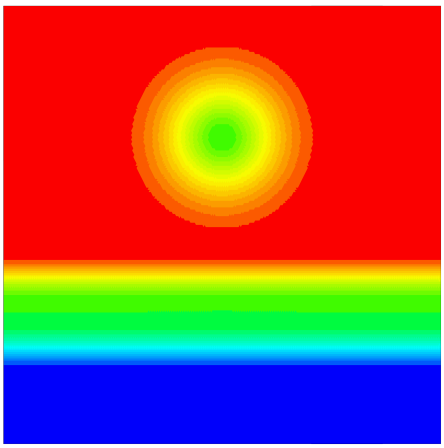
INSTITUTO SUPERIOR TÉCNICO

UNIVERSIDADE DE LISBOA

1º Semester 2023/2024

Computational Fluid Mechanics

2nd Computational Exercise: Stability, Error Analysis, and Mesh Influence in CFD Simulations



GROUP NUMBER 7

João Fernandes Marques, 93270

Miguel Dias Casalinho, 93307

João Pedro Sousa Gaspar, 96930

Pedro Afonso Simões Gonçalves, 105745

28 november 2023

Contents

1 Introduction

Computational Fluid Dynamics (CFD) serves as an engineering instrument utilized across diverse domains in project design. It enables the examination of various flow parameters within scenarios involving heat transfer and overall fluid dynamics. While CFD is now readily available through widely used commercial codes, it remains both an art and a precise science. This dual nature complicates the reliable assessment of its outcomes, particularly for individuals new to the tool or lacking expertise in fluid dynamics and numerical methods, who may struggle with the nuances required for establishing the appropriate configuration and input parameters.

The purpose of this project is to respond to the suggested work of HW2 that aims to provide some of this sensibility when dealing with stability and errors in both spatial and time discretization and the implication of different mesh types in the solution of a 2D heat transfer problem.

The first part presents the solution to the 1D convection equation with different initial conditions and periodic boundaries. Five different schemes are evaluated, which employ different combinations of discretization schemes. Discussions of the results are supported by Von Newman stability analysis as well as the study of the behaviour of diffusion and dispertion errors. The second part of the project relates to mesh quality and topology. By using START CCM+, different cell types are compared. Having the analytical solution, it is possible to evaluate the absolute error and, with this, extract valuable information about procedures to improve the accuracy of the results.

2 Convection schemes 1D

This section will focus on the analysis of the 1D convection equation $\frac{\partial u}{\partial t} = -a\frac{\partial u}{\partial x}$, with a=1 and applied to a set of 1D waves. Five discretization schemes will be analysed, all of them being different in either the space or the time discretization scheme. During the study of these schemes, several phenomena will be studied, namely their spectral behaviour, stability with Courant Number ($\sigma = a.(dt/dx)$) and overall adequacy to simulate the given waves.

As previously mentioned, five discretization schemes were used. Due to conciseness, only some of the discretized equations are shown bellow. The scheme presented in equation 1 is the Leapfrog Method which is 2nd Order accurate with the application of 2nd order central differences (SOCD) to both space and time. In contrast, equation2 shows the second scheme used, with a simpler approach, only 1st order accurate, applying Forward Euler in time and 1st order upwind (FOU) in space. In addition, 2 schemes applied Runge-Kutta of 4th order in time with the space being discretized by SOCD in one and FOU on the other. Furthermore, a custom TVD scheme was designed in order to have a non linear 2nd order scheme in space with Forward Euler in time. This scheme is presented in figure 1. The mathematical description of Ψ is given in the expression 3. To note that all of the studied methods are explicit methods.

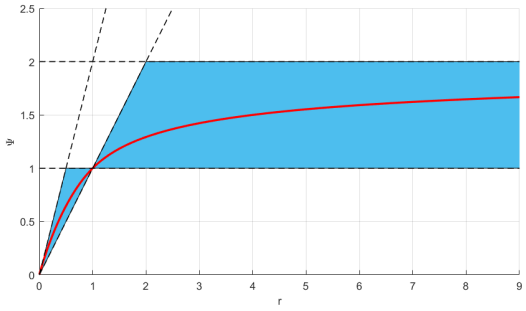


Figure 1: Graphical representation of custom TVD Scheme

$$u_i^{n+1} = u_i^{n-1} - \sigma(u_{i+1}^n - u_{i-1}^n) \qquad (Leapfrog\ Method\ O(\Delta x^2),\ O(\Delta t^2)) \tag{1}$$

$$u_i^{n+1} = u_i^n - \sigma(u_i^n - u_{i-1}^n) \qquad (Forward\ Euler\ with\ 1st\ Order\ Upwind\ Method\ O(\Delta x),\ O(\Delta t)) \tag{2}$$

$$\Psi(x) = \begin{cases} 0 & \text{if } r < 0 \\ r(2-\sqrt{r}) & \text{if } 0 \le x < 1 \\ 2 - \frac{1}{\sqrt{r}} & \text{if } x \ge 1 \end{cases} \tag{3}$$

2.1 Numerical Results

The application of the aforementioned schemes to a single wave after 1 revolution are presented in Figure 2. The Courant number chosen was 0.5 which leads to a stable solution for all 5 schemes. It can be seen that first order methods present severe amplitude errors, as is expected by the spectral analysis, similar to what is done in section 2.2. Furthermore, despite the better amplitude accuracy on second order space discretization schemes, oscillations before the wave are visible, which deteriorate the solution and increase error along time iterations. In addition, phase errors are also visible for SOCD, where the wave seems to drag behind the analytical solution. On the bottom right corner, the custom TVD scheme is presented, which demonstrates the efficacy of non linear schemes in preventing oscillations in the solution, adapting the scheme to the local gradients. Results for other initial waves are presented in the Appendix section 5.1.

