset meshes schematic highlighting (in orange) the faces where signed mass fluxes are being summed up to compute mass imbalance.

3 | VERIFICATION STUDIES

3.1 | Iterative uncertainty

Iterative uncertainty quantification has been done following the method presented in Eça et al.[44,29]. For this, a set of five computations on grid G2 with the rudder at 10 degrees angle of attack was done with a varying number of outerloops per time step of 25, 50, 75, 100 and 150. Each computation resulted in different residuals iterative convergence, and Figure 7 shows that, for each equation solved, the time-averaged residual level at the end of each time step depending on the number of outerloop per time step. First, apart from the pressure correction equation (denoted P), all the equations are not affected by the number of outerloops per time step. This is mainly because their convergence stall within the first 25 outerloops each time step.

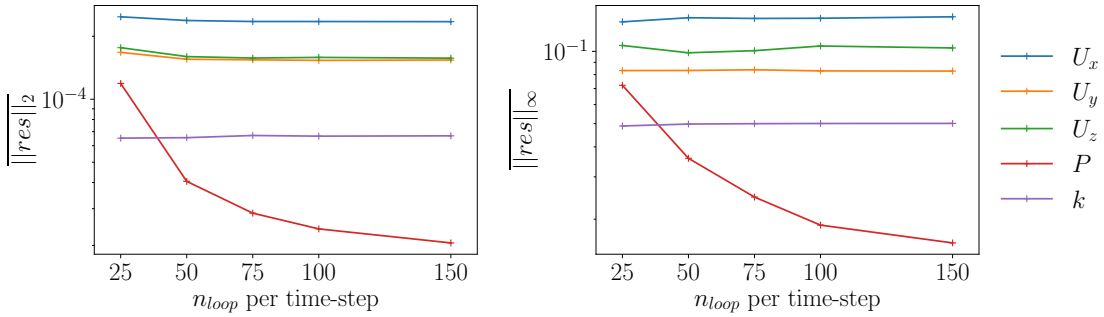


FIGURE 7 Time-averaged L_2 and L_∞ residuals for an increasing number of outerloops (n_{loop}) per time step. The second turbulence equation residuals are omitted for clarity, but they are relatively constant and three orders of magnitudes lower than the turbulence kinetic energy (k) residuals.

Following Eça et al.[44], a least squares fit is performed on the quantities of interest against their relative residuals levels. In this study, the L_∞ norm of residuals levels for the pressure correction equation were selected as they are the only one affected by the number of outerloops. The proposed fit function is:

$$\phi(\epsilon_i) = \phi_0 + \alpha e^{F(\epsilon_i)\beta}, \tag{3}$$

with ϕ the quantity of interest and ϵ_i the residuals level achieved. Both functions $F(\epsilon) = ln(\epsilon)$ and $F(\epsilon) = -1/\epsilon$ were tested and the first one is selected for its better fit to the data. Results are shown in Figure 8 for the propeller's K_T and rudder's C_L.

A converging trend is clearly achieved for the residuals as a function of the number of outerloops, and iterative uncertainty can be computed. In the rest of this work, 75 outerloops per time step will be used as a balance between iterative error (here its uncertainty is estimated at 0.2% for K_T and 0.02% for C_L) and computational cost. It has to be noted, however, that because only the pressure correction equation residuals are reduced with more outerloops, this study only partially evaluates iterative error. In fact, reduction of all the other residuals would be needed to fully assess the iterative uncertainty. Moreover, iterative error is here studied on a single mesh, but Eça et al.[44] showed that iterative error can influence the study of discretisation errors. Nonetheless, given the relatively low uncertainties found for integral quantities further study was considered outside of the scope of this work.

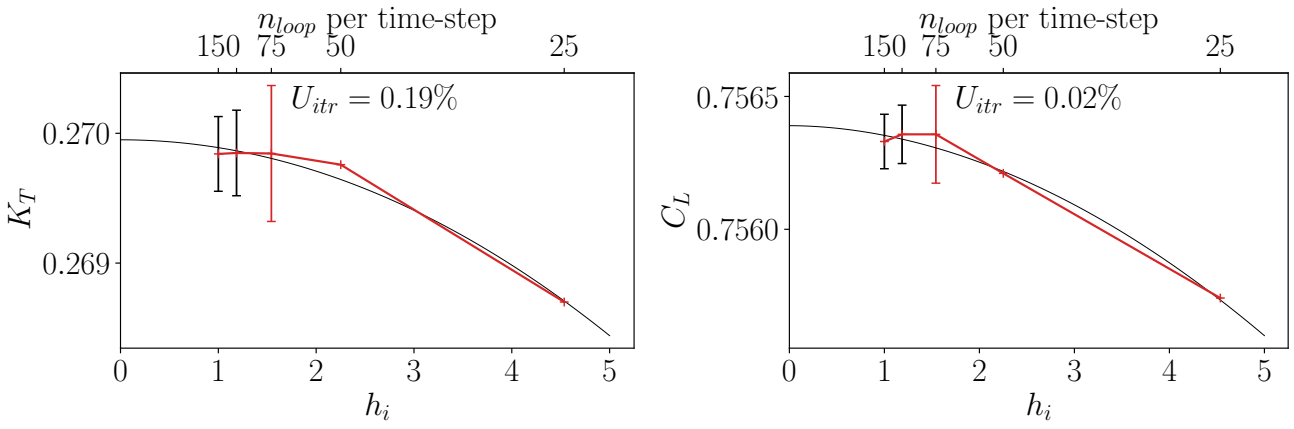


FIGURE 8 K_T and C_L convergence with the pressure correction residuals, $h_i = \frac{\overline{||P_{res\ i}||_\infty}}{||P_{res\ 0}||_\infty}$. Bars show the iterative uncertainty and in red the one corresponding to the number of outerloops selected for the rest of this work.

3.2 | Time discretisation uncertainty

To assess time discretisation errors, a set of five computations with varying time steps was used (see Table 2). The number of time steps ranged from 400 per propeller revolution to 50, leading to an average Courant number from 0.7 to 5. For computational resources reasons, this test has been performed on mesh G2 only even though some variations should be expected with finer meshes and the increase of Courant number. Figure 9 plots K_T and C_L against time step refinements, each quantity showing convergence trends. Using Eça et al.[29] method, the time step uncertainty on K_T is estimated at 0.15% when using 1.8° rotation per time step, and 2% for C_L. This time step is selected to be used in the rest of the current study. One should note, however, that for the finest mesh assembly (G4) this choice leads to an average Courant number of around 2.2 which would result in similar K_T uncertainty but, likely, a higher C_L uncertainty, looking at the trend of each convergence plot.

TABLE 2 Time steps tested to compute time discretisation uncertainty. The rightmost two columns show the time step compared to the propeller rotation speed ($J = 0.51$ leading to 1460 RPM).

h_i	t_{step}	$\overline{CFL}$	t_{step}/rot	$°/t_{step}$
1.00	$1.027 \cdot 10^{-4}$ s	0.66	400	0.9°
1.33	$1.370 \cdot 10^{-4}$ s	0.89	300	1.2°
2.00	$2.055 \cdot 10^{-4}$ s	1.33	200	1.8°
2.67	$2.740 \cdot 10^{-4}$ s	1.77	150	2.4°
4.00	$4.110 \cdot 10^{-4}$ s	2.66	100	3.6°
8.00	$8.219 \cdot 10^{-4}$ s	5.32	50	7.2°

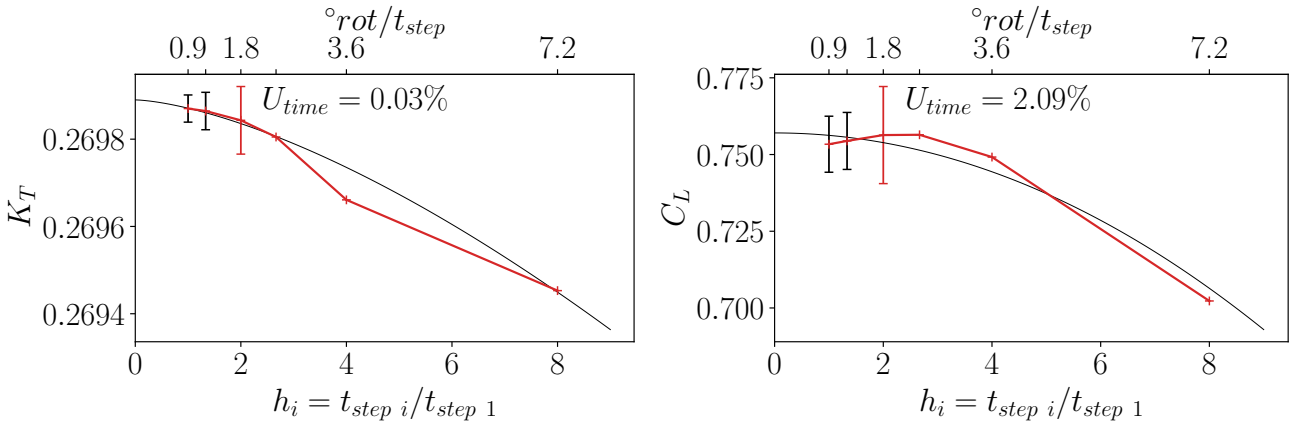
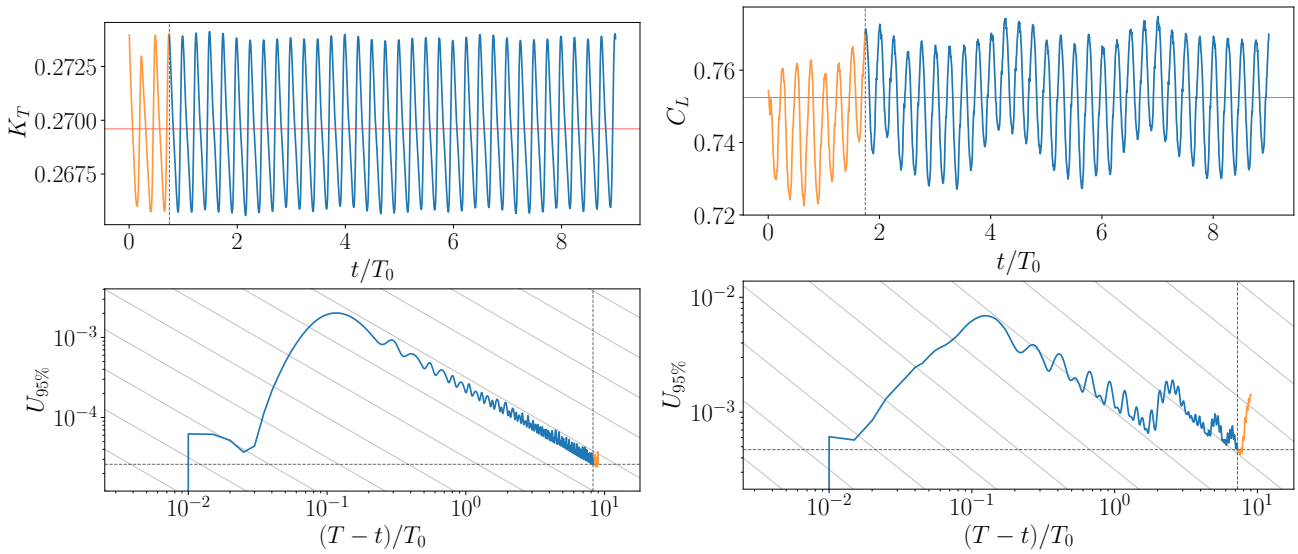


FIGURE 9 K_T and C_L convergence with time step refinement. The error bars show time discretisation uncertainties and are computed using Eça et al.[29] methodology.

