NAFEMS World Congress 2015
San Diego, June 2015, Proceedings, ISBN-13: 978-1-910643-24-2

COMPETING WITH A CAD EMBEDDED CFD SOFTWARE AGAINST TRADITIONAL CFD CODES IN A BLIND JSAE BENCHMARK TO PROVE RESULT ACCURACY

B. Marovic
(Mentor Graphics, Germany)

1. Abstract

It has been almost 20 years since the first CAD embedded CFD codes appeared on the market and yet the accuracy of such codes is still questioned. The current development cycles require fast simulation results in order to guide new design decisions based on these results. Unfortunately traditional CFD tools are not able to comply with the needs for the requirements of such simulation driven design methods. The attempts to increase the ease of use by simplifying and/or embedding these traditional tools in CAD systems often lack in technology to reduce the needs for highly sophisticated meshes, the need to wisely choose from a range of turbulence models and the lack of solver stability for a converged solution. These disadvantages increase the requirements in numerical knowledge of users of such tools and yet do not reduce the workload of manual meshing of most complex CAD geometries and monitor and control solver convergence.

CAD embedded CFD codes that comply with the requirements of the simulation driven design trend might often appear on first look as inaccurate based on the methods applied by them and therefore labelled as hoax to the industry and unable to deliver high result accuracy. What seems to be unsuitable methods of meshing and oversimplification of complex numerical tasks, however bare high-end numerical schemes behind the curtains of ease-of-use and algorithms that guarantee high accuracy results where CFD experts would not trust their results.

This paper will briefly show the applied technologies of meshing and solver technology for the code FloEFD™ that was applied in the blind JSAE benchmark [1] against traditional CFD codes. This blind benchmark is about an external aerodynamics simulation of a simplified car body similar to the famous Ahmed Body. Participating in this

benchmark were well known commercially available CFD codes shown by name with their applied meshing methods, cell count, turbulence models, CPU times and results compared to the actual physical test results. This paper will show the result accuracy of the applied CAD embedded CFD code with its meshing and solver technology that makes CFD simulation easy to define and execute for the ideal application in today's simulation driven design trend.

2. CAD Embedded CFD

The industry undergoes a drastically change at the moment where the term "Simulation Driven Design" is used by many simulation tools. The term basically describes the trend to use simulation results to make judgments on possible regions of the design that should be improved in order to increase the products efficiency, durability and reliability. The basis of every simulation is at least an idea of the design in order to start with a one dimensional sketch like a fluid network for a 1D simulation. In case of a 3D design a first 3D model is necessary and since those engineers that do these designs are working with CAD tools, the obvious choice is of course a CAD system to do that. Although most 3D CFD software are able to create 3D models either by a simple 3D modeller or with the mesh boundaries themselves, this is not the typical tool those engineers use.

The next step after a 3D model is created is of course to test if the designed idea or concept actually delivers the desired performance before going into even a more advanced and detailed design which usually takes several days if not weeks depending on how mature the design will be. But the most useful state of the process is the early state where not too much effort was put into the design in order to avoid wasting a lot of time creating a detailed design that has to be re-done almost completely because the performance is really poor. So with every simulation the design shapes into a better performing product and in such a concurrent process where design and simulation interpolate into a mature high performing product, the CFD method is also called "concurrent CFD".

Now such a concurrent CFD approach in the modern world has to work like its final product itself – highly efficient with the least amount of friction to the process. Such an efficient process is obviously a CAD embedded simulation approach. Since the CAD model is the basis, the simulation can directly build up on that basis and "escort" the design safely through the process dodging high costs, many prototypes and bad quality and efficiency. A separate approach of doing the CFD simulation will require to export the geometry often into a neutral format and then let it be analysed by a separate simulation team which is busy with other simulation from other teams already. This means usually the new project gets into the queue of projects to be calculated and if you are lucky you'll receive an answer in 3-4 weeks. That of course is not efficient and especially if the engineer has ideas of playing with parameter to see the influence on the performance and play with those ideas in parametric studies or what-if analysis. This would be almost impossible or very interrupted due to the handling of it by a different group. He would not have the direct influence to change anything if the performance trend goes into the wrong direction.

However, a CAD embedded CFD approach requires a user friendly user interface and a reduction of the numerical challenges that usually the experts face. Such challenges range from complex terminology that the engineer is not familiar with over manual mesh generation and modification to complex solver settings such as a vast range of turbulence models and solver convergence control parameters. Most engineers graduating from university or college with an engineering degree do have the fundamental understanding of thermodynamics and fluid flow but not necessarily the higher advanced numerical and mathematical understanding of such simulation tools like CFD. These engineers develop new devices such as valves, pumps, vacuum cleaner, automotive or aerospace engines, electronics, HVAC systems and ovens and therefore understand the way their products work and their boundary conditions. But they are not simulation experts.

So providing those engineers with a tool that makes them capable of reliably performing simulations during their design process and make early decisions on changes in the design is important for such a CAD embedded CFD approach. This requires special meshing and solver methods that enable the engineers to do exactly that without worrying about things they don't understand from the high end numerical technology.

a. Meshing Technology for CAD Embedded CFD

The meshing technology required for such a CAD embedded approach has to fulfil some important criteria in order to be suitable for this task. These criteria are mainly:

- Automated meshing in order to reduce the manual necessary work to a minimum which usually can take up to several weeks.