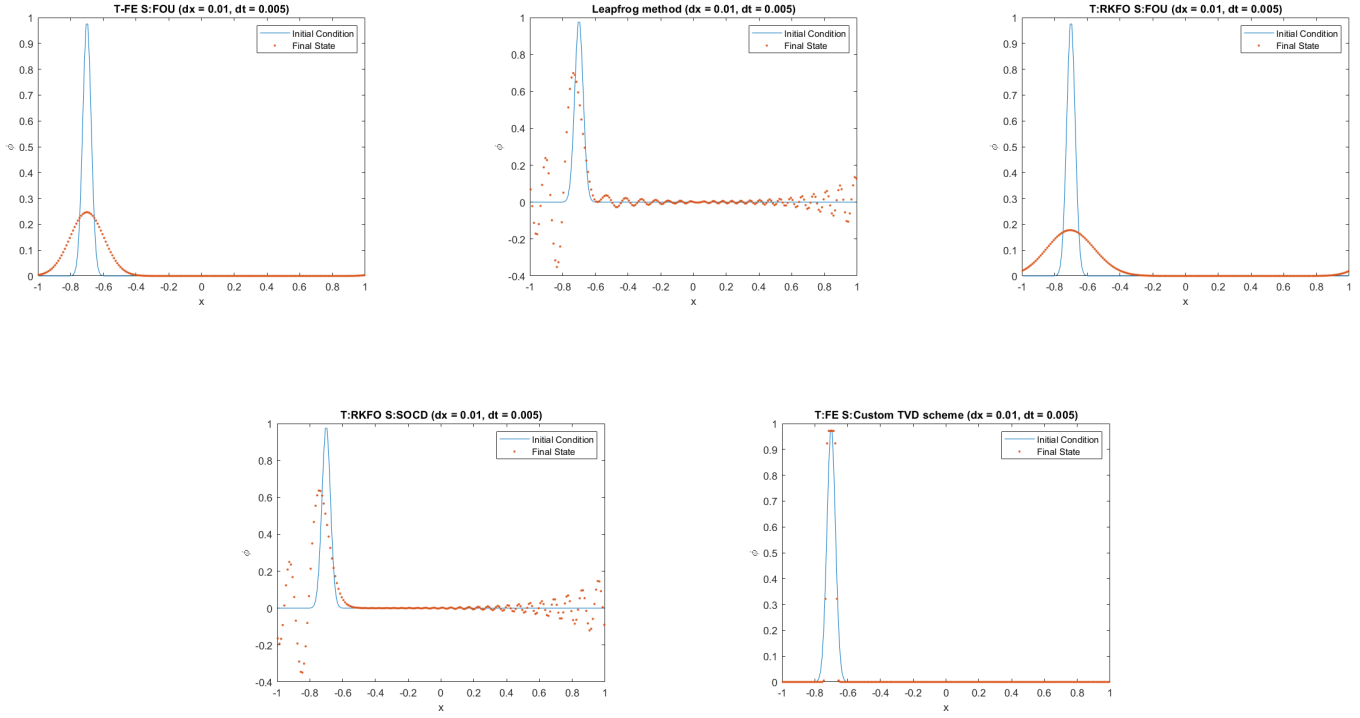
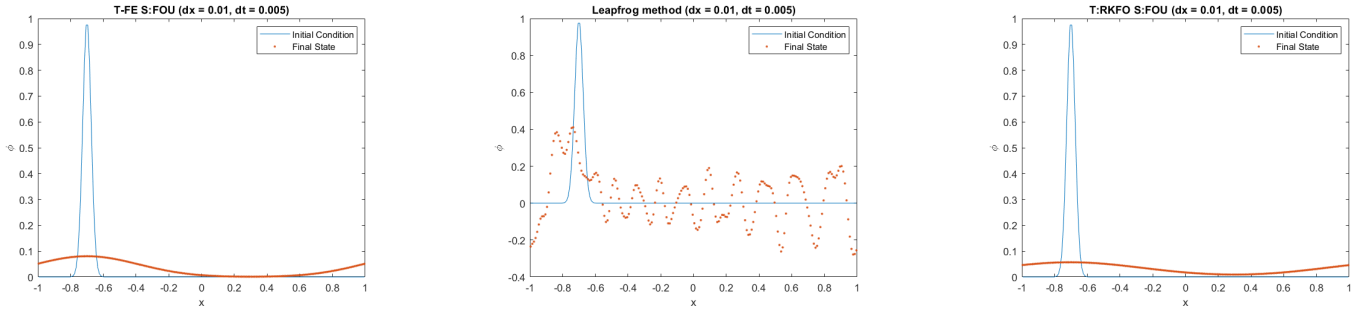


Figure 2: 1 revolution of 1D wave with CFL = 0.5 ($\Delta t = 0.005$ $\Delta x = 0.01$. From left to right, up to bottom: Explicit First Order Upwind; Leapfrog; Range Kutta 4th Order with First order Upwind; Range Kutta 4th Order with Second Order Central Difference; Explicit Custom TVD Scheme

In order to better understand the consequences of the application of each scheme, 10 full revolutions were calculated (Figure 3). Similar to the single revolution, Courant of 0.5 was considered. Through the visual analysis of the plots, it is possible to conclude that first order methods are not suitable for time dependent problems where accuracy is paramount. The high diffusive behaviour leads to intolerable amplitude errors, not suitable for engineering applications. Furthermore, SOCD oscillations present in one revolution led to the propagation of errors that "contaminated" all the domain (dispersive errors). This can also be seen on the results for the simulation with the complete set of waves in Figure 4. Once again, the TVD scheme seems to be best choice regarding accuracy, with no visible phase errors nor amplitude errors. Despite that, in figure 5 it shows the results obtained using the FOU method when $CFL = 1$, that follow exactly the initial condition with all the wave shapes. This occurs due to the fact that, for $CFL = 1$, this method has no diffusion errors nor dispersion errors for phase angles $\phi < 90$. In the Appendix 5.2, figures 29,30 and 31 show the stability region diagram, and diffusion and dispersive error plots, respectively, that prove the previous statement. The way to obtain this figures uses the same methodology used in section 2.2 for the leapfrog method.

The difference between Runge-Kutta 4th Order (RK4) and Forward Euler seems to be related with the stability range. For instance, for a Courant number of 1.75, RK4 with SOCD is still stable, whereas the other schemes cannot withstand this Courant number. This property of the RK4 method allows for solving problems where the wave speed is very high, this is, problems with high



Courant Numbers which would be unstable for other time discretizations. This way, by applying RK4, computational costs related with implicit methods can be avoided, as well as keeping the same cell size and time step. Despite this, they present higher diffusion errors when compared with Foward Euler as demonstrated by Figure 28 present in the Appendix.

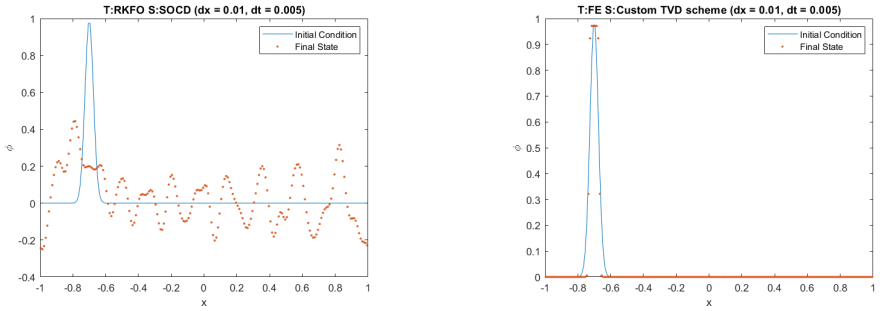


Figure 3: 10 revolutions of 1D wave with CFL = 0.5 ($\Delta t = 0.005$ $\Delta x = 0.01$. From left to right, up to bottom: Explicit First Order Upwind; Leapfrog; Range Kutta 4th Order with First Order Upwind; Range Kutta 4th Order with Second Order Central Difference; Explicit Custom TVD Scheme

2.2 Polar diagram and stability errors of Leapfrog method

By experimenting with different values of Courant, the stability limits can be estimated, but at this point it seems important to reach a more analytical value of this stability criterium in order to confirm if its implementation is good. With this in mind, and with the discretized leapfrog method equation 2, a stability analysis can be performed, with the Von Neuman methodology (I represents the imaginary unit), starting with:

$$V^{n+1}e^{Ii\phi} = V^{n-1}e^{Ii\phi} + \sigma V^n(e^{(i+1)\phi} - e^{I(i-1)\phi}) \tag{4}$$

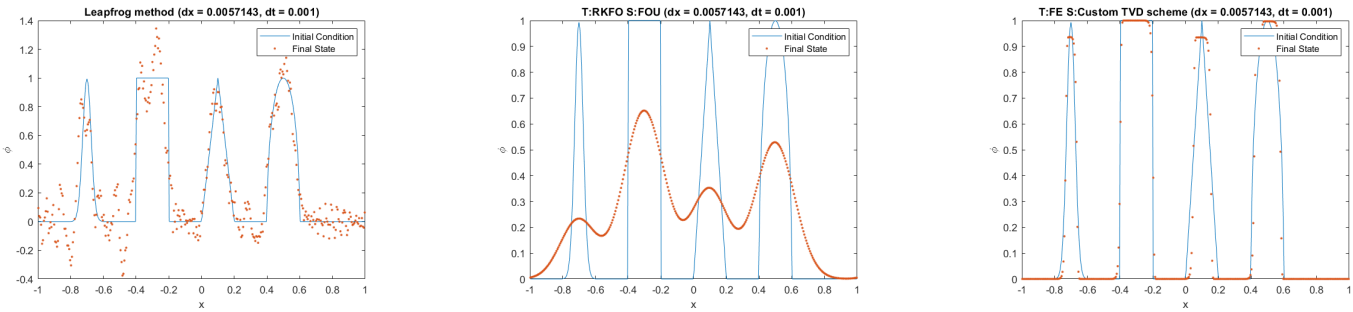


Figure 4: 1 revolution of 1D set of waves with CFL = 0.175. From left to right: Leapfrog; Range Kutta 4th Order with First Order Upwind; Explicit Custom TVD Scheme

