OpenFOAM and Fluent Features in CFD Simulations on CRESCO High Power Computing System

F. Ambrosino[1], A. Funel[1]
[1]ENEA, Via Vecchio Macello - 80055 Portici (Naples, Italy)
E-mail: fiorenzo.ambrosino@enea.it, agostino.funel@enea.it

***Abstract*—We present the results of a CFD simulation obtained with Fluent and OpenFOAM, an open source code. Our aim is to compare OpenFOAM capabilities to that of a well known commercial CFD code like Fluent. Both the two codes have been run on CRESCO, the new ENEA high power computing cluster.**

I. Introduction

IN the last years high quality CFD techniques have become necessary and essential tools for tasks such as aircraft industry aerodynamic development, combustion research, fluid-structure interaction problem solving etc. This requires an ongoing struggle for computational efficiency improvements where parallel computing is one of the major issues. As the CRESCO supercomputer has been integrated in ENEA GRID a considerable high power computing can be achieved. Unfortunately, commercial CFD solvers like Fluent have an expensive license cost and generally this cost is directly related to the number of parallel cores needed to run the program. OpenFOAM is an open source CFD solver, it can simulate anything from complex fluid flows involving chemical reactions, turbulence and heat transfer, to solid dynamics, electromagnetics and molecular dynamics. Being an open source toolbox it does not need any proprietary license and therefore it is a valid Fluent alternative during all the phases of a CFD solution development.

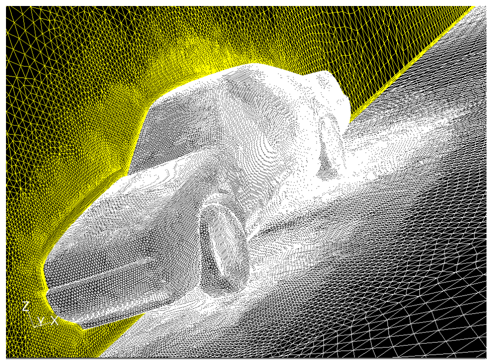

Fig. 1. Fluent mesh snapshot.

II. Problem Specification

THE case study examined here concerns the exterior flow field around a simplified passenger sedan geometry that proceeds with a steady and straight velocity in forward front direction.

The case was produced by ANSYS-Fluent and used as a benckmark of the standard benchmark collection provided by Fluent to test parallel performances on various high power computing systems [1]. It belongs to the “large” class of benchmarks: a viscous-hybrid grid with about 3.6 million prismatic cells is used to adequately model the region of interest. In Fig. 1 is shown the computational domain.

The geometry is outlined in Fig. 2. The car has a velocity v of 27.8 m/s, the pressure p is the atmospheric ambient pressure that is 101325

Pa. The flow is assumed to be steady turbulent. Moreover, due to its low velocity, it can also be considered incompressible.

The equations of motion are:

$$\nabla \cdot \mathbf{U} = 0 \tag{1}$$

$$\mathbf{U} \cdot \nabla \mathbf{U} + \frac{1}{\rho} \nabla p = \nu \nabla^2 \mathbf{U} + \mathbf{g} \tag{2}$$

where $\mathbf{U}$ is the velocity, ρ the density, p the pressure, ν the cinematic viscosity and $\mathbf{g}$ is the gravitational field.

Both for Fluent and OpenFOAM the adopted turbulence model is the standard K-ϵ.

III. OpenFOAM case setting

In order to run the case with OpenFOAM, the first step is to convert the computational domain in OpenFOAM format. This can be accomplished by using the `fluentMeshToFoam` utility [2]. In Fig. 3 is shown the result for the body car.

The case requires initial and boundary conditions settings for all the involved fields. For applying boundary conditions a boundary is broken up into a set of patches. In OpenFOAM physical boundary conditions are specified by setting appropriate keywords in a file which describes the type of the patches. Moreover, for each involved field a file exists where initial numerical conditions have to be set [2]. In this case the air enters perpendicularly the inlet (y direction) with an initial velocity of 27.8 m/s. The initial pressure field is 1 atm everywhere and at the boundaries we require the condition $\nabla p = 0$. The coefficients used for the turbulence model are $C_\mu = 0.09$, $C_1 = 1.44$, $C_2 = 1.92$, $\alpha_k = 1$, $\alpha_\epsilon = 7.69$. The value of the turbulent kinetic energy is $k = 0.02898$ m^2s^{-2} at the inlet, the same value has been set at the outlet for modeling the turbulence in case of backflow. The value of the energy dissipation rate is $\epsilon = 1.8239$ m^2s^{-3}.

The OpenFOAM solver used is `simpleFoam`, implemented for steady incompressible flow. We adopted as numerical scheme the standard finite volume discretization of Gaussian integration.

