Implementation of Segregated Solvers for Incompressible and Compressible Flow in Collocated Grids

Francisco de Lemos Cabral Granadeiro Martins
francisco.c.g.martins@ist.utl.pt

Instituto Superior Técnico, Universidade de Lisboa, Portugal

June 2018

Abstract

An extension to the SIMPLE algorithm was developed, with the objective of solving compressible flow problems. The implementation of the algorithm was explained in detail, and its main differences to the original SIMPLE for incompressible flows are emphasized. Special attention was given to the discretization, in cartesian and uniform grids, of the viscous terms in the energy and momentum equations, in compressible flow, with constant and variable viscosity.

To test the algorithm, several benchmark cases available in the literature were solved, in compressible flows over a wide range of Mach numbers.

A pressure correction stabilization procedure is included in the algorithm, where the main diagonal dominance of the matrix is guaranteed by an under relaxation factor calculated automatically.

Two programs featuring the algorithm were developed in Matlab: one to solve the quasi 1-D convergent-divergent nozzle problem, and another one to solve 2D test cases for both viscous and inviscid fluids. In the latter one, the upwind, linear and Gamma NVD schemes were implemented, and their performance compared.

The ability to solve unsteady flows was also included in the algorithm, wherein the implicit Euler scheme was used, and was tested on the Mach 3 inviscid flow over a forward facing step. The impact of the numerical errors generated on the step corner and the corrections suggested by other studies were also analysed.

Keywords: Finite Volume Method, Incompressible Flow, Compressible Flow, Pressure-Velocity Coupling Algorithms, Computational Fluid Dynamics

1. Introduction

1.1. Segregated Solvers

Pressure based solvers were originally designed to solve incompressible flows, by using pressure as the variable which ensures that continuity is respected. One of the most famous, and perhaps oldest, pressure based methods is the SIMPLE algorithm [1]. It is a guesser-corrector algorithm for pressure, that was proposed to be used in staggered grids (preventing the pressure checker board problem) and introduced the concept of pressure correction to guarantee the continuity constraint.

Since then, many different adaptations of this method have been proposed: like the SIMPLER [2], and the SIMPLEC [3]. To overcome the pressure checker board issue in colocated grids, [6] proposed what is known as the Rhie and Chow interpolation. In this interpolation, the face velocities are calculated by taking into account the pressure gradients not only in the face, but also in the control volumes that share it. In 1985 the PISO algorithm was proposed by[4], and, in the year after, all of these algorithms were compared by [5] for steady flows.

One of the first pressure based solvers that was able to handle both supersonic and subsonic flows was proposed by [7] which could predict the supersonic viscous flow near walls.

Later, an all-Mach solver was proposed by [8], which extended the incompressible SIMPLE-like algorithms in staggered grids. This was accomplished by writing the compressible form of the continuity equation and transforming it into a pressure correction one. Most pressure based all-Mach solvers since then follow this formulation.

In 1988, [9] presented an all-Mach version of the SIMPLER algorithm using staggered grids, and [10] devised an all-Mach pressure based inviscid flow solver with adaptive grids.

The SIMPLES was introduced by [11] for collocated and non-orthogonal meshes, which included the Rhie and Chow interpolation, and was tested using several schemes. A multigrid technique for staggered grids was proposed in the same year by

[12], something that had also been proposed earlier by [13] for the compressible version of the PISO algorithm.

Another boundary fitted method was presented in [14], using the Rhie and Chow interpolation and the upwind scheme. The algorithm devised by [15] had an ameliorated accuracy by using a deferred correction blending between the upwind and central differencing schemes.

An all-Mach pressure based solver was used to simulate turbulent flow by [16]. The concept of density retardation was introduced to improve stability. This concept was also used by [17] for turbomachinery flows.

A high-resolution scheme was implemented by [18] for compressible flows, and another extension to the algorithm, was proposed in [19], allowing it to solve turbulent cavitating flows.

1.2. Coupled Solvers

An incompressible pressure based solver on structured grids was developed by [20], who later developed two versions of all-Mach solvers: [21] for inviscid flows in unstructured grids, and [22] used to solve turbulent flows and employed a rotating frame of reference.

An algorithm for unstructured grids where the momentum, pressure and energy equations are all coupled was proposed by [23]. An alternative approach (also for unstructured grids) was proposed in [24], which involves solving the coupled momentum and pressure equations in a constant temperature cycle, followed by the computation of the energy equation.

2. Governing Equations

The set of equations that governs the compressible flow of newtonian fluids is comprised by the continuity equation, the Navier-Stokes equations and the energy equation:

$$D\left(\rho\mathbf{u}\right) = \frac{\partial \rho}{\partial t} + \nabla \cdot \left(\rho\mathbf{u}\right) = 0, \tag{1}$$

$$\frac{\partial \left(\rho\mathbf{u}\right)}{\partial t} + \nabla \cdot \left(\rho\mathbf{u} \otimes \mathbf{u}\right) = -\nabla p + \nabla \cdot \left(\boldsymbol{\tau}\right) + \rho\mathbf{g}, \tag{2}$$

$$\frac{\partial \left(\rho h_t\right)}{\partial t} + \nabla \cdot \left(\rho h_t \mathbf{u}\right) = \nabla \cdot \left(\boldsymbol{\tau} \cdot \mathbf{u}\right) - \nabla \cdot \left(k \nabla T\right) + \tag{3}$$

$$+ \frac{\partial p}{\partial t} + \mathbf{f}_b \cdot \mathbf{u} + q_v,$$

here $\boldsymbol{\tau} = \left(\mu \nabla \boldsymbol{u} + \mu \nabla \boldsymbol{u}^T\right) - \frac{2}{3}\mu\left(\left(\nabla \cdot \boldsymbol{u}\right)\boldsymbol{I}\right)$ is the viscous stress tensor. It can be simplified for flows in which the viscosity is considered to be constant, leading to: $\nabla \cdot \boldsymbol{\tau} = \mu\left(\nabla^2 \boldsymbol{u} + \frac{1}{3}\nabla\left(\nabla \cdot \boldsymbol{u}\right)\right)$.