- Handling most complex geometries, as they are often are created in CAD systems, automatically without manual meshing work.
- Creation of a high quality mesh without laborious manual work in order to provide high accuracy when compared to measurements.

The capability to mesh automatically complex CAD data and generate a high quality mesh easily is very important as the user should not be occupied by the usual long meshing times from manual meshing processes and also does not have the experience of creating a high quality mesh as it is often necessary with body fitted meshes where the boundary layer has to be resolved. Parameters such as Y+ are usually unknown to him and he wouldn’t know how to define them and consider for a high quality mesh.

Taking a step back and considering the most perfect mesh for the CFD solver itself we would find that the Cartesian mesh is the best suited mesh. Reason for that is the perfect alignment of the cell faces to the main coordinate system in which the equations of the solver are defined. Now the problem that would arise with any geometry that is not conform to a rectangular shape and aligned to the coordinate system would be the representation of such geometry with the mesh. Various solutions propose often a stair-step representation of the body with the mesh, a tetrahedral or similar [2] resolution at the body’s surface in order to keep its original surface and transition to the Cartesian mesh or an overset mesh or also known as Chimera (derived from the Greek mythology of a hybrid beast composed of parts from more than one animal). Often these solutions are good to represent a more complex body in a structured mesh but do also bare problems such as issues with skewed cells or interpolation errors in case of the overset mesh. Usually such solutions also don’t resolve the boundary layer automatically and manual work is necessary to create a suitable mesh.

The partial cell technology however enables the use of a Cartesian basic mesh that is simply cut at the intersection of a cell with a body splitting the cell into two sub-control volumes, one being solid and the other being fluid. This method can be applied for very small features such as thin layers of material with different properties as well and will create several sub-control volumes in order to represent the geometry even within one cell. Of course it is not possible to use one single cell for the whole computational domain as someone might start to dream. A somewhat reasonable resolution of the geometry is of course necessary as it does rely on cell or sub-control volume centre parameters for pressure and temperature etc. But with this method it is possible to use a Cartesian mesh and automatically mesh the geometry as the CAD model is present and can be used to split the cell into sub-control volumes it can be automated easily.

Similar approaches are known from the cut-cell method and do in some way fall under the Immersed Boundary (IB) method. However these methods are not the same through and through.

b. Solver Technology for CAD Embedded CFD and its Meshing Technology

The next step after a created mesh is of course the solver and its capabilities. Including all type of physics is more or a mathematical definition than a solver problem, except for the correct implementation in order to get accurate results and ease-of-use. Physics is defined by equations and can be implemented in every code, in some easier than in others but we wouldn’t be humans if we weren’t creative and love solving problems. So getting the necessary physics into a code is just some coding and mathematics and here and there a genius mind, but often these capabilities are implemented for experts and they make it possible to calculate it but also leave the user to the task to define often a large range of parameters that are hard to find in any handbook or datasheets if material or fluid parameters are necessary. So the effort of putting the physics and simplicity into the code is often not done by vendors of the traditional CFD codes.

Now considering the mentioned meshing technology we initially lack the capability of resolving the boundary layer in detail. There are already methods that separate the flow into far field flow and near wall flow but usually not applied in CFD, rather the user of traditional CFD tools use a very fine mesh to resolve the boundary layer and the flow near the wall is calculated together with the far field. Here the advantage of the single flow field definition is that you can use traditional methods and algorithms but the disadvantage is you need a lot of cells to resolve the boundary layer and a body fitted mesh is preferable but cost a lot of manual work for a good quality of mesh. The method of splitting the two flow region reduces the amount of cells necessary for the calculation of the flow and works on any mesh. But also has its disadvantages such as the need for special algorithms for the near wall flow and the need for boundary conditions of the far field flow or core flow.

The methods applied in FloEFD separate the boundary layer into two types, the thick boundary layer and the thin boundary layer [3], not to be mistaken with the actual thickness of the boundary layer. These two types are named thick and think relative to the number of cells as the Figure 1 shows.