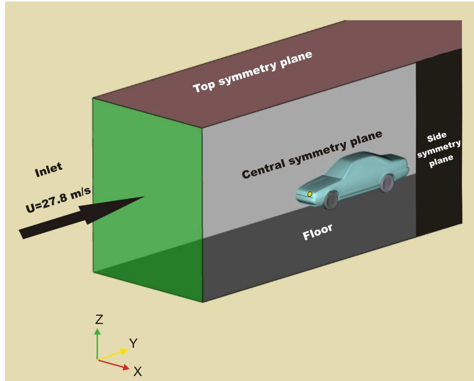

Fig. 2. The geometry.

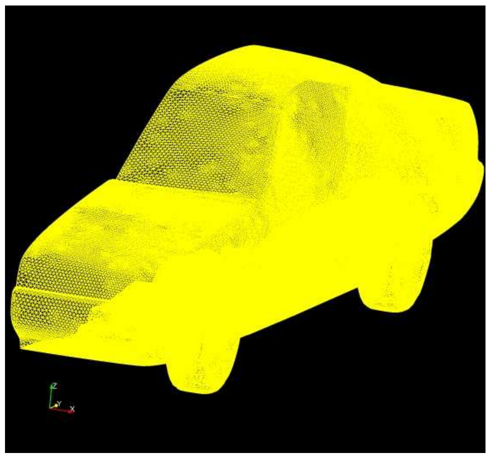

Fig. 3. Body car mesh obtained by converting the original GAMBIT file in OpenFOAM format.

IV. Cross Checking Results

Both Fluent and OpenFOAM got convergence. Fig. 4 shows the residuals of the Fluent simulation while Fig. 5 shows the OpenFoam ones.

The viscous-hybrid mesh used had not a good accuracy in modelling the behavior of the flow at the boundary layer. In fact, the y^+ scale value on the sedan surface is of order 100, much more of the viscous sublayer scale value. Fig. 6 shows a map of y^+ on the sedan surface. However, we do not want to study boundary layer effects. Our attention focused on comparing computational performances, thus we considered only global

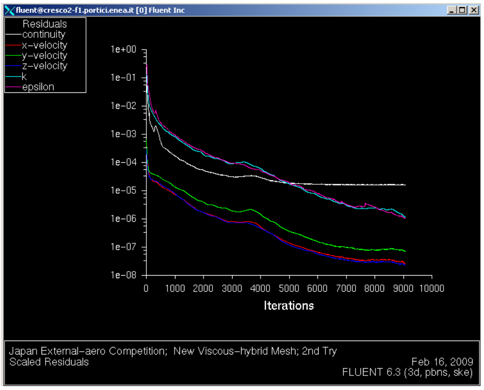

Fig. 4. Fluent residuals.

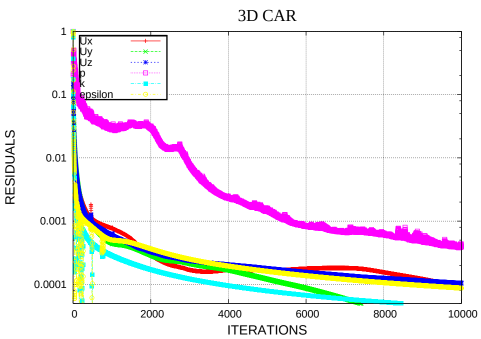

Fig. 5. OpenFOAM residuals.

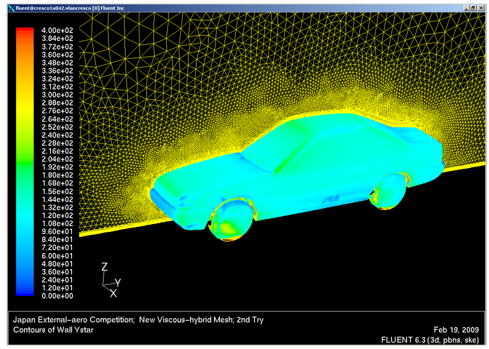

Fig. 6. Map of y^+.

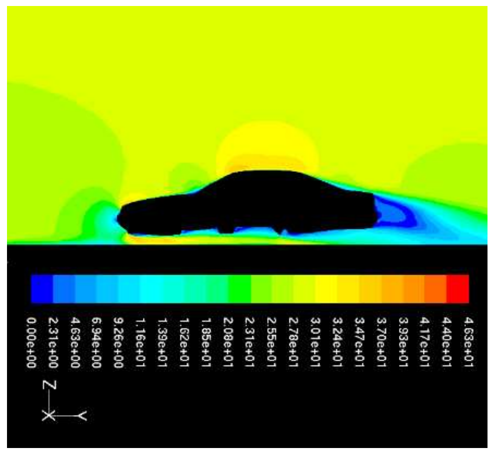

Fig. 7. Contour map of Fluent velocity field.

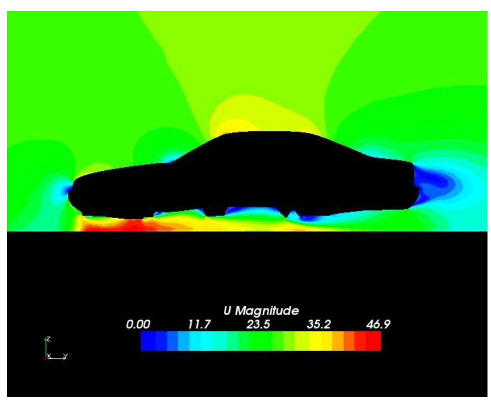

Fig. 8. Contour map of OpenFOAM velocity field.

properties of the flow motion.

