TRABAJO DE FIN DE MÁSTER

FACULTAD DE CIENCIAS || MÁSTER DE ASTROFÍSICA

NON-UNIFORM GRID AS A KEY ELEMENT TO GRASP MULTI-SCALE PHENOMENA IN THE SOLAR ATMOSPHERE

MATÍAS KOLL PISTARINI

SUPERVISED BY:

Dr. MIKHAIL MODESTOV
Dr. ELENA KHOMENKO

Julio 2022

AGRADECIMIENTOS

En primer lugar quería agradecer la labor de mis supervisores Mikhail y Elena por sus consejos y la amplia disponibilidad que me han brindado. He apendido mucho de ellos, sobre todo en las numerosas tutorías en la cafetería.

También agradecer a todos mis amigos y compañeros de carrera y máster que me han acompañado durante este viaje, sin ellos no hubiese sido lo mismo. En especial mencionar a Adri, Amy y María, quienes me han dado un apoyo fundamental en esta última etapa.

Y por último, y más importante, a mi familia. En especial a mi madre y mi padre, quienes han hecho posible que pueda seguir estudiando lo que me gusta y me han apoyado de forma incansabale, aguantándome en los momentos más difíciles. Este logro también es vuestro.

EXTENDED ABSTRACT

La atmósfera solar presenta una estructura compleja debido a la multitud de fenómenos físicos que tienen lugar en escalas características muy variadas. Además, al ascender en altura, las propiedades del plasma cambian bruscamente. En la fotosfera, la presión del gas es la que domina el movimiento del plasma, mientras que en la corona solar el movimiento está gobernado por el campo magnético. Estas regiones están conectadas por medio de la cromosfera, la cuál está caracterizada por importantes cambios en las propiedades del plasma. Estudiar en detalle esta capa es fundamental para entender cómo se transporta la energía desde la parte baja de la atmósfera hasta regiones externas, de manera que se produzca el calentamiento coronal observado.

En los últimos años, el estudio de ondas magnetohidrodinámicas (MHD) se ha convertido en uno de los mejores candidatos para explicar el transporte de energía que tiene lugar en la atmósfera solar. Son capaces de transportar energía a las capas superiores de la atmósfera, pudiendo liberarla y calentar el medio a través de diferentes mecanismos, como por ejemplo la formación de choques. Este tipo de perturbaciones se pueden generar tanto en regiones del Sol en calma (movimientos turbulentos en la base de la fotosfera), como en regiones activas con fuertes campos magnéticos (manchas solares). Durante su propagación, las ondas pueden experimentar diferentes procesos dependiendo de la propiedades locales del plasma. La variabilidad de las escalas espaciales características en la que tienen lugar dificulta su estudio a nivel observacional.

Gran cantidad de códigos numéricos se han desarrollado en los últimos años con el fin de reproducir los múltiples procesos que tienen lugar en la atmósfera solar, pudiendo compararse con datos observacionales. En general, la resolución espacial utilizada en las simulaciones numéricas viene determinada por la potencia computacional existente. De esta manera, se han llevado a cabo múltiples intentos para aumentar la eficiencia computacional en los códigos numéricos. La implementación de mallas no-uniformes o mallas adaptativas ha sido uno de los principales aspectos a mejorar, al reducir significativamente las extensiones de las mallas utilizadas. Durante este trabajo hemos implementado en el código MANCHA3D una malla fija no-uniforme en la dirección vertical, con el objetivo de reducir la potencia computacional empleada.

En primer lugar, hemos desarrollado un algoritmo para la generación de una malla no-uniforme. Utilizando una distribución uniforme pre-existente, se han reajustado los puntos de la malla a partir de una serie de criterios. Nuestro objetivo ha sido aumentar la resolución de la malla en aquellas regiones en las que la atmósfera solar presente importantes gradientes de temperatura, como la región de transición. A su costa, la resolución espacial en zonas como la corona solar se ha reducido. Durante el proceso de creación de la nueva malla, ha sido necesario limitar la distancia máxima entre puntos adyacentes. En caso contrario, los métodos numéricos pueden volverse inestables y ser una fuente de errores. De esta manera, presentamos distintas distribuciones dependiendo del número total de puntos utilizado y la resolución máxima deseada.

A continuación, hemos analizado los distintos módulos del código MANCHA3D en los que interviene la componente vertical. En concreto, se han llevado a cabo modificaciones en los módulos de derivadas espaciales, difusividad, filtrado, cálculo del paso temporal y PML. Hemos llevado a cabo un análisis en detalle de cuáles han de ser las modificaciones necesarias a introducir con respecto al código original. Todos los cambios introducidos se ha realizado de manera que se mantenga una buena compatibilidad con la estructura original del código, siendo posible su ejecución paralelo. Múltiples tests se han llevado a cabo para verificar que la implementación de todas las modificaciones funcionan correctamente.

Se ha estudiado la propagación de ondas acústicas y de Alfvén a lo largo de toda la atmósfera solar. Para el caso de ondas acústicas, se han analizado simulaciones en régimen lineal y no-lineal, de manera que dan lugar a la formación de choques. Estos resultados se han comparado con los obtenidos al utilizar una malla uniforme de alta resolución. En todos los casos, el comportamiento de las ondas en su evolución es prácticamente idéntico. Pequeñas diferencias numéricas se han observado como consecuencia de emplear métodos diferentes. Para el caso de ondas de Alfvén se ha utilizado una configuración similar a la de ondas acústicas. En este caso, ha sido necesario añadir un campo magnético uniforme a lo largo de toda la atmósfera. Los resultados obtenidos por las mallas uniforme y no-uniforme son muy similares. Además, se han utilizado expresiones analíticas para comparar los resultados. Se ha obsevado un acuerdo razonable con los resultados númericos en ciertas regiones, a pesar de las limitaciones que presentan estas ecuaciones analíticas. De esta manera, hemos verificado que los cambios introducidos funcionan correctamente para simulaciones de ondas acústicas y de Alfvén.

Finalmente, se ha realizado una comparativa de la eficiencia computacional que proporciona el uso de una malla no-uniforme. Hemos tomado como referencia el tiempo de computación de la malla uniforme utilizando el código original. Estos resultados, los hemos comparado con el tiempo empleado por simulaciones con mallas no-uniformes. Se han estudiado mallas con distinto número de puntos y resoluciones espaciales mínimas. En general, se observa una significativa mejora en la eficiencia computacional que introduce este tipo de mallas. La mejora computacional se hace más notable cuanto menor es la resolución espacial utilizada, suponiendo un gran ventaja.

Contents

1 Introduction

The solar atmosphere presents a complex structure in which a multitude of physical phenomena take place on a variety of scales. Its structure can be divided into four fundamental layers: photosphere, chromosphere, transition region and solar corona. The properties of the plasma change abruptly between each of these regions. In the photosphere the gas pressure dominates the motion of the plasma, while the corona is dominated by the magnetic field. Chromosphere connects the photosphere with the transition region and the lower corona, which make it a layer where the properties of the plasma change strongly.

A detailed study of the chromosphere is essential to understand how energy is transported from the photosphere to the corona in order to produce coronal heating. The magnetic field in this region presents a complex structure, playing an important role on heating process and energy transport. Besides, chromosphere shows a high variability and its fast temporal evolution makes it difficult to study in detail. A large amount of observational and theoretical efforts have been devoted to understand more in depth the physical processes that take place in this layer and the effect they have on the solar atmosphere.

In recent years, magneto-hydrodynamic (MHD) waves have become one of the main candidates as a mechanism for energy transport. This type of waves transport energy to the upper layers which can release and heat the medium mainly through the formation of shocks. The quiet Sun, which covers around 90% of the solar surface, is continuously perturbed by turbulent convection. This can produce the excitation of waves which will travel across the solar atmosphere contributing significantly to the heating of the outer layers (Muller et al. 1994; Trujillo Bueno et al. 2004; Martínez González et al. 2010). MHD waves are also associated with the active regions sunspots where stronger magnetic field are present. The complex structure of this magnetic field will lead to waves to experience different physical processes which may affect to the evolution and dissipation of them (Khomenko & Cally 2012, Tian et al. 2014, Grant et al. 2018).

During the wave propagation, several physical processes as mode-transformations, refraction and reflection can take place. This will depend on the magnetic field strength and the local plasma properties such as the equipartition layer where plasma $\beta = 1$, the cut-off frequency or the steep temperature gradient at the transition region. In addition, the plasma will experiment other processes such as reconnection, viscosity or thermal conductivity. Each of these phenomena take place at a specific length scale. In many cases, this introduces limitations when making ground-based observations, as the resolution of telescopes is limited. An alternative is the use of numerical simulations, which are based on theoretical developments of each of these processes. This allows an unlimited replication of the phenomena that take place, although this is limited by the current computational potential.

Several numerical codes that simulate the solar atmosphere have been developed in recent years (e.g. MURaM (Vögler, 2004); MANCHA3D (Khomenko & Collados 2006; Felipe et al. 2010); Bifrost (Gudiksen, B. V. et al., 2011); COBOLD (Freytag et al., 2012) and more). It has become clear how important and realistic they are with respect to ground-based observations, even allowing the generation of 3D simulations. Nevertheless, many simplifications have to be made in order to carry out the numerical calculations, as well as the spatial resolution allowed is not infinite. The resolution used has to be controlled to make the simulations computationally feasible. In this sense, numerous attempts have been made to improve the efficiency of numerical codes by reducing the number of grid points used such as the algorithms of adaptive mesh refinement (Keppens et al., 2012) or the usage of a non-uniform fixed grid (as used by Bifrost or COBOLD). The latter is in general simpler to implement than the former, as it only requires the generation of a fixed non-equidistant mesh at the beginning of the simulation, being able to increase the resolution in specific areas of the numerical domain. Adaptive mesh refinement requires the calculation and refinement of a mesh in a dynamic way, identifying at each iteration the regions with highest gradients in order to increase the spatial resolution in these regions. In this way, this type of method is more complex to implement.

One of the most modern and versatile numerical codes today is MANCHA3D . This code allows to perform simulations of the solar and other stellar atmospheres, being able to apply many realistic phenomena such as ambipolar diffusion, Hall's term and Biermann's battery, radiative transfer equation, realistic equation of state, etc. During this work, we will implement in MANCHA3D a non-uniform fixed grid in the vertical direction to sample more accurately regions where small-scale processes take place. In this way, it is intended to improve the precision of the results, as well as the computational efficiency of the code. Numerous simulations will be carried out to compare the implemented modifications with the original code.

1.1 MANCHA Code

MANCHA3D solves time-dependent equations of the MHD on a 3D Cartesian grid. It can solve either full or linearized MHD equations, being the following system of equation written in conservative form solved:

$$\begin{aligned}
&\frac{\partial\rho}{\partial t}+\nabla\cdot(\rho\mathbf{v})=\left(\frac{\partial\rho}{\partial t}\right)_{\text{diff}}, \\
&\frac{\partial\rho\mathbf{v}}{\partial t}+\nabla\cdot\left[\rho\mathbf{v}\mathbf{v}+\left(p+\frac{\mathbf{B}^2}{2\mu_0}\right)\mathbf{I}-\frac{\mathbf{B}\mathbf{B}}{\mu_0}\right]=\rho\mathbf{g}+\mathbf{S}(t)+\left(\frac{\partial\rho\mathbf{v}}{\partial t}\right)_{\text{diff}}, \qquad (1)\\
&\frac{\partial e}{\partial t}+\nabla\cdot\left[\mathbf{v}\left(e+p+\frac{|\mathbf{B}|^2}{2\mu_0}\right)-\frac{\mathbf{B}(\mathbf{v}\cdot\mathbf{B})}{\mu_0}+\frac{(\eta_A+\eta)\,\mathbf{J}_\perp\times\mathbf{B}}{\mu_0}-\frac{\nabla p_e\times\mathbf{B}}{en_e\mu_0}+\mathbf{q}\right]=(\rho\mathbf{g}+\mathbf{S}(t))\cdot\mathbf{v}+Q_{\mathbf{R}}+\left(\frac{\partial e}{\partial t}\right)_{\text{diff}}, \\
&\frac{\partial\mathbf{B}}{\partial t}=\nabla\times\left[\mathbf{v}\times\mathbf{B}-\eta\mathbf{J}-\eta_A\mathbf{J}_\perp+\frac{\nabla p_e}{en_e}-\eta_H\frac{(\mathbf{J}\times\mathbf{B})}{|B|}\right]+\left(\frac{\partial\mathbf{B}}{\partial t}\right)_{\text{diff}}.
\end{aligned}$$

where $\mathbf{I}$ is the identity tensor, ρ is the density, $\mathbf{v}$ is the velocity, p is the gas pressure, $\mathbf{B}$ is the magnetic field, $\mathbf{g}$ is the gravitational acceleration and $\mathbf{q}$ is heat flux vector. The dot $'\cdot'$ represents the scalar product of vectors, while the notation $'\mathbf{BB}'$ stands for the tensor product. The term $\mathbf{S}(t)$ in the momentum and energy equations represents a time-dependent external force. The term Q_R is the radiative energy exchange. Artificial diffusion terms have been added to all above equations to give numerical stability to the simulations. It is possible to use either the ideal gas equation of state or a more realistic equation of state containing a specific chemical mixture.

All the variables of the code (except for the velocities) are split into two parts: equilibrium (sub-index 0) and a perturbation (sub-index 1) as:

$$\rho = \rho_0 + \rho_1; \quad p = p_0 + p_1; \quad \mathbf{B} = \mathbf{B_0} + \mathbf{B}_1; \quad \mathbf{v} = \mathbf{v}_1 \ . \tag{2}$$

The code assumes a strictly background magnetohydrostatic equilibrium, so that all time derivatives and initial velocities are zero. In this way, the code actually solves the system of equations (1) for the nonlinear perturbations of density, momentum, magnetic field and energy. The time step is computed using an explicit fourth-order Runge-Kutta scheme.

