Project 3 - AST3310

Jonas Thoen Faber

May 20, 2020

Introduction

There exists a lot of captured images of the Sun showing multiple "bubble's" at the surface if taken at the right wavelength. These "bubble" patterns are better known as convection cells and can be modelled with an numerical approach. Because of the limitation on an everyday laptop, we will look at the convection zones in two dimensions only. That is in both an horizontal direction and a vertical direction. By vertical direction means looking towards the center of the Sun if you are standing at the surface. The goal of this project is to better understand how these convection cells occurs based on some hydrodynamic equations. We quickly notice that we must present some assumptions to make the code work properly. The simulations is therefore just an simplification of the real thing but still produce some interesting results.

Method

The model will cover a box in the outer part of the Sun at the surface. The box is two dimensional and covers 12 Mm and 4 Mm in the horizontal and vertical direction respectively. The horizontal direction will be referred to as the x-direction and will be split into 300 sections while the vertical y-direction is split into 100 sections. To make a model of a stellar convection, we must introduce some equations first. The equations will be used to write a two dimensional hydrodynamic code. The first one is the continuity equation with no sources or sinks (Equation 1), the second is the momentum equation in two dimensions including gravity with no viscous stress tensor (Equation 2) and the last equation is the energy equation (Equation 3). These three goes as following:

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

$$\frac{\partial \rho \mathbf{u}}{\partial t} + \nabla \cdot (\rho \mathbf{u}\mathbf{u}) = -\nabla P + \rho \mathbf{g} \tag{2}$$

$$\frac{\partial e}{\partial t} + \nabla \cdot (e\mathbf{u}) = -P\nabla \cdot \mathbf{u} \tag{3}$$

where ρ is mass density, ∇ is the vector differential operator, $\mathbf{u} = (u, w)$ is the flow vector in x- and y-direction respectively, P is the pressure, $\mathbf{g}$ is the gravity vector and e is the energy density. We will in this project use an explicit numerical scheme to solve the equation above. Before we can discretize the hydrodynamic equations, we need to split them into components of x- and y-directions. Equation 1 becomes:

$$\frac{\partial \rho}{\partial t} = -\nabla \cdot (\rho \mathbf{u}) = -\left(\frac{\partial \rho u}{\partial x} + \frac{\partial \rho w}{\partial y}\right) \tag{4}$$

Equation 2 is further split into an horizontal and vertical equation which refer to x- and y-direction respectively. Here we will define the direction on x and y with the unit vectors $\vec{i}$ and $\vec{j}$. To make the derivation a little simpler, each side of the equal sign is done separately starting off with the left side:

$$\frac{\partial \rho u}{\partial t}\vec{i} + \frac{\partial \rho w}{\partial t}\vec{j} + \left(\frac{\partial}{\partial x}\vec{i} + \frac{\partial}{\partial y}\vec{j}\right)\rho\left(u\vec{i} + w\vec{j}\right)\left(u\vec{i} + w\vec{j}\right) \tag{5}$$

$$= \frac{\partial \rho u}{\partial t}\vec{i} + \frac{\partial \rho w}{\partial t}\vec{j} + \left(\frac{\partial \rho u}{\partial x} + \frac{\partial \rho w}{\partial y}\right)\left(u\vec{i} + w\vec{j}\right) \tag{6}$$

$$= \frac{\partial \rho u}{\partial t}\vec{i} + \frac{\partial \rho w}{\partial t}\vec{j} + \frac{\partial \rho u^2}{\partial x}\vec{i} + \frac{\partial \rho u w}{\partial x}\vec{j} + \frac{\partial \rho u w}{\partial y}\vec{i} + \frac{\partial \rho w^2}{\partial y}\vec{j} \tag{7}$$

$$= \left(\frac{\partial \rho u}{\partial t} + \frac{\partial \rho u^2}{\partial x} + \frac{\partial \rho u w}{\partial y}, \frac{\partial \rho w}{\partial t} + \frac{\partial \rho u w}{\partial x} + \frac{\partial \rho w^2}{\partial y}\right) \tag{8}$$

Note that the gravity works only in the y-direction as the gravity cancel out in the x-direction assuming a star that is spherical symmetric. The right side of Equation 2 becomes then:

$$-\frac{\partial P}{\partial x}\vec{i} - \frac{\partial P}{\partial y}\vec{j} + \rho g\vec{j} \tag{9}$$

$$= \left(-\frac{\partial P}{\partial x}, -\frac{\partial P}{\partial y} + \rho g\right) \tag{10}$$

