

INSTITUTO SUPERIOR TÉCNICO

COMPUTATIONAL FLUID MECHANICS

HW2

Authors :

Diogo Carvalho N°83677

José Santos N°83704

Miguel Brito N°83711

Gonçalo Oliveira N°84256

Philip Widmaier N°95467

Professors :

Prof. José Carlos Fernandes Pereira

Prof. José Manuel da Silva Chaves Ribeiro Pereira

November 28, 2019

Contents

1	Convection Schemes	1
1.1	Von Neumann stability analysis	1
1.2	Spectral Analysis	1
1.3	Explicit first order time scheme	1
1.3.1	Euler with upwind interpolation (UDS)	1
1.3.2	Euler with quadratic upwind interpolation (QUICK)	3
1.4	Explicit forth order time scheme	3
1.4.1	Runge-Kutta with upwind interpolation (UDS)	3
1.4.2	Runge-Kutta with central linear interpolation (CDS)	4
1.5	Implicit first order time scheme	5
1.5.1	Euler with upwind interpolation (UDS)	5
1.5.2	Euler with quadratic upwind interpolation (QUICK)	6
1.6	Flux Limiter	7
1.6.1	Min mod	7
2	Heat Transfer in Unstructured Grids	7
2.1	Sources of error	7
2.2	Mesh1A: Trimmed	9
2.3	Mesh2D: Triangular and prism layer	10
2.4	Mesh3G: Polyhedral	10
2.5	Mesh4A: Error decay with respect to mesh base size	11
2.6	Mesh3G: Secondary gradient and Neumann boundary condition influences	11
2.7	Strategy implemented to get grid with lower error	13

1 Convection Schemes

1.1 Von Neumann stability analysis

The Von Neumann method is the most relevant tool for stability analysis. This method, however, can only be applied in problems without boundary influence. One way to circumvent it is to consider periodic boundaries such as in our simulation. This method begins by expanding the numerical solution into a Fourier series. A more detailed explanation is presented in [2]. Here we present the most important steps using as frame the Euler Explicit time scheme with upwind discretization in space. We start with the 1-D advection equation. Integrating in time and space and after the appropriate simplifications not treated here because they fall out of the scope of this report we have the final result:

$$\frac{d\phi}{dt} + a \frac{d\phi}{dx} = 0 \Rightarrow \phi_j^{n+1} - \phi_j^n + a \frac{\phi_j^n - \phi_{j-1}^n}{\Delta x} \Delta t = 0 \quad (1)$$

Replacing ϕ_{j+l}^{n+k} by $a^{n+k} e^{i\theta(j+l)}$, and knowing that $\frac{a\Delta t}{\Delta x}$ is equal to C , which is the Courant number. Then dividing by $e^{i\theta j}$ and by a^n and using Euler's trigonometric identity:

$$a^{n+1} e^{i\theta j} = a^n e^{i\theta j} - C(a^n e^{i\theta j} - a^n e^{i\theta(j-1)}) \Rightarrow G = \frac{a^{n+1}}{a^n} = 1 - C(1 - e^{-i\theta}) = 1 - C + C \cos \theta - Ci \sin \theta \quad (2)$$

According to the Von Neumann stability condition, the amplitude of any harmonic may not grow indefinitely in time. In order for that to happen, the amplification factor $|G|$ has to be smaller than 1. However, since we have an amplification factor with an imaginary part, we will use the expression $|G|^2 \leq 1$ which is similar to $|G| \leq 1$.

$$|G|^2 = 1 + C(-2 + 2 \cos \theta) + C^2(2 - 2 \cos \theta) \Rightarrow \begin{cases} -1 \leq 1 + C(-2 + 2 \cos \theta) + C^2(2 - 2 \cos \theta) \\ 1 + C(-2 + 2 \cos \theta) + C^2(2 - 2 \cos \theta) \leq 1 \end{cases} \quad (3)$$

While the first expression is always true, the second is only true for $0 < C \leq 1$, so this is our stability condition, which therefore means this method is conditionally stable. Besides the analytical way to achieve the stability condition, we can also plot G for different values of C in a polar diagram. If our plot leaves the circle of radius 1 we conclude the scheme is unstable for that Courant number. This is done with Matlab for the subsequent schemes analyzed in the report.

1.2 Spectral Analysis

The analytic solution to our equation can also be represented by a Fourier decomposition, just like we have done it on our numerical solution. For the convection equation, following the procedure in [2], we have an expression for the exact amplification factor, which is: $\tilde{G} = e^{-i\tilde{\omega}\Delta t}$ (for $\tilde{\omega} = a \frac{\theta}{\Delta x}$). This allows us to obtain more information on the error from our numerical solution. The error is decomposed into two types, diffusion errors and dispersion errors. Diffusion errors are evidenced by the difference in amplitude, while dispersion errors by the difference in phase between the exact solution and the numerical solution. These errors are calculated using the following expressions for convection dominated problems, such as ours:

$$\epsilon_D = \frac{|G|}{|\tilde{G}|} = |G| \quad (4) \quad \epsilon_\phi = \frac{\text{Arg}(G)}{\text{Arg}(\tilde{G})} = \frac{\text{Arg}(G)}{-C\theta} \quad (5)$$

1.3 Explicit first order time scheme

1.3.1 Euler with upwind interpolation (UDS)

In the Von Neumann chapter overviewing this methodology, we covered this scheme in our example. Rewriting here the expression obtained, for simplicity, in combination with the respective polar diagram and diffusive and dispersive error graphs we have:

