ORIGINAL PAPER

Development and Validation of an Open Source CFD Code for Analysis of Aerospace Vehicles

Carlos Henrique Melo Souza[1,*] (iD), Amanda Chenu Romano[2] (iD), Angelo Passaro[3] (iD), Danton José Fortes Villas Boas[1] (iD)

1. Departamento de Ciência e Tecnologia Aeroespacial ROR – Instituto de Aeronáutica e Espaço – Divisão de Aerodinâmica, Controle e Estruturas – São José dos Campos/SP – Brazil. 2. Departamento de Ciência e Tecnologia Aeroespacial ROR – Instituto Tecnológico de Aeronáutica – Laboratório de Combustão, Propulsão e Energia – São José dos Campos/SP – Brazil. 3. Departamento de Ciência e Tecnologia Aeroespacial ROR – Instituto de Estudos Avançados – Divisão de Física Aplicada – São José dos Campos/SP – Brazil.

*Correspondence author: souzachms@fab.mil.br

ABSTRACT

In this work, a review of the theoretical aspects and an assessment to validate a Computational Fluid Dynamics (CFD) open-source code for applications in aerospace problems are discussed. The code uses a finite volume method, with cell-centered implementation, and it is suitable for simulations of inviscid, laminar, and turbulent flows. The code considers two-dimensional cases with unstructured meshes and employs the turbulence model known as Spalart-Allmaras. The implementation is detailed presenting the spatial discretization, including the upwind scheme, the linear reconstruction algorithm, and the calculation applying the method of gradients. The temporal discretization considers the application of a multistage explicit algorithm using a 5 stages Runge-Kutta method. The validation was done considering three cases of study: the inviscid shock tube, the laminar flat plate, and the flow over a rocket fairing. These cases are simulated using the software developed and the results are compared with analytical and experimental results. The rocket fairing case is related to the analysis of the Brazilian VLS launch during its transonic flight and it exemplifies the effect of the shock wave/boundary-layer interaction in its pressure distribution. The simulation results present a good agreement with the experimental results.

Keywords: Computational fluid dynamics; Aerodynamics; Aerospace vehicles; Open source.

INTRODUCTION

The aerospace applications of aerodynamics are a challenging area. In some cases, it is possible to use simplified methodologies like Newton's method or tangent-wedge methods to model aerospace vehicles (Rolim *et al.* 2020). But it is not always possible and, therefore, the implementation of more complex methods based in Computational Fluid Dynamics – CFD – are necessary. These simulations can be highly complex when, for example, an entire aircraft is modeled in order to study the effect of the fuselage in the vortex dynamic (Sustrino *et al.* 2020) or even when an entire rocket is modeled with the grid using chimera technique (Oliveira Neto *et al.* 2011). It is also possible to consider an intermediary situation in that CFD is applied in a more simplified and accessible way. This corresponds to the analysis of two-dimensional cases.

The 2-D approach reduces the computational costs while keeping several fundamental characteristics of the problem. There are several studies that attest this fact. For example, Azevedo and Korzenowski (2009) have presented the modeling of hypersonic inlet where the sensibility of the solver to several characteristics were analyzed in order to evaluate the configurations with best performances. McNamara *et al.* (2010) has presented an approximate modeling of hypersonic aeroelasticity using the flow over a double-wedge airfoil. The results of several simplified approaches were compared with the results obtained from CFD simulations considering a bi-dimensional model. The work of Silva and Pimenta (2019) has presented an analysis of aerodynamic properties of a reentry capsule using a two-dimensional model. In another study, Farrokhfal and Pishevar (2014) developed a CFD optimization method for airfoils based on the adjoint method. The objective was to reduce the compressibility drag or pitch moment of transonic airfoils without compromising its lift coefficient. All these cases show the importance of bi-dimensional analyses in the research of aerospace problems.

The CSU (Code of Simulation for Unstructured meshes) is a code of CFD developed to be used in tests of new models and to verify the results from other CFD software, considering bi-dimensional problems. The CSU is based on the finite volume method, using the density-based formulation and unstructured meshes. The methods used are common methods of CFD and the objective of this study is not to contribute with originality, but to present a review of the methodology used to develop this kind of code and make a computational tool available for the scientific community.

The software was developed to be an accessible open-source code that allows the fast assimilation and use. The implementation of the computationally expensive parts of the algorithm is done in C and the managing of the compilation and data visualization is done in Python. The software has been released in Linux and can be obtained at https://github.com/CarlosCHMS/CSU. The implementation of CSU makes use of the OpenMP package to parallelize the calculation processing. The use of parallelization reduces the time of processing, which is quite helpful for a CFD code. OpenMP allows it to subdivide the process into several threads, which is efficient if this number is smaller than the number of physical processor cores. Special attention must be given to the memory access since processing in different threads and trying to write in regions of a vector close together can result in errors and reduction of the performance.

The visualization of the results corresponds to plotting the graphics of flow contours and data. The contours are plotted using specialized algorithms based on unstructured meshes. However, it is necessary to know the values at the cell nodes not at the cell- centers. This requires the cell-centered values obtained from the simulation to be converted to node-centered values and it is made using the inverse of distance interpolation method (Tasri and Susilawati 2021).

SPATIAL DISCRETIZATION

The complete system of Navier-Stokes equations constitutes the governing equations of the flow and are presented in the integral form in Eq. 1 (Blazek 2005):

$$\frac{\partial}{\partial t} \int W d\Omega + \oint \left(F_c - F_v \right) dS = \int Q d\Omega \tag{1}$$

where F_c is the vector of convective fluxes, F_v is the vector of viscous fluxes, Q are the source terms, Ω is the control volume and the vector W is called "vector of conservative variables". In the bi dimensional approach with a one equation, the turbulence model has 5 components as presented in Eq. 2:

$$W = [\rho, \rho u, \rho v, \rho E, \rho \bar{\eta}] \tag{2}$$

where ρ is the density, u is the velocity in the x direction, v is the velocity in the y direction, E is the total energy per unit of mass and $\bar{\eta}$ is the modified turbulent viscosity, that is part of the Spalart-Allmaras implementation, and it is used in the turbulent cases. The code uses the finite volume method to solve the governing equations. The basic idea is to divide the flow domain in a set of small control volumes and, in each of them, the system of Eq. 1 is solved. For each of these volumes, the Eq. 3 is solved:

$$\frac{dW_I}{dt} = \frac{-R_I}{\Omega_I} \tag{3}$$

where the index I refers to each of these volumes, Ω_I is the volume of the control volume (area in 2-D formulation) and R_I is the residuals that are calculated with the numerical integration of the convective and viscous fluxes over the surface of the volume. The source terms will be considered null for all variables, except for the modified turbulent viscosity.

In general, the discretization of the volume can be made using a structured or unstructured approach. The structured meshes have the advantage of being easier to be generated in simple domains and of having a more direct implementation in the code. However, these advantages do not overcome the difficulties in generating meshes for more complex domains. Indeed, it can be said that the maturation of unstructured mesh technologies has ushered CFD into a new era. This advancement enables an increased speed of mesh generation, along with the ability to automate specific processes and employ adjoint-based mesh adaptation and shape optimization techniques (Mani and Dorgan 2023). Thus, the CSU was developed to use unstructured meshes.

The mesh generation is performed using the gmsh software (Geuzaine and Remacle 2000). The mesh format used as input is .su2. This mesh format was developed for the use of the SU2 software, which is an open-source CFD solver also currently in development and that has its characteristics detailed in Economon *et al.* (2016). This format was chosen because it was considered a format simple to read.

