Modelling of Magnetohydrodynamic Waves in the Solar Corona

Supervisor: Mijie Shi Michaël Maex Rune Buckinx

March 2020

Abstract

This paper presents a summary of research for a bachelor's thesis on the modelling of magnetohydrodynamic waves in the solar corona using numerical simulations via the open source code PLUTO. Several types of magnetohydrodynamic waves are analysed both theoretically and by simulations in the context of a simple blast wave. Large structures on the solar corona are modelled and their interaction with waves are simulated and discussed.

Contents

List of Figures

1 Introduction

The Sun's corona, or solar corona, is the outer layer of the Sun's atmosphere, which consists of fully ionized material due to it having very high temperatures compared to the surface of the sun. As the particles of this fluid are charged, the fluid interaction with the magnetic field and electric currents in the fluid have to be taken into account. The plasma in the solar corona can be described using magnetohydrodynamic (MHD) equations, which combine the Navier-Stokes equations from fluid dynamics with the Maxwell equations from electromagnetism. This approach to model the plasma treats it as a single fluid, as opposed to the two-fluid model, which describes the positive and negative charges separately.

The governing differential equations that make up the MHD model, are difficult to solve analytically. It is therefore often favourable to use numerical computer modelling to study the equations, which is the main focus of this bachelor's project. The simulations will be performed with the open source code PLUTO, which is designed to solve systems of partial differential equations in the context of astrophysical fluid dynamics [Mig+11].

The goals of this bachelor's project are to gain familiarity with simulating and modelling phenomena, in particular MHD waves, using numerical models and code such as PLUTO, and to gain some basic knowledge on MHD waves by analysing simulations. Concretely, this consists of the following two problems, simulating a magnetohydrodynamic blastwave and simulating the interaction of magnetohydrodynamic waves with large scale structures in the solar corona, such as a coronal hole and plume.

All the code used can be found on GitHub at `https://github.com/MichaelMaex/eindproject-2`.

2 (M)HD Theory

First a brief introduction to non-magnetic hydrodynamics is given which will then be compared to magnetohydrodynamics.

2.1 Ordinary Hydrodynamics

Fluids are described using the Navier-Stokes equations. In the case of an ideal fluid these equations are referred to as the Euler equations and take the form [Ach90, section 1.3]:

$$\begin{aligned}
&\text{conservation of mass:} && \frac{\partial \rho}{\partial t} && + \rho \nabla \cdot \mathbf{v} && = 0 \\
&\text{conservation of momentum:} && \rho \frac{\partial \mathbf{v}}{\partial t} && + \nabla p && = 0 \\
&\text{conservation of enery/adiabatic equation} && \frac{\partial}{\partial t}\left(\frac{p}{\rho^\gamma}\right) && && = 0,
\end{aligned}$$

where $\mathbf{v}$ is the velocity, ρ is the denisity, p is the pressure and γ is the ratio of the specific heat capacities (constant pressure, constant volume) which in this report will always be 5/3. These equations are differential forms of conservation laws.

In a linear approximation (p, ρ are small deviations of some constant background pressure and density p_0, ρ_0) the group speed of a wave (the speed at which a wave packet travels, energy moves) is independent of wavelength and direction. It is usually referred to as the speed of sound and is given by [Ach90, section 3.6]

$$c_s = \sqrt{\gamma \frac{p_0}{\rho_0}}.$$

2.2 Magnetohydrodynamics and the Solar Corona

The classic Navier-Stokes equations do not suffice to describe the movement in the solar corona due to the influence of strong magnetic fields on the ionized particles. This magnetic field stems from convection in the photosphere, which is a physical process that helps transport the nuclear energy created in the core of the sun [BB17].

The use of the magnetohydrodynamic model to describe the plasma depends on three main assumptions, namely, high collisionality, large scales and the assumption of ideal fluids. The last assumption implies that the dissipation of large scale variables is neglected [GP04]. This model also does not take relativity into account, as opposed to the relativistic magnetohydrodynamic model, which will not be discussed further[1]. The non-relativistic approximation is valid when the speeds considered are much smaller than the speed of light, which will be the case throughout this report.

The system of PDE's/conservations laws is the following:

$$\begin{aligned}
&\text{conservation of mass:} && \frac{\partial \rho}{\partial t} && + \rho \nabla \cdot \mathbf{v} && = 0 && \text{(2.1a)} \\
&\text{conservation of moment:} && \rho \frac{\partial \mathbf{v}}{\partial t} && + \nabla p - \frac{(\nabla \times \mathbf{B}) \times \mathbf{B}}{\mu} && = 0 && \text{(2.1b)} \\
&\text{Faraday's law:} && -\frac{\partial \mathbf{B}}{\partial t} && + \nabla \times (\mathbf{v} \times \mathbf{B}) && = 0 && \text{(2.1c)} \\
&\text{conservation of energy:} && \frac{\partial}{\partial t}\left(\frac{p}{\rho^\gamma}\right) && && = 0 && \text{(2.1d)}
\end{aligned}$$

[1] More information about the relativistic MHD model can be found in [Kar05]

Here ρ is the plasma density, $\mathbf{B}$ is the magnetic field, $\mathbf{v}$ the velocity, p is the thermal pressure. The constant μ is the magnetic permeability.

There are similarities to Euler equations (section 2.1). The third equation models the interaction between the currents in the plasma and the magnetic field and orginates in the Maxwell equations. As magnetic fields can carry momentum [Gri17, section 8.2] a term has to be added to the momentum equation. This term incorporates the effect of magnetic pressure the size of which is given by $P_B = \frac{B^2}{2\mu}$. A very important quantity often used in plasma physics is the plasma-β which gives the ratio between ordinary/thermal pressure and magnetic pressure. It is defined as

$$\beta = \frac{p}{P_B} = \frac{2\mu p}{B^2}.$$

If $\beta \gg 1$ the thermal pressure dominates and the magnetic effects can be ignored. If $\beta \ll 1$ the magnetic pressure dominates.

2.3 Linear Approximations

To analyse small amplitude waves in the plasma, a linearised version of the magnetohydrodynamic (eq. (2.1a)-(2.1d)) equations is considered as in [Fit11], namely

$$\frac{\partial \rho_1}{\partial t} + \rho_0 \nabla \cdot (\mathbf{v}_1) = 0 \tag{2.2a}$$

$$\rho_0 \frac{\partial \mathbf{v}_1}{\partial t} + \nabla p_1 - \frac{(\nabla \times \mathbf{B_1}) \times \mathbf{B_0}}{\mu} = 0 \tag{2.2b}$$

$$-\frac{\partial \mathbf{B}_1}{\partial t} + \nabla \times (\mathbf{v_1} \times \mathbf{B_0}) = 0 \tag{2.2c}$$

$$\frac{\partial}{\partial t}\left(\frac{p_1}{p_0} - \frac{\gamma \rho_1}{\rho_0}\right) = 0. \tag{2.2d}$$

Where μ is the vacuum permeability constant, and initial flow velocity and plasma current are assumed to be zero. This linearisation is done by writing the physical quantities as

$$f(\mathbf{x}, t) = f_0(\mathbf{x}) + f_1(\mathbf{x}, t)$$

under the assumption that the perturbation (f_1) of the equilibrium (f_0) is small compared to this equilibrium. The further analysis is based on the online lecture notes by R. Fitzpatrick [Fit11].

Consider perturbations of the following form, $\exp(i(\mathbf{kx} + \omega t))$, substituting this in the equations results in the following system of equations,

$$-\omega \rho_1 + \rho_0 \mathbf{k} \cdot \mathbf{v}_1 = 0 \tag{2.3a}$$

$$-\omega \rho_0 \mathbf{v}_1 + \mathbf{k} p_1 - \frac{(\mathbf{k} \times \mathbf{B_1}) \times \mathbf{B_0}}{\mu} = 0 \tag{2.3b}$$

$$-\omega \mathbf{B}_1 + \mathbf{k} \times (\mathbf{v_1} \times \mathbf{B_0}) = 0 \tag{2.3c}$$

$$-\omega \left(\frac{p_1}{p_0} - \frac{\gamma \rho_1}{\rho_0}\right) = 0. \tag{2.3d}$$

Rewriting these equations in function of the equilibrium states and $\mathbf{v_1}, \mathbf{k}$ and ω leads to,

$$\rho_1 = \rho_0 \frac{\mathbf{k} \cdot \mathbf{v_1}}{\omega}$$

$$p_1 = \gamma p_0 \frac{\mathbf{k} \cdot \mathbf{v_1}}{\omega}$$

$$\mathbf{B_1} = \frac{(\mathbf{k} \cdot \mathbf{v_1}) \cdot \mathbf{B_0} - (\mathbf{k} \cdot \mathbf{B_0}) \cdot \mathbf{v_1}}{\omega}$$

These values can now be subsituted in the linearised moment equation yielding the following,

$$-\omega\rho_0\mathbf{v_1} + \mathbf{k}\cdot\frac{\gamma p_0}{\omega} - \frac{(k\times\frac{(\mathbf{k}\cdot\mathbf{v_1})\cdot\mathbf{B_0}-(\mathbf{k}\cdot\mathbf{B_0})\cdot\mathbf{v_1}}{\omega})\times\mathbf{B_0}}{\mu} = 0$$