3.3 | Statistical uncertainty

To speed up simulations, each computation presented in this work is initialised from a fully developed computation on a coarser mesh. After this initialisation, each computation is run for at least nine propeller rotations. The transient scanning technique's[30] implementation by Lemaire et al.[45] is then used on each quantity to detect and discard the transient portion of the results and compute the statistical uncertainty. As shown in Figure 10, computations done with 10 degrees rudder angle converge quickly to a steady state and show a clear oscillatory behaviour in sync with the blade passing frequency (4 times per rotation). Rudder forces (like C_L) additionally display secondary oscillatory behaviour about 10 times larger than the blade passing one (or 2.5 propeller period). As shown in Figure 10, the statistical uncertainty is below 0.5% for this set of computations and analysed



(a) $\overline{K_T} = 0.26960 \pm 2.6 \cdot 10^{-5}$ **(b)** $\overline{C_L} = 0.7525 \pm 4.7 \cdot 10^{-4}$

FIGURE 10 Propeller (K_T) and rudder (C_L) forces coefficients time histories with the rudder at 10 degrees angle of attack. The statistical uncertainty on the mean is computed using the transient scanning technique[43] and shown on the bottom plot for each quantity. The transient portion (orange) is removed from the computation of the mean. T_0 is the rotation period of the propeller and T is the simulation time.

quantities. While not shown here, this conclusion holds for the other mesh refinements, overset grid interpolation schemes and quantities.

When setting the rudder at 20 degrees, this behaviour changes for grid assemblies G1 to G3 (the coarsest ones) as they introduce an extra long transient phase lasting more than 40 propeller periods. As this transient portion needs to be simulated but discarded this set of computations was deemed too expensive to be computed. As a result, at 20 degrees rudder angle, only the results for mesh G4 (the finest one) will be shown and discussed. These were run for 12 propeller rotations and lead to similar statistical uncertainties as the 10 degrees rudder angle ones.

4 | IMPACT OF INTERPOLATION SCHEMES

4.1 | Integral quantities

As seen in section 3.3, propeller forces coefficients K_T or K_Q have a clear oscillatory behaviour in sync with the blade rotations. These oscillations are also consistent over time, likely because the propeller inlet, even in the wake of the angled centre board, is steady. The results obtained when using the three interpolation schemes tested share this characteristic, and, overall, have very close behaviour as seen in Figure 11. Rudder forces, however, exhibit larger differences and seem less smooth. Being in the wake of the propeller and having the flow cross at least one more overset interface brings more time variation and greater changes between the different interpolation schemes. The *Inverse distance* show up to 1% difference on C_L and 2.5% on C_D compared the *Least squares* computation, but only 0.05% on K_T. The same trend is also true for the *Nearest cell gradient*, albeit with smaller differences. Apart from different amplitudes and general shape of the oscillation, it appears that using the *Inverse distance* scheme results in 'noisier' forces, once again, specially on the rudder lift and drag.

To quantify these differences, and in particular the spurious oscillations between the interpolation schemes, a spectral analysis is shown in Figure 12. First, doing a Fourier transform on the forces coefficient time histories confirms that, overall, propeller force coefficients have less higher-frequency harmonics than rudder coefficients. Moreover, the high frequencies are mainly harmonics of the blade passing frequency. Beside having more harmonics, the rudder quantities have higher white noise levels

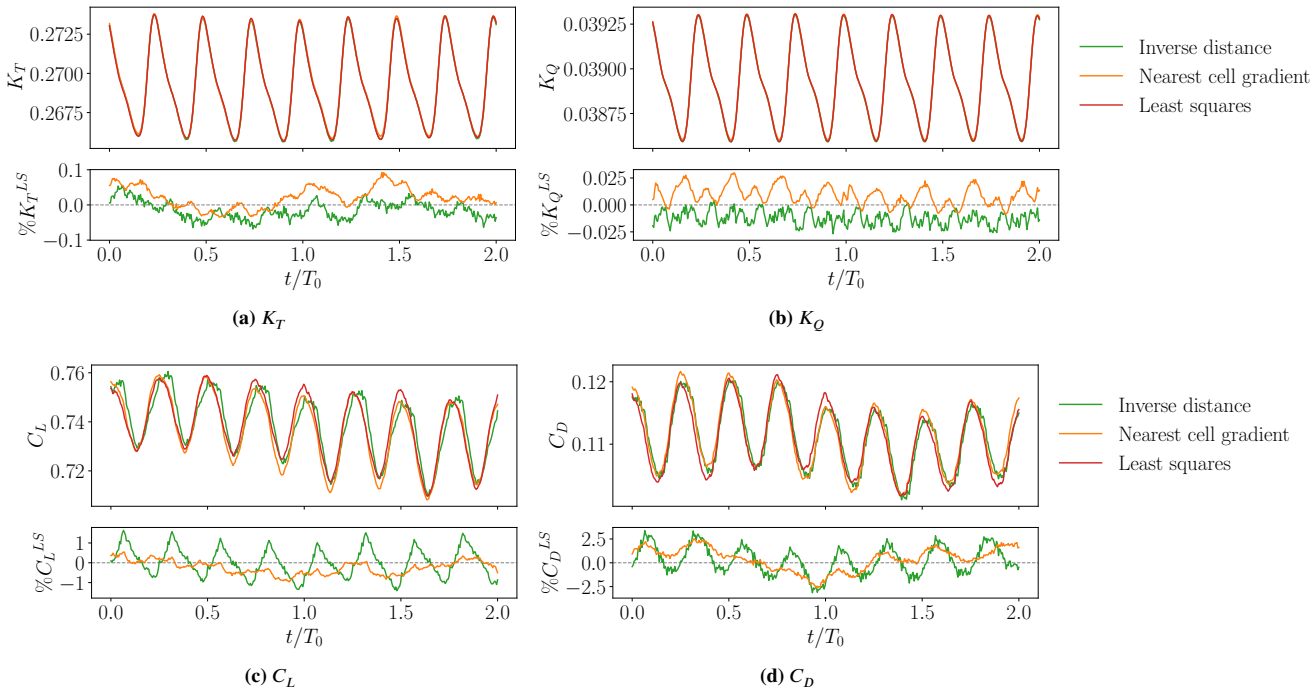
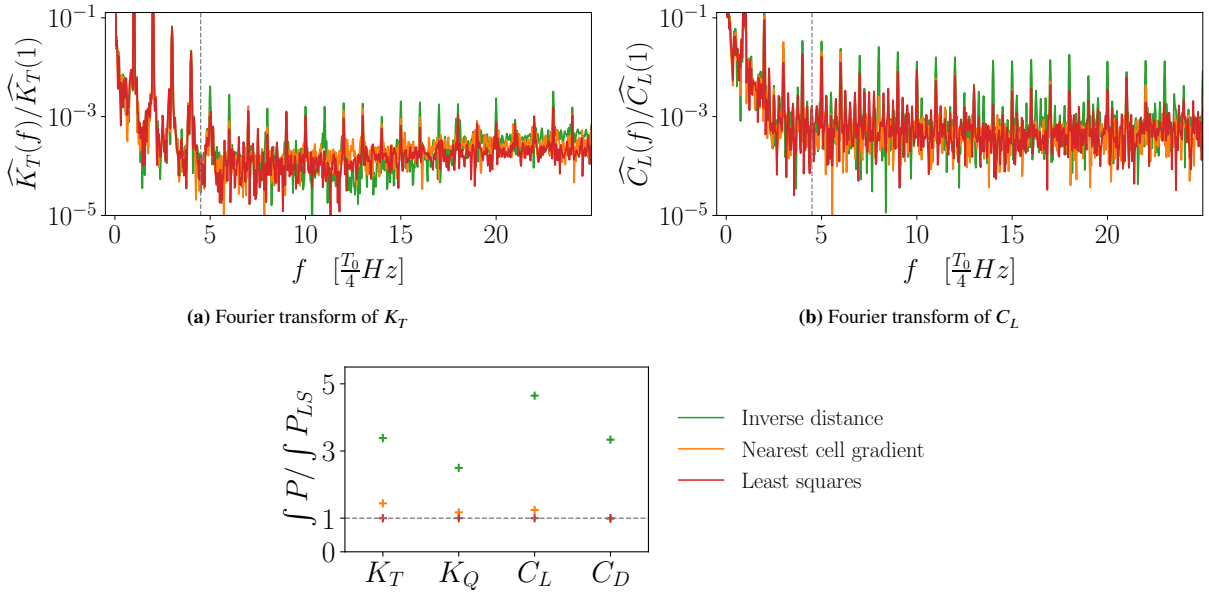


FIGURE 11 Force coefficients time history for the three interpolation schemes tested at 10 degrees rudder angle. The bottom plot is a comparison with the *Least squares* scheme results, computed with: $\%\phi^{LS} = 100 \cdot \frac{\phi - \phi^{LS}}{\phi^{LS}}$



(a) Fourier transform of K_T

(b) Fourier transform of C_L

(c) Power spectral density of high frequencies (higher than 4.5 BPF) normalised by the *Least squares* computation ones.