There are two alternatives to define the control volumes for an unstructured mesh: cell-centered scheme and cell-vertex scheme. In the cell-centered scheme, the control volumes are the same as the mesh cells and the values of the flow variables refer to the centroids of these cells. For the cell-vertex scheme, the variables are associated to the mesh vertices and the control volume is, in general, formed as some sort of combination among the cells that share the vertex. Both methods are similar in terms of accuracy and computational work but the cell-centered scheme has advantages in non-conformal grids (Blazek 2005). The cell-centered scheme was chosen in CSU software because its implementation was considered more direct and simpler. The calculation of the flow volumes and areas is made considering the alternatives of planar or axially symmetric solution. In the axisymmetric case, the symmetry axis is considered as the x axis.

For the calculation of the residual in the boundary of each mesh cell, the flux vectors must be presented. The convective flux vector can be written accordingly to Eq. 4:

$$F_c = \left[\rho V, \rho u V + n_x p, \rho v V + n_y p, \rho H V, \rho \bar{\eta} V \right] \tag{4}$$

where V is the component of the velocity normal to the cell face, n_x and n_y are the components of the vector normal to the cell face, p is the static pressure and H is the total enthalpy per unit of mass. In fact, there are Eq. 5 and 6:

$$V = u n_x + v n_y \tag{5}$$

$$H = E + \frac{p}{\rho} \tag{6}$$

and for perfect gases, the Eq. 7:

$$p = (\gamma - 1) \rho \left(E - \frac{u^2 + v^2}{2} \right) \tag{7}$$

where γ is the ratio of specific heats.

It is also important to observe that velocities and the moment fluxes must be considered with components normal and tangential to the cell face. This requires a transformation that consists in applying rotations before and after the flux calculation. That rotations make the flux coherent to the flux splitting method that will be applied (Mulder 1989). Making the rotation in the way that the u velocity component and moment component stay perpendicular to the cell border, the Eq. 4 became the Eq. 8:

$$F_c = [\rho u, \rho u^2 + p, \rho u v, \rho u H, \rho u \bar{\eta}] \tag{8}$$

Equation 8 together with a simple interpolation of variables in the cell face is not enough to calculate the flux correctly. It is well known that simple central schemes for convective terms are not stable. It is necessary to include an artificial dissipation to the central scheme or use an upwind scheme. In this work, the upwind scheme was chosen since it delivers a much better resolution of shocks and boundary layers than the central schemes (Blazek 2005).

The upwind schemes are also divided in several options of flux-vector splitting or flux-difference splitting schemes like Roe, HLLE, Osher, Leer, AUSM, among others. But each one of them has its limitations. Roe, for example, diverges at strong expansions even if entropy fix is used and it suffers from the carbuncle phenomenon, that consists of a numerical instability in capturing shock waves in a multidimensional domain. The HLLE does not incorporate information about contact discontinuities which makes it too dissipative. Osher's scheme fails in near vacuum condition. Leer scheme is very dissipative in 1-D contact discontinuities and generates glitches in the pressure near the edge of the boundary layer. AUSM (Advection upstream splitting method) is a quite interesting scheme since it can calculate strong shock waves. However, it generates numerical overshooting behind the shocks (Wada and Liou 1994). In fact, a variant of AUSM seems to be the best choice in terms of performance.

Wada and Liou (1994) proposed the AUSMDV that has some advantages like: high-resolution at contact discontinuities; conservation of enthalpy for steady flows and numerical efficiency. This scheme consists in a mixture of two other variants of AUSM that are the AUSMD and AUSMV. The AUSMD is a flux-Difference-splitting-biased scheme at the time that AUSMV is a flux-Vector-splitting-biased scheme. The convective flux coefficients are calculated as presented in the Eq. 9:

$$F_{c\frac{1}{2}} = 0.5 \left[(\rho u)_{\frac{1}{2}} (\Psi_L + \Psi_R) - \left| (\rho u)_{\frac{1}{2}} \right| (\Psi_R - \Psi_L) \right] \tag{9}$$

where the ψ assume the values 1, v, H and η-, for each of the sides left L or right R of the cell face. The value of $(\rho u)_{\frac{1}{2}}$ is calculated using the Eq. 10:

$$(\rho u)_{\frac{1}{2}} = u^+{}_L \rho_L + u^-{}_R \rho_R \tag{10}$$

where the values of $u^+{}_L$ and $u^-{}_R$ are calculated using a series of equations that takes into account sound velocity, normal velocity to the cell, pressure, and density in both sides of the cell face.

The only exception to Eq. 9 is the component referring to the moment normal to the cell face which is calculated using Eq. 11:

$$\left(\rho u^2 + p \right)_{\frac{1}{2}} = (0.5 + s) \left(\rho u^2 \right)_{\frac{1}{2} AUSMV} + (0.5 - s) \left(\rho u^2 \right)_{\frac{1}{2} AUSMD} + p_{\frac{1}{2}} \tag{11}$$

where $(\rho u)_{\frac{1}{2} AUSMV}$ corresponds to the calculation using the AUSMV scheme and $(\rho u)_{\frac{1}{2} AUSMD}$ the procedure related to the AUSMD scheme, S is a switching function calculated using a relation of pressures in the left and right side of the border and $\rho_{\frac{1}{2}}$ is calculated using a system of equations similar to the used for $u^+{}_L$ and $u^-{}_R$. The definition of $(\rho u^2)_{\frac{1}{2} AUSMV}$ and $(\rho u^2)_{\frac{1}{2} AUSMD}$ are presented in Eq. 12 and 13:

$$\left(\rho u^2 \right)_{\frac{1}{2} AUSMV} = u^+{}_L (\rho u)_L + u^-{}_R (\rho u)_R \tag{12}$$

$$\left(\rho u^2 \right)_{\frac{1}{2} AUSMD} = 0.5 \left[(\rho u)_{\frac{1}{2}} (u_L + u_R) - \left| (\rho u)_{\frac{1}{2}} \right| (u_R - u_L) \right] \tag{13}$$

The characteristic of the switch function was defined from numerical study cases based on the shock-tube problem. This analysis showed that AUSMV has a higher shock-capturing capability, however this scheme can result in strong oscillations in some cases. The switching function proposed is defined by Eq. 14:

$$s = 0.5 \min\left(1, K \frac{|p_R - p_L|}{\min(p_L, p_R)}\right) \tag{14}$$

where p_R and p_L are the pressures in the right and left side of the cell face and K is a constant parameter defined as 10. More details on the equations used in the implementation of the AUSMDV scheme can be obtained in the (Wada and Liou 1994).

The values of flow variables to be considered on the left and right sides of the cell face must be determined, considering the flow variables constant inside the cell result in solutions that are only first-order accurate. First-order-accurate solutions present a good convergence, however they are too much dissipative. Thus, it is necessary to achieve the second order accuracy. It can be done by using piecewise linear reconstruction of the solution (Barth and Jespersen 1989).

The basic idea of the method is to assume that the flow variables vary linearly inside the cell. So, the values at the cell faces are given by the Eq. 15 and 16:

$$U_L = U_I + \psi_I \left(\nabla U_I \cdot \vec{r_L}\right) \tag{15}$$

$$U_R = U_J + \psi_J \left(\nabla U_J \cdot \vec{r_R}\right) \tag{16}$$

where U_L and U_R represent the values on the left and right side of the cell face. These values are the ones used in the calculation of the flux. U_I and U_J are the flow variables at the center of the cells I and J. ∇U_I and ∇U_J are the gradients of the flow variables calculated for each cell-center. $\vec{r_L}$ and $\vec{r_R}$ are the vectors from the cell-centers to the center of the cells faces. Finally, ψ_I and ψ_J are the values of the limiters.

