25th ABCM International Congress of Mechanical Engineering
October 20-25, 2019, Uberlândia, MG, Brazil

COB-2019-1458
VALIDATION OF 2D AND 3D SUPERSONIC FLOWS SIMULATIONS IN OPENFOAM

Pedro Henrique Rosa dos Santos
Braulio Gutierez Pimenta
Universidade de Brasilia, UnB, Campus Darcy Ribeiro, Faculdade de Tecnologia, Departamento de Engenharia Mecanica
pedrorosa.eng@gmail.com, brauliogp@unb.br

Abstract. *The computational Fluid Dynamics helps the design of supersonic jets and the open-source software can show a good relationship between cost and benefit. In this context, the present work aimed to validate the results of OpenFOAM concerning the simulation of supersonic external flows.*
The methodology consisted of the simulation of 2D and 3D supersonic flows, with or without adaptive mesh refinement (AMR) based on the velocity gradient. OpenFOAM 1712 has been compiled in a Cluster. A MATLAB code was developed to calculate the parameters of the flow over the double wedge airfoil and its results as compared to the simulations with and without AMR. Moreover, experimental results obtained in a wind tunnel of wings with double-wedged sections were used as a reference to the 3D case.
The 2D case presented 0.94% of accuracy without AMR, and 0.10% with AMR. In both situations, the shock angle was in agreement with the theory. The 3D results followed the expected tendency and presented satisfactory convergence, although the critical performance penalties due to the AMR mechanism. It was noticed that good practice is no to use the AMR unless the convergence criteria are reached in a less dense mesh. .

Keywords: *OpenFOAM, Validation, Supersonic, Double-Wedge, Adaptive Mesh Refinement*

1. INTRODUCTION

To minimize the generation and the effects of shock waves is a necessary factor, among others, to the flight of supersonic airplanes over countries without unacceptable levels of noise and fuel consumption. Computational Fluid Dynamics (CFD) software should be prepared to deal with complex problems and to work in High-Performance Computing (HPC) architectures, especially in parallel computation.

The performance of open-source CFD software might not be better than private ones. Although, the relation between cost and benefit could be determinant in the decision of which to use. Moreover, OpenFOAM has the advantage that it was built to be flexible, due to the use of the objected-oriented C++ programming language, and it has a strong community of developers. However, it has the disadvantage that it does not have the best performance in every area, due to its generalist approach, and its learning curve can be stepper than proprietary software's learning curves.

In this context, the present work is dedicated to describing the precision of the results of OpenFOAM concerning the simulation of supersonic externals flows. The benchmark cases used were firstly the double wedge airfoil under supersonic steady flow (Fig. 1) because it has an analytical solution based on oblique shock and expansion wave theory.

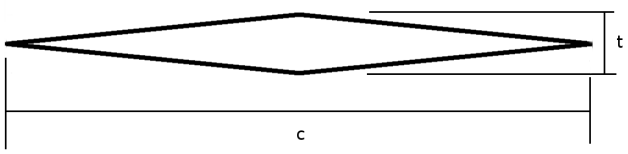

Figure 1. Double wedge airfoil. *c* is the chord length and *t* is the airfoil thickness.

According to Anderson (2003), the oblique shock can be predicted by the Eq. (1), where the Ma_1, Ma_2, θ and β are presented in Fig. 2. Also according to Anderson (2003), the expansion wave can be predicted by the Eq. (2) and Eq. (3), where the parameters Ma_1, Ma_2 and θ are presented in Fig. 3. In Eq. (1) and Eq. (3), γ is the ratio between the specific

heat at constant pressure and specific heat at constant volume.

