Stellar convection

Candidate number : 23
(Dated: May 31, 2023)

I. INTRODUCTION

In this report we aim to model convection in 2 dimensions in the photosphere of the sun. Such a model relies on three important equations within hydrodynamics: the continuity equation (1), the momentum equation (3) and the energy equation (5). In order to solve these equations, we use an explicit numerical scheme; specifically, we use the Forward Time Centered Space (FTCS) numerical scheme. However, FTCS is unconditionally unstable, meaning we must consider other numerical schemes in order to avoid numerical errors; such as the Upwind differencing scheme. We will dive further into what this entails in the method section.

For all of these equations to work, we need to make sure the gas we are dealing with behaves like a fluid. This can be found by finding the fluid's Knudsen number, defined as $Kn = \lambda/L$ where L is the scale of the system and λ is the mean free path of a neutral particle. In the photosphere $\lambda = 10^{-4}m$, and further below the surface it only increases. The scale of our system will be in the mega-meters and so the Knudsen number is much smaller than 1, which is the requirement for particles to be seen as a fluid.

In addition to our differential equations solver, we will also discuss the initialization of our system. The gas needs to be initialized in hydrostatic equilibrium and have $\nabla = \frac{d\ln T}{d\ln P} \approx 2/5$. Since the model is initialized in hydrostatic equilibrium, it is expected to remain so for a longer period of time once simulated also. Once such a model is stable, we will introduce different 2D Gaussian perturbations in the initial temperature and discuss how our model evolve the pertubation through time.

II. METHOD

To begin with, we need to define the governing equations of our model. The first equation we will look at is the continuity equation with no sources or sinks (1). Since we do not account for sources or sinks, we will later need to make sure no particles are able to escape at the top or enter at the bottom of our box. This box will have a physical size of 12Mm along the x-axis and 6Mm along the y-axis, where the y-axis is in the radial direction. The box represents an area below the photosphere, with the top of the box touching the photosphere. The box does not need to take the curve of the Sun' surface into account as the radius of the Sun is sufficiently large enough.

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \vec{u}) = 0 \tag{1}$$

Where $[\rho] = kg/m^3$ is the density of the fluid, $[t] = s$ is time and $\vec{u} = (u, w)$ with units $[\vec{u}] = m/s$ is the velocity vector. We wish to use this equation to progress ρ in time, and so we need to solve for its partial time derivative in equation (2).

$$\begin{aligned} \frac{\partial \rho}{\partial t} &= -\nabla \cdot (\rho \vec{u}), \quad \nabla \cdot (\rho \vec{u}) = \frac{\partial \rho u}{\partial x} + \frac{\partial \rho w}{\partial y} \\ \frac{\partial \rho}{\partial t} &= -\frac{\partial \rho u}{\partial x} - \frac{\partial \rho w}{\partial y} \end{aligned} \tag{2}$$

Next, we have the momentum equation excluding the viscous stress tensor, but including gravity (3). The viscous stress term has been excluded in order to simplify our model, as the term is often a source of much added complexity to a fluid model. As for the gravity term, the gravity can be assumed constant due as our system is small compared to the size of the sun.

$$\frac{\partial \rho \vec{u}}{\partial t} + \nabla \cdot (\rho \vec{u} \otimes \vec{u}) = -\nabla P + \rho \vec{g} \tag{3}$$

Where $[\nabla P] = Pa/m$ and $[\vec{g}] = m/s^2$ with $\vec{g} = (0, g_y)$. Since the y-direction of our system is the same as the radial direction, the gravity will only pull in the y-direction. Once again we want to solve for the time derivatives and after some algebra, as shown in Appendix A, we get equation (4).

$$\begin{aligned} \frac{\partial \rho u}{\partial t} &= -\frac{\partial \rho u^2}{\partial x} - \frac{\partial \rho u w}{\partial y} - \frac{\partial P}{\partial x} \\ \frac{\partial \rho w}{\partial t} &= -\frac{\partial \rho u w}{\partial x} - \frac{\partial \rho w^2}{\partial y} - \frac{\partial P}{\partial y} + \rho g_y \end{aligned} \tag{4}$$

Where $g_y = g = \frac{GM_\odot}{R_\odot^2}$. Finally, we have the energy equation (5).

$$\frac{\partial e}{\partial t} + \nabla \cdot (e \vec{u}) = -P \nabla \cdot \vec{u} \tag{5}$$

Where $[e] = J/m^3$ is the internal energy. After solving for the time derivative of e, as shown in appendix B, we arrive at the following equation 6

$$\frac{\partial e}{\partial t} = -\frac{\partial e u}{\partial x} - \frac{\partial e w}{\partial y} - P\left(\frac{\partial y}{\partial x} + \frac{\partial w}{\partial y}\right) \tag{6}$$

In order to use these equations to evolve our system, we also need initial conditions. To begin with, we want our system to be in hydrostatic equilibrium and have a temperature gradient with regards to pressure as shown in equation (7) where $\nabla^* \approx 2/5$. Using these two conditions and the ideal gas law, we arrive at an expression for

the initial T and P in equations (8) and (9) respectively. The calculations of which are derived in appendix C.

$$\nabla = \frac{\partial \ln T}{\partial \ln P} \tag{7}$$

$$T = T_0 - \nabla^* \frac{m_u \mu}{k_B} g(R - R_0) \tag{8}$$

Where T_0 is the temperature at the top of the computational box which corresponds to $T_0 = 5778K$, the temperature of the photosphere. And R_0 is the radius of the sun $R_0 = R_\odot$. The way we define our system, $R = y$. Using equation (7) and (8) we can get an expression for pressure also. The more detailed derivation is also contained in appendix C.

$$P = P_0 \left(\frac{T}{T_0}\right)^{1/\nabla^*} \tag{9}$$

When we have both T and P we can find the mass density $\rho = \mu m_u P / k_B T$ as derived from the ideal gas law. Lastly, we need also the internal energy e, which is defined as in equation (10).

$$\begin{aligned} e &= \frac{1}{(\gamma - 1)} \frac{\rho}{\mu m_u} k_B T \\ e &= \frac{1}{(\gamma - 1)} P \end{aligned} \tag{10}$$

Where $\gamma = 5/3$ for ideal gas.