The gradients calculation is not only necessary for linear reconstruction, but it is used in the calculation of viscous fluxes, making it a key feature of applying the finite volume method. The most popular methods for calculating gradients are Green-Gauss and least-squares. The Green-Gauss method is based in the theorem of Green in 2-D (or Gauss theorem in 3-D) and it requires the numerical integration of the variable over the cell's surface. The least-square method is based on linear regression using the cell and its neighbors. These methods are popular mainly because they are not dependent on a particular cell's geometry and can be used in cells with arbitrary numbers of faces. It is quite convenient for the application in unstructured meshes as the ones used in CSU. The choice between these two methods is guided by the findings of Syrakos *et al.* (2017), who analyzed their performance across several mesh cases. Their conclusion is that, for arbitrary grids, the least-squares method is first-order accurate, while the Green-Gauss method is zero-order accurate. Higher accuracy orders can be achieved for specific grids. Considering these results, the least-squares method was chosen for implementation in CSU.

The limiter function ψ is crucial to the implementation of the second order upwind algorithm. Limiters avoid the wiggles that occur in the solution near shocks or discontinuities. The basic idea is that the limiter is close to 1 in regions of smooth solution, which results in a second-order scheme. In regions of intense variation of the flow variables, the limiter is close to zero, which results in a first-order solution. The Venkatakrishnan's limiter was chosen to be implemented because of its superior convergence properties (Blazek 2005). The limiter can be defined accordingly to Eq. 17 (Venkatakrishnan 1993):

$$\psi_I = \min_J \begin{cases} f\left(\Delta_{1,max}, \Delta_2, \epsilon\right), & if\ \Delta_2 > 0 \\ f\left(\Delta_{1,min}, \Delta_2, \epsilon\right), & if\ \Delta_2 < 0 \\ 1, & if\ \Delta_2 = 0 \end{cases} \tag{17}$$

where the Eqs. 18–21 define the additional terms:

$$f\left(\Delta_1, \Delta_2, \epsilon\right) = \frac{1}{\Delta_2}\left[\frac{\left(\Delta_1^2 + \epsilon^2\right)\Delta_2 + 2\Delta_2^2\Delta_1}{\Delta_1^2 + 2\Delta_2^2 + \Delta_1\Delta_2 + \epsilon^2}\right] \tag{18}$$

$$\Delta_{1,max} = U_{max} - U_I \tag{19}$$

$$\Delta_{1,min} = U_{min} - U_I \tag{20}$$

$$\Delta_2 = \left(\nabla U_I \cdot \vec{r_L} \right) \tag{21}$$

U_{max} and U_{min} are the values maximal and minimum of the flow variable U considering the cell I and all its neighbors. The min_I in Eq. 16 means that the value of the limiter must be calculated considering all possible values of $(\nabla U_I \cdot \vec{r_L})$ for each cell face and choosing the smaller one. The parameter ϵ avoid a possible division by 0 in Eq. 17 and can be used to improve the convergence of the solution.

Until this point the formulation was focused in the calculation of the convective fluxes, however there is also the viscous fluxes. The current version of the software has the implementation of the viscous fluxes relative to the laminar and turbulent flow. The viscous flux is given by the Eq. 22:

$$F_v = \left[0, n_x \tau_{xx} + n_y \tau_{xy}, n_x \tau_{xy} + n_y \tau_{yy}, n_x \Theta_x + n_y \Theta_y, n_x \tau_x + n_y \tau_y \right] \tag{22}$$

where the Eqs. 23–29 define its components:

$$\Theta_x = u \tau_{xx} + v \tau_{xy} + k \frac{\partial T}{\partial x} \tag{23}$$

$$\Theta_y = u \tau_{xy} + v \tau_{yy} + k \frac{\partial T}{\partial y} \tag{24}$$

$$\tau_{xx} = 2 \mu \left[\frac{\partial u}{\partial x} - \frac{1}{3} \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} \right) \right] \tag{25}$$

$$\tau_{yy} = 2 \mu \left[\frac{\partial v}{\partial y} - \frac{1}{3} \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} \right) \right] \tag{26}$$

$$\tau_{xy} = \mu \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) \tag{27}$$

$$\tau_x = \frac{1}{\sigma} \left(\rho \left(\eta_{laminar} + \bar{\eta} \right) \frac{\partial \bar{\eta}}{\partial x} \right) \tag{28}$$

$$\tau_y = \frac{1}{\sigma} \left(\rho \left(\eta_{laminar} + \bar{\eta} \right) \frac{\partial \bar{\eta}}{\partial y} \right) \tag{29}$$

where μ is the effective dynamic viscosity coefficient and k is the effective thermal conductivity. The terms τ_{xx} and τ_{yy} are the normal stresses and τ_{xy} is the shear stress. The terms Θ_x and Θ_y correspond to the work of viscous stresses plus the heat fluxes. The terms τ_x and τ_y are associated to the components of the viscous fluxes related to the turbulence model. It is possible to observe that it is necessary to calculate the gradient of the primitive variables u, v, T and $\bar{\eta}$. The value $\eta_{laminar}$ is the laminar kinematic viscosity. The least-square method can be used for this, but the results are related to the cell-center. But, to calculate the fluxes, it is necessary to obtain the values of the gradients in the cell faces.

A first approach to calculate the gradient in the cell face could be to use the average of the gradient values of the neighboring cell to estimate the gradient in the faces. However, Haselbacher and Blazek (2000) have shown that this simple approach results in the decoupling of the solution on quadrilateral meshes. The solution proposed to tackle this problem is to interpolate the gradient according to Eq. 30:

$$\nabla U_{1/2} = \bar{\nabla} U - \left[\bar{\nabla} U \cdot \vec{t} - \frac{(U_J - U_I)}{l} \right] \vec{t} \tag{30}$$

where the Eq. 31–33 define its components:

$$\bar{\nabla} U = \frac{1}{2} \left(\nabla U_I + \nabla U_J \right) \tag{31}$$

$$\vec{t} = \frac{\vec{r_{IJ}}}{l} \tag{32}$$

$$l = \left| \vec{r_{IJ}} \right| \tag{33}$$

The value of U refers to the velocities, the temperature are inputs of the equations and $\vec{r_{IJ}}$ is the radii vector that goes from the center of the cell I to the center of the cell J.

TURBULENCE MODELING

The flows considered in the modeling of aerospace vehicles are, in most cases, turbulent. Therefore, it is necessary to include an approach in the CSU code to deal with these problems. There are several common ways to tackle this problem, such as the use of turbulence models associated with Reynolds-averaged Navier-Stokes equations (RANS), the use of Large Eddy Simulation (LES), or even the use of Direct Numerical Simulation (DNS). However, the last two methods have a higher computational cost and generally require three-dimensional domains, which are not available for CSU in its current state of development. In fact, the use of RANS and turbulence models is the preferred choice for an initial approach to modeling turbulence.

The definition of RANS equations can be obtained from Rodriguez (2019). It involves modeling the turbulent flow by assuming that the flow variables can be represented as a sum of mean and oscillatory components. The flow is then solved by considering only the mean terms and variables. However, it is still necessary to determine the values of the mean terms that appear in the RANS equations, and this is where turbulence models come into play.

There is a wide variety of turbulent models in the literature. They are classified as Zero, One, or Two-equation models, depending on the number of flow variables introduced in the RANS equations. According to Blazek (2005), some of the most popular models include the Spalart-Allmaras one-equation model, the $k - \epsilon$ two-equation model, and the Menter SST (shear stress transport) two-equation model. In their analysis, Zingg and Godin (2009) evaluated different turbulence models and their applicability, concluding that the Spalart-Allmaras model performs best for aerodynamic flows. It provides accurate results for attached and mildly separated flows. However, flows with large-scale separation may require the use of the Menter SST model. Given this, the Spalart-Allmaras model is chosen for implementation in CSU, with the possibility of a future implementation of the Menter SST model.