$$G = 1 - C + C \cos \theta - Ci \sin \theta \quad (6)$$

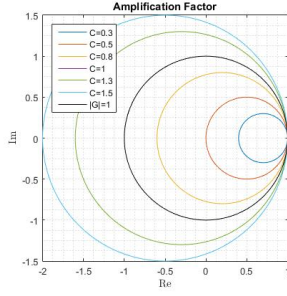


Figure 1: Amplification Factor

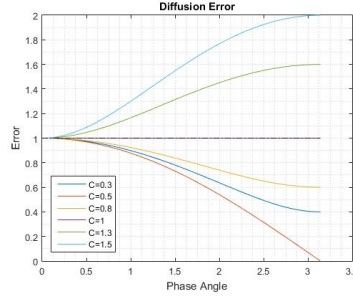


Figure 2: Diffusion Error

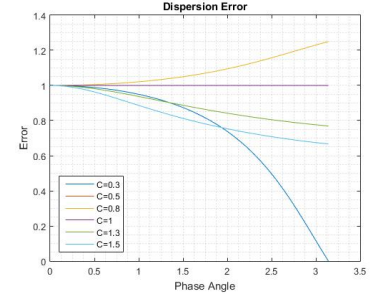
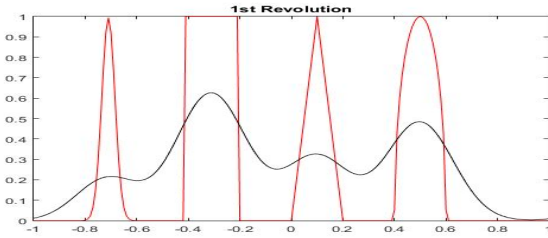
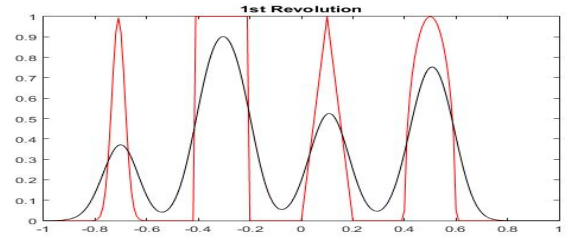
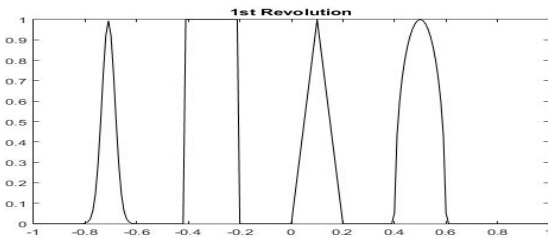
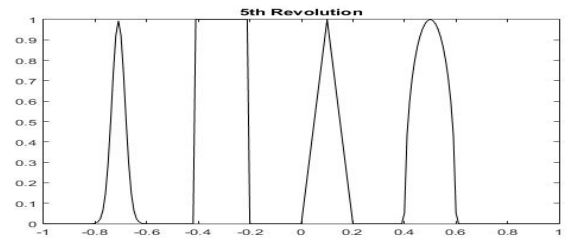


Figure 3: Dispersion Error

The stability condition was also derived in our example. As demonstrated, this scheme is conditionally stable. For $C \leq 1$ no harmonic grows in time. Looking at the polar diagram we reach the same conclusion. Furthermore, analyzing the diffusion error graph, we see that all wavelengths simultaneously (i.e. with the same Courant number) become unstable outside the stability region. Regarding the dispersion error, one can observe that for $C < 0.5$ and $C > 1$ the solution will lag compared to the correct solution. On the other hand, for $0.5 < C < 1$ our solution will lead the correct solution. Plotting the solution after one revolution for two values of C , each in one of these intervals, we reach this same conclusion (Figure 4 and Figure 5).


Figure 4: Solution after one revolution ($C=0.3$)

Figure 5: Solution after one revolution ($C=0.8$)

Figure 6: Solution after one revolution ($C=1$)

Figure 7: Solution after five revolutions ($C=1$)

Finally, when using $C=1$ this scheme displays neither a diffusion nor a dispersion error, meaning the solution calculated with this scheme will be exactly the same as the initial value for integer numbers of revolutions. In this example our convection velocity is small ($a = 1m/s$), however for large convection speeds keeping our Courant number small translates into a small time-step, worsened when working with a fine mesh. For example, simulating a supersonic flow with a convection speed of $a = 350m/s$ and an equally spaced mesh as ours, for $C=1$ our time-step restriction is $\Delta t < 3 \times 10^{-5}s$. This previous case underlines a major shortcoming of a first-order explicit method for time-dependent problems.

1.3.2 Euler with quadratic upwind interpolation (QUICK)

Following the Von Neumann methodology exposed and plotting the respective polar diagram, diffusion and dispersive error graphs for a number of values of C we get:

$$G = \left[1 - \frac{3}{8}C + \frac{1}{2}C \cos(\theta) - \frac{1}{8}C \cos(2\theta) \right] + \left[-\frac{5}{4}C \sin(\theta) + \frac{1}{8}C \sin(2\theta) \right] i \quad (7)$$

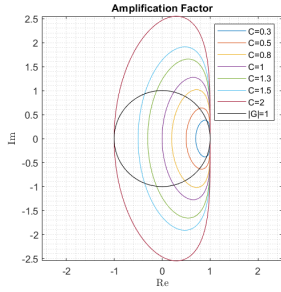


Figure 8: Amplification Factor

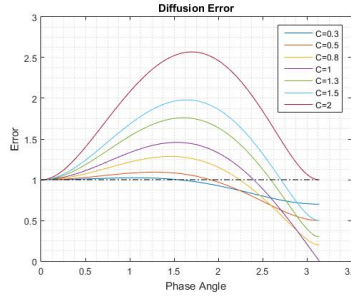


Figure 9: Diffusion Error

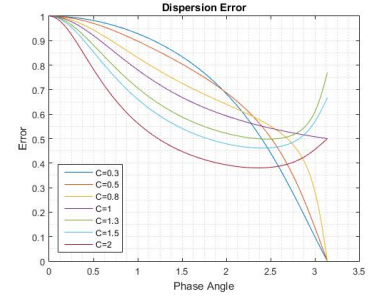


Figure 10: Dispersion Error

Examining the polar diagram near $(1, 0)$, regardless of our choice for C , all lines leave the circle of radius 1. Hence, this scheme is unconditionally unstable. Furthermore, the diffusion error graph shows that wavelengths larger than $4\Delta x$ cause the unstable behavior, for a small courant number. As C increases, $2\Delta x$ -waves become unstable last. Instability is present for all wave numbers when C is larger than 2. Lastly, the dispersion graph shows that all waves lag in comparison to the correct solution. Contrary to what we might have intuitively thought, combining a third-order spatial scheme with our explicit first-order time scheme, i.e., increasing the order of our spatial discretization scheme, does not lead to an overall more stable scheme.

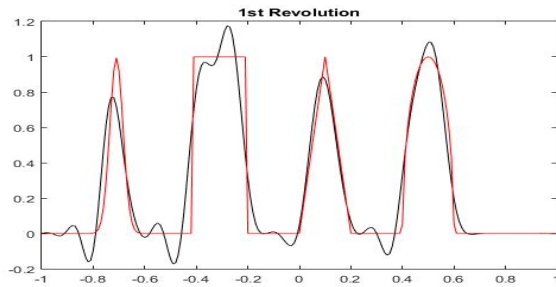


Figure 11: Solution after one revolution ($C=0.01$)

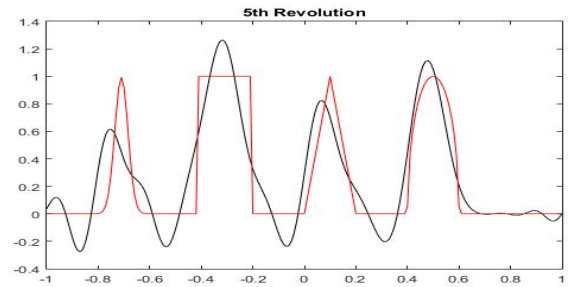


Figure 12: Solution after five revolutions ($C=0.01$)

Being unconditionally unstable, in order to present a meaningful simulation our choice of C had to be extremely small with $C = 0.01$ (for the present case $\Delta t = 1 \times 10^{-4} s$). In addition to the remark made at the end of the previous section we can conclude that regardless of the spatial scheme's order, first-order time schemes are not the most suitable discretization options in unsteady problems.

