Evaluating and Optimising the OpenFOAM Overset Mesh Solver for Offshore Renewable Applications

Archer2-eCSE07-7 Techinal Report

Ranjodh Rai, Ling Qian, Zhihua Ma, and Wei Bai

Centre for Mathematical Modelling and Flow Analysis, Department of Computing and Mathematics, Manchester Metropolitan University, Manchester M1 5GD, United Kingdom

Abstract

The overset mesh method has gained significant popularity for use in computational fluid dynamics (CFD) due to its ability to handle large amplitude motion of structures with complex geometries. However, solvers employing this method can be computationally expensive, so the primary objective of this work is to optimise the open-source ESI-OpenCFD OpenFOAM overset mesh solver for offshore renewable applications by improving its efficiency. A secondary objective is to also establish whether the updated hole-cutting procedure released in OpenFOAM v2212 improves the accuracy of solutions and reduces computational time. A profiling investigation of the solver establishes that the method for donor searching is the primary efficiency bottleneck; hence, a new optimised method is developed that reduces the number of donors that have to be found and consequently reduces the number of necessary inter-processor communications. The new method is validated through two test cases: regular wave interaction with a two-dimensional T-shaped floating body and the aerodynamic modelling of an offshore wind-turbine rotor in constant wind. It is found that the new method reduces computational time in the T-body case by 14 and 7 % on 128 and 256 cores respectively—a modest amount but to be expected for a simple case with only small amplitude motion. For the wind turbine case, the new method reduces computational time by 64.6 % on 1024 cores—approximately 3x faster. For the secondary objective, it is found that the updated hole-cutting method does indeed improve the accuracy of solutions for the T-body case but not necessarily for the wind turbine case. The affect on computational time is small.

Keywords: Archer2, eCSE, overset mesh, OpenFOAM, hydrodynamics, aerodynamics, renewable energy.

1 Introduction and Motivation

Accurate and efficient computational fluid dynamics (CFD) models are essential in the design of offshore renewable structures such wave energy converters and wind turbines. For hydrodynamic analysis, mesh-based methods require proper adjustment to the mesh to account for the movement of a rigid body. The most common method for doing this is using a dynamic mesh that can be deformed (morphed) according to the motion of the body [10, 12]. However, dynamic meshes are limited in the size of deformations possible because large deformations can result in severe mesh distortion and consequent failure of the simulation. This problem is amplified when angular, as well as linear, displacements are present. Remeshing could be used for mitigation, but this significantly increases computational overheads, particularly if the deformations occur quickly.

An alternative to a dynamic mesh approach is the use of an overset mesh [2]. The overset mesh methodology involves a composite mesh system with two or more overlapping meshes; one is always a fixed background mesh that encompasses the entire computational domain, and the others are body-fitted meshes for each 'body' that could be structures and/or their constituent components. Each constituent mesh is then modelled independently with the body-fitted mesh (es) moving simultaneously with their corresponding body as a single entity without any mesh deformations. For this to work, information must be exchanged between each body-fitted mesh and the background mesh at each time step. This approach has a major advantage over other techniques in that it allows for large amplitude motion of structures with complex geometries without changing the quality of the mesh. This also makes it useful for aerodynamic analysis of fully rotating structures such as wind turbines—an application that dynamic meshes cannot easily simulate. However, the primary disadvantage of the overset mesh approach is that it can be computationally expensive.

One of the most popular overset mesh solvers is the one developed in the open-source toolbox OpenFOAM. This powerful tool has been used for numerous applications such as simulating focused wave impacts on point absorber wave energy converters [6] and modelling the aerodynamics [9]—and hydrodynamics [7]—of a floating offshore wind turbine. Furthermore, comparisons have also been done between the solver and its dynamic mesh counterpart [1, 13, 5], all of which notably demonstrate that the overset approach in OpenFOAM is more computationally expensive. The overset solver has regularly been updated through later versions of OpenFOAM, but minimal updates have focused on the efficiency of the solver. Nevertheless, major improvements have been made to other aspects of the solver. For example, ESI-OpenCFD OpenFOAM v2212 (released in Dec. 2022 [3]) upgraded the 'hole-cutting' procedure which allows cells that are used for information exchange via interpolation to be done further away from solid body boundaries. This is beneficial as it allows interpolation further away from areas where the flow exhibits high gradients, consequently improving the flow solution around the body. This improvement is also hypothesised to increase the efficiency of the solver.

Given the problems with computational expense, the primary objective of this project is to optimise the OpenFOAM overset mesh solver for offshore renewable applications by improving its efficiency. In particular, the primary efficiency bottleneck is identified and a solution subsequently developed. The new method is validated for two test cases: regular wave interaction with a 2-D T-shaped floating body and the aerodynamic modelling of an offshore wind-turbine rotor in constant wind. A secondary objective of this project is to also establish whether the updated hole-cutting procedure introduced in ESI-OpenCFD OpenFOAM v2212 improves the accuracy of solutions and reduces computational time. This is also established through the two aforementioned test cases.

2 The Efficiency Bottleneck and Optimisation Strategy

2.1 Overset methodology

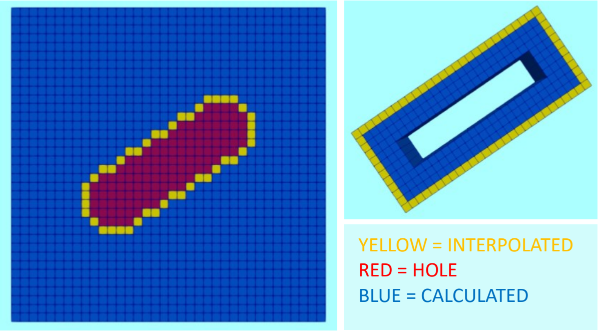

Figure 1: Composite mesh system for a simple 2-D rotor. The left image shows the background mesh; the right image shows the rotor and the body-fitted overset mesh.