The turbulence model is based on the one proposed by Spalart and Allmaras (1992) with the alterations proposed for a compressible implementation presented in Allmaras *et al.* (2012). The compressible version of the governing equation is given by Eq. 34:

$$\frac{\partial \bar{\eta}}{\partial t} + \nabla \left(\rho \, \vec{u} \, \bar{\eta} \right) = \rho \left(P - D \right) + \frac{1}{\sigma} \left[\nabla \left(\rho \left(\eta_{laminar} + \bar{\eta} \right) \nabla \bar{\eta} \right) + c_{b2} \rho \left(\nabla \bar{\eta} \right)^2 \right] - \frac{1}{\sigma} \left(\eta_{laminar} + \bar{\eta} \right) \nabla \bar{\eta} \, \nabla \rho \tag{34}$$

where the production and destruction terms are defined by Eqs. 35 and 36, respectively:

$$P = c_{b1} \tilde{S} \bar{\eta} \tag{35}$$

$$D = C_{w1} f_w \left[\frac{\bar{\eta}}{d} \right]^2 \tag{36}$$

The trip and laminar suppression terms are not considered. The variable S is the modified vorticity and it is calculated by using the modification proposed by Allmaras *et al.* (2012) in order to prevent negative values. The function f_w is defined by Eqs. 37–39:

$$f_w = g \left[\frac{\left(1 + C_{w3}{}^6 \right)}{\left(g^6 + C_{w3}{}^6 \right)} \right]^{\frac{1}{6}} \tag{37}$$

$$g = r + c_{w2} \left(r^6 - r \right) \tag{38}$$

$$r = min \left(\frac{\bar{\eta}}{\tilde{S} k^2 d^2}, 10 \right) \tag{39}$$

The eddy viscosity is calculated by using the Eq. 40–42:

$$\eta_{turb} = \bar{\eta} f_{v1} \tag{40}$$

$$f_{v1} = \frac{X^3}{X^3 + c_{v1}^3} \tag{41}$$

$$X^3 = \frac{\bar{\eta}}{\eta_{laminar}} \tag{42}$$

The effective viscosity and thermal conductivity are calculated by using the Eqs. 43 and 44:

$$\mu = \rho \left(\eta_{laminar} + \eta_{turb} \right) \tag{43}$$

$$k = C_p \rho \left(\frac{\eta_{laminar}}{Pr_{laminar}} + \frac{\eta_{turb}}{Pr_{turb}} \right) \tag{44}$$

where $Pr_{laminar}$ is the laminar Prandtl number assumed as 0.72 and Pr_{turb} is the turbulent Prandtl number assumed as 0.9. The values of the constants present in the model are $c_{b1} = 0.1355$, $\sigma = 2/3$, $c_{b2} = 0.622$, $k = 0.41$, $c_{w2} = 0.3$ and c_{w1} that is calculated by Eq. 45:

$$c_{w1} = \frac{c_{b1}}{k^2} + \frac{\left(1 + c_{b2} \right)}{\sigma} \tag{45}$$

TEMPORAL DISCRETIZATION

For the implementation of the temporal discretization there are two possible classes: the explicit and the implicit methods. The advantage of the implicit methods is the possibility of achieving faster and stable solutions by defining a higher CFL number. However, the explicit methods are easier to implement and parallelize. Explicit schemes generally require more calculation steps than the implicit ones, but each iteration is relatively cheap in computational terms (Leer *et al.* 1989). Because of these advantages, an explicit method was chosen to be implemented in the CSU.

The explicit method chosen uses a multistage scheme. The multistage method corresponds to the application of the Runge-Kutta method in which the new solution suffers several updates in a sequence of stages. This implementation can be seen in the equations of 5 stages presented in Eq. 46, below:

$$
\begin{aligned}
W_I^{(0)} &= W_I^n \\
W_I^{(1)} &= W_I^{(0)} - \alpha_1 \Delta t\, R_I^{(0)} \\
W_I^{(2)} &= W_I^{(0)} - \alpha_2 \Delta t\, R_I^{(1)} \\
W_I^{(3)} &= W_I^{(0)} - \alpha_3 \Delta t\, R_I^{(2)} \\
W_I^{(4)} &= W_I^{(0)} - \alpha_4 \Delta t\, R_I^{(3)} \\
W_I^{n+1} &= W_I^{(0)} - \alpha_5 \Delta t\, R_I^{(4)}.
\end{aligned}
\tag{46}
$$

Equation 46 show that in each stage, a preliminary solution $W_J^{(i)}$ is calculated and then used to calculate the residual $R_J^{(i)}$ of the next stage. At the end of the five steps, the final solution W_J^{n+1} is calculated. It is important to observe that this implementation of Runge-Kutta requires that only the initial solution and the newest preliminary solution need to be stored in the memory. It represents an advantage in terms of computational effort.

Using a multistage scheme is convenient since it improves the stability of the spatial discretization at the same time that allows an increase in the time-step used. The five order scheme applied to the software allows an increase of 2.5 times in the time step. The coefficients α can be defined considering first- or second-order spatial discretization schemes. However, for strong shocks, first-order schemes are preferred (Blazek 2005) since they prevent oscillations in the solution. The coefficients α_i implemented in the software are obtained from Leer *et al.* (1989) and are given in Table 1.

Table 1. Coefficients α_i for first order implementation.

Coefficient	α_1	α_2	α_3	α_4	α_5
Value	0.0533	0.1263	0.2375	0.4414	1.0

Source: Elaborated by the authors.

The time step is defined using the approach proposed by Vijayan and Kallinderis (1994) and modified by Blazek (2005). The time step is given by Eq. 47:

$$
\Delta t = min_I \left(\Delta t_I \right)
\tag{47}
$$

where Δt_I is the local time step and is given by Eq. 48:

$$
\Delta t_I = 2.5 \frac{\Omega_I}{\Lambda_{cI}^x + \Lambda_{cI}^y + C \left(\Lambda_{vI}^x + \Lambda_{vI}^y \right)}
\tag{48}
$$

The Λ are the spectral radii. The convective spectral radii are given by Eqs. 49 and 50:

$$
\Lambda_c^x = \left(|u| + c \right) d\hat{S}^x
\tag{49}
$$

$$
\Lambda_c^y = \left(|v| + c \right) d\hat{S}^y
\tag{50}
$$

where c is the velocity of sound in the cell, $d\hat{S}^x$ and $d\hat{S}^y$ are the projections of the cell volume in the direction x and y respectively. They can be calculated by Eqs. 51 and 52:

$$
d\hat{S}^x = \frac{1}{2} \sum_{J=1}^{N_I} \left| dS_J^x \right|
\tag{51}
$$

$$dŜ^y = \frac{1}{2} \sum_{J=1}^{N_f} \left| dS_J^y \right| \tag{52}$$

The summation corresponds to the sum over all the faces of the cell. The expressions for the viscous spectral radii Λ_{vl}^x and Λ_{vl}^y are more complex and can be found in Blazek (2005).

BOUNDARY CONDITIONS

The software supports four types of boundary conditions: wall, inlet, outlet and symmetry. The wall boundary condition is also divided in two cases, the inviscid and the viscous cases. In the inviscid case, all terms of the convective flux (Eq. 8) are null unless there is pressure. The pressure on the wall is calculated using the formulation presented by Whitfield and Janus (1984) that uses the characteristic boundary conditions. They use concepts of method of characteristics to estimate the pressure on the wall from the available flow variables in the closest cell-center. The expression of pressure is given by Eq. 53:

$$p_b = p_d + \rho_0 c_0 \left(n_x u_d + n_y v_d \right) \tag{53}$$

Where p_b is calculated using the values of pressure p_d, velocity u_d and v_d related to the cell adjacent to the border. The values of ρ_0 and c_0 are reference values of density and sound velocity, respectively, and are assumed from the cell adjacent to the border.