The code is written in modern Fortran 90 language and fully parallelized using the MPI library. MANCHA3D distributes data between a certain number of processors by a domain decomposition scheme, with data contained in each processor located in an accessible memory partition. The whole numerical domain is divided into a set of three dimensional subdomains, being the processors communicated with each other only through their common data boundaries. This communication is done using ”ghost” layers, cells which send and receives data from the neighbours subdomains. Currently, a fixed number of 5 ghost cells are used to avoid problems between the different modules implemented.

2 Objectives

The main objective of this work is the implementation of a non-uniform fixed mesh in a single direction within the MANCHA3D code. In this way we aim to study phenomena occurring at different scales in the solar atmosphere by reducing the computational power required. In particular, we will focus on the study of the propagation of acoustic and Alfvén waves through the solar atmosphere. Schematically, the following objectives will be accomplished:

- To develop a method for the generation of the non-uniform grid. A series of criteria have to be derived to obtain a mesh with a variable spatial step in the vertical direction, leading to different spatial resolution along the solar atmosphere. The idea is to increase the resolution where the solar magnitudes present large spatial gradients.

- Familiarisation with the code and the modular structure that it presents. Those modules in which aspects of the numerical grid are involved in the vertical component will be analysed. In this way we will have a clear idea of the parts of the code which need to be modified.

- Apply the necessary modifications within the code that affect the use of a non-uniform mesh. These changes have to be implemented following the same structure as the current code presents, in order to ensure backward compatibility. They must also be compatible to be run in parallel.

- Perform multiple tests to verify the correct implementation of the changes using the acoustic and Alfvén wave samples. This will be done by comparing the results with those provided by the original code.

- Analyse the computational efficiency of the new configuration compared to the original code.

3 Methodology

In this section, the methodology used in order to achieve the proposed objectives is described. Firstly, the background atmosphere model used will be discussed. From this, a new model containing a non-uniform grid along the vertical component will be generated. Then, the main modules of the MANCHA3D code that are necessary to modify will be described: spatial derivatives, filtering, diffusivity, time step calculation and PML. A detailed description is given of how they have to be adapted to take into account the non-uniform spacing in the vertical direction. Finally, some aspects of the code implementation and parallelisation are addressed.

3.1 Background model

A background model for the solar atmosphere is needed which describes the initial temperature, pressure and density distributions. This model is provided to the code and an initial perturbation is added to it. As we want to study phenomena that happen through the whole atmosphere we create this model in such a way that covers from the photosphere till the solar corona using an equidistant grid in all directions, characterized by a constant dx, dy, dz. A vertical extension ranging from -75 km to 4 Mm is used to cover the main layers of the solar atmosphere.

The photospheric and chromospheric layers are described by the VALC model (Vernazza et al., 1981) where the temperature distribution is interpolated to our uniform grid. An isothermal corona is created with a temperature of $3{\cdot}10^5$ Kelvin. This temperature is not the actual temperature of the solar corona (about 1 million K), but it is a good representation for the temperature gradient from the upper layer of the chromosphere to the corona. This distribution has been used because the computational time is reduced, since the time step depends on the sound speed across the whole domain. By using this coronal temperature, the wave amplitudes in this region decreases slightly. Nevertheless, the overall behaviour of the distribution is as valid as when using a more realistic temperature value. This layer is connected smoothly by hand to the chromosphere forming the transition region, a layer with a strong temperature gradient. Once the temperature distribution is created, we can compute the pressure scale height H_0 assuming the ideal gas equation:

$$H_0 = \frac{R_{gas}T_0}{\mu g} \,. \tag{3}$$

This quantity allows us to adjust the atmosphere to be in hydrostatic equilibrium. To that end, we can compute the pressure along the vertical direction by integrating numerically the hydrostatic equilibrium equation:

$$\frac{dp_0}{dz} + \frac{p_0}{H_0} = 0 \,, \tag{4}$$

and then we are able to compute the density profile for the whole atmosphere:

$$\rho_0 = \frac{p_0}{gH_0} \,. \tag{5}$$

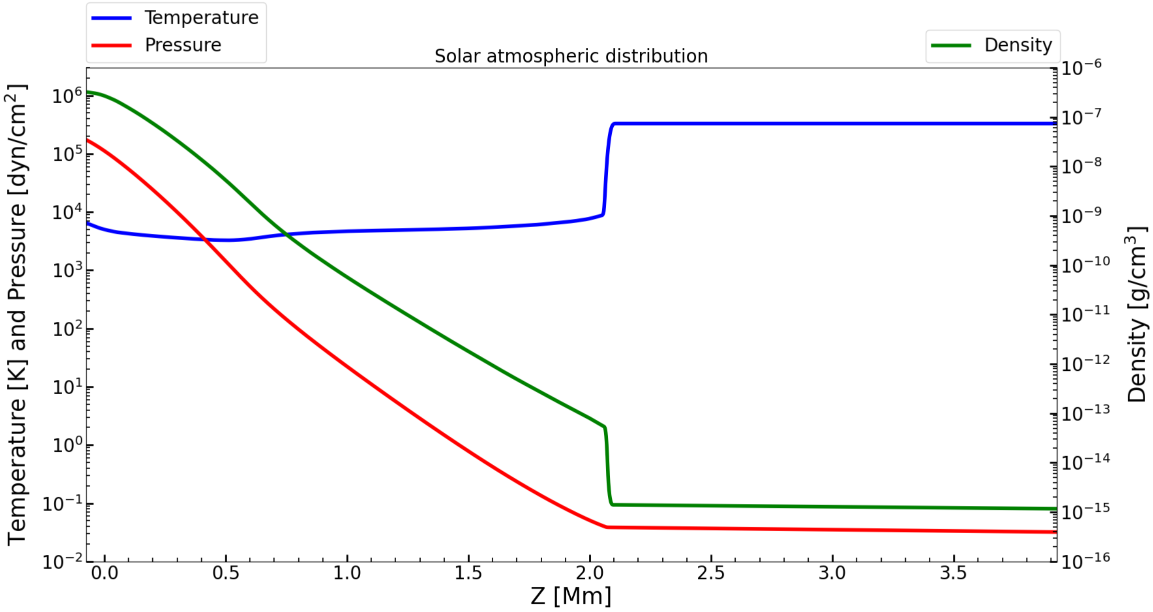

Figure 1: Background distribution of the solar atmosphere magnitudes used for the simulations. Left vertical axis gives temperature (blue) and pressure (red) values while the right vertical axis represents the density profile (green). Both axis are in logarithmic scale.

	Photosphere	Chromosphere	Corona
c_{s0} [km/s]	7	8.5	70
v_A [km/s]	2	1000	15000

Table 1: Characteristics sound and Alfvén speed values in each of the main layers in the equilibrium solar atmosphere.

Figure 1 shows the distribution of temperature, pressure and density that will be used as stratification of the solar atmosphere. It can be seen the large temperature gradient in the transition region, which is related to a large drop in density. This model of the atmosphere provides a good representation of the actual distribution of these quantities on the Sun despite not using very accurate values. An estimate of the sound speed in each layer can be computed from this model. The values of the sound speed are given in Table 1. To compute them we assume a Hydrogen monatomic gas with $\gamma = 5/3$ and $\mu = 10^{-3}$ kg along the whole atmosphere. This values will be of interest in the propagation of acoustic waves in the vertical direction. Besides, a constant magnetic field could be added to our hydrostatic model without breaking the hydrostatic equilibrium condition. In Section 4.2, Alfvén waves will be studied, for which a uniform magnetic field of 200 G has been used. Table 1 contains the characteristic Alfvén speeds for this particular situation. These values are far from being realistic, especially for the upper layers of the atmosphere, as these regions do not have such strong magnetic fields. Nevertheless, they are useful to test the code for the propagation of these waves in a simple way.

3.2 Non-Uniform grid

A non-uniform background model in the z direction has to be generated. To this end, we developed a Python script to generate a non-equidistant grid, using an iterative process, based on the atmosphere model described in the previous section . The main goal of this new grid is to have better resolution (lower values of dz) in those regions with large temperature gradients, but keeping both the number of points n_z used and the total extent of the uniform atmosphere.

To locate those regions with large temperature gradients, like the transition region, we compute the first spatial derivative of the background temperature distribution normalized to the local temperature. It is more useful to have an adimensional quantity which contains the information of where the dz should be increased or decreased. This can be reached just by multiplying the previous normalized derivative by the constant dz, obtaining a *scaled derivative*. Following the basic relation $dz = c_{s0} d\tau$, where $d\tau$ is the time it takes for a perturbation to cross each grid point, we can add the information of the scaled derivative to have a first approximation of the new dz as:

$$dz_{nonu} = \frac{c_{s_0}\, d\tau\, \varepsilon}{dT + \varepsilon}\,, \tag{6}$$

where dT is the temperature scaled derivative and ε is a free parameter. The value of dT will be higher in regions where the temperature gradient is larger, as is the case of the transition region. In the solar corona, $dT = 0$ as the temperature is uniform, so there the expression (6) is reduced to $dz = c_{s0} d\tau$. In this way, the value of dz_{nonu} will be decreased around the transition region, while in the solar corona it will remains large enough, as a high resolution is not needed there. On the other hand, ε is a variable and adimensional parameter which will change at each iteration. This is a small parameter with respect to dT, as it is responsible for readjusting the final distribution of dz until the initial constraints are fulfilled (number of points and atmosphere extension). ε is updated at each iteration using a bisection method which is sufficiently efficient and simple for what is required. This expression is valid for the study of acoustic waves, since we are using the value of the sound velocity to determine the new spatial step. For the case of Alfvén waves, the equation (6) must be modified to take into account also the Alfvén speed v_A, characteristic of this type of perturbations.

The calculation of the new space steps should be controlled through a stretching factor r_i, limiting the difference between two adjacent cells. We set the ratio between two contiguous cells to be as maximum ± 10%. Adding the stretching factor to the expression (6), the calculation of dz_{nonu} results:

$$\begin{aligned} dz^i_{nonu} &= 1.1\, dz^{i-1}_{nonu}, && \text{if } dz^i_{nonu} > 1.1\, dz^{i-1}_{nonu}\ , \\ dz^i_{nonu} &= 0.9\, dz^{i-1}_{nonu}, && \text{if } dz^i_{nonu} < 0.9\, dz^{i-1}_{nonu}\ , \\ dz^i_{nonu} &= dz^i_{nonu}, && \text{Otherwise}\ . \end{aligned} \tag{7}$$

This parameter is quite important because having large differences in spatial steps between adjacent cells introduces larger errors in the derivative schemes, as these are proportional to the grid spacing.

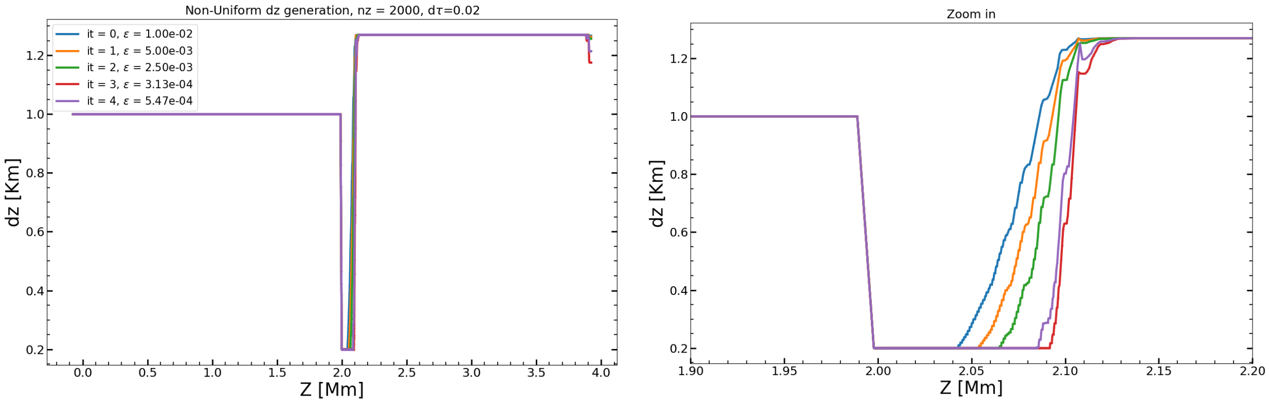

Figure 2: Example of how the non-uniform mesh generation algorithm works. A mesh with 2000 points and a minimum dz of 200 m is used. The spatial step up to 2 Mm has been kept constant. Each colour represents the successive iterations until the final solution for the grid is reached. Left panel shows the dz for the complete atmosphere, while the right panel plots a zoom in to the higher resolution region.

Figure 2 shows an example of the process of generating the non-uniform grid from an initial distribution given by (6). An initial ε parameter is needed to start the generation process. Besides, the distribution will depend on the value of $d\tau$, which takes a fixed value throughout the execution of the algorithm. Larger values of $d\tau$ will produce larger spatial steps in the corona, as in this region $dz = c_{s0}d\tau$. Thus, we can sample this region with a small number of points, concentrating more grid points around the transition region, where the dz will be significantly smaller. We can see how ε changes in order to fulfill the initial conditions of the problem. The distribution obtained may present sharp changes in the calculated dz. Therefore, we have applied a filter to smooth the distributions based on the calculation of the mean of two adjacent points on each side.