Further rearranging of the terms gives,

$$\left(\omega^2 - \frac{(\mathbf{k}\cdot\mathbf{B_0})^2}{\mu\rho_0}\right)\mathbf{v_1} = \left[\left(\frac{\gamma p_0}{\rho_0} + \frac{{\mathbf{B_0}}^2}{\mu\rho_0}\right)\mathbf{k} - \frac{(\mathbf{k}\cdot\mathbf{B_0})}{\mu\rho_0}\mathbf{B_0}\right](\mathbf{k}\cdot\mathbf{v_1}) - \frac{(\mathbf{k}\cdot\mathbf{B_0})(\mathbf{v_1}\cdot\mathbf{B_0})}{\mu\rho_0}\mathbf{k} \quad (2.4)$$

To simplify further notation, $c_s = \sqrt{\frac{\gamma p_0}{\rho_0}}$ will be used to represent the sound speed, and $v_A = \sqrt{\frac{{\mathbf{B_0}}^2}{\mu\rho_0}}$ will be used to represent the "Alfvén speed". Let θ be the angle between $\mathbf{B_0}$ and $\mathbf{k}$. Equation (2.4) then becomes

$$\left(\omega^2 - v_A^2\cos^2\theta\right)k^2\mathbf{v_1} - \left[(c_s^2+v_A^2)\mathbf{k} - v_A^2\cos\theta\frac{k}{B_0}\mathbf{B_0}\right](\mathbf{k}\cdot\mathbf{v_1}) + v_A^2\cos\theta\frac{k}{B_0}(\mathbf{v_1}\cdot\mathbf{B_0})\mathbf{k} = 0 \quad (2.5)$$

Assuming that the magnetic field lies in the x-direction, and the wave front vector k lies in the x,y-plane this can be rewritten as follows, where θ represents the angle between $\mathbf{B_0}$ and $\mathbf{k}$, i.e. $k_x = k\cos\theta,\ k_y = k\sin\theta,\ k_z = 0$.

$$\begin{aligned}
&\left(\omega^2 - k^2v_A^2\cos^2\theta\right)\mathbf{I}_3\begin{pmatrix}v_x\\v_y\\v_z\end{pmatrix}\\
&-k^2\left[(c_s^2+v_A^2)\begin{pmatrix}\cos\theta\\\sin\theta\\0\end{pmatrix} - v_A^2\cos\theta\begin{pmatrix}1\\0\\0\end{pmatrix}\right]\begin{pmatrix}\cos\theta & \sin\theta & 0\end{pmatrix}\begin{pmatrix}v_x\\v_y\\v_z\end{pmatrix}\\
&+k^2v_A^2\cos\theta\begin{pmatrix}\cos\theta\\\sin\theta\\0\end{pmatrix}\begin{pmatrix}1 & 0 & 0\end{pmatrix}\begin{pmatrix}v_x\\v_y\\v_z\end{pmatrix}\\
&= \mathbf{0}
\end{aligned}$$

Writing every term as a matrix yields:

$$\begin{aligned}
&\begin{pmatrix}\omega^2 - k^2v_A^2\cos\theta^2 & 0 & 0\\ 0 & \omega^2 - k^2v_A^2\cos^2\theta & 0\\ 0 & 0 & \omega^2 - k^2v_A^2\cos^2\theta\end{pmatrix}\begin{pmatrix}v_x\\v_y\\v_z\end{pmatrix}\\
&+\begin{pmatrix}-k^2c_s^2\cos^2\theta & -k^2c_s^2\cos\theta\sin\theta & 0\\ -k^2(c_s^2+v_A^2)\cos\theta\sin\theta & -k^2(c_s^2+v_a^2)\sin^2\theta & 0\\ 0 & 0 & 0\end{pmatrix}\begin{pmatrix}v_x\\v_y\\v_z\end{pmatrix}\\
&+\begin{pmatrix}k^2v_a^2\cos^2\theta & 0 & 0\\ k^2v_a^2\cos\theta\sin\theta & 0 & 0\\ 0 & 0 & 0\end{pmatrix}\begin{pmatrix}v_x\\v_y\\v_z\end{pmatrix}
\end{aligned}$$

This yields:

$$\begin{pmatrix}\omega^2 - k^2c_s^2\cos^2\theta & -k^2c_s^2\cos\theta\sin\theta & 0\\ -k^2c_s^2\cos\theta\sin\theta & \omega^2 - k^2(c_s^2\sin^2\theta - v_A) & 0\\ 0 & 0 & \omega^2 - k^2v_A^2\cos^2\theta\end{pmatrix}\begin{pmatrix}v_x\\v_y\\v_z\end{pmatrix} = \begin{pmatrix}0\\0\\0\end{pmatrix} \quad (2.6)$$

For this to be satisfied, the determinant of the matrix on the left hand side must be zero.

$$(\omega^2 - k^2v_A^2\cos\theta)\left[(\omega^2 - k^2c_s^2\cos^2\theta)(\omega - k^2c_s^2\sin^2\theta + k^2v_A) - k^4c_s^4\cos^2\theta\sin^2\theta\right] = 0.$$

Simplifying this results in the dispersion relation for plasma waves

$$(\omega^2 - k^2 v_A^2 \cos^2\theta)\left[\omega^4 - \omega^2 k^2 (v_A^2 + c_s^2) + k^4 v_A^2 c_s^2 \cos^2\theta\right] = 0. \tag{2.7}$$

This relation has three roots for ω^2 that correspond to three different kinds of waves:

$$\begin{aligned}
&\text{Alvén waves:} && \omega^2 = k^2 v_a^2 \cos^2\theta \\
&\text{fast magnetosonic waves:} && \omega^2 = \frac{k^2}{2}\left(v_A^2 + c_s^2 + \sqrt{(v_A^2 + c_s^2)^2 - 4 v_A^2 c_s^2 \cos^2\theta}\right) \\
&\text{slow magnetosonic waves:} && \omega^2 = \frac{k^2}{2}\left(v_A^2 + c_s^2 - \sqrt{(v_A^2 + c_s^2)^2 - 4 v_A^2 c_s^2 \cos^2\theta}\right).
\end{aligned}$$

2.4 Alfvén Waves

The previous derivation led to two different types of waves, namely, Alfvén waves and magnetosonic waves. An Alfvén wave is a magnetic tension wave caused by a restoring force. This force follows from Lenz's law, which states that the electrical currents caused by the motion of a conducting fluid in a magnetic field give rise to a force countering this motion, and Newton's second law for fluids [Fin07]. Properties of Alfvén waves are that they travel along the magnetic field lines and hence cannot propagate across field lines and that they do not perturb density [Fày16].

Alfvén waves occur when

$$\omega^2 = k^2 v_a^2 \cos^2\theta.$$

As can be seen from eq. (2.6) the corresponding eigenvector is in the z-direction

$$\mathbf{v_1} = v \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}.$$

As a result $\mathbf{k} \cdot \mathbf{v_1} = 0$ and eq. (2.3a) implies that the perturbation in density, $\rho_1 = 0$. Equation (2.3d) states that p_1 vanishes as well. As $\mathbf{B_0}$ and $\mathbf{V_1}$ lie in the x and z-direction respectively, *eq.* (2.2b) and *eq.* (2.2c) simplify to

$$\begin{aligned}
\rho_0 \frac{\partial v_z}{\partial t} - \frac{B_1}{\mu}\frac{\partial B_0}{\partial x} &= 0 \\
\frac{\partial B_x}{\partial t} + B_0 \frac{\partial v_z}{\partial t} &= 0.
\end{aligned}$$

This gives rise to the ordinary wave equation. The phase velocity of an Alfvén wave is equal to ω/k or,

$$v_{\text{ph}} = \omega/k = v_A \cos\theta.$$

The group velocity is equal to a constant, namely the Alfvén speed.

$$v_{\text{gr}} = v_A$$

The comparison of this kind of wave with the fast and slow acoustic waves is made in the Friedrichs diagrams below (see fig. 2.1).

2.5 Magnetosonic Waves

The second type of wave found in the dispersion relation are magnetosonic waves. There are two roots corresponding to this type of wave of which one is faster than the other, they correspond to what is called the fast magnetosonic wave and the slow magnetosonic wave.

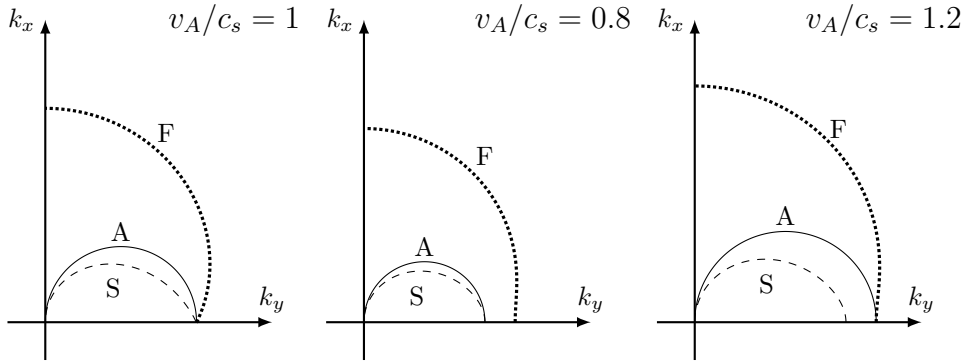

Figure 2.1: Diagrams Representing the phase speed of the different types of magnetohydrodynamic waves plotted as radial distance. Only the first quadrant is plotted as there is mirror symmetry with respect to the k_x, k_y-axes. The angle between k and B_0 is the angle away from the k_y axis.

