Universitat de les
Illes Balears

Facultat de Ciències

Memòria del Treball de Fi de Grau

Dinámica de gotas de lluvia coronal

Aleksandar Rasevic Lukic

Grau de Física

Any acadèmic 2017-18

DNI de l'alumne: 49922396E

Treball tutelat per Ramón Oliver Herrero
Departament de Física

S'autoritza la Universitat a incloure aquest treball en el Repositori Institucional per a la seva consulta en accés obert i difusió en línia, amb finalitats exclusivament acadèmiques i d'investigació	Autor		Tutor	
	Sí X	No	Sí X	No

Paraules clau del treball:
Coronal rain, Magnetohydrodynamics (MHD), PENCIL CODE, Thermal Non Equlibrium (TNE)

Coronal rain dynamics

Aleksandar Rasevic Lukic

September, 2018

Contents

Overview

Our Sun has proven to be an endless supply of interesting physical problems, which have required the aid of about every other branch of physics to find their answer. In relatively recent years, with the aid of spatial telescopes and new observation techniques, it has been found that even the seemingly calm quiescent sun is in fact full of activity. Even so, in this work we turn our attention to one of the oldest acquaintances of solar physicists: the impressive active sun, with its sunspots, flares, prominences and coronal loops. We will expose the existence of a very interesting phenomenon that takes place in the extremely hot solar corona, tightly linked to the flamboyant magnetic activity of the active sun: coronal rain.

Coronal rain involves a cyclic process in which heated plasma fills the arcade of a coronal loop, and, interestingly enough, as a result of such uneven heating, which only takes place in the feet of the coronal loop, a process called Thermal Non-Equlibrium (TNE) takes place, cooling down the plasma as it finds itself at the top of the arcade, falling down back to the sun's chromosphere.

Having understood what coronal rain is and how it is produced, we will get to our main concern: to describe the dynamics of a falling blob in the context of the Magnetohydrodynamic (MHD) equations, with the aid of the numerical analysis tools described below: the higher order finite-difference code PENCIL and the visualization program IDL. We will extract information and compare our results with previous work on this field, and attempt to extend the work of Oliver et al. (2014) to the 2D case.

Chapter 1

Introduction to coronal rain

In this section, we expose and discuss the theoretical background and motivation for the work at hand. We will begin by inspecting the most obvious structures of the active sun: sunspots and coronal loops. It is precisely in these regions that coronal rain takes place. Then, the process of TNE is explained in the measure that it is needed to understand the work at hand. Special emphasis is made in sections 1.4 and 1.5, which are, all in all, the main concern of this work: the magnetohydrodynamics governing the fall of a plasma condensation from the top of a typical coronal loop and a summary of previous work on this problem, drawn from Oliver et al. (2014), respectively.

1.1 Magnetic structure of active regions

We cannot emphasize enough the importance that the magnetic field has in solar physics. The Sun is made of plasma, and as such, it can be thought of as a fluid made up of electrical charges in the form of ions, protons and neutrals. Following this assertion, the myriad of phenomena that we can study in the active sun involve, in one way or another, the ways in which solar plasma is being altered by the underlying magnetic structure and its evolution.

Currently, the origin of this magnetic field is accounted by dynamo theory, and is formed in the depths of the convective zone of the Sun as large and coherent flux bundles. The way in which these magnetic bundles rise from the convective zone has been, of course, an interesting subject of study. The mechanism of magnetic buoyancy (Parker (1955*a*)) had been proposed as a possible explanation by which such bundles could rise from the convective zone in the so called Ω-shaped loops. This idea was later rejected, as it actually proved way *too* effective and put constraints on theories thought to be more fundamental; even so, it was brought up again by later authors (Acheson (1978)) as means to a more sophisticated mechanism of instability onset, reminiscing of a Rayleigh-Taylor instability in which a heavier fluid is suspended above a lighter one; as in this case the magnetic field is able to sustain more mass than it would be possible in its absence. We present here this idea of instability as opposed to that of non-equilibrium, which will be explained in later sections.

As these bundles of magnetic flux rise up from the convective zone into the photosphere, X-ray bright points of around 10 Mm in diameter appear in the corona. These are the so-called 'ephemeral active regions', which typically disappear within a few hours. It is possible that one such bright point swells into an 'active region', whose typical size is around 100 Mm and a magnetic field of 100 G or so. As these flux bundles are fragmented, other phenomena become visible, e.g., flux tubes, pores and sunspots, to name a few.

Let us focus now our attention in sunspots. They appear as dark regions in the photosphere which have been observed since ancient times, since they can even be seen with the naked eye. Nowadays, we know that sunspots are basically large concentrations of magnetic flux, with a wide range of shapes and sizes, ranging from 3.5 to 60 Mm, and, accordingly, their lifetimes vary from typically 2 to 3 days up to 80 to 90 days. They are dark due to the partial inhibition of normal energy transport as a result of their magnetic field, with two clearly distinguishable regions: the umbra, which accounts for about a

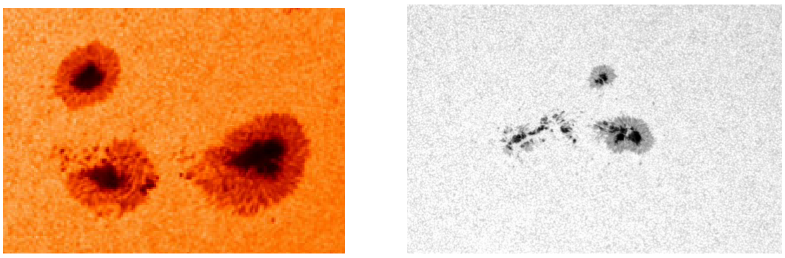

Fig. 1.: Left, a group of three sunspots. The umbra and penumbra zones can be clearly distinguished. Right, another group of sunspots merging. Both images have been taken from the official Solar Dynamics Observatory (SDO) gallery.

40 per cent of the spot radius, and the penumbra, which surrounds the umbra. Differences in brightness between umbra and penumbra are obvious, as the umbra radiates only around 30 per cent of the surrounding photosphere, while the penumbra gets to about a 75 per cent. Accordingly, differences in temperature are also present, as the umbra is around 1500 K cooler than the photosphere, while the penumbra is around 300 K cooler. Differences in the magnetic field are also noticeable, as the umbra exhibits something around 2.8 kG, and the penumbra has some 800 G. Upon closer observation, a very curious effect can be observed: a flow of material from the penumbra to the umbra. This is the so-called Evershed effect, and observations of this effect and its deviations (Beckers (1962)) helped determine the existence of downward material motions in sunspots. Coronal rain is a paradigmatic example of such flows.

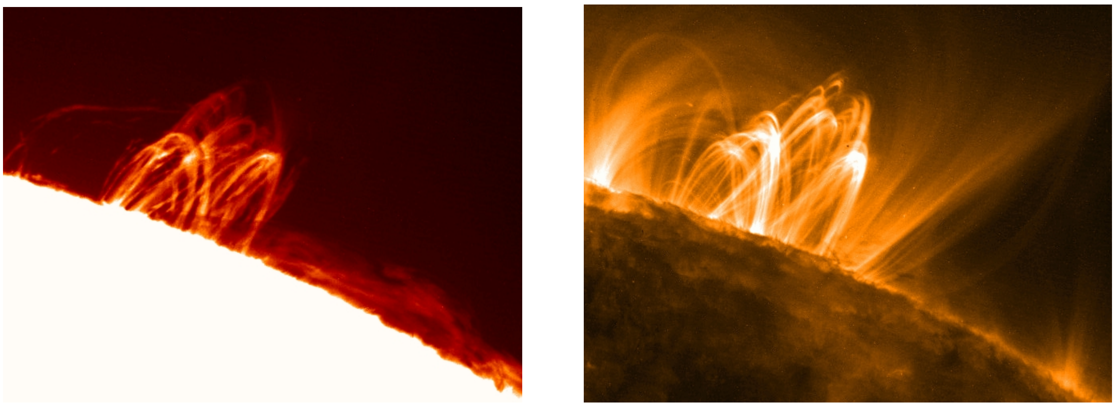

Fig. 2.: Left, a coronal loop as captured by TRACE, at 10000 K emission. Right, the same coronal loop, at 1 MK emission. Images taken from the TRACE archive at Stanford.

