

Study on the Capability of an Open Source CFD Software for Unsteady Vehicle Aerodynamics

Johannes Wojciak, Bastian Schnepf, Thomas Indinger and Nikolaus Adams
Technische Universitaet Muenchen

ABSTRACT

A wind-tunnel experiment investigating unsteady flow phenomena around a generic notchback during single crosswind gusts is modeled with the open source CFD package OpenFOAM®. The overall objective is to assess the capability and accuracy achieved by the simulation tool with respect to its potential for industrial usage. Transient yaw simulations apply a sliding interface between two computational grids, which are generated using the commercial software Spider®. It is shown that a stable simulation process is feasible but requires long computation times. The physical accuracy of the investigated phenomena depends on the computational grid and on the turbulence model used. Although the obtained aerodynamic loads qualitatively correspond with the experimental results, the absolute values are not satisfactory when working with a coarse grid with 6.2 million cells. Then, characteristic surface pressure distributions and their transient development differ from the experimental data. Consequently, the phase shifts of the computed load transients against the model motion are incorrect. In contrast, steady calculations for constant yaw on a refined mesh with 10.9 million cells using alternative turbulence models deliver better results. A realizable k- ϵ model proves to predict the wake flow most accurate, since solely this model captures the dominating leeward inflow at the rear lights observed in the experiment. As a consequence, the estimation of aerodynamic loads being relevant for the driving stability of vehicles in gusty crosswind is largely improved by using this setup. Considering the additional computational resources needed for performing transient yaw simulations with the improved setup, this was not accomplished in this study.

CITATION: Wojciak, J., Schnepf, B., Indinger, T. and Adams, N., "Study on the Capability of an Open Source CFD Software for Unsteady Vehicle Aerodynamics," *SAE Int. J. Commer. Veh.* 5(1):2012, doi:10.4271/2012-01-0585.

INTRODUCTION

Due to the increased willingness of industry to become independent of licensed computational fluid dynamics (CFD) software and to gain insight into the numerical models applied, open source tools have experienced more interest in the past years. Audi, for instance, presented a new aerodynamic development process with an in-house code based on numerical simulations using OpenFOAM® [1]. Some manufacturers followed successfully, such as presented recently by Ford [2]. Regardless of the CFD software, the common requirements of its users are solidity and reliability when predicting aerodynamic loads in industrial processes.

However, most of the OpenFOAM® simulation setups presented in the field of vehicle aerodynamics still lack of complex modeling techniques such as for example the treatment of moving wheels. This is particularly detrimental, since recent investigations on unsteady aerodynamic phenomena underlined the demand of being capable to simulate more complex flow scenarios. One of these complex flow scenarios appears when a crosswind gust hits a vehicle.

AIM OF STUDY

The aim of this study is to assess the capability of the open source CFD software OpenFOAM® to model a complex flow scenario during transient model motion, to quantify the accuracy of the results and finally to provide information on the computational performance and cost of the simulation. Thus, the objective is extended beyond the correct prediction of drag and lift so as to capture the influence of an unsteady change of wake structures on unsteady surface pressure distributions correctly.

EXPERIMENTAL DATA FOR VALIDATION

The experiment presented by Theissen and Wojciak et al. [3,4] is chosen as a benchmark case due to the detailed and accessible experimental results. There, a 50% scale vehicle model was oscillated with a frequency of 2Hz and an amplitude of $\pm 3.3^\circ$ around its vertical axes while being exposed to constant flow at 58.3m/s. This corresponds to a

Reynolds number of $8.9 \cdot 10^6$ and a Strouhal number of 0.08. The experiments were conducted in BMW's Aerolab wind tunnel featuring a nozzle cross section of 14m^2 , a single moving belt as well as a boundary layer scoop and an air suction and tangential blowing unit. Note that the model was lifted 30mm off the moving ground. The wheels were fixed to the model and did not rotate. The experimental setup is outlined in Figure 1. For detailed information on the setup refer to [3].

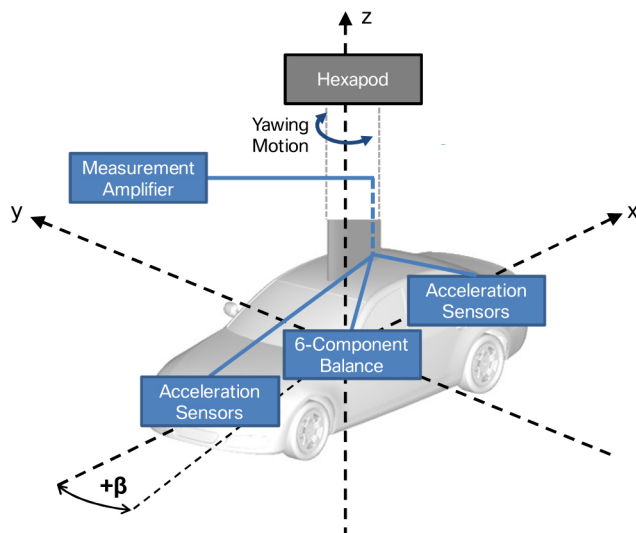


Figure 1. Schematic view of the experimental setup presented by Theissen et al. [16].