To summarize, we use equation 7 and hydrostatic equilibrium to find T and P, with these we can find ρ through the equation of state for an ideal gas and then internal energy e. The initial conditions values are summarized in table I Next, knowing how we initialize our system and our governing equation, we want to begin discretizing these equations. However, before that we need to discuss FTCS scheme and upwind differencing. The FTCS scheme uses forward Euler (11) to progress through time and central differencing (12) in space. As all numerical methods, these rely on discretized spatial and time parameters. We choose a resolution of $N_x = 300$ and $N_y = 100$ representing $L_x = 12Mm$ and $L_y = 4Mm$ respectively. We will talk more about time resolution later, for now you need to know it will start at 0 and end at a chosen simulation end time, ranging from 1 minute and up to 10 minutes. A given point in the computational point at a given time can be indexed as $\phi^n_{i,j}$, where n describes its location in time, $i \in [0, N_x - 1]$ is the x-coordinate and $j \in [0, N_y - 1]$ is the y-coordinate. These will be evenly spaced out with a spacing of $\Delta y = L_y / N_y$ in the y-direction and $\Delta x = L_x / N_x$ in the x-direction. $\phi^n_{0,0}$ is located in the bottom left corner of the computation box and $\phi^n_{N_y - 1, N_x - 1}$ is located in the upper right corner. We define $y \in [R_\odot - L_y, R_\odot]$ and $x \in [0, L_x]$.

Table I. The table contains values used to initialize the system.

Parameter	Value	Units
T_0	5778	[K]
P_0	$1.8 \cdot 10^4$	[Pa]
μ	0.61	[*]
γ	5/3	[*]
∇^*	$2/5 + 10{-}5$	[*]
$M_\odot$	$1.989 \cdot 10^{30}$	[kg]
$R_\odot$	$6.96 \cdot 10^8$	[m]

$$\phi^{n+1}_{i,j} = \phi^n_{i,j} + \left[\frac{\partial \phi}{\partial t}\right]^n_{i,j} dt \tag{11}$$

Where ϕ represents ρ, u, w and e.

$$\begin{aligned} \left[\frac{\partial \phi}{\partial x}\right]^n_{i,j} &\approx \frac{\phi^n_{i+1,j} - \phi^n_{i-1,j}}{2\Delta x} \\ \left[\frac{\partial \phi}{\partial y}\right]^n_{i,j} &\approx \frac{\phi^n_{i,j+1} - \phi^n_{i,j+1}}{2\Delta y} \end{aligned} \tag{12}$$

However, as mentioned, central differencing is unstable and another scheme is needed depending on the equation. Upwind differencing is a more stable, but less accurate, scheme we can use. While there is a mathematical explanation, we will only go into an intuition level approach later.

We can now begin to discretize equation (2), (4) and (6). We must first do a few more derivations. The flow of (u, w) is not assumed constant and so the product rule is needed. In equation (4), ρu is considered one quantity and not the product of two different ones. With this, equations (2), (4) and (6) becomes equations (13), (14) and (15) when discretized..

$$\left[\frac{\partial \rho}{\partial t}\right]^n_{i,j} = -[\rho]^n_{i,j} \left(\left[\frac{\partial u}{\partial x}\right]^n_{i,j} + \left[\frac{\partial w}{\partial y}\right]^n_{i,j}\right) - [u]^n_{i,j} \left[\frac{\partial \rho}{\partial x}\right]^n_{i,j} - [w]^n_{i,j} \left[\frac{\partial \rho}{\partial y}\right]^n_{i,j} \tag{13}$$

$$\begin{aligned}\left[\frac{\partial \rho u}{\partial t}\right]_{i,j}^{n} &= -\left[\rho u\right]_{i,j}^{n}\left(\left[\frac{\partial u}{\partial x}\right]_{i,j}^{n}+\left[\frac{\partial w}{\partial y}\right]_{i,j}^{n}\right)-\left[u\right]_{i,j}^{n}\left[\frac{\partial \rho u}{\partial x}\right]_{i,j}^{n}-\left[w\right]_{i,j}^{n}\left[\frac{\partial \rho u}{\partial y}\right]_{i,j}^{n}-\left[\frac{\partial P}{\partial x}\right]_{i,j}^{n}\\ \left[\frac{\partial \rho w}{\partial t}\right]_{i,j}^{n} &= -\left[\rho w\right]_{i,j}^{n}\left[\left(\frac{\partial u}{\partial x}\right]_{i,j}^{n}+\left[\frac{\partial w}{\partial y}\right]_{i,j}^{n}\right)-\left[u\right]_{i,j}^{n}\left[\frac{\partial \rho w}{\partial x}\right]_{i,j}^{n}-\left[w\right]_{i,j}^{n}\left[\frac{\partial \rho w}{\partial y}\right]_{i,j}^{n}-\left[\frac{\partial P}{\partial y}\right]_{i,j}^{n}+\left[\rho\right]_{i,j}^{n} g_y\end{aligned} \tag{14}$$

$$\left[\frac{\partial e}{\partial t}\right]_{i,j}^{n} = -(\left[e\right]_{i,j}^{n}+\left[P\right]_{i,j}^{n})\left(\left[\frac{\partial u}{\partial x}\right]_{i,j}^{n}+\left[\frac{\partial w}{\partial y}\right]_{i,j}^{n}\right)-\left[u\right]_{i,j}^{n}\left[\frac{\partial e}{\partial x}\right]_{i,j}^{n}-\left[w\right]_{i,j}^{n}\left[\frac{\partial e}{\partial y}\right]_{i,j}^{n} \tag{15}$$