Sunspots are anchored in the photosphere. Regardless, their effects are notorious in the upper layers of the solar atmosphere and, among other phenomena, give raise to coronal loops. These coronal loops are but magnetic structures that connect different sunspots, or, more generally, any number of regions with opposite polarity, as there exist deviously complicated configurations that can give rise to these awe-inspiring arcades. Their length varies greatly with the kind of phenomena that has originated them: from around 1 Mm for X-ray bright point loops up to 1000 Mm for giant arches. A crude distinction between loops may be made on their temperatures: cool loops are some 10^5 K, around 10^6 K for warm loops in active regions and a few 10^7 K for flaring loops. Each of these loops emits in a different wavelength, which is what prompted such classification: UV for cool loops, EUV for warm loops, and soft X-rays for hot loops. The temperature of the coronal loop has a decisive impact on its lifetime: hotter loops are relatively steady, while warm or cool loops exhibit a variety of phenomena, such as cooling collapse or siphon flows. The magnetic fields in these structures go up to about 10 G. It is worth stating here that these loops are heated almost exclusively at their footpoints by the sunspots in which they are anchored, by a process yet to be fully understood, and the results of this uneven heating will be brought back later, as it plays a fundamental role in this work. The extraordinary phenomena that take place in these last examples will occupy the next section of this work: Thermal Non-Equilibrium (TNE).

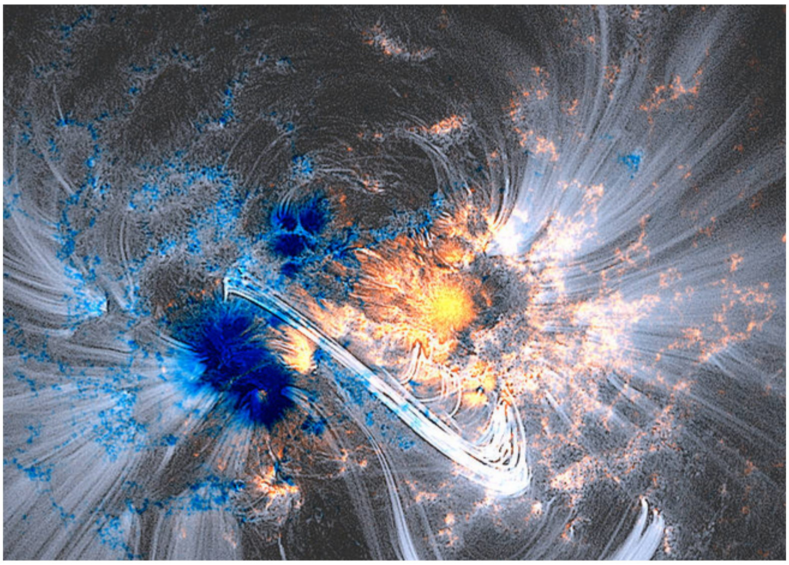

Fig. 3.: sunspots joined by a coronal loop, represented white. The colors of the sunspots represent opposite magnetic polarities. Taken from the official Solar Dynamics Observatory (SDO) gallery.

1.2 Thermal Non-Equilibrium

Thermal Non-Equilibrium, henceforth referred as TNE, was a concept introduced in the theory of solar dynamics to account for the seemingly contradictory existence of cool plasma in the solar corona and its coronal loops, which we have already mentioned is already of the order of a few MK. Experimental evidence was found for the existence of these 'plasma knots', blobs of material cooler than their surroundings (Antiochos et al. (1999), see references therein), at around chromospheric temperature to be able to find themselves in the 'condensed' state, even though, as we have stated, the temperature in the corona is usually of around a few MK, while the chromosphere is around 4000 K.

At first, it was thought that these knots were formed due to a failure of the heating mechanism of the coronal loops, which would then eventually form clumps of cool material which would fall down. Such mechanism was deemed incorrect, because of its privileged conditions and strong siphon flows, that would led to the immediate drainage of the material. As these authors were concerned with the formation of plasma knots inside of prominences, this was not an acceptable process, as these are structures able to last for days. In later work, it was proposed a seemingly contradictory mechanism: the plasma knots were formed because the material was getting *hotter*.

This assertion might be a bit too bold; let us rephrase this idea. It is not that the loop is 'cooling', rather, the very nature of the uneven heating that the loop is experiencing, as stated before, leads naturally to the process of condensation of these cold plasma blobs. The source of this heating is currently still under study; nonetheless, if we admit that coronal loops are subject to heating, a stratification of temperatures is then created, which in turn leads to the process of condensation. This hypothesis has been confirmed time and time again in numerical simulations (Antiochos & Klimchuk (1991), Antiochos (1999), Müller et al. (2004)). Even though this process was first studied under flare and prominence conditions, it has been put forth that even in quiescent, non-flaring conditions, non-uniform heating would lead to this process in coronal loops (Müller et al. (2004)).

We just need to put forth a final piece of data before we can tie up our explanation of TNE in a tidy form: solar plasma experiences radiation loss in a complicated way. One of the widely accepted models, which includes the ranges in which certain assumptions are acceptable, is provided by Rosner et al. (1978). We include the following figure:

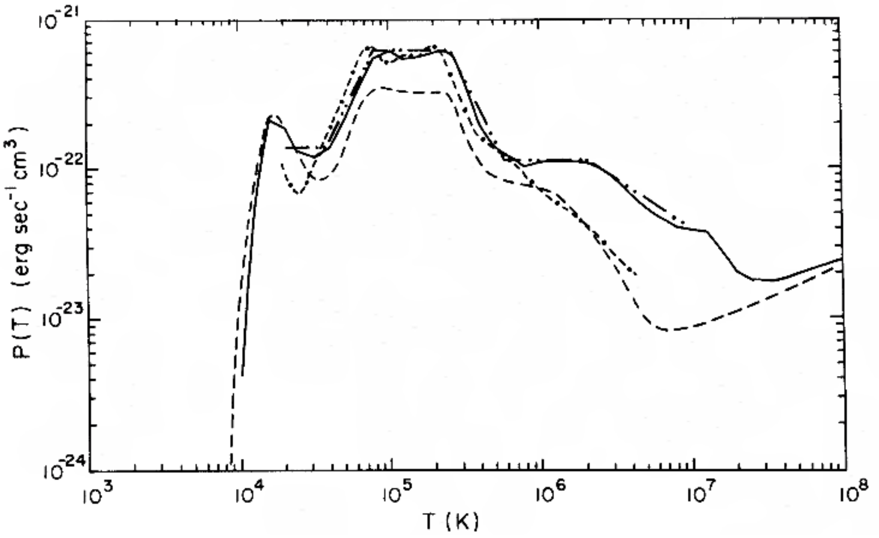

Fig. 4.: Radiative loss function for solar plasma. Courtesy of Rosner et al. (1978), Fig 10.

The important feature we should notice from such function, is that the radiative losses can go up by as much as an order of magnitude in the regime where we cool down from 10 MK to 1 MK.

Let us condense the general process of TNE, which is so elegantly put forth by Froment et al. (2017). The plasma found near sunspots is heated up by an unknown process. This plasma then 'evaporates', as in gets less dense that the plasma around it, and begins to fill the coronal loop. Logically, the more mass we have in our loop, the bigger the heating needed to sustain such mass: the loop has become denser. When the loop becomes too dense, the increased radiation losses caused by the high temperature and density of the loop material give rise to an imbalance between the energy input and the losses, hence the loop cools. Such temperature drop comes hand in hand with an increase in the radiation losses, as explained before, which results in even faster cooling. Thus, a dense blob forms and subsequently falls to the chromosphere. Notice that this phenomenon is cyclical, as once the loop loses enough mass, the energy needed to evaporate new material will be lower and thus attainable by the heating sources.