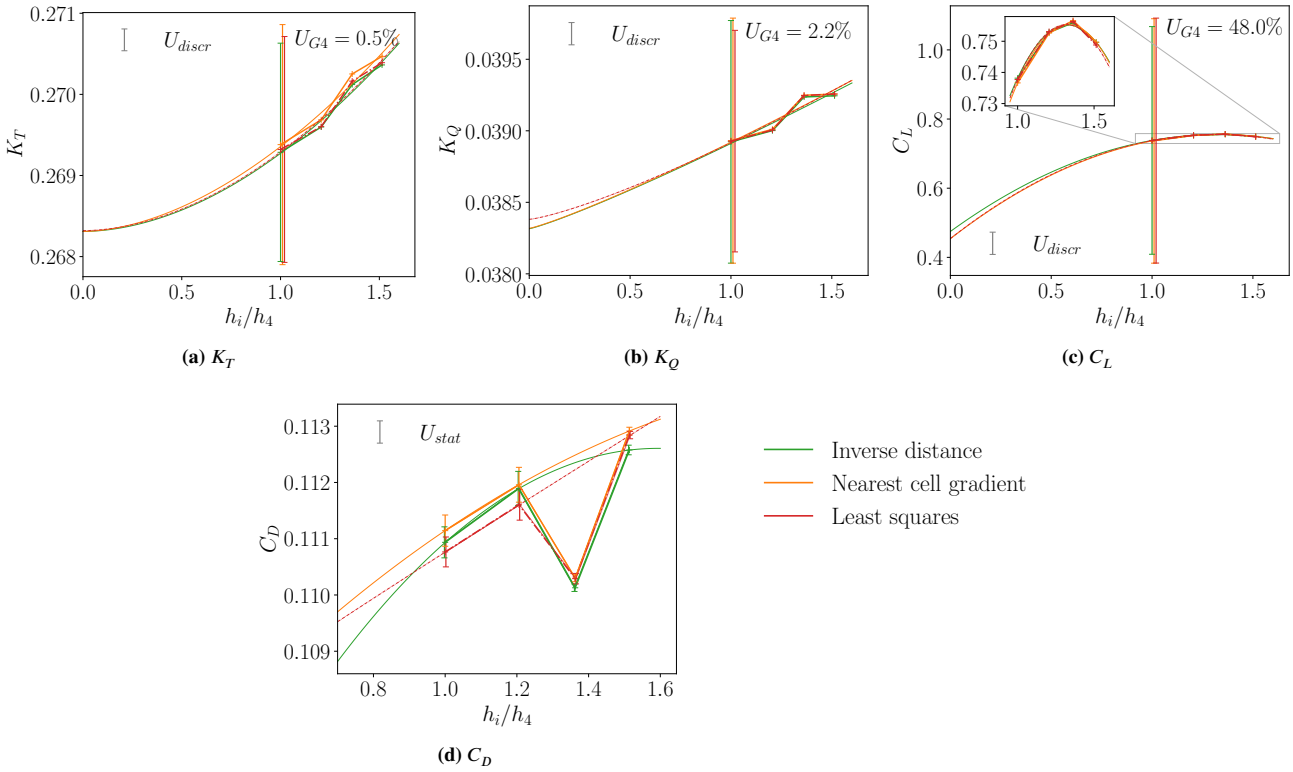
FIGURE 12 Fourier transforms of K_T and C_L. Frequencies are normalised by the blade passing frequency ($4/T_0$) and spectra by the level of the first harmonic (corresponding to the blade passing frequency). Plot 12c compares the integration of the power spectra for high frequencies (higher than 4.5 times the blade passing frequency, denoted with the vertical dashed line on the Fourier transform plots).

too. Differences between the interpolation schemes can also be observed as the *Inverse distance* computation appears to have more harmonics than the other two tested, as well as overall higher intensities outside of harmonics.
Computing the power spectral density of high frequencies (higher than 4.5 times the blade passing frequency) highlights the energy stored in these high frequencies and is here used as a measure of the overall 'smoothness' of the signals. Such quantity, normalised by the *Least squares* density, is plotted in Figure 12c. It shows the *Nearest cell gradient* computation having very similar levels to the *Least squares* one, but the *Inverse distance* computation, regardless of the quantity, to have two to five times more energy in these high frequencies. These results are here displayed for 10 degrees angle of attack on grid G4 but are similar to the 20 degrees rudder angle or other grid refinements. The reason this noise is higher with *Inverse distance* scheme is likely coming from its lower overall accuracy. Indeed, larger interpolation errors come with higher variations around the exact interpolated value, which, applied to every single interpolated value on *fringe* cells every time step, leads to larger noise levels.

Time averaging the force coefficient eliminates these high frequencies and leads to very similar results for the three interpolation schemes. This is observed in Figure 13, where each quantity is plotted against grid refinement. K_T, for example, tends to be only marginally over-predicted on the *Nearest cell gradient* results compared to the other two schemes but with a difference of less than 0.03% it can be considered as an insignificant change in an engineering context. For propeller forces, the convergence with grid refinement is clear and monotonic which leads to low uncertainty of 0.5% and 2.2% for K_T and K_Q respectively. In comparison, the rudder forces show non-monotonic behaviour with less clear convergence trends. On C_L the 'bell shaped' convergence leads to 48% uncertainties. C_D, on the other hand, has a different behaviour. In fact, in figure 13d, the statistical uncertainty is this time plotted as it is of similar magnitude to the mesh refinements changes, which then confounds the fitting and computation of the discretisation uncertainty. Even though the three schemes show very close trends, the uncertainty varies from 0.5% to 11% and this results in very different fitting functions. This issue could be solved by either running the simulation for longer to decrease the statistical uncertainty, or by using more meshes with different refinements to make the fit less influenced by small variations[1].

[1]Eça et al. methodology is known to be pessimistic and highly sensitive to small spread of refinements and small number of refinement levels.

Overall, this shows that the spurious oscillations shown in the previous section mainly lead to zero mean noise, and do not lead to sustained changes to forces as the three interpolation schemes result in very similar time-averaged values. Also, propeller forces show monotonic 2^{nd} order converging trends whereas rudder ones are non-monotonic. The discretisation uncertainty estimation, however, is one order of magnitude larger than the difference between the three schemes suggesting that the discretisation error is higher than the interpolation error made on overset interfaces. This also explains why computations done with the 1^{st} order *Inverse distance* scheme manage to maintain an overall 2^{nd} order convergence on propeller related integral quantities, as overset related interpolation errors are negligible compared to the discretisation errors, even on the finest sets of grids.

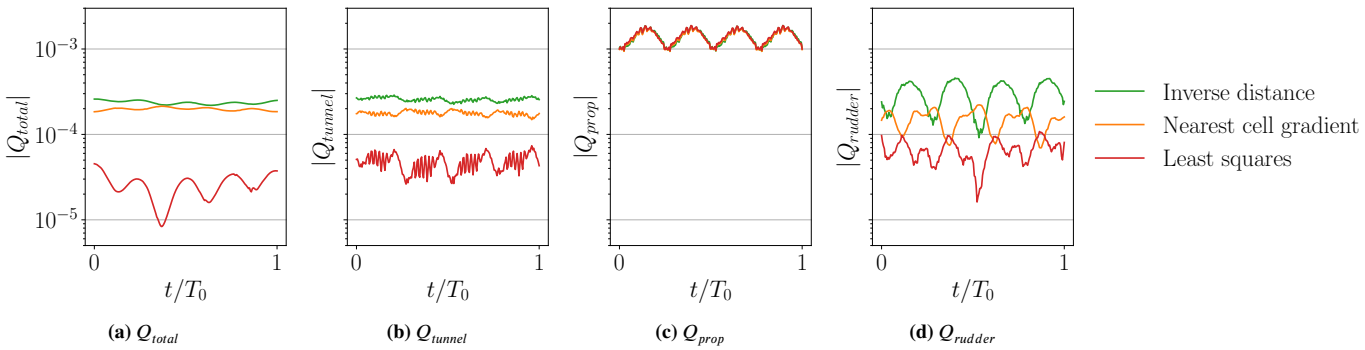


(a) K_T **(b)** K_Q **(c)** C_L **(d)** C_D

FIGURE 13 Time average force coefficients against grid refinement for the three interpolation schemes tested at 10 degrees rudder angle. For each of them, except C_D, discretisation uncertainty (U_{discr}) is displayed. For C_D, statistical uncertainty (U_{stat}) is shown instead.

As explained in section 2.2.4, mass fluxes are not theoretically conserved by the overset grid method and consequently a mass imbalance is introduced and influenced by the interpolation scheme. Figure 14 shows the sum of mass fluxes going through each mesh over time and for each scheme. First, once again, the blade passing frequency is discernible on each of the quantities, even though it is not the only frequency component. The comparison of the different schemes shows that, for the *total* mass imbalance (Figure 14a), the *Least squares* scheme shows results an order of magnitude lower than the other two, but they all share similar oscillation amplitudes (albeit not obvious due to the log scale). The *tunnel* interface mass imbalance has, then, levels similar or slightly higher than the *total* ones but with extra frequencies components. Next, the *prop* interface shows a drastically different behaviour with higher flux imbalance and only minor differences between the interpolation schemes. Finally, the *rudder* mesh has a behaviour close to the *tunnel* one with, however, larger oscillations amplitudes.

The difference in smoothness and higher frequency components between the *total* mass imbalance and those on the *tunnel* or *rudder* is explained by the direct proximity of the latter two to *fringe* cells. The inlet and outlet of the domain are further away from overset meshes, which means that the oscillations are dampened due to numerical diffusion and iterative errors. The higher imbalance seen on the propeller mesh is either related to the iterative error, the highest pressure correction residuals



(a) Q_{total} **(b)** Q_{tunnel} **(c)** Q_{prop} **(d)** Q_{rudder}

FIGURE 14 Sum of fluxes going through each overset interface (Q_{tunnel}, Q_{prop} and Q_{rudder}) and by the inlet and outlet of the domain (Q_{total}) for the three interpolation schemes tested on the finest mesh (G4) and at 10 degrees rudder angle. Fluxes are normalised by the inlet mass flux.

being close to the blade tips, or due to the fact that the propeller mesh is the only one moving. Regardless of the reason, the error source driving this phenomenon appears to be different from the one seen on the other three quantities as it is not influenced by the interpolation scheme, it has a different level and it is not influenced in the same way by mesh refinements (not shown here). The interpolation made on the propeller mesh still deteriorates the mass imbalance, but its effect is overshadowed by an other error source. Overall, it is, however, not clear how and if errors made on fluxes propagate through an overset interface. Here, for example, the high flux imbalance of the propeller mesh does not seem to affect the other two ones. One beginning of explanation resides with the fact that fluxes are computed on cell faces while the interpolation uses exclusively cell centre information partially dissociating the different meshes.

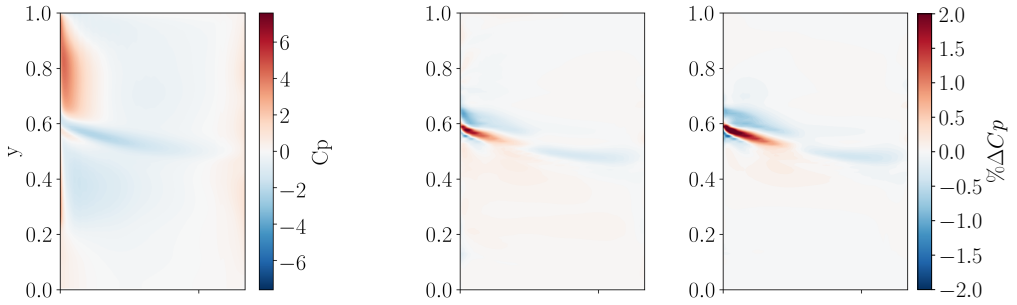
4.2 | Pressure on the Rudder

The time-averaged pressure coefficient on the rudder surface is shown in Figure 15 at 10 degree rudder angle and in Figure 16 at 20 degrees. In both figures, the *Least squares* results are displayed in the first column, and the subsequent two plot differences with *Inverse distance* and *Nearest cell gradient* respectively. Since the height or span of the rudder is $s = 1m$, no distinction will be made between the height of a section and its span-wise ratio (y/s) in this section.