The eigenvector linked to the magnetosonic waves is in the x, y-plane, as can be seen from eq. (2.6). In this case $\mathbf{k} \cdot \mathbf{v}$ and $\mathbf{v} \cdot \mathbf{B_0}$ do not vanish implying that the waves cause perturbations in the density and the pressure of the plasma.

The phase velocity can again be calculated by dividing ω by k, this results in,

$$\begin{aligned} v_{\text{phF}} &= \omega/k = \sqrt{\frac{1}{2}\left(v_A^2 + c_s^2 + \sqrt{(v_A^2 + c_s^2)^2 - 4v_A^2 c_s^2 \cos^2\theta}\right)} \\ v_{\text{phS}} &= \omega/k = \sqrt{\frac{1}{2}\left(v_A^2 + c_s^2 - \sqrt{(v_A^2 + c_s^2)^2 - 4v_A^2 c_s^2 \cos^2\theta}\right)} \end{aligned} \tag{2.8}$$

Figure 2.1 shows the phase speed of the slow, fast and Alfvén waves for different ratios of v_A/c_s. The derivation for the group velocity takes more steps, and is based on the derivation in [Lyu14]. For any wave it is defined as

$$v_{\text{gr}} = \frac{d\omega}{d\mathbf{k}} = \hat{k}\frac{\partial\omega}{\partial k} + \hat{\theta}\frac{1}{k}\frac{\partial\omega}{\partial\theta}$$

The k-component can be calculated as follows,

$$\begin{aligned} \frac{\partial\omega}{\partial k} &= \frac{\partial}{\partial k}\left(k v_{\text{ph}}\right) \\ &= v_{\text{ph}} \end{aligned}$$

Which is a function of θ.
The θ-component can be calculated by rewriting the term,

$$\frac{1}{k}\frac{\partial\omega}{\partial\theta} = \frac{\partial}{\partial\theta}\left(\frac{\omega}{k}\right) = \frac{\partial}{\partial\theta}(v_{\text{ph}}) = \frac{1}{2v_{\text{ph}}}\frac{\partial}{\partial\theta}(v_{\text{ph}}^2)$$

The remainder of the calculations will be done for the fast wave, the slow wave case is analogous. The partial derivative of the phase speed of the fast wave squared becomes,

$$\begin{aligned} \frac{\partial}{\partial\theta}(v_{\text{phF}}^2) &= \frac{1}{2}\frac{\partial}{\partial\theta}\sqrt{(v_A^2 + c_s^2)^2 - 4v_A^2 c_s^2 \cos^2\theta} \\ &= \frac{2v_A^2 c_s^2 \cos\theta \sin\theta}{\sqrt{(v_A^2 + c_s^2)^2 - 4v_A^2 c_s^2 \cos^2\theta}} \end{aligned}$$

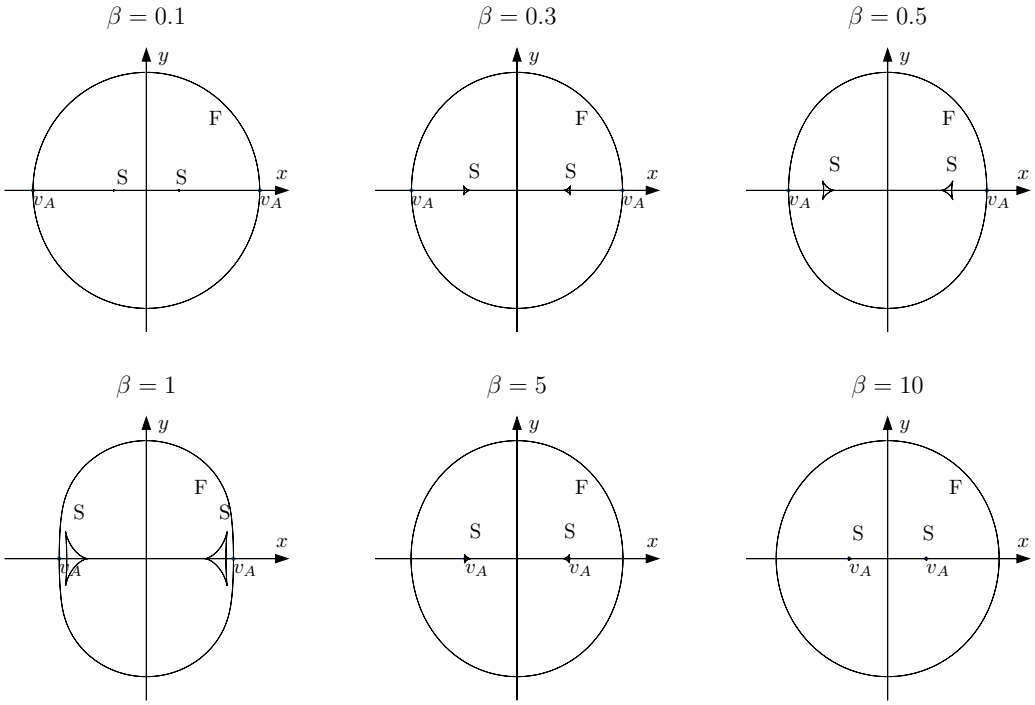

Figure 2.2: The possible group speeds for fast, slow and Alfvén waves for different values of β. The magnetic field lines are horizontal.

Substituting this back in to the equation gives,

$$\begin{aligned}\frac{1}{k}\frac{\partial\omega}{\partial\theta} &= \frac{1}{v_{\mathrm{phF}}}\frac{v_A^2 c_s^2 \cos\theta\sin\theta}{\sqrt{(v_A^2+c_s^2)^2 - 4v_A^2c_s^2\cos^2\theta}} \\ &= \frac{1}{\sqrt{\frac{1}{2}\left(v_A^2+c_s^2+\sqrt{(v_A^2+c_s^2)^2-4v_A^2c_s^2\cos^2\theta}\right)}}\frac{v_A^2 c_s^2 \cos\theta\sin\theta}{\sqrt{(v_A^2+c_s^2)^2 - 4v_A^2c_s^2\cos^2\theta}}\end{aligned}$$

Combining this with the k-component gives the group velocity,

$$\begin{aligned}v_{\mathrm{grF}} =& \hat{k}\sqrt{\frac{1}{2}\left(v_A^2+c_s^2+\sqrt{(v_A^2+c_s^2)^2-4v_A^2c_s^2\cos^2\theta}\right)} \\ &+ \hat{\theta}\frac{1}{\sqrt{\frac{1}{2}\left(v_A^2+c_s^2+\sqrt{(v_A^2+c_s^2)^2-4v_A^2c_s^2\cos^2\theta}\right)}}\frac{v_A^2 c_s^2 \cos\theta\sin\theta}{\sqrt{(v_A^2+c_s^2)^2 - 4v_A^2c_s^2\cos^2\theta}} \\ v_{\mathrm{grS}} =& \hat{k}\sqrt{\frac{1}{2}\left(v_A^2+c_s^2-\sqrt{(v_A^2+c_s^2)^2-4v_A^2c_s^2\cos^2\theta}\right)} \\ &- \hat{\theta}\frac{1}{\sqrt{\frac{1}{2}\left(v_A^2+c_s^2-\sqrt{(v_A^2+c_s^2)^2-4v_A^2c_s^2\cos^2\theta}\right)}}\frac{v_A^2 c_s^2 \cos\theta\sin\theta}{\sqrt{(v_A^2+c_s^2)^2 - 4v_A^2c_s^2\cos^2\theta}}\end{aligned}$$

This can be simplified with eq. (2.8):

$$v_{\mathrm{grF}} = v_{\mathrm{phF}}\hat{k} + \frac{v_A^2 c_s^2 \cos\theta \sin\theta}{v_{\mathrm{phF}}\sqrt{(v_A^2 + c_s^2)^2 - 4v_A^2 c_s^2 \cos^2\theta}}\hat{\theta}$$
$$v_{\mathrm{grS}} = v_{\mathrm{phS}}\hat{k} - \frac{v_A^2 c_s^2 \cos\theta \sin\theta}{v_{\mathrm{phS}}\sqrt{(v_A^2 + c_s^2)^2 - 4v_A^2 c_s^2 \cos^2\theta}}\hat{\theta}$$

Figure 2.2 shows the group speed of the slow, fast and Alfvén waves for different values of the plasma-β

3 Pluto Code

The Pluto library consists completely out of uncompiled code. For each problem/simulation a binary needs to be made specific to that problem. This allows Pluto to be run on a large variety of devices, provided that the necessary compilers are available. The files specific to a problem are referred to as the definition of the problem. This definition typically consists of three files, `definitions.h`, `init.c`, `pluto.ini`.

1. `definitions.h` contains all the information necessary for compiling the right code for the project. This mostly consists of information for the solver: what solver is used (HD, MHD, relativistic MHD, ...), what kind of grid geometry is used (number of dimensions, cartesian, cylidrical, polar), what code units represent (see section 3.1). It also contains what user-definable variables there are.

2. `init.c` contains the C-functions that specify initial and boundary conditions. The function `void Init (double *v, double x1, double x2, double x3)` sets up stores the initial conditions at the point `(x1,x2,x3)` and stores them in at the pointer `*v`. The function `void UserDefBoundary (const Data *d, RBox *box, int side, Grid *grid)` can be used to define custom boundary conditions when the build in boundary conditions (outflow, reflective, periodic, ...) are not sufficient.

3. `pluto.ini` contains information that the binary reads before starting the computation. This constists of grid size and resolution, simulation time, what output needs to be stored where and the values of the user defined parameters.

The compiler uses `definitions.h` to only compile the right solver and `init.c`. One binary can use multiple `pluto.ini`'s which makes it easy to run multiple simulations of the same setup with different input variables.