The upwind scheme is dependent the flow velocity in the same direction, meaning upwind in the x-direction depends on u and in the y-direction on w. So when the gas is moving in positive x-direction, it finds the derivative of the variable ϕ by comparing where the gas $pos = i, j$ is now and where it was $pos = (i-1, j)$. And when moving in the opposite direction it takes $pos = (i+1, j)$ instead for the second position. This is shown in equations (16) and (17).

$$\frac{\partial \phi}{\partial x} = \begin{cases} \frac{\phi_{i,j}^{n}-\phi_{i-1,j}^{n}}{\Delta x} & \text{if } u_{i,j}^{n} \geq 0 \\ \frac{\phi_{i+1,j}^{n}-\phi_{i,j}^{n}}{\Delta x} & \text{if } u_{i,j}^{n} < 0 \end{cases} \tag{16}$$

$$\frac{\partial \phi}{\partial y} = \begin{cases} \frac{\phi_{i,j}^{n}-\phi_{i,j-1}^{n}}{\Delta y} & \text{if } w_{i,j}^{n} \geq 0 \\ \frac{\phi_{i,j+1}^{n}-\phi_{i,j}^{n}}{\Delta y} & \text{if } w_{i,j}^{n} < 0 \end{cases} \tag{17}$$

The use of upwind over central is determined by whether the spatial derivative is multiplied by the flow in the same direction as the spatial derivative. An example of this is equation (14), where the time derivative of ρu and ρw are calculated. In the case for ρu, only the spatial derivative of P and w are calculated using central, the rest use upwind. This is due to u being multiplied by the spatial derivative of u in x-direction and ρu in x-direction and w is multiplied by the spatial derivative of ρu in the y-direction. For the time derivative of ρw, the spatial derivative of P is again calculated using central, but this time the spatial derivative of u is calculated using central.

Now that we have the time derivatives, we can find the next timestep's values. To do so we use forward Euler, which is derived from using forward differencing to define a derivative. A quick derivation of this is shown in equation (18).

$$\begin{aligned}\left[\frac{\partial \phi}{\partial t}\right]_{i,j}^{n+1} &= \frac{\left[\phi\right]_{i,j}^{n+1}-\left[\phi\right]_{i,j}^{n}}{\Delta t} \\ \text{Solve for } \left[\phi\right]_{i,j}^{n+1} & \\ \left[\phi\right]_{i,j}^{n+1} &= \left[\phi\right]_{i,j}^{n}+\left[\frac{\partial \phi}{\partial t}\right]_{i,j}^{n+1}\Delta t\end{aligned} \tag{18}$$

First we find $\left[\rho\right]_{i,j}^{n+1}$, which using forward Euler can be written as equation (19).

$$\left[\rho\right]_{i,j}^{n+1} = \left[\rho\right]_{i,j}^{n}+\left[\frac{\partial \rho}{\partial t}\right]_{i,j}^{n}\Delta t \tag{19}$$

Next we use forward Euler to progress $\left[\rho w\right]_{i,j}^{n+1}$, and since we have already found $\left[\rho\right]_{i,j}^{n+1}$ we can divide by it on both side to get an expression for the updated value of $\left[w\right]_{i,j}^{n+1}$, as shown in equation (20).

$$\begin{aligned}\left[\rho w\right]_{i,j}^{n+1} &= \left[\rho w\right]_{i,j}^{n}+\left[\frac{\partial \rho w}{\partial t}\right]_{i,j}^{n}\Delta t \\ \left[w\right]_{i,j}^{n+1} &= \frac{\left[\rho w\right]_{i,j}^{n}+\left[\frac{\partial \rho w}{\partial t}\right]_{i,j}^{n}\Delta t}{\left[\rho\right]_{i,j}^{n+1}}\end{aligned} \tag{20}$$

The same logic can be applied to $\left[u\right]_{i,j}^{n+1}$ and we get equation (21)

$$\left[u\right]_{i,j}^{n+1} = \frac{\left[\rho u\right]_{i,j}^{n}+\left[\frac{\partial \rho u}{\partial t}\right]_{i,j}^{n}\Delta t}{\left[\rho\right]_{i,j}^{n+1}} \tag{21}$$

Lastly, the new internal energy e can be found by the same logic as for ρ, giving us equation (22).

$$\left[e\right]_{i,j}^{n+1} = \left[e\right]_{i,j}^{n}+\left[\frac{\partial e}{\partial t}\right]_{i,j}^{n}\Delta t \tag{22}$$

All these equations depend on Δt, which we have yet to define. We will set a few criteria. First, we do not want to progress less than $\Delta t = 0.01$ as it would only serve to slow down the program. We also want to make sure the timestep is not big enough to displace gas more than one pixel over, meaning $rel(\vec{x}) = |\vec{u}/\Delta\vec{x}$ where $\Delta\vec{x} = (\Delta x, \Delta y)$. Additionally, we do not want the changes to physical quantities to be more than a given percentage of the quantities. We set this percentage $p = 0.01$ and define $\Delta t = p/\delta$, where $\delta = max(max(rel(\phi)))$. We find $rel(\phi)$ by calculating the relative change $\Delta\phi/\phi$ per time step Δt, which comes out to equation (23).