So if we decompose Equation 8 and Equation 10 into two equations vector-wise, we get:

$$\frac{\partial \rho u}{\partial t} = -\frac{\partial \rho u^2}{\partial x} - \frac{\partial \rho u w}{\partial y} - \frac{\partial P}{\partial x} \tag{11}$$

and:

$$\frac{\partial \rho w}{\partial t} = -\frac{\partial \rho u w}{\partial x} - \frac{\partial \rho w^2}{\partial y} - \frac{\partial P}{\partial y} + \rho g \tag{12}$$

And lastly Equation 3 becomes:

$$\frac{\partial e}{\partial t} = -\frac{\partial eu}{\partial x} - \frac{\partial ew}{\partial y} - P\left(\frac{\partial u}{\partial x} + \frac{\partial w}{\partial y}\right) \tag{13}$$

Now that we have split all the hydrodynamic equations into components of x and y, the next step is to discretise them so that we get $\rho^n_{i,j}$, $u^n_{i,j}$, $w^n_{i,j}$ and $e^n_{i,j}$. i and j are the index to be advanced in x- and y-direction and n are the index in the direction of time. The continuity part $\rho^n_{i,j}$ and momentum part $u^n_{i,j}$ is already known so we need only to find the last two terms. First we look at Equation 12 by expanding it using the product rule:

$$\begin{aligned}\frac{\partial \rho w}{\partial t} = &-\rho w\left(\frac{\partial u}{\partial x} + \frac{\partial w}{\partial y}\right) \\ &- u\frac{\partial \rho w}{\partial x} - w\frac{\partial \rho w}{\partial y} - \frac{\partial P}{\partial y} + \rho g\end{aligned} \tag{14}$$

$$\begin{aligned}\Rightarrow \left[\frac{\partial \rho w}{\partial t}\right]^n_{i,j} = &-[\rho w]^n_{i,j}\left(\left[\frac{\partial u}{\partial x}\right]^n_{i,j} + \left[\frac{\partial w}{\partial y}\right]^n_{i,j}\right) - u^n_{i,j}\left[\frac{\partial \rho w}{\partial x}\right]^n_{i,j} \\ &- w^n_{i,j}\left[\frac{\partial \rho w}{\partial y}\right]^n_{i,j} - \left[\frac{\partial P}{\partial y}\right]^n_{i,j} + \rho^n_{i,j} g\end{aligned} \tag{15}$$

The left-hand side of Equation 15 is discretised using forward time. To prevent occurrence of numerical errors, all the gradients in the discretised equation will be calculated separately. The pressure gradient are approximated by using central differencing and the spatial gradients by using upwind differencing as stability may be a concern. The method is chosen based on whether each time differential are a product with the horizontal or vertical velocity u or w. We get:

$$\left[\frac{\partial u}{\partial x}\right]^n_{i,j} = \begin{cases} \frac{u^n_{i,j} - u^n_{i-1,j}}{\Delta x} & \text{if} \quad w^n_{i,j} \geq 0 \\ \frac{u^n_{i+1,j} - u^n_{i,j}}{\Delta x} & \text{if} \quad w^n_{i,j} < 0 \end{cases} \tag{16}$$

$$\left[\frac{\partial w}{\partial y}\right]^n_{i,j} = \begin{cases} \frac{w^n_{i,j} - w^n_{i,j-1}}{\Delta y} & \text{if} \quad w^n_{i,j} \geq 0 \\ \frac{w^n_{i,j+1} - w^n_{i,j}}{\Delta y} & \text{if} \quad w^n_{i,j} < 0 \end{cases} \tag{17}$$

$$\left[\frac{\partial \rho w}{\partial x}\right]^n_{i,j} = \begin{cases} \frac{[\rho w]^n_{i,j} - [\rho w]^n_{i-1,j}}{\Delta x} & \text{if} \quad w^n_{i,j} \geq 0 \\ \frac{[\rho w]^n_{i+1,j} - [\rho w]^n_{i,j}}{\Delta x} & \text{if} \quad w^n_{i,j} < 0 \end{cases} \tag{18}$$

$$\left[\frac{\partial \rho w}{\partial y}\right]^n_{i,j} = \begin{cases} \frac{[\rho w]^n_{i,j} - [\rho w]^n_{i,j-1}}{\Delta y} & \text{if} \quad u^n_{i,j} \geq 0 \\ \frac{[\rho w]^n_{i,j+1} - [\rho w]^n_{i,j}}{\Delta y} & \text{if} \quad u^n_{i,j} < 0 \end{cases} \tag{19}$$

$$\left[\frac{\partial P}{\partial y}\right]^n_{i,j} = \frac{P^n_{i,j+1} - P^n_{i,j-1}}{2\Delta y} \tag{20}$$