In order to compare Fluent with OpenFOAM results we shall analyse the shape of the fields on the central symmetry plane. Fig. 7 and Fig. 8 show the contour map of the velocity magnitude field around the sedan for Fluent and OpenFOAM solvers, respectively.

We observe that the magnitude of the velocity fields is of the same order. The velocity field around the car has a similar shape in both simulations.

Fig. 9 anf Fig. 10 show the contour map of the pressure field for Fluent and OpenFOAM solvers, respectively. The Fluent picture reports the pressure values referred to 1 atm, thus negative values indicate a pressure lower than 1 atm. On the other hand, the OpenFOAM picture reports the absolute pressure values as displayed in the scale. In this case an overall agreement is observed, too.

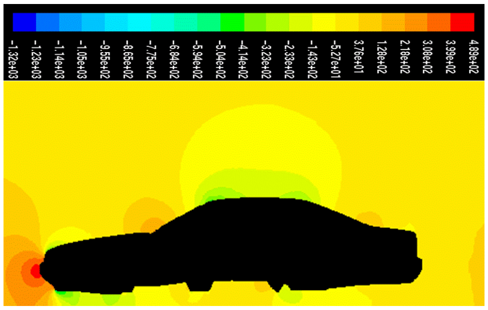

Fig. 9. Contour map of Fluent pressure field.

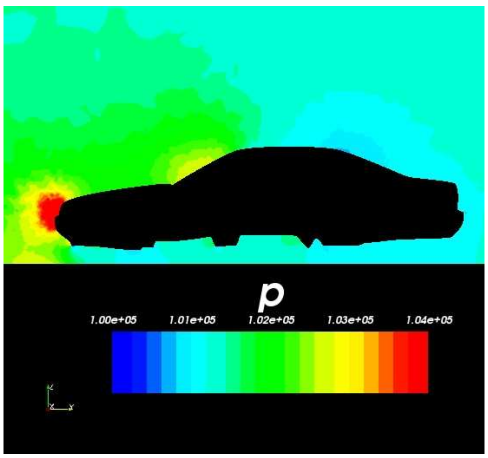

Fig. 10. Contour map of OpenFOAM pressure field.

All simulations have been run on the CRESCO cluster architecture at the ENEA-Portici Research Center. Performances and scalability tests were made both for Fluent and OpenFOAM solvers. Parallel efficiency decreases as the number of computing cores increases. Moreover, parallel efficiency enhances with the mesh size. Fig. 11 shows the speed-up of the simulations up to 512 cores. The speed-up is defined as a ratio of serial to parallel execution time.

The shape of the speed-up depends on the method used for decomponing the computational domain. In the case of Fluent we used a standard decomposition, as comes with the benchmark, where the mesh is subdivided according to Cartesian directions, while in the OpenFOAM case we used an optimized decomposition method which attempts to minimize the number of common faces between processors. As Fig. 11 shows, the speed-up increases up to 256 cores with the OpenFOAM solver and up about 64 cores with Fluent.

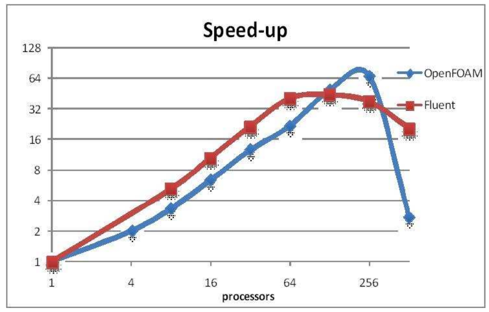

Fig. 11. Speed-up for Fluent and OpenFOAM solvers. The simulations have been run on the ENEA CRESCO high power computing system.

V. Conclusion

In summary we have shown that OpenFOAM may be suitable for running CFD simulations in place of a commercial code like Fluent. The main advantage is that of not having to pay any license cost. This aspect is particularly important when one wishes to run a CFD simulation with many cores. Moreover, being an open source toolbox OpenFOAM is fully extensible and provides the possibility of writing new libraries. We have shown similar results are obtained by running OpenFOAM with a Fluent benchmark. Tests have been made on the ENEA CRESCO high power computing system which allowed us to establish that OpenFOAM code has a good scalability.

Acknowledgments

We thank Silvio Migliori, Salvatore Podda, Pietro D'Angelo and Paolo Roccetti for helping us with their support. We also wish to thank all the staff of the ENEA-Portici Research Center.

References

[1] http://www.fluent.com/software/fluent/fl5bench/flbench_6.3/problems/fl5l2.htm
[2] http://www.opencfd.co.uk/openfoam/doc/user.html