As mentioned previously, the overset mesh method consists of a composite mesh system. Figure 1 shows a simple example of this with a 2-D rotor. Here we can see a fixed background mesh on the left and a body-fitted mesh on the right which 'fits' around the 2-D mixer. The blue cells are 'CALCULATED' cells on which the governing equations are solved and a flow solution is found. The red cells on the background mesh represent cells that lie within the mixer's surface when the two meshes are overlapped, and thus are marked as inactive or as 'HOLE' for which no flow solution is required. Establishing these cells is the first step during the overset procedure and is called 'hole cutting'. The yellow cells are known as 'INTERPOLATED' cells. These are cells through which data is exchanged between the two meshes at each time step. To do this, a 'donor' cell must be found on each analogous mesh from which an interpolation stencil can be created and data subsequently interpolated using a chosen interpolation method (such as inverse distance weighted). Note that the initial donor cell is specifically the cell whose centre lies closest in space to the centre of the corresponding INTERPOLATED cell.

The process of finding the donor cells is called 'donor searching', and collectively hole cutting and donor searching comprise the 'overset grid assembly procedure' (OGA). It is the OGA that takes up most of the computational time in overset methods.

2.2 Identifying the efficiency bottleneck

A profiling investigation of the OpenFOAM overset solver indicates that the donor searching part of the OGA procedure is the primary efficiency bottleneck. In particular, the `markDonors` function is found to be the main culprit due to its inefficiency and consequent negative affect on parallel efficiency. The reason for this is that donor searching is done for every cell on both background and overset meshes regardless of whether the cell is actually marked as INTERPOLATED. Referring back to Figure 1, this means that donors are found for every cell, including HOLE (red) and CALCULATED (blue), even though this is not strictly necessary. This causes two major problems:

1. The number of donors that have to be found is drastically increased, particularly for large-scale problems with potentially millions of mesh cells.

2. The number of necessary inter-processor communications drastically increases if donors have to be found on processors different to the ones containing the corresponding INTERPOLATED cell. This is particularly true if continuously large displacements are present (such as for rotating bodies).

2.3 A new optimised method for donor searching

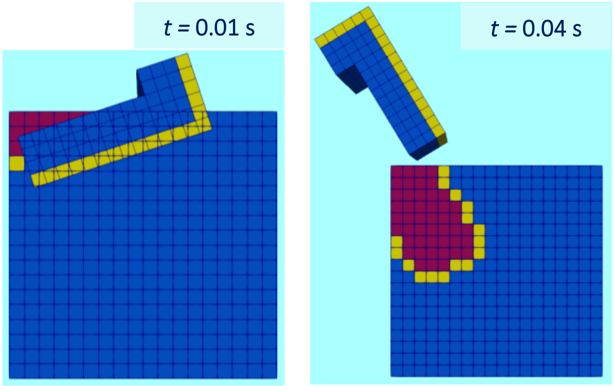

Figure 2: Hierarchically decomposed parts of the background and overset meshes on a single processor at two points in time. The left image shows both meshes overlapping on said processor at $t = 0.01$ s, whereas the right image shows the two meshes no longer overlapping at $t = 0.04$ s.

To combat the two aforementioned problems, a new optimised method is developed which aims to optimise the `markDonors` function so that it only finds donors for INTERPOLATED cells. This new method is hypothesised to improve computational efficiency by:

1. reducing the number of cells for which donors are found

2. reducing the number of inter-processor communications

This optimised method works in conjunction with the 'hierarchical' decomposition method in OpenFOAM which is previously known to be optimal for the overset solver when compared to the default 'scotch' method. An explanation for why this is, is given in Appendix A. To understand in detail how inter-processor communications are actually reduced, consider Figure 2. The mesh from Figure 1 has been hierarchically decomposed onto four processors with Figure 2 showing processor 4 in particular. At time $t = 0.01$ s, the decomposed parts of the background and overset meshes overlap, meaning that most donors can be searched for **locally** on the same processor. This implies low inter-processor communications. In contrast, at time $t = 0.04$ s, the background and overset meshes no longer overlap, meaning that all donor must be found on **remote** processors. This implies high inter-processor communications. Therefore, by only finding donors for INTERPOLATED cells at $t = 0.04$ s, the number of times that remote processors have to be searched is greatly reduced. This then implies low inter-processor communications.

3 Regular Wave Interaction With a T-shaped Floating Body

3.1 Computational setup

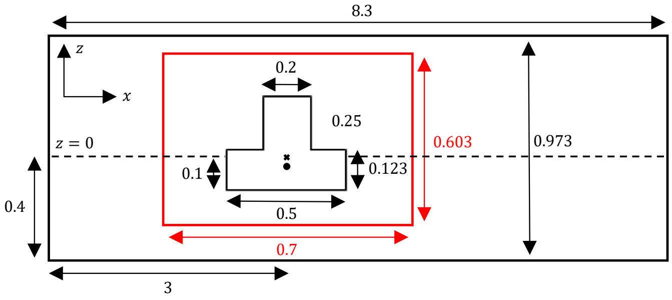

Figure 3: Schematic of the NWT for the 2-D T-shaped floating body test case (not to scale; all dimensions are in metres). The black cross at $y = 0$ marks the centre of rotation and the dot at $y = -0.0204$ marks the centre of mass. Red indicates the body-fitted mesh and dimensions.

This first test case considers the wave-structure interaction of fifth-order Stokes waves with a 2-D T-shaped floating body which is considered as a simplified midship section with superstructure. The case itself is based on the physical and numerical experiments of Zhao and Hu [14]. This test case was originally done using an older OpenFOAM version in Hao et al. [2], hence for comparison, the same geometric and computational setups are used here. The geometric setup of the 2-D numerical wave tank (NWT) is shown in Fig. 3. The background mesh has dimensions $x \times y \in [0,\ 8.3] \times [-0.4,\ 0.573]$ (m), with width in the transverse direction being 0.3 m (one computational cell thick). Its uniform mesh configuration is $N_x \times N_y = 1660 \times 195$. In addition, the body-fitted overset mesh has dimensions $x \times y \in [2.65,\ 3.35] \times [-0.2,\ 0.403]$ and has an initial mesh configuration of $N_x \times N_y = 160 \times 121$. After the T-shaped body has been cut from the overset mesh, the overall number of cells is approximately 3.4×10^5. The body is placed at a distance of 3 m away from the inlet and, as in the experiments, is constrained to only heave and roll motions. The input wave parameters are also shown in Tab. 1 and the simulation time is 15 s. For validation, the heave and roll motions of the body are measured relative its centre of rotation. Finally, each simulation was carried out on 2 Archer2 nodes (256 cores) using the 'overInterDyMFoam' solver and the inverse distance interpolation method.