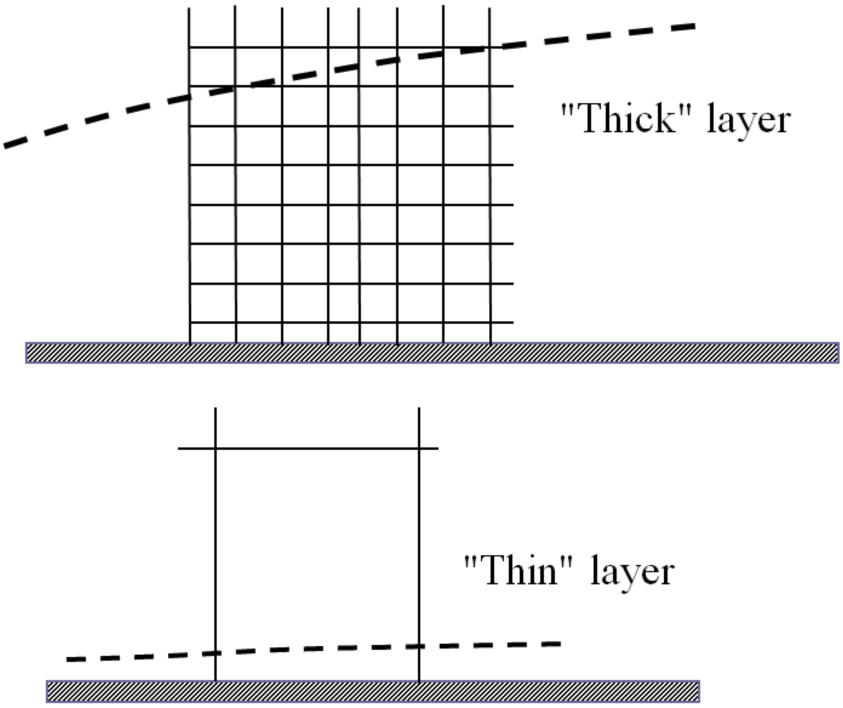

Figure 1: Thick and thin boundary layer as used in FloEFD.

With a thick boundary layer type the boundary layer is resolved by more than 6 cells and with the thin boundary layer the boundary layer has less than 3 cells. For the range in between 3 and 6 cells an interpolation is applied and the two types need different algorithms.

The thick boundary layer uses the wall function of the Van-Driest Profile which accounts for the laminar, turbulent as well as the transitional buffer zone. The usual wall functions work for either the laminar or turbulent but not both boundary layer types, neither for the transition zone. The wall function for the thick boundary layer type in FloEFD includes the influence of wall roughness, shear stress and heat flux, compressibility effects and the k-ε turbulence parameters are also considered.

The tin boundary layer type applies an integral approach of the boundary layer in which the Prantl equations are integrated over the thickness of the boundary layer. For that the assumptions are made that the physical properties over the thickness are constant and the vector of the shear stress on the wall is parallel to the velocity vector outside the boundary layer. This method has the advantage that it can calculate very thin boundary layers and can be adapted to Cartesian meshes very easily. It also enables the consideration of a large range of Reynolds numbers and a calculation with sufficient accuracy. It considers also the wall roughness, compressibility effects and buoyancy effects from gravitation and centrifugal forces. The limitation however is that a 2D boundary layer model is used and it does not allow any correct evaluation of flow parameters near the wall.

This approach enables the use of the meshing technology mentioned before and gives accurate enough results for about 90% of all real industry applications engineers are designing and need simulating, neglecting any physics more than flow and heat transfer. Additional physics as mentioned before can be implemented such as radiation models, film condensation, combustion, joule heating, rotation, cavitation etc. With the single turbulence model applied in FloEFD, the k-ε turbulence model, it uses the most used turbulence model also applied in the industry for industry case simulations. The applied k-ε turbulence model is not the standard model or but a modified model that increases the use of the model to a wider range of applications compared to its known types.

3. Proof of Validity in a Blind JSAE Benchmark

FloEFD and its applied methods and technologies are often initially dismissed as “Mickey Mouse” tool or not accurate. But as the countless industry applications and successes have proven it works very well and highly efficient with very good result accuracy. This is again proven in the JSAE blind benchmark “Benchmark of Aerodynamics CFD of Simplified Road Vehicle Model”.

The benchmark describes a similar benchmark model such as the Ahmed Body (Figure 2) and the geometry was made available by the organizers without and result information, hence a blind benchmark.