The vehicle model is called DrivAer and was first introduced by Heft et al. [5]. It is a generic mid-range passenger car model, but exhibits realistic and relevant geometry details. For the experimental study [3,4] a preliminary version of the DrivAer with a slightly more rounded rear was used, see Figure 2. The 50% scale model has a total length $L_{\text{ref}}=2.305\text{m}$, a wheelbase length $L_{\text{WB}}=1.393\text{m}$ and a frontal cross section $A_{\text{ref}}=0.542\text{m}^2$. In agreement with the experiment, the vehicle length is used for the calculation of the Reynolds and Strouhal number, whereas the cross section and the wheelbase are used for the non-dimensionalization of pressures and aerodynamic loads.

During steady flow conditions, experimental data were averaged over more than 100s. During model rotation, data were periodically averaged over more than 400 periods resulting in a maximum error of 2N and 1Nm for the three force and three moments, respectively. Load evolutions were eventually 11Hz low pass filtered.

Using this setup, unsteady aerodynamic loads were observed to differ significantly from quasi-steady loads. In particular, the unsteady yaw moment exceeded the maximum value encountered during steady flow conditions. It was shown that the wake flow of a vehicle being subjected to constant crosswind is characterized by a region of high lateral velocity at the leeward side and two counter-rotating vortices

establishing above and below that region. This structure reacts with a time delay to a change of the oncoming flow. The delayed response of the wake flow causes phase shifted pressure fluctuations at the rear side of the vehicle and thus causes an unsteady increase of the yaw moment and a decrease of the side force and roll moment.

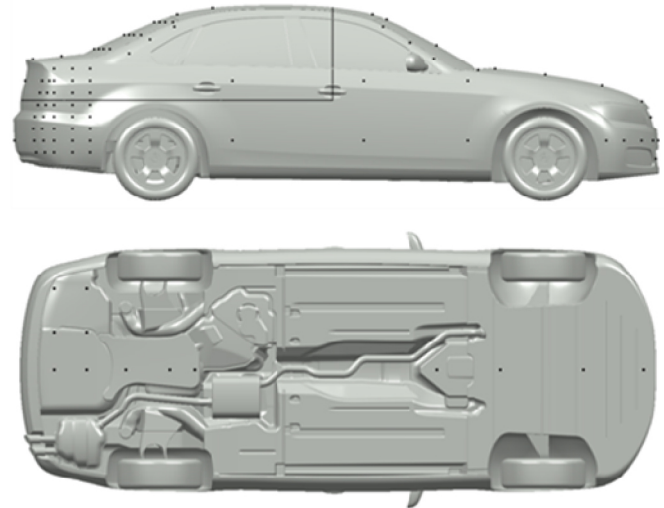


Figure 2. Notchback model with experimental probe locations presented by Theissen and Wojciak et al.[3,4].

NUMERICAL METHOD

In this study, steady and unsteady Reynolds-averaged Navier -Stokes simulations are performed using the open source software package OpenFOAM® Version 1.5 dev.

SOLVERS

OpenFOAM®'s built-in transient solver for simulations involving moving grids is TurbDyMFoam. The pressure-velocity coupling is based on the PISO algorithm by Issa [21]. After updating the moving mesh, the momentum equations are solved with the pressure from the old timestep. Then, in the PISO-loop, the Laplacian pressure correction equation is solved and the velocity is corrected in order to satisfy the continuity equation. Finally, the turbulence model can be applied. Since the pressure velocity coupling is only valid for small Courant numbers, the chosen timestep has to be sufficiently small.

For steady-state simulations without mesh motion the solver SimpleFoam is used. Following the SIMPLE algorithm by Caretto et al. [22] the solution procedure is very similar to the PISO algorithm described above. The momentum correction, however, is only performed once and the solution is being under-relaxed. The latter is necessary to accelerate convergence. In this study, the pressure is under-relaxed by a factor of 0.1, whereas the other solution variables are under-relaxed by a factor of 0.3. Thereby, an under-relaxation factor close to zero corresponds to a dominance of the old solution.