The rest of deduction until the solution is found for the amplification factor G is:

$$V^{n+1} = V^{n-1} + \sigma V^n (e^{I\phi} - e^{-I\phi}) \Rightarrow \frac{V^{n+1}}{V^n} = \frac{V^{n-1}}{V^n} - \sigma(e^{I\phi} - e^{-I\phi}) \Rightarrow G = \frac{1}{G} - \sigma.2Isin\phi \Rightarrow G^2 + (2\sigma Isin\phi)G - 1 = 0 \quad (5)$$

Applying the quadratic formula to the deducted equation 5, the solution for G is shown in equation 6 and this solution is also represented in the complex plane, for different values of σ by attributing $0 < \phi < 2\pi$ (figure 6). In this plot, it is also represented the unit circle where inside it $|G| \leq 1$, i.e., the discretized equation is considered stable if all the points of the solution of G for a certain Courant number are inside the circle.

$$G = -I\sigma sin\phi \pm \sqrt{1 - \sigma^2 sin^2\phi} \quad (6)$$

Because the equation that is being discretized is hyperbolic, the diffusion and dispersion errors can be calculated with equations 7 and 8, respectively. The results are plotted in figures 7 and 8.

From figure 6 it can be seen that for $\sigma > 1$ the leapfrog method becomes unstable, since for a certain phase angle the absolute value of G becomes greater than 1. The phase angles for which this happens can be identified in figure 7 to be between 80 and 100 degrees, when the negative branch of the solution of G goes to absolute value higher than 1. In the dispersion error figure it can be seen that, for stable values of Courant, this type of error becomes more significant the lower the σ.

So in conclusion, for the leapfrog method, since there's no diffusion error for any stable value of Courant, the Courant number should be the highest possible stable value in order to minimize dispersion errors.

$$\varepsilon_D = |G| \quad (7) \qquad \varepsilon_\phi = \frac{\Phi}{\sigma\phi} = \frac{arctan(-\frac{ImG}{ReG})}{\sigma\phi} \quad (8)$$

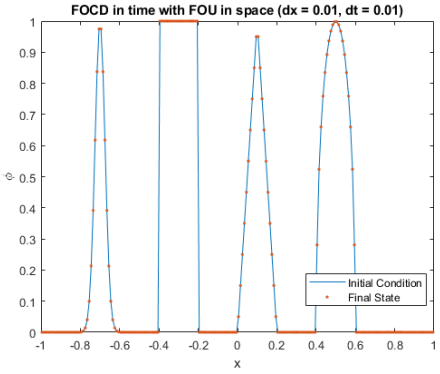


Figure 5: 1 revolution of 1D set of waves with CFL=1 for Explicit First Order Upwind method

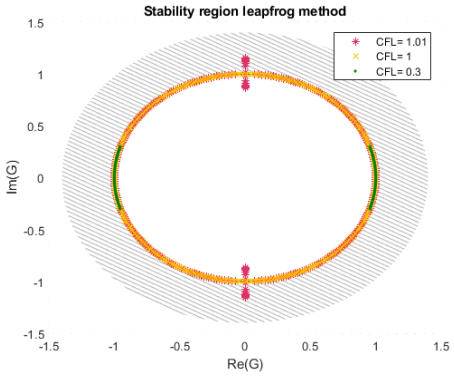


Figure 6: Stability

With the same Von Neuman method, the stability conditions can be obtained for the other discretizations used in this homework. So, in summary, by looking at the literature [1] and [2] and confirming by experimenting with our program, those stability conditions are:

- Temporal first order explicit with spatial first order upwind $\rightarrow 0 \leq \sigma \leq 1$
- Leapfrog method $\rightarrow |\sigma| \leq 1$
- Runge-Kutta 4th order with spatial central differences $\rightarrow |\sigma| \leq 2\sqrt{2}$
- Runge-Kutta 4th order with spatial first order upwind $\rightarrow 0 \leq \sigma \leq 1.39265$

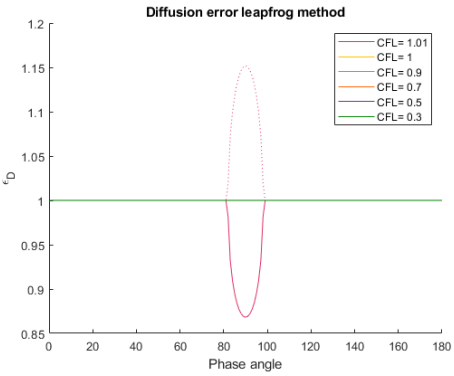


Figure 7: Diffusion error

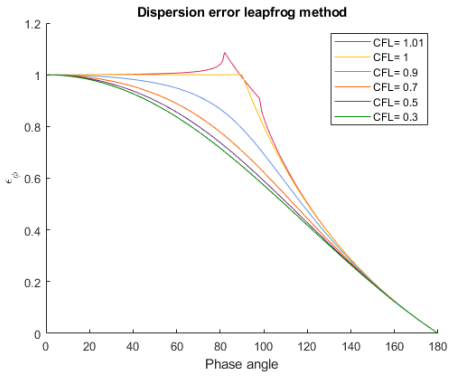


Figure 8: Dispersion error

It is relevant to note that the methods that use spatial first order upwind discretization methods are never stable for negative values of Courant number, since the way the differences are made needs to be oriented with the direction of the flow for them to work, and be in better overall accordance with the physics of the problem.

2.3 Grid size, Courant and time discretization analysis of Leapfrog method

Following the previous section, where the stability analysis was done for the Leapfrog method and also considering the results shown in section 2.1, it seems necessary to study how the grid size and the Courant number influence the solution. The results obtained for the Leapfrog method shown previously in figure 2 are replicated here in figure 9 in **pink**, while the colors **black** and **green** are the results obtained by either increasing the CFL to 1 (the stability limit for this method) without changing the grid size, and the results obtained by decreasing the grid size by a factor of 10 while keeping $CFL = 0.5$, respectively. Figure 10 shows the results after 10 revolutions of the wave.

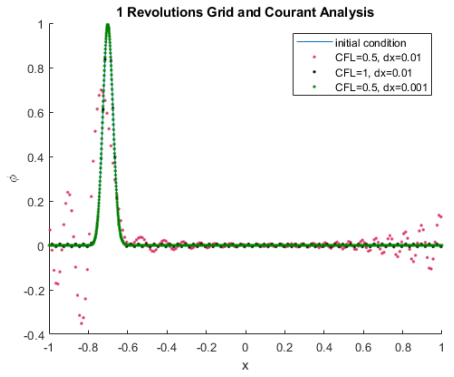


Figure 9: Comparison after 1 revolution

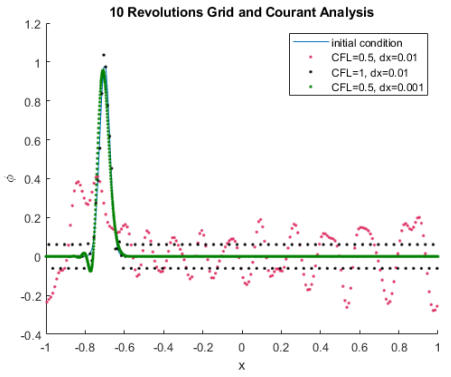


Figure 10: Comparison after 10 revolutions

It is apparent that for the same grid size, increasing CFL makes improves significantly the solution after 1 revolution. There are still some oscillations where the initial condition is 0, as there were in the baseline figure, but they are much smaller. This happens because, in this discretization method, as it was shown in figure 7, the diffusive error is not influenced by changing the Courant, but the dispersive error is more significant for lower CFL (figure 8). So, the oscillations are still present, but in general, there is no dispersion of the solution,i.e, it follows the initial condition without lagging behind. After 10 revolutions of the wave the amplitude of this oscillations becomes 10 times bigger.