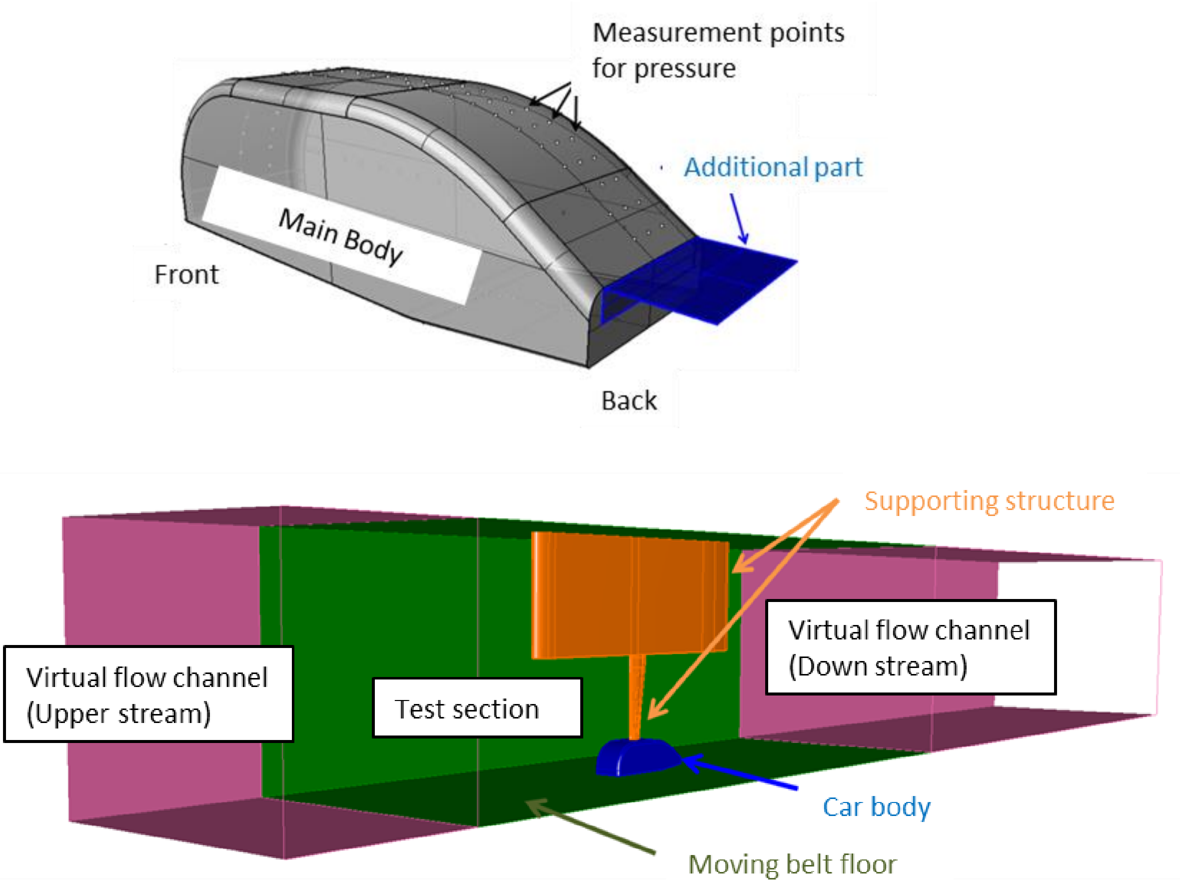

Figure 2: JSAE blind benchmark model geometry and testing setup.

a. The Benchmark Task

The model in Figure 2 consists of the vehicle body itself with a full length of 1,100mm without the additional part at the end of the vehicle and with a length of 1,250mm with the additional part. The height of the vehicle was 355mm, the width 320mm and the underfloor height was 15mm. The model was simulated in a test chamber similar to a wind tunnel at a velocity of 25.0 m/s. The fluid properties were given with a density of 1.17 kg/m^3 and a kinematic viscosity of 1.56 x 10^{-5} m^2/s which results in a Reynolds number for the test of 1.76 x 10^6.

As for the result, the simulations were to provide results for drag, lift and pitch-moment coefficient as well as pressure coefficient at various sections of the body. Sections vertically to the car were compared to measurements at the centre plane (y/W=0.0), 12.5% off centre (y/W=0.125) and 25% off centre (y/W=0.25) where W is the width of the body. The underfloor section was vertically only analysed for the centre plane as it was not disturbed by the wind tunnel fixture as the top side was. The section horizontally to the car was compared at 25% (z/H=0.25) of the car height as shown in Figure 3 where H is the height of the body.