In the energy equation, $h_t = c_p T + \frac{1}{2}\|\mathbf{u}\|^2$ is the total enthalpy and $\nabla \cdot \boldsymbol{\tau} \cdot \mathbf{u}$ is the energy lost due to viscous effects, which is usually neglected for subsonic flows.

These equations can be simplified for incompressible flow, but, even without those simplifications, they are still valid in that regime.

Another important equation is that of transport of a general scalar. After the integration over a control volume, and some simplifications:

$$V\frac{\partial \left(\rho\phi\right)}{\partial t} + \sum_{f=1}^{F} \rho_f \phi_f \left(\mathbf{u}_f \cdot \mathbf{n}_f\right) A_f = \tag{4}$$

$$\sum_{f=1}^{F} \Gamma_f \left(\nabla\phi\right)_f \cdot \mathbf{n}_f A_f + V\dot{q}_\phi$$

(5)

3. Discretization Procedure

In this section, the discretization schemes used in this work will be briefly detailed by using the general scalar ϕ:

3.1. Convective Term

Upwind Differencing Scheme A very common way to discretize the convective term is by using the upwind differencing scheme (UDS), which approximates ϕ_f as:

$$F_e\phi_e = Fe \begin{cases} \phi_E : Fe < 0 \\ \phi_P : Fe > 0 \end{cases} \tag{6}$$

Linear Interpolation An alternative to the UDS is the linear interpolation scheme LIN. In the case of a uniform grid $F_e\phi_e = Fe\left(\frac{\phi_E + \phi_P}{2}\right)$.

Gamma NVD Scheme A third alternative, the Gamma scheme presented by [28], allows the upwind scheme to be used in regions where gradients are substantial and LIN wherever this is not the case. It also allows the use of a blending scheme to smoothen the transition region between the two previous schemes, and is based on the NVD concept.

3.2. Diffusive Term

The most common way to discretize the diffusive terms is via the central differencing scheme, where the derivatives are calculated as:

$$\Gamma_e \left(\frac{\partial \phi}{\partial x}\right)_e A_e = \Gamma_e \frac{\phi_E - \phi_P}{\Delta x} A_e \tag{7}$$

3.3. Temporal Term

Implicit Euler In this work, the temporal discretization was made utilizing the implicit Euler scheme:

$$V\frac{\partial \rho\phi}{\partial t} + R = V\frac{\left(\rho\phi\right)^{n+1} - \left(\rho\phi\right)^{n}}{\Delta t} + R^{n+1} \tag{8}$$

In this equation R represents the remaining terms, n the present time step and $n+1$ the future time (to which the equation is being solved to).

3.4. Final Form of the Governing equations
After discretizing the governing equations, they can be written in a general way:

$$\frac{A_P}{\alpha}\phi_P + \sum_{f=1}^{F} A_L\phi_L = S_\phi + \frac{(1-\alpha)}{\alpha}A_P\phi_P^* \quad (9)$$

here, A_P and A_L are the (implicit) coefficients, which are multiplying by the CV centered values of either ϕ_P or $\phi_E, \phi_W, \phi_N, \phi_S$, respectively. The term S_ϕ is the source term where all the explicit terms and boundary conditions are included, ϕ_P^* is the value of ϕ_P resulting from the last iteration, and α is the under relaxation factor.

4. Incompressible Flow SIMPLE Algorithm

After discretizing the equations, they can to be solved in a sequence. Here, a brief review of SIMPLE algorithm's steps will be presented:

The first step is the initialization one, where the fields of all variables are created. It is followed by the preparation and solution of the momentum equations, which should be done sequentially, and yield the CV centered velocity values. These velocities are used to calculate the velocity values in the CV faces via the Rhie and Chow interpolation [6], that prevents the pressure checker board issue. Next, the pressure correction equation is assembled, solved, and the resultant pressure correction field is used to correct the face velocities, ensuring that continuity is respected in every CV. This quantity is also used to correct the CV centered values of pressure and velocity. After this step, the residuals are checked for convergence: if it wasn't achieved, the cycle is restarted by taking the latest values of every variable, otherwise, the cycle is stopped.

This algorithm is only valid for incompressible flow, though. The proposed extension to this algorithm, so that it is able to solve compressible flows, will be presented in the next section.

