Fourth international conference in numerical and experimental aerodynamics of, road vehicles and trains (Aerovehicles 4), Berlin, Germany, August 23-25, 2021

Aerovehicles 4-2021

Improvement of Arbitrary Mesh Interface (AMI) Algorithm for External Aerodynamic Simulation with Rotating Wheels

S. Vilfayeau*, C. Pesci*, S. Ferraris**, A. Heather**, and F. Roesler***
Corresponding author: sebastien.vilfayeau@esi-group.com

* Engineering System International GmbH, Germany
** ESI-OpenCFD, UK
***AUDI AG, Germany

Abstract: The new WLTP (Worldwide harmonized Light vehicles Test Procedure) [1] requires an evaluation of the aerodynamic performance of all vehicle configurations within a vehicle program under real conditions. Wheel rims present a challenge due to the number of variants, typically ten or more. A popular method in terms of accuracy and speed, to simulate rotating wheels is to use a sliding mesh approach., e.g. using Arbitrary Mesh Interfaces (AMI) [2]. However, it introduces additional computational costs compared to static mesh approximations. In this study, several improvements to the Arbitrary Mesh Interface (AMI) algorithm [3] have been implemented and tested. Initial results highlight that each development can decrease the computational time by up to 10%. The different developments are currently being assembled to evaluate the overall gain with a view to integration into a future OpenFOAM release.

Keywords: Aerodynamics, WLTP, Arbitrary Mesh Interface (AMI), numerical simulation.

1. Introduction

To include the WTLP regulation within the vehicle engineering process, vehicle manufacturers have to extend these requirements to the simulation processes, so that simulations include the effect of rotating wheels and moving ground. Wheels can contribute up to 25% of the total aerodynamic drag of a car [4]. Therefore, an accurate resolution of the physical effect of the wheel rotation is necessary. The simulation software used is the open source CFD toolbox OpenFOAM, developed and distributed by ESI-OpenCFD [5]. The pimpleFoam application used for these simulations is based on a Finite Volume Method solution of the incompressible Navier-Stokes equations. With the previous OpenFOAM version, i.e. v2012 (20 12), the turnaround time of cases with rotating wheels using sliding mesh and AMI was two to three times longer than standard cases with static wheels [6]. The increased turnaround time hinders its use in the early design phase of a car. This contribution describes developments that aim to significantly reduce the cost of the AMI algorithm without sacrificing accuracy. Improvements include optimization of the face area intersection algorithm, implicit treatment of the cyclicAMI boundary condition, and mesh motion optimization. Gain in performance attributed to each development will be evaluated on simplified test cases, such as a propeller and single wheel assembly, as well as industrial cases with a full car model and different set of rims.

2. Numerical Method

2.1. Face Area Intersection Algorithm for AMI

Arbitrary Mesh Interface (AMI) cyclic patches enable communication across topologically dissimilar patches, by constructing interpolation weights based on the fractional overlap of faces on either side of the coupled patch. Interpolation weights and addressing are assembled using an advancing front of patch faces that, for each candidate pair of source and target faces includes all edge connected neighbor faces. This repeats until all patch faces have been visited. The fractional overlap calculation is the most expensive part of the algorithm, typically accounting for half of the cost of the AMI. The current implementation uses an 'ear clipping' method based on the Sutherland-Hodgman algorithm [7] that requires splitting face (triangles) with geometric planes aligned with the face edges. Although effective, the computation is reasonably costly and prone to numerical error. Within this contribution we will present alternative methods to calculate the intersection areas and assess their performance to replace the current approach if shown to be beneficial. Four new different methods are implemented and tested:

- faceAreaWeightAMI_2D: 2-D variant of current (3-D) algorithm; advancing front search
- faceAreaWeightAMI_2D_octree: 2-D variant of current (3-D) algorithm; octree search
- faceAreaWeightAMI_3D_octree: 3D algorithm; octree search
- faceAreaWeightAMI_2D_CGAL: 2-D variant using the CGAL algorithm; octree search

2.2. Implicit Treatment of cyclicAMI Boundary Condition

The current cyclicAMI boundary condition is treated semi-implicitly. This approach introduces errors that make the system of equations more difficult to solve, often requiring additional solver iterations or introducing perturbations that can lead to unboundedness and non-physical behavior, with a net effect of significantly increased run times. Indeed, the sum of the interpolation weights per AMI patch face should sum to unity. However, these weights often incur an error, typically arising due to coupling across curved patches or poorly matched geometrical features. We have investigated how to implement a fully implicit version of the cyclicAMI boundary condition. This should lead to increased stability with much fewer solver iterations, and reduced run times.