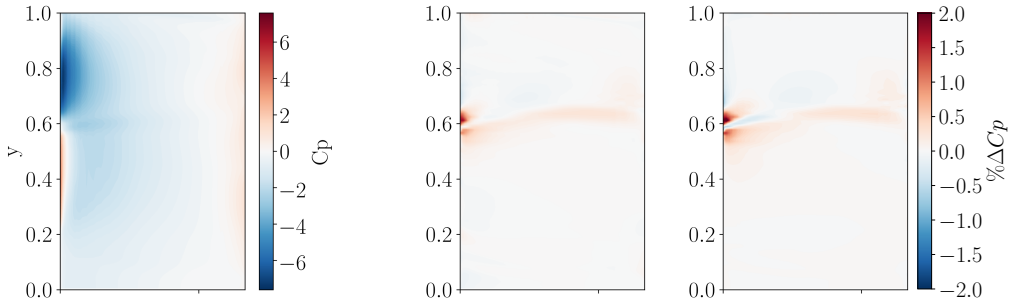
At both 10 and 20 degrees, from 0.2 m to 1 m height, the leading edge of the rudder sees high and low pressure regions directly related to the propeller rotation and at 0.6 m height, and the impact of the hub vortex is clearly visible and creates a lower pressure trail. On the suction side, this trail goes down due to the high pressure region above it and goes up on the pressure side, again due to the higher pressure region this time in the lower part of the rudder. Being in the race of the propeller, the high pressure regions have levels higher than the stagnation pressure. Moreover, as expected, a higher rudder angle leads to higher pressure differences between the two sides of the rudder, generating more lift.

When comparing the different interpolation schemes the main variations are in the hub vortex region with at most 3% differences at 10 degrees angle of attack. In comparison, at 20 degrees, differences are lower with maximum around 1% or lower. Surprisingly, however, in both cases, the *Inverse distance* results look closer to *Least squares* than *Nearest cell gradient* are. These differences, being very localised and of relatively small magnitude, have likely a marginal contributions to the overall pressure lift and drag. For a finer comparison, line plots along the rudder's chord are shown in Figure 17. They highlight the fact that, outside the hub vortex region, *Least squares* results are actually closer to *Nearest cell gradient* ones than they are to *Inverse distance*, though, again, the overall small differences explain the minimal variation in lift and drag observed in the previous section.

Finally, similarly for the integral quantities, using the four sets of meshes, space discretisation uncertainties can be computed for the rudder's pressure distribution. Figure 18 displays it at several slices and on the rudder's pressure side. On average, the uncertainty is 0.6 Cp overall, but it is a lot higher in the hub vortex region. This means that, regardless of the interpolation scheme, a reduction of cell sizes in this region would be needed to reduce discretisation errors, and doing so would likely contribute to lowering the lift and drag errors as well. Finally, the differences between the schemes may also be reduced in this region with further grid refinements.

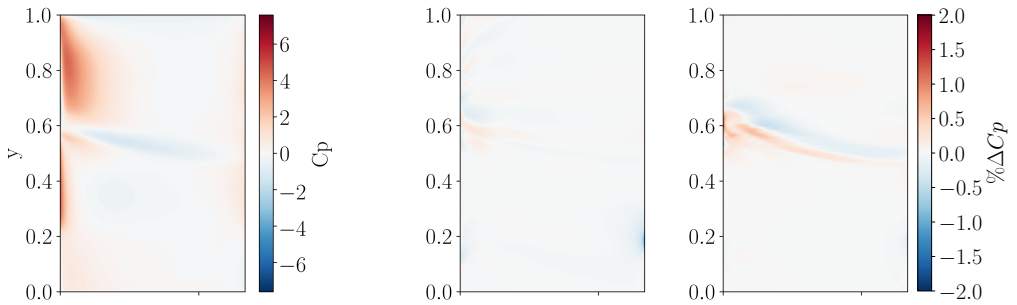


(a) *Least squares* (Cp_{LS}), pressure side **(b)** $Cp_{InvDist} - Cp_{LS}$ **(c)** $Cp_{NCGrad} - Cp_{LS}$

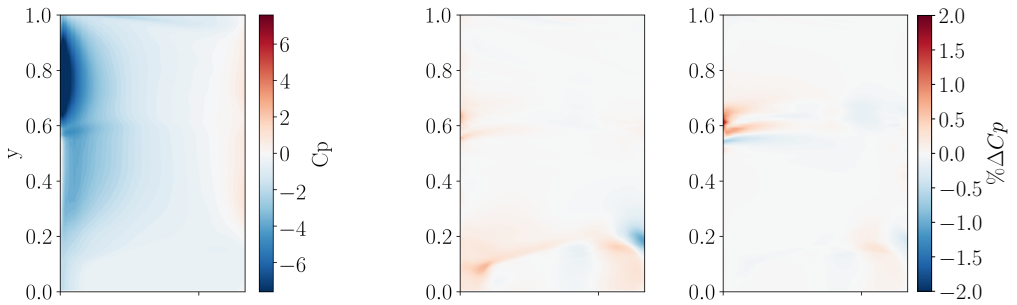


(d) *Least squares* (Cp_{LS}), suction side **(e)** $Cp_{InvDist} - Cp_{LS}$ **(f)** $Cp_{NCGrad} - Cp_{LS}$

FIGURE 15 Pressure coefficient on the rudder's surfaces at 10 degrees angle of attack for grids G4. The leading edge is shown on the left of the frame. The first column shows *Least squares* results while the other two display its difference with *Inverse distance* and *Nearest cell gradient* respectively, normalised by the amplitude of Cp over the rudder surface.

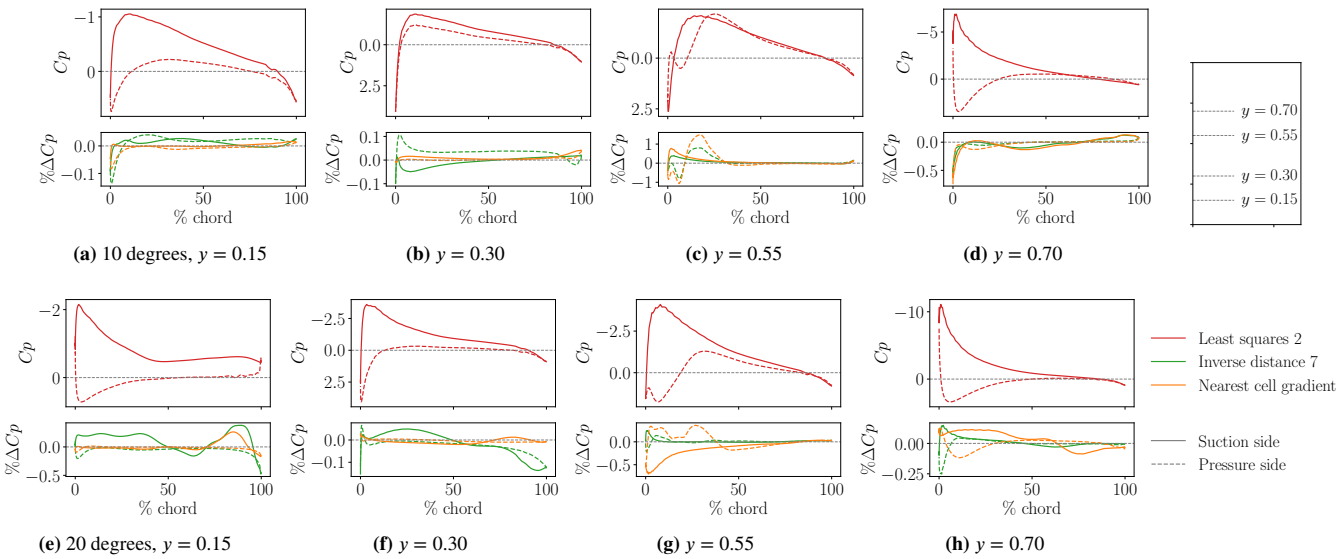


(a) *Least squares* (Cp_{LS}), pressure side **(b)** $Cp_{InvDist} - Cp_{LS}$ **(c)** $Cp_{NCGrad} - Cp_{LS}$



(d) *Least squares* (Cp_{LS}), suction side **(e)** $Cp_{InvDist} - Cp_{LS}$ **(f)** $Cp_{NCGrad} - Cp_{LS}$

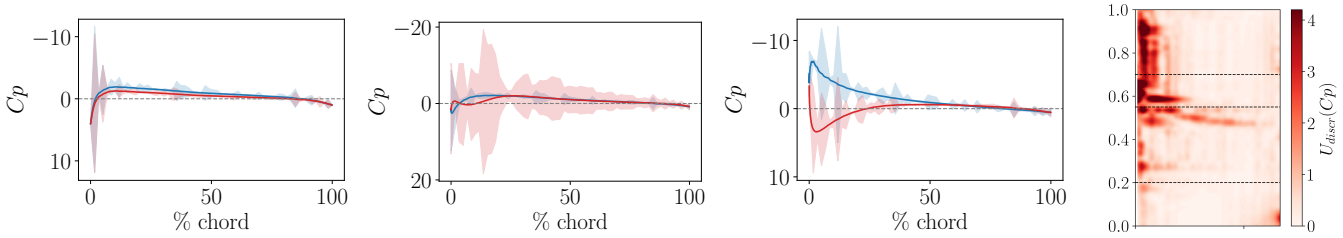
FIGURE 16 Pressure coefficient on the rudder's surfaces at 20 degrees angle of attack for grids G4. The leading edge is shown on the left of the frame. The first column shows *Least squares* results while the other two display its percentage of difference with *Inverse distance* and *Nearest cell gradient* respectively, normalised by the amplitude of Cp over the rudder surface.



(a) 10 degrees, $y = 0.15$ **(b)** $y = 0.30$ **(c)** $y = 0.55$ **(d)** $y = 0.70$

(e) 20 degrees, $y = 0.15$ **(f)** $y = 0.30$ **(g)** $y = 0.55$ **(h)** $y = 0.70$

FIGURE 17 Pressure coefficient on the rudder's surface at various height. The first line at 10 degrees angle of attack, and the second one at 20. For each section, the bottom plot is a comparison with *Least squares* results.



(a) $y = 0.20$ **(b)** $y = 0.55$ **(c)** $y = 0.70$ **(d)** Discretisation uncertainty on the pressure side

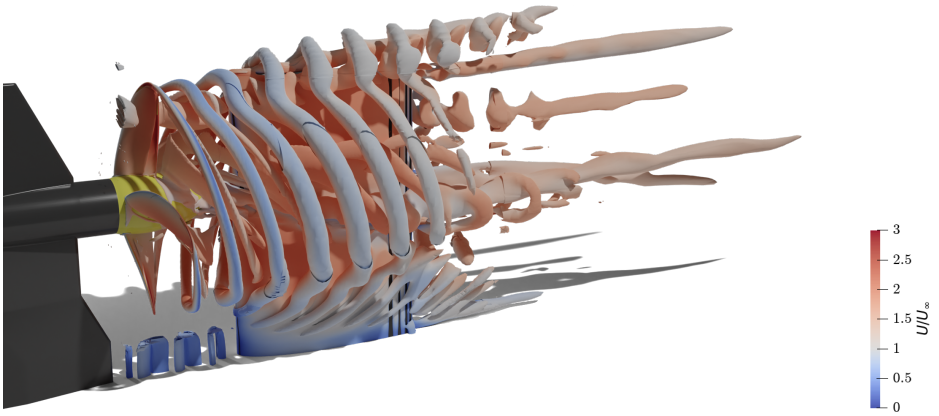
FIGURE 18 Space discretisation uncertainties for the local pressure coefficient. Data for the *Least squares* computation at 10 degrees rudder angle. In line plots, the pressure side is colored in red when the suction side is in blue.