1.4 Explicit forth order time scheme

1.4.1 Runge-Kutta with upwind interpolation (UDS)

With a first-order explicit time scheme, increasing the order of the spatial discretization scheme does not produce the expected results as demonstrated in the previous section. The next logical step is to increase the order of our time scheme instead. In our simulation we resort to forth-order RK method. This method involves the calculation of 4 stages, each stage dependent on the former. In order to study the stability and spectral behavior of the scheme, following the procedure proposed before is impractical. For this reason, as explained in [1], for the advection equation

it is possible to introduce linear case RK methods. These methods behave completely similar to the classical ones when applied to the convection equation. Both upwind and central difference scheme stability expressions are achieved following the procedure in [1]. The expression for this scheme is:

$$G = 1 + C(e^{-i\theta} - 1) + \frac{C^2}{2}(e^{-i\theta} - 1)^2 + \frac{C^3}{6}(e^{-i\theta} - 1)^3 + \frac{C^4}{24}(e^{-i\theta} - 1)^4 \quad (8)$$

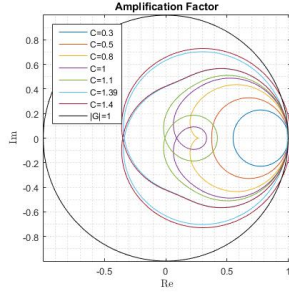


Figure 13: Amplification Factor

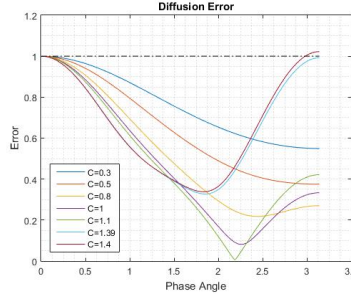


Figure 14: Diffusion Error

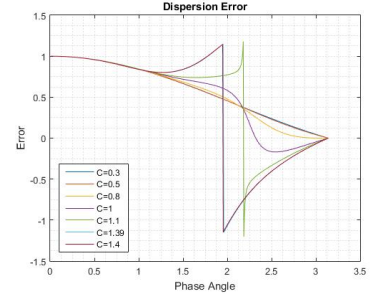
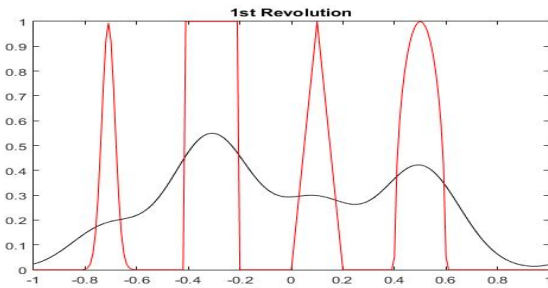
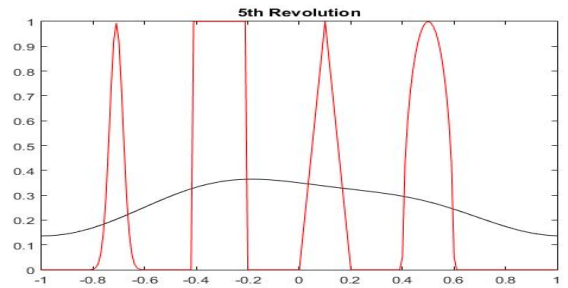


Figure 15: Dispersion Error

As presented in [1], the stability condition for RK4 with upwind first-order is $C < 1.3926...$. This can be verified both in the polar diagram as well as in the diffusion error graph. Furthermore, by inspection of this graph we can also note that $2\Delta x$ -waves first become unstable. This translates into an abrupt variation of the error between two contiguous mesh points. Looking at the dispersion error graph, we see that at $C = 1.1$ a discontinuity appears. As C grows, the wavelength responsible for this will approach the $4\Delta x$ value. This discontinuity means $\arg(G)$ becomes positive. Because we are working with the real values, the real part of $e^{-i\theta}$, which is $\cos(-\theta)$, does not change sign and the impact in the solution is meaningless. Moreover, the wavelengths close to this value are most diffused, thus, its contribution in the final solution is negligible. For these reasons the final solution is lagged when put against the correct one. Comparing it with Euler explicit we can highlight RK4 has a wider range of values inside the stability condition.


Figure 16: Solution after one revolution ($C=0.8$)

Figure 17: Solution after five revolutions ($C=0.8$)

1.4.2 Runge-Kutta with central linear interpolation (CDS)

One important takeaway from the Euler explicit method was that increasing the order of our spatial discretization scheme does not necessarily lead to a more stable overall scheme. Nonetheless, this is not a general rule meaning the same behavior might not occur with a fourth-order time scheme. Using once more the method in [1] to obtain the expression of G and plotting the respective polar diagram and diffusive/dispersive error graphs, one gets:

$$G = \left[1 - \frac{C^2}{2} \sin^2(\theta) \right] + \left[-C \sin(\theta) + \frac{C^3}{6} \sin^3(\theta) \right] i \quad (9)$$

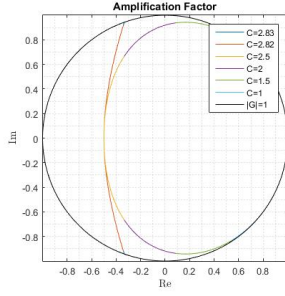


Figure 18: Amplification Factor

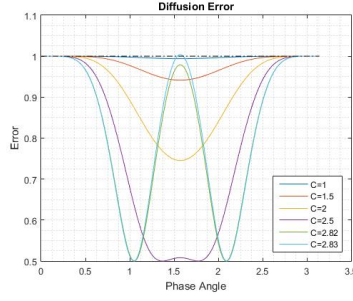


Figure 19: Diffusion Error

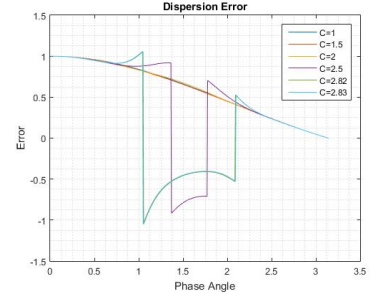
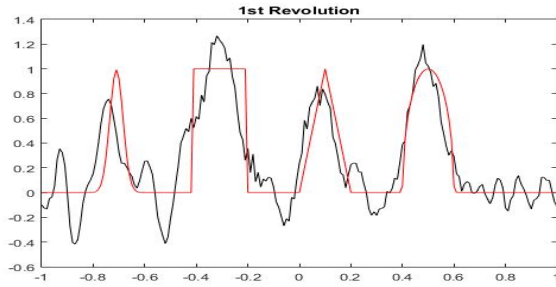
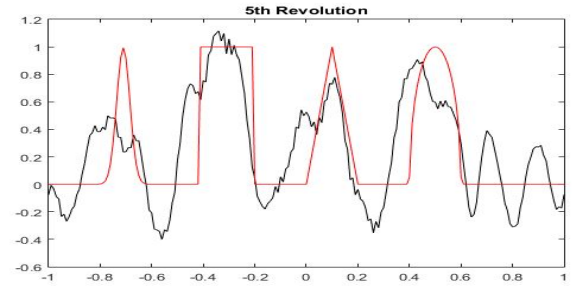


Figure 20: Dispersion Error

The stability condition for this scheme is presented both in [1] and in [2] as $C < 2.8284\dots$. The range of stable values of C is larger. This is also verified in the polar diagram. Looking at the diffusive error graph, we see that $4\Delta x$ -waves ($\theta = \frac{\pi}{2}$) become unstable first. In addition to this, Courant numbers up to 1 also display a much lower diffusion when compared to upwind RK4 scheme. In conclusion, in contrast to Euler explicit, increasing the spatial scheme order with RK4 leads to an overall better scheme.