To do so, we have examined the possibility to convert the AMI patch faces into equivalent internal mesh faces. The approach will be based on reformulating the pressure equation matrix using a new addressing structure such that AMI contributions are treated fully implicitly. Investigations aim to determine the optimal method to connect cells across cyclicAMI patches, based on the application of the existing AMI addressing and interpolation weights to formulate new 'artificial' faces. This has the potential advantage of avoiding the need to perform topological mesh changes to enforce a consistent view across the patches. In addition, the fully implicit method should perform much better than the current semi-implicit method. A similar approach has been successfully applied to assemble the energy equation for conjugate heat transfer (CHT) cases, where the convergence rate was significantly improved. This proof-of-concept has been generalized to other solvers (GAMG), and the final code will be integrated in the standard OpenFOAM library.

2.3. Solid Body Motion Optimization for AMI

Moving mesh cases incur significant costs compared to their static mesh counterparts even when the motion is set to zero. For instance, consider the propeller tutorial run with fixed mesh; for this case it can be shown that the dynamic mesh version performs approximately 40% slower. Contributions to the cost of moving mesh calculations include:

- the clear-out and recreation of mesh data, e.g. base mesh information, AMI weights and addressing, calculation stencils, agglomeration maps, etc.
- calculation of derived fields, e.g. the mesh flux (meshPhi), field mapping etc.

Solid body motion does not change the mesh topology, e.g. for rotating wheel cases the mesh outside the wheel region remains fixed whilst the inner region simply rotates without changing its internal structure. Nevertheless, being a dynamic mesh case, the mesh will undergo all the costly updates listed above, including the non-moving parts of the domain.

We have implemented a new update process for solid body rotation cases that keeps static geometric data intact and only updates necessary components, in the expectation that this will reduce run-times. Note that the mesh updates contribute to the cost of each time step, but the overall cost is dominated by solving the flow equations (and potentially post-processing operations). Therefore, in a case where, for instance, more pressure iterations are required, the contribution of the mesh updates to the run time will be smaller. Thus, in our presentation, we will consider cases for which we can show the performance gain just due to this development, including both small tests like the propeller or single wheel, as well as the industrial test case with the full car geometry.

3. Reference Cases

The propeller case is a standard OpenFOAM tutorial where the AMI patch-pair is described by 18496 and 18720 source and target faces, respectively. Calculations are performed in parallel on 4 CPUs using scotch domain decomposition.
The single wheel assembly case was supplied by Audi AG as illustrated in Fig. 1. The AMI patch-pair (shown in purple) is described by 23729 and 24117 source and target faces, respectively. Calculations were performed in parallel on 120 CPUs using the ptscotch domain decomposition method.
Please note that performance of the sliding mesh simulation depends on the face count of the AMI interface rather than the overall cell count. Therefore, we expect that AMI developments will show better performance as the case size increases.

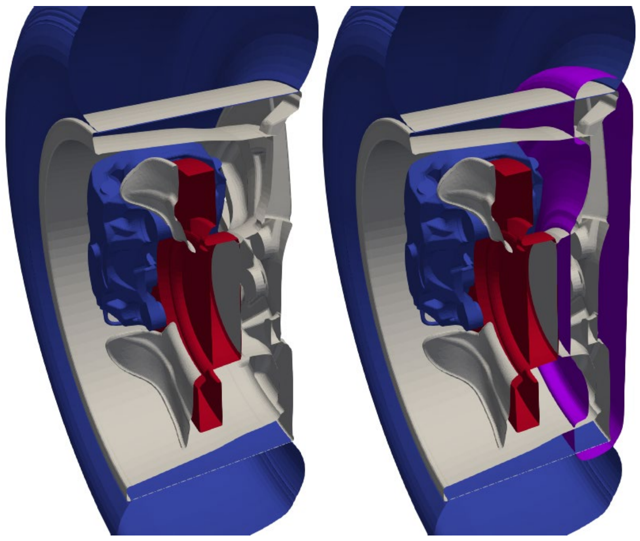

Figure 1 Single wheel assembly. The AMI interface is depicted in purple, tire in blue, and rim in grey.