TURBULENCE MODELLING

The closure problem of the RANS equations is dealt with by using two-equation eddy-viscosity models which are already implemented in OpenFOAM® 1.5-dev. The two-equation models presented in this study are the $k-\omega$ -SST model by Menter [15], a linear realizable $k-\epsilon$ model by Shih et al. [16] and a quadratic nonlinear $k-\epsilon$ model by Zhu et al. [17].

Combining the near-wall behavior of a standard $k-\omega$ model and the far field behavior of a standard $k-\epsilon$ model through blending, the $k-\omega$ -SST model incorporates advantages of both models. However, as an isotropic model it is known to have deficiencies in predicting flows with strong streamline curvature. The linear realizable $k-\epsilon$ model is supposed to perform better when dealing with shear layers. Being an anisotropic model, the quadratic nonlinear $k-\epsilon$ model is also used in this study to benefit from its improved capability of predicting rotating flows.

Wall Functions

Since high-Re flows would require a very fine resolution of the boundary layer to achieve accurate results, wall functions are used in order to limit the computational effort. A logarithmic velocity profile is applied for dimensionless wall-distances $y^+ \leq 11$,

$$U_{wall} = \frac{1}{\kappa} \ln(Ey^+) + B, \quad (1)$$

$$y^+ = \frac{c_\mu^{1/4} \sqrt{k}}{\nu} y, \quad (2)$$

where $\kappa=0.4187$, $E=9.0$, $B=5.25$ and C_μ is specified by the turbulence model.

LINEAR SOLVERS

The equations for U , k , ϵ and ω are solved using a Gauss-Seidel smoothed solver. During unsteady simulations, the PCG (Preconditioned Conjugate Gradient) algorithm is applied for the pressure equation. A diagonal based incomplete Cholesky factorization is chosen for preconditioning. For steady runs the GAMG (Geometric-Algebraic Multi-Grid) solver replaces the PCG solver due to slight advantages in calculation performance. It first solves the equation on a very coarse grid and then step-by-step increases the grid's resolution. By mapping the coarse solution on the fine grid it can achieve good rates of convergence. For more details on linear solvers see Ferziger et al. [23].

DISCRETIZATION SCHEMES

Convective terms are discretized by a first order upwind scheme in order to ensure numerical stability. A blended scheme, called Gamma scheme in OpenFOAM®, was employed as well, but did not deliver more accurate results in

steady calculations. Therefore, calculations using the Gamma scheme are not presented in this paper. Laplacian terms are discretized by second order central differencing. Furthermore, surface-normal gradients are computed using an explicit non-orthogonal correction. In the case of unsteady simulations, the temporal derivatives are approximated using a first order implicit Euler scheme.

SLIDING MESH INTERFACE

A sliding mesh interface, called Generalized Grid Interface (GGI) in OpenFOAM®, is used to couple the static and the moving part of the mesh. It establishes the transfer of field variables between the cylindrical boundary faces of both regions through weighted interpolation. The package OpenFOAM® 1.5-dev contains a prebuilt interface for handling constant speed rotations. It has been modified to support the oscillatory motion of the model according to the experiment by implementing an incremental change of yaw per timestep

$$\Delta\beta(t) = -2\pi f\beta_0 \sin(2\pi f(t - t_{start}))\Delta t. \quad (3)$$

Thereby, the computational effort of updating the locations of all grid points in the moving region depends on the complexity of the motion equation and the number of moving points. Additionally, the interface covers all processors in a parallel computation which limits the potential of scalability. A detailed description of the GGI is given by Beaudoin et al. [14]

NUMERICAL SETUP

MODEL

The geometrical setup is, wherever possible, identical to the experimental setup. A surface mesh of the 50% scale vehicle model is generated based on CAD data. Differences between the CAD geometry and the final assembled geometry are regarded as negligible and are not accounted for. As mentioned in [3,4], the strut geometry used in the experiments is slightly asymmetrical. It has been scanned during the experiments and a surface mesh is generated based on the scan data instead of the CAD data. However, the scan was not performed right after the test runs and thus differs to a certain degree from the real test status. In the simulations presented in this study, only the components being located in the free jet are considered, i.e. only the vehicle and the strut are modeled. The wind tunnel itself is not reproduced. Since the nozzle cross-section of BMW's Aerolab wind tunnel is very large and since experimental blockage corrections are not necessary in the Aerolab wind tunnel [8], it is correct to omit its replication in the simulation.

NUMERICAL DOMAIN AND BOUNDARY CONDITIONS

The simulation volume is a box with 3 vehicle lengths in front and 5 vehicle lengths behind the model. The height of the simulation domain is 6m and the width is 20m, see [Figure 3](#). Hence, the blockage ratio calculates to 1%, whereas it is 0.8% in the experiment.