3.1 Units in Pluto

Internally Pluto uses dimensionless code units. All units are derived from the constants `UNIT_DENSITY` (ρ_0), `UNIT_LENGTH` (L_0), `UNIT_VELOCITY` (v_0), all of which are denoted in cgs units. All other units are derived from these units. Note that the unit for magnetic field is defined as $B_0 = v_0\sqrt{4\pi\rho_0}$.

Table 1 shows some important dimensions in (M)HD and what one code unit (up to three figures) represents when `UNIT_DENSITY`, `UNIT_LENGTH` and `UNIT_VELOCITY` have their default values.

3.2 Initial Example

The first problem that will be analysed is a simple hydrodynamic blastwave generated by a high-pressure region. The viscosity, heat conduction and dissipation of the fluid will be neglected. This problem is meant to be an initial example of how to set up a non-predefined simulation and is a modified version of the initial example described in section

Table 1: Important dimensions and their default code units in PLUTO

one code unit of	corresponds to
density	$m_p\ \mathrm{cm}^{-3} = 1.67 \times 10^{-24}\,\mathrm{g\,cm}^{-3}$
lenth	$1\,\mathrm{AU} = 1.50 \times 10^{11}\,\mathrm{m}$
velocity	$1\,\mathrm{km\,s}^{-1}$
time	$1.50 \times 10^{8}\,\mathrm{s}$
magnetic field strength	$4.58 \times 10^{-7}\,\mathrm{Gs}$
pressure	$1.67 \times 10^{-14}\,\mathrm{Ba} = 1.67 \times 10^{-15}\,\mathrm{Pa}$

0.4 of PLUTO's User Guide [Mig+18]. The blastwave is simulated in two dimensions and uses three user-defined parameters, namely, the ambient pressure, the pressure in the high-pressure region, and γ which in this report will always be 5/3. The domain is a rectangle going from -10 to 10 (code units), in both the x and the y direction, which is represented by a 1024×1024 grid.

The following initial condition is used, a high-pressure region inside a circle of radius 0.3 is surrounded by an ambient pressure region. The simulation is run for 2 code units of time, and every 0.01 code units, a double-precision file is saved.

The file `pluto.ini` stores parameters for the simulation, including grid size, grid resolution, simulation time, output format and user defined parameters. In this case can be set up in PLUTO by making the following adjustments to the `pluto.ini` file.

```
[Grid]

X1-grid    1    -10.0    1024    u    10.0
X2-grid    1    -10.0    1024    u    10.0

[Static grid output]

dbl        0.01  -1   multiple_files

[Parameters]

P_IN                        15.0
P_OUT                       8.0
GAMMA                       1.66666666666666667
```

The file `init.c` contains the code to generate the initial and boundary conditions. To define the problem the following adjustments have to be made to the `void init(...)` function, which generates the initial conditions for a given point on the grid.

```
void Init (double *v, double x1, double x2, double x3)
{
  double r;
  g_gamma = g_inputParam[GAMMA];
  r = x1*x1 + x2*x2;
  r = sqrt(r);

  v[RHO] = 1.0;
  v[VX1] = 0.0;
  v[VX2] = 0.0;
  v[VX3] = 0.0;
```

```
  #if HAVE_ENERGY
  v[PRS] = g_inputParam[P_OUT];
  #endif
  if (r <= 0.3) v[PRS] = g_inputParam[P_IN];
  v[TRC] = 0.0;
}
```

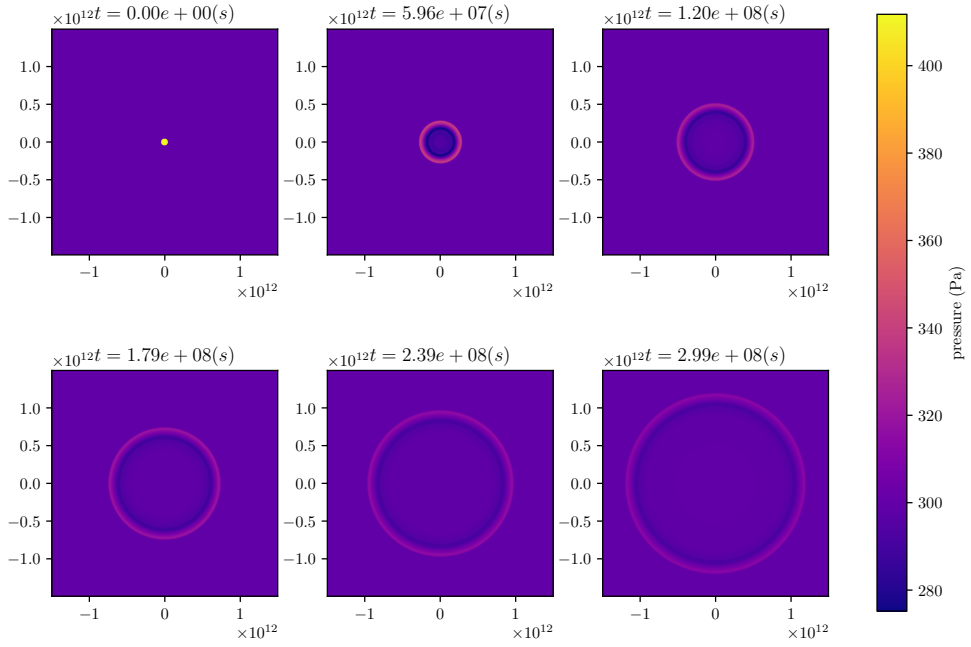

Figure 3.1: Evolution of the pressure of the simulated blast wave. The x and y-axis are in m. Note that the colormap is cut off at 412 Pa (11 code units) and that the inital high pressure region is not faithfully represented.

PLUTO's hydrodynamics module solves the following system of equations

$$\begin{aligned}
\frac{\partial \rho}{\partial t} &+ \nabla \cdot (\rho \mathbf{v}) &&= 0 && \text{(mass)} \\
\frac{\partial \mathbf{m}}{\partial t} &+ \nabla \cdot (\mathbf{m}\mathbf{v} + \mathbf{I}p)^T &&= -\rho \nabla \Phi && \text{(moment)} \\
\frac{\partial (E_t + \rho\Phi)}{\partial t} &+ \nabla \cdot ((E_t + p + \rho\boldsymbol{\Phi})\mathbf{v}) &&= 0. && \text{(energy)}
\end{aligned}$$

Here is ϕ the potential, ρ the mass density, $\mathbf{m} = \rho\mathbf{v}$, the momentum density, $\mathbf{v}$ the speed, p the thermal pressure and E_t the total energy density. These can be derived from the standard Euler equations by changing the total derivatives to partial derivatives with respect to time. This is necessary because PLUTO uses a static grid to run simulations, and hence uses local change.

Figure 3.1 shows pressure of the simulated fluid at multiple points in time. The speed at which the wave front moves is the group speed. According to section 2.2 this is given by $c_s = \sqrt{\frac{5}{3}\frac{p_0}{\rho_0}}$. Hence the speed (in code units) is $\sqrt{\frac{5}{3}8} = 3.61$, which corresponds to $3.61 \times 10^5\,\mathrm{cm\,s^{-1}}$. A comparison of the theoretical radius against the simulated radius can

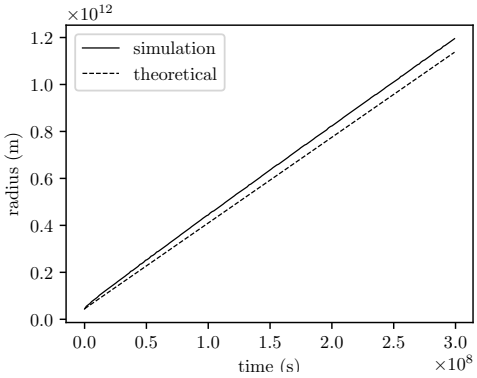

Figure 3.2: The position of the simulated wave front compared to the theoretical speed

be found in fig. 3.2. Initially the simulated wave moves faster than the theoretical group speed. This could be an result of the linear approximation used in determining the group speed. However for the remainder of the simulation the wave front matches the theoretical group speed quite well.

3.3 Non-linear Effects

Due to our current understanding of the (magnetic) navier-stokes equations, linear approximations are necessary for many theoretical results. This is one limitation of the theory that numerical simulations aim to solve. A problem was setup with the same initial conditions as in section 3.2, except that the initial pressure is 100 (code units) in the centre and 8 elsewhere. Like in the previous setup the radius of the blast over time was compared to the theoretically linear group speed. Figure 3.3 shows concave radius/time relation against the linear theoretically predicted relation. It can be clearly seen that the theory matches the low pressure simulation better than the high pressure difference. Especially in the beginning of the simulation the wave front moves a lot faster than the theoretical wave front. When the wave is more dispersed and the pressure difference at the wave front is lower the speed does match the theoretical speed. This shows that the theory is not sufficient for a complete description of MHD fluids but can be used to gain insights non the less.

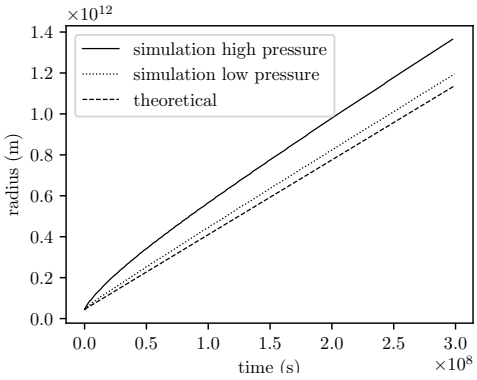

Figure 3.3: A comparison of the theoretically predicted radius of the blast wave (using a linear approximation) versus the simulated result with a high pressure difference.