Case	Wave	Amplitude: A(m)	Period: T(s)	Wavelength: λ(m)	Water depth: h(m)
T-body	Stokes 5^{th}	0.031	1	1.4637	0.4

Table 1: Input wave parameters for the 2-D T-shaped floating body test case.

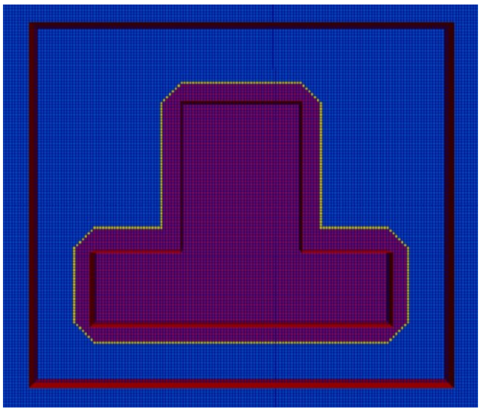

Figure 4: Background mesh with the hole pushed back 7 layers of mesh cells from the T-shaped body boundary.

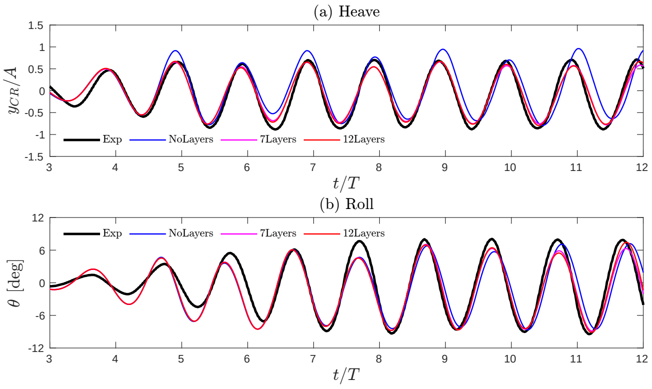

Figure 5: Time histories of (a) heave motion and (b) roll motion for no layers, 7 layers, 12 layers, and the experimental solution.

3.2 Numerical results

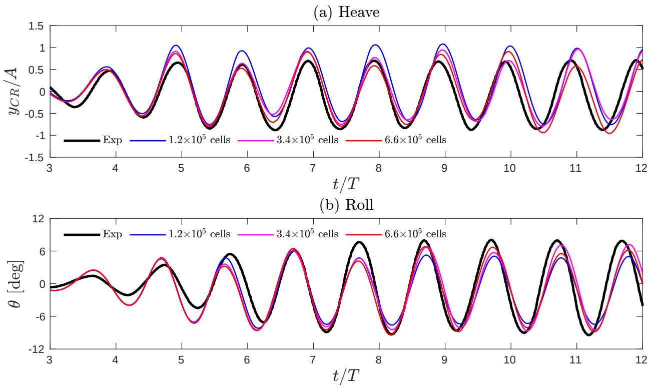

Figure 6: Time histories of (a) heave motion and (b) roll motion for mesh with 1.2×10^5 cells, 3.4×10^5 cells, 6.6×10^5 cells, and the experimental solution.

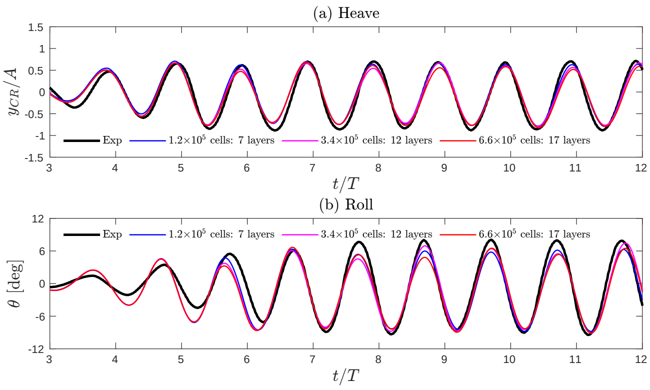

Figure 7: Time histories of (a) heave motion and (b) roll motion for mesh with 1.2×10^5 cells, 7 layers; 3.4×10^5 cells, 12 layers; 6.6×10^5 cells, 17 layers; and the experimental solution.

We consider first whether the updated hole-cutting procedure improves the accuracy of solutions, hence the original OpenFOAM v2212 donor-searching method is used first and not the new optimised method detailed in Section 2.3. To illustrate what this hole-cutting update actually does, consider Figure 4 which shows the T-shaped body boundary, and the overset mesh boundary, overlapped with the background mesh. Here we can see that the hole on the background mesh has been pushed back 7 'layers' of mesh cells away from the body with the interpolated cells being placed on the eighth layer. This is in contrast to older versions of OpenFOAM where the interpolated cells would automatically be placed on the first layer around the body. Figure 5 then shows the time histories of heave and roll motions for the OpenFOAM v2212 overset solver with the hole pushed back no layers, 7 layers, and 12 layers, along with the experimental Zhao and Hu solutions [14]. It is clear from Fig. 5 (a) that the heave motion with no layers is inaccurate with a combination of overprediction and underprediction. However, this is greatly improved when pushing the hole back 7 layers, with 12 layers showing a similar improvement. This suggests that pushing the hole back does indeed increase the accuracy of heave results but only up to a certain point at which the improvement remains static. In contrast, the corresponding results for the roll angle in Fig. 5 (b) are very similar, with slight discrepancies in the period of rotation beginning to show $t/T > 9$. This suggests that pushing the hole back does not necessarily improve the roll results. However, these results for the roll angle are significantly more accurate than the older OpenFOAM version in [2], regardless of whether the hole is pushed back from the T-body boundary or not. Therefore, there is indeed a significant improvement for this test case.

It is also useful to consider how changing the mesh resolution, as well as pushing the hole away from the body boundary, affects the corresponding solutions. To do this, a 'coarse mesh' with approximately 1.2×10^5 cells and a 'fine mesh' with approximately 6.6×10^5 cells are considered. Figure 6 then shows the time histories for each mesh along with the experimental solution. The solutions are to be expected with the coarsest mesh giving the most inaccurate results. However, Fig. 7 then shows analogous results with the hole now pushed back a number of layers equivalent to 12 layers on the original 3.4×10^5 mesh. The results here are now very similar, even on the coarsest mesh. This clearly indicates that pushing the hole back has a greater impact on improving accuracy than increasing the mesh resolution, meaning that a coarser mesh can be used to achieve accurate results—consequently increasing computational efficiency.