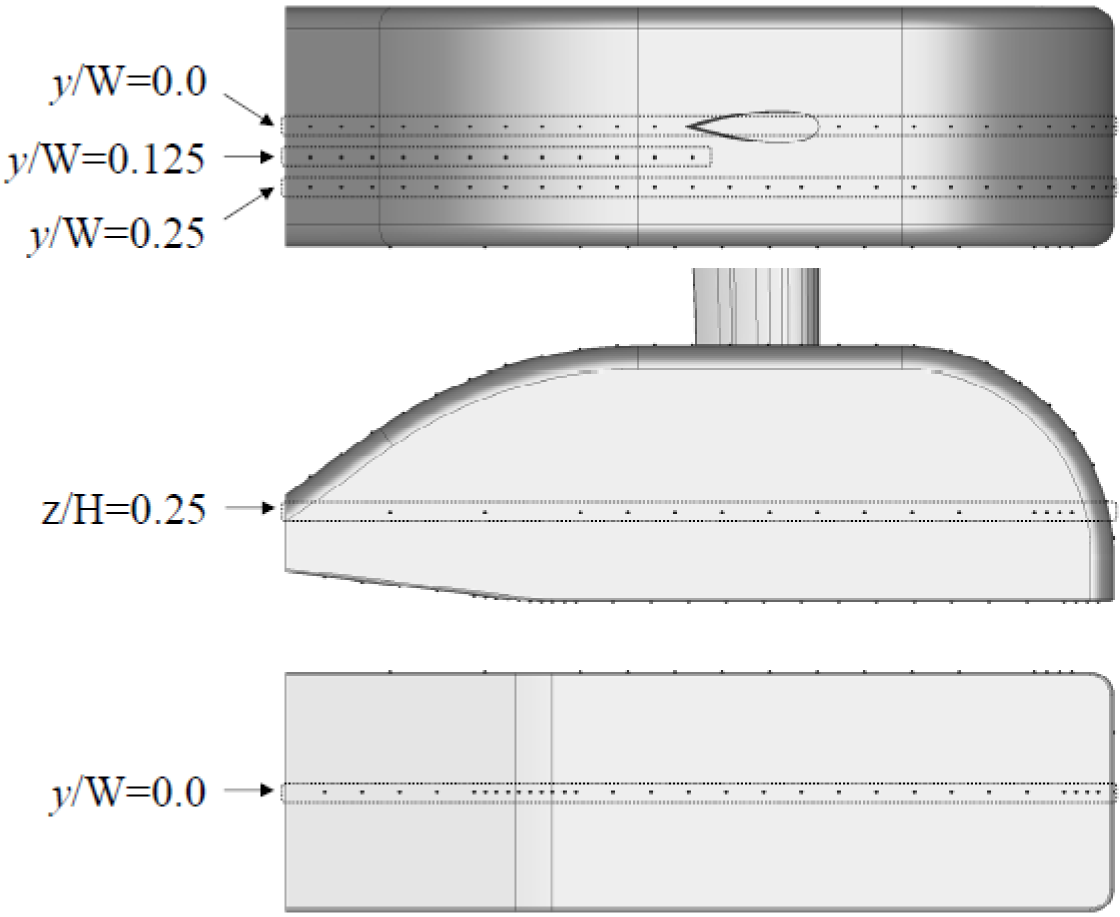

Figure 3: Pressure coefficient measurement point distribution on the body.

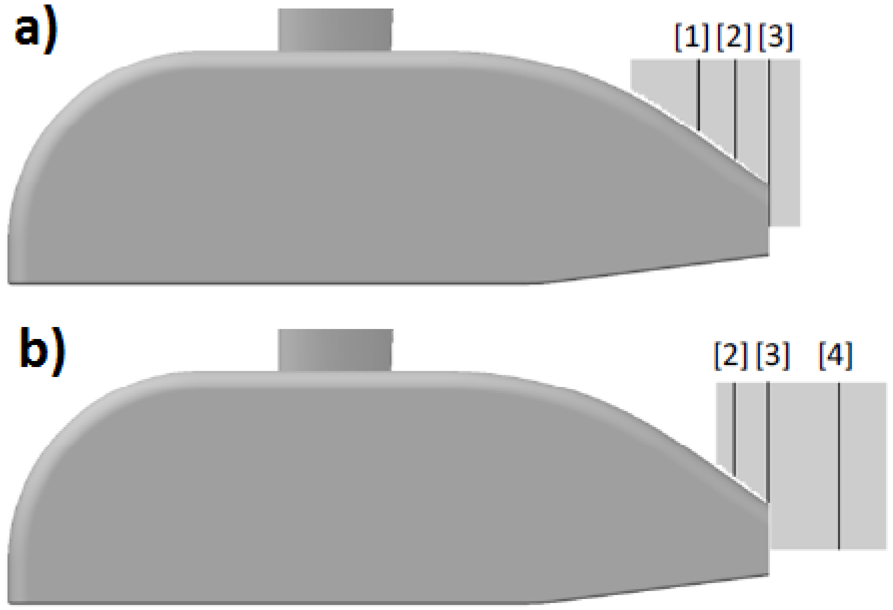

Figure 4: Wake measurements at y/W=0.0 for a) the case without the additional part and b) the case with additional part.

The wake of the model was analysed at the vertical lines of x=1,000mm (line 1), x=1,050mm (line 2), x=1,100mm (line 3) and x=1,200mm (line 4) and compared to measurements.

b. The Competition

The following tables show the participating companies with their selected tools of choice for the benchmark and the applied methods from meshing to turbulence models.

Participants	Software	Compressible/ Incompressible	Steady State/ Transient	Turbulence model
JSOL Corporation	AcuSolve	Incompressible	Steady State	Spalart Allmaras
ANSYS Japan	ANSYS Fluent R14.5	Incompressible	Transient	Scale Adaptive Simulation (SAS)
KKE	FloEFD	Compressible	Steady State	Modified k-ε
Icon Technology & Process Consulting Ltd.	iconCFD	Incompressible	Transient	Spalart Allmaras
ESI Group	PAM-FLOW	Incompressible	Transient	SGS
CRADLE	SCRYU/Tetra	Incompressible	Transient	SST-DES
				SST-SAS
CD-adapco	STAR-CCM+ v7.06	Compressible	Transient	IDDES (SST)
		Incompressible	Steady State	SST k-ω

Table 1: Participating companies and their tool of choice including solver method and turbulence model.

Software	**Mesh type**	**Number of layers in boundary layer**	**Number of cells**	**Mesher**
AcuSolve	Tetrahedral mesh	7 layers	Case 1: 24,755,000 Case 2: 25,795,000	AcuConsole 1.8b
ANSYS Fluent R14.5	Unstructured grid	17 layers	Case 1: 16,000,000 Case 2: 16,700,000	ANSYS Meshing R14, TGrid R14
FloEFD	Cartesian mesh based on octree technology	-	3,520,000	FloEFD
iconCFD	Hexahedral dominant mesh	7 layers	Case 1: 37,640,000 Case 2: 38,300,000	foamProMesh
PAM-FLOW	Tetrahedral mesh	6 layers	38,260,000	PAMGEN3D
SCRYU/Tetra (DES, SAS)	Tetrahedral mesh with prisms	10 layers	27,000,000	SCRYU/Tetra
STAR-CCM+ v7.06 (IDDES, SST k-ω)	Hexahedral dominant mesh	20 layers	Case 1: 16,690,000 Case 2: 16,835,000	STAR-CCM+ v7.06

Table 2: Mesh type, boundary layer resolution and meshing tool.