$$rel(\phi) = \left|\frac{\partial \phi}{\partial t}\cdot\frac{1}{\phi}\right| \tag{23}$$

As the system has progressed to the next timestep, a new problem arises however. As mentioned previously, our continuity system has no sources or sinks. This means we cannot allow gas to escape or enter the system, as it would throw off the system. We have four places

this can happen, the surface, the two sides and the bottom. We fix the problem with the two sides by connecting them, effectively turning our computational box into the surface of a cylinder. However, this is only a functional change, in reality we are assuming that our computational box is surrounded by similiar boxes with the exact same initial conditions. The top and bottom are however more complex as there are a few problems. First, as evident in equation (12) and (17) in the y-direction, both schemes rely on the row above and below the boundary at the top and bottom of the computational box respectively. Second, no gas can escape out of the surface or enter from the bottom. For the first problem, we introduce the 3-point forward (24) and backward (25) differencing approximations in the y-direction.

$$\left[\frac{\partial \phi}{\partial y}\right]^n_{i,j} = \frac{-\phi^n_{i,j+2} + 4\phi^n_{i,j+1} - 3\phi^n_{i,j}}{2\Delta y} \tag{24}$$

$$\left[\frac{\partial \phi}{\partial y}\right]^n_{i,j} = \frac{3\phi^n_{i,j} - 4\phi^n_{i,j-1} + \phi^n_{i,j-2}}{2\Delta y} \tag{25}$$

The two problems are interconnected, so we can use this 3-point differencing together with how we want the boundaries to behave. In order to smooth the horizontal velocities along the surface and bottom we set its vertical gradient equal to zero at the boundary. This gives us equation (26), which can be solved for the $u_{i,0}$, a similar method can be applied at the lower boundary to give us equation (27) to give us u_{i,N_y-1}.

$$\begin{aligned} 0 = \left[\frac{\partial u}{\partial y}\right]^n_{i,0} &= \frac{-u^n_{i,2} + 4u^n_{i,1} - 3u^n_{i,0}}{2\Delta y} \\ u^n_{i,0} &= \frac{-u^n_{i,2} + 4u^n_{i,1}}{3} \end{aligned} \tag{26}$$

$$u^n_{i,N_y-1} = \frac{-u^n_{i,N_y-3} + 4u^n_{i,N_y-2}}{3} \tag{27}$$

Since no gas is escaping or entering at these points, the system should also be in static equilibrium at these points. With this, the pressure gradient is given. By extension, through equation (10), the internal energy gradient is also set. Combining this with equations (24) and (25), we have a way to determine the internal energy at the edges. Equations (28) and (29) shows the new definition of internal energy at the boundaries.

$$\begin{aligned} \left[\frac{\partial e}{\partial y}\right]^n_{i,0} &= \frac{-e^n_{i,2} + 4e^n_{i,1} - 3e^n_{i,0}}{2\Delta y} \\ e^n_{i,0} &= \frac{-2\Delta y \left[\frac{\partial e}{\partial y}\right]^n_{i,0} - e^n_{i,2} + 4e^n_{i,1}}{3} \end{aligned} \tag{28}$$

$$\begin{aligned} \left[\frac{\partial e}{\partial y}\right]^n_{i,N_y-1} &= \frac{e^n_{i,N_y-3} - 4e^n_{i,N_y-2} + 3e^n_{i,N_y-1}}{2\Delta y} \\ e^n_{i,N_y-1} &= \frac{2\Delta y \left[\frac{\partial e}{\partial y}\right]^n_{i,N_y-1} - e^n_{i,N_y-3} + 4e^n_{i,N_y-2}}{3} \end{aligned} \tag{29}$$

This in turn also effects the mass density ρ, as we require ideal gas. By solving equation (10) for ρ we get equation (29).

$$\rho = (\gamma - 1)\frac{\mu m_u}{k_B T} e \tag{30}$$

Lastly, in order for no gas to escape through the bottom or top, their vertical velocity needs to be zero.

With everything in order, we can make a summary of the algorithm.

(a) We begin by finding the initial conditions, defining T, P and e from equation (8), (9) and (10) respectively. With these variables we can also find the mass density ρ from the equation of state for ideal gas (29).

(b) Next, we start progressing the system forward in time using forward Euler. To do so we must find the time derivatives of ρ (13), ρu(14), ρw (14) and e (15).

(c) These all rely on spatial derivatives. How these spatial derivatives depend on what time derivative we are calculating. The central scheme is used by default, while upwind is used when a spatial derivative is multiplied by the flow velocity in the same direction as the derivative.

(d) With time derivatives calculated, we find our timestep, which cannot be too big. We first find $\rho^{n+1}_{i,j}$ (19), as both u and w rely on it. Then we can find w, u and e with equations (20), (21) and (22) respectively.

(e) After finding the new updated variables we must enforce the boundary conditions. We do this by converting our computational box to a cylinder, set vertical velocities to $w = 0$ at the vertical boundary, use equation (28) and (29) in combination with equation (30) to find e and ρ at the border. Lastly, we also set the vertical component of the gradient of the horizontal velocity to zero and use the 3-point scheme to find the new value at the boundary using equations (26) and (27).