Next, looking at decreasing the grid size while maintaining Courant number at 0.5 shows that the solution is much more accurate after 1 revolution without the lagging. This happens because the grid points are much closer to one another, the phase angle of the harmonics of the solution is lower, so we are in the region of the plot of figure 8 where there's almost no error, i.e., the beginning of the curve. Despite not being apparent after 1 revolution, there's still some dispersion errors that become more noticeble after 10

revolutions (see Figure 10).

3 Heat transfer in unstructured grids

In the second phase of this study, this work focus on solving the two-dimensional Laplace equation, incorporating a diffusion term ($\nabla^2\phi + f = 0$), subject to Dirichlet boundary conditions taking temperature as the unknown variable, using Siemens' STAR-CCM+ software. The adopted approach involves implementing a manufactured solution, 9, and a forced term, 10.

$$\phi = 300 + 100\left\{\tanh\left[(y+2)^3\right] - \exp\left[-\frac{13}{20}x^2 - \frac{13}{20}(y-2)^2\right]\right\} \tag{9}$$

$$\begin{aligned}\frac{f}{100} = {} & 3\cdot(2y+4)\left[\tanh^2(y+2)^3 - 1\right] - \frac{13}{5}\exp\left[-\frac{13}{20}(y-2)^2 - \frac{13}{20}x^2\right] + \exp\left[-\frac{13}{20}(y-2)^2 - \frac{13}{20}x^2\right]\cdot\left(\frac{13}{10}y - \frac{13}{5}\right)^2 \\ & + \frac{169}{100}x^2\exp\left[-\frac{13}{20}(y-2)^2 - \frac{13}{20}x^2\right] - 18\tanh\left[(y+2)^3\right]\left[\tanh^2(y+2)^3 - 1\right](y+2)^2\end{aligned} \tag{10}$$

By employing Computational Fluid Dynamics (CFD) techniques to solve numerical problems, we emphasize the robustness of this approach. The research is centered on evaluating the effectiveness of the CFD mesh in solving the mentioned equation. The primary focus of this study is to obtain a CFD mesh that minimizes both the maximum and average errors in the solution. Initially, it is conducted a comparative analysis of different types of meshes, including polyhedral, triangular, and trimmed meshes, all generated under the same conditions. Subsequently, the mesh that provides the best resolution is selected for a process of mesh refinement, aiming to further minimize its errors.

3.1 Simulations Star CMM+

To present three different meshes, Star CMM+ was utilized, and simulations were conducted for polyhedral, trimmed, and triangular meshes. In order to achieve meshes with approximately 5000 cells, certain parameters in the software, such as base size and target surface size, were varied.

A uniform mesh was also examined by adjusting the surface growth size to a value of approximately 1. The different uniform meshes are depicted in figures 32, 33, and 34 of the Appendix, for the polyhedral, triangular, and trimmed meshes, respectively.

3.1.1 Numerical Error

After obtaining various meshes with a cell count close to 5000, the simulations were proceeded to generate 2D plots illustrating the obtained numerical error. These plots are depicted for the three meshes in figures 11, 12, and 13.

The meshes are comparable due to the selection of a similar cell count. To achieve the equivalent number of cells as in the trimmed and polyhedral meshes, the base size of the triangular volumes had to be increased.

To compare the different meshes and provide insights into the results, the maximum error and mean error were calculated for each mesh, as presented in table 1.

Before conducting simulations in Star CCM+, the expectation was that cells with more faces would exhibit greater accuracy due to a better representation of multi-directional gradients. In other words, according to this expectation, polyhedral cells were anticipated to be more accurate when using the same number of cells, as having more neighboring cells would lead to improved gradient approximations.

	Base Size	Cells	Maximum Error	Mean Error
Polyhedral Mesh	0.15 m	5006	0.5974	0.09987
Triangular Mesh	0.2 m	4950	0.5223	0.09533
Trimmed Mesh	0.14 m	5184	0.6679	0.1089

Table 1: Mean and Maximum error for different mesh base elements and cell count of roughly 5000.

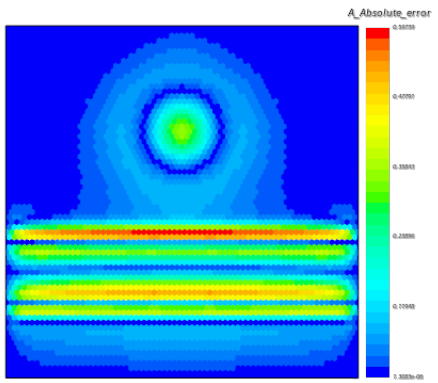


Figure 11: Absolute Error for Polyhedral Mesh.

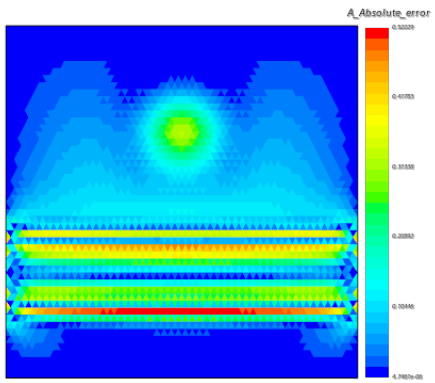


Figure 12: Absolute Error for Triangular Mesh.

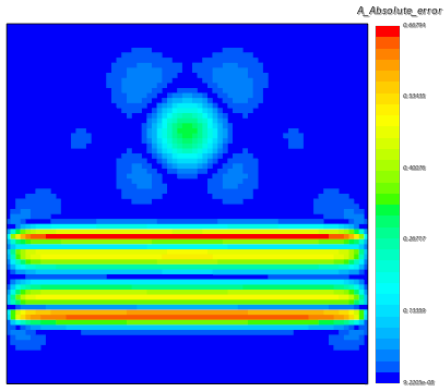


Figure 13: Absolute Error for Trimmed Mesh.

However, as evident from table 1, the findings deviated from the initial expectation. The triangular mesh emerged as the most accurate, exhibiting the lowest mean error, followed by the polyhedral mesh, and lastly, the trimmed mesh. Regarding the maximum error, the trimmed mesh exhibited the highest error, followed by the polyhedral mesh, and finally, the triangular mesh. The triangular mesh presents the best results since it is the most compact cell shape. More centroids are within the same area for the same cell size, which decreases distances between centroids and decreases the error.

3.1.2 Decay order for Mean and Maximum Error

In order to ascertain the corresponding decay order for both maximum and mean errors, the polyhedral mesh was chosen. A series of simulations with varying base sizes was conducted, and the resulting data was utilized to generate the graph depicted in figure 14.

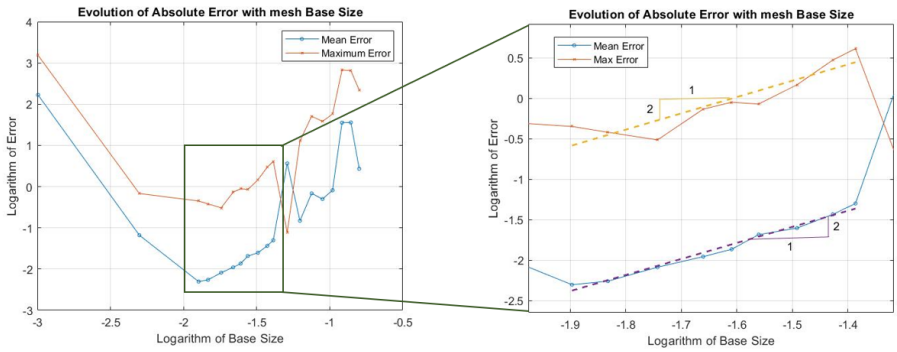


Figure 14: Evolution of Mean and Maximum Error with Base Size.