Green highlighted in the tables is FloEFD and as one can see the cell count is extremely (4.5 to 11 times) lower than the other CFD tools, which is due to the use of the partial cells that can contain several sub-control volumes and does not need a fine resolution of the boundary layer as other tools do. Also one can see that a large number of calculations have been conducted in transient which usually results in very high CPU time for the calculation even with a high number in cores considering the large number of cells. More information on the setup can be found in the JSAE benchmark paper.

c. The Results

The results of FloEFD are based on a Cartesian mesh as already described with a solution adaptive refinement on octree basis and local meshes around the body. The mesh used for the calculation can be seen in Figure 5 below. Each cell level refinement is set very easy with a number of the level defined by the user. The approach is therefore very simple to understand and automated in the rest of the mesh generation. The adaptive refinement can also be limited by a maximum level for the cells and a limited cell number in order to not explode in the mesh size and to cause too high CPU times.

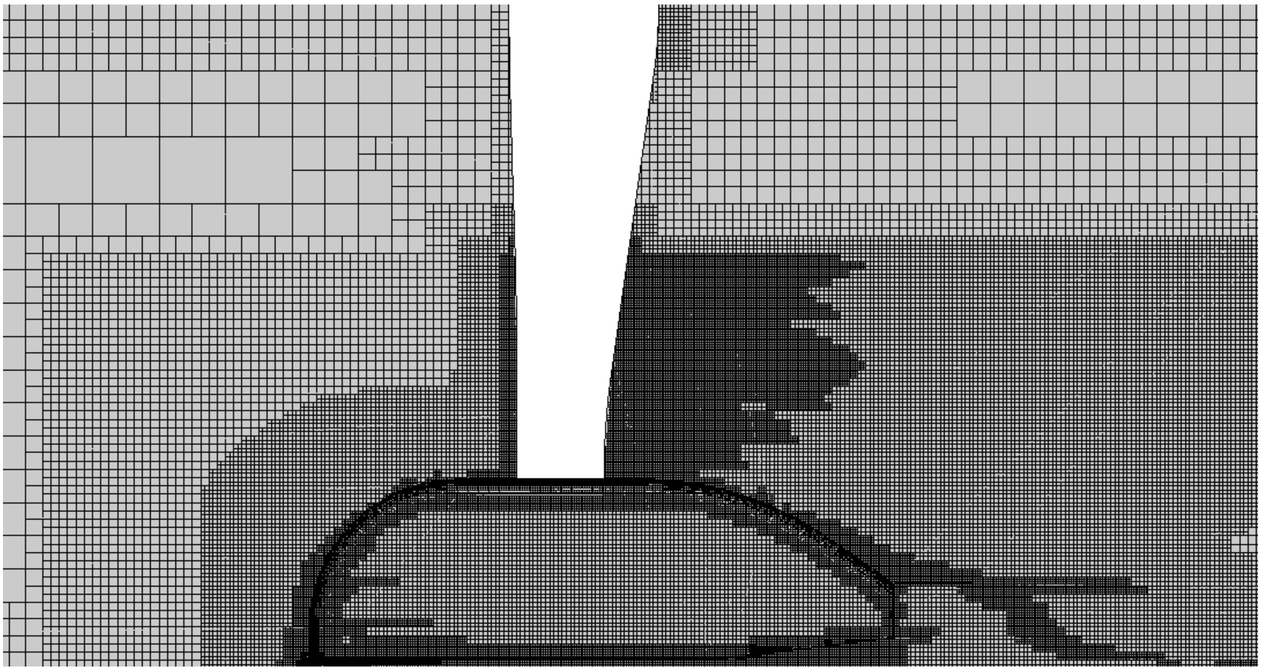

Figure 5: Computational mesh as it is used by FloEFD for the benchmark model.

Software	Computer characteristics	Cores	Calculation time [h]		Time step [s]
			Steady State	Transient	
AcuSolve	HP ProliantDL360p Gen8, Xeon E5-2660 (2.2GHz)	16	Case 1: 4.2 Case 2: 5.9	-	-
ANSYS Fluent R14.5	Dell Power Edge R720 (2.9GHz)	32	4	60	2.0×10^{-4}
FloEFD	HP Z600, Intel Xeon X5670 (2.93GHz)	6	17	-	-
iconCFD	Intel® Xeon® Processor E5645 (2.4GHz)	72	-	Case 1: 254 Case 2: 267	5.0×10^{-5}
PAM-FLOW	HP BL460c, Intel Xeon E5-2680 (2.7GHz)	16	40	155	5.352×10^{-5}
SCRYU/Tetra (DES)	Intel Xeon E5-2690 (2.9GHz)	48	-	33	1.0×10^{-4}
SCRYU/Tetra (SAS)			-	34	1.0×10^{-4}
STAR-CCM+ v7.06 (IDDES)	Dell Power Edge, Intel(R) Xeon(R)	120	-	~200	1.0×10^{-4}

STAR-CCM+ v7.06 (SST k-ω)	CPU X5675 (3.07GHz)	12	17.5	-	-

Table 3: Processing power used and calculation times for the benchmark model.