Figure 21: Solution after one revolution ($C=1$)

Figure 22: Solution after five revolutions ($C=1$)

1.5 Implicit first order time scheme

1.5.1 Euler with upwind interpolation (UDS)

Returning again to the Von Neumann methodology we get the expression for the amplification factor and plots for the polar diagram and diffusive/dispersive error graphs as:

$$G = \frac{1}{1 + C - C \cos \theta + Ci \sin \theta} \quad (10)$$

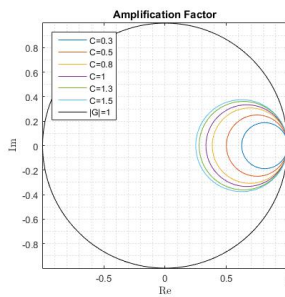


Figure 23: Amplification Factor

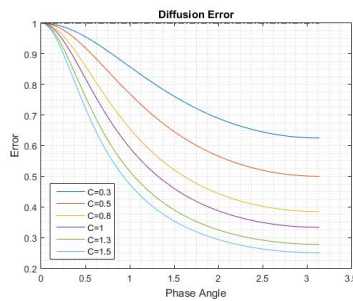


Figure 24: Diffusion Error

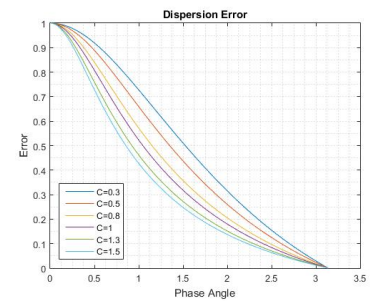
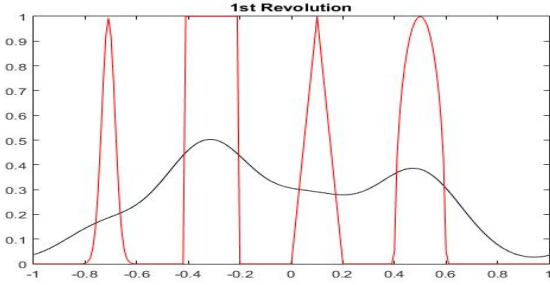
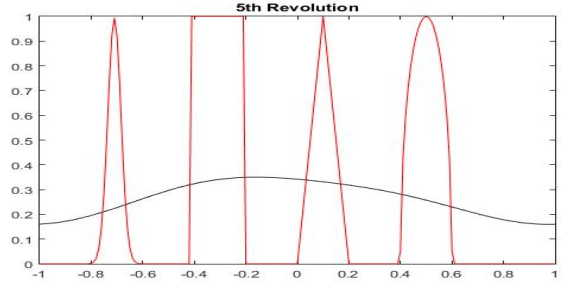


Figure 25: Dispersion Error

Observing the polar diagram together with the diffusive error graph we conclude the scheme is unconditionally stable, in other words it is stable for all values of C . In the diffusion error graph we see that high-frequency waves decay more rapidly than low-frequency waves, regardless of our Courant number. Dispersion error graph also presents a lagging in the solution which is transversal to all wavelengths.


Figure 26: Solution after one revolution ($C=0.3$)

Figure 27: Solution after five revolutions ($C=0.3$)

Because the higher the Courant number the greater is the diffusion, we chose a low value ($C=0.3$) in our simulation. Again this low value of C presents no great impasse because we are dealing with a small convection speed.

1.5.2 Euler with quadratic upwind interpolation (QUICK)

Following once again the Von Neumann methodology we obtain the following expression and plots of the polar diagram and diffusion/dispersion error graphs for G :

$$G = \frac{1}{\left[1 + \frac{3}{8}C - \frac{1}{2}C \cos(\theta) + \frac{1}{8}C \cos(2\theta)\right] + \left[\frac{5}{4}C \sin(\theta) - \frac{1}{8}C \sin(2\theta)\right]i} \quad (11)$$

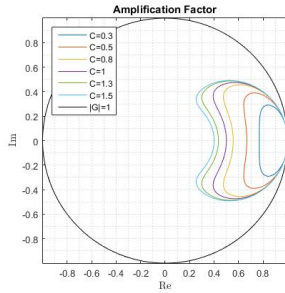


Figure 28: Amplification Factor

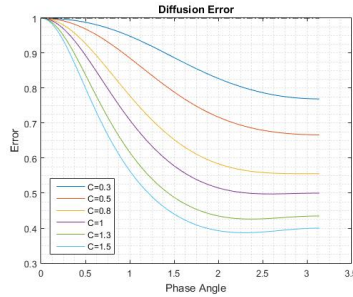


Figure 29: Diffusion Error

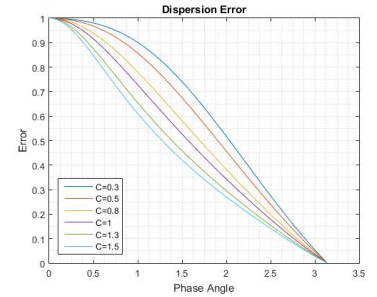


Figure 30: Dispersion Error

Examining both the polar diagram and the diffusion error graph, we quickly ascertain this scheme is, like the previous one, unconditionally stable. In much the same manner, high number waves also disappear faster than low number waves. Not only this, but there is also a lagging in the calculated solution. The main difference between these schemes is the rate of diffusion in low frequency waves, for a given C . Because these are the less diffused waves they will have the greater impact in the sum of waves composing the solution. For this reason, the final solution also displays less diffusion using the QUICK scheme. We saw that, contrary to what one would have initially thought as logical, in explicit first-order scheme increasing the order of our spatial scheme led to worse results. With this implicit time scheme, however, we achieved better results with a higher spatial discretization scheme. In conclusion, an implicit scheme has the advantage compared to an explicit scheme of not having any time-step restriction in regards to stability. Nevertheless, it requires more CPU usage than an explicit scheme of the same order. This additional burden comes from having to solve a system of linear equations, at each iteration, the same size as our number of grid points.

1.6 Flux Limiter

1.6.1 Min mod

We will now be applying a flux limiter to the Explicit Euler discretization in time, to help stabilize the solution and get a oscillation-free solution, of all TVD schemes, *MinMod* was chosen. In order to compare the influence of the flux limiter, we ran the solution for the same Courant number, 0.3, as the one used previously in the Explicit Euler Upwind scheme. The importance of the grid size was also investigated, therefore we ran the solution for the grid given to us, with 200 cells ($H=0.01$), and with 2000 cells ($H=0.001$).