Thus, we have explained so far how is it possible that we have a knot of cool, dense plasma suspended at such height in the corona. In this work, we are concerned with the dynamics of such blob, but first, let us turn our attention to previous work on the field of coronal rain dynamics, and, at the same time, we will try to further polish our definition of this phenomenon.

1.3 Coronal rain

Coronal rain manifests itself as cold plasma knots that fall down to the chromospheric surface with an acceleration lower than the speed of free fall. Their temperature range allows them to be seen in neutral atom bands, such as $Ly\alpha$ or $H\alpha$ or in ionized elements, such as Ca II or He II, as pointed out by Oliver et al. (2014).

Early reports on this phenomenon date to Beckers (1962)) and others, who accounted for the existence of coronal rain and estimated a falling speed of around 40 to 60 km s^{-1}. Later authors, such as Kawaguchi (1970) and Leroy (1972) pointed out that indeed, such range of velocities could not be entirely due to the gravitational pull experienced by the blobs; their speeds would have been much greater had this been true. Detailed kinematics would still have to wait quite some time. Also, as pointed out above, for quite some time such condensations of plasma were thought to take place only in flare-like conditions, but simulations by Müller et al. (2004) ultimately proved otherwise: even in the calm quiescent sun, coronal plasma rain can take place as long as there exists a coronal loop in which TNE can form.

Antolin et al. (2010) and coetaneous authors found agreement in the simulations conducted and the observations made: such blobs formed at the apex of the arcade of a coronal loop, where they would begin falling with free fall speed. As they progressed along the loop's leg making their way to the chromosphere, the acceleration would stop, reaching falling speeds of around 150 km s^{-1}. A question remains: what is the source of this decrease in acceleration?

It is this question that was mainly addressed in the work of Oliver et al. (2014): the sole existence of the condensation of the blob leads to a rearrangement of the initially stable stratification of the atmosphere, creating a pressure gradient that acts opposite to gravity. Such gradient eventually equates the force of gravity, resulting in constant falling speeds.

To aid in the visualization of this astonishing phenomenon, we provide two movies, which can be visualized *here* and *here*. In the first movie, we see the surface of the sun on the Hα spectral line, which is relatively cold enough, revealing the presence of material at chromospheric temperatures, in contrast to the much hotter corona. The Doppler velocity of these blobs is against their background. In the second video, we notice how material suddenly 'appears' at some height above the chromospheric surface. As we know, it is not that it was not there, it just means that the plasma was at a much higher temperature, emitting radiation that could not be collected by the filter used. As the plasma cools down, it starts emitting in other spectral lines and its light can be seen.

Having commented on the atmospheric stratification of the Sun's corona, let us briefly expose the equations that govern charged fluids in presence of magnetic fields: Magnetohydrodynamics, from whose viewpoint we will try to derive equations that account for the state of the corona and its evolution.

1.4 Magnetohydrodynamics

Magnetohydrodynamics, henceforth MHD, is a theory that deals with the motion of charged fluids in the presence of magnetic fields. In its crudest form, it can be thought of a non-relativistic mixture of the Maxwell equations and the continuity equations of an ideal fluid, even though extraordinarily more complex (and, arguably, more mathematically justified) derivations exist, such as those found in the relevant chapters of *Principles of Magnetohydrodynamics*, by Goedbloed & Poedts. Nonetheless, we will stick to the simplest derivation and further simplify whatever is needed to justify the model we use, under, of course, more than reasonable assumptions.

We can write down the Maxwell equations as:

$$\nabla \wedge \vec{B} = \mu \vec{j} + \frac{1}{c^2}\frac{\partial \vec{E}}{\partial t} \tag{1.1}$$

$$\nabla \cdot \vec{B} = 0 \tag{1.2}$$

$$\nabla \wedge \vec{E} = -\frac{\partial \vec{B}}{\partial t} \tag{1.3}$$

$$\nabla \cdot \vec{E} = \frac{\rho^*}{\epsilon} \tag{1.4}$$

Vacuum values for the constants are widely accepted. All quantities represent their usual meanings. The plasma is assumed to be approximately electrically neutral. This seems reasonable, taking into account that given the order of magnitude of the particle densities found in the Sun, charge imbalances would result in tremendous electrical fields, which are not present in reality.

We add Ohm's law to this mix, under the form:

$$\vec{j} = \sigma \left(\vec{E} + \vec{v} \wedge \vec{B} \right) \tag{1.5}$$

Ohm's law is possible and even necessary to account for some phenomena, but this is outside of our region of interest, both theoretically and physically, as in the corona we have no such worries, and Ohm's law will be used just as a means to connect the current density and the electric field, as we shall see later.

The ideal fluid equations we will be managing take the form of:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \vec{v}) = 0 \tag{1.6}$$

$$\frac{\partial \vec{v}}{\partial t} + (\vec{v} \cdot \nabla)\,\vec{v} = -\frac{1}{\rho}\nabla p + \vec{g} + \frac{1}{\rho}\left(\vec{J} \wedge \vec{B} \right) \tag{1.7}$$

$$\frac{\partial p}{\partial t} + (\vec{v} \cdot \nabla)\,p + \gamma p \nabla \cdot \vec{v} = 0 \tag{1.8}$$

which is the usual system of equations that describe mass, balance and energy evolution in time. At first glance, the key point in common between both systems of equations supplemented seems straightforward: magnetic field. A further requirement, or rather, a further supposition, as mentioned above, is that of non-relativistic motions. If we suppose that the electromagnetic variations are in this regime, a great deal of simplification takes place: first, simple numerical analysis tells us that the displacement current has magnitude:

$$\frac{E_0}{c^2 t_0} \approx \frac{V_0^2}{c^2} |\nabla \wedge \vec{B}| \tag{1.9}$$

which, by virtue of non relativistic motion, is immediately much smaller than the first term of equation 1; therefore, it can be neglected. We will also assume that the plasma has high conductivity, so that $E \approx 0$

By taking the divergence of equation 1, we would obtain the well known law of conservation of current, which in front of no displacement currents, is 0, which in turn implies that local accumulations of charge are negligible.

We now turn our attention to the case at hand: a blob falling along a coronal loop leg. As long as coronal plasmas are involved, the so-called β parameter of the plasma is low. This value correlates the magnetic pressure, given by $p_{mag} = \frac{B^2}{2\mu_0}$, and the gas pressure, supplied in our case by the ideal gas law, $p_{gas} = 2\rho RT$, with $R = k_b/m_p$ and their usual meanings. As such, $\beta = p_{gas}/p_{mag}$ tries to account for the measure by which the plasma is affected by the magnetic field. In the solar corona, $\beta << 1$, so that even if we are dealing with magnetic fields of the order of 10G or less, the plasma will stick to the magnetic field lines strongly. This explains why coronal rain blobs delineate magnetic field lines as they fall.

In turn, as the movement of the particles follows the magnetic field lines, we can immediately go check our equations above, and notice that terms that involve the wedge product between the magnetic field and the velocity of the plasma element will be immediately 0, as they parallel one another.

Now it is about time to study the coronal rain motion. For the 1D case, which we will firstly analyze, we can follow the derivation of Oliver at al. (2014); whose final assumption for this case is that the set of variables with which we are concerned depend so strongly on t and z, that the movement is effectively confined on the z-axis. This in turn leads to a rewriting of equations 6 to 8, resulting in:

$$\frac{\partial \rho}{\partial t} = -v\frac{\partial \rho}{\partial z} - \rho\frac{\partial v}{\partial z} \tag{1.10}$$

$$\rho\frac{\partial v}{\partial t} = -\rho v\frac{\partial v}{\partial z} - \frac{\partial p}{\partial z} - gp \tag{1.11}$$

$$\frac{\partial p}{\partial t} = -v\frac{\partial p}{\partial z} - \gamma p\frac{\partial v}{\partial z} \tag{1.12}$$

Following Oliver et al. (2014), to determine the state of the initial state of the loop we ignore temporal variations and assume $v = 0$ in equations 10 through 12. This leads to two exponential solutions, whose form clearly indicates the stratified nature of the field we are working in, in density and in pressure. These exponential solutions correspond to equation (11) of the same paper, where the derivation has shown that:

$$p\left(z, t = 0\right) = p_0 e^{-z/H}, \rho\left(z, t = 0\right) = \rho_0 e^{-z/H} \tag{1.13}$$

where H is the vertical scale, which in turn depends on the equilibrium temperature as $H = 2\frac{RT_0}{g}$, and $p_0 = 2\rho_0 RT_0$, which follows from the state equation written beforehand.