Similar to previous analyses in Homework 1, the determination of the order of accuracy can be achieved graphically. This approach involves fitting a linear regression to the data within the range where asymptotic behavior is evident. For this analysis, attention was focused solely on data points demonstrating asymptotic behavior, as illustrated in the graph on the right in figure 14.

The graphical analysis led to the conclusion that both the decay order for maximum error and mean error was approximately 2. This finding provides empirical support for the understanding that Star CCM+ program employs second order discretization.

3.2 Mesh Refinement

In the previous section, the study focused on the use of various types of meshes for heat transfer problem. It was concluded, as shown in table 1, that the triangular mesh yielded the best results with a maximum error of 0.5223 and a mean error of 0.09533. Consequently, the aim now is to establish a strategy for mesh refinement with the objective of minimizing the error.

Firstly, it is essential to emphasize the vertical symmetry present in the solution depicted in figures 11, 12 and 13. Due to this symmetry, it opens the possibility for the simulation of only half of the domain. This reduction in domain and, considering only 3000 total cells, already provides a finer mesh than the previous cases. Achieving the same level of precision in the simulation would require placing at least twice the number of cells, which, from a computational standpoint, would have significant implications.

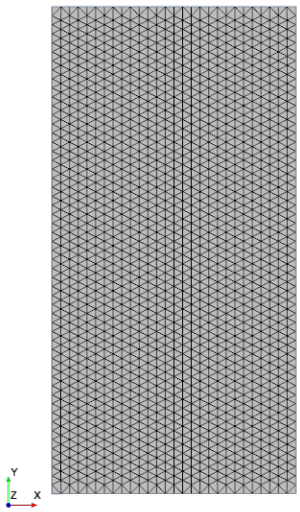


Figure 15: Baseline mesh for refinement.

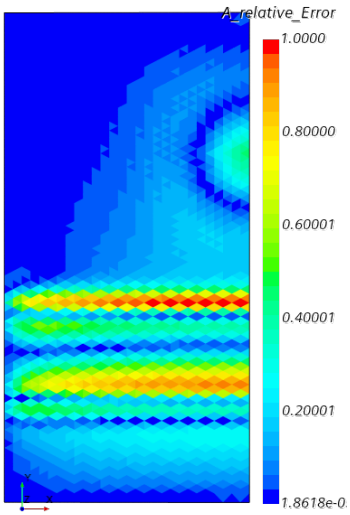


Figure 16: Relative error.

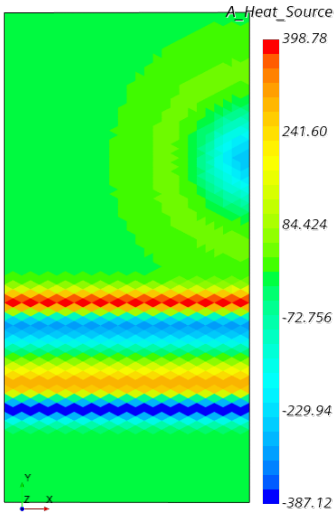


Figure 17: Heat Source

After this improvement, the focus changes to local refinements of the mesh in figure 15 in order to minimize the obtained error. As a refinement criterion, two initial hypotheses will be considered: 1) relative error and 2) heat sources. The first hypothesis arises for obvious reasons, by identifying error concentration regions and refining these, improvements in the results are expected. On the other hand, the second hypothesis arises from the physics of the problem, as the solution is strongly influenced by second derivatives.Because linear interpolation is used between cells, errors will be greater in the presence of high second derivatives, thus utilizing mesh refinements can be a good method to mitigate errors.Second Derivatives can be known by the source term which, by definition is equal to the Laplacian of Temperature

To implement these two different strategies, two functions were created. Each of these functions aims to generate a refinement table that will be applied to generate a new mesh, addressing the reduction and enlargement of cell sizes to refine areas of higher error. This reduction and enlargement of specific cells aim to ensure restrictions on the number of cells while always keeping their count under the maximum allowed. As a result of applying the functions described earlier, figures 18 and 19 depict the refinement regions in blue, and in red regions where refinement is not as necessary. Therefore, based on these regions and once the cell growth rate is defined according to the criterion, the new meshes are generated.

The mesh shown in figures 20 and 21 are the resultant meshes from the refinement process for each hypothesis. Inside the blue area, as explained earlier, it is possible to observe that the cell size has decrease and inside the red area the cell size has increased. In both cases, refined was obtained in the desired area with total number of cells of 2996 and 2917 for the relative error and heat source methods respectively.

The results highlighted in table 2 show the superiority of heat source-based refinement, indicating a more efficient and localized approach to enhancing simulations. This conclusion justifies what was indicated earlier: the heat transfer problem is strongly influenced by second derivatives. After this results, the mesh of Figure 21 is subjected to another refinement process by improving the refinement function and creating an additional condition to make transition between cell sizes more gradual. This way, while regions of increasing and decreasing mesh size remain, a third in between region is defined in which cell size is maintained which will allow a smoother transition between cell sizes. This function is shown bellow. To enhance the mesh in Figure 21, we optimize the refinement function with regions based on different values (Figure 22). This ensures a gradual mesh generation: refining in the

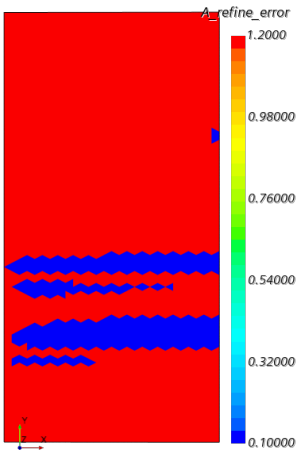


Figure 18: Region where the relative error is above 40 %.

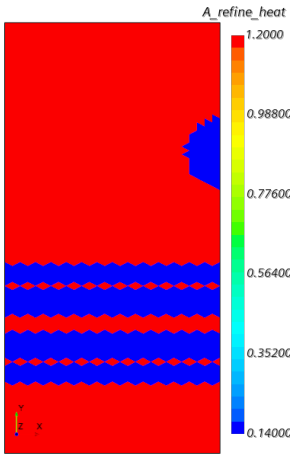


Figure 19: Region where the heat source is above than 100 K.

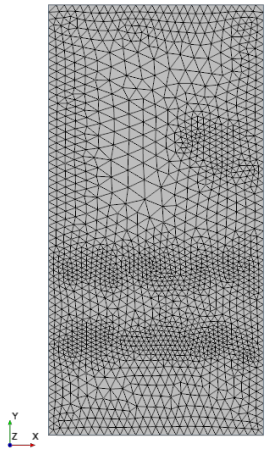


Figure 20: Mesh generated from the relative error refinement condition

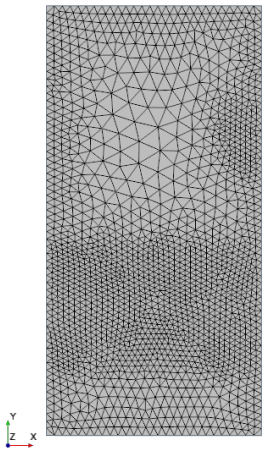


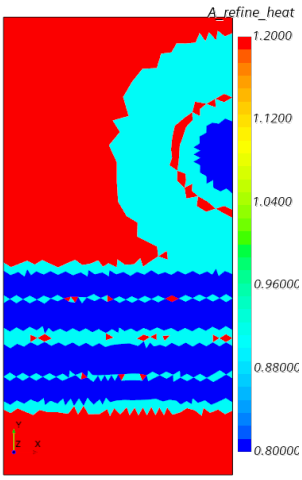
Figure 21: Mesh generated from the heat source refinement condition

Refinement Method	**Maximum Error**	**Avg. Abs. Error in Volume**
Error-based Refinement	0.7888	0.1450
Heat Source-based Refinement	0.4447	0.0773

Table 2: Comparison of Refinement Methods in Simulation

first zone, transitioning smoothly in the second, and increasing cell size in the third, where the heat source has minimal influence.

```
abs($A_Heat_Source) > 100 ? 0.8 : abs($A_Heat_Source) < 10 ? 1.2 : 0.9
```



This function looks for the value in the cell and if the absolute value of the heat source is above 100, cell size is decreased by a factor of 0.8, whereas if this value is bellow 10, mesh size can be increased (factor of 1.2), regions in between will have a slight refinement of factor 0.9. The function has two key parameters. The first, vital for refinement, sets values above 90 for the blue-highlighted refinement zone and below 5 for the less critical zone. The transition region spans values between 5 and 90 for the heat source. The second parameter, the refinement criterion, takes values of 0.165, 0.85, and 1.05 for the refinement, transition, and less crucial refinement zones, respectively, in the example.