Both central scheme and upwind scheme is defined as separate functions in the code for x and y directions as the methods are used multiple times. They are called `central_x`, `central_y`, `upwind_x` and `upwind_y`. We now have the discretised expressions on the right-hand side of Equation 15 and since the left-hand side is discretised using forward time, we are then left with:

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

where we already have calculated $\rho_{i,j}^{n+1}$. Now we need to discretise the last expression, Equation 13. As the flow (u, w) is non-constant, the spatial terms need to be split into additional terms:

$$\frac{\partial e}{\partial t} = -(e+P)\left(\frac{\partial u}{\partial x} + \frac{\partial w}{\partial y}\right) - u\frac{\partial e}{\partial x} - w\frac{\partial e}{\partial y} \tag{22}$$

$$\begin{aligned} \Rightarrow \left[\frac{\partial e}{\partial t}\right]_{i,j}^n &= -(e_{i,j}^n + P_{i,j}^n)\left(\left[\frac{\partial u}{\partial x}\right]_{i,j}^n + \left[\frac{\partial w}{\partial y}\right]_{i,j}^n\right) \\ &\quad - u_{i,j}^n \left[\frac{\partial e}{\partial x}\right]_{i,j}^n - w_{i,j}^n \left[\frac{\partial e}{\partial y}\right]_{i,j}^n \end{aligned} \tag{23}$$

Both $\partial u/\partial x$ and $\partial w/\partial y$ is found using central differential scheme while $\partial e/\partial x$ and $\partial e/\partial y$ is approximated by using upwind differential scheme as it is more stable. Each of the gradients are given by the following:

$$\left[\frac{\partial u}{\partial x}\right]_{i,j}^n = \frac{u_{i+1,j}^n - u_{i-1,j}^n}{2\Delta x} \tag{24}$$

$$\left[\frac{\partial w}{\partial y}\right]_{i,j}^n = \frac{w_{i,j+1}^n - w_{i,j-1}^n}{2\Delta y} \tag{25}$$

$$\left[\frac{\partial e}{\partial x}\right]_{i,j}^n = \begin{cases} \frac{e_{i,j}^n - e_{i-1,j}^n}{\Delta x} & \text{if} \quad u_{i,j}^n \geq 0 \\ \frac{e_{i+1,j}^n - e_{i,j}^n}{\Delta x} & \text{if} \quad u_{i,j}^n < 0 \end{cases} \tag{26}$$

$$\left[\frac{\partial e}{\partial y}\right]_{i,j}^n = \begin{cases} \frac{e_{i,j}^n - u_{i,j-1}^n}{\Delta y} & \text{if} \quad w_{i,j}^n \geq 0 \\ \frac{e_{i,j+1}^n - u_{i,j}^n}{\Delta y} & \text{if} \quad w_{i,j}^n < 0 \end{cases} \tag{27}$$

Now that we have everything we need to estimate the right-hand side of Equation 23, we get using forward time on the energy equation:

$$e_{i,j}^{n+1} = e_{i,j}^n + \left[\frac{\partial e}{\partial t}\right]_{i,j}^n \Delta t \tag{28}$$

We now have the algorithms for solving $\rho_{i,j}^{n+1}$, $u_{i,j}^{n+1}$, $w_{i,j}^{n+1}$ and $e_{i,j}^{n+1}$. We need to determine the initial conditions for the temperature T, pressure P, density ρ and the energy e assuming it to vary only in the vertical direction at first. This is done by assuming an ideal gas in hydrostatic equilibrium in the beginning. Then we know that:

$$\frac{dP}{dy} = -\rho g \tag{29}$$

And by disctretizing, we get:

$$dP = -\rho g dy \tag{30}$$

$$\Rightarrow P^n_{i,j+1} - P^n_{i,j} = -\rho^n_{i,j} g \Delta y \tag{31}$$

$$\Rightarrow P^n_{i,j+1} = P^n_{i,j} - \rho^n_{i,j} g \Delta y \tag{32}$$

To find an expression for the density, we will use the internal energy that is given by the equation of state for an ideal gas:

$$e = \frac{1}{(\gamma - 1)} \frac{\rho}{\mu m_u} k_B T \tag{33}$$

Where γ is the ratio between the specific heats c_P and c_V, μ is the mean molecular weight, m_u is the atomic mass unit and k_B is the Boltzmann constant. γ may also be written as $\gamma = 1 + 2/f$ where f is the degrees of freedom. For an ideal gas, the degrees of freedom is given as $f = 3$. By flipping the energy equation so that we have the density on the left side and discretize it, we get:

$$\rho^n_{i,j} = \frac{e^n_{i,j}(\gamma - 1)\mu m_u}{k_B T^n_{i,j}} \tag{34}$$

If we look at Equation 33 and remember that we are assuming an ideal gas, we have that

$$P = (\gamma - 1)e \tag{35}$$

and when discretized, it leaves us with:

$$e^n_{i,j} = \frac{P^n_{i,j}}{(\gamma - 1)} \tag{36}$$

In this project, we assume the temperature to be slightly super adiabatic and since we are dealing with an ideal gas, the temperature gradient is found by:

$$\nabla = -\frac{H_P}{T}\frac{\partial T}{\partial y} \tag{37}$$

where the temperature gradient ∇ in this project is defined as $\nabla = 0.40001$ to simulate a slightly super adiabatic process. H_P is the pressure scale height and is defined as:

$$H_P = -P\frac{\partial y}{\partial P} = -P\left(\frac{\partial P}{\partial y}\right)^{-1} = -P\left(-\rho g\right)^{-1} = \frac{P}{\rho g} \tag{38}$$

Implementing the last equation into the previous one and discretize it gives:

$$\nabla = -\frac{P}{\rho g T}\frac{\partial T}{\partial y} \tag{39}$$

$$\Rightarrow \partial T = -\rho g \nabla \frac{T}{P}\partial y \tag{40}$$

$$\Rightarrow T^n_{i,j+1} = T^n_{i,j} - \rho^n_{i,j} g \nabla \frac{T^n_{i,j}}{P^n_{i,j}}\Delta y \tag{41}$$

The measured values of the temperature and the pressure on the photosphere of the Sun is used. With these parameters, we can update the initial values of T, P, e and ρ starting from the surface of the Sun and updating the values towards the center. When using the values found through the initialisation, the code should be stable for at least 60 seconds. This will be implemented as a sanity check. After the sanity has passed, an Gaussian perturbation is implemented in the initial temperature to provoke the gas to become unstable. At this point we need to recalculate the density in the box but will assume the pressure and therefore also energy density to stay the same.

Since the hydrodynamic equations can not be solved at the boundaries, some boundary conditions must be implemented. The horizontal boundary conditions is implemented as periodic such that $\phi^n_{-1,j} = \phi_{nx-1,j}$ and $\phi^n_{0,j} = \phi_{nx,j}$ where $nx - 1$ is the last index in the horizontal direction. This condition applies for all of the primary parameters. The vertical velocity at the vertical boundary is set to equal zero at all time as we assume no heat enter or escapes the box. The gradient of the horizontal velocity at the vertical boundary should be zero. With this assumption, we may use the forward and backward difference approximation to find the appropriate variables. The forward approximation goes like this:

$$\left[\frac{\partial \phi}{\partial y}\right]^n_{i,j} = \frac{-\phi^n_{i,j+2} + 4\phi^n_{i,j+1} - 3\phi^n_{i,j}}{2\Delta y} \tag{42}$$

and the backward approximation:

$$\left[\frac{\partial \phi}{\partial y}\right]^n_{i,j} = \frac{3\phi^n_{i,j} - 4\phi^n_{i,j-1} + \phi^n_{i,j-2}}{2\Delta y} \tag{43}$$

The boundary condition for both density and energy must be implemented with cautious as they are coupled. An assumption of an ideal gas at hydrostatic equilibrium at the vertical boundary at all time is set. We may then use Equation 35 to determine the energy. Also, constant temperature and pressure at the vertical boundaries are implemented after the initialisation as we do not have enough information or equations to update all the primary variables. Now that temperature is constant and the energy is found at the boundaries, the density is calculated using Equation 34. The boundary conditions is called upon as a separate function after the update of the primary variables in each iteration as `boundary_conditions`.