The convective flux used for the viscous wall is similar to the one used for the inviscid wall. The main difference is that now there are the viscous fluxes to be calculated. In order to do that, the gradient of the velocities must be calculated considering the velocity in the wall boundary null. That corresponds to the no slip condition on the wall. The heat flux is another condition to be defined in the viscous wall. There are some possibilities as: constant temperature wall, radiative wall or adiabatic wall. The adiabatic wall corresponds to the case in which no exchange of heat occurs through the wall, which implies in considering the heat flux or the gradient of temperature in the Eq. 23 and 24 to be zero. The adiabatic wall is the condition currently implemented in CSU.

The boundary conditions for subsonic inlets and outlets have also been implemented. The implementation follows the same approach proposed by Whitfield and Janus (1984) and is also based on the characteristic method approach. For the supersonic cases, the implementation differs. The input values are directly inserted as boundary values for the supersonic inlet boundary. For the supersonic outlet, the conditions at the boundary correspond to the flow variables of the cells adjacent to the boundary. This approach ensures that information flows upwind, ensuring the stability of the code.

The final boundary condition is symmetry, which assumes that the boundary divides the flow in two symmetrical and opposite regions. The implementation is done by assuming the existence of dummy cells. This approach supposes the existence of auxiliary cells outside the boundary with flow variables that are used in the calculation of the fluxes. For the symmetry condition, the flow variables at the adjacent wall are considered accordingly, Eq. 54:

$$\begin{aligned} \rho_{dummy} &= \rho \\[4pt] u_{dummy} &= -u \\[4pt] v_{dummy} &= v \\[4pt] E_{dummy} &= E \\[4pt] \bar{\eta}_{dummy} &= \bar{\eta} \end{aligned} \tag{54}$$

where the velocity u is considered normal and v tangent to the cell face. Using the flow variables of the cell and its dummy counterpart, it is possible to calculate the flux in the face using the same approach used for the interior cells.

When considering the turbulent model, the condition at the wall is $\bar{\eta} = 0$ and for the inlet the conditions is $\bar{\eta} = 5\eta_{laminar}$. These conditions agree with what is proposed in Allmaras *et al.* (2012) for a fully turbulent Spalart-Allmaras model.

NUMERICAL SIMULATIONS AND VALIDATION

The cases chosen for code validation have the objective of evaluating the behavior of the software considering different aspects. The shock tube problem can be used to verify the performance of the inviscid solver and the accuracy of the first and the second order algorithm. The flat plate problem has the objective of evaluating the performance of the laminar viscous implementation. Finally, the rocket fairing cases have as objective to evaluate the software globally, considering inviscid and turbulent solutions and comparing them to experimental results.

Sod shock tube

A typical shock tube is a tube generally made of metal with a circular or rectangular cross-section, in which two regions with gas at low pressure and high pressure are separated by a diaphragm. Under certain conditions, the diaphragm bursts and produces a shock wave that propagates in the low-pressure region. A schematic representation of the problem is presented in Fig. 1. This problem is closely related to the Riemann problem and is commonly used to test CFD algorithms. The problem can be solved using the Euler equations in one spatial dimension; however, it can also be solved using a rectangular tube in two dimensions.
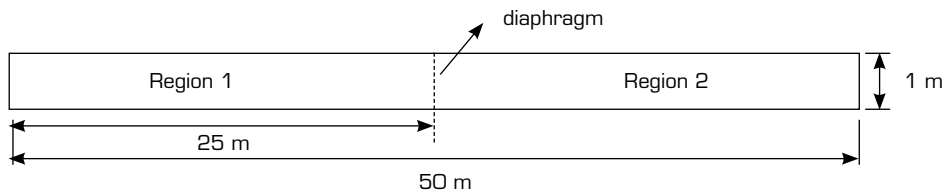
Source: Elaborated by the authors.

Figure 1. Schematic representation of the shock tube problem. Figure is not in scale.

The initial conditions used in the problem were obtained from Sod (1978) and correspond to what is called Sod shock tube problem. They correspond to assume in the region 1 $\rho_1 = 1.0$ and $p_1 = 1.0$ of the reference values, and for the region 2 the values are $\rho_2 = 0.125$ and $p_2 = 0.1$ of the reference values. The initial velocity is equal to zero for both regions. The CSU code works with atmospheric air and requires a different set of variables in the input data. The inputs are the initial values of pressure, temperature and both velocity components. Considering the Sod inputs and the reference values of pressure and temperature of $1.0 \cdot 10^5 Pa$ and $300K$, the initial conditions for the two regions of the 2-D case are given by Eq. 55:

$$\left[p_1, T_1, u_1, v_1 \right] = \left[1.0 \cdot 10^5\,Pa, 300\,K\,0.0\,\frac{m}{s}, 0.0\,\frac{m}{s} \right] \text{ and}$$

$$\left[p_2, T_2, u_2, v_2 \right] = \left[1.0 \cdot 10^4\,Pa, 240\,K, 0.0\,\frac{m}{s}, 0.0\,\frac{m}{s} \right]$$

(55)

The mesh used covers a rectangular domain measuring 50m x 1m and is generated using 3022 triangular elements. The division of the initial regions occurs at the position of 25m. The simulation results are compared with the analytical solution obtained by applying the method of characteristics. The transient simulation is conducted until a physical time of 0.02 seconds. This time was chosen to be sufficiently long to observe the discontinuities in the solution well separated, yet short enough to avoid wave reflections with the walls at the tube's extremities, which could complicate the solution using the characteristics method.

The impact of using a first or second-order method can be observed in Fig. 2. While the first-order method presents some discrepancies regarding the characteristic solution in the shock regions, the second-order solution exhibits good agreement across all regions. These results align with expectations for this type of simulation and demonstrate the software code's effectiveness. Additionally, it should be noted that some preliminary results of this problem exhibited overshoots in the second-order solutions near the discontinuities. This issue was resolved by reducing the value of the parameter ϵ in Venkatakrishnan's limiter in Eq. 18. A value of $\epsilon = 0.001$ resulted in solutions without overshoots. This observation is consistent with Venkatakrishnan (1993), which states that the increasing of ϵ can lead to discrepancies in the solutions.