$$\tan(\theta) = 2\cot(\beta)\left[\frac{\mathrm{Ma}_1 - 1}{\mathrm{Ma}_1(\gamma + \cos(2\beta)) + 2}\right] \tag{1}$$

$$\theta = \nu(\mathrm{Ma}_2) - \nu(\mathrm{Ma}_1) \tag{2}$$

$$\nu(\mathrm{Ma}) = \left(\frac{\gamma+1}{\gamma-1}\right)^{\frac{1}{2}} \tan^{-1}\left(\left[\frac{\gamma-1}{\gamma+1}(\mathrm{Ma}^2-1)\right]^{\frac{1}{2}}\right) - \tan^{-1}\left(\left[\mathrm{Ma}^2-1\right]^{\frac{1}{2}}\right) \tag{3}$$

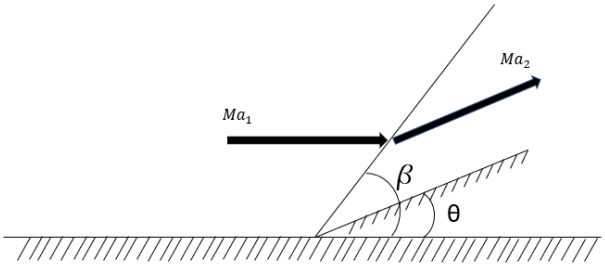

Figure 2. Oblique shock over a wedge.

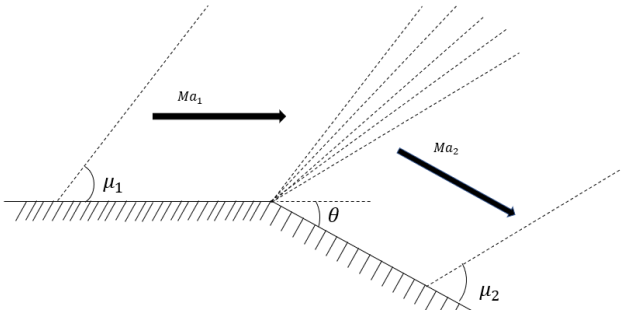

Figure 3. Expansion wave over a wedge.μ_1 and μ_2 are the mach wave angles.

The second benchmark case was based on the experimental work of Love (1949), which tested wings with double wedge sections on a supersonic wind tunnel. Figure 4 shows their geometry.

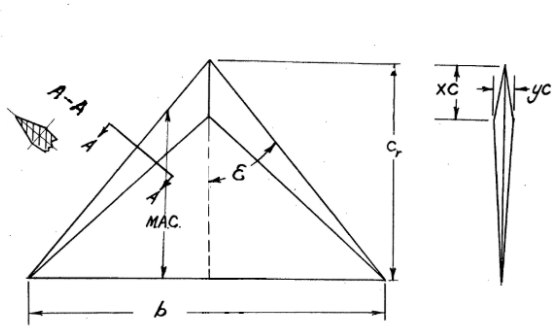

Figure 4. Projected views of the wing utilized in the simulations. *b* is the wingspan, *Cr* is the root chord, *XC* is the position of the maximum thickness in the x direction, ϵ is the apex angle, *M.A.C.* is the mean aerodynamic chord, *YC* is the position of the maximum thickness in the y direction. Adapted from Love (1949).

In the first case, the simulation results of pressure over the airfoil can be compared with the analytical solution obtained with the aid of a MATLAB routine. In the second case, the available pressure data from the wing tunnel experiment can be compared with the simulation results, but it is limited to some sections of the wing.

2. SIMULATION METODOLOGY

OpenFOAM is an open-source CFD software based on the finite volumes method, it is distributed under the GNU license and works in a wide range of problems. In the present methodology, the chosen algorithm solves the governing equations sequentially to a dominant variable using the iteration procedure, which keeps running until the convergence criteria. Moreover, it uses implicit methods to the time stepping.

The cluster utilized has an internal network composed of InfiniBand cables, whose nominal transference is 56 Gb/s. It has 8 computing nodes, and each node has 16 GB MCDRAM, 96GB RAM DDR4, and1 Intel Xeon Phi *Knights Landing* with 64 cores. This processor has 4 Vectoral Processing Units (VPU) capable of 16 floating-point operations per cycle. Moreover, each it can allocate up to 4 instances of a process, using Hyper-Threading technology. However, OpenFOAM has not come with vectorization capabilities, so it has not used the VPUs.