Figure 22: Refinement regions example

This function above looks for the value in the cell and if the absolute value of the heat source is above 100, cell size is decreased by a factor of 0.8, whereas if this value is bellow 10, mesh size can be increased (factor of 1.2), regions in between will have a

slight refinement of factor 0.9. The function has two key parameters. The first, vital for refinement, sets values above 90 for the blue-highlighted refinement zone and below 5 for the less critical zone. The transition region spans values between 5 and 90 for the heat source. The second parameter, the refinement criterion, takes values of 0.165, 0.85, and 1.05 for the refinement, transition, and less crucial refinement zones, respectively, in the example.

The current focus is on fine-tuning the analysis to identify optimal parameters for addressing the heat transfer problem with the goal of minimizing errors. Initial insights from Table 3 suggest a refinement strategy's impact on reducing both average and maximum errors. The key observation is the identified refinement region, suggesting that values above 50 for heat sources should be considered, while values below 20 are deemed part of the non-refinement zone. The transition zone is consequently established between 20 and 50.

To enhance mesh refinement, the size of the refinement zone was reduced to 15% of the initial base size, resulting in an increased number of cells within the domain, now totaling 2849. Additionally, a shift was made in the stopping criterion, favoring a count of iterations over the error between iterations. A sufficiently large error between iterations was defined, leading to the conclusion that the desired level of accuracy was achieved within a fixed maximum of 111 iterations.

Finally, after executing the mesh refinement process explained throughout this section, the result obtained is shown in figures 23 and 24. Regarding errors, the maximum error was reduced from 0.5223 to 0.38182, approximately a 27% reduction, and the average error decreased from 0.09533 to 0.06074, around a 36% reduction.

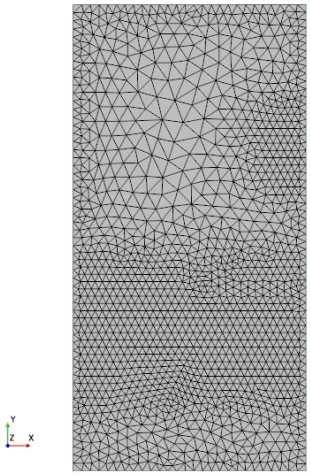


Figure 23: Final mesh after refinement

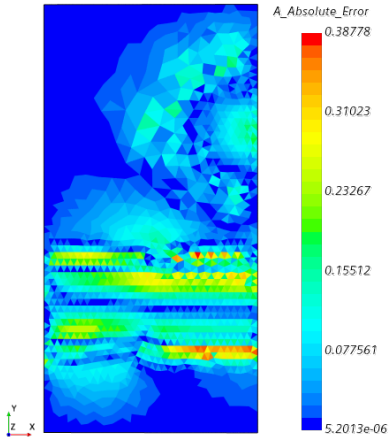


Figure 24: Absolute error after refinement

4 Conclusion

Computational Fluid Dynamics can be defined as the application of numerical methods on the resolution of fluid flow and heat transfer problems. In order to understand and apply this tool to its full potential, knowledge on what is behind the code is important and many times disregarded. This way, this project aimed to analyse the behaviour of different discretization schemes as well as the intricacies of mesh typologies and refinement.

The first part of this work looked into the stability of different discretization schemes analysing effects of Courant, grid spacing and number of revolutions. Spectral analysis was also performed on the leapfrog method. In turn, the second part of this work involved a study of different mesh types for the given problem, and it was concluded that the most suitable mesh would be triangular. Subsequently, a mesh refinement procedure was carried out, making it possible to reduce the maximum error by 27% and the average error by 36%. It can be stated that the refinement process yielded good results.

References

[1] M. Baldauf, "Stability analysis for linear discretisations of the advection equation with runge–kutta time integration," *Journal of Computational Physics*, vol. 227, no. 13, pp. 6638–6659, 2008.

[2] C. Hirsch, "Numerical computation of internal and external flows: The fundamentals of computational fluid dynamics," in Elsevier, 2007, ch. 7.

5 Appendix

5.1 Results for Single Revolution

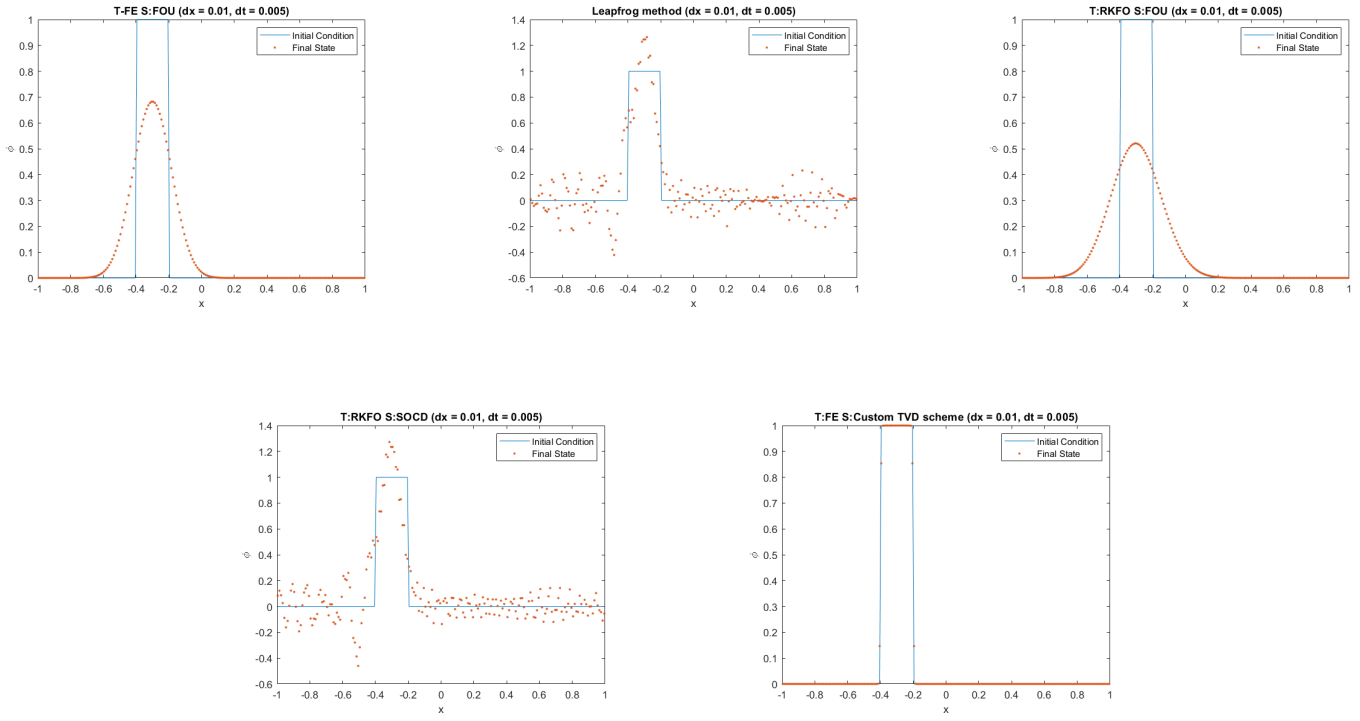


Figure 25: 1 revolution of 1D wave with CFL = 0.5 ($\Delta t = 0.005$ $\Delta x = 0.01$. From left to right, up to bottom: Explicit First Order Upwind; Leapfrog; Range Kutta 4th Order with First order Upwind; Range Kutta 4th Order with Second Order Central Difference; Explicit Custom TVD Scheme