Source: Elaborated by the authors.
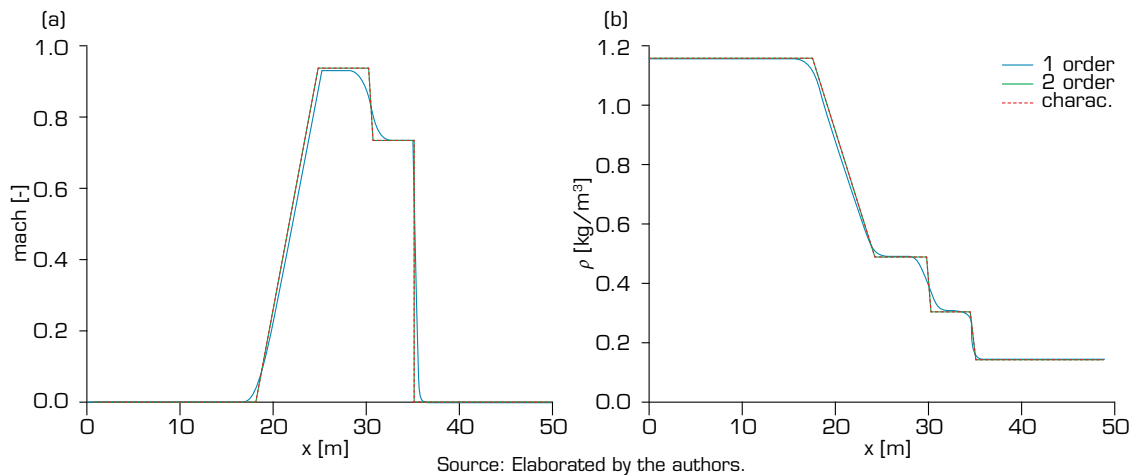
Figure 2. Results from the simulation of the shock tube for the instant of 0.02s. (a) Mach number along the shock tube; (b) density along the shock tube.
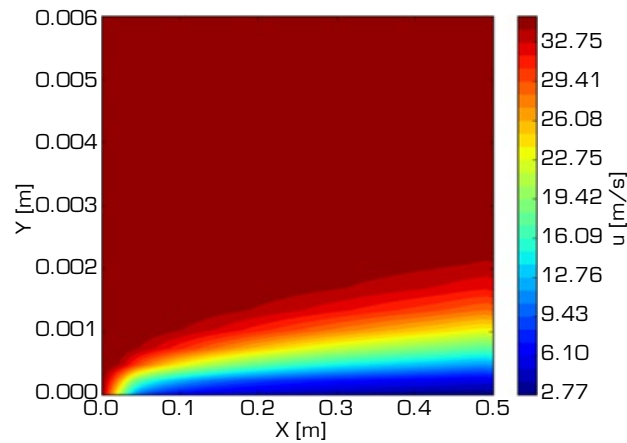
Flat plate

The laminar flow on a flat plate is a convenient problem to verify the implementation of the viscous flux since the Blasius analytical solution can be used for comparisons.

The problem is set as a flow with velocity $u = 34.75^{m/s}$ or Mach number of 0.1. The pressure is set as $p = 1 \cdot 10^5 Pa$ and the temperature is $T = 300K$. The domain is rectangular of 0.5 m x 0.006 m and the mesh has 2,786 elements. The results are presented in Figs. 3 and 4.

Fig. 4 on the left, shows a good agreement for the velocity profile at $x = 0.5m$ obtained from the CFD code and the Blasius solution. The Fig. 4 on the right, also shows a good agreement of the shear stress in the region far from the plate leading edge, however the curves do not fit well for values lower than 0.07 m. This disagreement is related to the fact that, close to the leading edge, the boundary layer has a thickness of the same order of the height of the cell adjacent to the wall. Because of this, the solver is not able to resolve the boundary layer close to the edge and incorrect results are obtained.

The validation will be made by using wind tunnel results related to the VLS (Veículo Lançador de Satélites), a former Brazilian project of vehicle launchers. The experiments were conducted at the ONERA (Office National d'Etudes et de Recherches Aérospatiales), France, using a 1/15 scale model (Gauthier, 1989). The CFD simulations considered the same geometry of the scale model used in the wind tunnel experiments.

The experiments generated pressure coefficient c_p distributions for several cases of Mach and Reynolds numbers. However, a detailed analysis of these results permitted the verification of two cases that have a remarkable effect of interaction between shock and boundary layer. These cases are: first one $Mach = 0.9$ and $R_e = 1.68 \cdot 10^6$ and the second one $Mach = 1.05$ and $R_e = 1.57 \cdot 10^6$. It is desirable to use these cases in the validation process because they allow the observation of the effect of the boundary layer that is linked to the turbulence model implemented. The Reynolds numbers were calculated using the vehicle diameter as reference length.

Source: Elaborated by the authors.

Figure 3. Velocity contours for the flat plate laminar flow. Out of scale figure.
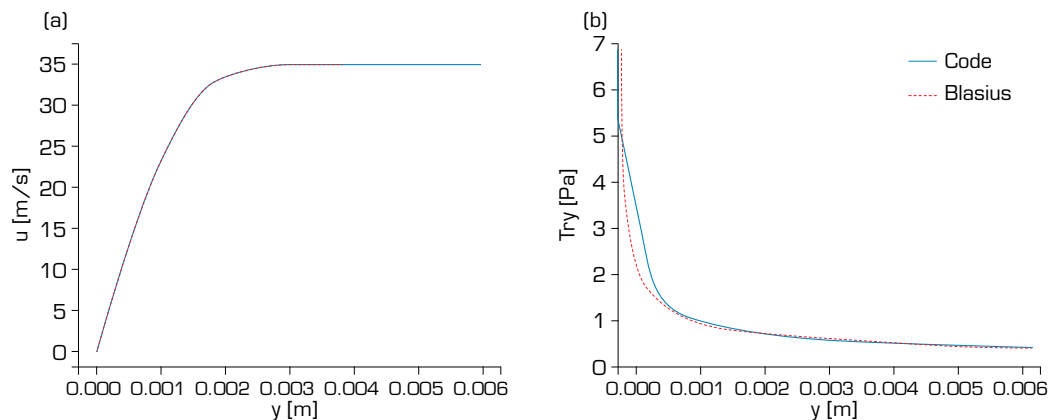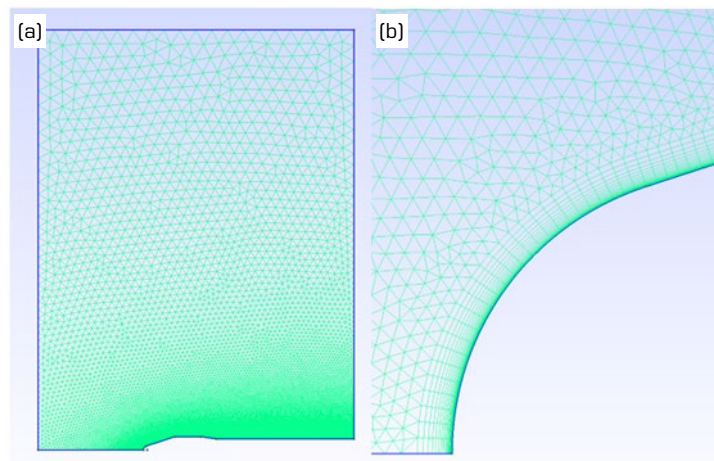

Source: Elaborated by the authors.

Figure 4. Results of the flat plate simulation. (a) The velocity profile on the flat
plate at position x=0.5 m; (b) The shear stress along the flat plate.

The explanation of the effects observed in these two cases is as follows: the case of *Mach* = 0.9 is transonic, with two near normal shocks being formed after the expansion corners in the fairing. These two shocks are strong enough to generate observable effects in the distribution of C_p when compared with the inviscid case. The case of *Mach* = 1.05 has a unique shock interacting with the boundary layer at the end of the fairing. Curiously, greater Mach numbers do not generate remarkable interactions. The reason is that, despite the increase in the external flow Mach number, the pressure ratio in the shock at the end of the fairing is reduced, resulting in weak shocks and a less pronounced interaction with the boundary layer.

The mesh created has 87,210 triangular and quadrilateral elements. The quadrilateral elements are used close to the surface of the vehicle that has as objective to improve the solution of the boundary layer. The quadrilateral layer has a growth rate of 1.2 and was carefully made in order to result in y+~1 in the wall. The meshes can be seen in Fig. 5. The domain is rectangular with the following boundary conditions: the left side being the input, the right side being the output, the upper side being wall and the lower side being the symmetry and wall. The walls are considered adiabatic.