2.1 Fisical model and boundary conditions

The governing equations are based in the principle of conservation of mass, Eq. (4), of moment, Eq. (5), and energy, Eq. (6). ρ is the specific mass, u is velocity, P is pressure, τ is shear stress, E is total internal energy, q is heat flux and f is field forces in a fluid particle.

$$\frac{\partial \rho}{\partial t} + \nabla \bullet (\rho \mathbf{u}) = 0 \tag{4}$$

$$\frac{\partial (\rho \mathbf{u})}{\partial t} + \nabla \bullet (\rho \mathbf{u}\mathbf{u}) = -\nabla P + \nabla \bullet \underline{\tau} + \rho \mathbf{f} \tag{5}$$

$$\frac{\partial \rho E}{\partial t} + \nabla \bullet ([\rho E]\mathbf{u}) = -\nabla \bullet \mathbf{q} - \nabla \bullet (P\mathbf{u}) + \nabla \bullet (\underline{\tau} \bullet \mathbf{u}) \tag{6}$$

Equation (7) gives the shear stress, and viscosity is calculated by the Sutherland law. I is the identity matrix and μ is the dynamic viscosity of air.

$$\underline{\tau} = \mu \left(\nabla \mathbf{u} + (\nabla \mathbf{u})^T - \frac{2}{3} (\nabla \bullet \mathbf{u}) \underline{I} \right) \tag{7}$$

The constitutive relations are the ideal gas, Eq. (8), calorically and thermally perfect, Eq. (9) and Eq. (10). R is the perfect gas constant, T is temperature, e is internal energy, h is entalphy, Cv is specific heat at constant volume and Cp is specific heat at constant pressure.

$$P = \rho R T \tag{8}$$

$$e = C_v T \tag{9}$$

$$h = C_p T \tag{10}$$

Total slip condition was used on the airfoil surface, Eq. (11). n is the normal vector to the airfoil surface. Moreover, turbulence does not play a major influence in the case, so there is no turbulence model.

$$\mathbf{u} \bullet \mathbf{n} = 0 \tag{11}$$

2.2 Solution Algorithm

The solution algorithm of the simultaneous equation system utilized was the *sonicFoam*, whose description was obtained in Asproulias (2014) and is presented next.

The initial conditions of the field are initialized ($\rho^{(0)}, \mathbf{u}^{(0)}, P^{(0)}$ and $e^{(0)}$), including the mass flux $\mathcal{M}_f^{(0)}$, Eq. (12), where $\mathbf{S}_f$ is the surface normal vector of each finite volume. The subscript f indicates that the value is obtained at the face of the finite volume.

$$\mathcal{M}_f^{(0)} = \mathbf{S}_f \bullet (\rho^{(0)} \mathbf{u}^{(0)})_f \tag{12}$$

After the initialization of these variables, the time loop begins. Firstly, the specific mass $\rho_p^{(init)}$ is obtained with the Eq. (4) in the form of Eq. (13). The subscript *p* indicates that the value is calculated at the center of each volume. *t* indicates the time and *V* indicates the volume of each element.

$$V_p \frac{\rho_p^{(init)} - \rho_p^{(0)}}{\Delta t} + \sum_f \mathcal{M}_f^{(0)} = 0 \tag{13}$$

Secondly, Eq. (5) is utilized in the form of the Eq. (14) to obtain $\mathbf{u}^{(init)}$.