5. All-Mach SIMPLE Algorithm

5.1. Algorithm Description
The algorithm is started by initializing all the required fields (u, v, ρ, p, T). These values are then used to prepare and solve the momentum equations, which are solved sequentially for each direction. The following step is the density field recalculation, where the density values are computed by using the ideal gas equation. The CV centered velocity and pressure fields are then used to interpolate the face velocity values with the Rhie and Chow interpolation. The next step is to prepare and solve the pressure correction equation, which yields the corrections both to density and velocity fields, that are necessary to guarantee continuity. The pressure values are also corrected in this step. Afterwards, the energy equation must be prepared and solved (by using the corrected fields), yielding the temperature distribution. Convergence is then assessed: if it wasn't achieved, the cycle starts over, taking the last available fields of all variables. Otherwise, the program is stopped.

5.2. Pressure Correction Equation for Compressible Flow
Many of the steps are similar to the SIMPLE algorithm for incompressible flows. The biggest change occurs in the pressure correction equation:

$$\sum_{F=1}^{f} u_f^* \left(\overline{\frac{1}{RT}}\right)_f p'_f A_f - \rho_f^* \left(\overline{\frac{V_P}{A_P}}\right)_f (\nabla p')_f A_f = -Q_m^* \quad (10)$$

where p' is the pressure correction, which is used to update pressure and density:

$$u'_f = -\left(\overline{\frac{V_P}{A_P}}\right)_f (\nabla p')_f \quad (11)$$

$$\rho'_f = \left(\overline{\frac{1}{RT}}\right)_f p'_f \quad (12)$$

5.3. Stability of the Pressure Correction Equation
The previously described equation is not as stable as the one incompressible flow. Therefore, the following procedure was adopted, to tackle this issue:

- Evaluate the matrix regarding its main diagonal dominance. If it is diagonally dominant, solve the equation as usual, otherwise:
  1. Find the CV where $||A_P|| - \sum ||A_L||$ is largest.
  2. Find the value of α which guarantees $\frac{A_P}{\alpha} = ||A_P|| + \sum ||A_L||$, i.e. the under relaxation required to turn the matrix in a diagonally dominant matrix.
  3. Divide the matrix diagonal by α and add include the source term $\frac{1-\alpha}{\alpha}A_P p'_P$. Here the value of$A_P$ before the under relaxation was considered, otherwise, the source term becomes $(1-\alpha)\,A_P p'_P$.
  4. Solve the system of equations numerically.
  5. Update the value of p'_P, and repeat **4** and **5** until $(p'_P)^{n+1} - (p'_P)^n \leq tol$.

This tolerance was usually set to be 10^{-7}, as that was the tolerance when checking the matrices for the dominance of the main diagonal .

6. Comparison Between the Two Algorithms

The main differences between the original SIMPLE and the all-Mach algorithm are:

- Altered, more general, governing equations with additional terms that are neglected for incompressible flows.

- Recalculating the density values at the cells, and faces.
- A more complicated pressure correction equation with an additional convective term that depends on the local Mach number.
- Solving an energy equation of the flow.
- Possibility of having extra variables, when considering variable μ.

On top of this, the algorithm solves at least 4 variables: u, v, p, T (and additionally ρ, which is explicitly calculated from p and T), instead of the 3 present in the SIMPLE for incompressible flows: u, v, p.

7. Results

7.1. Converging Diverging Nozzle

The first test case solved with the proposed algorithm was the flow in an inviscid isentropic quasi-1D converging diverging nozzle. This test case is often used to validate all-Mach solvers, since in the nozzle, the flow can be subsonic, supersonic and transonic, allowing a demonstration of an all-Mach solver's full capabilities.

The nozzle shape was presented by [24]:

$$S(x) = 0.1\left[1 + 4\left(\frac{x}{l}\right)^2\right] \tag{13}$$

Here $-0.5 \leq x \leq 0.5$ and $l = 1$. Because of the symmetry of the nozzle, the fully-subsonic (or fully-supersonic) flow inside it also becomes symmetric, with the symmetry line being the throat.

If the inlet and outlet pressures are different, one of two different phenomena might happen:

1. The flow can be subsonic at the inlet and until it reaches the throat, become sonic at the throat location and become supersonic beyond it, until the outlet.
2. The same behaviour as previsouly, but a shockwave might appear beyond the throat. In the shock wave, the flow goes from supersonic to subsonic and stays that way until the outlet.

The appearance, and the location of shockwave can be controlled by altering the pressure difference between the inlet and the outlet.

7.2. Results

The obtained results are compared to those available in the literature. It is important to note that all the simulations utilized the upwind scheme for the convective fluxes and the central differencing scheme for the diffusive terms.

7.2.1 Fully Subsonic Flow

The fully subsonic test was run by setting the steady state inlet Mach=0.25, and outlet pressure. The solution ran on a uniform 400 cells mesh.

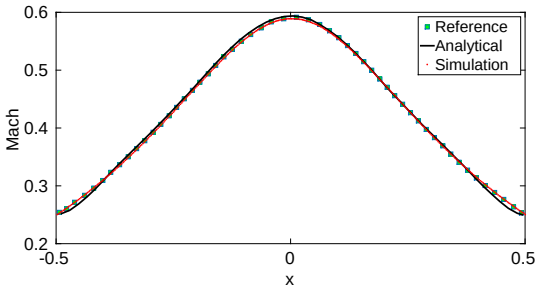


Figure 1: Fully subsonic predicted Mach profile.

Figure 1 shows a very good agreement with the results by [24], which were also obtained using a 100 control volume uniform mesh.