We shall not make here an explicit derivation of the equations that define the dynamics of a 2D blob, as nothing would be gained from this. It suffices to say that it would get much more difficult to solve, as new spatial dependences and derivatives would arise. It suffices to say that this system will be carried over by our code, hence, no analytical expression will be sought.

One last point should be clear from now: the equations presented in this section describe an equilibrium atmosphere, no dynamics should be present. What we will do, again following previous work from Oliver et al. (2014) is forcefully input a blob with given characteristics, which will perturb this equilibrium state. Then, we follow the perturbation to derive the desired results.

1.5 Summary of Oliver et al. (2014)

Throughout this introduction to the theory behind coronal rain and its dynamics, we have invoked several times the work of Oliver et al. (2014), though this information has been scattered through the entirety of the paper at hand and only when it was relevant to be mentioned. Considering that our work is, all in all, a different approach to a very similar problem and a possible extension of the results in there obtained, it seems fitting to write a brief summary of their work, which will also aid in understanding the main similarities and differences.

Henceforth, we will refer to Oliver et al. (2014) simply as O2014; even though extensions of this work exist concerning the coupling of neutral charges to the protons and ions, (Oliver et al. (2016)), it is to be understood that we are drawing most of our information from the 2014's paper.

The main concern of O2014's work is to study the dynamics of a cold plasma condensation upon which is acting the gravity force. The equations for this movement have been presented as equations 10 to 13 of this very same work, in the section labeled 'Magnetohydrodynamics'; solutions for this system are also included. Knowing the coronal stratification, a condensation is forcefully introduced, modeled as:

$$\rho_b\left(z, t = 0\right) = \rho_{b0} exp\left[-\left(\frac{z - z_0}{\Delta}\right)^2\right] \tag{1.14}$$

where ρ_{b0} is the central mass density of the blob, z_0 accounts for the initial height and its length is of order 2Δ. As the total density is now the total sum of the density of the loop in which the blob appears and that of the blob, but such sum is not accounted by the loop pressure, as it only takes for granted the loop's density, the mechanical equilibrium is broken from the very beginning.

Numerical calculations are performed using the PDE2D code cited in the paper, and measurements of the position, total density, height and velocity are taken, as well as the acceleration evolution and the distribution of the plasma pressure. They take place in a range of heights of $0 \leq z \leq 60$ Mm, and the blob is released at an initial height of 50 Mm, using diverse numerical domains in which these integrations are performed, which are larger than the range in which the dynamics are explored because, as pointed out in O2014, the breakage of equilibrium by the sudden insertion of the blob density leads to a pressure gradient that translates to the emission of two sound waves from the blob. Numerically, these sound waves or perturbations could bounce back on the upper and lower bounds of the numerical domain, leading to interaction with the blob that is still falling as these sound waves are much faster than the blob speed. This effect is absolutely unphysical, as normally the sound wave would just be absorbed by the chromospheric surface. As to avoid this, the numerical domains are expanded as to ensure no autointeraction or other unphysical events.

These calculations are explored in mainly two cases whose only distinction is the density involved; which in turn has great effect on the dynamics of the blob and the speed at which the phenomena that we will now cite take place. The base parameters are $T_0 = 2x10^6$ K as the corona equilibrium temperature temperature, so that $H \approx 120$ Mm; this is, the vertical scale at which an important variation of the parameters presented upwards takes place, namely the stratus pressure and density. The density at the coronal base is $\rho_0 \approx 5x10^{-12}$ kg/m^3; and the base pressure is 0.165 Pa. The two densities explored are $\rho_{b0} = 10^{-10}$ kg/m^3 and $\rho_{b0} = 10^{-9}$ kg/m^3. The blob elongation is $\Delta = 0.5$ Mm.

Results are clear from these simulations: a pressure gradient is formed and the two sound waves are emitted. The pressure gradient (Fig. 6, (b)))gives raise to a force that opposes gravity, (Fig. 6, (a)), effectively diminishing the effects of gravity on the plasma knot, after a phase of free-fall speeding (Fig. 5, (c)). The inertia term of the fluid equations presented plays little to no role in the smallest density. To sum up the graphic behavior, we show these figures for the knot of lesser density:

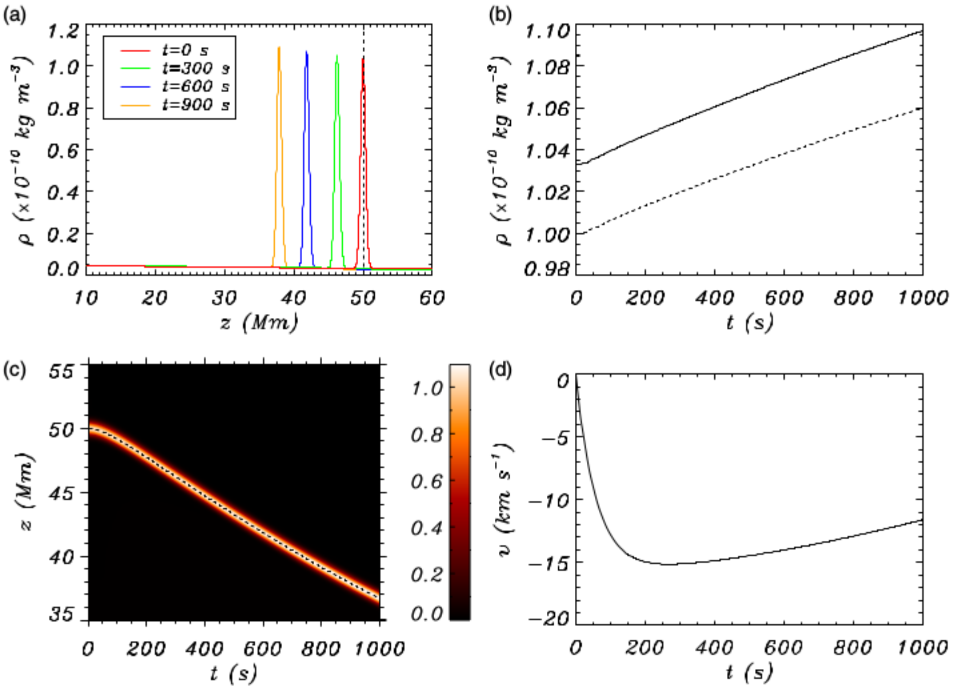

Fig 5.: (a) Position of maximum density, (b) maximum density value over time, (c) position evolution and (d) velocity evolution for a blob of density $\rho_{b0} = 10^{-10}$ kg/m^3. Courtesy of Oliver et al. (2014).

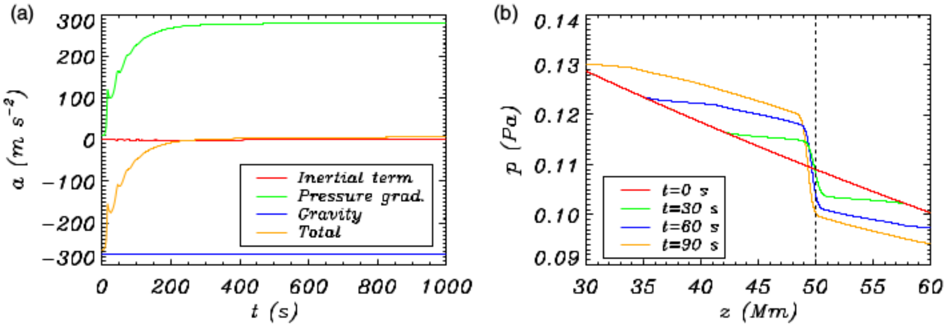

Fig 6.: (a) Acceleration evolution, (b) pressure distribution evolution for a blob of density $\rho_{b0} = 10^{-10}$ kg/m^3. Courtesy of Oliver et al. (2014).