4.3 | Velocity field

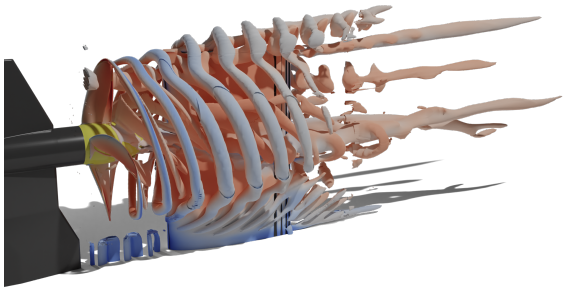
To understand better the error patterns and their influence on the flow field, this section focuses on the analysis of the velocity field. Figure 19 shows, for the different interpolation schemes, iso-surfaces of Qcriterion. This quantity, computed from the velocity components, highlights coherent structures by displaying shear layer vortices. For such flow, it makes some features clearly identifiable like helices formed by blade tip vortices or the hub vortex, the later being split in two vortices by the rudder's leading edge. At 10 degrees angle of attack, the rudder also generates two vortices at the leading edge and the trailing edge of its tip section, the one on the leading edge then blends with the wake of the centre board placed upstream of the propeller. Finally, flow acceleration in the propeller wake is visible by the higher velocity (in red), but also, by the increase of helix pitch for vortices closer to the hub vortex. Besides these flow features, visualisation artefacts from the overset meshes are also visible. On blades' tip vortices, for example, a 'step' is seen wherever the vortex crosses an overset boundary. These discontinuities are intrinsic to the overset method as separate iso-surfaces are constructed for each mesh. The difference in sizes and cell locations on both sides of the overset interface means that the iso-surfaces do not perfectly match across the boundary.

When comparing the different interpolation schemes, however, only marginal changes can be seen on such visualisation. The three schemes all show the same main flow features as described in the previous paragraph, moreover, lower order schemes (like *Inverse distance*) are not excessively diffusing vortices at the overset interface compared to higher order ones like is sometimes the case for free surface flows[46]. In fact, only the wake and hub vortex are slightly different, but this is likely due to the chaotic characteristics of the flow in these regions rather than to the interpolation method itself.

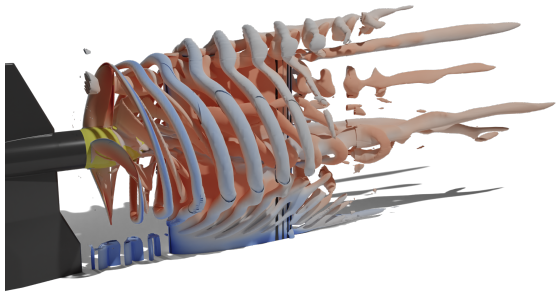
In order to analyse the flow more quantitatively, Figure 20 shows a top view of the domain with, from top to bottom, iso-Qcriterion, the velocity field at the propeller hub height with *Least squares* interpolation scheme, and a comparison with *Inverse distance* and *Nearest cell gradient* computations, the left column showing 10 degrees rudder angle and the right ones 20 degrees. From the iso-Qcriterion and velocity slice, Figures 20a to 20d, the flow features highlighted in the previous section are still



(a) *Least squares*

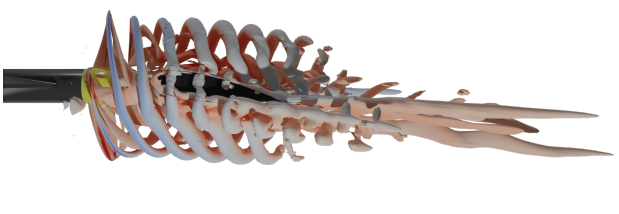


(b) *Inverse distance*

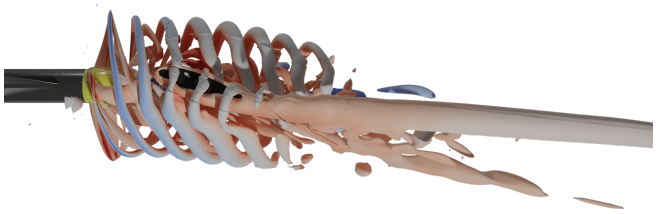


(c) *Nearest cell gradient*

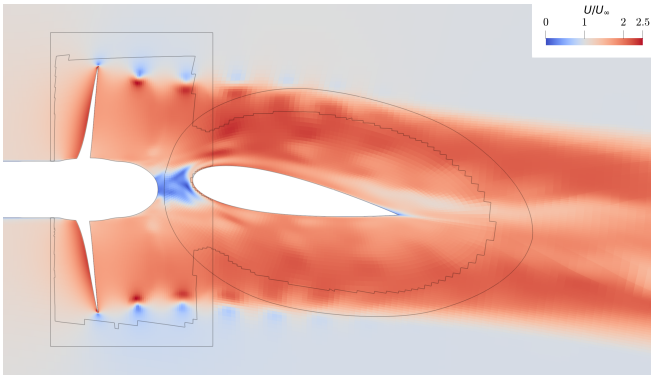
FIGURE 19 Iso-surface of Qcriterion coloured by velocity for the tested interpolation schemes with a 10 degrees rudder angle on mesh assemblies G4.



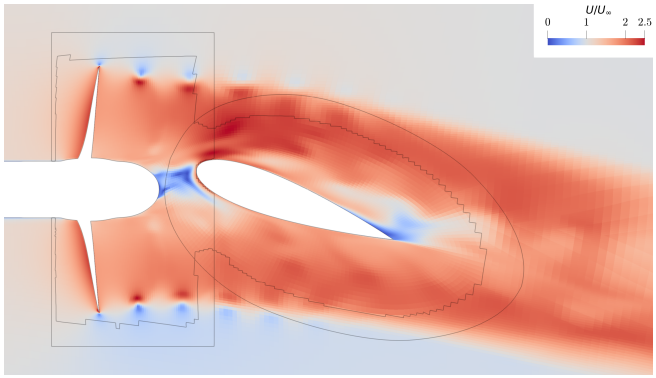
(a) *Least squares* iso-Qcriterion, 10 degrees



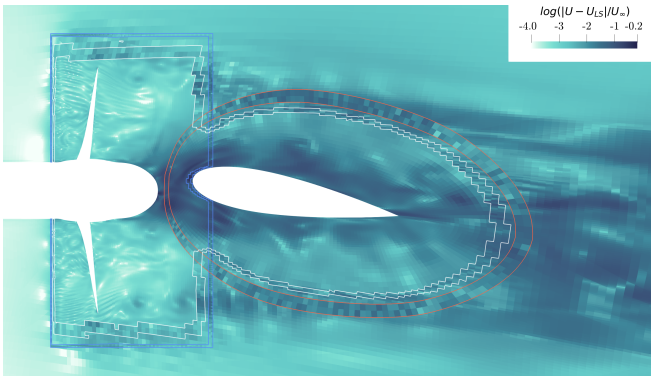
(b) *Least squares* iso-Qcriterion, 20 degrees



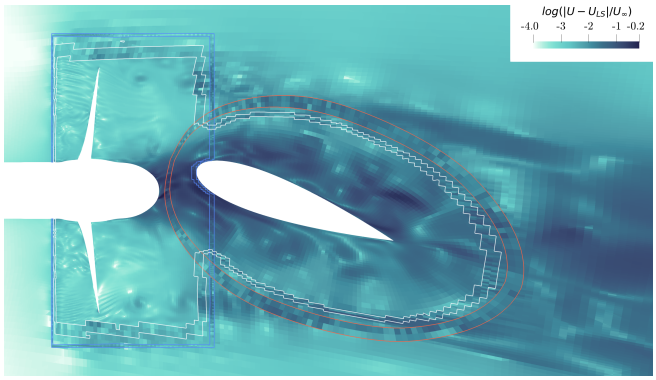
(c) *Least squares*, 10 degrees



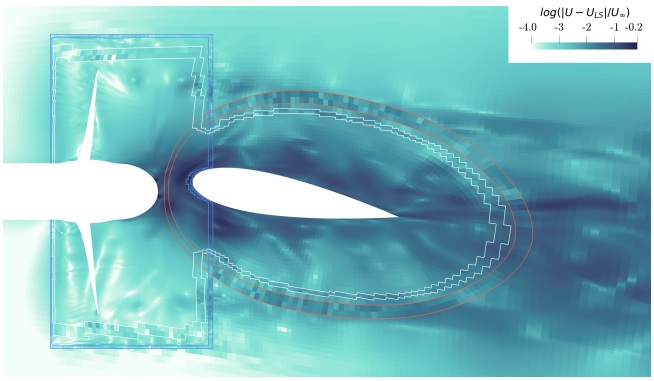
(d) *Least squares*, 20 degrees



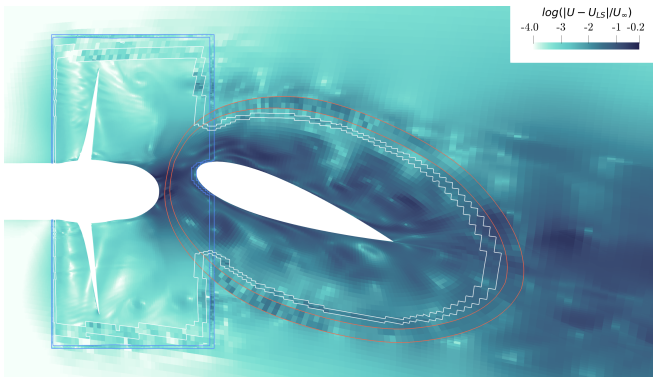
(e) *Inverse distance*: $log(|U_{InvDist} - U_{LS}|/U_\infty)$, 10 degrees