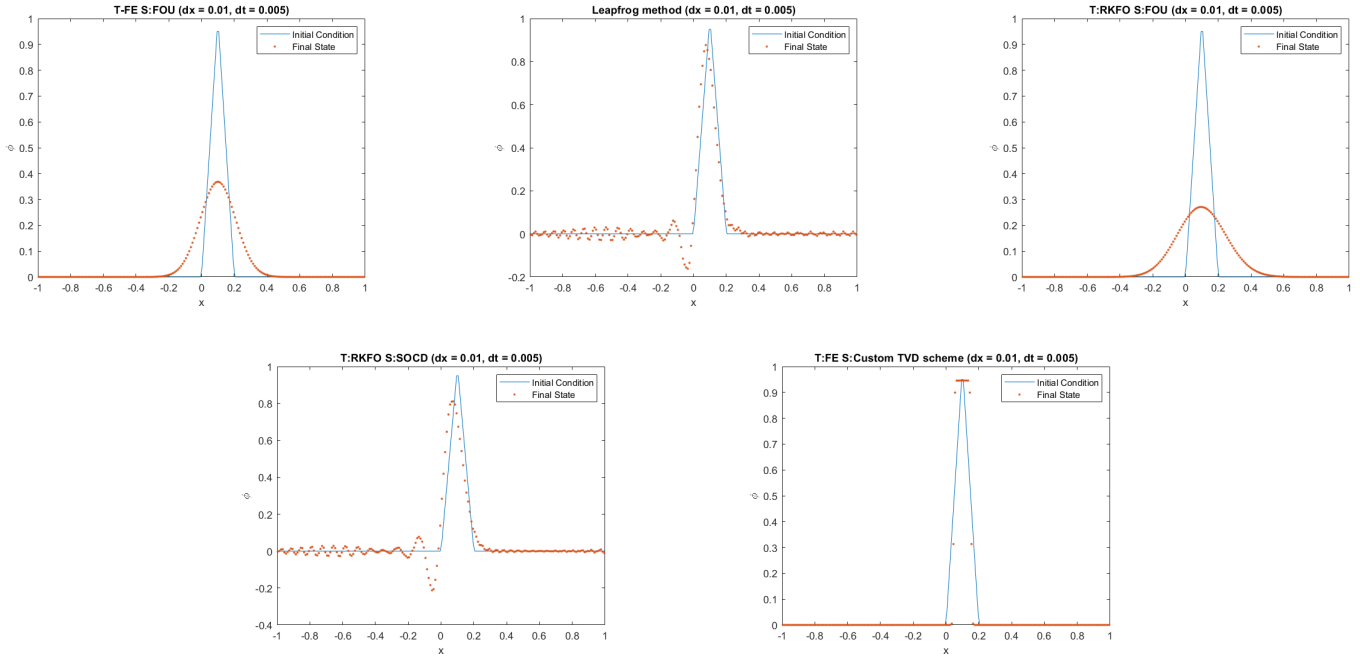


Figure 26: 1 revolution of 1D wave with CFL = 0.5 ($\Delta t = 0.005$ $\Delta x = 0.01$. From left to right, up to bottom: Explicit First Order Upwind; Leapfrog; Range Kutta 4th Order with First order Upwind; Range Kutta 4th Order with Second Order Central Difference; Explicit Custom TVD Scheme

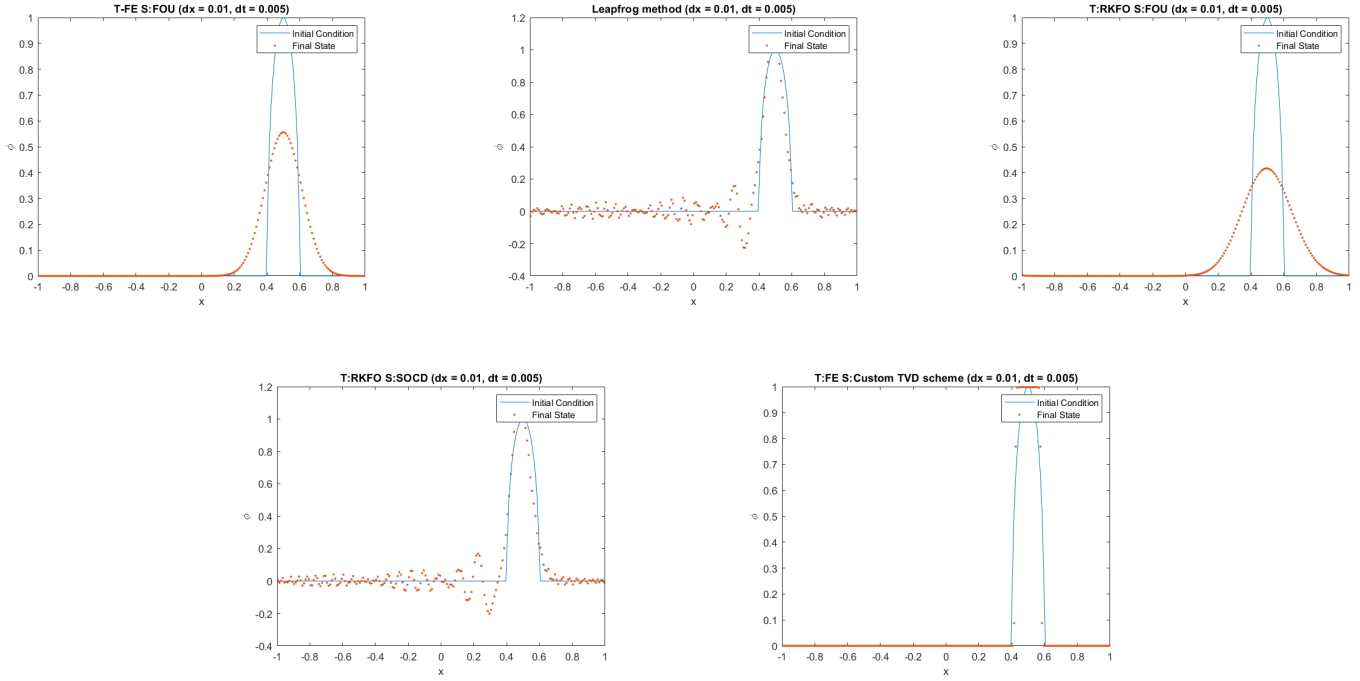


Figure 27: 1 revolution of 1D wave with CFL = 0.5 ($\Delta t = 0.005$ $\Delta x = 0.01$. From left to right, up to bottom: Explicit First Order Upwind; Leapfrog; Range Kutta 4th Order with First order Upwind; Range Kutta 4th Order with Second Order Central Difference; Explicit Custom TVD Scheme