$$V_p \frac{\rho_p^{(init)}\mathbf{u}_p^{(init)} - \rho_p^{(0)}\mathbf{u}_p^{(0)}}{\Delta t} + \sum_f \mathcal{M}_f^{(0)}\mathbf{u}_f^{(init)} - \sum_f \mu_f^{(0)}\mathbf{S}_f \bullet (\nabla \mathbf{u})_f^{(init)} - \sum_f \mu_f^{(0)}\mathbf{S}_f \bullet \left[(\nabla \mathbf{u})^T - \frac{2}{3}\nabla \bullet u\underline{I}\right]_f^{(0)} = -\sum_f \mathbf{S}_f P_f^{(0)} \tag{14}$$

Thirdly, it uses the Eq. (6) without the term second term on the right side in the form of the Eq. (15) to obtain the internal energy $e^{(n)}$. $k^{(0)}$ is the thermal conductivity coefficient.

$$V_p \frac{\rho_p^{(init)} e^{(n)} - \rho_p^{(0)} e_p^{(0)}}{\Delta t} + \sum_f \mathcal{M}_f^{(0)} e_f^{(n)} - \sum_f \left(k_f^{(0)}/C_v\right)\mathbf{S}_f \bullet (\nabla e)_f^{(n)} = -\sum_f \left(\mathcal{M}_f^{(0)}/\rho_f^{(n)}\right) P_f^{(0)} \tag{15}$$

At this point, the temperature is atualized using $e^{(n)}$ in Eq. (9), the viscosity $\mu^{(n)}$ using $T^{(n)}$ by the Sutherland's law and the thermal conductivity coefficient $k^{(n)}$ using $T^{(n)}$. The solver then proceeds to the pressure and velocity corrector loop, inside the time loop. The algorithm used is the Pressure-implicit with Splitting of Operation (PISO) algorithm (Issa (1986),Ferziger and Peric (2012)).

The solver repeats the PISO's loop until it reaches the convergence criteria. When it is achieved, the solver advances one step in time, and it continues until the end time of the simulation.

2.3 Adaptive Mesh Refinment Procedure

The adaptive mesh refinement (AMR) algorithm utilized can be described in the following phases: firstly, an initial coarse mesh is created with few volumes; secondly, the system of governing equations is solved in the available mesh for a number of iterations specified by the user; thirdly, the volumes within the refinement criteria are refined, which creates 8 new volumes for each refined one; fourthly, the most recent numerical solution is mapped to the new mesh, and it comes back to the first phase. This process continues until the end of the simulation.

The refinement criteria utilized was the velocity gradient magnitude, whose level of magnitude included only the oblique shock, which was the interest region for the adaptive refinement. The authors had to included the calculation of the velocity gradient in the object registry of the OpenFOAM's solver utilized and they recompiled it, so it was possible to use the cell refinement library already available in the OpenFOAM 1712 version. Moreover, the authors defined a maximum Courant number about 0.8, so the software could calculate the time step accordingly to the smallest volume, which changes due to the refinement.

2.4 Simulation Campaign

A double wedge airfoil was used to perform the 2D simulations because of its simplicity, which allows developing an analytical solution. The wings presented in Love (1949) were used to perform the 3D simulation because they have double wedge sections and were tested in a wind tunnel.

The 2D static mesh (without AMR) utilized was generated by the Salome software (Fig. 5). The tips of the airfoil were truncated to improve the mesh quality, and it is in the order of 1 to 3 control volumes near the airfoil. Three meshes were generated with a different number of volumes to investigate the mesh convergence.

There was also another mesh with AMR capabilities, which was made of hexagons. The regions which was marked to be refined were divided by 8 subregions, because the refinement process is three dimensional, but the numerical solutions only took account to two dimensions and ignored the variation in the third one. In Tab. 1 there are the parameters of the airfoil simulated, according to Fig. 1, and in Tab. 3 there are the parameters of the flow over the airfoil.