If we amp up the density to the second value, ten times larger than the first experience, similar results appear, this time though, they take longer to appear. The acceleration phase takes longer, and so the speed of the blob is significantly bigger. From the results of O2014 we can conclude a few things: the inertia term of the acceleration presents oscillations, that are due to the emission of sound waves. Notice from the system at hand that the inertia term depends on the product of the velocity times its derivative. In the case at hand, the velocity is significantly bigger, which leads to a higher inertia influence. The pressure gradient starts to act later in the simulation, mainly due to two reasons, as argued by O2014: first, the pressure term depends inversely on the density. An increase of a factor of 10 reduces the effect of this term by a factor of 10. Secondly, the limits of the pressure gradient are limited by the heights perturbed by the sound waves, which in turn depends on the depth these sound waves have achieved. The slower they go, the smaller the step, resulting in a slower formation for the gradient. From this second experience, O2014 obtained the following figures:

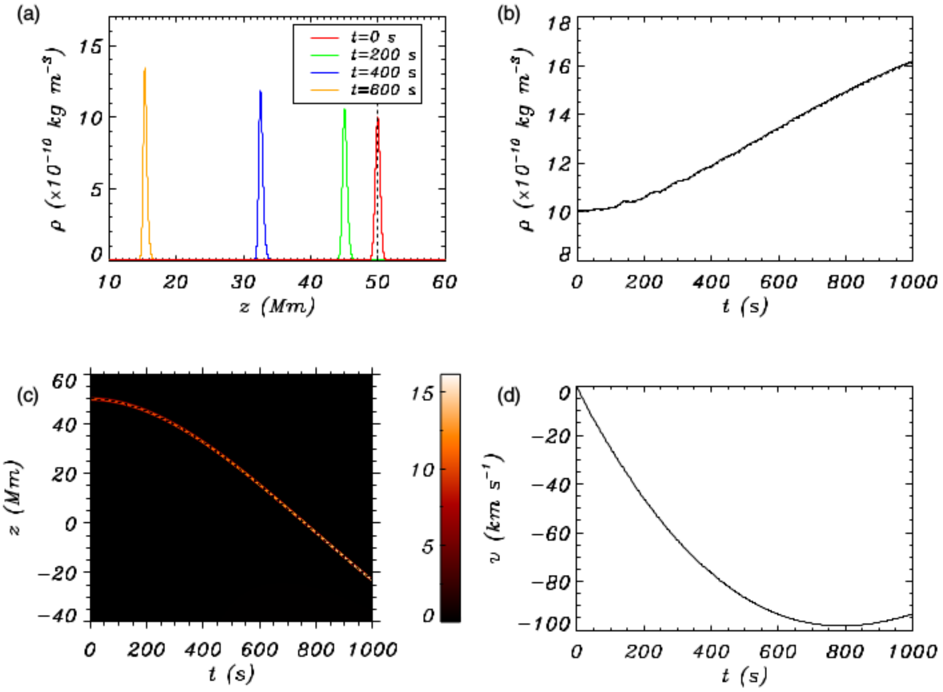

Fig 7.: (a) Position of maximum density, (b) maximum density value over time, (c) position evolution and (d) velocity evolution for a blob of density $\rho_{b0} = 10^{-9}$ kg/m^3. Courtesy of Oliver et al. (2014).

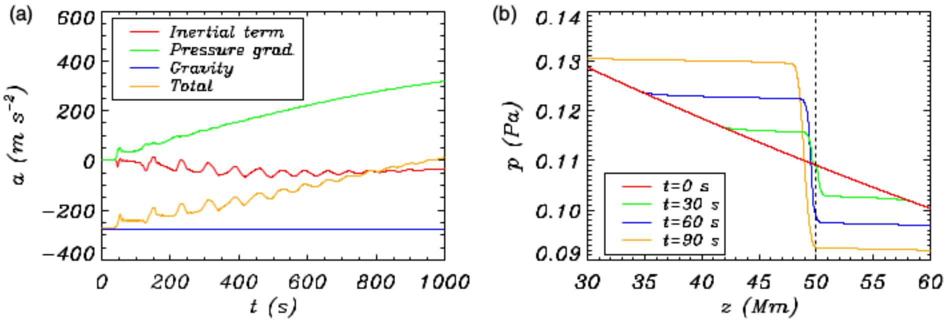

Fig 8.: (a) Acceleration evolution, (b) pressure distribution evolution for a blob of density $\rho_{b0} = 10^{-9}$ kg/m^3. Courtesy of Oliver et al. (2014).

1.6 Aim of this work

In our work, we want to reproduce some of the results obtained by O2014 using a different numerical code with a twofold purpose: first, to test the results obtained by O2014 in a different code, which in itself would be further proof of the strength and conclusions derived from this study, and secondly, we are interested in checking if the code used, PENCIL CODE, is good enough to run viable 2D simulations, as the dimensionality is quite restricted in the code used by O2014. Of course, we would also like to check if the extra dimensionality has any significant effect on the simulations at hand or if it works as expected from the theory, with the assumptions that have been explained above.

For this purpose, we will first try to reproduce the height evolution and velocity evolution for the same equilibrium conditions and the same blob densities that O2014 used in their article.

Second, we will run simulations in a 2D magnetohydrodynamic scenario and will derive conclusions from these results and try to tie them in with the theory exposed thus far. For example, we will check if the blob deformation takes place as it is expected from the 1D experiments. This effect is, as O2014 points out, no mere curiosity: in a partially ionized plasma this effect could lead to recombination or ionization of the flow of material on the blob, which would add an extra layer of complexity for the coupling of neutrals and charged particles. This is of special interest if one considers partially ionized plasmas; let us remember here that we are always working with fully ionized plasmas in this work.

Chapter 2

Numerical and analytic tools

In this chapter we will present the numerical tools that have been used to conduct our analysis and derive our conclusions, we use a combination of the numerical code PENCIL CODE and the visualization code IDL. We will explain what these tools are and will give a brief introduction to their functioning as well as their limitations and strengths. We will also explain how a typical execution of the could would run.

2.1 PENCIL CODE

PENCIL CODE, henceforth simply Pencil, as written in their webpage, is a code originally developed at the Turbulence Summer School of the Helmholtz Institute in Potsdam in 2001. As per popular request, it was released to the public in late 2008 via github. It is a code designed to solve hydrodynamics and magnetohydrodynamics problems using a higher order finite-difference method. Its original use is focused on dealing with weakly compressible turbulent flows. The kind of problem Pencil is designed to deal with involves, for example, problems such as driven MHD turbulence in a periodic box, convection with non periodic boundary conditions, etc. If the problem has to deal with magnetohydrodynamics, the magnetic field is implemented through the well known magnetic potential, which ensures the divergence-free condition. Even though we had no use for such data in our conditions, Pencil developers claim that their code is especially advantageous were one to deal with magnetic helicity in their simulations.

The name Pencil is derived from the way in which the calculations take place: an entire time-step is calculated along a one-dimensional array of data, all within the grid of the problem at hand. It is claimed by the coders that this is made to ensure maximum efficiency of the caché memory of the computer.

Pencil code is highly modular, but we should not understand the name module in the same way as in the one that coders familiar with Fortran should. The modules involved in Pencil represent what conditions will our run have. As such, for example, the module 'magnetic' will tell our simulation that it should fetch the files necessary to be able to run a simulation in which there is a magnetic field. On the contrary, one must specify the module 'nomagnetic' for the obvious instance. This capacity to switch on and off modules as desired is one of the most user-friendly features that Pencil has, as a quick modification, as will be explained later, can easily set our desired parameters.