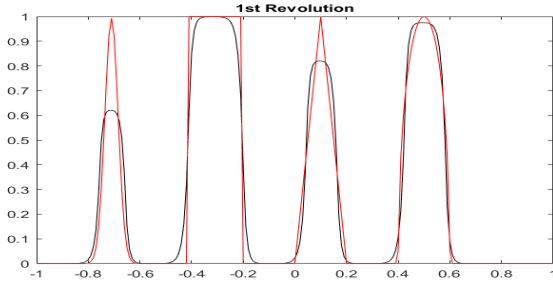


Figure 31: Solution after one revolution ($C=0.3, H=0.01$)

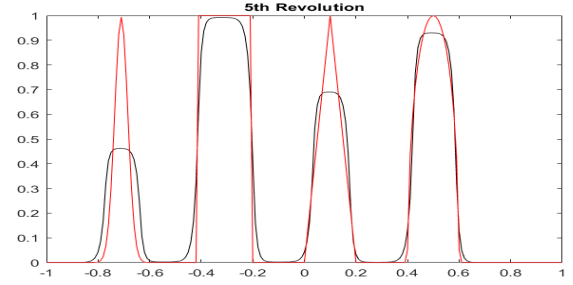


Figure 32: Solution after five revolutions ($C=0.3, H=0.01$)

Comparing the solutions from the flux limiter and from the upwind explicit scheme after 1 revolution, it is clearly seen that the diffusion error was mitigated, with our solution being very close to the exact one. This was expected, because TVD schemes are supposed to have less diffusion error than their upwind explicit counterparts.

Analysing the effect of the changes in grid size, through comparing the results after 1 and 5 revolutions, we can see that our solution with a smaller step in space has a significantly better solution, very close to the exact one, with no diffusion error and a small dispersion error, with the solution leading the exact. The improvement with a smaller grid size although significant is also costly in terms of computational resources, because for the same courant number, a smaller grid leads to a smaller time-step, ergo a larger number of iterations.

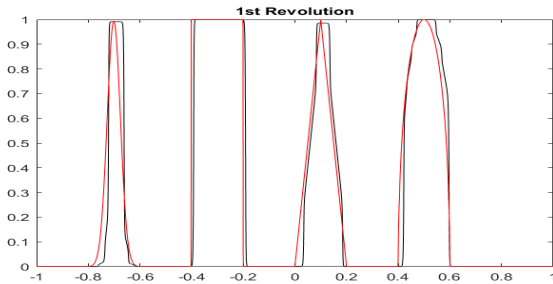


Figure 33: Solution after one revolution ($C=0.3, H=0.001$)

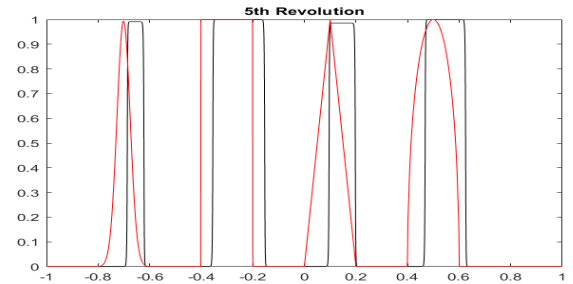


Figure 34: Solution after five revolutions ($C=0.3, H=0.001$)

2 Heat Transfer in Unstructured Grids

2.1 Sources of error

The use of non-uniform grid in CFD simulations is quite popular but it should be used with caution since it can bring loss of accuracy, particularly with the schemes generally used, which are second order accuracy for structured orthogonal grids. However, with the finite volume discretization of non orthogonal meshes, the accuracy is no longer

of second order. One can notice that for a non orthogonal mesh the line that connects the centroids of two adjacent cells is not aligned with the normal of the shared edge. This phenomena, referred to as mesh skewness, quantifies the distortion of a cell compared to a normalized cell. Another distortion indicator and reason for the emergence of errors is the so called warp angle, the angle between the normal vector of a face and the centroid-connecting line (see figure 35).

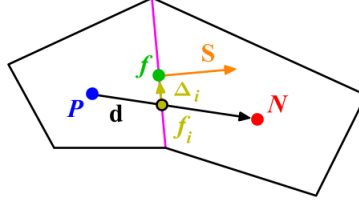


Figure 35: Skewness (Δi) and warp angle (between \vec{S} and \overrightarrow{PN}). [5]

Thus, a higher cell distortion will lead to higher calculus errors. There are ways of correcting this issue, e.g. implementing secondary gradients. This particular method is common in simulation software and will be discussed in section 2.6. In general, correction approaches for unstructured meshes approximate the surface vector \vec{S} in different ways. That is necessary because the normal gradient cannot be written as a function of only Φ_P and Φ_N (like in structured grids), since it has a component perpendicular to \overrightarrow{PN} . Hence, \vec{S} (per definition normal to the cell face) is expressed as a sum of \vec{E} (pointing in \overrightarrow{PN} direction) and \vec{T} (parallel to the cell face):

$$\vec{S} = \vec{E} + \vec{T} \quad (12)$$

Now the linearization of the flux in non-orthogonal grids can be expressed analytically:

$$\begin{aligned} \nabla \Phi \cdot \vec{S} &= \overbrace{\nabla \Phi \cdot \vec{E}}^{\text{orthogonal to cell face}} + \overbrace{\nabla \Phi \cdot \vec{T}}^{\text{non-orthogonal to cell face}} \\ &= \vec{E} \cdot \frac{\Phi_N - \Phi_P}{d_{PN}} + \nabla \Phi \cdot \vec{T} \end{aligned} \quad (13)$$

For computational purposes, the analytical decomposition (equation 12) is approached by expressing \vec{E} as a function of \vec{S} , e.g. $\vec{E} = S \frac{\overrightarrow{PN}}{\|\overrightarrow{PN}\|}$ ¹. In doing so, the different approaches each emphasize either the orthogonal or non-orthogonal term of equation 13 which leads to different error distributions. [3]

As previously said, the scheme used by the software is a second order central difference scheme, which implies that the truncation error will be proportional to the fourth derivative of the field function. Thus, it is important to know how the fourth derivative behaves in the domain in order to know where higher error values are expected and vice versa. Figure 37 shows a contour of the fourth derivative of the field function. One can already suppose that, when truncation errors have greater influence than other sources of error, the higher values will appear near the corners (0,0) and (0.5,0.5). On the other hand, the diagonal connecting the corners (0,0.5) and (0.5,0) should have the smaller values for the error.

¹Orthogonal Correction Approach: This approach keeps the contribution of the term involving Φ_P and Φ_N the same as on an orthogonal mesh irrespective of the degree of grid non-orthogonality. [3]

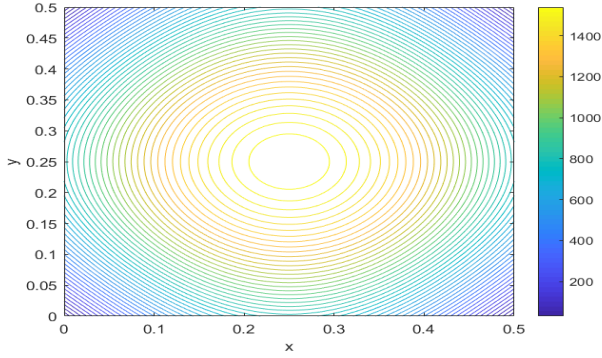


Figure 36: Contour of the gradient of the field function

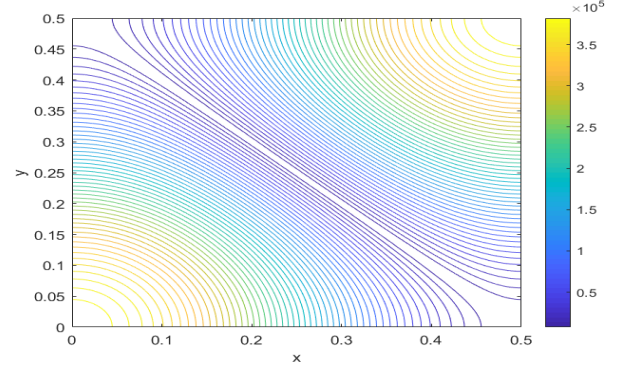


Figure 37: Contour of the fourth derivative of the field function

The meshes used in the following experiments are summarized in table 3.