(f) With boundary conditions enforced, we can move forward in time after making sure all relevant values are stored for later. This continues until the $\sum^N \Delta t = x$, where x is the number of seconds we wish to simulate for and N is the amount of time steps taken to get to x seconds.

Once we have a model, we can verify it to some extent by running it for a minute and there should be no changes. This means the model is in hydrostatic equilibrium, which is should be according to how we initialized it. We can now begin to add pertubation into the system in order to get convective motions. Convective motions appear when hot gas rises to the top. Thus, we need to

add a pertubation to the temperature. However, the way we define both P and ρ, a pertubation added to T will affect both. However, adding the pertubation to T before we define P gives very extreme values for P. Therefor we add it after P has been defined, effectively adding a pertubation to T and ρ.

Next, we need to talk about the type of pertubation and its scale. First the type, we use a 2D Gaussian to get a smooth 2D curve that transitions from very hot and into the surrounding temperatures. A 2D gaussian is the product of two Gaussians along different axes. Meaning, where both Gaussians have high values, the 2D gaussian will also and where one has a large value while the other is close to zero, the resulting is also close to zero. A Gaussian, as shown in equation (31), is dependent on spatial coordinate x, mean position μ_x and standard diviation σ_x.

$$g(x) = \exp\left\{-\frac{(x-\mu_x)^2}{\sigma_x^2}\right\} \tag{31}$$

We multiply equation (31) by an equivalent, but in the y-direction and we get equation (32).

$$\begin{aligned} G(x,y) = g(x)h(y) &= \exp\left\{-\frac{(x-\mu_x)^2}{\sigma_x^2}\right\} \cdot \exp\left\{\frac{-(y-\mu_y)^2}{\sigma_y^2}\right\} \\ G(x,y) &= \exp\left\{-\left(\frac{(x-\mu_x)^2}{\sigma_x^2} + \frac{(y-\mu_y)^2}{\sigma_y^2}\right)\right\} \end{aligned} \tag{32}$$

Using equation (32), we can decide the location and shape of our pertubation. μ_x and μ_y decides the spatial coordinates for the centre of the pertubation. While σ_x decides the shape of the distribution in the x-direction and σ_y in the y-direction. It is also possible to add severa pertubations by summing the different gaussians up, before applying the pertubation to T and ρ.

Next we need to talk about the scale. We use the surface temperature $T_0 = 5778K$ to scale and a factor $a \in [1, 20]$. The factor a will be used to test different degrees of pertubation. We then define $T_{pert} = aT_0G(x,y)/\max G(x,y)$ and add it into the initial temperature T_{init} after P_init has been found. By dividing the gaussian by its maximum, the peak of the 2D pertubation gets a temperature $T = aT_0$.

Once the system has been initialized with the pertubations, we let the model simulate until a desired time or a crash. The desired time $t_f \in [60, 600]$, where $[t_f] = s$, will vary depending on how a and the 2D gaussian $G(x,y)$ is designed.

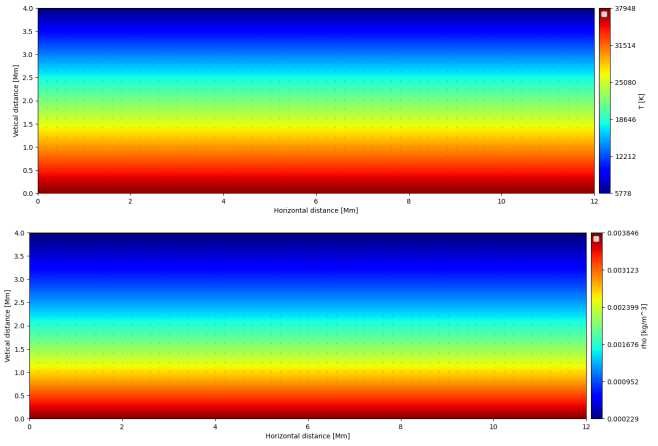

Figure 1. Shows the initial temperature $[T] = K$ and density $[\rho] = kg/m^3$ in the top and bottom plot respectively. Each plot has their own color bar scale, where its units and values are specified. The plot covers a physical space of $4Mm \times 12Mm$. Gravity is parallel to the y-axis.

III. RESULTS

First, in order to compare with, we have a figure which includes the initial state of both temperature T and mass density ρ. The vertical velocity w plot has been excluded as the system is in hydrostatic equilibrium.

Worth noting, the scales of the different plots within a figures vary from plot to plot, meaning you have to be careful when just looking at colors.

We begin by adding a single pertubation to the bottom of the computational box in the middle of the x-axis as seen in the top part of figure 2. After testing a few different values of a, we landed on $a = 14$. The pertubation has been dragged out such that it appears the pertubation has come from a hot gas bouble traveling upwards and losing heat to the surroundings as it reached the top.

After 4 min, it appears the heat has mostly spread out, while there is still remaining movement.

Next we add more perturbations, for now they are the same shape. We space them evenly, so that the distance between each is $4Mm$. We can see how it looks in figure 3. After adding additional perturbations we see that the system is less in rest than before. Even after 10 minutes there is significant structures.

We wish to study this further and will require more information about the other parameters. For this we have figures 4, which shows the mass density ρ, and figure 5.