The equations solved by Pencil code are, as claimed by the users manual, in which exists an exhaustive list of the systems of equations involved, the standard MHD compressible equations. The switching on and off of modules, as mentioned earlier, will accordingly modify these equations, and of course, it is expected that if the code is able to run on the full-blown equations it will be able to do so with simpler versions.

Finally, data are stored in what the developers have come to call 'video files' or 'slices'. These are nothing but files that contain the time series of values of a variable in a given plane, and their output frequency may be modified at will, as we will comment later on.

2.2 Typical execution

In this section we will illustrate a typical execution of the Pencil code. For this, a total of six files have been used every time: these files are 'Makefile.local', 'cparam.local', 'run.in', 'start.in', 'video.in' and 'print.in'.

Just to grasp a feel of the hierarchal structure of the code, we include an image provided by the Pencil code manual:

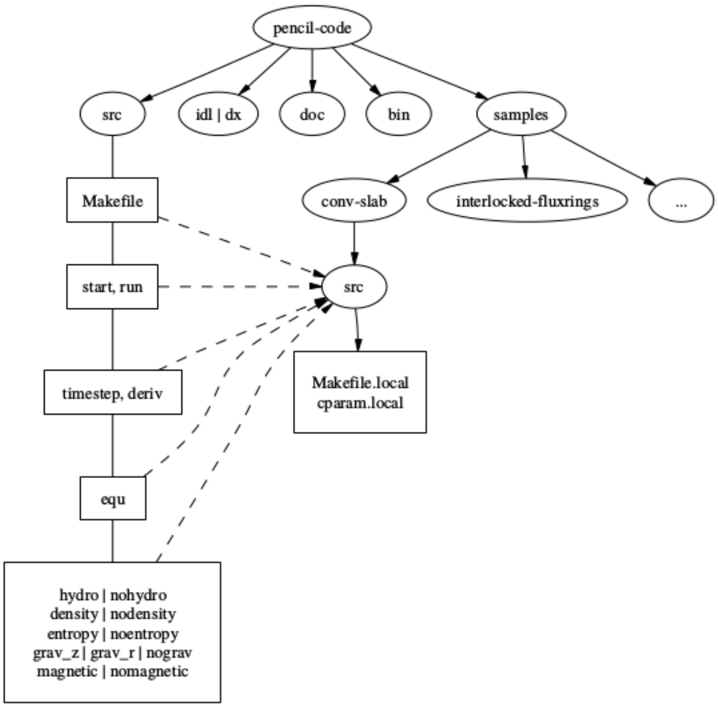

Fig. 9.: The hierarchy tree of the Pencil code. Courtesy of pencil-code.nordita.org.

First, one should define a source folder, henceforth named src. In this folder, two files are allocated: 'Makefile.local' and 'cparam.local'. In Makefile.local we control the switching off or on of the modules involved in an execution, as well as the precision of our run, whether it be simple precision or double precision. In cparam.local one controls the numerical grids, defining the number of processors as well as the number of grid points in any direction desired. Notice that Pencil code is highly parallelized and originally meant to be run on supercomputers, where a large number of processors could be available at any given moment. We have not used this feature in our case.

The other four files are allocated outside this src folder, and each one regulates different aspects of our simulation: 'start.in' has all the data referring to the initial conditions and parameters of our run. From here we can modify, for example, the knot's initial position or its initial density, for example. In 'run.in', we address more abstract concepts that regard the numerical bounds of our simulation. One can define the shape of our domains, whether there exist periodic boundaries, as well as the rate in which information is stored, for example, or the magnitude of the units used in the definitions. The last two files simply deal with the way in which the information is shown. In 'print.in', we can chose which measures are displayed and in what format as the simulation goes on; in 'video.in' we can select which variables are to be stored in the slices mentioned above. Notice that in the users manual, comprehensive lists of all the possible variables exist, as well as their meanings. Finally, we create a symbolic link to a folder (this is done out of convenience, one could just put this folder inside the run directory) which contains the initial conditions for the blobs at hand. In our case, these were .f90 programs which contained the necessary instructions to initialize the simulation.

To run the code, it is only needed then to have the Pencil code installed on one's computer, as well a Fortran compiler, a C compiler and Perl, which is, by any standard, pretty standard. Typical execution would go as follows: create a directory in which the files are allocated, create the src folder, and set the desired parameters in Makefile.local an cparam.local. Outside src, modify as needed the other four files. Then, we need to 'setup' the source.

By setting up the source, we mean to link the necessary files which are located elsewhere by the Pencil code. This is done almost automatically by the Pencil scripts supplemented by the code; the instruction 'pc_setupsrc' will suffice. If no errors appear at this stage, one would continue with the compilation by means of the script 'pc_build -f Intel', which is the compilation instruction used in case one has Intel compilers, for example. Multiple variants of this instruction are possible depending on the compilers at hand, as well as the option to further specify '_MPI' to deal with multi-processor executions. Finally, all that remains is to create a data folder in this directory and run the script 'pc_run'. So, once one has some understanding of the underlying code, a typical execution would take about four lines of text input by the user, provided one has already made all the necessary modifications and proper setup has taken place.

The code will then run and output the data specified in 'print.in' at a rate specified in 'run.in', and will save slices as specified in 'video.in'. Finally, in the folder data, all the information regarding the slices is stored, which then is awaiting treatment to be interpreted.

2.3 IDL

IDL is a scientific programming language used for data analysis. It is a vectorized, numerical and interactive language used mainly for the processing of large quantities of data. Taking into account that the heavier simulations we are dealing with occupy around 40 Gb of disk space, its use seems more than justified.

IDL accounts for Interactive Data Language, and its syntax is similar to that of C and Fortran. Pencil code routines can be invoked so that they may be included to build up routines to retrieve specific kinds of data. For example, Pencil code does not account for the 'position' of a blob, as this concept seems kind of strange when talking about a fluid. We have then the possibility of running a routine in an IDL program which locates the position of the maximum of the density distribution according to the data that actually is retrieved by the Pencil code.

Therefore, data visualization is run completely by running prepared scripts that retrieve the data needed for our outputs, or, in other cases, try to optimize the large data files we are managing.

Having explained the analytical and numerical tools used in our program, it is time to expose the results from our simulations.

Chapter 3

Results

In this section we will study the cases proposed in section 1.6 and relate them to the study cases and theory derived by the work of Oliver whenever possible. Discrepancies and comments related to the efficiency or validity of our results will be saved up for the next section, 'Conclusions', and we will focus primarily here on exposing our results and showing graphs and numerical data concerning them.

Physical and simulation parameters will be explained in each section as needed.

3.1 1D case

This is a 1D execution in which the parameters are the ones that have been explained for the case of Oliver further above. Just as a reminder, we will explain them again here: initial conditions are given by solving the system of equations that describes the motion of the blob, under assumption of low β and vertical motion that takes place fully in the z-axis. Then, the physical data of importance are: acceleration of gravity in the corona, $g_\odot = -274 m/s^2$, plasma knot density, $\rho_{b0} = 10^{-9} kg/m^3$, plasma knot elongation $\Delta = 0.5$ Mm, initial blob position, $z_0 = 50$ Mm. Also, the dimensional bottom density is supplied as $\rho_0 = 0.05\text{x}10^{-10} kg/m^3$ and sound speed of the waves emitted is also supplied as $cs0 = 234.58328 km/s$.

As for the run parameters, we chose one of the grids proposed by Oliver: we will study the dynamics of the blob in the regime $0 \leq z \leq 60$ Mm, whereas the numerical domain comprises $-200 \leq z \leq 250$ Mm with a number of grid points in this direction of $nzgrid = 50000$. This results in a parameter dz of $9km$. Finally, we further supply an end of simulation time of $950s$. The rate at which we collect the data is set to be of one slice per second of simulation. We will discuss possible computational limitations further below. Then, we run the simulation and collect the data.