4 Waves in Magnetohydrodynamic Fluids

To simulate magnetohydrodynamic waves, PLUTO solves the following system of differential equations, which is a slightly different formulation of the one defined in section 2.

$$\text{mass:} \quad \frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{v}) = 0 \tag{4.1a}$$

$$\text{moment:} \quad \frac{\partial \mathbf{m}}{\partial t} + \nabla \cdot \left[\mathbf{mv} - \mathbf{BB} + \mathbf{I}\left(p + \frac{\mathbf{B}^2}{2}\right)\right]^T = -\rho \nabla \Phi + \rho \cdot g \tag{4.1b}$$

$$\text{Farady:} \quad \frac{\partial \mathbf{B}}{\partial t} + \nabla \times (cE) = 0 \tag{4.1c}$$

$$\text{energy:} \quad \frac{\partial (E_t + \rho\Phi)}{\partial t} + \nabla \cdot \left[\left(\frac{\rho \mathbf{v}^2}{2} + \rho e + p + \rho \Phi \right) v + c\mathbf{E} \times \mathbf{B} \right] = m \cdot g. \tag{4.1d}$$

Here ρ is the plasma density, $\mathbf{B}$ the magnetic field, $\mathbf{v}$ the velocity, $\mathbf{m} = \rho \mathbf{v}$ the momentum density, Φ the potential and p is the thermal pressure. E_t is the total energy, which is given by

$$E_t = \rho e + \frac{m^2}{2\rho} + \frac{B^2}{2}. \tag{4.2}$$

Here is c the speed of light and ρe is the internal energy, which for ideal fluids is given by $\frac{p}{\gamma - 1}$ with γ the ratio of heat capacities.[Mig+18]. Only ideal fluids will be discussed.

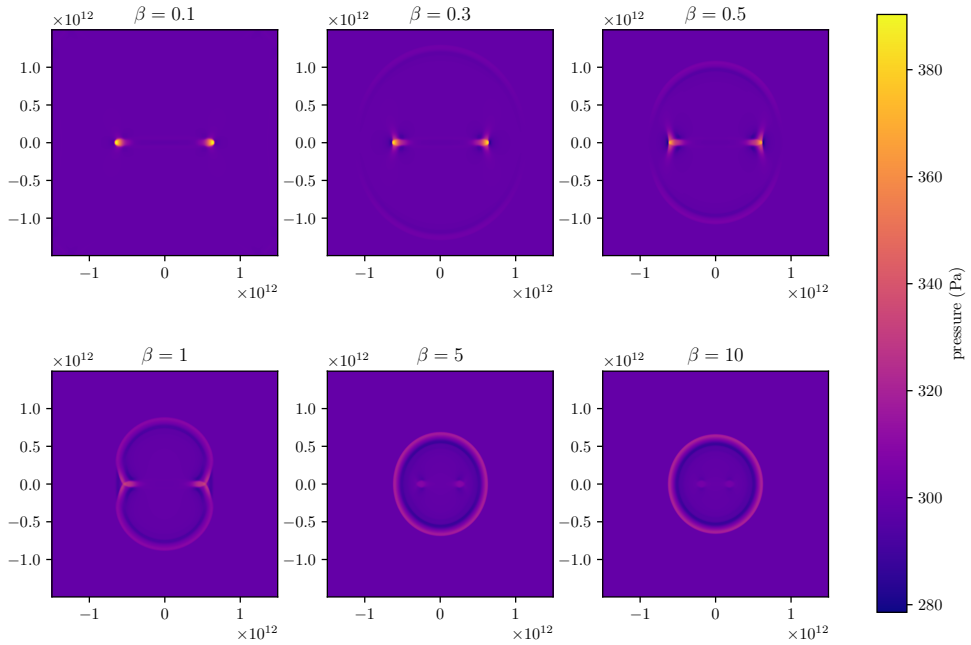

Figure 4.1: The shape of the blast wave for different plasma beta's, for the low pressure simulations. Pressure is shown. The value for β is modified with the thermal pressure held constant at 299 Pa (8 code units) and constant density of 1 proton mass per cm^3 (1 code unit). All figures are the simulation after 149 597 892 s (1 code unit). The axes are in m.

In order to study the effects of a constant magnetic background field, the same initial conditions as in section 3.2 were used but this time a magnetic field in the x direction was

added. This can be done by adding a parameter β, and adjusting the `void init(...)` function in the `init.c` file as follows,

```
void Init (double *v, double x1, double x2, double x3){
  ...
  #if PHYSICS == MHD || PHYSICS == RMHD
  double g_beta = g_inputParam[BETA];
  v[BX1] = sqrt(g_inputParam[P_OUT]*2.0/g_beta);
  v[BX2] = 0.0;
  v[BX3] = 0.0;
  #endif
  ...
}
```

The parameter $\beta = \frac{2p}{B^2}$ (in code units) gives the ratio of magnetic pressure over mechanical pressure. If $\beta = \infty$ the magnetic field is 0. If $\beta \ll 1$ the magnetic pressure dominates.

The simulation was run for

- 6 values for β : $0.1, 0.3, 0.5, 1, 5, 10$
- 2 values for the high pressure region (in code units): $15, 100$

the background pressure was always 8 code units.

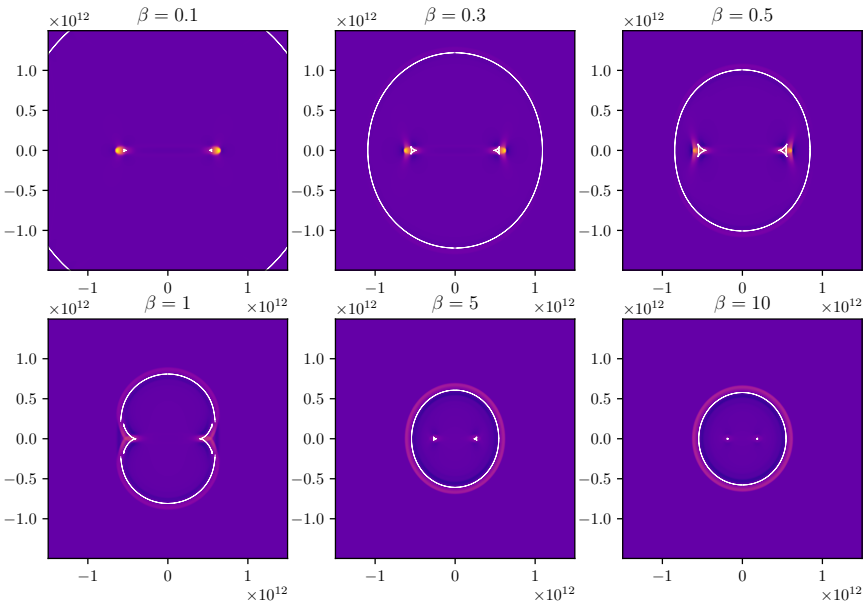

Figure 4.2: The group speed diagrams (fig. 2.2) overlayed on the simulations (fig. 4.1). The white lines are the theoretical positions of the slow and fast magnetosonic wavefronts.

Figure 4.1 shows how the value of β influences the shape of the blast wave. When β is high, the blastwave resembles the ordinary non-magnetic hydrodynamics(fig. 3.1). The shape very closely resembles the shape of the group speeds diagrams in fig. 2.2. One can

multiply the theoretical group speeds with the elapsed time to find the position of the high pressure region at that time. Overlaying fig. 2.2 with fig. 4.1 in this way results in fig. 4.2. One can see that they match very well. The experimental results are slightly ahead of the theoretical wavefront. The authors suspect that this is due two reasons:

1. the theoretical position determines the movement of the centre of the initial high pressure region. But this has a non-zero radius.

2. the initial pressure difference (15 over 8 code units) is slightly too high for linear approximations. As a result the real group speed is higher than the linear theory would suggest.

4.1 Sound Waves

In fig. 4.3 the simulation results for the pressure of a magnetohydrodynamic blastwave with $\beta = 0.1$ are plotted, for different points in time. The simulation does not resemble that of a hydrodynamic blastwave, because β is relatively small, which implies that the magnetic pressure dominates over the mechanical pressure. In this simulation the fast magnetosonic wave is barely visible. Hence this simulation can be used to compare the theoretical and experimental group speed of the slow wave.