In figure 4 we see that a pertubation temperature re-

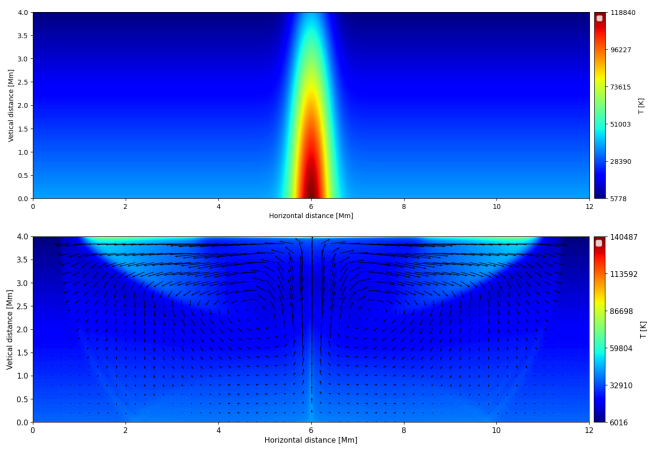

Figure 2. Shows the temperature $[T] = K$ plot of a gaussian pertubation with $\mu_x = 6Mm$, $\mu_y = 0$, $\sigma_x = 3 \cdot 10^5 Mm$ and $\sigma_y = 3 \cdot 10^6$ after 0 seconds, at the top, and 4 minutes, at the bottom. The arrows show the direction of gas flow and has a scale $0.189Mm/s$ per Mm.

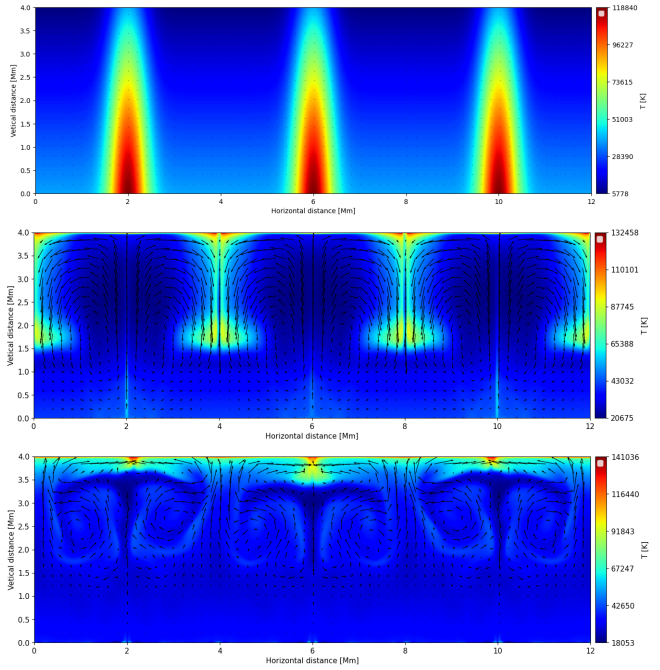

Figure 3. Shows the temperature $[T] = K$ plot of three gaussian pertubation evenly spaced with $\mu \in [3Mm, 6Mm, 9Mm]$, $\mu_y = 0$, $\sigma_x = 3 \cdot 10^5 Mm$ and $\sigma_y = 3 \cdot 10^6$ after 0 seconds, at the top, 4 minutes, in the middle, and 9 minutes and 55 seconds at the bottom. The arrows show the direction of gas flow and has a scale $0.191Mm/s$ per Mm in the middle plot and $0.197Mm/s$ per Mm in the bottom plot.

duces the density, essentially creating an inverted picture of the initial temperature. After four minutes the system has begun stabilizing, but is separated into two regions and after ten minutes two regions have started melting into each other.

Worth noting is that w has been simulated one single timestep, so that it isnt 0 as initialized, the other plots in the figure 5 have otherwise the exact same time. In the figure we see that the initial reaction of the system is to push the hot gas upwards. Later, after four minutes in the second plot in the figure, there have appeared distinct regions of gas being pushed up and down. Towards the end of the 10 minutes, the regions have become less distinct and more complex as visible in the last plot.

IV. DISCUSSION

To begin with, we establish a baseline from our initial conditions, shown in figure 1. While both images look similar, they are distinguished by their color bar scale. Both have a difference of one magnitude from their highest and lowest value. Unintuitively, we do not see the same resemblance when we compare figure 3 with figure 4. Instead, we see an inverted version, meaning where the temperature has risen in the temperature plot, it has fallen for density plot. This can be attributed to the order we initialize our pertubation. If we had added it before finding P_init the density would have looked similar to the temperature. What this means is that when looking at later snapshots of density ρ when adding perturbations, we would expect the density in the system settles on overall lower values. Which is apparent in the last plot of figure 4.

When we add more perturbations to our model, we effectively add more energy into the model. With more energy, which has nowhere to go, the system takes longer to settle. It also reveals some flaws in our model.

As evident in figure 3, where temperature is for multiple perturbations is shown, the hot gas has nowhere to go. Normally, such hot gas would be thrown out in the form of solar flares. Instead, since we have imposed velocity restrictions on our boundaries, the gas cannot continue moving upwards or downwards. Already from the initial frame of vertical velocity, we can see that gas quickly begins to accelerate towards the top. Assuming this acceleration would continue for at least a second more, velocities would reach $1.5km/s$ and later even higher velocities. This gas is however not allowed to continue to rise and so our model fails to simulate more realistic scenarios.

At the end of the 10 min simulation convection cells have formed, however they are below the surface due to being pushed down by the very light hot gas. The convection within the cells have slowed down somewhat in the six minutes that have passed since the second plot as shown in figure 5. Additionally, the computational box has seemingly divided itself in two, with colder heavy gas on the bottom and lighter gas on the top. Seeing how the bottom part would grow over the next few minutes or hours would be interesting. Presumably, while unable to check due to numerical constraints, the system would most likely continue to stabilize until some new hydrostatic equilibrium has formed.