Next we need to find an appropriate time step length. This will be calculated through every iteration by insisting that none of the primary variables gets changed by more than a defined percentage. A method that includes the already calculated time derivatives of the primary variables will be used. The method goes like so:

$$\text{rel}(\phi) = \left| \frac{\partial \phi}{\partial t} \cdot \frac{1}{\phi} \right| \tag{44}$$

Since we do not have $\partial u/\partial t$ or $\partial u/\partial t$, we need to calculate $\text{rel}(\rho u)$ and $\text{rel}(\rho w)$ first and then divide each term by $\text{rel}(\rho)$. Another important implementation is that a particle does not exceed a whole grid point. To find this, we will use the calculated velocities u and W like so:

$$\text{rel}(x) = \left| \frac{u}{\Delta x} \right| \tag{45}$$

and:

$$\text{rel}(y) = \left| \frac{w}{\Delta y} \right| \tag{46}$$

Next is to find the maximum value δ of all the relative changes to determine the next time step length. This is done by insisting that the product of δ and Δt is equal to the acceptable defined percentage p, or in this case $p = 0.1$. The next time step length is determined by:

$$\Delta t = \frac{p}{\delta} \tag{47}$$

The time step length is calculated at every iterations in the solver and is called upon as a function `timestep`. A limitation is set so that we do not get an extreme value of the time step length. As most of the elements in the u and w arrays is stationary, they may cause a division by zero or nearly zero and result in a time step length which is not of interest.

We now have everything we need to create a model of a two dimensional convection in a star. The algorithm will solve the hydrodynamic equations in such a way that we can plot it and see the evolution in time.

Results

From Figure 1 we notice how the temperature has not changed during the 60 seconds of simulation. The result tells us that the code stays stable in hydrostatic equilibrium. We should take notice that the vertical velocity is not zero at all time as seen from Figure 2. The velocities are indeed small relative to the size of the box. The simulations of the vertical velocity becomes

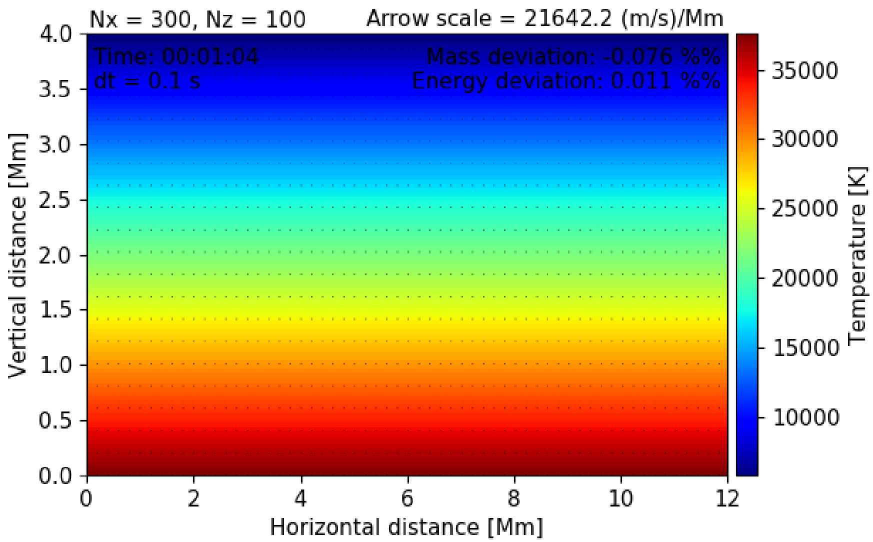

Figure 1: Plot of the temperature after 60 seconds of simulation without perturbation. The plot shows a section of the Sun near the surface.

more and more unstable as time goes by as a result of an numerical approach. Figure 3 show the result as an Gaussian perturbation is added to the initial temperature. After some time the convection becomes noticeable as the arrow shows the direction of the flow as the hydrodynamic solver progress with time. The convection cell rises towards the surface of the Sun and since nothing escapes the box, the gas spreads towards the horizontal direction of the box.