(f) *Inverse distance*, 20 degrees

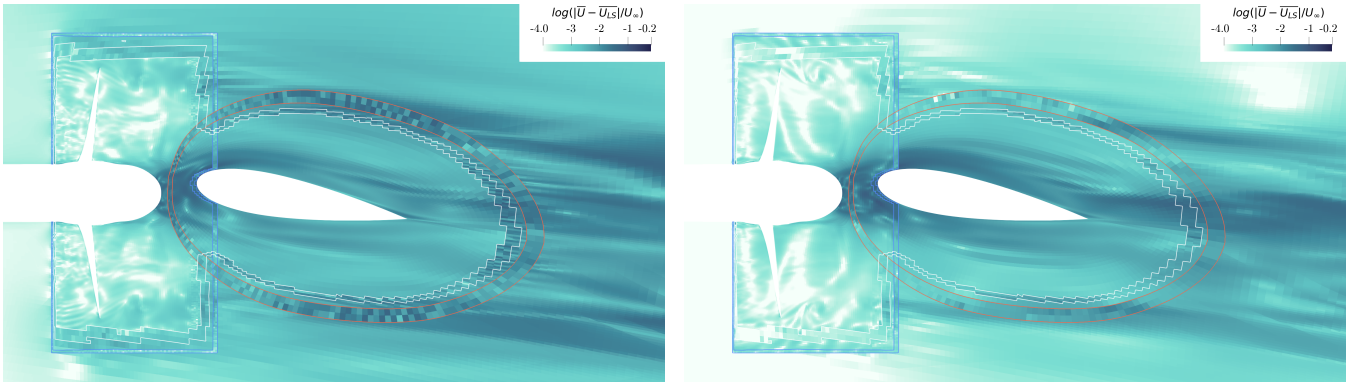


(g) *Nearest cell gradient*: $log(|U_{NCGrad} - U_{LS}|/U_\infty)$, 10 degrees



(h) *Nearest cell gradient*, 20 degrees

FIGURE 20 Comparison of the different interpolation schemes flow. 20a and 20a show iso-Qcriterion for the *Least squares* computations, 20c and 20d show the velocity field for the same computation, and finally 20e to 20h compares *Inverse distance* and *Nearest cell gradient* velocity fields to *Least squares* ones. In the last set of plots, *fringe* cells are highlighted in white for the tunnel mesh, blue for the propeller one and finally red for the rudder. 10 and 20 degrees rudder angles are shown in the left and right columns respectively.



(a) *Inverse distance* **(b)** *Nearest cell gradient*

FIGURE 21 Top view of the domain showing a comparison of the time-averaged velocity fields with *Least squares* computation, at 10 degrees rudder angle. *fringe* cells are highlighted in white for the tunnel mesh, blue for the propeller and red for the rudder.

noticeable. Furthermore, at 20 degrees, flow separation can be observed at the rudder's trailing edge, the wake being diverted towards the suction side even more. In both cases this wake orientation is not only due to the rudder angle but also to the entire system orientation the propeller and centre board being at a drift angle of 7.5 degrees and therefore misaligned with the tunnel and domain inlet. When comparing the other interpolation schemes with the *Least squares* computation, the hub vortex area shows the largest differences, once again likely due to the chaotic property of the flow at that location. Larger differences are also noticeable on each propeller tip vortex also coinciding with the rudder *fringe* cells region. These differences are finally convected downstream. On *fringe* cells themselves, the difference field is not as smooth as elsewhere because of interpolation errors, the *Inverse distance* computation showing more interpolation artefacts than the *Nearest cell gradient* one with larger cell-to-cell variations. Just like with the noise sources analysed in time in section 4.1, this increased spacial noise produced by the *Inverse distance* interpolation is likely originated from the higher interpolation error it produces, each *fringe* cell generating a larger error that is not correlated to its neighbouring *fringe* cell leads to more visible cell-to-cell differences. Moreover, since the set of *donor* cells associated to each *fringe* cell changes every time-step when moving meshes are involved, the interpolation error generated on *fringe* cells is also not correlated in time. This is then the source of the high frequency oscillation discussed in section 4.1.

Finally, it is interesting to notice that the comparison of time-averaged velocity fields, as seen in Figure 21, intensifies these cell-to-cell variations. Indeed, while other differences seem to average out, the differences located on *fringe* cells are still present and emphasised. This can be explained by the fact that the rudder and tunnel meshes are not moving, *fringe* cells then always have the same set of *donor* cells, leading to consistently over or under predicting their interpolated field values. These differences then, once again, follow streamlines and propagate downstream. This effect, however, affects less the *Nearest cell gradient* plot as it has both overall lower differences and smoother *fringe* cells regions.

Even though these plots do not directly show interpolation or discretisation errors, the large difference in methodology between the three schemes demonstrates that the *Nearest cell gradient* and *Least squares* methods create fewer artefacts, produce overall smoother interpolated fields and have lower interpolation errors compared to the *Inverse distance* scheme. This conclusion is in line with previous work done using manufactured solution for which errors could be probed directly [1]. With the present study, however, it is not possible to state which of *Nearest cell gradient* or *Least squares* computations have less interpolation errors and lead to a smoother interpolated fields.

4.4 | Performance considerations

Because comparing accuracy of the different schemes is only relevant in light of their respective performance, this section analyses the different computational overheads. With overset grid computations, these overheads come from the DCI computation (determining the locations of *fringe* and *hole* cells), the *donor* cell search and the interpolation computation itself. Depending on the interpolation scheme selected, different amount of *donor* cells need to be gathered per *fringe* cell, and the complexity of the interpolation computation also varies. The DCI, however, is the same regardless of the scheme, and is, in this implementation,

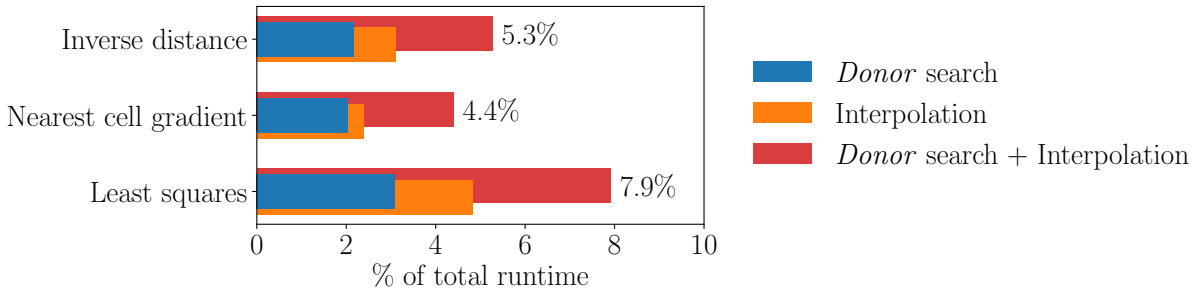


FIGURE 22 Time spent in overset related functions as a percentage of the total computation runtime for mesh G4 when using 500 cores.

computed by the external library Suggar++[36] as stated before. Moreover, for performance reasons and because in this test case the motion is periodic, the computation of the DCI for each time step is precomputed and stored to file to be read during the simulation which is not the case of the *donor* search and interpolation themselves. These are recomputed every time step and outerloop respectively (even though similar cache systems could be implemented). For this reason, this discussion will focus on the *donor* search and the interpolation steps only, with more details on the implementation, parallel performance and scalability given in Lemaire et al.[1].

Performance is shown for the finest set of meshes (G4) using 500 cores on Marclus4 cluster, a 20 cores per node system with InfiniBand inter-connect. Using domain decomposition this results in about 75,000 cells per core. In the present configuration, the *Inverse distance* scheme requires seven *donor* cells per *fringe* (i.e. one cell and all its direct neighbours), the *Nearest cell gradient* uses a single *donor* cell and its gradient, and the *Least squares* using a polynomial function of degree two here uses 25 *donor* cells per *fringe* cell (2.5 times the number of polynomial coefficients). Wall clock times of the *donor* search and interpolation phases as a percentage of the total runtime are plotted in Figure 22. First, one can notice that the *donor* search does not scale with the number of required *donor* cells. Indeed, for each *fringe* cell, finding the first *donor* cell is a lot more time-consuming than looking for its neighbours (which uses mainly the topology of the meshes and is better parallelised). For that reason, the *Nearest cell gradient* computation spends around 2% of its time looking for a single cell per *fringe* cell when, for the *Least squares* one, searching for 25 of them takes only around 3% of total runtime. The actual computation of the interpolation is also very different depending on the scheme, the *Inverse distance* is just computing a weighted average based on the distance between *fringe* and *donor* cells, which is seen to take longer than computing a *Nearest cell gradient* interpolation likely due to the larger number of cells involved (and hence involving more parallel communications). The *Least squares* interpolation is here more expensive than the other two, once again due to the larger number of *donor* cells involved in each interpolation.

In conclusion, this means that the benefits given by the *Least squares* schemes compared to the *Inverse distance* method are at the cost of about 2.5% longer runtime. Moreover, the *Nearest cell gradient*, which shares a lot of the benefits of the *Least squares* approach, is even slightly less expensive than the *Inverse distance* scheme. Lastly, the overset grid computational overhead is at most 8% of the total runtime (with *Least squares*).

5 | RUDDER FLOW VALIDATION

In this section, the validation is focussed on the rudder because as long as the thrust loading generated by the propeller is correct[18] then the rudder's performance is the most sensitive to errors induced by the interpolation scheme as onset flow goes through propeller and then into rudder overset mesh itself.

As discussed in section 2, the propeller geometry used in the CFD analysis is different from the experimental one, as a result, direct validation of the entire setup is not possible. Rudder characteristics can however be validated using K_T equivalent computations. Experiments done at $J = 0.51$ with a -7.5 degrees drift angle and a 10 degrees rudder angle led to $K_T = 0.242$[16]. Thus, in order to replicate the propeller wake, albeit with a different propeller geometry, a CFD computation has been done at $J = 0.61$ with similar drift and rudder angles which resulted, on grid G4, in $K_T = 0.235$. This is less than 3.5% away from the experimental value and, considering the verification study done at $J = 0.51$, it was deemed close enough for the purpose of this validation. The rest of this section, then, compares experimental data with this CFD computation done at $J = 0.61$, starting with

Table 3 showing integral quantities. As said, K_T is, by design, close to the experimental one, K_Q is, however, quite far off with 20% difference. Which could partially be expected when using such methodology. On the other hand, rudder force coefficients are within 10% of the experimental ones, with C_L even below 1%. These values justify the K_T-equivalent methodology and concur in the validation of the rudder flow. The moment coefficient taken at 30% chord sees a larger difference which led to a 4.3% chord length offset in the centre of pressure C_{Pc}.

TABLE 3 Comparison between CFD and experimental [16] integral quantities for the K_T equivalent validation.