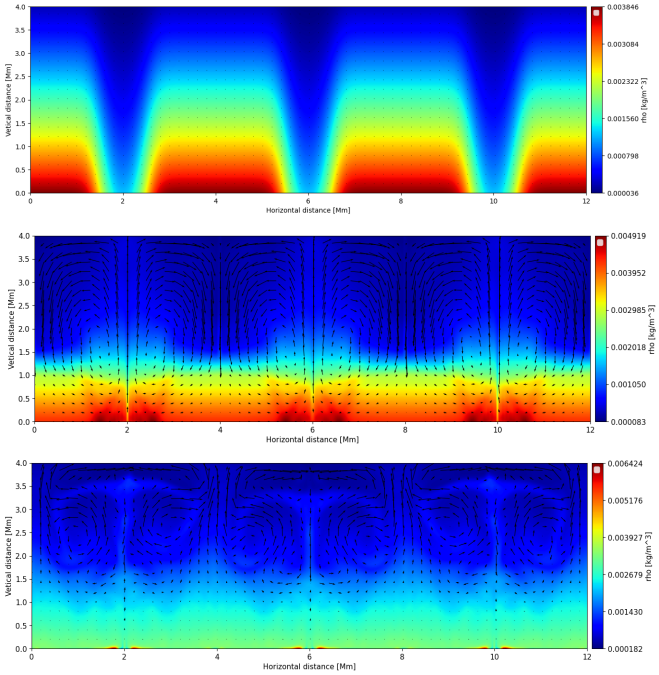

Figure 4. Shows the density $[\rho] = kg/m^3$ plot of three gaussian pertubation evenly spaced with $\mu \in [3Mm, 6Mm, 9Mm]$, $\mu_y = 0$, $\sigma_x = 3 \cdot 10^5 Mm$ and $\sigma_y = 3 \cdot 10^6$ after 0 seconds, at the top, 4 minutes, in the middle, and 9 minutes and 55 seconds at the bottom. The arrows show the direction of gas flow and has a scale $0.191 Mm/s$ per Mm in the middle plot and $0.197 Mm/s$ per Mm in the bottom plot.

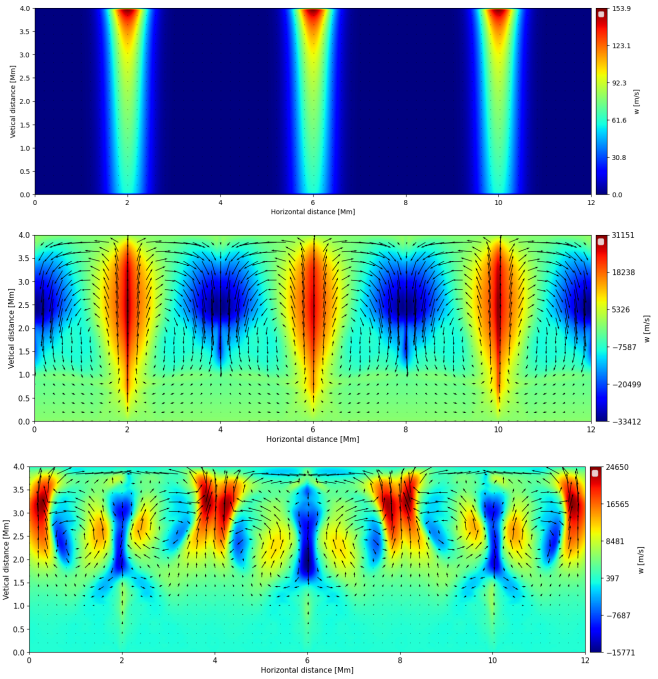

Figure 5. Shows the vertical velocity $[w] = m/s$ plot of three gaussian pertubation evenly spaced with $\mu \in [3Mm, 6Mm, 9Mm]$, $\mu_y = 0$, $\sigma_x = 3 \cdot 10^5 Mm$ and $\sigma_y = 3 \cdot 10^6$ after 0 seconds, at the top, 4 minutes, in the middle, and 9 minutes and 55 seconds at the bottom. The arrows show the direction of gas flow and has a scale $0.191 Mm/s$ per Mm in the middle plot and $0.197 Mm/s$ per Mm in the bottom plot.

V. CONCLUSION

While solving such a big project, the main challenge was the initial architecture of the model. While the skeleton code provided was unintuitive at first, it did prove usefull in the end. It did however impose an extra challenge of using object oriented programing in a way I wasnt used to. After spending some time getting my equations in order, it was mostly smooth sailing from there. Smooth as in the bumps were small and easy to overcome once discovered, but there were a lot of bumps. A lot of time was spent reviewing the differntial schemes, which never seemed to be perfect. The main difficulty came from piecing together the algorithm and hints while the math and programming was relatively easy. Especially unintuitive was the use of the boundary condition hint together with the vertical boundary for horizontal velocity. In order to effectively compare my initial conditions and evolution with what was happening in the animation, I developed a plotting function which later became very usefull in generating the plots used in this paper.

Through this project I have learnt how to better structure code for medium size projects. I do not think the added difficulty of having to interpret what the project description is talking about gave me a better understanding. Instead I wish there was more physical problems we had to solve before being able to produce code. And if numerical schemes are to be used they should also be talked about in the lecture notes.

In the end our model was able to successfully simulate convection when pertubations were added. While the convection was not entirely realistic, it did happen. The main issue with the model seems to be our upper vertical boundary, which stops solar flares and thus heat from escaping. A model which includes the lower photosphere would be interesting to look at.

A point of further exploration is how smaller pertubations affect the model, and pushing the limits on what is needed for convection to occur.