7.2.2 Shockwave Flow

The case that featured a shock wave was made by imposing a ratio between the inlet and outlet pressures: $\frac{p_{in}}{p_{out}} = 1.2$. To enforce this pressure difference at steady state the pressure and temperature were specified at the inlet and the pressure was specified at the outlet. A 500 control volume uniform mesh was utilized to solve this problem, in hopes that the shock intensity and its location were well predicted. The results were compared in figure 2 to those obtained by [24] who ran the simulation on a 400 control volume uniform mesh and compared their own results to the analytical solution.

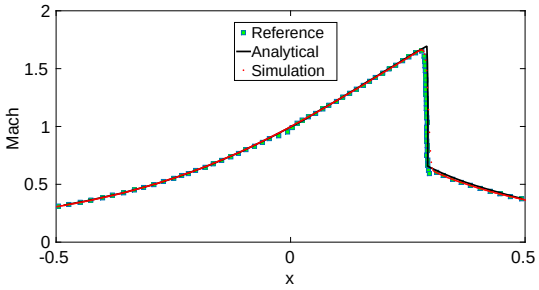


Figure 2: Flow with a shock.

The figure shows that the shock location and its intensity are well captured. It also shows that no oscilations occured, unlike in the result predicted by [24], but, on the other hand, the shock is predicted to be slightly wider. These differences are doubtlessly related to the diffusivity of the upwind scheme.

7.2.3 Subsonic-Supersonic Flow

To simulate the subsonic-supersonic flow across the nozzle, the inlet temperature and pressure were specified and the outlet was left free, using the supersonic outflow boundary condition. 400 control volumes were used and the result compared to the one obtained by [24] in figure3, where the proximity between the solutions is patent, overlapping in most of the domain.

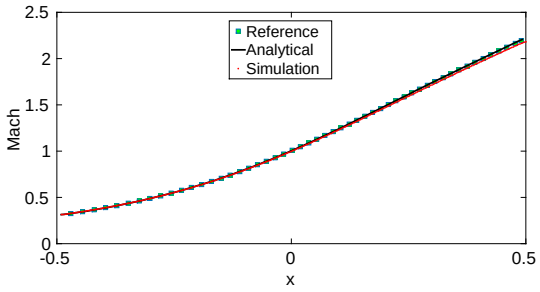


Figure 3: Subsonic-Supersonic flow.

7.2.4 Fully Supersonic

The fully supersonic case was simulated using the supersonic inlet and supersonic outflow boundary conditions and a uniform mesh with 400 cells. The inlet Mach number is reported to be 2.5, albeit the data plotted by [24] is closer to 2.52 at the inlet, hence, this was the value that was used.

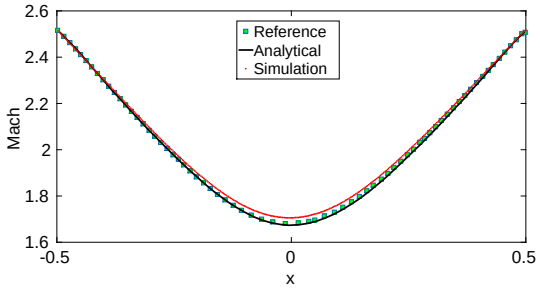


Figure 4: Fully supersonic flow.

The two results are very close in the vicinity of the nozzle's inlet and outlet, where the two Mach profiles overplap. Near the nozzle's throat, however, the results are somewhat different, which might be related to the upwind scheme's large diffusivity.

7.3. Convection Driven Cavity

One of the 2D test cases solved is the convection driven cavity. Once again, the fluid is trapped inside a square cavity, yet, instead of being set in motion by the movement of top wall, the west and east walls' temperatures (T_h and T_c, respectively) are specified. The temperature difference sparks a density change in the fluid, which, due to the force of gravity, makes the fluid rotate in the cavity (whose remaining walls are treated as adiabatic).

Tm	Pr	ϵ	p_o	μ	k
293	0.71	0.034	10^5	$1.79 \cdot 10^{-5}$	0.02540

Table 1: Convection driven cavity initial conditions.

Grid	Ra	ϵ	$\overline{Nu}$	Reference $\overline{Nu}$
100x100	10^3	0.034	1.1330	1.105 - 1.119
100x100	10^4	0.034	2.2683	2.201 - 2.302
40x40	10^5	0.034	4.7242	
60x60	10^5	0.034	4.6492	
80x80	10^5	0.034	4.6207	4.430 - 4.646
100x100	10^5	0.034	4.6064	
80x80	10^6	0.034	9.1784	
100x100	10^6	0.034	9.0985	
120x120	10^6	0.034	9.0526	8.754 - 9.012
140x140	10^6	0.034	9.0234	

Table 2: Constant viscosity convection driven cavity results (UDS).

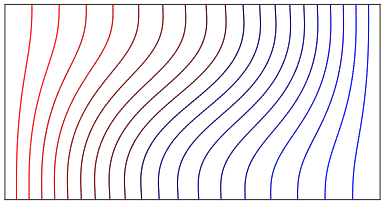
To set up the convection driven cavity correctly and compare the results to the ones available in the bibliography [30, 31, 32, 34, 33], two non-dimensional numbers must be calculated: The Rayleigh number $Ra = Pr\frac{\rho_o^2 g(T_h - T_c)L^3}{T_o \mu_o^2}$ and the non-dimensional temperature difference: $\epsilon = \frac{T_h - T_c}{T_h + T_c}$. The initial values for this test case are listed in table 1. In both cases, by fixating Ra and ϵ, the size of the cavity L can be computed.