Name	Grid type	Base size	Number of cells	$\epsilon_{maximum}$	ϵ_{mean}
Mesh1A	Trimmed	0.0115	1936	0.08708	0.05217
Mesh2D	Triangular + Prism layer	0.0137	1918	0.10016	0.03524
Mesh3G	Polyhedral	0.0095	2003	0.08757	0.04785

Table 1: Specifications and error values for the different meshes considered

2.2 Mesh1A: Trimmed

The mesh and error results obtained for the first test are shown in the following figures and are results of convergence with a residual of order 10^{-16} . As expected for an almost structured and orthogonal grid, the truncation error has the biggest influence on the function field error and it is reflected on the similarity between the fourth order derivative graph and the obtained solution for the error. On the other hand, the error value for the boundary condition area was expected to be the smaller value possible since the value of the function at that location is the function itself. For unknown reasons, that is not what the plot shows.

The analytical solution for a point in the red region (maximum error) is 340. Therefore, the maximum error 0.087 is approximately 0.025 % of the function value, and by performing an average volume report for the absolute error, the average error was found to be 5.217×10^{-02} , the highest value of the implemented meshes, because as can be observed in figure 39, a lot of cells have the maximum absolute error.

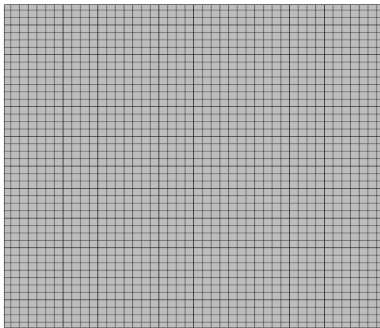


Figure 38: Trimmed mesh for the domain considered

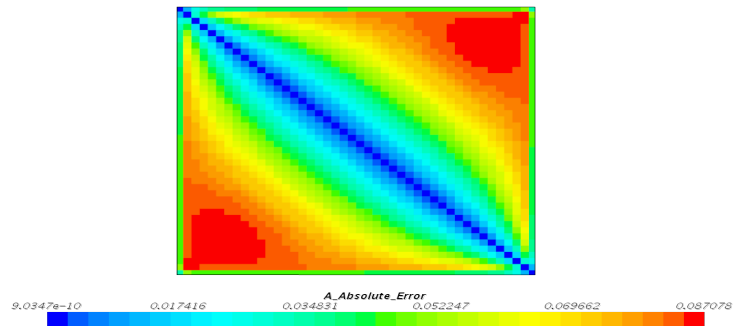


Figure 39: Absolute error for trimmed mesh

2.3 Mesh2D: Triangular and prism layer

This 2D mesh consists of triangles concealed by layers of rectangles disposed around the edges. Prism layer meshes are usually implemented to refine the mesh where there is boundary layer, since it is a complex phenomenon with very small scale and the solution gradients are very high. A fine mesh could be used but would be very costly, so, instead, prism layers are added up near the wall, which is a time and cost effective solution that gives good convergence and fairly accurate results. The prism layer incorporates two layers and 60% of the base size. As for the base size, a value of 0.017 was used and growth factor was set to 1.01 in order to improve size continuity between elements.

With this mesh the obtained error is no longer symmetric. It can be said mesh quality is good because there are no extremely acute or obtuse angles and therefore, little skewness effect. Even though the maximum error in this case is higher than for the trimmed mesh, the average is better, as can be seen in table 3. Implementing prism layer at the edges reduces both the maximum and average error, in particular close to the boundaries. The maximum error that without prism layer would be close to the boundary, in this case is pushed to the second generation of triangular cells.

The results obtained for the analytical error of this mesh are shown in figure 41.

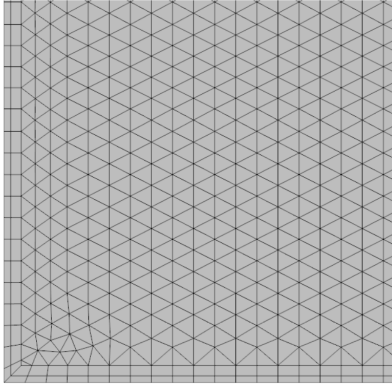


Figure 40: Zoom about the corner of the square domain

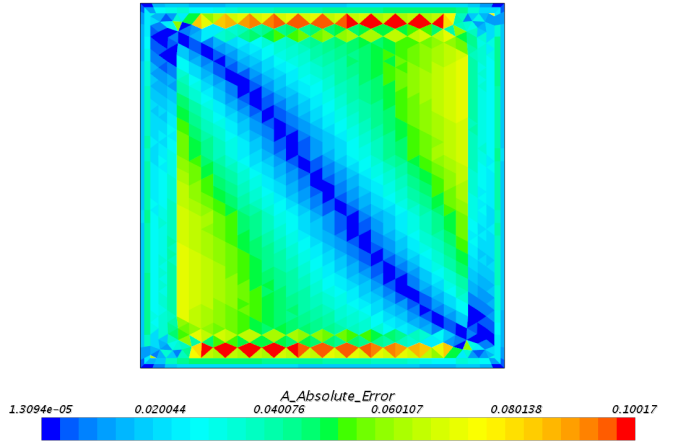


Figure 41: Absolute error

2.4 Mesh3G: Polyhedral

The polyhedral mesh shows some advantages when compared to the triangular (i.e. tetrahedral) mesh, which (under the right circumstances) enables it to deliver better quality results. For our particular problem, the major benefit of the polyhedral mesh is that each cell has a relatively high number of neighbour cells, so the approximation via gradients works better than in tetrahedral meshes. Obviously, while this leads to a higher accuracy, computing for a higher number of neighbour cells also demands more computational resources. [4]

This mesh's error distribution is similar to the trimmed case, in line with the truncation error. It produced a maximum error very close the the trimmed mesh and a lower average error.

The results obtained for the analytical error of this mesh are shown in figure 43.