Photosphere and chromosphere are defined by the VALC model. As a result, the calculated dz_{nonu} in these layers shows significant variability (see panel a of Figure 3). In general, having this variation in the lower layers of the solar atmosphere is not of interest, since we are interested in increasing the spatial resolution mainly in the transition region. In this way, it has been added the possibility to fix the dz along the photosphere and chromosphere, forcing the variation in the transition region, as is also shown in panel a of Figure 3.

In panel b of Figure 3, different distributions of dz are plotted for different $d\tau$ values. It can be seen that decreasing $d\tau$ implies increasing the resolution in the corona, as discussed above. In addition, a different number of points have been used in each of the cases. Increasing this number allows to sample a larger area of the transition region, at the cost of increasing the computational time. In this way, an agreement has to be reached between the number of points to be used and the extent of maximum spatial resolution. For all of these cases, the maximum resolution (minimum dz) is located over the transition region as it can be seen with the help of panel c of Figure 3.

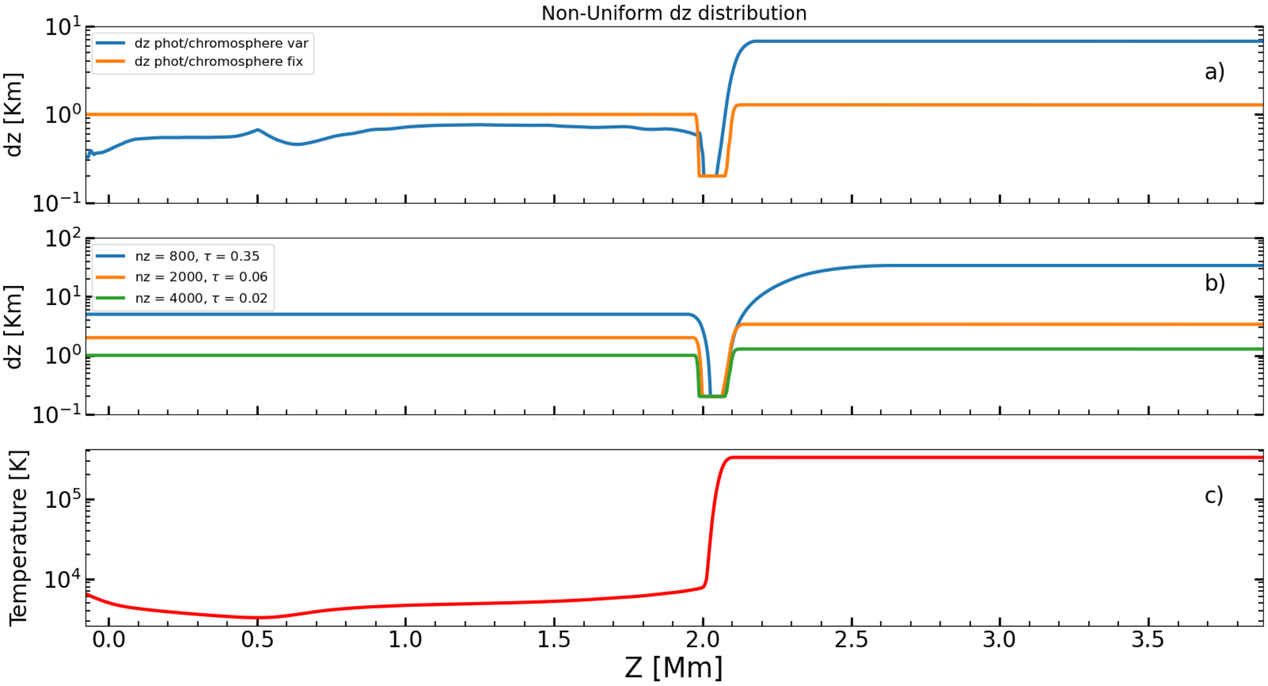

Figure 3: Panel a) shows two non-uniform dz distributions in the case of using a constant and variable spatial step throughout the photosphere and chromosphere. Panel b) illustrates the effect of having a different number of points (nz) used which implies a variation in the $d\tau$ value. The last panel shows the temperature distribution for a better comparison between the regions with the larger temperature gradients and the dz distributions computed in panels a) and b).

3.3 Spatial Derivatives

One of the essential aspects to be implemented in the code is the numerical scheme for the calculation of spatial derivatives. Currently, MANCHA3D allows the use of two different numerical schemes: a centered, fourth order accurate, explicit finite difference scheme using five grid points (Vögler et al., 2005); and a six order accurate explicit scheme using 10 grid points (Nordlund & Galsgaard, 1995). Both schemes employ a three-dimensional Cartesian grid discretization of the spatial domain with a constant spatial step in each dimension. These spatial derivatives are applied independently to each of the spatial coordinates. This makes it possible to easily implement a new numerical scheme that only affects one of the spatial dimensions, as it is the case we are interested in.

Developing an analytical expression of a numerical scheme for the non-uniform spatial derivatives calculation is not trivial. Such expressions are often complex, and there are situations where it is not even possible to find an analytical solution, depending on the order of accuracy required. This is why we have chosen to implement a flexible method for calculating derivatives of different orders and with a given order of accuracy. The basis of this method relies on Taylor series expansions, as discussed in Taylor (2016). The higher the number of points used, the higher the order of accuracy of the numerical scheme. As an example, the procedure for the calculation of the numerical scheme using a five-point stencil is described below. The idea is to compute a set of coefficients $\{A, B, C, D, E\}$ for each of the grid points used:

$$\left(\frac{\partial f}{\partial z}\right)_i = A\, f_{i-2} + B\, f_{i-1} + C\, f_i + D\, f_{i+1} + E\, f_{i+2}\,, \tag{8}$$

where f_{i-2} denotes a variable at a point $f(z_{i-2}) = f(z_i - (dz_{i-2} + dz_{i-1}))$, being $dz_i = z_{i+1} - z_i$ the non-equidistant step. The same notation follows for rest of the terms. It is possible to develop by Taylor series each of these f terms around z_i as follows:

$$\begin{aligned}
&f(z_{i-2}) = f(z_i) - f'(z_i)(dz_{i-2} + dz_{i-1}) + \frac{f''(z_i)}{2}(dz_{i-2} + dz_{i-1})^2 - \frac{f'''(z_i)}{6}(dz_{i-2} + dz_{i-1})^3 + \dots\,,\\
&f(z_{i-1}) = f(z_i) - f'(z_i)dz_{i-1} + \frac{f''(z_i)}{2}dz_{i-1}^2 - \frac{f'''(z_i)}{6}dz_{i-1}^3 + \dots\,,\\
&f(z_i) = f(z_i)\,,\\
&f(z_{i+1}) = f(z_i) + f'(z_i)dz_i + \frac{f''(z_i)}{2}dz_i^2 + \frac{f'''(z_i)}{6}dz_i^3 + \dots\,,\\
&f(z_{i+2}) = f(z_i) + f'(z_i)(dz_i + dz_{i+1}) + \frac{f''(z_i)}{2}(dz_i + dz_{i+1})^2 + \frac{f'''(z_i)}{6}(dz_i + dz_{i+1})^3 + \dots\,.
\end{aligned} \tag{9}$$

As in this case we are using five grid points, it will be necessary to expand the Taylor series to the fourth-order derivative term. This is because we want to calculate five coefficients and this requires the use of five independent equations (from zero to fourth order). Taking into account the expression (8), a set of coefficients $\{A, B, C, D, E\}$ which nullify the terms $f(z_i)$, $f''(z_i)$, $f'''(z_i)$ and $f''''(z_i)$ is searched, as in this case we are interested in the first order derivative. In such a way, the following matrix equation can be constructed:

$$\begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ -(dz_{i-2}+dz_{i-1}) & -dz_{i-1} & 0 & dz_i & (dz_i+dz_{i+1}) \\ (dz_{i-2}+dz_{i-1})^2 & dz_{i-1}^2 & 0 & dz_i^2 & (dz_i+dz_{i+1})^2 \\ -(dz_{i-2}+dz_{i-1})^3 & -dz_{i-1}^3 & 0 & dz_i^3 & (dz_i+dz_{i+1})^3 \\ (dz_{i-2}+dz_{i-1})^4 & dz_{i-1}^4 & 0 & dz_i^4 & (dz_i+dz_{i+1})^4 \end{pmatrix} \begin{pmatrix} A \\ B \\ C \\ D \\ E \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} . \tag{10}$$

By calculating the inverse matrix of the left-hand matrix of the above expression, it is possible to directly obtain the required coefficients. This procedure can be performed for a stencil with either an even or odd number of points, as well as for any derivative order. In the case of using an even number of points (as is the case of the diffusivity module, see Section 3.4), the generated matrix will not be equally symmetric to that of expression (10). Thus it is necessary to indicate the direction in which an extra point is taken, i.e. forward or backward, while the rest of the procedure is the same.

It is possible to show that in the case of using a uniform grid with $dz_i = h$, the known analytical expressions are recovered. Applying this to the previous case, we would have to obtain the same fourth-order scheme used by MANCHA3D . In this way, the matrix system results:

$$\begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ -2 & -1 & 0 & 1 & 2 \\ 4 & 1 & 0 & 1 & 4 \\ -8 & -1 & 0 & 1 & 8 \\ 16 & 1 & 0 & 1 & 16 \end{pmatrix} \begin{pmatrix} A \\ B \\ C \\ D \\ E \end{pmatrix} = \frac{1}{h} \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} . \tag{11}$$

Computing the inverse matrix of the left-hand side matrix and multiplying it by the right-hand side term, this set of coefficients are obtained:

$$A = 1/12 \; ; \quad B = -8/12 \; ; \quad C = 0 \; ; \quad D = 8/12 \; ; \quad E = -1/12 \; , \tag{12}$$

which are exactly the same coefficients of the numerical scheme implemented in the original code.

3.3.1 Derivative in the borders

The calculation of the derivatives at the borders of the domain requires a special treatment. This is because the number of points on either side of the point over which the derivative is to be evaluated is not the same. MANCHA3D currently uses a specific numerical scheme according to the number of points that can be used. Exactly at the border of the domain a forward or backward formula of first order will be used. In any other point close to the border that does not contain enough points on each side to apply the chosen numerical scheme, a centered derivative of second order of accuracy will be used (see Figure 4 (left)).

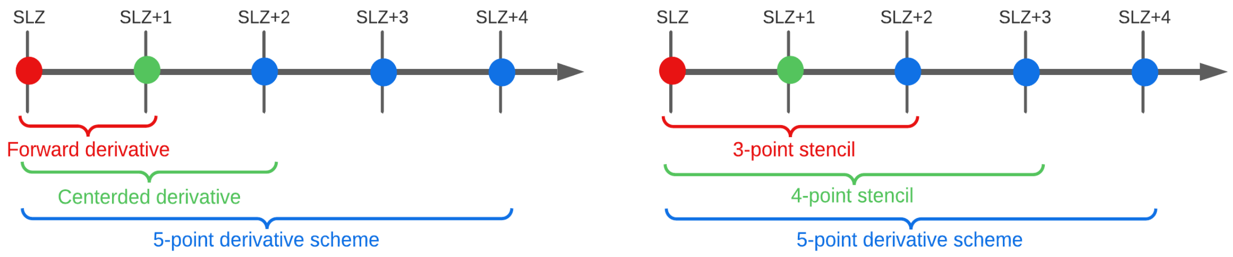

Figure 4: Left: current numerical schemes used by MANCHA3D in the border of the domain. Right: specific numerical schemes by computing a new set of coefficients with a different number of points in each case. In both plots it is represented the bottom part of the domain in z, being SLZ the initial point.

We can compute a specific numerical scheme for each situation by determining a new set of coefficients following a similar procedure as described above. For instance, if we use a numerical scheme with five grid points, the derivative at the border of the domain would be calculated by considering just three grid points forwards or backwards, depending whether it is at the bottom or the top of the domain. The adjacent point will use four grid points which leads to a new numerical scheme and so on. This idea is what it is illustrated in Figure 4 (right) particularised to the bottom z domain.

Analysing the behaviour of these derivatives, it has been seen that the results are less stable than those obtained through the fixed numerical schemes currently used by MANCHA3D . The use of these schemes produces more ringing in these regions, which is undesirable. This is why we keep using the simple numerical schemes with known analytical expressions.

3.4 Diffusivity

In the solar atmosphere, different processes take place on a variety of spatial scales. In some cases, these phenomena have very small characteristic length scales, such as viscosity, magnetic diffusivity or thermal conductivity. To solve these scales in numerical simulations is still a computational challenge today. Therefore, an alternative strategy is used: to replace these physical terms by artificial quantities that simulate their effect. This philosophy is followed by MANCHA3D , solving the equations (1) where artificial diffusivity terms were introduced. These quantities have their physical equivalents in the momentum, energy and induction equations, but not in the continuity equation. The purpose of these terms is to give stability to the code against small-scale oscillations, which are not resolved by the spatial resolution of the grid.

