Steady-state aerodynamics of a Porsche Taycan: A numerical investigation at

Name: LEI, Zhen

Student ID: 20937857

Course: Computational Fluid Dynamics

Instructor: Prof. Larry Li

Department of Mechanical and Aerospace Engineering,

The Hong Kong University of Science and Technology

05/2023

Content

[1. Introduction and engineering problem 3](#_Toc134974917)

[1.1 Background information 3](#_Toc134974918)

[1.2 Case application 3](#_Toc134974919)

[2. Numerical method 3](#_Toc134974920)

[2.1 Case assumptions 3](#_Toc134974921)

[2.2 Governing equations 3](#_Toc134974922)

[2.3 Numerical approach 4](#_Toc134974923)

[3. Numerical setup 4](#_Toc134974924)

[3.1 Computational domain 4](#_Toc134974925)

[3.2 Boundary/Initial conditions 5](#_Toc134974926)

[4. Grid convergence test 6](#_Toc134974927)

[4.1 Test matrix 6](#_Toc134974928)

[4.2 Results and discussion. 7](#_Toc134974929)

[5. Conclusions 7](#_Toc134974930)

[Reference 7](#_Toc134974931)

## 1. Introduction and engineering problem

In this case, there is steady-state aerodynamics of a *Porsche Taycan*: A numerical investigation at . The description of the engineering problem is as below:

1. The car *Porsche Taycan* moving at a speed of 20m/s.
2. Air flows around the body of the car.
3. Assume sea-level air properties at 15°C.

The schematic of the engineering problem could be seen as Figure 1:



Figure : schematic of the engineering problem

### 1.1 Background information

### 1.2 Case application

## 2. Numerical method

### 2.1 Case assumptions

The flow of the CFD case satisfies the conservation of mass and conservation of momentum.

Also, there are some basic assumptions which is used to simplify the engineer problem.

1. The flow is incompressible, and it is constant.
2. The flow is steady, which means .

### 2.2 Governing equations

As the case is a 3D case, should be taken into consideration. As a result, the governing equations for the case are:

Mass:

Momentum:

### 2.3 Numerical approach

Just as mentioned, this case is the numerical simulation of the steady flow around a 3D car, so steady solvers are in need. Based on the governing equations, simpleFoam is selected for simulating this case in unsteady flow.

SimpleFoam is a steady-state solver for incompressible, turbulent flows using the SIMPLE algorithm. SIMPLE algorithms are based on evaluating some initial solutions and then correcting them. SIMPLE only makes 1 correction. The grid convergence test will be conducted for different mesh to get the result.

## 3. Numerical setup

The flow properties used in this case are as Table 1.

Table : flow properties

|  |  |
| --- | --- |
| Parameters | Value |
| Reynolds Number () |  |
| Diameter () |  |
| Air density () |  |
| Kinematic Viscosity (=) |  |
| Velocity () |  |

### 3.1 Computational domain

The computational domain could be as Figure 2.



Figure : 3D case for the computational domain



Figure : breakdown of the computational domain

### 3.2 Boundary/Initial conditions

Here are summaries of the boundary conditions and initial conditions:

|  |  |  |  |
| --- | --- | --- | --- |
| Turbulence Model | | *k- SST* | |
|  | |  | |
| BC | Boundary | Type | Value |
| inlet | U,k,omega:fixedValue  p:zeroGradient  nut:calculated  boundary:patch | U= (0 20 0)  nut=0  k=0.06  omega=0.089 |
| outlet | U,k,omega: inletOutlet  p: fixedValue  nut:calculated  boundary:patch | U: inletValue 0,  value (0 20 0)  p=0  nut=0  k: inletValue 0.06,  value 0.06  omega: inletValue 0.089,  value 0.089 |
| BC(continued) | LowerWall | U: fixedValue  p:zeroGradient  nut: nutkWallFunction  k: kqRWallFunction  omega: omegaWallFunction  boundary:wall | U= (0 20 0)  p=0  nut=0  k=0.06  omega=0.089 |
| walls | U: Noslip  p: zeroGradient  nut: nutkWallFunction  k: kqRWallFunction  omega: omegaWallFunction | nut=0  k=0.06  omega=0.089 |
| upperWall | U,P,k,omega:slip  nut:calculated  boundary:patch | Nut=0 |
| frontAndBack | U,P,k,omega: slip  nut:calculated  boundary:patch | Nut=0 |
| IC | sides | - | |
| inlet | U = (0,20,0) | |
| topBot | - | |
| outlet | p = 0 | |
| wall | - | |

## 4. Grid convergence test

### 4.1 Test matrix

In this part, we are going to conduct the grid convergence test to get the optimal grid. Using the IC and BC, set different refinement levels as Table 2.

Table : test matrix

|  |  |  |
| --- | --- | --- |
| mesh | cells |  |
| coarse |  |  |
| medium 1 |  |  |
| medium 2 |  |  |
| fine |  |  |

Figure : grid test

### 4.2 Results and discussion.

Get from the equation

## 5. Conclusions

## Reference