3.3 Analysis of computational time

It is also hypothesised that pushing the hole back may help to increase the efficiency of the solver. However, the results from this test case do not necessarily indicate this. Indeed, a pattern emerges when looking at the total

Case	No layers	7 layers	12 layers	17 layers
Execution time (s)	6057	5666	5883	5944

Table 2: Execution times for 2-D T-shaped floating body simulations with the hole pushed back 0, 7, 12, and 17 layers.

execution time for no layers, 7 layers, 12 layers, and an additional simulation using 17 layers (all listed in Tab. 2); a small but noticeable reduction in time can be seen between 0 and 7 layers. However, as the number of layers are increased further, a steady increase in time occurs, so much so that the execution time between 0 an 17 layers is almost the same. This indicates that the hypothesised reduction in execution time due to pushing the hole back does indeed occur but only up to a point. After this the time increases gradually due to the increased time spent pushing the hole back further. Regardless, these increases/decreases are very small and it is unclear whether these would be significant on a larger scale.

Now considering the new optimised donor-searching method, Table 3 lists the execution times and corresponding reduction in computational time as a percentage when using both the original (which we refer to as old) and optimised (referred to as new) methods on 1 and 2 nodes (128 and 256 processors respectively). On two nodes, a moderate reduction in computational time of 7.4% can be seen when using the new method over the old. In contrast, on one node, the reduction is almost double at 14.9 %. Both these results confirm what is hypothesised about the new method. First, the moderately sized reduction is to be expected for this simple 2-D case because the amplitude of motions is small, meaning that the necessity for remote donor searching is minimal. Moreover, the number of mesh cells is also small, meaning that the reduction in the number of cells for which donors have to be found is also minimal compared to much larger cases. Nevertheless, the reduction is still apparent. Second, the reduction being larger on one node is also expected for similar reasons in that a higher number of processors results in increased local donor searching rather than remote searching, so the primary effect of reducing remote searching is already happening. Table 4 also shows the execution times on 1, 2, 3, and 4 nodes (128, 256, 384, and 512 processors respectively). This indicates that the scalability for this case is poor; however, the case is simple and the number of mesh cells is not too high, so this sort of behaviour is to be expected because the time taken up by inter-processor communication is far exceeding the time taken for the OGA procedure. Note that the heave and roll results for both methods were identical so the figure has been omitted here.

Case	Old (2)	New (2)	Old (1)	New (1)
Execution time (s)	5666	5247	7841	6672
Reduction (%)		7.4		14.9

Table 3: Execution times and corresponding reduction in computational time as a percentage when using the old and new methods on 1 and 2 nodes.

No. of nodes	1	2	3	4
Execution time (s)	6672	5247	5367	5306

Table 4: Execution times when using the new method on 1, 2, 3, and 4 nodes (128, 256, 384, and 512 processors respectively).

4 Aerodynamic modelling of an offshore wind-turbine rotor

4.1 Computational setup

This second—more complex—test case considers the aerodynamics of the National Renewable Energy Laboratory (NREL) 5 megawatt (MW) wind turbine rotor [4] in constant wind. This test case was originally done using an older OpenFOAM version in Lin et al. [9], hence for comparison, the same geometric and computational setups are used here. The background mesh has dimensions $x \times y \times z \in [-30,\ 75] \times [-100,\ 100] \times [-100,\ 100]$ (m) with

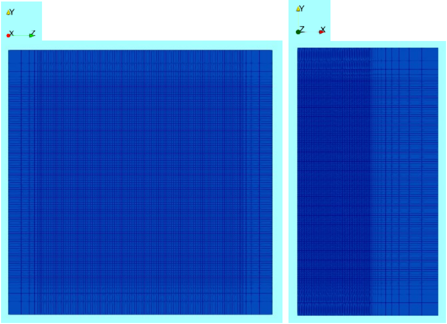

Figure 8: Background mesh in the yz (left) and xy (right) planes for the wind turbine test case.

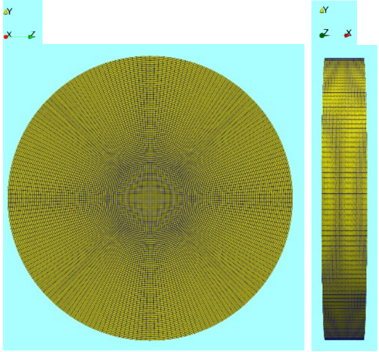

Figure 9: Overset mesh in the yz (left) and xy (right) planes for the wind turbine test case.

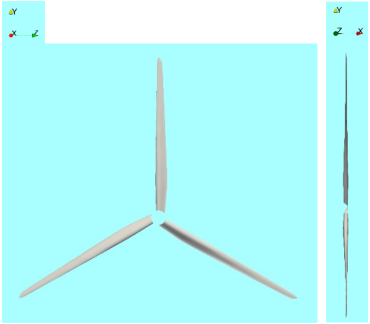

Figure 10: Wind turbine rotor blades in the yz (left) and xy (right) planes.

mesh configuration $N_x \times N_y \times N_z = 150 \times 120 \times 120$; Figure 8 shows the background mesh in the yz and xy planes. In addition, the cylindrical body-fitted overset mesh has a radius of 75 m in the yz plane with $x \in [-10,\ 10]$; the mesh configuration can be seen in Figure 9. The turbine rotor blades can also be seen in Figure 10 and are cut from the overset mesh using the `snappyHexMesh` utility. When modelling wind turbine aerodynamics, it is essential to accurately capture the turbulent boundary layer flow around the rotor, hence a sufficiently fine mesh is needed in the region close to rotor blade surfaces. In this work, to maintain a $y+$ value within a reasonable range of 30–600, the boundary layer mesh is generated in the vicinity of the rotor boundaries by extruding 15 layers of cells from the surfaces with a growth rate of 1.1 and first layer thickness of 0.001m. Figure 11 shows a cross section in the xz plane of the overset mesh close to the vertical rotor blade surface at $y = 16$ m; the boundary layer mesh can clearly be seen. The shear-stress transport (SST) $k-\omega$ model is used for turbulence modelling in the present work. The total number of cells is 2.16 million in the background mesh and approximately 2.44 in the overset mesh. Finally, a constant wind speed of 11 m s^{-1} and rotor speed of 1.2567 rad s^{-1} are adopted in the simulations, and the simulation time is 5 s (one revolution). Simulations are carried out on 8 Archer2 nodes (1024 cores) using the 'overPimpleDyMFoam' solver and the inverse distance interpolation method.