We desire to analyze the graphs depicting the time evolution and velocity evolution of this blob. We find:

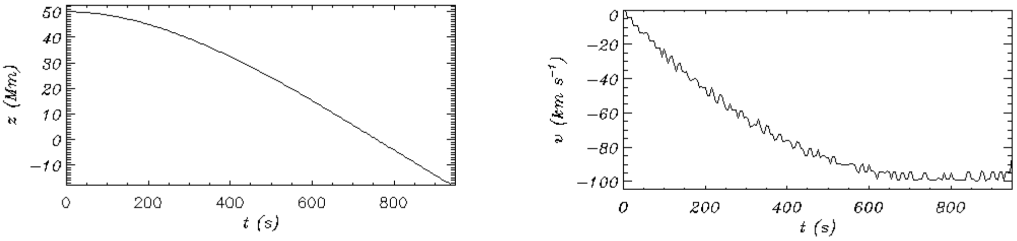

Fig 10.: (a) Position evolution of the blob with the described specifications and (b) velocity evolution for the very same blob. Obtained via IDL.

We remark here some details about the graphs here displayed. First, let us take our attention to the first of these two graphs: the position evolution. If we compare our figure to figure 3c of this very work, which had been obtained previously from Oliver as stated therein, we can see that our results greatly coincide, as was to be expected. Just as a matter of visuals, we can notice that a fall of about 50 Mm has taken about 800 s. Turning our attention to the second graph, the velocity function, things do not seem as clear. For one second, let us not pay attention to the oscillations that the function is suffering and just concentrate on the shape of the slope. If one is willing to ignore this for a second, the slopes of the graphs presented here and the one presented in figure 3d of this work are also very similar: indeed, one can appreciate a slope that is getting less steep as time goes by. One should take into account this fact, for as it was explained in Oliver and related work, we are observing the effects of the slowly building up gradient of pressure acting on the blob, which cushions the effects of gravity. As this takes place rather slowly due to effects already described such as the speed of the sound waves emitted or the factor of then in the density described in its corresponding section, gravity is nonetheless reduced until we reach a quasi-constant fall speed at the mark of about 650 s.

If we wish to investigate the cause of such wobbles in the velocity, we must understand how calculations take place using the Pencil code and IDL. As defined in its section, position is not such a well-defined quantity when we are using this code, as it deals mostly with flows, and as it is known, flows are better spoken of in terms of averages and so on. As such, we've defined a simple script whose solution consists of computing where the maximum density is located at each time. All in all, we are working with a discrete grid, which can entail the problems we are seeing here. If the maximal density found in the grid points available is not the actual maximal density in the time step were the grid continuous enough, the correct value will not be selected, resulting in an incorrect velocity. This problem of course could be addressed by continuing to make the grids smaller, but we also have to consider efficiency when dealing with the code and simulations we are trying to partake in. Let us then accept such wobbles as a purely computational effect that is due to the way the code acts rather than some real physical meaning in clear difference to the wiggles in the inertia term, which were explained in the summary of Oliver.

Just for the record, we will include here the case with lower blob density as shown in figures 2 and 3 of this work, which in turn had been taken from Oliver and explained earlier in this work. All conditions are identical, density is ten times lower:

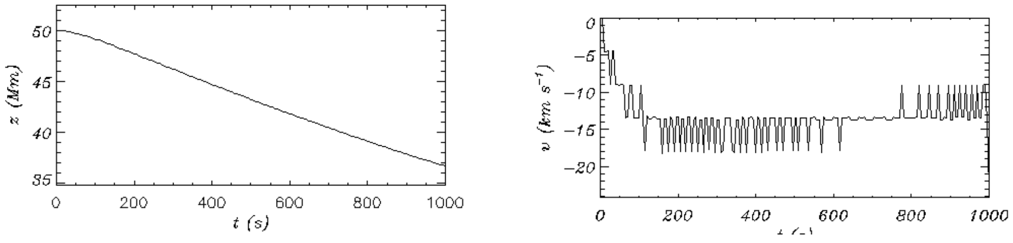

Fig 12.: (a) Position evolution of the blob with the described specifications and (b) velocity evolution for the very same blob. Density is ten times smaller than the previous figure. Obtained via IDL.

Notice the extreme fluctuations that take place in the velocity graph. Even though position evolves as expected if we compare our results to those derived from Oliver, the data shape that we obtain for the velocity cannot possibly be, under any reasonable assumptions, be ignored. It is in fact the wobble described a little further above, but taken to a magnitude big enough that the result is the complete distortion of the graph at hand.

Movies illustrating both study cases can be found *here* and *here*. The first movie represents a blob of density $\rho_{b0} = 10^{-10}$ kg/m-3 falling subject to gravity, as the second represents a blob of density $\rho_{b0} = 10^{-9}$ kg/m-3. In these movies we can appreciate better the dynamics involved in our study; although some parameters differ for the sake of representation, like cutting off the total fall time of the second blob to 600s, as it reaches the end of the assigned representation space in the IDL execution programmed, the description of the phases is the same we've mentioned earlier. This movies can help understand better the development of the pressure gradient and its effect on the blob's motion.

3.2 2D case

We now turn our attention to the bi-dimensional case of study. As this is not a case treated by Oliver, we will focus most of our attention here, and we will divide this section in further subsections that try to look for expected or unexpected behaviors in the simulations we will run. We will specify for each subsection the parameters considered, the aim of the experience and its results.

3.2.1 Comparison with 1D with no magnetic field

In this section, we aim to compare the effects of a vertical magnetic field in the z-axis on a blob that has actually two dimensions respect the previous experience. We expect that the effects of this should be negligible, as in the 1D case the vertical magnetic field was also taken to be vertical, and, as explained before, such assertion led to the magnetic terms disappearing from the magnetic fluid equations.

All initial parameters are exactly the same as in the previous run, with the exception of the extra conditions needed to further supply the 2D case: the x-axis, which has a numerical domain extension of $-4.5 \leq x \leq 4.5$ Mm, and the initial position in this axis is taken to be 0 Mm. The external magnetic field is taken to be purely vertical, with a magnitude of 1G for this first experience.

Computationally, the number of grid points rapidly rises, and because of the way in which the Pencil code works, the number of pencils involved in the calculations greatly rises in comparison with the previous case. As such, the number of points in the x grid is set to $nxgrid = 200$, so $dx = 45$ km, and $nzgrid = 4500$, so $dz = 44.4$ km.

Lastly, we have to account for the bi-dimensionality of our initial density distribution. This is taken to be the simplest possible extension to our original assumption for this distribution, solely including an x-axis term:

$$\rho_b\left(z,t=0\right)=\rho_{b0}exp\{-\left[\left(\frac{z-z_0}{\Delta}\right)^2+\left(\frac{x-x_0}{\Delta}\right)\right]\} \tag{3.1}$$

where the parameter Δ, which as we know accounts for blob elongation, is the same in both directions, resulting in a circular mass distribution. We now proceed to show the results extracted from this simulation, starting with the lesser density:

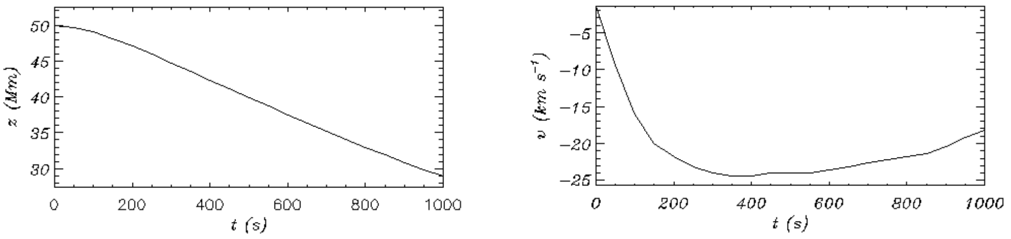

Fig 13.: (a) Position evolution of the blob with the described specifications and (b) velocity evolution for the very same blob. Obtained via IDL.