In Figure 6 the blue dashed line shows the upper and lower error margin of the case 1 which is without the additional part and the red dashed lines show the margin for case 2 (with the additional part). This graph shows the drag coefficient C_D of the simulation for both cases. The three codes AcuSolve (Inflow 2), FloEFD and STAR-CCM+ (IDDES) where within the margin for case 1 followed by SCRYU/Tetra (DES) with a larger gap and for case 2 none of the codes were exactly in the margin but the codes iconCFD, STAR-CCM+ (IDDES) and AcuSolve (Inflow 2) were the closest in this order.

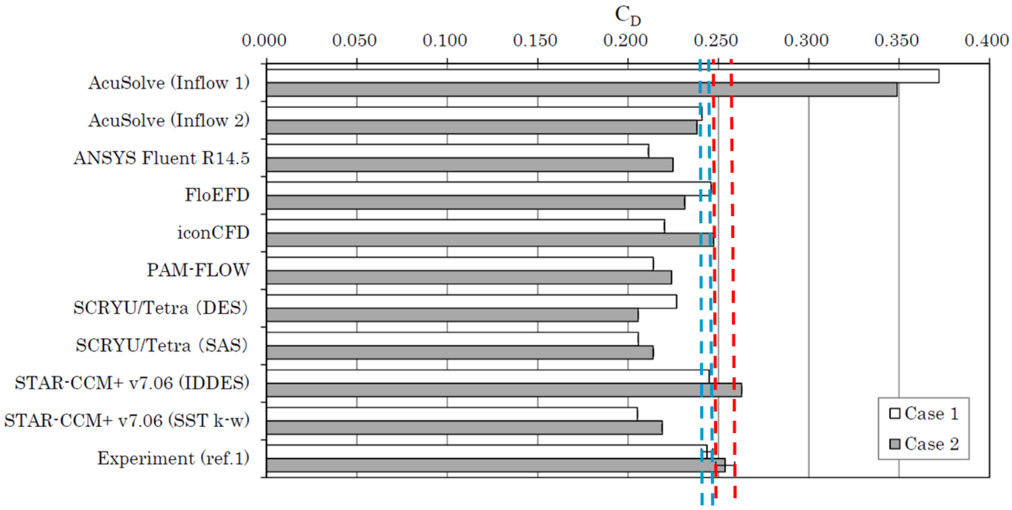

Figure 6: Drag coefficients for all CFD codes for case 1 and 2 with the experiment error margins.

In Figure 7 the same dashed lines show the margins also for case 1 and 2 but here it is for the lift coefficient C_L. For case 1 only FloEFD was exactly in the margin slightly out of margin were STAR-CCM+ (SST k-ω) and SCRYU/Tetra (SAS) at about the same level as AcuSolve (Inflow 1). For case 2 the margin was very narrow and none of the codes were exactly in it. Only STAR-CCM+ (SST k-ω) was very close, followed by ANSYS Fluent and AcuSolve (Inflow 1) but with a larger distance in that order.

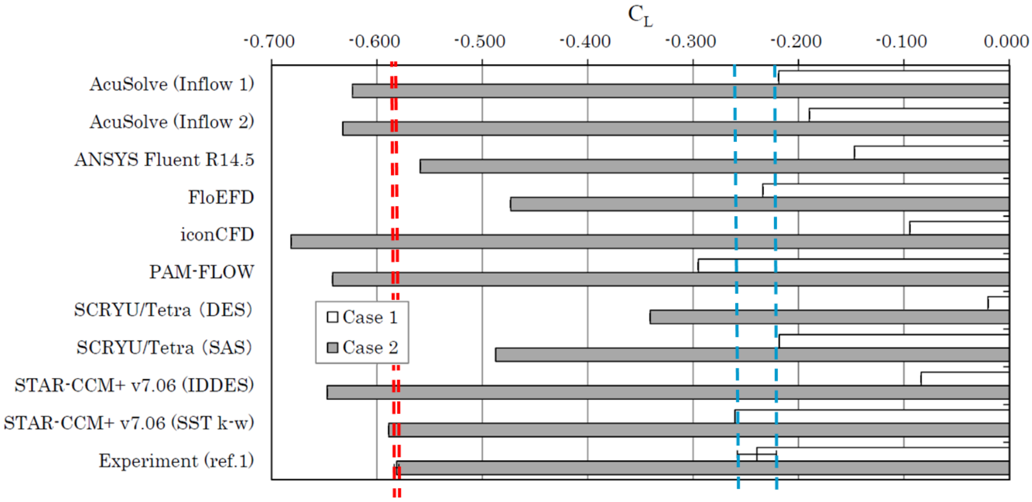

Figure 7: Lift coefficients for all CFD codes for case 1 and 2 with the experiment error margins.

The pitch-moment coefficient is shown in Figure 8 and has the same margin colours for case 1 and 2. In this graph the error margin of case 1 are very narrow and also none of the codes made it exactly in the margin but the closest are STAR-CCM+ (IDDES) followed by PAM-FLOW and then SCRYU/Tetra (SAS) and STAR-CCM+ (SST k-ω) equally distant but on the lower margin compared to PAM-FLOW. Case 2 on the other hand has a larger margin and two codes made it in the margin. FloEFD was fully in the margin followed by ANSYS Fluent on the upper margin line and then a little outside the margin is SCRYU/Tetra (SAS) and AcuSolve (Inflow 2) both about the same distance but one to the upper and the other to the lower margin respectively.