Following a methodology similar to that proposed by Stein & Nordlund (1998), Caunt & Korpi (2001), Vögler et al. (2005), three artificial diffusivity terms are considered: constant, variable (hyper) and shocks. For a certain variable $\mathbf{u}$ (scalar or vector) and in a spatial direction i (x, y, z) the total diffusivity can be written as a sum of these three terms:

$$\nu_i(\mathbf{u}) = \nu_i^{cte}(\mathbf{u}) + \nu_i^{hyp}(\mathbf{u}) + \nu_i^{shk}(\mathbf{u}) \ . \tag{13}$$

Constant diffusivity means a constant in time diffusion coefficient and it is defined as:

$$\nu_i^{cte}(\mathbf{u}) = c^{cte}(\mathbf{u})\,(c_{s0} + v_{A0})\,\Delta x_i\,F^{cte}(x,y,z) \ , \tag{14}$$

where $c^{cte}(\mathbf{u})$ is a coefficient of proportionality which is set in the control file and it can be different for each of the primary variables, increasing or decreasing the diffusivity intensity. c_{s0} and v_{A0} are the equilibrium sound and Alfvén speeds, respectively. The function $F^{cte}(x,y,z)$ is a spatial variable profile which can be used to increase diffusivity in a specific region. Δx_i represents the constant spatial step in the i direction.

As this coefficient is defined independently for each spatial direction, the non-uniform case implementation becomes simple. Modifying only the spatial step on the vertical component z by the corresponding value on the non-uniform grid, the new expression for the constant diffusivity is obtained:

$$\boxed{\nu_z^{cte}(\mathbf{u}) = c^{cte}(\mathbf{u})\,(c_{s0} + v_{A0})\,dz_{\text{nonu}}\;F^{cte}(x,y,z) \ .} \tag{15}$$

The sound and Alfén speeds and the $F^{cte}(x,y,z)$ profile are already evaluated on the non-uniform mesh, so they do not require any change when computing this constant diffusivity.

An analogous procedure has been followed for the **shock diffusivity** term. In this case, the diffusivity is applied where there are strong gradients with sudden variations in the velocity between nearby points. It is only non-zero in those regions where there are converging flows, being defined as:

$$\begin{aligned} &\nu_i^{\text{shk}}(\mathbf{u}) = c^{\text{shk}}(\mathbf{u})\,(\Delta x_i)^2\,|\nabla\cdot\mathbf{v}| && \text{if} \quad \nabla\cdot\mathbf{v} < 0 \ , \\ &\nu_i^{\text{shk}}(\mathbf{u}) = 0 && \text{if} \quad \nabla\cdot\mathbf{v} \geq 0 \ , \end{aligned} \tag{16}$$

where $c^{shk}(\mathbf{u})$ is also a coefficient of proportionality which is set through the control file for each variable $\mathbf{u}$. This term can be modified to the non-uniform case by just changing the spatial step, same as in the constant diffusivity term. The divergence of the velocity is computed using the numerical schemes described in the previous section, so no further changes are necessary. Therefore, the shock diffusivity term in the z direction can be written as:

$$\boxed{\begin{aligned} &\nu_z^{\text{shk}}(\mathbf{u}) = c^{\text{shk}}(\mathbf{u})\,(dz_{\text{nonu}}\ )^2\,|\nabla\cdot\mathbf{v}| && \text{if} \quad \nabla\cdot\mathbf{v} < 0 \ , \\ &\nu_z^{\text{shk}}(\mathbf{u}) = 0 && \text{if} \quad \nabla\cdot\mathbf{v} \geq 0 \ . \end{aligned}} \tag{17}$$

Variable (Hyper) diffusivity shows a higher complexity to adapt it to the non-uniform configuration than the two previous cases. The idea of this term is to apply a higher diffusivity in the regions of higher fluctuation, damping them more. To detect these regions, this term is defined proportional to the ratio of the third derivative to the first derivative of the variable $\mathbf{u}$, so the diffusivity may be different for each quantity $\mathbf{u}$. In such a way, it is defined as:

$$\nu_i^{\text{hyp}}\left(\mathbf{u}\right) = c^{\text{hyp}}\left(\mathbf{u}\right)\left(v + c_{\text{S}} + v_{\text{A}}\right)\Delta x_i \frac{\max_3 \Delta_i^3 \mathbf{u}}{\max_3 \Delta_i^1 \mathbf{u}} F^{\text{hyp}}(x,y,z)\ , \tag{18}$$

where again c^{hyp} is a scaling factor and $F^{\text{hyp}}(x,y,z)$ has the same purpose as $F^{cte}(x,y,z)$. The term $(v + c_{\text{s}} + v_{\text{A}})$ is computed from the current flow, sound and Alfvén velocities and thus it will change from one time instant to another. As in the previous cases, Δz must be changed by dz_{nonu} to take into account the non-uniform distribution. The ratio $(\max_3 \Delta_i^3 \mathbf{u} / \max_3 \Delta_i^1 \mathbf{u})$ represent a proportionality to the ratio between the third and first derivative, but they are computed as absolute third (Δ_i^3) and first (Δ_i^1) differences. They can be calculated as:

$$\begin{aligned}\left(\Delta_i^3 u\right)_{j+1/2} &= \left|3\left(u_{j+1} - u_j\right) - \left(u_{j+2} - u_{j-1}\right)\right|\ ,\\ \left(\Delta_i^1 u\right)_{j+1/2} &= \left|u_{j+1} - u_j\right|\ ,\end{aligned} \tag{19}$$

being the j-th value a symmetric combination of the terms $\left(\Delta_i^3 u\right)_{j-1/2}$ and $\left(\Delta_i^3 u\right)_{j+1/2}$. The $\max_3$ indicates the maximum over 3 adjacent points.

To generalise this diffusivity term to the non-uniform case, it will be necessary to compute the absolutes third and first differences taking into account that the $\mathbf{u}$ variable is not uniformly distributed. Thus a new set of coefficients must be calculated using the methodology described in the previous section. In particular, two set of coefficients are needed for both third and first absolute differences, one for point $j - 1/2$ point and one for $j + 1/2$.

When computing these coefficients by using the non-uniform derivative scheme, we are not calculating the coefficients of the absolute differences, but we actually obtain these coefficients divided by the spatial step at each point. Therefore, the set of coefficients computed from the matrix inversion are multiplied by the spatial step at each j point. This multiplication is performed using the maximum dz_{nonu} in the 4-point stencil to increase the diffusivity in those regions where dz_{nonu} changes (mainly in the transition region). The same strategy is done for the first absolute difference, where it is only used 2 points. As an example, this is how the absolute third difference terms look like for the z direction:

$$\boxed{\begin{aligned}\left(\Delta_z^3 u\right)_{j+1/2} &= \left|A_{\text{fwd}}\, u_{j-1} + B_{\text{fwd}}\, u_j + C_{\text{fwd}}\, u_{j+1} + D_{\text{fwd}}\, u_{j+2}\right|\ ,\\ \left(\Delta_z^3 u\right)_{j-1/2} &= \left|A_{\text{bwd}}\, u_{j-2} + B_{\text{bwd}}\, u_{j-1} + C_{\text{bwd}}\, u_j + D_{\text{bwd}}\, u_{j+1}\right|\ ,\end{aligned}} \tag{20}$$

where the set of coefficients are $\{A, B, C, D\} = \{A', B', C', D'\} \times dz_{\text{nonu}}^3$, being the primed coefficients the resulting values of the derivative calculation. The fwd and bwd notation stands for forward and backward in order to differentiate if we are computing the derivative taking into account the forward $(j + 1/2)$ or the backward $(j - 1/2)$ point with respect j. For the first order differences, the coefficients are only multiplied by dz_{nonu} since the power depends on the derivative order. Using these new expressions, equation (18) can still be used by simply changing the spatial step to dz_{nonu} when using the vertical direction.

3.5 Filtering

In some simulations, it is not possible to apply high diffusivity values, as in the case of wave propagation. This is because the diffusivity affects the amplitude of the waves, resulting in less accurate results. By reducing this artificial term, small-scale high-frequency oscillations start to develop, which amplify over time, making the simulation unstable. An alternative mechanism is therefore used to damp this type of fluctuations: the filtering. Following the method described by Parchevsky & Kosovichev (2007), a sixth-order digital filter is applied in order to remove the unresolved short-wavelength components in this way:

$$u^{\text{filt}}\,(x_j) = u(x_j) - \sum_{m=-3}^{3} d_{\text{m}} u(x_j + m\Delta x) \tag{21}$$

where u is a variable before filtering and u_{filt} is after filtering. The coefficients d_{m} are computed by the Fourier analysis, assuming a constant spatial step; they are symmetric and take the following values:

$$d_0 = \frac{5}{16}, \quad d_1 = d_{-1} = -\frac{15}{64}, \quad d_2 = d_{-2} = \frac{3}{32}, \quad d_3 = d_{-3} = -\frac{1}{64}. \tag{22}$$

This filtering function can be applied independently to each of the three spatial directions. This is an advantage when implementing a non-uniform filtering only in the vertical direction. Instead of computing a new set of d_m coefficients for each point over the non-uniform grid, the values of u can be interpolated into a uniform mesh keeping the same d_m coefficients. This uniform grid is constructed using the minimum dz_{nonu} within each 7-point stencil. The values of u in this new mesh are obtained through a linear interpolation:

$$\begin{aligned} &u^{\text{interp}}\,(z_{j+m}) = \alpha_m\, u(z_{j+m}) + \beta_m\, u(z_{j+m+1}), \;\; \text{if} \;\; m = -3, -2, -1\;, \\ &u^{\text{interp}}_{z_j} \; = u_{z_j}, \;\; \text{if} \;\; m = 0\;, \\ &u^{\text{interp}}\,(z_{j+m}) = \alpha_m\, u(z_{j+m-1}) + \beta_m\, u(z_{j+m}), \;\; \text{if} \;\; m = 1, 2, 3\;, \end{aligned} \tag{23}$$

where α_m and β_m are the interpolation coefficients which satisfies the relation $\alpha_m + \beta_m = 1$ and only depend on the distribution of the z array, independently of the variable u to filter. This set of coefficients can be computed as:

$$\begin{aligned} \alpha_m &= \frac{z_{j+m+1} - z^{\text{unif}}_{j+m}}{z_{j+m+1} - z_{j+m}}, \;\; \text{if} \;\; m = -3, -2, -1\;, \\ \alpha_m &= \frac{z_{j+m} - z^{\text{unif}}_{j+m}}{z_{j+m} - z_{j+m-1}}, \;\; \text{if} \quad m = 1, 2, 3\;, \end{aligned} \tag{24}$$

where z^{unif} is the local uniform grid which is created from the minimum dz_{nonu} . In this way, the same filtering expression can be applied to these new values:

$$u^{\text{filt}}\left(z_j\right) = u(z_j) - \sum_{m=-3}^{3} d_{\text{m}} u^{\text{interp}}\left(z_{j+m}\right) , \tag{25}$$

These two steps can be merged into one, reducing the number of operations to be calculated and preserving the same structure as in the uniform filtering. To this end, a new set of coefficients is defined to take into account the interpolation as well as the uniform filtering with the d_m coefficients:

$$\boxed{u^{\text{filt}}\left(z_j\right) = u(z_j) - \sum_{m=-3}^{3} C_{\text{m}} u(z_{j+m}) ,} \tag{26}$$

where the new C_m coefficients have been defined as:

$$\boxed{\begin{array}{ll} C_m = \alpha_m \, d_m, & \text{if } \; m = -3 \; , \\ C_m = \alpha_m \, d_m + (1-\alpha_{m-1}) \, d_{m-1}, & \text{if } \; m = -2, -1 \; , \\ C_m = \alpha_{m+1} \, d_{m+1} + d_m + (1-\alpha_{m-1}) \, d_{m-1}, & \text{if } \; m = 0 \; , \\ C_m = \alpha_{m+1} \, d_{m+1} + (1-\alpha_m) \, d_m, & \text{if } \; m = 1, 2 \; , \\ C_m = (1-\alpha_m) \, d_m, & \text{if } \; m = 3 \; . \end{array}} \tag{27}$$

3.6 Time step

The time step used by MANCHA3D is the minimum value of three terms: the advective and diffusive time step, and the time step set by the user in the control file. The advective time step for the MHD non-linear system of equations is a modified CFL condition, which depends on the characteristic speeds. It can be approximated by:

$$\Delta t_{\text{adv}} = c_{\text{adv}} \left[\frac{1}{1/\Delta x^2 + 1/\Delta y^2 + 1/\Delta z^2} \right]^{1/2} \frac{1}{v_{\text{max}}} , \tag{28}$$

where c_{adv} is a proportionality coefficient and v_{max} is the maximum value of the sound or Alfvén speeds or the computed velocity perturbation in each of the three spatial directions. In the non-uniform configuration, the computation of the time step should be modified. As we have a specific value of dz at each point in space, we have to evaluate the above expression locally. In addition, we will have to differentiate this expression for each of the velocities, so that we have a value of the time step for each point in space and for each of the velocities. If these changes are not taken into account, the advective time step would be underestimated, making the time step significantly smaller than necessary. This is because the region with minimum dz_{nonu} does not necessarily have to be located where the maximum velocity is reached. Thus, we compute the time step as:

$$\Delta t_{\mathrm{adv}}^{c_s,v_A,v_i} = c_{\mathrm{adv}} \left[\frac{1}{1/\Delta x^2 + 1/\Delta y^2 + 1/dz_{\mathrm{nonu}}^2} \right]^{1/2} \frac{1}{(c_s, v_A, v_i)} \, , \tag{29}$$

where the sub-index i denotes each of the three spatial directions. In order to ensure that during the whole simulation the CFL condition is verified, the minimum time step of the 5 above values is used:

$$\boxed{\Delta t_{\mathrm{adv}} = \min(\Delta t_{\mathrm{adv}}^{c_s}, \Delta t_{\mathrm{adv}}^{v_A}, \Delta t_{\mathrm{adv}}^{v_i})} \tag{30}$$