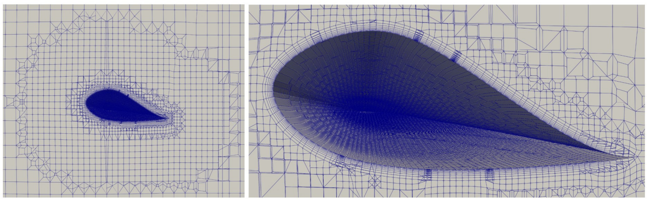

Figure 11: Cross section in the xz plane of the overset mesh close to the vertical rotor blade surface at $y = 16$ m.

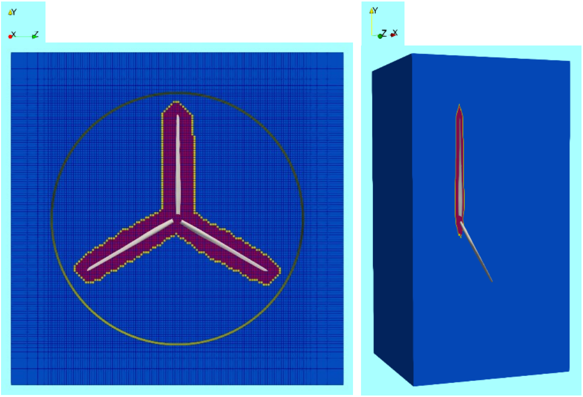

Figure 12: Background mesh with the hole pushed back 5 layers of mesh cells from the turbine rotor boundaries in the yz (left) and xy (right) planes.

4.2 Numerical results

We again first consider whether the updated hole-cutting procedure improves the accuracy of solutions; however, given the prior analysis from the T-body case, the new donor-searching procedure is now used instead of the original OpenFOAM v2212 method. In addition, it is well known that numerical simulations of complex unsteady flow problems are highly sensitive to the time step used in the model, so this needs to be considered alongside pushing the

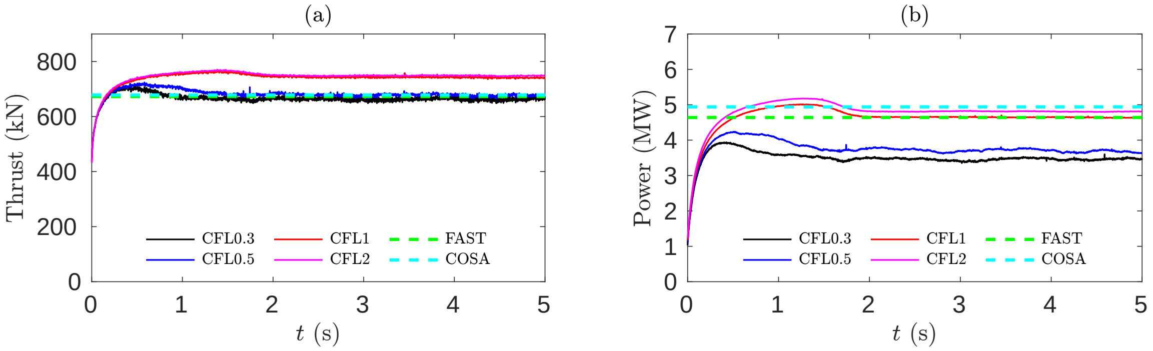

Figure 13: Time histories of (a) thrust and (b) power when pushing the hole back 5 layers for maximum CFL numbers 0.3, 0.5, 1 and 2, alongside the benchmark FAST and COSA code results [11].

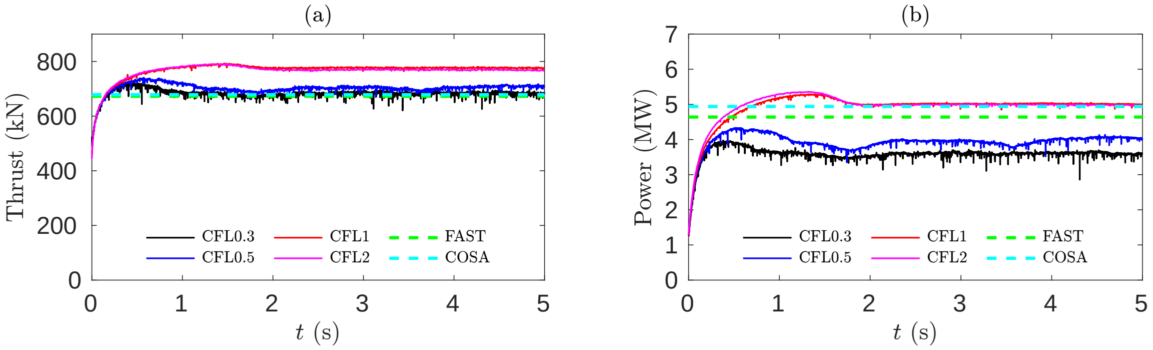

Figure 14: Time histories of (a) thrust and (b) power when not pushing the hole back for maximum CFL numbers 0.3, 0.5, 1 and 2, alongside the benchmark FAST and COSA code results [11].