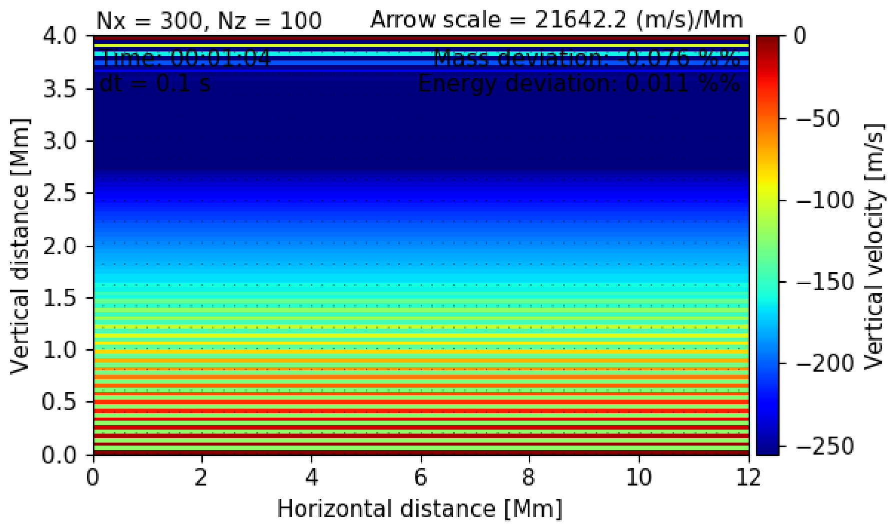

Figure 2: Plot of the vertical velocity after 60 seconds of simulation without perturbation. The plot shows a section of the Sun near the surface.

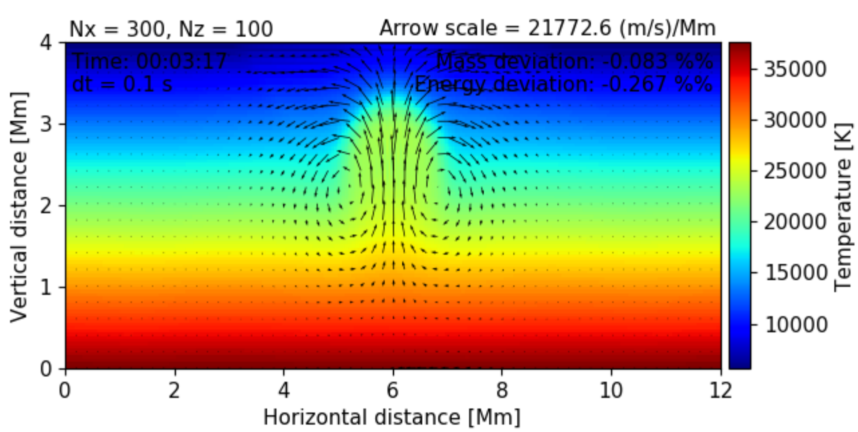

Figure 3: Plot of a convection near the surface of the Sun. An Gaussian perturbation is added to the initial temperature. The figure shows how the convection as evolved after about 200 seconds of simulation.

Discussion

Using the continuity Equation 1, momentum Equation 2 and the energy Equation 3 does give some interesting plots, especially when the Gaussian perturbation is implemented. We observe from Figure 3 that after about 200 seconds in real time, a convection in the gas occurs which was our main goal in this project. The flow is greatest at the center of the perturbation where it goes up toward the surface of the Sun. This makes sense as the perturbation is an implementation to make the gas unstable. The part of the box that is not changed stays in hydrostatic equilibrium until the flow of the gas spreads out. The gas spreads up and outside of the perturbation spot and then inward to the center of the Sun and again towards the center of the gas convection. The gas behaves like an bubble under water as it is "lighter" than the surroundings and pushes itself up towards the surface.

We notice from Figure 2 that the horizontal velocity is not zero even without perturbation. If we look at Equation 15, it contains two terms that should cancel out at hydrostatic equilibrium. That is:

$$-\left[\frac{\partial P}{\partial y}\right]_{i,j}^{n} + \rho_{i,j}^{n} g = 0 \tag{48}$$

This is not the case when we are solving this problem numerically. There will be an slight difference between these two terms which result in a small change in the horizontal velocity. As the solver keeps updating our values, this change gets eventually larger and causes the solver to become unstable. Therefore the algorithm is only stable in hydrostatic equilibrium for a limited period of time.

Conclusion

Calculating the time step length did not work properly as it often became too large or too small. This often had something to do with the elements in the horizontal velocity u matrix. The time step length is therefore calculated through every iterations but almost never used.

There has been some difficulties trying to make the code work properly. That is since we need to be careful whether to update the parameters with respect to space or time and therefore use the correct equations. With trial and errors we finally got to satisfying results even with slightly numerically errors.

It has been very interesting to use the hydrodynamic equations to simulate a convection in the gas within a star. It resulted in better understanding of why the convection cells occurs on the surface of the Sun when looking at it with a telescope. The project demanded a lot of testing and failures but it was interesting and educational nevertheless. Animations is added in the folder for both cases of hydrostatic equilibrium and perturbation in the initial temperature. The first case visualize temperature and vertical velocity only while the last case visualize for temperature, pressure, density, energy, horizontal velocity and vertical velocity.