The diffusive time step is computed as the minimum diffusion time across the three directions. Besides, it takes into account the time step imposed by the Ohmic, ambipolar, Hall and battery diffusion terms. For these four terms, it is only necessary to consider the minimum dz_{nonu} in the whole domain to implement them in the non-uniform configuration. The diffusion time through each spatial direction are defined as:

$$\Delta t_{\mathrm{diff}} = c_{\mathrm{diff}} \min \left(\frac{\Delta x^2}{\nu_x}, \frac{\Delta y^2}{\nu_y}, \frac{\Delta z^2}{\nu_z} \right) \, , \tag{31}$$

where $\nu_{x,y,z}$ are the artificial diffusion coefficients defined in (13) and c_{diff} is a scale factor. The same strategy has been followed as for the advective time step. In this case, it is only necessary to evaluate locally the diffusivity time step in the vertical direction ($dz_{\mathrm{nonu}}^2/\nu_z$). Therefore, for the non-uniform configuration, the diffusion time step can be set as follows:

$$\boxed{\Delta t_{\mathrm{diff}} = c_{\mathrm{diff}} \min \left(\frac{\Delta x^2}{\nu_x}, \frac{\Delta y^2}{\nu_y}, \min \left(\frac{dz_{\mathrm{nonu}}^2}{\nu_z} \right) \right) \, .} \tag{32}$$

3.7 PML

The Perfect Matched Layer (PML) method is used by MANCHA3D on all boundaries of the simulation domain (Berenger (1994); Hu (1996); Parchevsky & Kosovichev (2007)). This technique is used to absorb perturbation at boundaries using a certain number of grid points; for the waves simulations it is used to prevent spurious wave reflections. The MHD equations in the borders are modified to introduce new terms which will damp the perturbation that reach the boundaries in each direction. These terms depend on the following damping coefficients:

$$\begin{aligned} \sigma_x &= a \frac{c_{S0} + v_{A0}}{\Delta x} \left(\frac{x - x_{\mathrm{PML}}}{x_{\mathrm{PML}}} \right)^2 \, , \\ \sigma_y &= b \frac{c_{S0} + v_{A0}}{\Delta y} \left(\frac{y - y_{\mathrm{PML}}}{y_{\mathrm{PML}}} \right)^2 \, , \\ \sigma_z &= c \frac{c_{S0} + v_{A0}}{\Delta z} \left(\frac{z - z_{\mathrm{PML}}}{z_{\mathrm{PML}}} \right)^2 \, , \end{aligned} \tag{33}$$

where a, b and c are constants controlling the damping amplitude, and x_{PML}, y_{PML} and z_{PML} are the thickness of the PML domain in each spatial direction. This is assuming a uniform grid in which all points of the PML grid are spaced by an amount of Δz. In a rigorous way, the non-uniform configuration should consider that all the points of the grid are separated by a varying dz. In wave propagation simulations we are interested in using PML at the upper edge. As seen in section 3.2, the value of dz in the solar corona is essentially uniform. In some configurations it may slightly change because of adjustments in the grid generation so that the total extent of the atmosphere remains unaffected. However, in none of the configurations used, the difference between the dz in the corona is greater than 5% and it can be considered almost uniform. Therefore, the damping coefficient in the vertical direction at the top has been set as the maximum dz_{nonu} of the corona:

$$\boxed{\sigma_z = c\frac{c_{S0} + v_{A0}}{\max(dz_{\mathrm{nonu}}\ )}\left(\frac{z - z_{\mathrm{PML}}}{z_{\mathrm{PML}}}\right)^2 .} \tag{34}$$

The same philosophy can be applied to the bottom part of the domain. As a constant dz is used in the photosphere and chromosphere (as discussed in Section 3.2), this value can be used for the PML coefficient in that region of the domain.

3.8 Code implementation

All modifications have to be implemented following the parallel design of MANCHA3D to preserve back compatibility in the code. An important feature is the reading and writing of the data which are also done in parallel from h5 files. MANCHA3D reads the background model generated from a Python or an IDL script that contains the distribution of magnitudes discussed in Section 3.1 and other general parameters of the simulation such as the number of points in each spatial direction, the spatial step, simulation time, etc. These data are distributed between the different processors following a memory spacing which depends on the subdomain location respect to the whole domain. When writing data to the h5 files, each processor writes their data following the same spacing in memory, so that all the information is reconstructed into a single array. This allows to analyse the results in a simple way for post-processing.

As it has been seen in Section 3.2, the generation of the non-uniform grid is also performed from an external script to MANCHA3D . The information of the non-uniform array in the vertical direction has to be added to the h5 files together with the rest of the magnitudes, as the code is not able to generate this distribution. Besides, this data is included in the h5 output files in order to analyse the results in a more efficient way. In contrast with the rest of the magnitudes, this new dataset is one-dimensional, which means that the distribution among the rest of the processors has to be done in a different way.

When decomposing the total domain of the simulation, each of the subdomains can be identified by three values representing their coordinates over the total domain. Figure 5 shows an example of a 3D domain decomposition, indicating the global coordinates of each subdomain. The values of the non-uniform z array must be distributed along the vertical direction. That is, subdomains with the same global coordinate value in z must contain the same portion of the array, regardless of their x

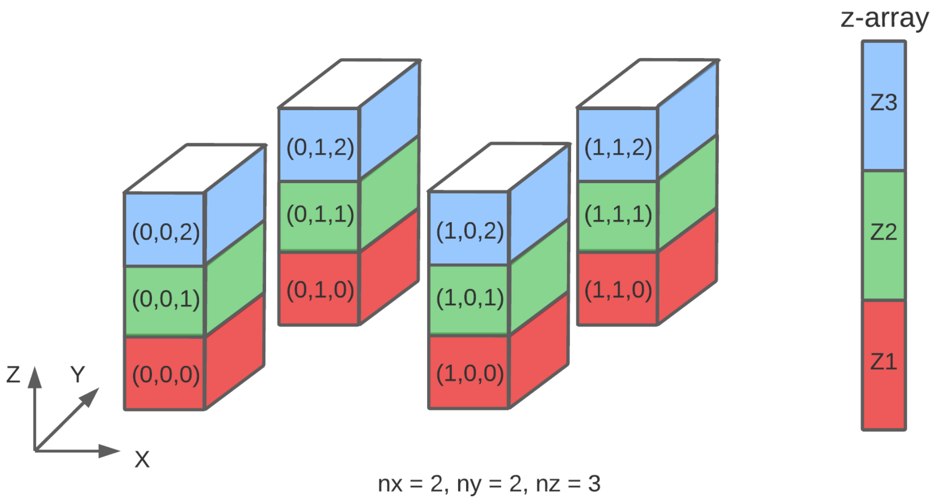

Figure 5: 3D domain decomposition when using nx x ny x nz = 12 processors. Each processor is identified by three coordinate number. The z-array illustrates how data is distributed and sent to each processor according to its z-coordinate (represented by different colors).

and y coordinates, which is also illustrated in Figure 5. Thus, memory spacing must be treated with care when distributing the data during reading.

In addition, the PML module requires each processor to store the complete z-array for some calculations. From each processor, this array has to be reconstructed. To do this, a new MPI communicator has been created that transfers information only along the processors that have the same global coordinates of x and y, i.e. along the vertical direction of each of the columns in Figure 5. In this way each one sends and receives each part of the complete array in z. This is done because it is more efficient than each processor reading a much larger array and then distributing the information to each of the subdomains.

Another important issue is the computational time. The changes have to be implemented in such a way that they do not introduce an excessive number of operations, while keeping the efficiency of the code. The spatial derivatives, diffusivity and filtering modules now require the calculation of a set of coefficients. These coefficients are computationally expensive as they involve the inversion of a matrix during their computation, but their values depend only on the distribution of the non-uniform z grid. Thus, it is possible to compute them only once during the initialisation process. Apart from keeping the number of operations almost invariant during each iteration, the general structure of the code in each of these modules is preserved.

4 Results and Discussion

In this section we present and discuss the results obtained after all the modifications of the previous section have been applied. First, we check that all the changes done have been well implemented by comparing the results with those of the original code. Then, simulations of acoustic and Alfvén waves which propagates along the vertical direction are performed. Different initial configurations have been studied and compared with the results obtained using high resolution uniform meshes. In addition, an analysis of the computational efficiency of the non-uniform configuration with respect to the uniform one has been carried out.

4.1 Acoustic Waves

Acoustic waves are longitudinal waves, so the velocity vector **u** is parallel to the direction of propagation **k**. Our interest is to study the variation with height of the velocity field, so we will only analyse waves that propagate along the vertical direction z in 1D. These waves start at the bottom part of the atmosphere as a perturbation to the equilibrium conditions. In the control file, the initial amplitude and period of the wave can be selected. We have studied waves with periods of 50 and 180 s and initial amplitudes of 0.01 and 10 m/s to see how the code behaves with this type of waves.

We differentiate the wave propagation between two regimes: linear and non-linear (shock formation). The studied regime will depend on the initial conditions of period and velocity of the perturbation selected. In both cases we are using an atmosphere stratified by gravity and with a non-uniform temperature distribution (same as in Figure 1). There is no analytical solution for acoustic waves under these conditions. Nevertheless, the solutions of the linear case can be compared with the analytical expression for an acoustic wave in an isothermal medium stratified by gravity. This expression for a wave, that propagates along the vertical direction, is known from Ferraro & Plumpton (1958). The vertical velocity in this situation for a wave of frequency $\omega = 2\pi/P$ is given by:

$$v_z(z,t) = V_0\, e^{z/2H_0} \sin\left[\left(\frac{\sqrt{\omega^2-\omega_c^2}}{c_s}\right) z + \omega\, t\right] , \tag{35}$$

where $\omega_c = \gamma g/(2\, c_s)$ is the cut-off frequency, H_0 is the pressure scale height and V_0 is initial amplitude of the wave. From this expression, it can be seen that the amplitude of the waves increases with height, as long as the frequency of the wave is above the cut-off frequency.

4.1.1 Code Verification

Firstly, it has been confirmed that all the modifications introduced work correctly. For this purpose, the same uniform grid has been used, applied on both configurations of the code: the uniform one, i.e. the original code; and the non-uniform one, after all the changes of the previous section have been applied. In this way, differences introduced by the new methods can be detected.

In order to check this, a sample of 1D acoustic waves that propagate along the vertical direction has been analysed. A uniform grid with $nz = 800$ points and $dz = 5$ km is used. In this case, we study

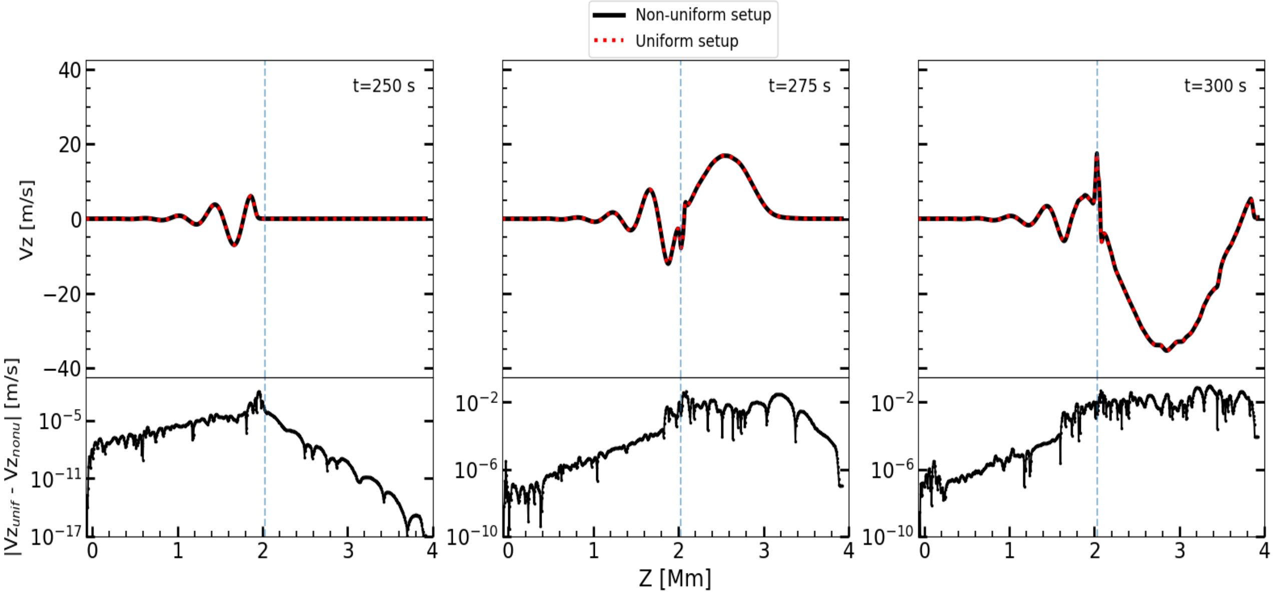

Figure 6: Comparison between the results of the acoustic wave sample in the linear regime with initial amplitude of 0.01 m/s and $P = 50$ s using both setups: uniform (red dashed line) and non-uniform (black solid line). The same grid distribution is used in both configurations. Top panel shows the amplitude wave distribution at different time steps. Bottom panel contains the differences between both solution in logarithmic scale. The vertical dashed blue line illustrates the location of the transition region.

the evolution of a wave in the linear regime with an initial amplitude of 10 m/s and period $P = 50$ s. For the calculation of spatial derivatives, the fourth-order numerical scheme has been used.

Figure 6 shows the results of this comparison for different time instants. Lower panels show that the difference between the two configurations is small in relation to the amplitudes of the waves. At the initial stages, the differences are of numerical origin as a consequence of using different methods. These differences increase as the wave advance, since these numerical errors are accumulated over time. It is possible to see that the differences between the two solutions become more accentuated once the wave crosses the transition region. This is reasonable as this is a region where the calculation of derivatives, filtering and diffusivity is more critical, allowing higher numerical fluctuations.