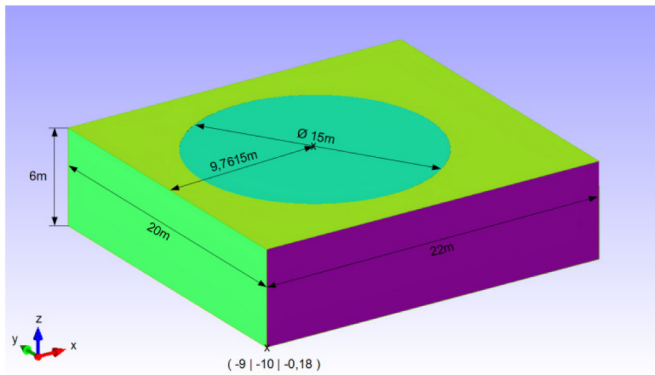


Figure 3. Numerical domain and sliding mesh interface.

At the inlet the velocity is set to $U=58.3\text{m/s}$, the kinetic turbulent energy to $k=0.082\text{m}^2/\text{s}^2$ and the specific dissipation to $\omega=0.71\text{s}^{-1}$ and $\epsilon=0.0387\text{m}^2/\text{s}^3$, respectively. This is in line with the inlet conditions during the experiment. The same holds true for the static pressure prescribed at the outlet, which is exactly as measured in the plenum chamber during the experiments. The top of the simulation volume as well as the side walls are defined as frictionless walls. The floor is defined as a moving wall with the same velocity as the inlet velocity of the flow. In line with the experiment, the wheels do not rotate.

MODEL MOTION

The sinusoidal oscillation of the vehicle in the wind tunnel is modeled using a sliding mesh interface between a cylindrical rotating and rectangular non-rotating domain, compare [Figure 3](#). The mesh interface is set up on a cylinder with 15m radius around the rotating axis of the model and reaches from the moving belt up to the top of the strut. Amplitude and frequency of the rotation are identical to the experiment. [Figure 4](#) illustrates the evolution of the model position and its angular velocity during the simulation. Accordingly, two oscillations are modeled, but results are only taken from the last oscillation. By doing this, the unsteady flow field has already adapted to the model oscillation, whereas it is influenced by its former constant yaw position during the first oscillation period. This procedure is very similar to the experimental method, where many oscillation periods were phase averaged.

The flow field at constant yaw is evaluated before and after the two model oscillations when utilizing an unsteady solver, as marked in [Figure 4](#). When utilizing a steady solver, additional simulations are performed at constant yaw.

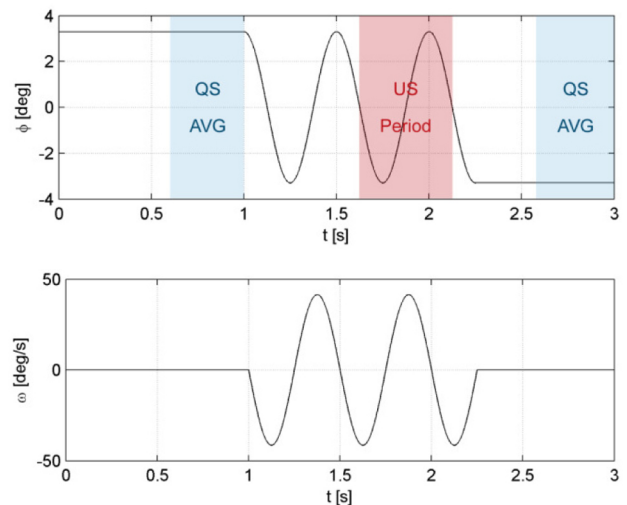


Figure 4. Model motion (yaw angle and angular velocity) during numerical simulation (QS=quasi-steady, US=unsteady, AVG=average period for constant yaw).

NUMERICAL GRIDS

The generation of surface and volume meshes is performed using the third party software Spider® [9, 10, 11]. In this study results of two different meshes are presented, named “Mesh 1” and “Mesh 2” in the following.

Mesh 1

Mesh 1 is a coarse mesh. The total number of cells of Mesh 1 amounts to 6.2 million. Two prism layers with a size of 6mm surround the whole vehicle leading to a dimensionless distance between the first cell center and the wall of 100. Another refinement region around the car and a rectangular box in the wake flow region are resolved with 12mm polyhedral cells, compare [Figure 6](#). Then, the cell size is gradually increased towards the edges of the simulation volume. The cylindrical mesh interface consists of 4680 patches.