To analyse the accuracy of the $\epsilon = 0.034$ simulations, the temperature contour plots are displayed in figures 5, and the adimensionalized velocity profiles in the symmetry sections are displayed in figures 6 and 7. The non-dimensional velocity values are given by: $\frac{v}{\sqrt{g\beta\Delta TL}}$, with $\beta = \frac{\rho R}{p}$. In addition, the average Nusselt number in the hot and cold walls were also calculated (since $\overline{Nu}_w \simeq \overline{Nu}_e$ in all cases, the average value was considered) and documented in tables 2 and 3, using both upwind and linear interpolation . This way, a more general inspection of the flow field can be made through the plotted variables and a quantitative confirmation on the correctness of the solution can be made through the Nusselt number, which is calculated as: $\overline{Nu} = \frac{1}{L}\int_{y=0}^{y=L} Nu(y)\,\mathrm{d}y$, where $Nu(y) = \frac{L}{K_o(T_h - T_c)} k \frac{\partial T}{\partial x}|_w$.

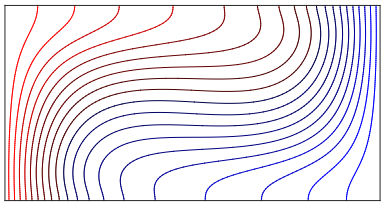
The integral was computed using Simpson's rule and the temperature derivative via either forward or backward finite differences using 2 points.

Grid	Ra	ϵ	$\overline{Nu}$	Iter.
40x40	10^3	0.034	1.1300	357
100x100	10^3	0.034	1.1291	1687
100x100	10^4	0.034	2.2641	965
100x100	10^5	0.034	4.5854	885
100x100	10^6	0.034	9.0082	1266
140x140	10^6	0.034	8.9597	2095

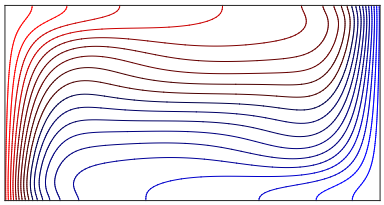
Table 3: Constant viscosity convection driven ity results (LIN).



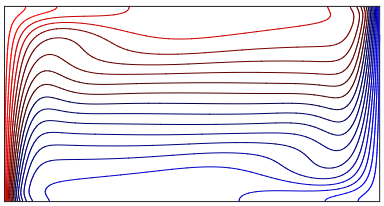
(a) $Ra = 10^3$



(b) $Ra = 10^4$



(c) $Ra = 10^5$



(d) $Ra = 10^6$

Figure 5: Temperature contour plots for several Rayleigh numbers ($\epsilon = 0.034$). 19 contours are shown, between $T = 303K$ and $T = 293K$.

The limiting reference values in table 2 for $\epsilon = 0.034$ have been obtained by [31, 32, 33, 34], using the Boussinesq approximation. Despite the differences in the modelling equations, generally speaking, the results are satisfactory, since the average Nusselt number and velocity profiles are almost always close to, or even inside the interval of reference values.

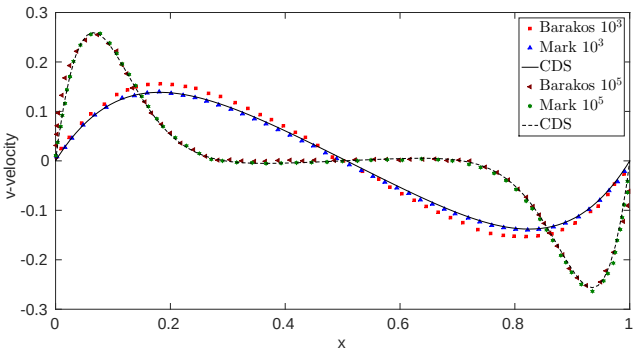


Figure 6: V velocity along horizontal symmetry line for $Ra = 10^3$ and $Ra = 10^5$ ($\epsilon = 0.034$).

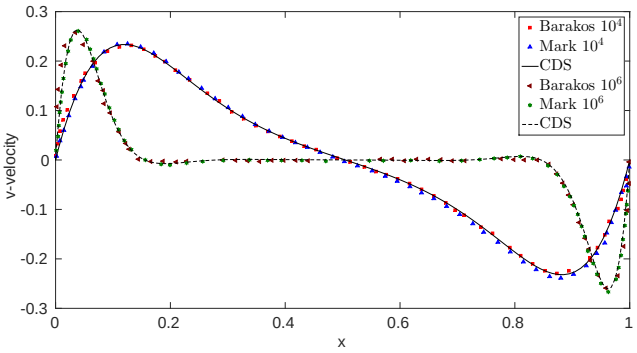


Figure 7: V velocity along horizontal symmetry line for $Ra = 10^4$ and $Ra = 10^6$ ($\epsilon = 0.034$).

The biggest divergence surprisingly happens for $Ra = 10^3$, where the velocity gradients are smaller and farthest from the wall, when compared to the other simulations. Although, the non-dimensionalized velocity profile follows the one obtained by [30] very closely, seemingly underpredicting the velocity values, by comparison to the one obtained by [31]. This in itself might be accountable for the difference in the prediction of the average Nusselt number on the wall, and, even for more refined meshes, the results did not change significantly.