4. Results and Discussion

4.1. Face Area Intersection Algorithm for AMI

In this section we compare the various algorithm introduced in 2.1. As shown in Table 1, changing to 2D intersection algorithm improved timings by approximately 6%. However, the largest improvement of 30% was observed when replacing the advancing front by an octree method. This is partially due to less loops in the algorithm, less intermediate storage and less data management. By combining both development (2D intersection algorithm and octree search), a cumulative improvement of 36% for the static single wheel case has been observed. Unfortunately, CGAL (Computational Geometry Algorithms Library) has shown to be very slow compared to the Sutherland-Hodgman algorithm and therefore not viable for AMI. With the moving propeller case, approximately 7% reduction in execution time is observed, as reported in Table 2.

Table 1 : static single wheel case (100 iterations, fixed time step, no FO, no I/O)

Algorithm	Average Time over 5 runs (s)	Delta Average (%)
faceAreaWeightAMI	44.614	0.00
faceAreaWeightAMI_2D	41.901	-6.08
faceAreaWeightAMI_3D_octree	30.838	-30.88
faceAreaWeightAMI_2D_octree	28.312	-36.54
faceAreaWeightAMI_2D_CGAL	2767.06	+6100.00

Table 2 rotating propeller case (100 iterations, fixed time step, no FO, no I/O)

Algorithm	Average Time over 5 runs (s)	Delta Average (%)
faceAreaWeightAMI	761.50	0.00
faceAreaWeightAMI_2D	756.50	-0.6
faceAreaWeightAMI_3D_octree	713.75	-6.3
faceAreaWeightAMI_2D_octree	709.75	-6.8

Table 3 rotating single wheel case (100 iterations, fixed time step, no FO, no I/O)

Algorithm	Average Time over 5 runs (s)	Delta Average (%)
faceAreaWeightACMI	1299	0.00
faceAreaWeightAMI_2D_octree	1248	-3.9

Initial investigations undertaken to reduce the cost of constructing AMI addressing and weights and improve the overall procedure have been successful. Thus, the development presented here has been employed to simulate a full car geometry with different set of rims. Initial tests have shown a 10% improvement in turnaround time.

4.2. Implicit Treatment of cyclicAMI Boundary Condition

The implicit treatment of cyclicAMI has been implemented in OpenFOAM and initial tests with the propeller and single wheel cases have shown an improvement of 5 to 10% in the execution time. Based on these successful preliminary tests, the new implicit AMI framework has been integrated into the development branch aiming for release in v2112. Further tests with the full car model are currently being performed and will be presented at the conference.

References

[1] https://www.wltpfacts.eu/

[2] L. Haag, M. Kiewat, T. Indinger, T. Blacha, *Numerical and experimental investigations of rotating wheel aerodynamics on the DrivAer model with engine bay flow*, ASME, 2017

[3] P. E. Farrell and J. R. Maddison, *Conservative interpolation between volume meshes by local Galerkin projection*, Comput. Methods Appl. Mech Engrg, vol. 200, p. 89-100, 2011

[4] B. Schnepf, T. Schütz, and T. Indinger, *Further Investigations on the flow around a rotating, isolated wheel with detailed tread pattern*, SAE International Journal of Passengers Cars, vol. 8(1), p. 261-274, 2015

[5] https://www.openfoam.com/

[6] F. Rösler, *External aerodynamics with rotating wheels*, German OpenFOAM User Meeting, 2018

[7] I.E. Sutherland, G.W. Hodgman, *Reentrant polygon clipping*, Communications of the ACM, vol. 17, no 1, p. 32-42, 1974