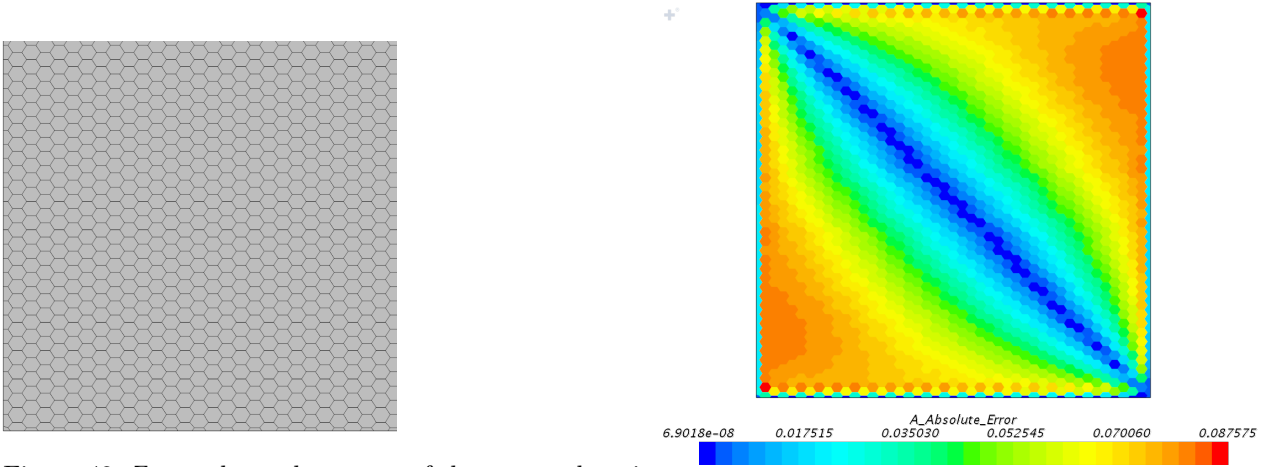


Figure 42: Zoom about the corner of the square domain

Figure 43: Absolute error

2.5 Mesh4A: Error decay with respect to mesh base size

In this subsection it is required to evaluate the evolution of the mean and maximum error in the same mesh grid with different number of cells. Thus, several simulations were made using a trimmed grid, decreasing the base size value in order to increase the number of cells. One could question the effect of turning of the secondary gradients correction method, however, for the trimmed mesh, it produces no changes.

Cells	Base size	$\epsilon_{maximum}$	ϵ_{mean}
196	0.035	1.0728	0.5641
400	0.025	0.4114	0.2565
1025	0.016	0.1688	0.0982
1936	0.0115	0.0871	0.0522
4096	0.008	0.0421	0.0259

Table 2: Specifications and error values for the different meshes considered

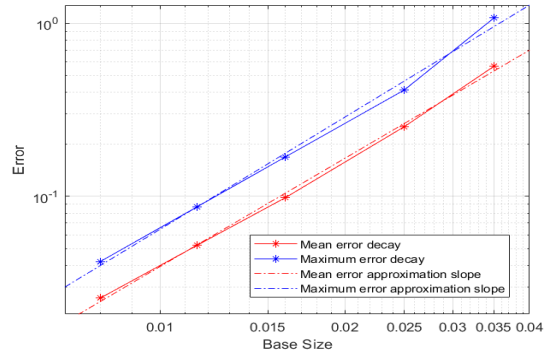


Figure 44: Error decay in a logarithmic graphic

With the aid of MATLAB and the function *polyfit* the slope of the obtained results is computed. The value for the mean error decay slope is 2.077, showing, as expected a proportionality between the error and the square of the base size. For the maximum error the slope has a value of 2.153, showing that the maximum value also has a relation with the base size value. This relation was expected since the errors are only influenced by Taylor series truncation errors as the trimmed mesh has no warp or skewness.

2.6 Mesh3G: Secondary gradient and Neumann boundary condition influences

The Star-CCM+ software implements the secondary gradient method by default to correct the errors evoked by the non-orthogonality of the mesh. Hereby, to reduce the errors introduced by the cell face gradients of skewed/warped cells (figure 35), secondary gradients for diffusion are added to the term. By deactivating this option, the software does not solve for the secondary gradients, neither at the boundaries of the domain nor at interior mesh faces. As expected, turning the correction method off leads (under otherwise equal conditions) to larger errors as shown by

table 3.

Cells	Base size	Grad. Off		Grad. On	
		$\epsilon_{maximum}$	ϵ_{mean}	$\epsilon_{maximum}$	ϵ_{mean}
218	0.03	1.9377	0.7686	0.9116	0.4602
431	0.022	2.9090	1.1625	0.4844	0.2296
1073	0.013	2.4736	0.9515	0.1671	0.0896
2003	0.0095	2.5973	0.9686	0.0875	0.0478
3938	0.0067	2.6899	1.0312	0.0438	0.0242

Table 3: Specifications and error values for the different meshes considered

While the distribution of the error with the secondary gradient correction ON (figure 45) displays according to ones expectations, the error with the secondary gradients OFF (figure 46) shows a different pattern. Without the secondary gradients, the term experiences no correction tangent to the cell face. Hence, for areas where the gradient field (figure 36) includes a large tangent proportion, the error increases. For areas with the flow almost normal to the cell face, the error reaches a minimum. Thus, there is a distinct minimum along the bottom-left to top-right diagonal, combined with two articulate maxima in the boundary-unaffected areas above and below said diagonal.

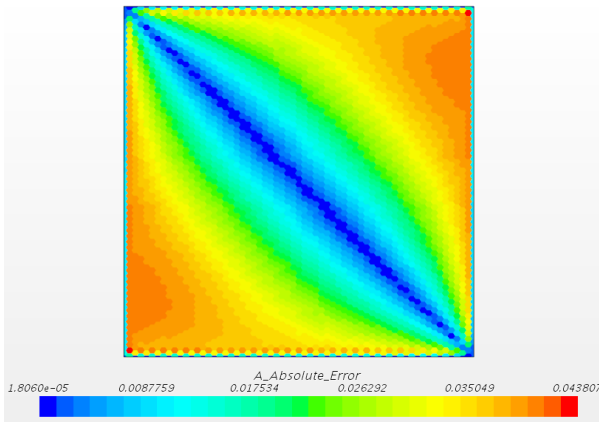


Figure 45: Error distribution with secondary gradient on in a polygonal mesh with 3938 cells

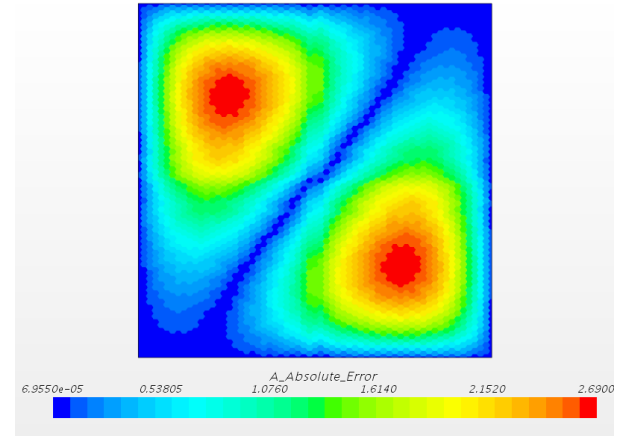


Figure 46: Error distribution with secondary gradient off in a polygonal mesh with 3938 cells

To determine the influence of Neumann boundary conditions, the boundary is divided and adiabatic condition is imposed to the top, left and bottom sides. As for the right side, the boundary condition is the same, e.g. Dirichlet, with value equal to the solution considered itself.