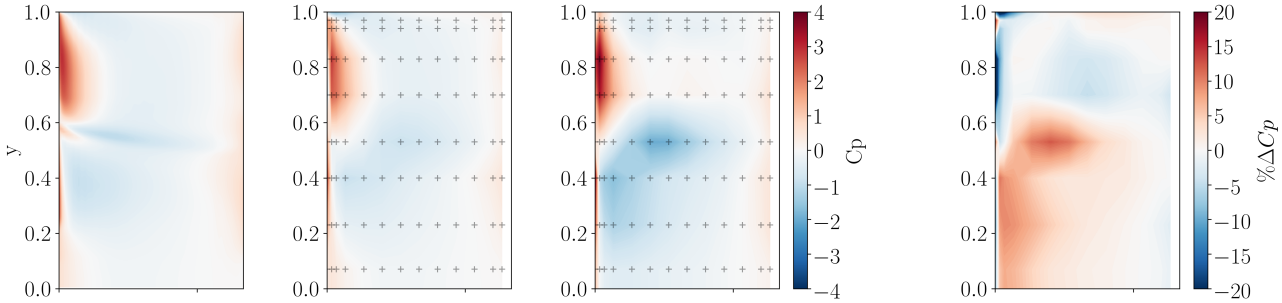
	CFD	Exp	comp
J	0.61	0.51	
K_T	0.2350	0.243	-3.28%
K_Q	0.0354	0.045	-21.32%
C_L	0.5903	0.587	0.56%
C_D	0.1055	0.114	-7.46%
C_m	-0.0520	-0.078	-33.37%
C_{Pc} (in % chord)	21.3174	17.02	4.29

In the experimental campaign, the rudder was equipped with pressure probes allowing comparison of the surface pressure. This pressure was, however, recorded only on cases without the upstream centre board which has some effects on the pressure distribution downstream. It should then be noted that the comparison done on the pressure is comparing data with a centre board (CFD) and without (Exp). Figure 23 displays the pressure on the rudder for both CFD and experimental results. The first column shows raw CFD pressure coefficient on both sides when the second one downsamples it at the locations where the experimental pressure probes were placed for easier comparison with the third column, i.e. the experimental results. Finally, the last column compares the two sets of results.

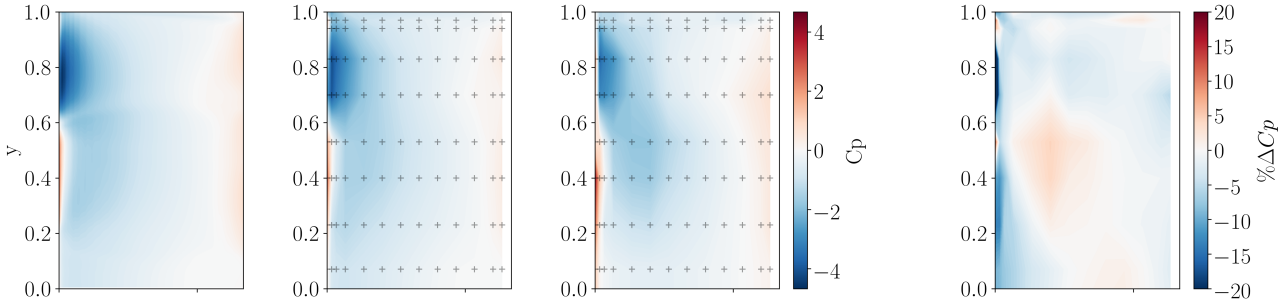
Due to the low amount of pressure probes, some features disappear from the experimental and downsampled plots like the distinct lower pressure region of the hub vortex or the extent of the higher pressure region at the trailing edge. Overall, however, both the CFD and experimental plots show a high pressure region at the top of the pressure side, a similar low pressure region on the suction side etc. The CFD computation has lower C_p amplitude with high C_p regions not as high as experimental ones, and, in general, higher C_p in low pressure regions. These observations are highlighted in Figure 24 which shows chord-wise slices of C_p at various span heights. From them, it is more clearly visible that the top part of the rudder shows better correlation with experimental data and the hub vortex section ($y = 0.53$) significantly underestimate the pressure side C_p while the rest of the slope is better predicted.

The differences seen on the pressure profile can have various origins. Firstly, it can be related to the lack of match between the experimental and CFD setups, with the difference in propeller geometry, the K_T equivalent methodology or the lack of centre board in this sets of results in the experiments. Secondly, it can also come from modeling errors directly as the CFD simulations use unsteady RANS approach to model turbulence (using the $k - \sqrt{k}L$ model). Thirdly, it can come from discretisation errors, as seen in section 4.2, the hub vortex region is where the discretisation uncertainty is the largest. Finally, it can also come from the overset interpolation upstream, and interpolation errors combined with the hub vortex could here create larger differences. It should also be noted that the literature related to this set of experiments did not provide uncertainty quantifications on the measurements, though it would likely be below the differences seen here.

To conclude, even though some differences can be originating from the K_T-equivalent methodology and different geometries, the very close correlation of rudder coefficients like C_L, C_D or the centre of pressure as well as the overall good capture of the features present on the rudder surface lead toward a favorable validation of the flow. The circumstances of this study, however, make it harder to quantify the degree of confidence of this validation.

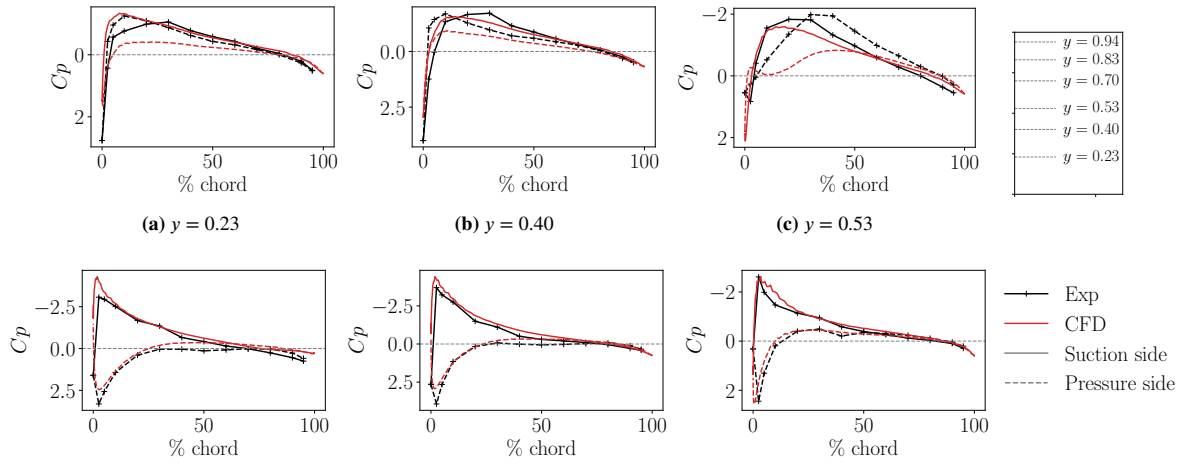


(a) CFD, pressure side **(b)** CFD: downsampled (like Exp) **(c)** Exp, pressure side **(d)** $Cp_{CFD} - Cp_{Exp}$



(e) CFD, suction side **(f)** CFD: downsampled (like Exp) **(g)** Exp, pressure side **(h)** $Cp_{CFD} - Cp_{Exp}$

FIGURE 23 Pressure coefficients on both sides of the rudder surface, comparison of CFD and experimental data from Molland and Turnock[16]. The first two columns show the CFD results, both raw and downsampled to the probes locations, then the third one show experimental data. Finally, the last column compares the two data sets.



(a) $y = 0.23$ **(b)** $y = 0.40$ **(c)** $y = 0.53$

(d) $y = 0.70$ **(e)** $y = 0.83$ **(f)** $y = 0.94$

FIGURE 24 Pressure coefficient on the rudder's surface at various span sections comparing CFD and experimental data from Molland and Turnock[16]

6 | CONCLUDING REMARKS

In this work, the accuracy and performance of an overset implementation has been assessed using three interpolation schemes from first to third order accurate on a rudder-propeller flow case at a drift angle. This has been achieved by comprehensive solution verification and error source analysis as well as validation against experiments done by Molland and Turnock[16]. This study, aimed at characterising and quantifying the impact of the different interpolation schemes on industry standard overset computations, analysed integral quantities, pressure coefficients, overset-related mass imbalance and field data, and led to the following conclusions:

- The first order *Inverse distance* produces more artefacts on *fringe* cells leading to high frequency oscillations on the different force coefficients of the propeller and rudder. A spectral analysis quantified these oscillations as having more than twice the energy as the ones produced by the second order *Nearest cell gradient* or third order *Least squares* schemes. Beside impacting the accuracy, those spurious oscillations can be an error source when doing common propeller acoustics studies, and cavitation flow simulations.
- These oscillations do not, however, result in significant changes in time-averaged force coefficients as the differences produced is far smaller than the discretisation uncertainty. Similarly, time-averaged rudder's pressure coefficients were also comparable for the three tested schemes, with the largest difference in the hub vortex region which is also where the space discretisation uncertainty is the largest. One can note that these little changes when comparing the different interpolation schemes are similar for all four mesh refinement tested, suggesting that this conclusion is not significantly influenced by the discretisation error.
- Another source of error of the overset method is its lack of mass flux conservation at the overlap interface, even in a theoretically conservative finite volume discretisation. By inspecting each mesh individually, it was observed that the third order accurate *Least squares* interpolation achieved mass flux imbalance one order of magnitude lower than the other two schemes, and the *Nearest cell gradient* performed better than the *Inverse distance* scheme.
- Comparison of the velocity fields showed clear effects of the interpolation errors on *fringe* cells propagating downstream, particularly for the *Inverse distance* scheme. These interpolation errors are not cancelled out by time average, highlighting the fact that errors on non moving *fringe* cells are consistently over- or under-predicted.
- In the current implementation, the use of the third order *Least squares* scheme only leads to 2.5% longer runtime compared to first order *Inverse distance*. Moreover, the second order *Nearest cell gradient*, which shares most of the benefits of the *Least squares* is cheaper than the *Inverse distance* scheme, reducing the total runtime by about 1%.
- The validation of the rudder flow, done with K_T equivalent computations, showed promising results with integral quantities within 10% of the experimental ones and Cp on the rudder surface capturing the main complex flow features such as the hub vortex, high and low pressure regions due to the propeller rotation.

This work largely confirms the conclusions drawn in Lemaire et al.[1] which performed a similar study but using manufactured solutions. Both studies show the production of smoother fields when employing high order interpolation schemes, and/or lower numerical errors. The main difference lies on the mass imbalance behaviour; while in this work it is seen decreasing with higher order interpolation schemes, similar error levels were observed regardless of the interpolation order used in Lemaire et al.[1]. Moreover, the presented study expanded the previous conclusions to moving meshes and real industrial applications.

Finally, while this paper mainly focuses on verification and comparison of the different interpolation schemes, the validation exercise here performed could be improved by reducing the uncertainties related to the difference in the geometries (propeller and centre board) used. Performing a similar study on free surface flows should be the next step in verification and validation of the overset method, as this is essential for maritime world applications.

Acknowledgements

The authors would like to thank Dr. Artur K. Lidtke from MARIN for his help with the generation of the propeller mesh. This research was financially supported by the EPSRC Centre for Doctoral Training in Next Generation Computational Modelling (EP/L015382/1) at the University of Southampton including financial support from MARIN. The authors also acknowledge the

use of the IRIDIS High Performance Computing Facility, and associated support services at the University of Southampton as well as MARIN HPC Marclus4 and blueOASIS resources and facilities, in the completion of this work.

Data Availability Statement

The data that support the findings of this study are available from the corresponding author upon reasonable request.