Appendix A: Momentum equation

Starting with equation A1, we wish to find A4.

$$\frac{\partial \rho \vec{u}}{\partial t} + \nabla \cdot (\rho \vec{u} \otimes \vec{u}) = -\nabla P + \rho \vec{g} \tag{A1}$$

We start by looking at $\nabla \cdot (\rho \vec{u} \otimes \vec{u})$. $\otimes$ is an outer product and results in the matrix in equation (A2):

$$\nabla \cdot (\rho \vec{u} \otimes \vec{u}) = \nabla \cdot \begin{bmatrix} \rho u^2 & \rho uw \\ \rho uw & \rho w^2 \end{bmatrix} \tag{A2}$$

We then perform the dot product, which will then result in a vector:

$$\nabla \cdot \begin{bmatrix} \rho u^2 & \rho uw \\ \rho uw & \rho w^2 \end{bmatrix} = \left(\frac{\partial \rho u^2}{\partial x} + \frac{\partial \rho uw}{dy}, \frac{\partial \rho uw}{\partial x} + \frac{\partial \rho w^2}{dy} \right) \tag{A3}$$

Using $\nabla P = \left(\frac{\partial P}{\partial x}, \frac{\partial P}{\partial y}\right)$, , inserting equation (A3) into equation (A1) and dividing the two dimensions into two seperate equations we get:

$$\begin{aligned} \frac{\partial \rho u}{\partial t} &= -\frac{\partial \rho u^2}{\partial x} - \frac{\partial \rho uw}{\partial y} - \frac{\partial P}{\partial x} \\ \frac{\partial \rho w}{\partial t} &= -\frac{\partial \rho uw}{\partial x} - \frac{\partial \rho w^2}{\partial y} - \frac{\partial P}{\partial y} + \rho g_y \end{aligned} \tag{A4}$$

Appendix B: Energy equation

Starting with equation (B1), we wish to find (B4.

$$\frac{\partial e}{\partial t} + \nabla \cdot (e\vec{u}) = -P \nabla \cdot \vec{u} \tag{B1}$$

$$\nabla \cdot (e\vec{u}) = (\frac{\partial eu}{\partial x} + \frac{\partial ew}{\partial y} \tag{B2}$$

$$P \nabla \cdot \vec{u} = P \left(\frac{\partial u}{\partial x} + \frac{\partial w}{\partial y} \right) \tag{B3}$$

By moving $\nabla \cdot (e\vec{u})$ to the right side in equation (B1) and inserting equations (B2) and (B3) we end up with equation (B4).—x

$$\frac{\partial e}{\partial t} = -\frac{\partial eu}{\partial x} - \frac{\partial ew}{\partial y} - P \left(\frac{\partial y}{\partial x} + \frac{\partial w}{\partial y} \right) \tag{B4}$$

Appendix C: Initial T and P

We begin with equation (C1):

$$\nabla^* = \frac{\partial \ln T}{\partial \ln P} \tag{C1}$$

This can be rewritten as equation (C2) using the chain rule:

$$\begin{aligned} \nabla^* &= \frac{P}{T} \frac{\partial T}{\partial P} \\ \nabla^* &= \frac{P}{T} \frac{\partial T}{\partial y} \frac{\partial y}{\partial P} \end{aligned} \tag{C2}$$

Using the condition of hydrostatci equilibrium in equation (C3) we get equation (C4).

$$\frac{\partial P}{\partial y} = -\rho g_y \tag{C3}$$

$$\nabla^* = -\frac{P}{T} \frac{\partial T}{\partial y} \frac{1}{\rho g_y} \tag{C4}$$

Using the ideal gas law, we have that $P = \frac{\rho}{\mu m_u} k_B T$. We insert this into equation (C4) and then set dT to be alone to get equation (C5).

$$\begin{aligned} \nabla^* &= -\frac{\rho}{\mu m_u} k_B T \frac{1}{T} \frac{\partial T}{\partial y} \frac{1}{\rho g_y} \\ \nabla^* &= -\frac{k_B}{\mu m_u} \frac{\partial T}{\partial y} \frac{1}{g_y}, \quad \text{solve for dT} \\ dT &= -\nabla^* \frac{\mu m_u}{k_B} g_y dy \end{aligned} \tag{C5}$$

Next we integrate from the top of the computational box to a radius R with temperature T:

$$\begin{aligned} \int_{T_0}^{T} dT &= \int_{R_0}^{R} -\nabla^* \frac{\mu m_u}{k_B} g_y dy \\ T - T_0 &= -\nabla^* \frac{\mu m_u}{k_B} g_y (R - R_0) \\ T &= T_0 - \nabla^* \frac{\mu m_u}{k_B} g_y (R - R_0) \end{aligned} \tag{C6}$$

We now need an expression for pressure, which we find from the first part of equation (C2).

$$\frac{1}{P} dP = \frac{1}{\nabla} \frac{1}{T} dT \tag{C7}$$

We integrate again, with the same temperature limit as in equation (C6) and a similiar pressure limit:

$$\begin{aligned} \int_{P_0}^{P} \frac{1}{P'} dP' &= \int_{T_0}^{T} \frac{1}{\nabla} \frac{1}{T'} dT' \\ \ln \frac{P}{P_0} &= \ln \left(\frac{T}{T_0} \right)^{1/\nabla} \\ \frac{P}{P_0} &= \left(\frac{T}{T_0} \right)^{1/\nabla} \\ P &= P_0 \left(\frac{T}{T_0} \right)^{1/\nabla} \end{aligned} \tag{C8}$$