The resolution used is not sufficient to resolve the behaviour of the waves as they pass through the transition region, so it will be necessary to sample this region with more precision in order to resolve their evolution correctly. Nevertheless, we can say that the modification of all modules for the implementation of the non-uniform grid is correct. In the following sections, different simulations with higher resolutions in uniform and non-uniform configurations will be analysed in order to correctly solve the transition region.

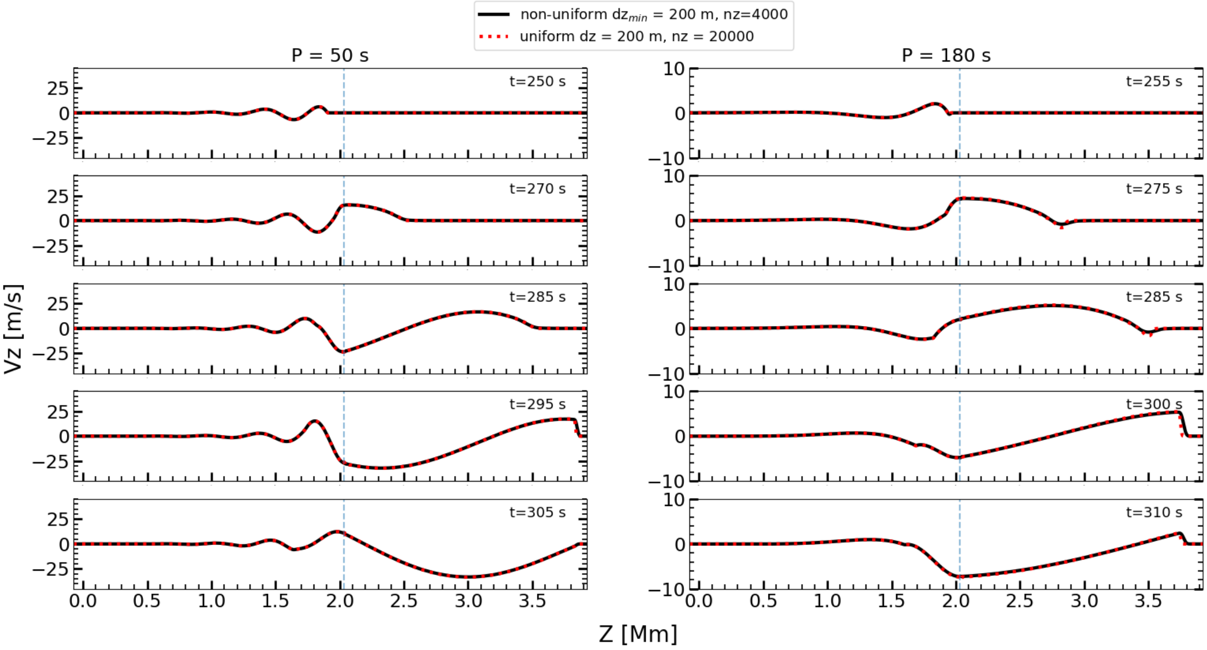

Figure 7: Time evolution of acoustic waves in linear regime for an initial amplitude of 0.01 m/s. Left panels contains the solutions for a wave of P = 50 s, while right panel for a wave with P = 180 s. The numerical results using a non-uniform grid (black solid lines) are compared with the results of an uniform grid in the original code (red dotted lines).

4.1.2 Linear case

Figure 7 shows the time evolution of two acoustic waves of initial amplitude 0.01 m/s and with periods of 50 and 180 s. These results have been obtained using a uniform grid with $nz = 20000$ points and constant $dz = 200$ m for the original code; and a non-uniform grid with $nz = 4000$ points and $dz_{min} = 200$ m for the modified code (equal to the green distribution in Figure 3).

The results are almost identical in the two configurations used for both waves. Although we are using 5 times less points in the non-uniform mesh, the wave behaviour is the same. The resolution used only in the transition region for the non-uniform grid is the same to that used in the uniform mesh for the whole domain. When this region is not well sampled, strong jumps appear in the velocity distribution, modifying the actual behaviour of the waves as they cross the transition region (as it can be seen in Figure 6). The non-uniform grid allows to correctly simulate waves propagating from the photosphere to the solar corona, reducing significantly the computational time required. In particular, computation with the uniform grid takes around 13 hours, while the non-uniform grid with the minimal dz to be same as with the uniform grid ($dz_{\rm min} = 200$ m) takes only 6 hours, i.e. a time reduction of more than 50%.

In this case we study the wave propagation in linear regime. For this, a sufficiently low initial amplitude in velocity has been used, in order to prevent shock formation. Although the initial ampli-

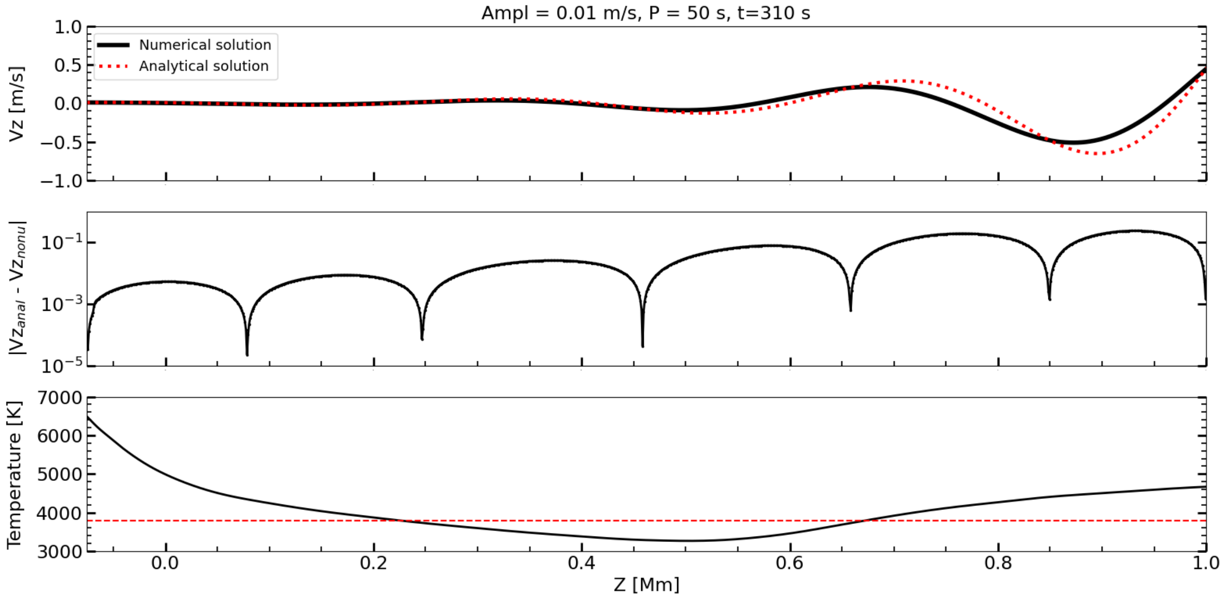

Figure 8: Top panel shows the numerical solution using the non-uniform grid and the analytical solution from equation (35). An initial amplitude of 0.01 m/s and a period of 50 s has been used in both cases. Middle panel contains the absolute difference between the numerical and the analytical solution in logarithmic scale. Bottom panel illustrates the temperature distribution of the photosphere and the lower chromosphere (black line). The red dashed line is the uniform temperature used for the analytical solution.

tudes are the same for both waves, the velocities differs significantly with time due to the difference in periods. This is because shorter period waves tend to increase faster their velocity profile than longer period waves (Ulmschneider, 1971). Once the waves cross the transition region, there is a significant increase in velocity. This is due to the large temperature gradient that present this layer. Nevertheless, in both cases the linear regime is preserved when waves reach the solar corona.

The number of points used in PML is different between the two setups. For the wave with $P = 50$ s, the uniform case uses 500 grid points, while the non-uniform case only 80 grid points were required. This is due to the difference in size of the dz because we want to cover 0.1 Mm of extension in the upper boundary for both simulations to dump the waves. Thus, the number of points needed to absorb the waves at the borders of the domain have been reduced.

We can compare these results with those of analytical equation (35) (see Figure 8). This expression assumes an isothermal atmosphere, so we will limit the comparison up to 1 Mm in the vertical direction. In this region the temperature variation is not as extreme as in the whole atmosphere. We have used a uniform temperature value of 3800 K since it is an intermediate value to the one in the region considered. It can be seen that the analytical solution reproduces the results till the photosphere (about 0.5 Mm). After that, the differences between both solutions start to increase. This difference is explained because the numerical solution does not have a uniform temperature distribution. It can

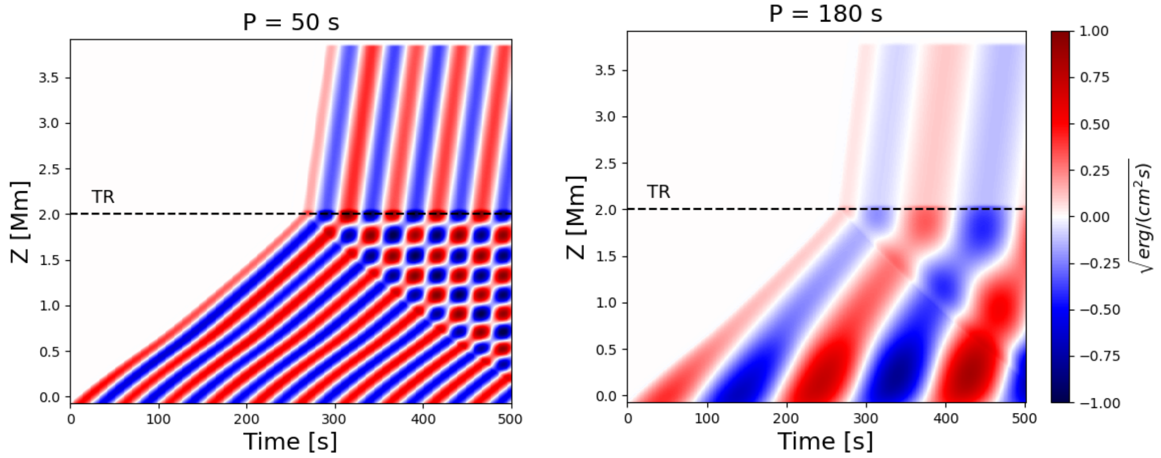

Figure 9: Time-distance diagram of the longitudinal velocity showing the complete evolution of the perturbation in the whole domain. The velocity is scaled by a factor $\sqrt{\rho_0 c_{s0}}$ which gives an idea of the energy carried out by the waves. Left panel shows the evolution of the $P = 50$ s wave and the right panel is for the $P = 180$ s wave. The black dotted line shows the location of the transition region.

be seen that the analytical solution is faster than the numerical solution after 0.5 Mm, since the fixed value of temperature used is higher than the temperature of the numerical solution in that region. It also explains why the velocity of the analytic solution is higher than the numerical one. When reaching the chromosphere, the opposite happens, so the analytical solution is delayed with respect to the numerical one.

Figure 9 shows the complete time evolution of the longitudinal component of the velocity for both acoustic waves. In this case, the velocity has been scaled by a factor $\sqrt{\rho_0 c_{s0}}$ for a better visualisation (same as used by Santamaria et al. (2015)). This quantity gives an approximation to the energy contained by the waves. The simulations have been run up to $t = 500$ s so several oscillations reach the domain boundary.

The slope of the lines in the figures reflect the propagation speed of the waves. The steeper the slope, the faster they are. It can be seen that the slope is nearly constant until the transition region is reached. In this region, due to the high increase in temperature and the drop in density, the waves increase significantly in speed. It can be seen that the results are perfectly sampled in this region as the transition between the chromosphere and the corona is quite smooth. Hereafter, the waves continue travelling with a constant velocity (due to the uniform coronal temperature) until they reach the edge of the domain where they are absorbed by PML. Furthermore, it is verified that PML works correctly, since it does not produce any type of reflection in the results.

When the wave front reaches the transition region, a wave reflection happens due to the strong change in the properties of the medium. This produces a perturbation which travels in the opposite direction, towards the interior of the solar atmosphere. This perturbation interferes with the waves that continue travelling in the vertical direction, so a clearly visible interference pattern is generated in the wave of $P = 50$ s. This effect is less visible in the $P = 180$ s because a smaller number of reflections are produced due to the longer wave period. In addition, the slope of the reflected waves is constant, so that they all travel at the same speed. The reflection of the waves always happens in the same region regardless of the time instant.

4.1.3 Non-Linear case

For the study of non-linear effects in wave propagation, we use a similar methodology to the previous case. The same grids have been applied as in the linear case. However, the initial amplitudes of the wave have been set to 10 m/s. In this way, the amplitudes are intended to grow sufficiently during the propagation of the wave to form shocks. As we are using the same grid as in the linear case, the variation in computational time is the same of the previous section.

Figure 10 shows the comparison of the time evolution of two waves with periods of 50 and 180 s. It can be seen that the results between both configurations are nearly identical. The regions of shock formation are well resolved on both grids, despite the strong changes in these regions. When crossing the transition region, there is a large jump in velocity. In this region we have the highest grid resolution in the non-uniform grid to properly resolve the strong variations that take place. Thereafter, the shock continues propagating to the boundary of the domain with constant velocity, since the temperature gradient in the corona is zero. PML cells absorb the waves perfectly despite of being more energetic than in the linear case.