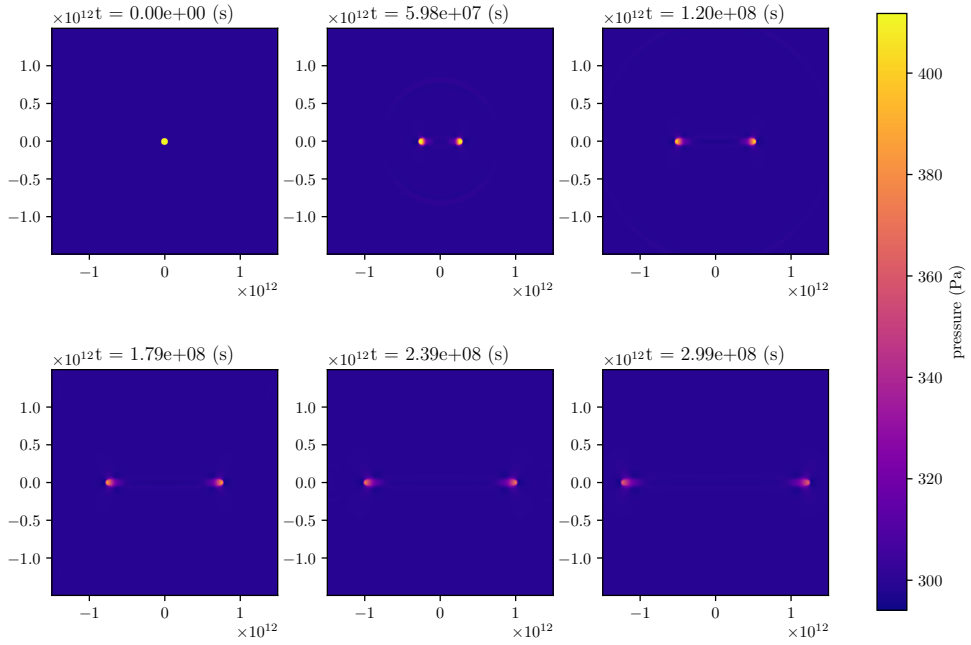

Figure 4.3: The propagation of a slow wave. In this example $\beta = 0.1$ Axes are in m.

Taking the right half of a horizontal slice and plotting over time is a good way of visualizing the propagation of the slow magnetosonic waves. This is visualised fig. 4.4 along with the theoretical boundary of the wave packet determined using the theoretical speed of the slow wave. Again the wave is slightly ahead of what theory predicts in the low pressure case. This is even more so for the high pressure blast. Again the authors think this discrepancy is due to non-linear effects.

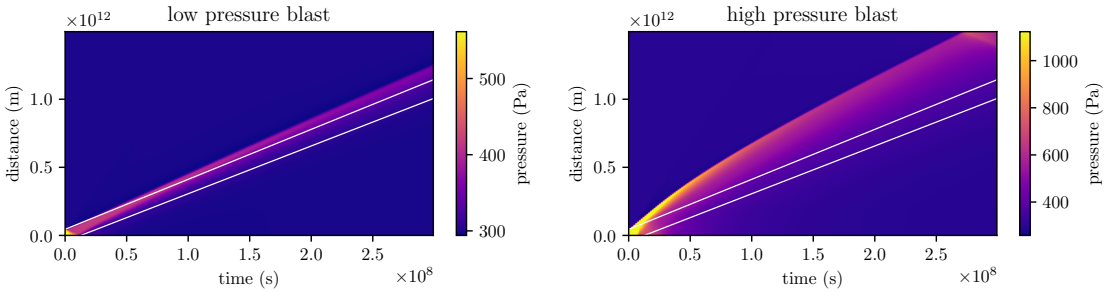

Figure 4.4: The pressure of the a horizontal slice plotted over time of the HMD blastwave simulation with $\beta = 0.1$ for both the low and high pressure difference case. The white lines represent the theoretical bounds of the wave packet. The possible group speeds of a slow wave form a triangle as can be seen in fig. 2.2. Taking let v_m, v_M be the minimum and maximum speed in the x-direction of such a triangle. Let r be the initial radius of the high pressure region. Then $x = v_m \cdot t - r, x = v_M \cdot t + r$ represent are the white boundaries.

5 Interaction of MHD Waves with Large Scale Structures

This task focuses on the interaction of large scale structures in the solar corona with magnetohydrodynamic waves and is based on the paper titled "Propagation of a global coronal wave and its interaction with large-scale coronal magnetic structures" by Afanasyev and Zhukov. [AZ18]

The shock wave simulated in section 4 is a good description of a wave in a uniform plasma, however it does not account for various large scale non-uniformities, such as coronal holes and plumes. The interaction of coronal waves with these structures results in wave phenomena such as wave transmission and reflection, which have been observed in extreme ultraviolet images of the Sun (see for example [Gop+09]).

5.1 Coronal Hole Model

A coronal hole, is a region of lower temperature and density compared to the surrounding plasma. This results in a region of higher Alfvén speed. They occur when a magnetic field does not fall back, and is open to interplanetary space .

To simulate a coronal hole, the following initial conditions are used,

$$n = n_{\text{out}} - (n_{\text{out}} - n_{\text{in}}) \exp\left[-\left(\frac{r}{d}\right)^8\right]$$
$$T = T_{\text{out}} - (T_{\text{out}} - T_{\text{in}}) \exp\left[-\left(\frac{r}{d}\right)^8\right]$$

Here n denotes the plasma number density, T denotes the temperature, and r is the distance from the center. The plasma number density counts the number of particles per volume and thus is related to the density by $\rho = m \cdot \eta$, where m is the mass of one particle. In this case m is taken to be the mass of a proton. The parameter d is used to describe the characteristic size of the non-uniformity and can hence be used to adjust the simulation to the correct order of magnitude. In this simulation d will be equal to 150 Mm (see also

table 2).

The magnetic field in the simulation is directed in the z-direction while the simulation takes place in the x, y-plane. An initial equilibrium is achieved by keeping the total pressure constant, using the following equation,

$$p^t = p_{\text{gas}} + \frac{B^2}{8\pi} = \text{constant}.$$

Here p^t represents the total pressure, which is the sum of the gas pressure and the magnetic pressure.

The simulation takes place on a domain stretching from -10^9 m to 10^9 m in both the x and y direction. The grid has a resolution of 1024×1024.

A wave is induced by modifying the velocity in the x-direction in the boundary condition at the left boundary edge. The velocity at the edge is given by

$$v_x = v_{\text{max}} \tanh\left(\frac{t}{a_0}\right) - \frac{v_{\text{max}}}{2}\left(\tanh\left(\frac{t - t_0}{a_1}\right) + 1\right),$$

where t is the elapsed time, v_{max} the maximum speed obtained, t_0 the duration of the wave. To avoid shocks, a hyperbolic tangent function is used to smoothly interpolate from 0 to v_{max} and back. The values a_0, a_1 determine how fast it v_x moves in between 0 and v_{max}. In this report a_0 and a_1 will always be 20 s and 60 s respectively. Figure 5.1 shows the velocity profile for these curves for both the coronal hole model and the coronal plume model (see section 5.2).

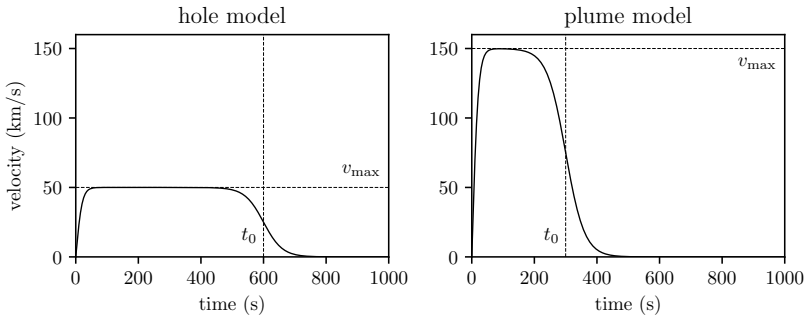

Figure 5.1: The velocity profile of the wave generated at the boundary of the coronal hole and coronal plume model.

The values that are used for the parameters are summarised in table 2.

Table 2: Parameters used to simulate the coronal hole model

Parameter	**Value**	**Unit**
d	150	Mm
n_{out}	1.0×10^9	cm^{-3}
n_{in}	1.0×10^8	cm^{-3}
T_{out}	1.5	MK
T_{in}	1.0	MK
B_{out}	4.0	G
v_{max}	50	km s^{-1}
t_0	600	s

5.1.1 Execution in Pluto

To reflect the more typical length/velocity/density scales in the solar corona the code units were redefined to $1\,\mathrm{Mm}, 1\,\mathrm{Mm\,s^{-1}}$ and 10^9 proton massses per cm^3 respectively. This is done by adding the following code to the `defintions.h` file.

```
/* [Beg] user-defined constants (do not change this line) */

#define  UNIT_DENSITY                  (1.e9*CONST_mp)
#define  UNIT_LENGTH                   1.e8
#define  UNIT_VELOCITY                 1.e8

/* [End] user-defined constants (do not change this line) *
```

To define these initial conditions the following adjustments have to be made to the `void init(...)` function in the `init.c` file. This code is an modified version of code kindly provided by Mijie Shi.

```
void Init (double *v, double x1, double x2, double x3)
{ double rho_out, rho_ins, rho_all, T_out, T_ins, T_all, B_out, p_out,
↪  p_all;  //define some variables
  double d;
  double r;

  d = g_inputParam[DIAM];
  rho_out = g_inputParam[RHO_OUT];
  rho_ins = g_inputParam[RHO_INS];
  T_out = g_inputParam[T_OUT];   //Physical value (Kelvin)
  T_ins = g_inputParam[T_INS];   //Physical value (Kelvin)

  r = sqrt(x1*x1 + x2*x2);
  rho_all = rho_out - (rho_out - rho_ins)*exp(-(pow(r/d,8))); //density
  ↪  profile
  T_all = T_out - (T_out - T_ins)*exp(-(pow(r/d,8)));
  ↪  //temperature profile

  p_out = rho_out*T_out/(KELVIN*0.5);                          //converitng
  ↪  from density and temperature to pressure
  p_all = rho_all*T_all/(KELVIN*0.5);                          //KELVIN is
  ↪  defined by PLUTO

  B_out =
  ↪  g_inputParam[B_OUT]/sqrt(4.0*CONST_PI*UNIT_DENSITY)/UNIT_VELOCITY;
  ↪  //normalized magnetic field outside of the structure (physical value
  ↪  is 4 Gauss)


  v[RHO] = rho_all;
  v[VX1] = 0.0;
  v[VX2] = 0.0;
  v[VX3] = 0.0;
  #if HAVE_ENERGY
  v[PRS] = p_all;
```

```
  #endif
  v[TRC] = 0.0;

  #if PHYSICS == MHD || PHYSICS == RMHD
  v[BX1] = 0.0;
  v[BX2] = 0.0;
  v[BX3] = sqrt(B_out*B_out + 2*(p_out - p_all));            //deriving
  ↪  the magnetic field from pressure balance, i.e., p + B^2/2 = const

  v[AX1] = 0.0;
  v[AX2] = 0.0;
  v[AX3] = 0.0;
  #endif
}
```

To generate a plane wave, the following time-dependent boundary condition is used at the right boundary of the simulation box. Firstly, two parameters are defined that make up the general shape of the wave namely, `WAVE_HEIGHT`, which specifies the top speed of the wave, in km/s, and `WAVE_DURATION`, which specifies the length of the wave.