A grid convergence study was made for $Ra = 10^5$ and $Ra = 10^6$. The gradients near the walls for all variables tend to increase with the Rayleigh number, something that, until $Ra = 10^5$ is reached, doesn't seem to alter the nature of the simulation greatly, so, by doing the grid convergence for this value of the Ra, the grid size for all the lower values was set. However, upon reaching $Ra = 10^6$, the

flow becomes much harder to predict. An increase in the number of grid points to obtain a reasonable accuracy was also required.

The temperature contour plots for the several simulations demonstrate that, as the Rayleigh number is increased, convection becomes increasingly important (relative to conduction), especially near the walls, where the velocities are largest. For $Ra = 10^6$, the once diagonally aligned lines at the center obtained for $Ra = 10^3$ have become completely horizontal, meaning that the heat transfer in the central area is negligible, when compared to the one near the walls (further demonstrating the increase in importance of convection).

The discretization schemes can also be compared: the linear interpolation (as expected) always outdid the upwind differencing scheme, to the point that, in the $Ra = 10^3$ case, the results using LIN and a 40x40 grid, proved to be better than the results using the UDS with a 100x100 CV grid, thoroughly justifying the extra iterations needed to converge.

7.4. Inviscid Supersonic Flow Over Forward Facing Step

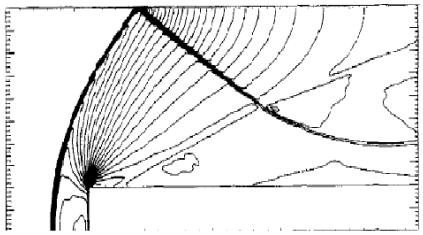
In this test case, an inviscid Mach 3 fluid flows over a forward facing step. This problem must be simulated assuming unsteady flow, in order to be compared to the results in the literature at $t = 4s$, before it reaches steady state at around $t = 12s$.

The fluid's properties are different from the usual: $\gamma = 1.4$, $R = 0.714 J/KgK$, and $c_p = 2.499 J/KgK$, allowing simplified boundary conditions: $T = 1K$, $p = 1Pa$, $u = 3m/s$, which are both the initial values and the ones specified at the inlet. At the outlet, the supersonic outflow boundary condition is used, and, every wall in the domain is a symmetry plane (including the step).

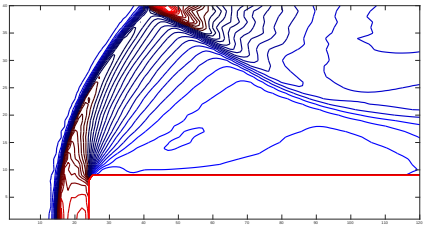
As mentioned by [35] and further detailed by [36] and [37] , the top corner of the step is a paramount region of this test case, for it generates numerical errors, which, if not corrected, will impact the solution. This correction is only valid when the flow near the corner becomes steady state, and the procedure to implement it is fully detailed by [36] , but, loosely, it involves correcting the entropy and enthalpy values of the cells nearest to the corner.

7.4.1 Results

The solution obtained with the Gamma NVD at every $1s$ is now presented, as a way to compare it to the one obtained by [35]. The figures feature 30 contours of density.Due to the amount of time that these simulations take to compute, the grid is less refined than the reference's ($\Delta x = \Delta y = \frac{1}{40}$), and the courant number is 0.8.

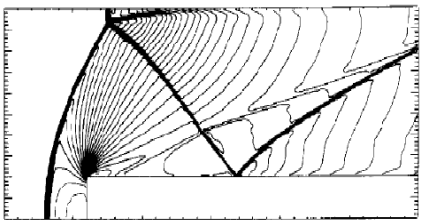


(a) Reference.

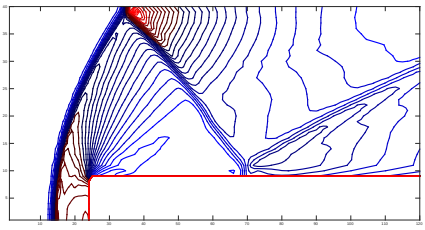


(b) Gamma scheme with correction.

Figure 8: Density contours at $t = 1s$.

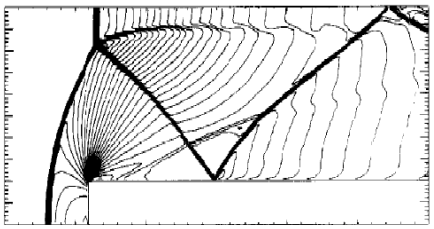


(a) Reference.

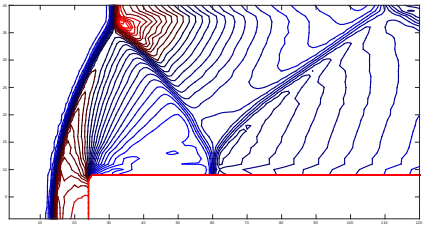


(b) Gamma scheme with correction.

Figure 9: Density contours at $t = 2s$.

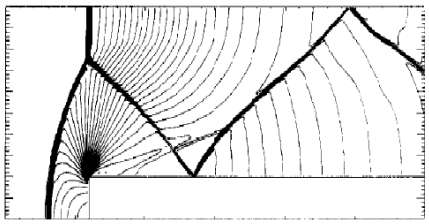


(a) Reference.