The higher frequency wave starts to develop non-linearities around 1.2 Mm, while for the lower frequency wave this occurs at a greater height, close to the transition region (2 Mm). This is the same phenomena observed in the linear case, as shorter period waves tend to form shocks earlier than longer period waves. For this situation there is no analytical expression to verify the results. Given the small differences between the uniform and non-uniform mesh, all modifications can be considered valid for this type of waves.

Figure 11 shows the same time-distance diagram as in the previous section (Figure 9). This allows to analyse the complete temporal evolution of the waves as they pass through the different layers of the atmosphere. The general behaviour is similar to the linear case, except for some differences. The waves travel at constant speed until they develop non-linearities. At this point the slope increases slightly and continues travelling up to the transition region. In this layer a reflection happens, generating a wave that propagates into the solar interior. The amplitude interference pattern shown in the figure differs in structure from the linear case. This is because the wave hitting the transition region is a shock wave, making the shape of the reflected wave sharper than in the linear case.

These waves carry more energy than those studied in the linear case. They can push the transition region each time a shock wave hits it. This phenomenon is visible in the high-frequency wave, since there are more shocks per unit time. From Figure 10, it can be clearly seen that reflections are produced each time at higher layers of the atmosphere. The transition region is moving upward because it is continuously hit by shock waves. This is not observed in the 180 s period wave, as a less number of reflections take place, avoiding the transition region to shift back to the equilibrium state. In both cases it is observed that the PML cells works properly as they correctly absorb waves that reach the boundaries of the domain without generating reflections.

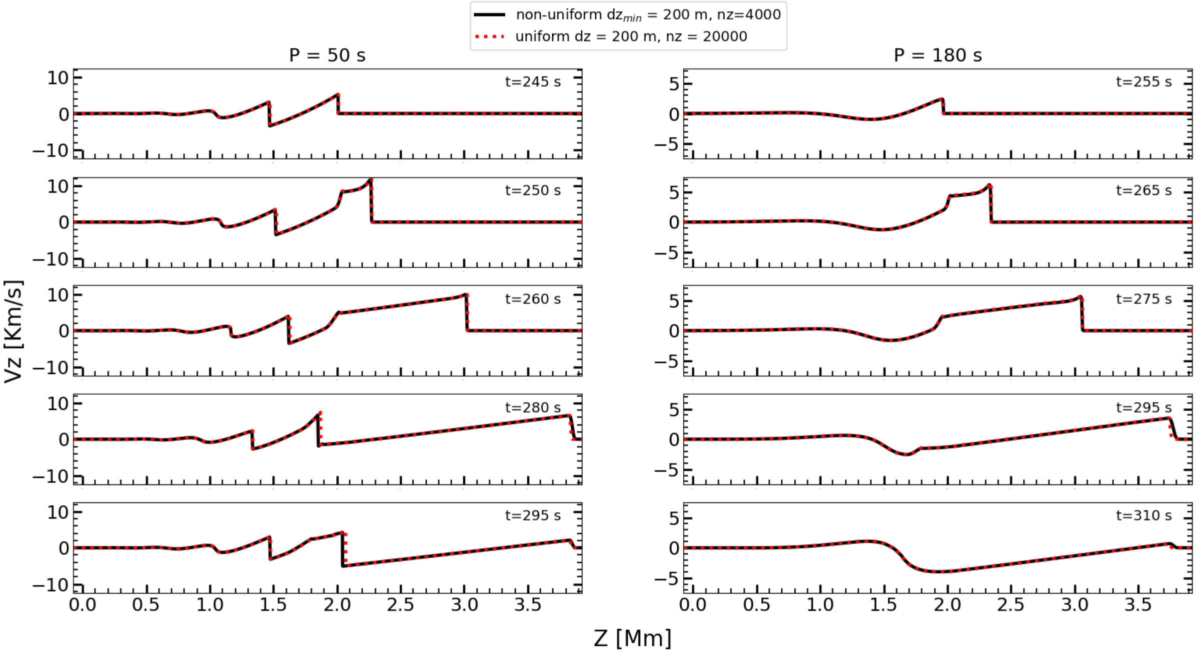

Figure 10: Same as Figure 7 but for waves in non-linear regimen. An initial amplitude of 10 m/s is used.

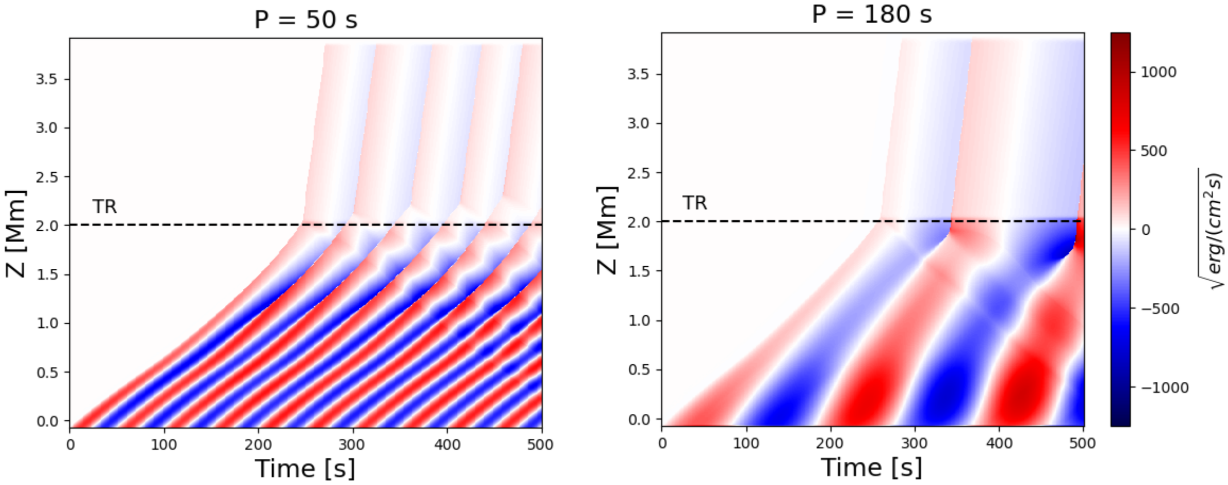

Figure 11: Same as Figure 9 but for waves in non-linear regimen. An initial amplitude of 10 m/s is used.

4.2 Alfvén Waves

Alfvén waves are transverse waves oscillating in the direction perpendicular to the direction of the background magnetic field. In this section we will study the propagation of this type of waves along a uniform magnetic field contained in the vertical direction. In particular, we have that $\mathbf{k} \,||\, \mathbf{B_0}$ and $\mathbf{u} \perp \mathbf{k}$. We can add a constant magnetic field throughout the whole atmosphere, without breaking the hydrostatic equilibrium condition. Besides, a transverse velocity perturbation is added at the bottom of the domain which will be evolved in time.

A comparison of the results between the uniform and non-uniform configurations have been carried out. For the uniform setup, we use a uniform grid with $nz = 800$ and $dz = 5$ Km, while for the non-uniform configuration a non-equidistant grid with $nz = 400$ and $dz_{\text{min}} = 5$ Km have been generated using a constant $dz = 10$ km for the photopshere and chromsophere. This values are quite different from those used in the acoustic wave section. When studying Alfvén waves, the time step is essentially determined by the maximum Alfvén speed ($V_A \propto B/\sqrt{\rho}$). For these simulations, a uniform magnetic field of 200 Gauss has been used. Besides, the value of the density is highly variable, being of the order of 10^{-15} g/cm^3 in the corona. This leads to very large Alfvén velocities, producing a very small time step (as seen in equation (29)). Thus, a low number of grid points has been used in order to be able to carry out the simulations in a reasonable time. Specifically, for the uniform grid it required around 8 hours to reach $t = 500$ s, while for the non-uniform grid it took about 7 hours. This small difference in time will be discussed further.

Figure 12 shows the results of the time-distance plots for both configurations. In this case, we use an initial amplitude of 100 m/s and $P = 100$ s. Left panels represents the longitudinal velocity while right panels shows the transverse velocity. The behaviour in both configurations is quite similar despite of using different grids. The main difference is in the intensity of the energy waves. This is because, in lower layers, the spatial step is the double of that in the uniform case, which means that precision can be lost when generating the initial perturbations. Nevertheless, the same reflections and refractions occur throughout the domain.

Initially there is only velocity in the transverse component of the perturbation. The velocity of the waves is constant, but as they approach the equipartition layer ($c_s = v_A$), they increase gradually their velocity (steeper slope). In this layer, the waves experience a reflection associated with a mode-transformation. Part of the energy of the waves continues propagating into the upper layers, while part of the energy travels in the opposite direction. The interference pattern at the bottom of the figures is a consequence of these reflections. After the equipartition layer, the slope in the diagrams is constant and purely vertical until the edge of the domain. A drop in the wave energy can be observed when crossing the transition region. This is due to the strong drop in density in the transition region, since the velocity of the waves is actually constant. On the other hand, we observe that a small component of the longitudinal velocity has been generated (left panels of Figure 12). The development of this perturbation is a consequence of non-linear effects in the equations. These waves evolve in a similar way to the acoustic waves studied in the previous section. They experience reflections both in the equipartition layer and in the transition region, generating waves that propagate into the solar interior.

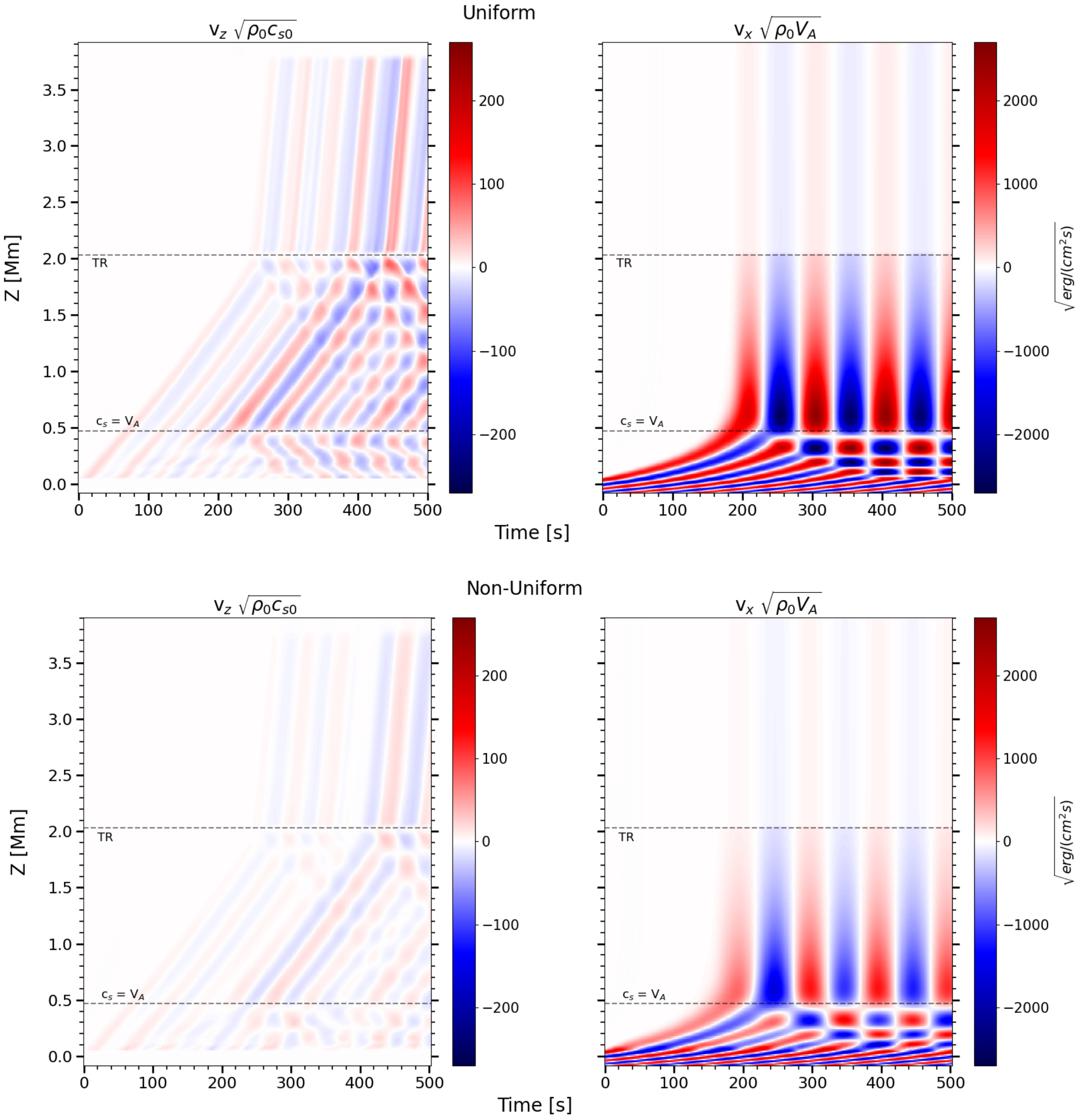

Figure 12: Time-distance plot for Alfvén waves using a uniform and non-uniform grids. Longitudinal (left panels) and transverse (right panels) velocities are scaled by $\sqrt{\rho_0 c_{s0}}$ and $\sqrt{\rho_0 V_A}$, respectively in order to have a better visualization of the results. The upper plots shows the results of using a uniform grid, while the bottom panels shows the same for a non-uniform grid. An initial amplitude of 100 m/s and $P = 100$ s have been used. The lower dotted line represents the equipartition layer and the upper one represents the location of the transition region.