References

1. Lemaire S, Vaz G, Deij-van Rijswijk M, Turnock SR. On the Accuracy, Robustness, and Performance of High Order Interpolation Schemes for the Overset Method on Unstructured Grids. *International Journal for Numerical Methods in Fluids* 2022; 94(2): 152–187. doi: 10.1002/fld.5050

2. Chandar DD, Boppana B, Kumar V. A Comparative Study of Different Overset Grid Solvers Between OpenFOAM, Star-CCM+ and Ansys-Fluent. In: *2018 AIAA Aerospace Sciences Meeting* American Institute of Aeronautics and Astronautics; 2018; Reston, Virginia

3. Chen HC, Chen M. Chimera RANS Simulation of a Berthing DDG-51 Ship in Translational and Rotational Motions. *International Journal of Offshore and Polar Engineering* 1998; 8(3): 182–191.

4. Boger DA, Dreyer J. Prediction of Hydrodynamic Forces and Moments for Underwater Vehicles Using Overset Grids. In: *44th AIAA Aerospace Sciences Meeting* No. January. American Institute of Aeronautics and Astronautics; 2006; Reston, Virigina: 1–13

5. Carrica PM, Ismail F, Hyman M, Bhushan S, Stern F. Turn and Zigzag Maneuvers of a Surface Combatant using a URANS Approach with Dynamic Overset Grids. *Journal of Marine Science and Technology* 2013; 18(2): 166–181. doi: 10.1007/s00773-012-0196-8

6. Shen Z, Wan D, Carrica PM. Dynamic overset grids in OpenFOAM with application to KCS self-propulsion and maneuvering. *Ocean Engineering* 2015; 108: 287–306. doi: 10.1016/j.oceaneng.2015.07.035

7. Windt C, Davidson J, Chandar DD, Ringwood JV. On the Importance of Advanced Mesh Motion Methods for WEC Experiments in CFD-based Numerical Wave Tanks. In: Ringsberg RB, J. ., eds. *VIII International Conference on Computational Methods in Marine Engineering MARINE 2019* International Center for Numerical Methods in Engineering (CIMNE); 2019; Gothenburg, Sweden: 145–156.

8. Völkner S, Brunswig J, Rung T. Analysis of Non-Conservative Interpolation Techniques in Overset Grid Finite-Volume Methods. *Computers & Fluids* 2017; 148: 39–55. doi: 10.1016/j.compfluid.2017.02.010

9. Chandar DD. Assessment of Interpolation Strategies and Conservative Discretizations on Unstructured Overset Grids in OpenFOAM. In: *2018 AIAA Aerospace Sciences Meeting* No. January. American Institute of Aeronautics and Astronautics; 2018; Reston, Virginia: 1–15

10. Chandar DD. On Overset Interpolation Strategies and Conservation on Unstructured Grids in OpenFOAM. *Computer Physics Communications* 2019; 239: 72–83. doi: 10.1016/j.cpc.2019.01.009

11. Chandar DD, Sitaraman J. A Flux Correction Approach for the Pressure Equation in Incompressible Flows on Overset Meshes in OpenFOAM. *Computer Physics Communications* 2022; 273: 108279. doi: 10.1016/j.cpc.2021.108279

12. Brazell MJ, Sitaraman J, Mavriplis DJ. An Overset Mesh Approach for 3D Mixed Element High-Order Discretizations. *Journal of Computational Physics* 2016; 322: 33–51. doi: 10.1016/j.jcp.2016.06.031

13. Gomes T, Lemaire S, Vaz G. Code and Solution Verification of Sliding and Overset Grid Methods on Wind Turbine Flows. In: *ASME 41th International Conference on Ocean, Offshore and Arctic Engineering* ; 2022; Hamburg, Germany.

14. Dooley G, Ezequiel Martin J, Buchholz JH, Carrica PM. Ship Airwakes in Waves and Motions and Effects on Helicopter Operation. *Computers & Fluids* 2020; 208: 104627. doi: 10.1016/j.compfluid.2020.104627

15. Gatin I, Vukcevic V, Jasak H, Lalovic I. Manoeuvring Simulations using the Overset Grid Technology in FOAM-extend. In: *32nd Symposium on Naval Hydrodynamics* No. August. ; 2018; Hamburg, Germany: 10.

16. Molland AF, Turnock SR. Wind Tunnel Tests on the Effect of a Ship Hull on Rudder-Propeller Performance at Different Angles of Drift. tech. rep., University of Southampton; Southampton, UK: 1995.

17. Turnock SR. A Test Rig for the Investigation of Ship Propeller/Rudder Interactions. tech. rep., University of Southampton; Southampton, UK: 1990.

18. Molland AF, Turnock SR. *Marine Rudders, Hydrofoils and Control Surfaces*. Elsevier. butterwort ed. 2022

19. Phillips AB, Turnock SR, Furlong M. Accurate Capture of Propeller-Rudder Interaction using a Coupled Blade Element Momentum-RANS Approach. *Ship Technology Research* 2010; 57(2): 128–139. doi: 10.1179/str.2010.57.2.005

20. Villa D, Franceschi A, Viviani M. Numerical Analysis of the Rudder–Propeller Interaction. *Journal of Marine Science and Engineering* 2020; 8(12): 990. doi: 10.3390/jmse8120990

21. Badoe CE, Phillips AB, Turnock SR. Influence of Drift Angle on the Computation of Hull–Propeller–Rudder Interaction. *Ocean Engineering* 2015; 103(0): 64–77. doi: 10.1016/j.oceaneng.2015.04.059

22. Yilmaz N, Aktas B, Atlar M, Fitzsimmons PA, Felli M. An Experimental and Numerical Investigation of Propeller-Tudder-Hull Interaction in the Presence of Tip Vortex Cavitation (TVC). *Ocean Engineering* 2020; 216(May): 108024. doi: 10.1016/j.oceaneng.2020.108024

23. Di Mascio A, Dubbioso G, Muscari R, Felli M. CFD Analysis of Propeller-Rudder Interaction. In: *International Ocean and Polar Engineering Conference* No. June 21-26. International Society of Offshore and Polar Engineers (ISOPE); 2015; Kona, Big Island, Hawaii, USA: 946–950.

24. Zhirong S, Decheng W, Carrica PM. RANS Simulations of Free Maneuvers with Moving Rudders and Propellers Using Overset Grids in OpenFOAM. In: *SIMMAN workshop on Verification and Validation of Ship Maneuvering Simulation Methods* No. December. ; 2014; Lyngby, Denmark.

25. Benek JA, Buning PG, Steger JL. A 3-D Chimera Grid Embedding Technique. In: *7th Computational Physics Conference* American Institute of Aeronautics and Astronautics; 1985; Reston, Virigina: 10

26. Schreck E, Peric M. Overset Grids in STAR-CCM+: Methodology, Applications and Future Developments. In: *STAR Japanese Conference* ; 2012.

27. Deng G, Leroyer A, Guilmineau E, Queutey P, Visonneau M, Wackers J. CFD Simulation of PMM Motion in Shallow Water for the DTC Container Ship. In: *4th International Conference on Ship Manoeuvring in Shallow and Confined Water with Special Focus on Ship Bottom Interaction* ; 2016; Hamburg, Germany: 93–98

28. Noack RW. DiRTlib: A Library to Add an Overset Capability to your Flow Solver. In: *17th AIAA Computational Fluid Dynamics Conference* No. June 2005. American Institute of Aeronautics and Astronautics; 2005; Toronto, Ontario, Canada: 1–20

29. Eça L, Hoekstra M. A Procedure for the Estimation of the Numerical Uncertainty of CFD Calculations Based on Grid Refinement Studies. *Journal of Computational Physics* 2014; 262: 104–130. doi: 10.1016/j.jcp.2014.01.006

30. Brouwer J, Tukker J, Klinkenberg Y, Rijsbergen vM. Random Uncertainty of Statistical Moments in Testing: Mean. *Ocean Engineering* 2019; 182(April): 563–576. doi: 10.1016/j.oceaneng.2019.04.068

31. Molland AF, Turnock SR. Wind tunnel test results for a model ship propeller based on a modified Wageningen B4.40. tech. rep., University of Southampton; Southampton, UK: 1990.

32. Turnock SR. Computer Aided Design and Numerically Controlled Manufacture of a Split Mold for a Composite Model Ship Propeller. tech. rep., University of Southampton; 1990.

33. Vaz G, Jaouen F, Hoekstra M. Free-Surface Viscous Flow Computations: Validation of URANS Code FreSCo. In: *Volume 5: Polar and Arctic Sciences and Technology; CFD and VIV* ASMEDC; 2009: 425–437

34. Klaij CM, Vuik C. SIMPLE-type Preconditioners for Cell-Centered, Colocated Finite Volume Discretization of Incompressible Reynolds-Averaged Navier-Stokes equations. *International Journal for Numerical Methods in Fluids* 2013; 71(7): 830–849. doi: 10.1002/fld.3686

35. Noack RW. SUGGAR: a General Capability for Moving Body Overset Grid Assembly. In: *17th AIAA Computational Fluid Dynamics Conference* . 5117. ; 2005; Toronto, Ontario, Canada: 1–21

36. Noack RW, Boger DA. Improvements to SUGGAR and DiRTlib for Overset Store Separation Simulations. In: *47th AIAA Aerospace Sciences Meeting* No. January. ; 2009; Orlando, Florida: 5–9

37. Cadence . Pointwise. .

38. PDC . GridPro. .

39. Foeth Ej, Deij-van Rijswijk M. Remodeling the B-series Geometry in a CAD Environment. In: Felli M, Leotardi C. , eds. *6th International Symposium on Marine Propulsors* No. May. National Research Council of Italy, Institute of Marine Engineering (CNR-INM), Via di Vallerano 139, 00128 Rome, Italy; 2019; Rome.

40. Robert McNeel & Associates . Rhinoceros. .

41. Lammeren vW, Manen vJD, Oosterveld M. The Wageningen B-screw Series. *Schip en Werf* 1970; 5: 88–103.

42. Menter FR, Egorov Y, Rusch D. Steady and Unsteady Flow Modelling Using the k-skL Model. In: *Proceedings of the International Symposium on Turbulence, Heat and Mass Transfer* Begell House; 2006; New York: 403–406

43. Brouwer J, Tukker J, Rijsbergen vM. Uncertainty Analysis and Stationarity Test of Finite Length Time Series Signals. In: *4th International Conference on Advanced Model Measurement Technology for the Maritime Industry* ; 2015.

44. Eça L, Vaz G, Hoekstra M. On the Role of Iterative Errors in Unsteady Flow Simulations. In: *21st Numerical Towing Tank Symposium (NuTTS)* ; 2018; Cortona, Italy: 2–7.

45. Lemaire S, Klapwijk M. PyTST: Python Library and Command Line Tool Performing the Transient Scanning Technique. 2021. doi: 10.5281/zenodo.4428158

46. Marin Lauber , Pandeli Temarel . Acquisition of Maneuvring Characteristics of Ships using RANS CFD. In: *22nd Numerical Towing Tank Symposium (NuTTS)* ; 2019; Tomar, Portugal: 1–6.