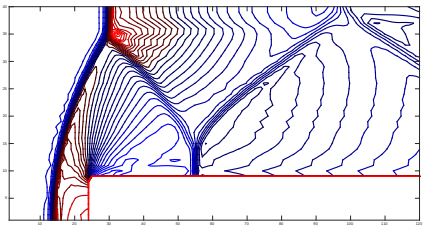


(b) Gamma scheme with correction.

Figure 10: Density contours at $t = 3s$.



(a) Reference.



(b) Gamma scheme with correction.

Figure 11: Density contours at $t = 4s$.

At $t = 1s$, the weak shock generating at the corner of the step was not predicted by the Gamma simulation, yet, other than that the flow seems to follow the reference closely.

A Kelvin-Helmholtz instability starts to appear at $t = 2s$ near the reflection on the top wall, which begins to separate itself from said wall. However, the Gamma scheme simulation does not recognize this structure at this time. However, at time $t = 3s$, this instability can be clearly seen, and the flow seems to follow the reference, in a general way. These are the first visible effects of the numerically induced boundary layer. At the same time, the gamma scheme also features a increase in the height of that shock reflection.

At the end of the simulation ($t = 4s$), the Kelvin-Helmholtz instability has subsided in the reference solution, unlike in the Gamma solution, where it can still be traced. On the other hand, the shock induced at the corner still has not been captured by this scheme, and the shock reflection on the bottom is too vertically stretched. However, most of these issues might be related to the less refined grid.

8. Conclusions

The SIMPLE algorithm and an extension to allow the computation of all-Mach flows have been implemented and tested in a 1D and several 2D cases with uniform grids. This was accomplished through the addition of several steps to the algorithm, as well as the expansion of the pressure correction equation, which takes the density changes into account when they are relevant. The stabilization issues of this equation were fixed by including an iterative procedure that involves a self-calculating relaxation factor.

The test cases allowed for a gradual assemblage of the algorithm, i.e. by starting with steady incompressible flow and gradually building up the algorithm to encompass viscous supersonic flow, and unsteady flows.

The implementation of the upwind, linear, and NVD Gamma convective schemes, as well as their applicability to the test cases that were solved have been documented. The upwind scheme provides stability to a level beyond any of the other schemes, yet, the amount of detail lost when applying sometimes cannot be tolerated. The Gamma NVD scheme implemented allows a much better resolution, yet, especially at higher Mach numbers, it sometimes required the added stability of the unsteady terms to converge, for these test cases.

All in all, the implementation of the algorithm can be considered to have been successful, as the results for both compressible, and incompressible flow simulations were close to the references'. This

meaning that the algorithm can to solve flows at all Mach numbers.

References

[1] S. V. Patankar, D. B. Spalding, *A Calculation Procedure for Heat,Mass and Momentum Transfer in Three-Dimensional Parabolic Flows* Int. J. Heat Mass Transfer Vol. 15, pp. 1787-1806 (1972)

[2] S. V. Patankar, *Numerical Heat Transfer* Hemisphere, Washington, D.C. (1980)

[3] J.P. Van Doormal, G.D. Raithby *Enhancements of the SIMPLE Method for Predicting Incompressible Fluid Flows* Numerical Heat Transfer, vol.7, pp.147-163 (1984)

[4] R. I. Issa *Solution of the Implicitly Discretised Fluid Flow Equations by Operator-Splitting* Journal of Computational Physics 62, 40-65 (1985)

[5] D. S. Jang , R. Jetli, S. Acharya *Comparison of the PISO, SIMPLER, and SIMPLEC Algorithms for the Treatment of the Pressure-Velocity Coupling in Steady Flow Problems* Numerical Heat Transfer: An International Journal of Computation and Methodology, 10:3, 209-228 (1986)

[6] C. M. Rhie , W. L. Chow, *A Numerical Study of the Turbulent Flow past an Isolated Airfoil with Trailing Edge Separation*, AIAA J., vol. 21, pp. 15251532, 1983.

[7] R. I. Issa, F. C. Lockwood *On the Prediction of Two-Dimensional Supersonic Viscous Interactions Near Walls*, AIAA JOURNAL VOL. 15, NO. 2 (1977)

[8] J.P.Van Doormal, G.D.Raithby, B. H. McDonald *The Segregated Approach to Predicting Viscous Compressible Fluid Flows*, Journal of Turbomachinery, vol.109, pp.268-277 (1987)

[9] K. C. Karki, S. V. Patankar *A Pressure Based Calculation Procedure For Viscous Flows at All Speeds In Arbitrary Configurations*, AIAA-88-0058 26th Aerospace Sciences Meeting January 11-14, Reno, Nevada (1988)

[10] W. Shyy, Mark E. Braaten *Adaptive Grid Computation For Inviscid Compressible Flows using a Pressure Correction Method*, 1st National Fluid Dynamics Conference Cincinnati, OH,U.S.A. (1988)

[11] M. H. Kobayashi, J. C. F. Pereira *Predictions of Compressible Viscous Flows at all Mach Number Using Pressure Correction, Collocated Primitive Variables and Non-Orthogonal Meshes*, AIAA-92-0426 30th Aerospace Sciences Meeting Exhibit, January 6-9, Reno, Nevada (1992)