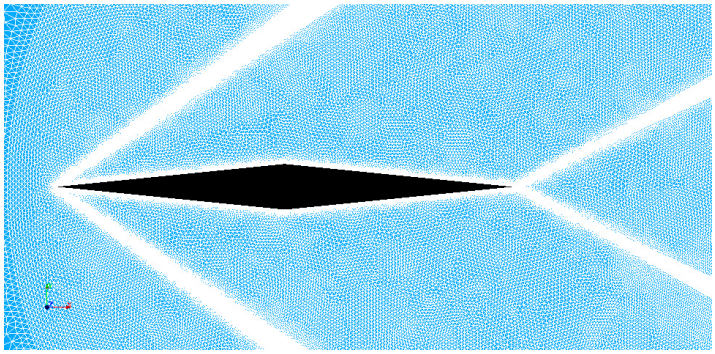

Figure 5. Static mesh generated in Salome with incresead resolution on the predicted shock line.

Table 1. Parameters of the double wedge airfoil simulated

t (mm)	c (mm)	Maximum thickness position
100.0000	1000	50% chrod

In the wing case, it was used only the mesh with AMR capabilities, based on the velocity gradient. An interval was settled to apply the refinement only in the shock region. The *snappyHexMesh* and *blockMesh* utilities were used to generate all the meshes with AMR capacity. The parameters of the wing are presented in Tab. 2, according to Fig. 4. Also, the Tab. 3 presents the parameters of the flow over the wing.

Table 2. Parameters of the wing simulated

M.A.C.(m)	b(m)	cr(m)	XC	YC	ϵ(degree)
0.046634	0.141122	0.070104	0.012619	0.005608	45.186226

Table 3. Parameters of the flow over the airfoil and over the wing

	U(m/s)	T(K)	P(Pa)	Ma	Re	gamma
Wing	545	303	24250	1.5620	3.7941E+5	1.4
Airfoil	650	300	100000	1,8722	4.12E+07	1.4

3. RESULTS AND DISCUSSION

In all simulations, the convergence was verified by the drag coefficient and the final residual error.

3.1 2D case

Figure 6a shows the variation between the drag coefficient and the number of cells, and the difference between the two most dense is only 0.6%. It suggests that the 2D static mesh with approximately 2.5 million cells presented an acceptable level of grid convergence,

Moreover, Fig. 6c shows that the results of pressure over the airfoil follow the theoretical value, except in the regions of leading and trailing edge, and in the middle of the airfoil. This is due to the change in the flow caused by the shock and expansion wave. In the 2D mesh with AMR case, it is noticed by the Fig. 6b that the AMR mechanism was capable to perform the shock fitting refinement where it was expected. Also, it did refinement in the expansion wave region.

Finally, Tab. 4 shows the precision of OpenFOAM in both 2D cases, and it suggests that the AMR mechanism can lead to better results, at least when there is high gradient regions and steady flow, such as in these cases.

3.2 3D case

Figure 7 shows the pressure distribution over the section located at 25.5% span of the wing simulated and it shows the result presented in Love (1949) by comparison. As one can notice, there is a satisfactory convergence between simulation and experimental results, and it suggests that AMR in steady flow applied in OpenFOAM can lead to actually good results.

Table 4. Results of the precision of the drag coefficient in the 2D case.