Figure 6 shows the Mach number contours obtained from the solution for Mach 0.9, considering the turbulence model, alongside a Schlieren photograph of the flow under the same conditions. It is possible to observe the formation of a shock wave caused by the flow acceleration, which becomes supersonic after the first corner. After that, the shock happens and makes the flow subsonic again. It is also remarkable that a relatively small region of higher Mach closes the second corner. This is associated with a second expansion through the corner and a second shock. In the Mach contours, it is also possible to observe a detail of the boundary layer in the shock region. The comparison of the contours and the photography allows the conclusion that the flow patterns were captured by the CFD simulation.

Source: Elaborated by the authors.

Figure 5. Mesh used in the simulations: (a) the entire domain; (b) a detail of the quadrilateral layer on the fairing tip.
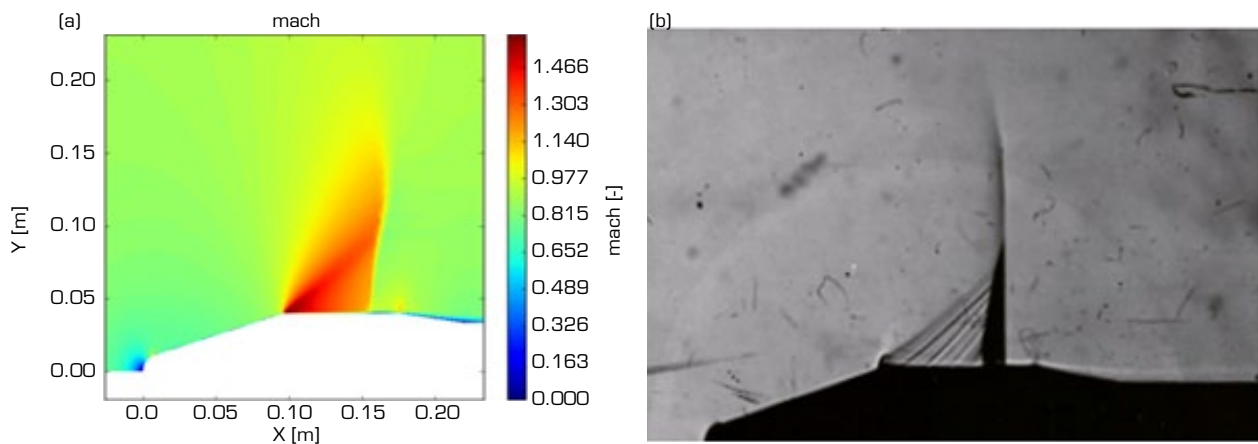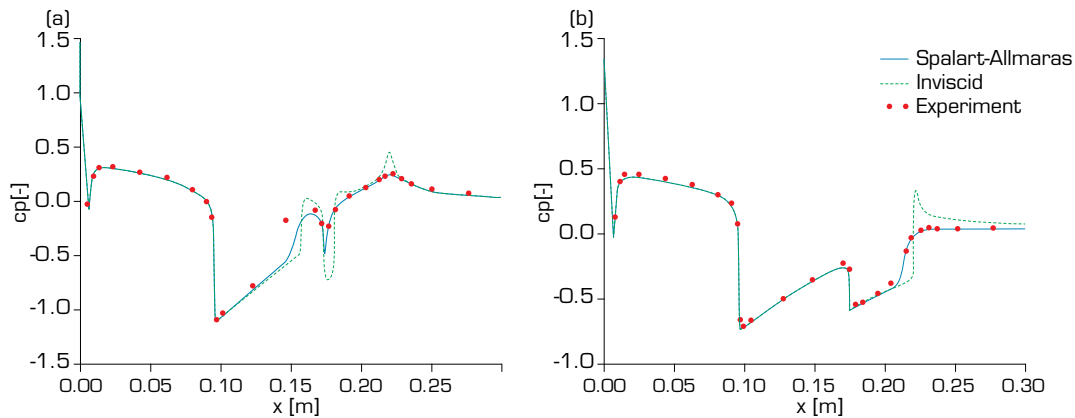

Source: Elaborated by the authors using data from Gauthier (1989).

Figure 6. Comparison between computational and experimental results: (a) contours of Mach number for the case of Mach 0.9 for the fairing considering the turbulence model; (b) the corresponding Schlieren photography.

Figure 7 presents a comparison between the simulation and experimental results for the pressure coefficient, cp. Up to x = 0.15 m for Mach 0.9 and x = 0.2 m for Mach 1.1, both inviscid and viscous solutions match the experimental data. However, beyond these positions, disparities emerge between the inviscid solutions and experimental results, while the results obtained with the Spalart-Allmaras turbulence model continue to exhibit good agreement. The cause of these discrepancies can be attributed to the occurrence of shocks in those regions and their interaction with the boundary layer. This phenomenon is well documented in Houghton and Carpenter (2003) and has been observed in a rocket fairing by Mata *et al.* (2017).

After the shocks, there is an increase in pressure that propagates upstream through the subsonic part of the boundary layer. This alteration in the flow around the shock region leads to a smoother pressure transition compared to what would be expected in an inviscid shock. This clarifies the differences obtained by the inviscid solutions and shows the importance of incorporating a turbulence model in these simulations. In fact, the agreement observed is not exclusive to the Spalart-Allmaras model and could be obtained by any other turbulence model that correctly represents the turbulent boundary layer. However, the results permit the conclusion that the Spalart-Allmaras turbulence model, as implemented in CSU, was sufficiently accurate to represent the flow and match the experimental results.

Source: Elaborated by the authors using data from Gauthier (1989).

Figure 7. Distribution of c_p on the fairing surface. (a) case of Mach number 0.9. (b) case of Mach number 1.05.

Additionally, Fig. 8 illustrates the temperature distribution on the fairing obtained from CSU using the Spalart-Allmaras model. It reveals a peak near the initial position, corresponding to the stagnation point on the nose, followed by a valley representing the circular part of the nose. Around x = 0.1 m, there is another valley attributed to the first fairing corner, which corresponds to an expansion fan. The curves remain relatively close until approximately x = 0.12 m, after which they begin to diverge due to differences in shock configurations for Mach 0.9 and 1.05 cases. Even with these differences, the curves approximate each other again after the end of the fairing at x = 0.23 m. It is also remarkable that the total range of temperature variation was around 16 K.
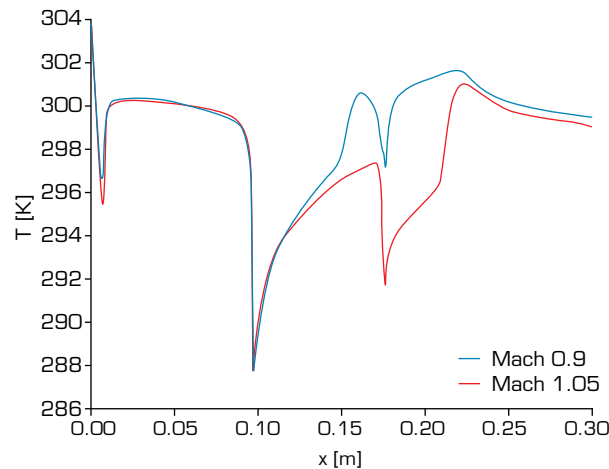
Source: Elaborated by the authors.

Figure 8. Temperature distribution on the fairing surface. Results from the application of CSU with the Spalart-Allmaras turbulence model.

CONCLUSIONS

The study presented the theory involved in the implementation of the code CSU and its validation through some test cases. The spatial discretization was detailed using the AUSMVD scheme combined with the application of least-squares for gradient calculation and Venkatakrishnan's limiter. The turbulence modeling based on Spalart-Allmaras is presented. The temporal discretization using multistage scheme based in the application of the Runge-Kutta method is also detailed.