hole back. The adaptive time-stepping procedure we use in OpenFOAM is dependent on the user-defined maximum CFL number, hence we consider different max CFL numbers—namely, 0.3, 0.5, 1, and 2—in the present work. The hole in this case is initially pushed back 5 layers; this is illustrated in Figure 12. Figure 13 then shows the time histories of (a) thrust and (b) power when pushing the hole back 5 layers for maximum CFL numbers 0.3, 0.5, 1 and 2, alongside the benchmark FAST and COSA code results [11]. The analogous 'no layer' results where the hole is not pushed back are shown in Figure 14. Considering the change in max CFL first, it is clear that using a smaller max CFL number (and consequently smaller time step on average) improves the thrust results in relation to the FAST and COSA codes, but the power results are majorly under predicted when using max CFL = 0.3 and 0.5. These results are somewhat counter-intuitive in that a smaller time step should lead to a more accurate and convergent solution, like we see with the thrust results. However, the time step is not the only factor that affects convergence—so too does the mesh: aerodynamics simulations like this are very sensitive to the quality of the computational mesh, particularly in and around the turbulent boundary layer. Unfortunately, it is difficult to create a high-quality mesh using the `snappyHexMesh` utility because it creates a mix of polyhedra cells rather than solely hexahedral; this causes problems with the aspect ratio, orthogonality, skewness, etc. Moreover, it is also difficult to add a proper boundary layer that keeps $y+$ values within a good—rather than sufficient—range. We hypothesis that a better quality mesh is required to properly predict the results for this case. Furthermore, the CFL = 1 and 2 power results look almost artificial as they appear to reach a quasi-steady state near the correct value even though the thrust is greatly over predicted. The aerodynamic relationship between thrust and power is correct in that a higher thrust will lead to higher torque, so given that the rotational speed and angle of attack are constant, the higher values must be caused by artificially high flow velocity. Given that the first-order Euler method is used in our OpenFOAM simulations, it is possible this artificial velocity is caused by temporal errors arising from the higher max CFL number (and time step). Looking at Figure 13, it is clear that the power will always reach a quasi-steady state so long as the thrust goes past a certain point, and to reach this point, the velocity has to be artificially high. In contrast, if the thrust does not go

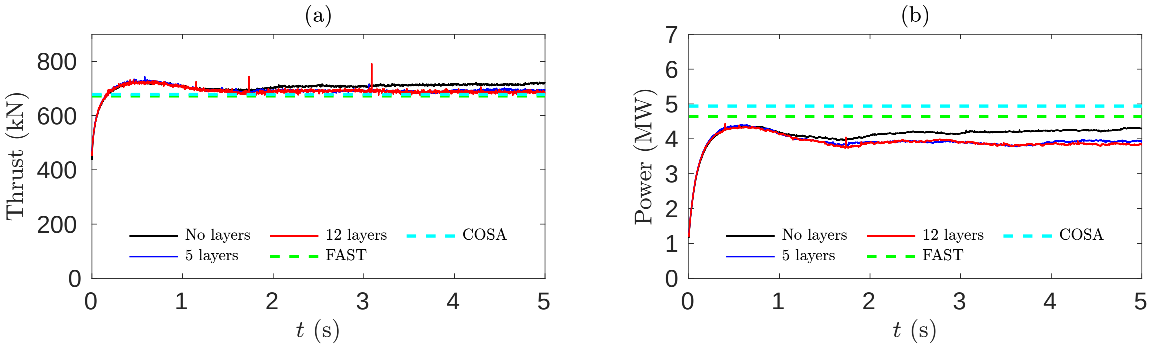

Figure 15: Time histories of (a) thrust and (b) power when pushing the hole back no, 5, and 12 layers for max CFL = 0.5, alongside the benchmark FAST and COSA code results [11].

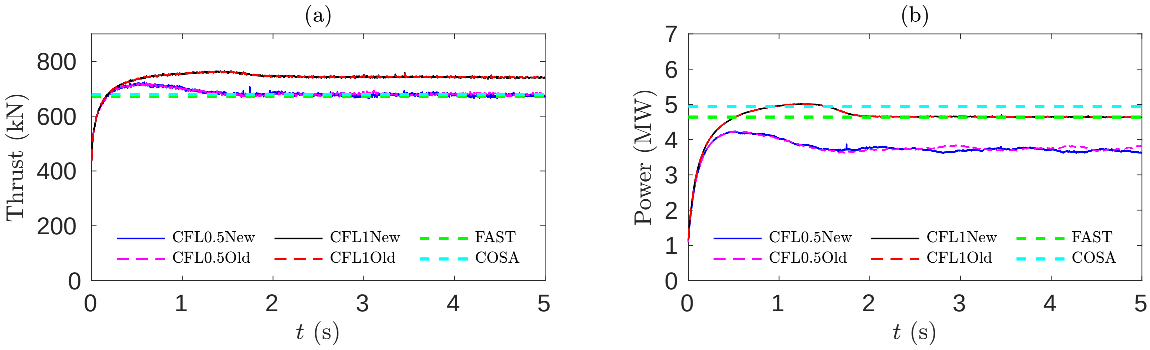

Figure 16: Time histories of (a) thrust and (b) power for both old and new methods when pushing the hole back 5 layers and for maximum CFL numbers 0.5 and 1, alongside the benchmark FAST and COSA code results [11].

past this point, the torque (and power) drops. This hypothesis is also backed up by results from previous work on this case [8] using the same circular disk shaped overset mesh. However, the authors also used a ‘body-fitted’ mesh where the overset interpolation both ways happened very close to the turbine body surface, something which is also likely to cause artificially high velocities and which consequently did turn out to be the case.

Now considering the affect of pushing the hole back, the overall picture looks similar for both no layer and 5 layer results. However, small improvements can nevertheless be seen when pushing the hole back. The most immediate improvement can be seen in the filtering effect observed by moving the hole back where the results are much smoother and less oscillatory. In addition, the thrust results are slightly improved for every CFL number, so much so that the results for CFL = 0.3 and 0.5 for 5 layers are almost identical to the benchmark results. It is also useful to see what happens if the hole is pushed back even further than 5 layers. To do this, both the background and overset meshes are extended so that the background mesh has new dimensions $x \times y \times z \in [-30,\ 75] \times [-120,\ 120] \times [-120,\ 120]$ (m) and the overset mesh has a radius of 90 m in the yz plane with $x \in [-12,\ 12]$. The number of mesh cells is also increased accordingly to 3.1 and 2.8 million for the background and overset meshes respectively. Figure 15 then shows the time histories of (a) thrust and (b) power when pushing the hole back no, 5, and 12 layers for max CFL = 0.5, alongside the benchmark FAST and COSA code results [11]. The behaviour here is similar to the T-body case in that the difference between the results for 5 layers and 12 layers is minimal, but both are different to the no layer results. Interestingly, the no layer results for this extended case are smoother than the original results in Figure 14 which suggests that the location of interpolation into the overset region can have a noticeable affect on the results. However, the overall behaviour of the no layer results are the same as the original.