[12] W. Shyy, M. Ghent,C. Sun *Pressure-Based Multigrid Algorithm for Flow at All Speeds*, AIAA Journal, Vol. 30, No. 11, November 1992

[13] C., M. Rhie *Pressure-Based Navier-Stokes Solver Using the Multigrid Method*, AIAA Journal 1017, Vol. 27, NO. 8, August 1989

[14] C.H. Marchi, C. R. Maliska, *A Non-orthogonal Finite-Volume Methods for the Solution of All Speed Flows Using Co-Located Variables*, Numerical Heat Transfer, Part B, vol. 26, pp. 293-311, 1994

[15] I. Demirdzic , Z. Lilek , M. Peric, *A Collocated Finite Volume Method For Predicting Flows At All Speeds*, International Journal for Numerical Methods in Fluids, vol.16, 1029-1050(1993).

[16] F. S. Lien, M. A. Leschziner, *A Pressure-Velocity Solution Strategy for Compressible Flow and Its Application to Shock/Boundary-Layer Interaction Using Second-Moment Turbulence Closure*, Journal of Fluids Engineering, vol. 115, pp. 717-725, 1993.

[17] E. S. Politis, K. C. Giannakoglou, *A Pressure-Based Algorithm for High-Speed Turbomachinery Flows*, International Journal For Numerical Methods in Fluids, vol. 25, 63-80 (1997)

[18] M. Darwish , F. Moukalled, *A High-Resolution Pressure-Based Algorithm for Fluid Flow at All Speeds* Journal of Computational Physics 168, 101133 (2001)

[19] I. Senocak and W. Shyy, *A Pressure-Based Method For Turbulent Cavitating Flow Computations* Journal of Computational Physics 176, 363-383 (2002)

[20] M. Darwish , I. Sraj, and F. Moukalled, *A Coupled Incompressible Flow Solver on Structured Grids*, Numerical Heat Transfer, Part B: Fundamentals, 52:4, 353-371 (2007)

[21] M. Darwish , F. Moukalled, *A Fully Coupled Navier-Stokes Solver For Fluid Flow At All Speeds*, Taylor & Francis Group, LLC, 2013.

[22] M. Darwish , F. Moukalled, *An OpenFOAM Pressure-Based Coupled CFD Solver for Turbulent and Compressible Flows in Turbomachinery Applications*, Taylor & Francis Group, LLC, 2013.

[23] Z. J. Chen, A. J. Przekwas, *A coupled pressure-based computational method for incompressible/compressible flows*, Elsevier, 2010.

[24] C. Xiao , F. Denner , B. G. M. van Wachem, *Fully-Coupled Pressure-Based Finite-Volume Framework for the Simulation of Fluid Flows at All Speeds in Complex Geometries*, Journal of Computational Physics 346 (2017) 91130.

[25] Joel H. Ferziger , Milovan Perić, *Computational Methods for Fluid Dynamics*, Springer, 3rd edition, 2002.

[26] L. Mangani, *Development and Validation of an Object Oriented CFD Solver for Heat Transfer and Combustion Modeling in Turbomachinery Applications*, Univerist degli Studi di Firenze, 2008.

[27] H. K. Versteeg, W. Malalasekera *An Introduction to Computational Fluid Dynamics*, Pearson Education Limited 1995, 2nd edition, 2007

[28] H. Jasak, *Error Analysis and Estimation for the Finite Volume Method with Applications to Fluid Flows* PHD thesis, Imperial College, University of London, 1996

[29] John D. Anderson, *Computational Fluid Dynamics*, McGraw-Hill Education

[30] A. Mark, E. Svenning, F. Edelvik, *An immersed boundary method for simulation of flow with heat transfer*, International Journal of Heat and Mass Transfer 56 (2013) 424435

[31] G. Barakos, E. Mitsoulis, D. Assimacopoulos, *Natural Convection Flow in a Square Cavity Revisited: Laminar and Turbulent Models with Wall Functions*, International Journal for Numerical Methods in Fluids, vol. 18, 695-719 (1994)

[32] N. C. Markatos and K. A. Pericleous, *Laminar And Turbulent Natural Convection In An Enclosed Cavity*, Int. Journal of Heat and Mass Transfer, vol. 27. No. 5, pp. 755 772 (1984)

[33] G. De Vahl Davis *Natural Convection of Air In a Square Cavity, A Bench Mark Numerical Solution* International Journal For Numerical Methods In Fluids, VOL. 3, 249-264 (1983)

[34] T. Fusegi, J. M. Hyun, K. Kuwaharas and B. Farouk, *A Numerical Study of Three-Dimensional Natural Convection in a Differentially Heated Cubical Enclosure*, Int. Journal of Heat and Mass Transfer, vol. 34, No. 6. PP. 1543-1557 (1991)

[35] P.Woodward, P.Colella, *The Numerical Simulation of Two-Dimensional Fluid Flow with Strong Shocks* Journal of Computational Physics, vol. 54, p. 115-173 (1984)

[36] R. Donat, A. Marquina, *Capturing Shock Reflections: An Improved Flux Formula* Journal of computational physics, vol. 125, pp.4258 (1996)

[37] R. Sanders, A. Weiser, *High Resolution Staggered Mesh Approach for Nonlinear Hyperbolic Systems of Conservation Laws* Journal of computational physics,vol. 101, pp.314-329 (1992)