To initialise these boundary conditions in PLUTO, the following code is added to the `void UserDefBoundary(...)` function in the `init.c` file, again a modified version of the code by Mijie Shi.

```
void UserDefBoundary (const Data *d, RBox *box, int side, Grid *grid){
  ...
  double v0, v1, v2, height, duration;
  height = g_inputParam[WAVE_HEIGHT];
  duration = g_inputParam[WAVE_DURATION];
  v0 = -height*1.e5/UNIT_VELOCITY;
  v1 = -v0;
  v2 = 0.0;

  if (side == X1_BEG){  /* -- X1_BEG boundary -- */
    if (box->vpos == CENTER) {
      BOX_LOOP(box,k,j,i){
      d->Vc[RHO][k][j][i] = d->Vc[RHO][k][j][2*IBEG - i -1];
      d->Vc[PRS][k][j][i] = d->Vc[PRS][k][j][2*IBEG - i -1];

      d->Vc[BX1][k][j][i] = d->Vc[BX1][k][j][2*IBEG - i -1];
      d->Vc[BX2][k][j][i] = d->Vc[BX2][k][j][2*IBEG - i -1];

      d->Vc[VX1][k][j][i] = v1 + (v0 -
      ↪  v1)*(0.5*(1.-tanh((g_time-0.)/20.))) + (v2 -
      ↪  v1)*(0.5*(1.+tanh((g_time-duration)/60.)));

      d->Vc[VX2][k][j][i] = 0.0;

      }
    }else if (box->vpos == X1FACE){
      BOX_LOOP(box,k,j,i){  }
    }else if (box->vpos == X2FACE){
      BOX_LOOP(box,k,j,i){  }
```

```
    }else if (box->vpos == X3FACE){
      BOX_LOOP(box,k,j,i){  }
    }
  }
  ...
}
```

The user defined parameters, the gridsize and resolution, output file type and timescale for the problem, can be defined in the `pluto.ini` file, similarly as in section 3.2.

5.1.2 Results and Analysis

Figure 5.2 shows the pressure at multiple points in time during the simulation.

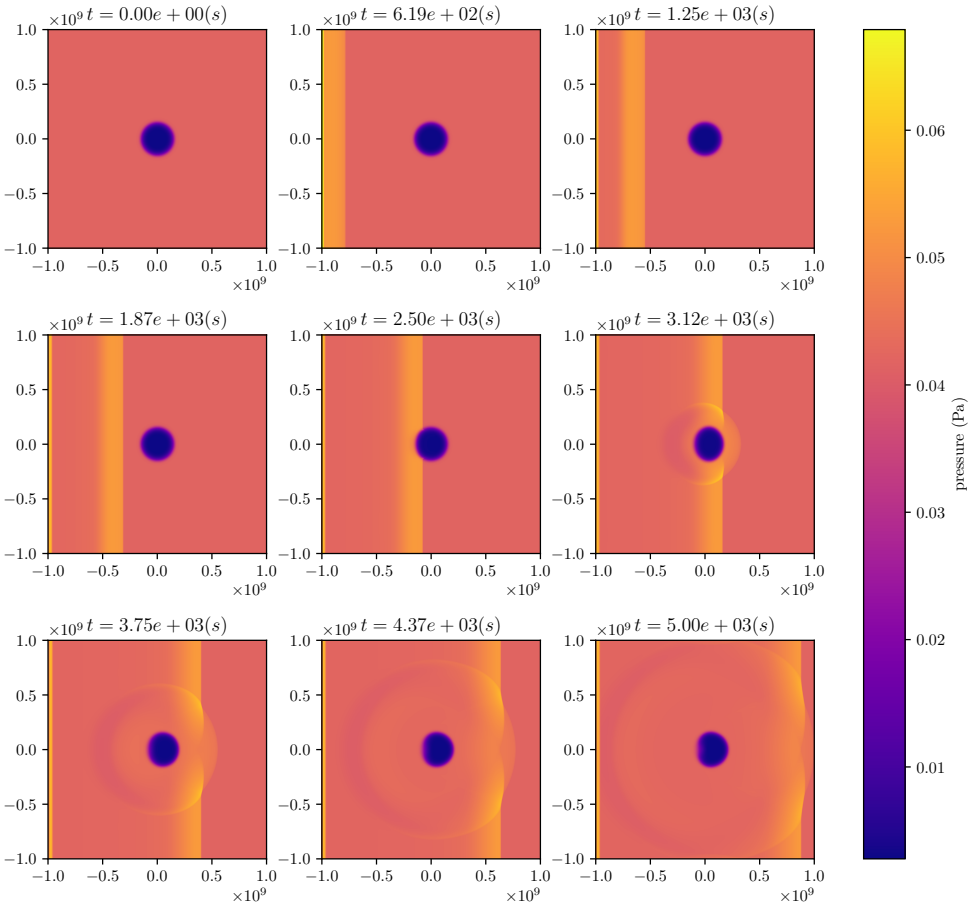

Figure 5.2: The pressure of the coronal hole simulation at multiple points in time. Note that the axes are in m.

Equation (4.2) gives the energy per volume (or area in a 2D simulation). Integrating this a region gives the energy in that region. This can be used to calculate the reflection coefficient. The energy of the left half, E_0, E_1, E_2 (excluding the slim line at the very left which appears to be an artefact) was calculated at three points in time $t_0 = 0, t_1, t_2$ such that at t_1 the wave is entirely located in that region and at t_2 the entirety of the reflection is captured. The specific points in time and integrated area can be found in fig. 5.3. Then

$E_i - E_0$ gives the energy that was added due to the wave at t_i. Hence the reflection coefficient is

$$R = \frac{E_2 - E_0}{E_1 - E_0} = 0.17.$$

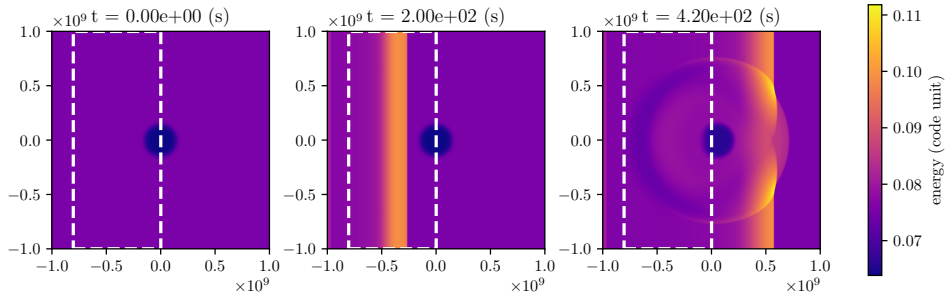

Figure 5.3: The total energy is plotted at the initial state, before the wave hits the hole and after the wave as passed the hole. The white box shows what part was integrated to obtain the total enegry in the wave/reflection.

5.2 Coronal Plume Model

To contrast the coronal hole model, the plume model is considered. It is defined as a region of high temperature and density, and as a result a lower Alfvén speed. This model is less applicable to real plumes, or collections of plumes, as the coronal hole model is to real coronal holes, but is still considered for completeness [AZ18].

The simulation of the coronal plume consists of the same initial conditions as in the coronal hole simulation, namely,

$$n = n_{\text{out}} - (n_{\text{out}} - n_{\text{in}}) \exp\left[-\left(\frac{r}{d}\right)^8\right]$$

$$T = T_{\text{out}} - (T_{\text{out}} - T_{\text{in}}) \exp\left[-\left(\frac{r}{d}\right)^8\right]$$

$$p^t = p_{\text{gas}} + \frac{B^2}{8\pi} = \text{constant}.$$

However, this time other values will be used for the parameters, which can be found in table 3.