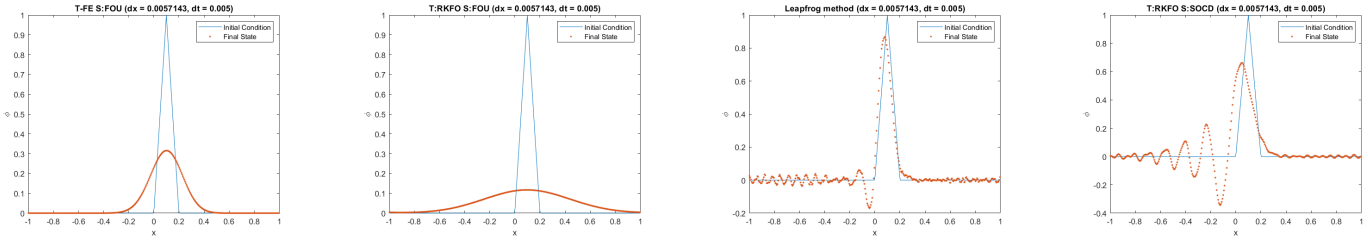


Figure 28: 10 revolutions of 1D wave with CFL = 0.875 ($\Delta t = 0.005$ $\Delta x = 0.01$. From left to right: Explicit First Order Upwind; Range Kutta 4th Order with First order Upwind; Leapfrog; Range Kutta 4th Order with Second Order Central Difference

5.2 Polar diagram and stability errors of FOU method

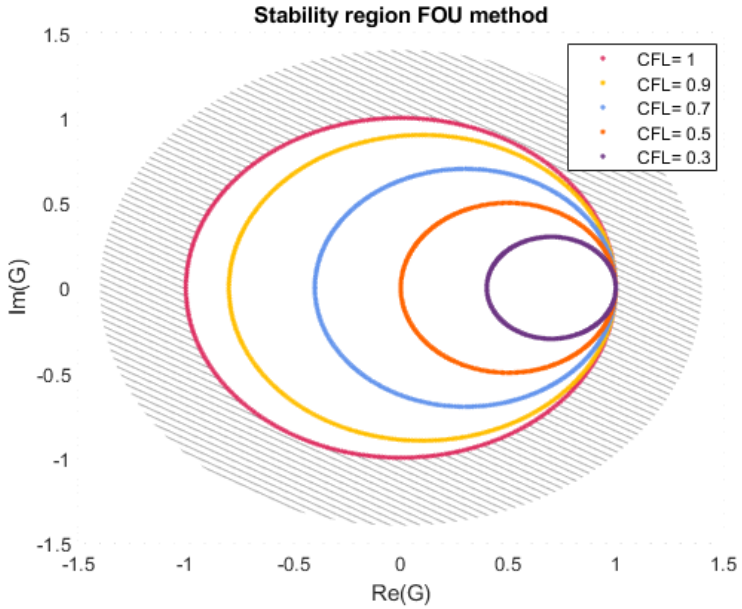


Figure 29: Stability diagram FOU method

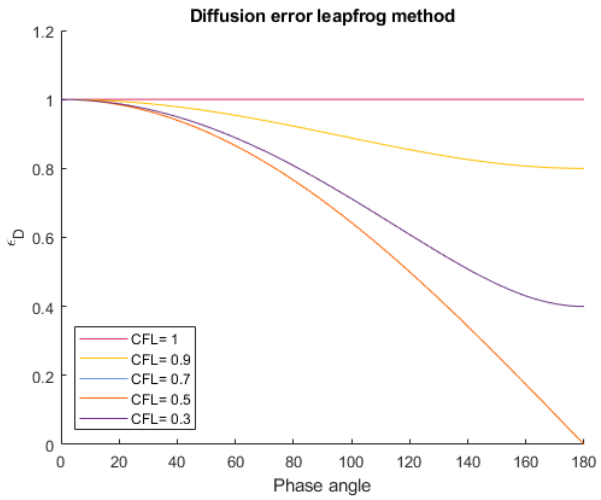


Figure 30: Diffusion error FOU method

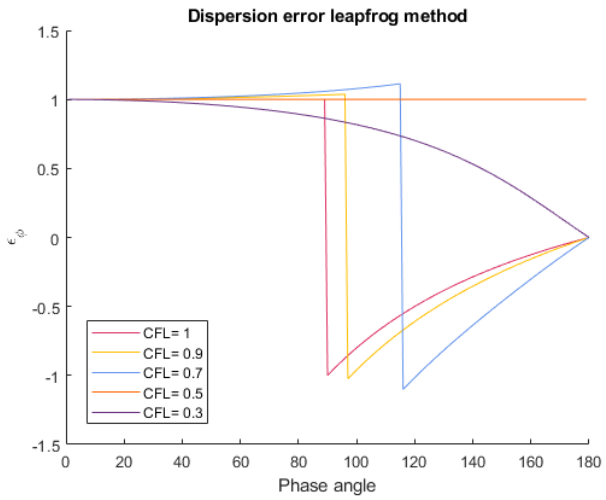


Figure 31: Dispersion error FOU method

5.3 Differente Uniform Meshes

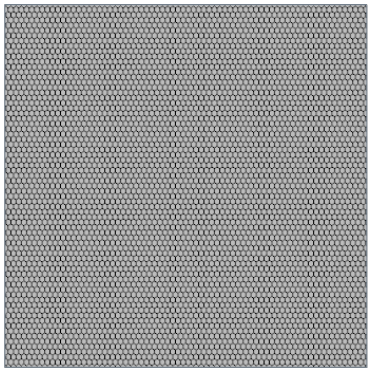


Figure 32: Polyhedral Mesh.

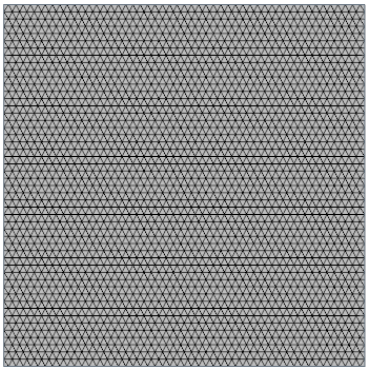


Figure 33: Triangular Mesh.

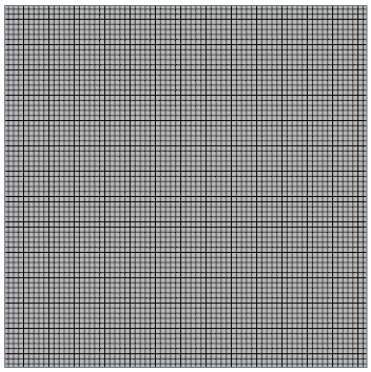


Figure 34: Trimmed Mesh.

5.4 Different Simulation Parameters

	Sup. Value	Inf. Value	C.S. Refinement	C.S. Transition	C.S. No Refinement	No. Cells	Mean E.	Max. E.	Diff. Mean E.	Diff. Max. E.
Variation of the refinement region	200	5	0.14	0.85	1.05	2743	0.13267	0.60315	-39%	15%
	150	5	0.16	0.9	1.05	2851	0.11885	0.41492	25%	-21%
	100	5	0.16	0.9	1.05	2978	0.14586	0.79444	53%	52%
	50	5	0.17	0.9	1.05	2909	0.09434	0.56470	-1%	8%
Variation of the transition region	50	30	0.17	0.9	1.05	2922	0.13653	0.67228	43%	29%
	50	20	0.17	0.9	1.05	2901	0.07195	0.41629	-25%	-20%
	50	10	0.17	0.9	1.05	2922	0.13653	0.67228	43%	29%
	50	2	0.17	0.9	1.05	2922	0.13653	0.67228	43%	29%
	50	1	0.17	0.9	1.05	2922	0.13653	0.67228	43%	29%
Combination	150	30	0.17	0.9	1.05	2675	0.08427	0.35956	-12%	-31%
	150	20	0.17	0.9	1.05	2657	0.08427	0.35956	-12%	-31%
	50	30	0.17	0.9	1.05	2922	0.13653	0.67228	43%	29%

Table 3: Refinement mean error and maximum error study for different regions size (Legend: Sup. = Superior; Inf = Inferior; C.S. = Cell size; Diff. = differnece; E. = Error; Max. = Maximum)