The results are kind of astonishing. By visual comparison, we can actually notice that the blob has fallen much more than in the case with no magnetic field action. Let us remind the reader that, as the magnetic field is purely vertical, it should not directly affect the downward falling speed of the blob in any relevant way. Even though the velocity graph of figure 12 (b) is difficult to interpret, it oscillates somewhere around -15 km/s. The presence of the magnetic field seems to also increase the downward velocity component, making it so that the fall velocity is bigger in this case.

We are not entirely sure how these effects have taken place, as the expected behavior was to be almost identical to the 1D case. Further investigation and commentary will be needed to identify the source of this increases, of around -20 km/s.

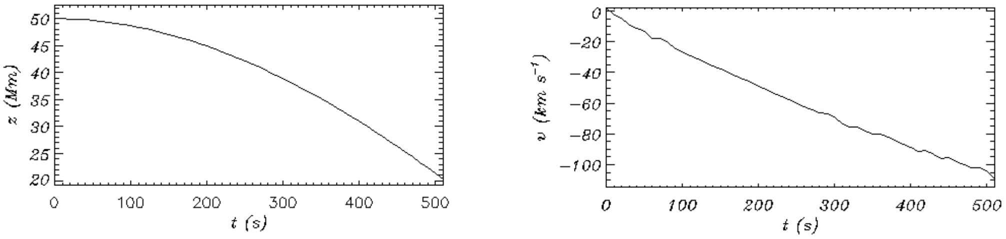

Fig 14.: (a) Position evolution of the blob with the described specifications and (b) velocity evolution for the very same blob. Density is ten times larger than the previous blob. Obtained via IDL.

We were not able to show further evolution of this case due to computational constraints, as the time step had become too short to be able to keep track of the evolution. Nevertheless, for the first 500 s, by comparing with the 1D case, we can easily observe that the blob has fallen an extra 10 Mm in the same span of time. Then, the unknown effect on the velocity seems to be related to the blob's density.

Chapter 4

Conclusions

We will now supplement all the study we have made thus far with a brief overview of the results obtained, the questions they answer and the ones they open. First of all, we have run the 1D case trying to check whether we would get the same results as O2014. This turned out good for the most part, with the great exception that we discovered a potent flaw of the Pencil Code, as we turned to the results obtained for the lowest density of the plasma knot considered, $\rho_{b0} = 10^{-10}$ kg/m^{-3}: the resulting graph for the velocity evolution was, admittedly, not very good. As the results for the position evolution for both densities in this case gave results that were very much in agreement by the work of O2014, and the velocity evolution also seemed to have some kind of spiked fluctuations, this erratic behavior has been attributed to a computational source. Indeed, finding the position of the maximum density of a fluid element is not a built-in function of the Pencil code, and as such, a simple program was to be made to be able to track this density clog.

But, as it seems, by this method we can only find the maximum density out of all the grid points where it is calculated. If the maximum density happens to be outside of this points, unphysical spikes appear. Even if some workarounds have been proposed while this work was taking place, like increasing the number of grid points in the vertical position or ignoring the slices in which the registered velocity had wild, sudden and isolated variations, currently the best results are the ones shown.

This effect seems all the more dramatic the lower the density, as when we made our analysis for the bigger density, at least the curve of the graph is well shown and in nice agreement with O2014. Thus, we can deem the results obtained by the combination of the Pencil code and IDL as physically significant, but we should remain on the watch for this kind of numerical fluctuations.

When we compared the magnetic field blob with the non-magnetic one, our original assumption was that there would not be any appreciable difference. This was mainly due to the vertical nature of the magnetic field in the MHD case and its' absence in the first case. Unexpectedly, we found out that this assumption is not quite true, there seems to be some effect of the magnetic field on the plasma knot, as it travels a greater distance downward in the same time, while also exhibiting larger overall velocity. The true nature of this phenomenon is still not understood, and we expect further research on this topic to be able to find the origin of the deviation seen.

All in all, Pencil Code is advertised, with reason, as a code optimized for super-computer runs with the option to use high-efficiency parallel computing. In a home PC, some of these simulations have been very space demanding and heavy on resources. We had to limit most of the results, images and videos shown throughout this work, as especially in the 2D MHD case, they were so demanding we were unable to find some efficient workaround to achieve graphical representation, if not by lowering the resolution of the executions, which of course would take its toll on the precision of our results. Just to get an idea, even for the simplest runs on the 2D MHD case (and by simple, we just mean relatively low parameter values) occupied a hefty 150 Gb. This skyrocket in computer resources seems obvious if we remember the way in which Pencil Code tries to solve the given equations: by solving one-dimensional large arrays of data. This in turn implies that the extension to the 2D case immediately multiplies by an enormous amount the number of pencils solved in each run, so much that the parameters we chose to show in this work were actually chosen in terms of their feasibility to be reproduced, instead of their pure, physical interest. For example, an external magnetic field of 10 G would have been of some interest, given that, as told before, this is approximately the order of magnitude found on coronal loops. Nevertheless, such value gave rise to the presence of physical fast moving Alfvén waves, that would lead to the unphysical auto-interaction described also here on this work. As to avoid this effect, we found two possible workarounds: either the spatial numerical domain is big enough that the waves have no time to re-interact with the blob, in the same way we dealt with the sound waves emitted by the blob, or else, weakening the magnetic field.

Our first strategy did not yield the desired results, as we had not enough computer resources to be able to simulate a domain big enough with a small enough step size as to avoid this re-interaction. Then, the second option had to be accepted, which consisted in lowering the magnetic field, thus slowing down the Alfvén waves emitted, thus giving enough time for the dynamics of the blob to take place.

The precision has been found to be, to say it bluntly, lacking. Even as we were able to run heavy simulations given enough time, visualization, as commented above, prove to be impossible. Given the way that IDL manages data structures, it brings data to the RAM memory, where the necessary calculations are made. Nevertheless, when the simulation is too big, the length of the structures proves to be way too big to be dealt with in any way. Just to give an example, the computer that carried out these simulations had around 24 Gb or RAM memory plus another 7 Gb of swap memory, and we could not run the visualizations before causing a memory related exception. IDL needs some expertise in its usage, but we were unable to find a way to visualize the most accurate (or heavy, for that matter) simulations. Thus, we had to step down the rate at which data was stored: from 1 slice per second of simulation for 1000 seconds of simulation to 1 slice per 10 seconds per 1000 seconds of simulation.

This had, of course, drained our precision very fast, and quantitative results could prove to be deceitful. Even so, some theorized effects could be observed and brought to light, such as the unaccounted extra speed of the blob subject to a vertical magnetic field. We further encourage then studies in the dynamics of coronal rain, as it seems there is much more to it than it seemed. We hope this gives more than enough justification to use a supercomputer on future investigations, as to keep unraveling the secrets of coronal rain.

References

Antiochos, S. K., Klimchuk, J. A. 1991, *ApJ*, **378**, 372

Antiochos, S. K., MacNeice, P. J., Spicer, D. S., & Klimchuk, J. A. 1999, *ApJ*, **512**, 985

Antolin, P., & Rouppe van der Voort, L. 2012, *ApJ*, **745**, 152

Beckers, J. M. 1962, *AuJPh*, **15**, 327

Goedbloed, J. P. H.; Poedts, S. (2004) *Principles of Magnetohydrodynamics* Cambridge University Press

Kawaguchi, I. 1970, *PASJ*, **22**, 405

Leroy, J. 1972, *Sol.Phys.*, **25**, 413

Müller, D. A. N., Peter, H., & Hansteen, V. H. 2004, *A&A*, **424**, 289

Oliver, R., Soler, R., Terradas, J., Zaqarashvili, T. V., & Khodachenko, M. L. 2014, *ApJ*, **784**, 21

Parker, E.N.: 1955a, *Astrophys.J.*, **121**, 491

Priest, E. (2004) *Magnetohydrodynamics of the Sun.* University of St Andrews, Scotland: Cambridge University Press