In terms of validation of the software, CSU was applied to three problems. The first was the shock tube problem and the results were compared with the analytical results obtaining an excellent agreement considering the second order algorithm. The simulation

of the flat plate was made considering the laminar solver, which resulted in a good agreement with the Blasius solution. Finally, the solver using inviscid and Spalart-Allmaras approach is used in the simulation of the flow over a rocket fairing. The CFD results agreed with the experimental data and permitted the validation of the implementation of the turbulent model.

CONFLICT OF INTEREST

Nothing to declare.

AUTHOR CONTRIBUTIONS

Conceptualization: Souza CHM; **Acquisition of funding:** Passaro A; **Research:** Souza CHM and Romano AC; **Methodology:** Souza CHM and Romano AC; **Software:** Souza CHM; **Supervision:** Passaro A; **Validation:** Souza CHM and Villas Boas DJF; **Visualization:** Souza CHM; **Writing - Preparation of original draft:** Souza CHM and Villas Boas DJF; **Writing - Proofreading and editing:** Souza CHM and Passaro A;

FUNDING

ACKNOWLEDGMENTS

REFERENCES

Allmaras SR, Johnson FT, Spalart PR (2012) Modifications and Clarifications for the Implementation of the Spalart-Allmaras Turbulence Model. Paper presented at Seventh International Conference on Computational Fluid Dynamics (ICCFD7). Big Island, Hawaii.

Azevedo JLF, Korzenowski H (2009) An assessment of unstructured grid finite volume schemes for cold gas hypersonic flow calculations. J Aerosp Technol Manag 1(2):135-152. https://doi.org/10.5028/jatm.2009.0102135152

Barth TJ, Jespersen DD (1989) The design and application of upwind schemes on unstructured Meshes. Paper presented at 27th Aerospace Sciences Meeting. AIAA; Reno, United States. https://doi.org/10.2514/6.1989-366

Blazek J (2005) Computational Fluid Dynamics: Principles and Applications. Amsterdam: Elsevier.

Economon TD, Palacios F, Copeland SR, Lukaczyk TW, Alonso JJ (2016) SU2: An open-source suite for multiphysics simulation and design. AIAA J 54(3):828-846. https://doi.org/10.2514/1.J053813

Farrokhfal H, Pishevar AR (2014) Optimization of airfoils for minimum pitch moment and compressibility drag coefficients. J Aerosp Technol Manag 6(4):395-406. https://doi.org/10.5028/jatm.v6i4.403

Gauthier J (1989) Essais de mesure de pression sur la maquette complète au 1/15 du lanceur Brésilien dans les souffleries S2MA et S3MA, ONERA.

Geuzaine C, Remacle J-F (2000) A three-dimensional finite element mesh generator with built-in pre- and post-processing facilities. Gmsh. [accessed Jan 15 2022]. http://www.geuz.org/gmsh/

Haselbacher A, Blazek J (2000) On the Accurate and Efficient Discretization of the Navier-Stokes Equations on Mixed Grids. AIAA J 38(11):2094-2102. https://doi.org/10.2514/2.871

Houghton EL, Carpenter PW (2003) Aerodynamics for Engineering Students. Burlington: Butterworth Heinemann.

van Leer B, Tai C-H, Powell KG (1989) Design of optimally smoothing multi-stage schemes for the Euler equations. Paper presented at 9th Computational Fluid Dynamics Conference. AIAA; Buffalo, United States. https://doi.org/10.2514/6.1989-1933

Mani M, Dorgan AJ (2023) A perspective on the State of Aerospace Computational Fluid Dynamics Technology. Annu Rev Fluid Mech 55:431-457. https://doi.org/10.1146/annurev-fluid-120720-124800

Mata HO, Falcão Filho JBP, Avelar AC, Carvalho LMMO, Azevedo JLF (2017) Visual Experimental and Numerical Investigations Around the VLM-1 Microsatellite Launch Vehicle at Transonic Regime. J Aerosp Technol Manag 9(2):179-192. https://doi.org/10.5028/jatm.v9i2.676

McNamara JJ, Crowell AR, Friedmann PP, Glaz B, Gogulapati A (2010) Approximate Modeling of Unsteady Aerodynamics for Hypersonic Aeroelasticity. J Aircr 47(6):1932-1945. https://doi.org/10.2514/1.C000190

Mulder WA (1989) High-resolution Euler solver. Paper presented at 9th Computational Fluid Dynamics Conference. AIAA; Buffalo, United States. https://doi.org/10.2514/6.1989-1949

Oliveira Neto JA, Basso E, Azevedo JLF (2011) Aerodynamic study of sounding rocket flows using Chimera and patched multiblock meshes. J Aerosp Technol Manag 3(1):87-98. https://doi.org/10.5028/jatm.2011.03016810

Rodriguez S (2019) Applied computational fluid dynamics and turbulence modeling. Cham: Springer. https://doi.org/10.1007/978-3-030-28691-0

Rolim TC, Cintra SC, Pellegrini MMC (2020) Development and Application of Computational tool using local surface inclination methods for preliminary analysis of hypersonic vehicles. J Aerosp Technol Manag 12:e2120. https://doi.org/10.5028/jatm.v12.1123

Silva LM, Pimenta AP (2019) Computational analysis of the stability of the Brazilian atmosphere reentry satellite (SARA). Braz J Phys 49:360-371. https://doi.org/10.1007/s13538-019-00654-9

Sod AG (1978) A survey of several finite difference methods for systems of nonlinear hyperbolic conservation laws. J Comput Phys 27(1):1-31. https://doi.org/10.1016/0021-9991(78)90023-2

Spalart PR, Allmaras SR (1992) A One-Equation Turbulence Model for Aerodynamic Flows. Paper presented at 30th Aerospace Sciences Meeting and Exhibit. AIAA; Reno, United States. https://doi.org/10.2514/6.1992-439

Sustrino S, Deendarlianto D, Rohmat TA, Wibowo SB, Iswahyudi S (2020) Vortex Dynamics Analysis of Straight-Body-Type-Fuselage Fighter Using CFD simulation. J Aerosp Technol Manag 12:e1020. https://doi.org/10.5028/jatm.v12.1104

Syrakos A, Varchanis S, Dimakopoulos Y, Goulas A, Tsamopoulos J (2017) A critical analysis of some popular methods for the discretization of the gradient operator in finite volume methods. Phys Fluids 29(12):7103. https://doi.org/10.1063/1.4997682

Tasri A, Susilawati A (2021) Accuracy of cell centres to vertices interpolation for unstructured mesh finite volume solver. J Inst Eng 7(4):E06875. https://doi.org/10.1016/j.heliyon.2021.e06875

Venkatakrishnan V (1993) On the Accuracy of limiters and convergence to steady state solutions. Paper presented at 31st Aerospace Sciences Meeting. AIAA; Reno, United States. https://doi.org/10.2514/6.1993-880

Vijayan P, Kallinderis Y (1994) A 3D Finite-Volume Scheme for the Euler Equations on Adaptive Tetrahedral Grids. J Comput Phys 113(2):249-267. https://doi.org/10.1006/jcph.1994.1133

Wada Y, Liou M-S (1994) A flux splitting scheme with high-resolution and robustness for discontinuities. Paper presented at 32nd Aerospace Sciences Meeting and Exhibit. AIAA; Reno, United States. https://doi.org/10.2514/6.1994-83

Whitfield DL, Janus JM (1984) Three-dimensional unsteady Euler equations solution using flux vector splitting. Paper presented at 17th Fluid Dynamics, Plasma Dynamics, and Lasers Conference. AIAA; Snowmass, United States. https://doi.org/10.2514/6.1984-1552

Zingg DW, Godin P (2009) A perspective on turbulence models for aerodynamic flows. Int J Comput Fluid Dyn 23(4):327-335. https://doi.org/10.1080/10618560902776802