4.3 Analysis of computational time

Now considering computational time, Table 6 lists the execution times and corresponding reduction in computational time as a percentage when using the old and new methods on 8 nodes (1024 processors) for max CFL = 0.5 and 1; Figure 17 (a) displays this in the form of a chart. (Note that, although we have discussed that CFL = 1 results may be artificial, they are listed here to demonstrate that the reduction in computational time is consistent with differing max CFL numbers.) For both CFL numbers, the new method reduces computational time by approximately 65 %, making it approximately 3x faster than the old method. Figure 16 also shows the time histories of (a) thrust and (b) power for both old and new methods when pushing the hole back 5 layers and for maximum CFL numbers 0.5 and 1, alongside the benchmark FAST and COSA code results [11]. This shows that the results for both methods are almost identical, but the new method is 3x times faster. This reduction in computational time is huge and clearly demonstrates the ability of the new method to speed up the computation for complex problems with continuously large displacements.

Case	CFL0.5: Old	CFL0.5: New	CFL1: Old	CFL1: New
Execution time (s)	344,512	122,014	171,603	57,566
Reduction (%)		64.6		66.6

Table 6: Execution times and corresponding reduction in computational time as a percentage when using the old and new methods on 8 nodes (1024 processors) for max CFL = 0.5 and 1.

Table 5 also shows the execution times and corresponding parallel speedup on 1, 4, 8, and 12 nodes (128, 512, 1024, and 1536 processors respectively); Figures 17 (b) and (c) illustrate this reduction in computational time and corresponding speedup respectively in the form of a graph. Linear speedup can be seen from 1 to 4 nodes but then drops off as the number of nodes increases further, indicating approximately 75 % parallel efficiency on 8 nodes and just 50 % on 12. This is clearly not ideal but is to be expected given our prior analysis during the T-body case which showed that the number of inter-processor communications will begin to take over the computational time after a

No. of nodes	1	4	8	12
Execution time (s)	667,425	164,102	112,014	107,405
Speedup	1	4.08	5.96	6.21

Table 5: Execution times and corresponding parallel speedup on 1, 4, 8, and 12 nodes (128, 512, 1024, and 1536 processors respectively).

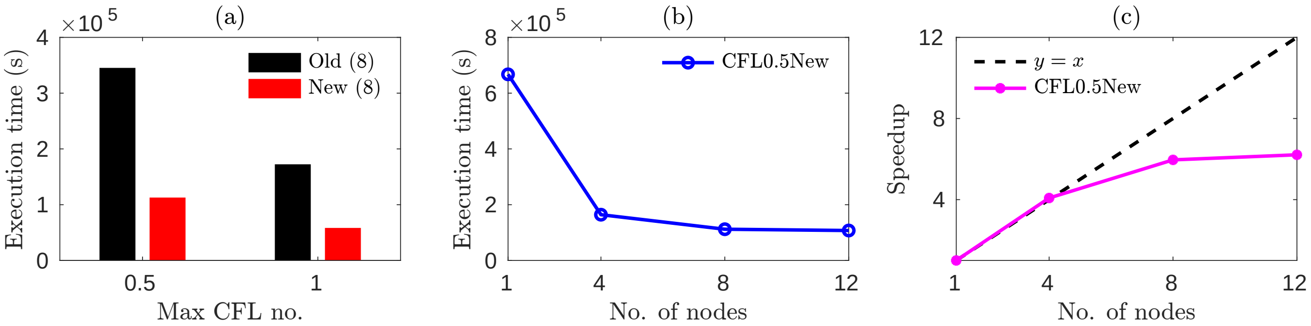

Figure 17: (a) Execution times for the old and new methods on 8 nodes with max CFL = 0.5 and 1; (b) Execution times on 4, 8, and 12 nodes for the new method with max CFL = 0.5; (c) Parallel speedup between 4, 8, and 12 nodes for the new method with max CFL = 0.5.

Case	CFL0.5: No layers	CFL0.5: 5 layers	CFL1: No layers	CFL1: 5 layers
Execution time (s)	112,014	115,522	54,766	57,566
Reduction (%)		3		-5.1

Table 7: Execution times and corresponding reduction in computational time as a percentage when the hole is pushed back no layers and 5 layers on 8 nodes (1024 processors) for max CFL = 0.5 and 1.

point. Indeed, there is practically no improvement between 8 and 12 nodes here. However, the execution time on 4 nodes is still less than half the time taken than the old method on 8 nodes, so the new method essentially cuts the required computational resources in half in addition to the reduction in computational time.

Similar to the T-body case, we can also check whether pushing the hole back helps to increase the efficiency of the solver. Table 7 shows the execution times and corresponding reduction in computational time as a percentage when the hole is pushed back no layers and 5 layers on 8 nodes for max CFL = 0.5 and 1. For max CFL = 0.5, there is 3 % reduction is computational by pushing the hole back. This looks small but still translates to approximately 1 hour, hence is not insignificant. In contrast, the max CFL = 1 results actually show a 5.1 % increase in computational time. However, this just backs up the point that max CFL = 1 results for this case are indeed artificial given that prior results for the T-body case also show a reduction in time when pushing the hole back. Results using the old method also show an identical pattern.

5 Conclusion

This technical report presents a new optimised version of the ESI-OpenCFD OpenFOAM overset mesh solver. A profiling investigation of the solver established that the method for donor searching was the primary efficiency bottleneck; hence, a new optimised method was developed that reduced the number of donors that must be found and consequently reduced the number of necessary inter-processor communications. The new method was validated for two test cases: regular wave interaction with a 2-D T-shaped floating body and the aerodynamic modelling of an offshore wind-turbine rotor in constant wind. It was found that the new method reduced computational time in the T-body case by 14% and 7 % on 128 and 256 cores respectively—a modest amount but to be expected for a simple case with only small amplitude motion. However, for the wind turbine case, the new method reduced computational time by 64.6 % on 1024 cores—approximately 3x faster whilst achieving practically identical results. The method was also show to have sufficiently good parallel efficiency; indeed, the execution time on 4 nodes took half the computational time of the old method on 8 nodes, so the new method essentially cut the required computational resources in half.