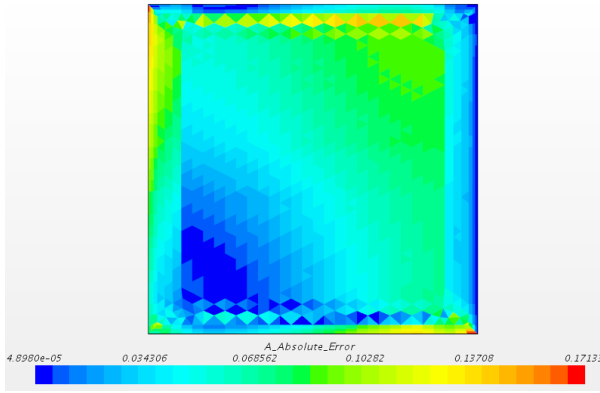


Figure 47: Error distribution with triangular-prism-layer mesh with 2158 cells

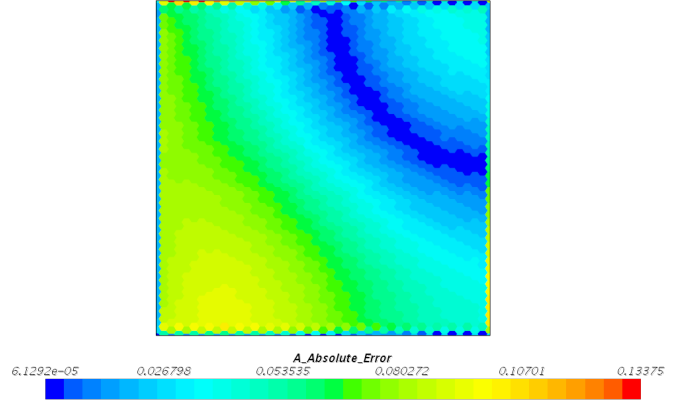


Figure 48: Error distribution with secondary gradient off in a polygonal mesh with 2003 cells

2.7 Strategy implemented to get grid with lower error

In this subsection, the objective is to build up a strategy in order to obtain a grid with less than 700 cells and the best possible results, i.e. lowest mean error. The mean error is the one used to compare results since it gives a good estimate along the domain, whereas the maximum error only allows a local estimation. With that in mind, a prism layer was applied to a triangular mesh. The strategy implemented was based on the following points:

1. Obtain different grid types with around 700 cells and determine the smallest mean error in order to choose a the mesh to improve.
2. Closely analyse the areas where the error distribution has the greatest values and determine the reason for this values. In this part, one should, once again, observe figures 36 and 37 and try to find irregularities on the mesh related to warp and distance between cell centroids.
3. Refine locally the mesh at desired locations.
4. Adjust parameters such as the base size, growth rate, number of prism layers, target surface size to find a good compromise between refined area and the rest of the mesh.

After the first step, the grid that showed best values for the mean error is a triangular with prism layer mesh. It shows a mean error of 0.0863 with 782 cells. The error distribution is similar to that of the figure 41. Clearly, the transition from the prism layer to the triangular elements lead to the maximum error value. This cells error distribution show that, as further away from the diagonal with fourth gradient equal to zero, the error increases. Thus, one can assume that truncation error plays a role on these values, since the distance between elements is also the biggest registered. Also, the non-orthogonality between the elements and the heat flux will provide *skewness* and *warp* errors. The mesh is then refined at these areas as shown in figure 49. The reduction of cell size in the respective areas allows for a better discretization of higher gradient values. The final mesh and error distribution is then showed in the next figures.

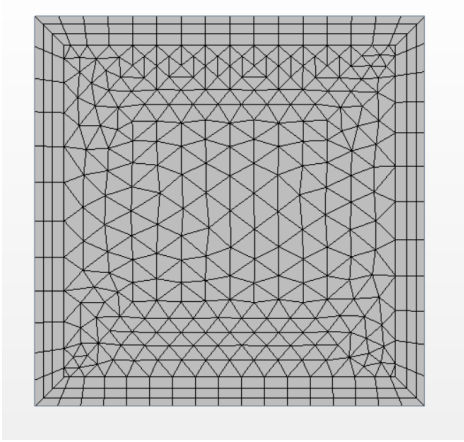


Figure 49: Mesh grid of the best solution obtained

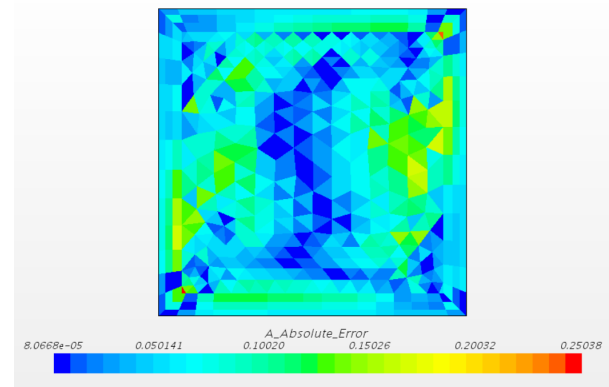


Figure 50: Error distribution for the lowest mean error

Table 4 shows some of the parameters of the mesh. The base size of the refined mesh is defined as a percentage of the base size of the triangular mesh itself. The final result has an improvement of 27% and we should also consider that the number of cell sizes was decreased almost 100 cells. The prism layer was used to take advantage of the lower errors in the boundary domain. Thus, a larger layer was used since it covers a broad area of the domain close to the boundaries and allows for a higher number of cells in the required locations.

Cells	Number of prism layers	Base size of refined cells	Growth rate	$\epsilon_{maximum}$	ϵ_{mean}
684	3	60%	1.25	0.2504	0.0629

Table 4: Specifications for the final mesh

Once again, it should be noticed that despite the maximum error being a high value, it only occurs locally due to the bad shape of the cell, proving that the mean error has a lot more influence on the mesh quality. Having improved the mesh quality in this region, the maximum error is now at the worst cell and, as it can be seen, it has at least one bad angle. It is hard to achieve a perfect mesh when the cell number is limited, since when one region's resolution is increased, results will deteriorate somewhere else in the mesh as number of cells will reduce and size will increase.

References

- [1] Baldauf M. *Stability analysis for linear discretisations of the advection equation with Runge-Kutta time integration*, Journal of Computational Physics 227 (2008) 6638-6659
- [2] Hirsch C. *Numerical Computation of Internal and External Flows. Fundamentals of Numerical Discretization, vol. 1*. Wiley: New York, 1988.
- [3] Moukalled et al. *The Finite Volume Method in Computational Fluid Dynamics : An Advanced Introduction with OpenFOAM (R) and Matlab*. Springer International Publishing AG, 2015.
- [4] Peric, M.; Ferguson, S. *The advantages of polyhedral meshes*. www.cd-adapco.com, 2004.
- [5] Guerrero, J. *Introduction to Computational Fluid Dynamics: Governing Equations, Turbulence Modeling Introduction and Finite Volume Discretization Basics*. Research Gate, 2015.