	Error
OpenFOAM without AMR	0.94%
OpenFOAM with AMR	0.10%

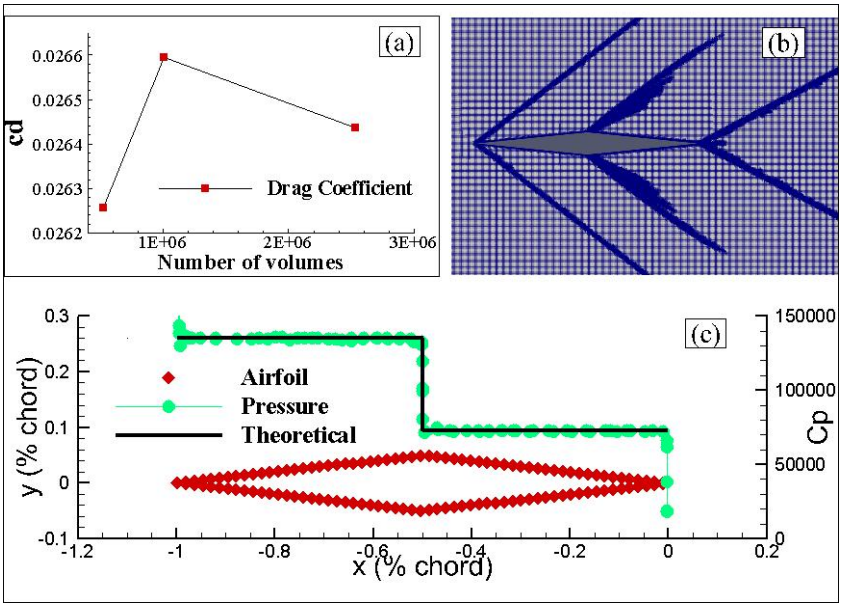

Figure 6. Simulation results for the 2D case: a) drag coefficient versus number of volumes b) final mesh from the AMR process c) pressure distribution over the airfoil and comparison to the theoretical value.

Table 5 presents the result of the drag coefficient over the wing simulated. It shows that the results are in a 7.5% range, so it is in good agreement with the experiments. It is important to mention that the Mach number of the simulation is 3.5% less than the experiment.

Table 5. Result of the drag coefficient in the 3D case

OpenFOAM 3D simulation with AMR	CD = 3.68E-02
Experimental results (LOVE ,1949)	CD = 3.99E-02
Percentual difference	7.65%

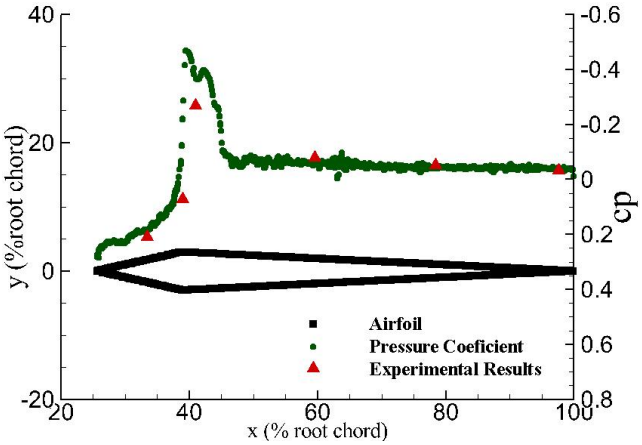

Figure 7. Pressure coefficient distribution obtained by the simulation and obtained by wind tunnel experiment (Love, 1949).

4. CONCLUSION

The results are satisfactory and they suggest that OpenFOAM has an acceptable level of precision in the simulation of supersonic external flows. Although, the problem is the performance of the software, and as a future work one could test different methodologies of process parallelization, such as the utilization of openMP- the present work used MPI- and Local Time Stepping.

5. ACKNOWLEDGEMENTS

The authors would like to thank CAALab and our family for their support.

6. REFERENCES

Anderson, J.D., 2003. *Modern compressible flow*. McGraw-Hill Series in Aeronautical and Aerospace Engineering. McGraw-Hill, New York, 3rd edition.

Asproulias, I., 2014. *RANS modelling for compressible turbulent flows involving shock wave boundary layer interactions*. Ph.D. thesis, University of Manchester.

Ferziger, J.H. and Peric, M., 2012. *Computational methods for fluid dynamics*. Springer Science & Business Media.

Issa, R.I., 1986. "Solution of the implicitly discretised fluid flow equations by operator-splitting". *Journal of computational physics*, Vol. 62, No. 1, pp. 40–65.

Love, E.S., 1949. "Investigations at supersonic speeds of 22 triangular wings representing two airfoil sections for each of 11 apex angles". In *NACA research memorandum L9D07*. National Advisory Committee for Aeronautics.

7. RESPONSIBILITY NOTICE

The authors are the only responsible for the printed material included in this paper.