These results can also be compared with theoretical expressions. An analytical solution for an isothermal atmosphere stratified by gravity was developed by Ferraro & Plumpton (1958), and, according to Khomenko et al. (2003), the solution for the horizontal velocity can be written as:

$$v_y(z,t) = i\xi_0\omega\sqrt{J_0^2+Y_0^2}\exp\left[i\left(\omega t + \text{arctang}\,\frac{Y_0}{J_0}\right)\right] \tag{36}$$

where ξ_0 is the initial amplitude and $J_0 = J_0(\eta)$ and $Y_0 = Y_0(\eta)$ are the Bessel function of first and second kind, respectively. $\eta = 2\omega H/V_A$ being V_A the Alfvén speed. This expression will be used to compare the numerical results with the theoretical ones (see Figure 13). We have compared both results around the photosphere, since in this region the temperature does not present high spatial gradients (as we did in the acoustic wave section). It can be seen that initially, the numerical solution (using the non-uniform mesh) and the analytical one agree quite well. From 0.1 Mm both results start to deviate, as a consequence of the limitations of the analytical expression. The temperature and pressure variations of the atmospheric model we are using are not appropriate for this type of expression. Nevertheless, the general evolution and the amplitudes are slightly analogous, becoming equal in the lower part of the domain.

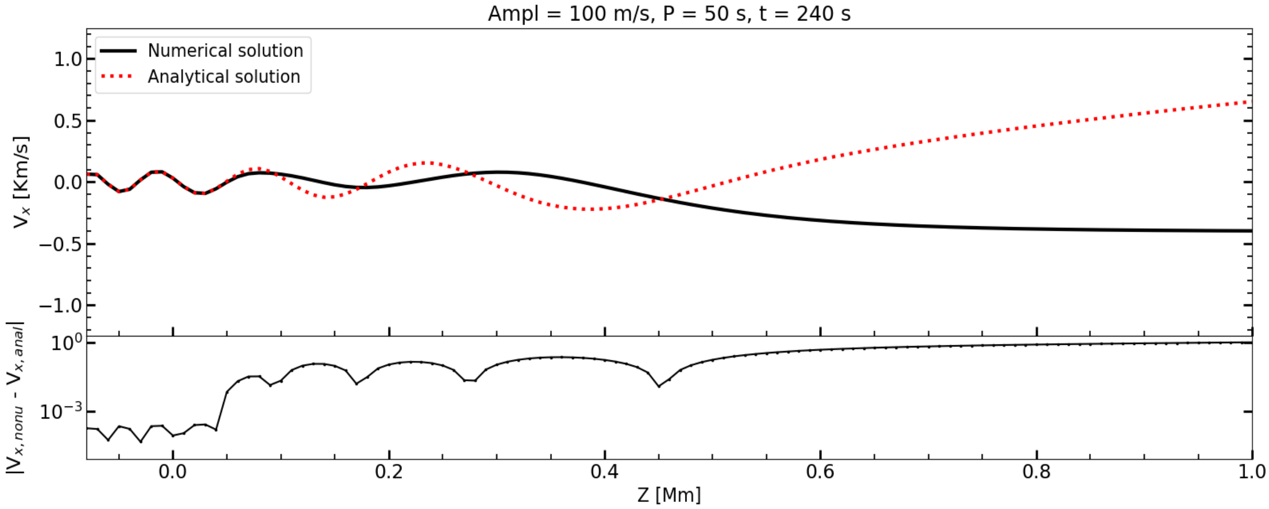

Figure 13: Top panel shows the numerical solution (black solid line) using the non-uniform grid and the analytical solution (red dotted line) from equation (36). An initial amplitude of 100 m/s and a period of 100 s has been used. Bottom panel contains the absolute difference between the numerical and the analytical solution in logarithmic scale.

4.3 Computational time

We perform an analysis of the computational efficiency of the code comparing the non-uniform configuration with the uniform one. For this purpose, uniform meshes of different resolutions have been used to obtain a reference of the computational time of the original code. Then, we measure the computational time for different grid sizes using the non-uniform configuration. For each of them, different values for the minimum dz have been used. All the simulations have been run with 4 processors in a virtual machine of a laptop with Intel Core i7 Processor (4x 2.6 GHz) and 10 GB DDR4 RAM.

Figure 14 shows the results of this analysis. A significant improvement in the computation time used by the non-uniform setup can be observed. This setup allows to sample with high resolution specific regions of the numerical domain, without the need to use such resolution in areas where it is not necessary. In this way, the number of points used is considerably reduced, resulting in an improvement in computational time. We observe that the reduction in computational time is more accentuated when a lower minimum dz is used, because the number of points needed when using a uniform grid grows proportionally to the desired resolution. By using a similar number of points, e.g. just twice, the differences in computation time are not as important, as it is observed from Figure 14 for the case of a minimum resolution of 1 km. The same effect was found in the case of Alfvén waves.

In the previous sections it has been seen that the results of using a uniform and a non-uniform mesh are very similar, as long as the regions with large gradients are sampled with the appropriate resolution. But there is a compromise between the number of points used and the minimum resolution imposed. As seen in Figure 3, using a low number of mesh points with a high resolution in dz is not convenient, since the region of minimum dz would have a very small extension. This might introduce errors in the results. Thus, it is necessary to pay attention to the configuration of the generated grids.

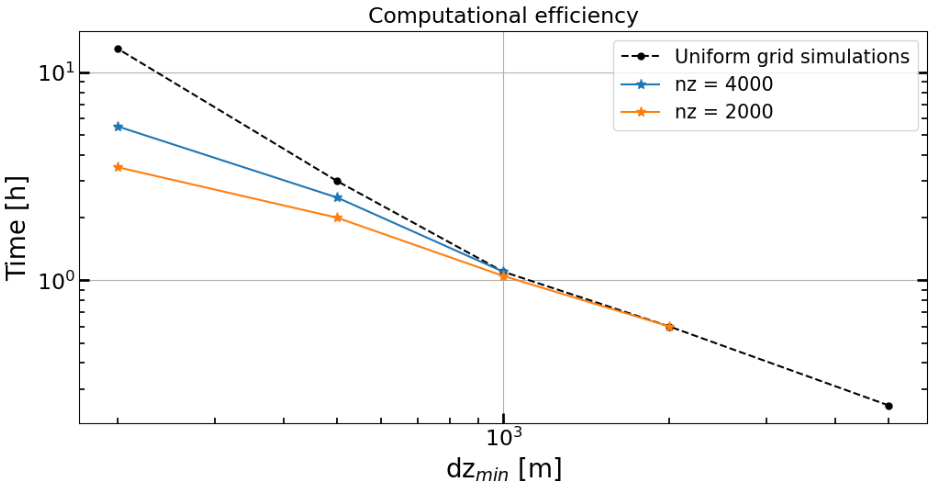

Figure 14: Comparison of the improvement in computation time when using a non-uniform grid (blue and orange lines) and using a uniform grid (black dotted line). The computation time is plotted against the minimum dz used in each distribution, being constant for the uniform grid simulations. The time shown is the time taken by each of the simulations to reach 500 s. Axes are in log scale.

5 Conclusions and Future work

The study of the propagation of MHD waves is fundamental to understand the energy transport and the heating of the different layers of the solar atmosphere. As they propagate to the outer atmosphere, these perturbations experience different processes, such as reflection or refraction, which cause them to change their properties. Such phenomena occur at different scales, requiring high computational power to study them properly. During this work we have used the MANCHA3D code to simulate such waves and study their evolution. In order to improve the computational power required, we have implemented a fixed non-uniform grid along the vertical component. In this way it is possible to solve spatial structures of different scales without the need to use a high resolution over the whole domain.

We have developed an algorithm to generate a non-uniform distribution from a previous uniform model. Our interest has been based on increasing the spatial resolution in areas with large gradients of solar magnitudes. In particular, we have paid special attention to achieve a proper sampling over the transition region, where the temperature gradient and the density drop happen in a small region of the space. At the expense of this, the resolution of the solar corona has been reduced, since it is a region with a uniform temperature distribution and there is no need to sample it carefully. Different grid models can be generated based on the number of grid points provided and the desired minimum dz. An agreement between these two quantities has to be reached, as to sample the transition region accurately, it is necessary to use a sufficient number of points. Otherwise, the benefits of the non-uniform mesh would not be fully exploited.

Several modifications have been carried out to the code for the use of the non-uniform grid in the vertical direction. Among them, we modify the spatial derivatives, diffusivity, filtering, time step calculation and PML. All modifications have been introduced following the same structure as the code in order to ensure compatibility. In addition, the number of numerical operations performed by the code is practically the same. In this way, the computational efficiency is not affected. We have performed multiple tests to verify that all the implementations work properly. In particular, we have studied the propagation of acoustic and Alfvén waves. For this purpose, uniform grids have been used within the non-uniform configuration of the code. This has allowed us to directly compare the results. In all cases, we have seen that the differences introduced are mainly numerical. These differences increase as the simulations evolve, since the errors accumulate over time. Moreover, it has been observed that when crossing the transition region, the differences become more accentuated, as in this region the calculation of derivatives is more critical. These small differences do not introduce significant changes in the results of the simulations, so we have validated all the modifications introduced in the code.

A study of the propagation of acoustic and Alfvén waves from the photosphere until the corona have been carried out. We analyze acoustic wave in linear and non-linear regime. No major differences have been observed between the uniform and non-uniform configurations in both regimes. Phenomena such as wave reflection, refraction or mode-conversion have been observed in all of the test. The regions where this processes take place should have a higher spatial resolution, as is the case of the transition region. For the Alfvén wave similar processes have been observed. In this case it has been observed that waves experience a reflection in the cut-off layer. Besides, the original transverse perturbation gives rise to a longitudinal velocity component as a consequence of non-linear effects. For both types of waves, we have compared the results using the analytical solutions constrained to the solar photosphere. The limitations of such expressions when considering an atmosphere with a non-uniform temperature distribution are shown.

The computational improvement introduced by the use of the non-uniform grid has been analysed. We compared the execution times of simulations of uniform meshes with those of non-uniform meshes of different resolutions. The numerical results in both cases are very similar, but the computational time has been greatly reduced. This is essentially because we can significantly reduce the number of grid points used in the simulations without losing precision due to the resolution used. This is an important step forward in the future of MANCHA3D , as numerical simulations are becoming more expensive due to the computational power required.

With a view to continuing this work in the future, several aspects can be improved and implemented. One of them is the algorithm for generating the non-uniform grid. It would be convenient to implement it in such a way that it is possible to select a larger or smaller range over which to increase the resolution, since now this width can only be controlled according to the minimum dz used. In addition, it would be interesting to implement the generation of the non-uniform mesh within the Fortran code, to avoid using external files for its generation. Throughout this work we have been working with a distribution for the solar atmosphere which is not completely accurate to the actual distribution. Thus, it would be convenient to perform simulations with more realistic values in order to be able to compare the results with observations. On the other hand, we have only focused on the study of wave propagation. The tests performed in this work would have to be carried out on other types of simulations such as Kelvin Helmholtz or Rayler Taylor instabilities or convection simulations.

References

Berenger, J.-P. 1994, Journal of Computational Physics, 114, 185

Caunt, S. E., & Korpi, M. J. 2001, A&A, 369, 706

Felipe, T., Khomenko, E., & Collados, M. 2010, Astrophys. J, 719, 357

Ferraro, C. A., & Plumpton, C. 1958, Astrophys. J, 127, 459

Freytag, B., Steffen, M., Ludwig, H.-G., et al. 2012, Journal of Computational Physics, 231, 919

Grant, S. D. T., Jess, D. B., Zaqarashvili, T. V., et al. 2018, Nature Physics, 14, 480

Gudiksen, B. V., Carlsson, M., Hansteen, V. H., et al. 2011, A&A, 531, A154

Hu, F. Q. 1996, Journal of Computational Physics, 129, 201

Keppens, R., Meliani, Z., van Marle, A. J., et al. 2012, Journal of Computational Physics, 231, 718

Khomenko, E., & Cally, P. S. 2012, The Astrophysical Journal, 746, 68

Khomenko, E., & Collados, M. 2006, Astrophys. J, 653, 739

Khomenko, E. V., Collados, M., & Rubio, L. R. B. 2003, The Astrophysical Journal, 588, 606

Martínez González, M. J., Manso Sainz, R., Asensio Ramos, A., & Bellot Rubio, L. R. 2010, Astrophys. J, Letters, 714, L94

Muller, R., Roudier, T., Vigneau, J., & Auffret, H. 1994, Astron. Astrophys., 283, 232

Nordlund, A., & Galsgaard, K. 1995, Technical Report, Astronomical Observatory, Copenhagen University

Parchevsky, K. V., & Kosovichev, A. G. 2007, Astrophys. J, 666, 547

Santamaria, I. C., Khomenko, E., & Collados, M. 2015, A&A, 577, A70

Stein, R. F., & Nordlund, Å. 1998, Astrophys. J, 499, 914

Taylor, C. R. 2016, Finite Difference Coefficients Calculator, https://web.media.mit.edu/~crtaylor/calculator.html

Tian, H., DeLuca, E., Reeves, K. K., et al. 2014, The Astrophysical Journal, 786, 137

Trujillo Bueno, J., Shchukina, N., & Asensio Ramos, A. 2004, Nature, 430, 326

Ulmschneider, P. 1971, Astron. Astrophys., 12, 297

Vernazza, J. E., Avrett, E. H., & Loeser, R. 1981, Astrophys. JS, 45, 635

Vögler, A. 2004, dissertation, Georg-August-Universität, Göttingen

Vögler, A., Shelyag, S., Schüssler, M., et al. 2005, A&A, 429, 335