A secondary objective of this project was to evaluate the updated hole-cutting procedure released in OpenFOAM v2212. For the T-body case, it was found that pushing the hole back from the body boundary improved the solution for heave and roll motion of the body but had little impact on computational efficiency. For the wind turbine case, small but noticeable improvements were seen to both the accuracy of solutions and computational efficiency. However, it was concluded that a better computational mesh is required for a more rigorous analysis.

6 Acknowledgements

This work was funded under the embedded CSE programme of the ARCHER2 UK National Super-computing Service (https://www.archer2.ac.uk/ecse/). This project also made use of time on ARCHER2 granted via the UK High-End Computing Consortium for Wave Structure Interaction (HEC-WSI) (http://hec-wsi.ac.uk), supported by EPSRC (grant no. EP/X035751/1).

References

[1] M. Alletto. "Comparison of Overset Mesh with Morphing Mesh: Flow Over a Forced Oscillating and Freely Oscillating 2D Cylinder". In: *OpenFOAM® Journal* 2 (2022), pp. 13–30. DOI: 10.51560/ofj.v2.47.

[2] H. Chen, L. Qian, Z. Ma, W. Bai, Y. Li, D. Causon, and C. Mingham. "Application of an Overset Mesh Based Numerical Wave Tank for Modelling Realistic Free-Surface Hydrodynamic Problems". In: *Ocean Engineering* 176 (Mar. 2019), pp. 97–117. DOI: 10.1016/j.oceaneng.2019.02.001.

[3] ESI-OpenCFD. *v2212: New and improved numerics.* 2022. URL: https://www.openfoam.com/news/main-news/openfoam-v2212/numerics (visited on 09/12/2024).

[4] J. Jonkman. "Definition of a 5-MW Reference Wind Turbine for Offshore System Development". In: *National Renewable Energy Laboratory (NREL)* (2009).

[5] E. Katsidoniotaki and M. Göteman. "Numerical modeling of extreme wave interaction with point-absorber using OpenFOAM". In: *Ocean Engineering* 245 (2022), p. 110268. DOI: 10.1016/j.oceaneng.2021.110268.

[6] Z. Lin, H. Chen, L. Qian, Z. Ma, D. Causon, and C. Mingham. "Simulating focused wave impacts on point absorber wave energy converters". In: *Proceedings of the Institution of Civil Engineers-Engineering and Computational Mechanics* 174.1 (2021), pp. 19–31. DOI: 10.1680/JENCM.19.00038.

[7] Z. Lin, L. Qian, and W. Bai. "A coupled overset CFD and mooring line model for floating wind turbine hydrodynamics". In: *ISOPE International Ocean and Polar Engineering Conference.* ISOPE. 2021, ISOPE–I.

[8] Z. Lin, L. Qian, W. Bai, Z. Ma, H. Chen, J. Zhou, and H. Gu. "A Finite Volume Based Fully Nonlinear Potential Flow Model for Water Wave Problems". In: *Applied Ocean Research* 106 (2021), p. 102445. DOI: 10.1016/j.apor.2020.102445.

[9] Z. Lin, L. Qian, M.S. Campobasso, W. Bai, Y. Zhou, and Z. Ma. "Modelling aerodynamics of a floating offshore wind turbine using the overset mesh solver in openfoam". In: *International Conference on Offshore Mechanics and Arctic Engineering.* Vol. 85932. American Society of Mechanical Engineers. 2022, V008T09A031. DOI: 10.1115/OMAE2022-79230.

[10] P.J. Martínez-Ferrer, L. Qian, Z. Ma, D.M. Causon, and C.G. Mingham. "Improved Numerical Wave Generation for Modelling Ocean and Coastal Engineering Problems". In: *Ocean Engineering* 152 (Mar. 2018), pp. 257–272. DOI: 10.1016/j.oceaneng.2018.01.052.

[11] A. Ortolani, G. Persico, J. Drofelnik, W.A. Jackson, and M.S. Campobasso. "Cross-comparative analysis of loads and power of pitching floating offshore wind turbine rotors using frequency-domain Navier-Stokes CFD and blade element momentum theory". In: *Journal of Physics: Conference Series.* Vol. 1618. 5. IOP Publishing. 2020, p. 052016.

[12] R.S. Rai, Z. Ma, L. Qian, W. Bai, Z. Lin, and A. Khait. "A New Integrated Finite Volume–Finite Volume Numerical Model for Wave-Structure Interactions". In: *International Conference on Offshore Mechanics and Arctic Engineering.* Vol. 86892. American Society of Mechanical Engineers. 2023, V007T08A003. DOI: 10.1115/OMAE2023-103590.

[13] C. Windt, J. Davidson, B. Akram, and J. Ringwood. "Performance Assessment of the Overset Grid Method for Numerical Wave Tank Experiments in the OpenFOAM Environment". In: June 2018. DOI: 10.1115/OMAE2018-77564.

[14] X. Zhao and C. Hu. "Numerical and Experimental Study on a 2-D Floating Body Under Extreme Wave Conditions". In: *Applied ocean research* 35 (2012), pp. 1–13. DOI: 10.1016/J.APOR.2012.01.001.

Appendix A

The problem with scotch decomposition is that it decomposes the domain in such a way that the overset and background meshes lie on as many different processors as possible. For example, if using 10 processors, it may put all background mesh cells on processors 1-4, all overset mesh cells on processors 5-9, and a mixture of both background and overset cells on processor 10. This then consequently slows down the donor searching process because the part of the overset mesh that overlaps the background mesh—and on which the donor is found—lies on a different processor (and vice versa). This then increases the number of inter-processor communications because the donor has to be found remotely on a different processor. However, with hierarchical, the decomposition ensures that overlapping sections of both the background and overset meshes lie on the same processor, so only the local processor has to be searched to find the donor, reducing the number of inter-processor communications. This disparity in efficiency then grows as the number of processors increases because the number of inter-processor communications increases with the number of processors for scotch, whilst the number of cells which have to be searched locally goes down for hierarchical. One thing to note here is that the above applies for small to medium sized motion. For consistently large displacements (such a turbine), the difference in efficiency between each method becomes smaller because the donor cells move further away from their corresponding interpolated cells anyway, meaning that they are more likely to lie on different processors (when using hierarchical). This in turn increases the number of inter-processor communications as remote processors have to be searched. However, the new donor-searching method implemented in this works mitigates this.