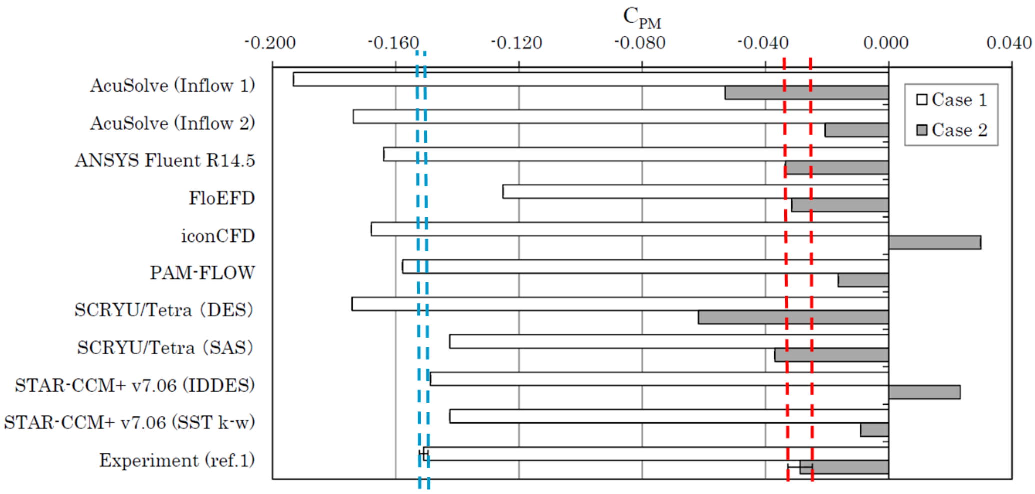

Figure 8: Pitch-moment coefficients for all CFD codes for case 1 and 2 with the experiment error margins.

Putting these in an Olympic ranking with giving ties to those who are too close together to clearly distinguish the better one, it would look like in Table 4 if just the winner for the three coefficients and the two cases are shown and in Table 5 if the participants were given medals from brace to gold and counting the number of medals overall.

Software	**Case 1 Winner**	**Case2 Winner**
Drag	STAR-CCM+ (IDDES) & AcuSolve 2	IconCFD
Lift	FloEFD	STAR-CCM+ (SST k-ω)
Pitch Moment	STAR-CCM+ (IDDES)	FloEFD

Table 4: Winner of the three coefficients and two cases.

Software	Gold	Silver	Bronze	Total
FloEFD	2	1		3
STAR-CCM+ v7.06 (IDDES)	2	1		3
STAR-CCM+ v7.06 (SST k-ω)	1	1	1	3
AcuSolve 2	1		2	3
iconCFD	1			1
Fluent		2		2
PaMFlow		1		1
SCRYU/Tetra(SAS)			3	3
AcuSolve 1			2	2
SCRYU/Tetra(DES)			1	1

Table 5: Olympic gold medal ranking of CFD codes in their accuracy compared to measurements.

Further results of the wake and body pressure measurements can be found in the JSAE paper.

4. Conclusion

Although the meshing and solver technology of FloEFD is unusual, it has proven its capability in this benchmark by more than being just accurate. It has shown it's extremely high competiveness compared to the high end traditional codes known in all industries. Ranking together at 1st place with STAR-CCM+, using fewer cells, a less sophisticated turbulence model and lower CPU time to get the same level of achievements is indeed proof of advanced CFD technology. And that considering it is built for regular engineers that do not have high numerical understanding and still would be able to achieve such a goal with an automated mesher and without a selection of highly advanced turbulence models.

Unfortunately this benchmark did not require recording the time necessary to setup the simulation and meshing the model. This should be considered with the same interest as the solver time as it actually measures the users time spent on the model and the overall process and would give a good metric for the efficiency of the code in addition to its accuracy.

5. Acknowledgements

I would like to thank the JSAE to allow the use of the benchmark information as well as showing the result outcome comparing the various codes used in the benchmark.

Also I want to thank the team from Kozo Keikaku Engineering Inc. for their excellent benchmark work they did with FloEFD.

6. References

[1] Nakashima, Takuji; Sasuga, Nobuhiro; Ito, Yuichi; Ikeda, Masami; Ueda, Iichiro; Kato, Yoshihiro; Kitayama, Masashi; Kito, Kozo; Koori, Itsuhei; Koyama, Ryutaro; Shimada, Yoshihiro; Hanaoka, Yuji; Higaki, Tatsuhiko; Fukuda, Kota; Yamamura, Jun; Li, Ye (2013). *Benchmark of Aerodynamics CFD of Simplified Road Vehicle Model*: JSAE. Paper Number: 20134343. Pages 8 - 28

[2] *Advanced Immersed Boundary Cartesian Meshing Technology in FloEFD*™: Mentor Graphics, http://go.mentor.com/2gogl

[3] *Enhanced Turbulence Modelling in FloEFD*™: Mentor Graphics, http://go.mentor.com/2glzd