Table 3: Parameters used to simulate the coronal plume model

Parameter	**Value**	**Unit**
d	150	Mm
n_{out}	1.0×10^8	cm^{-3}
n_{in}	1.0×10^9	cm^{-3}
T_{out}	1.5	MK
T_{in}	1.0	MK
B_{out}	3.0	G
v_{max}	150	km s^{-1}
t_0	300	s

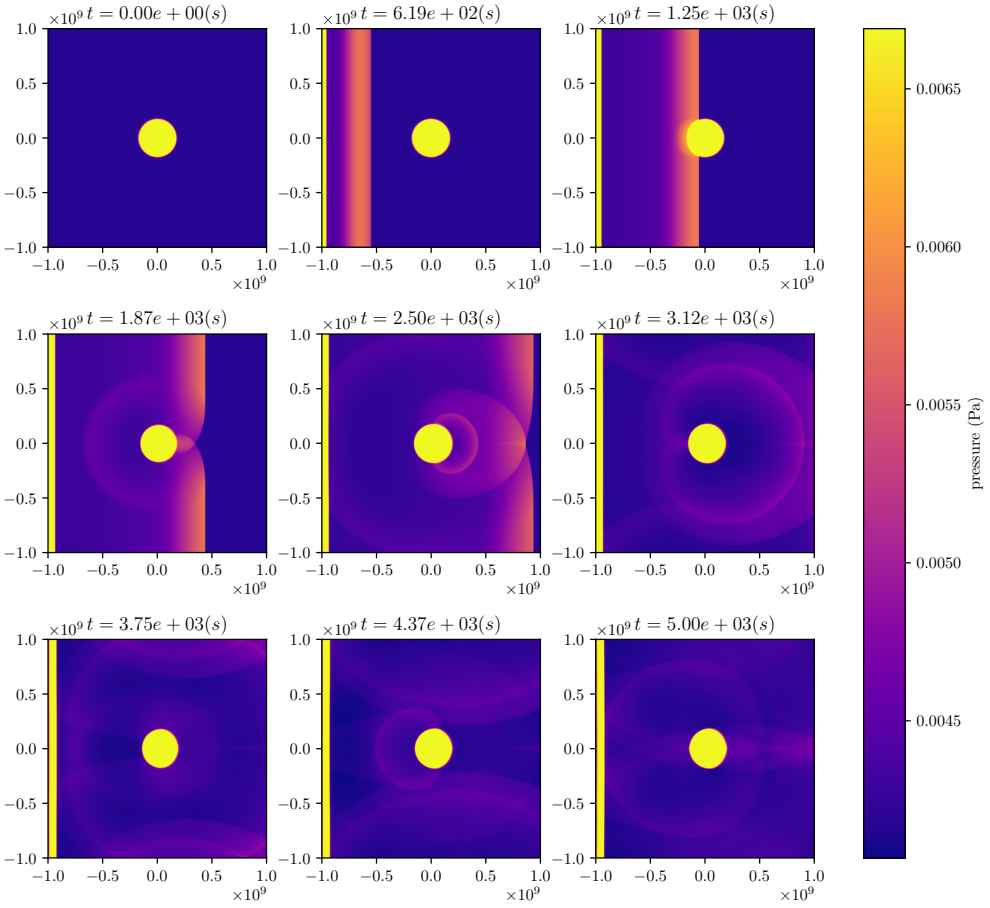

Figure 5.4: The pressure of the coronal plume simulation at multiple points in time. Note that the pressure is cut off above at 6.7×10^{-3} Pa (0.004 code units) to increase the contrast of the waves outside of the plume

To simulate the coronal plume, the same definition can be used as the one for the coronal hole by changing the parameters in the `definitions.h` file.

When the wave front passes through the plume, it is reflected because of the large difference in density of the plume and its surroundings. This reflection is in turn reflected again on the other side of the plume, causing a resonating effect visible in fig. 5.5. This "captured" wave, generates secondary wave fronts, visible in fig. 5.4.

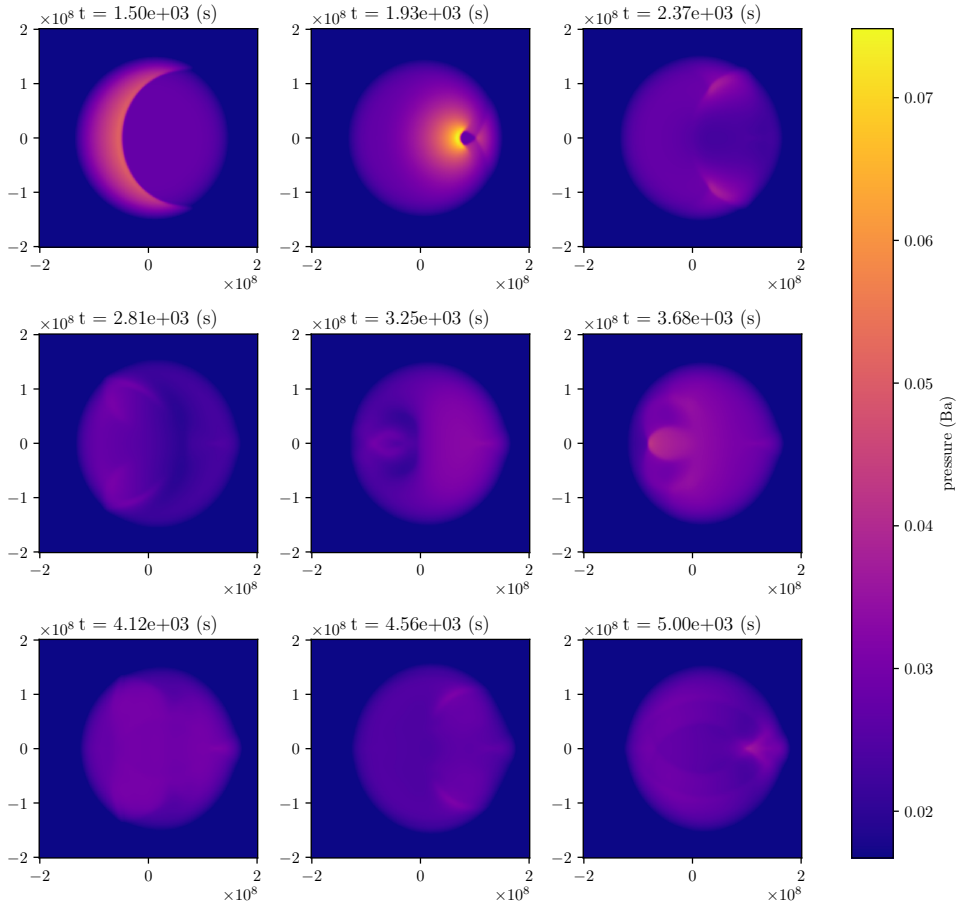

Figure 5.5: The within the coronal plume at multiple points in time. Note that the pressure is cut off below 1.6×10^{-2} Pa (0.01 code units) to increase the contrast of the waves inside of the plume.

6 Summary and Conclusion

In this paper, several different hydrodynamic and magnetohydrodynamic waves were simulated using PLUTO, which were then analysed by comparing the results to theoretical models. Propagation of the three main kinds of waves was studied in a uniform environment both with various magnetic fields and in the absence of magnetic fields. The interaction of waves with models of large scale structures, hole and plumes, commonly found on the solar corona were modelled and the simulations discussed.

References

[Ach90] D. J. Acheson. *Elementary Fluid Dynamics*. Oxford Applied Mathematics and Computing Science Series. Oxford University Press, USA, 1990.

[AZ18] AN Afanasyev and AN Zhukov. "Propagation of a global coronal wave and its interaction with large-scale coronal magnetic structures". In: *Astronomy & Astrophysics* 614 (2018), A139.

[BB17] Allan Sacha Brun and Matthew K Browning. "Magnetism, dynamo action and the solar-stellar connection". In: *Living Reviews in Solar Physics* 14.1 (2017), p. 4.

[Fày16] Robertus von Fày-Siebenürgen. *MHD Waves*. http://www.robertus.staff.shef.ac.uk/gian/mhd2.pdf. 2016.

[Fin07] Christopher Finlay. "Alfvén Waves". In: *Encyclopedia of Geomagnetism and Paleomagnetism*. Ed. by David Gubbins and Emilio Herrero-Bervera. Dordrecht: Springer Netherlands, 2007, pp. 3–6. ISBN: 978-1-4020-4423-6. DOI: 10.1007/978-1-4020-4423-6_3. URL: https://doi.org/10.1007/978-1-4020-4423-6_3.

[Fit11] Richard Fitzpatrick. *MHD Waves*. https://farside.ph.utexas.edu/teaching/plasma/lectures1/node65.html. 2011.

[Gop+09] N Gopalswamy et al. "EUV wave reflection from a coronal hole". In: *The Astrophysical Journal Letters* 691.2 (2009), p. L123.

[GP04] JP Hans Goedbloed and Stefaan Poedts. *Principles of magnetohydrodynamics: with applications to laboratory and astrophysical plasmas*. Cambridge university press, 2004.

[Gri17] David Jeffery Griffiths. *Introduction to electrodynamics*. 4th ed. Cambridge: Cambridge University Press, 2017. ISBN: 9781108420419.

[Kar05] Vladimír Karas. "An introduction to relativistic magnetohydrodynamics I. The force-free approximation". In: *RAGtime 6/7: Workshops on black holes and neutron stars*. 2005, pp. 71–80.

[Lyu14] Ling-Hsiao Lyu. "Linear Waves in the MHD Plasma". In: *Elementary Space Plasma Physics Second Edition*. Taipei: Airiti Press, 2014, pp. 91–92. URL: http://www.ss.ncu.edu.tw/~lyu/lecture_files_en/lyu_SPP_Book_A4format_pdf_html/pdf_1_Ch/lyu_SPP_Chapter_6.pdf.

[Mig+11] Andrea Mignone et al. "The PLUTO code for adaptive mesh computations in astrophysical fluid dynamics". In: *The Astrophysical Journal Supplement Series* 198.1 (2011), p. 7.

[Mig+18] Andrea Mignone et al. *PLUTO User's guide*. http://plutocode.ph.unito.it/documentation.html. 2018.