AST3310 - Project 3: Stellar convection

Candidate number: 15015
(Dated: May 16, 2021)

We have performed 2D simulations of the gas near the surface of our sun, and have shown that adding a temperature perturbation to the gas' hydrostatic state causes motion similar to convection if we let the pressure stay as in hydrostatic equilibrium, or causes motion similar to an explosion if we update the pressure after adding the temperature perturbation.

INTRODUCTION

Stars are concentrations of gas held together by gravity. They are generally stable on human timescales and the gas they are composed of must therefore as a whole be in hydrostatic equilibrium. However, the gas that makes up a star is not in general completely static. In a star's convective zones hot pockets of gas will flow upwards, carrying energy toward the surface before they sink down again. In this project we will attempt to simulate the gas in our sun's convection zone as a fluid, and observe how small perturbations in temperature may induce convection in the gas. The simulation will be in 2D, and to model the gas we will make use of a set of partial differential equations which describe the motion of fluids, called the continuity equation (with no sinks/sources) (1), the momentum equation (2) and the energy equation (3).

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

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

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

Where ρ is the mass density, $\mathbf{u}$ is the velocity, e is the energy density, P is the pressure and $\mathbf{g}$ is the gravitational acceleration. The continuity equation without sinks/sources is the equation for mass conservation; the change in mass density in a volume must come from mass flowing in/out of the volume. The momentum equation makes sure that the change of momentum in a volume comes from either internal- and external forces like the pressure gradient and gravity, or from momentum flowing in/out of the volume. Finally the energy equation makes sure that the change in energy density in a volume comes from energy flowing in/out of the volume or from pressure compressing/expanding the gas in the volume, which increases/decreases the energy density. To solve these equations numerically we will need to discretise them and choose reasonable boundary conditions. We will also need initial conditions for the variables in the equations, which will be chosen to be that of hydrostatic equilibrium. Finding hydrostatic equilibrium requires that we know the relationships between variables such as mass density, pressure, energy and temperature T. In this project we will assume we are dealing with an ideal gas, which gives the following relations:

$$e = \frac{1}{(\gamma - 1)} \frac{k_B T}{\mu m_u} \rho \tag{4}$$

$$P = (\gamma - 1) e \tag{5}$$

Where $\gamma = 5/3$ is the ratio of specific heats for an ideal gas, k_B is Boltzmann's constant, m_u is the atomic mass unit and μ is the mean molecular weight of the gas. We will assume here that $\mu = 0.61$. After having finding hydrostatic equilibrium, we will add a Gaussian temperature perturbation observe how the gas reacts and see if we get convection.

METHODS

Discretising

To make discretising easier we can rewrite the equations (1),(2),(3). Starting with the continuity equation (1) we have

$$\frac{\partial \rho}{\partial t} = -\nabla \cdot (\rho \mathbf{u})$$

With $\mathbf{u} = (u, w)$ and knowing the nabla operator in two dimensions can be written as $\nabla = (\frac{\partial}{\partial x}, \frac{\partial}{\partial y})$ we have

$$\frac{\partial \rho}{\partial t} = -\frac{\partial \rho u}{\partial x} - \frac{\partial \rho w}{\partial y} \tag{6}$$

The momentum equation (2) is a vector equation and can thus be split into two equations; one with the vertical- and one with the horizontal components. The second term on the left hand side contains the product $\mathbf{uu}$, which called an outer vector product and returns a matrix with components $(\mathbf{uu})_{ij} = u_i u_j$ so that

$$\mathbf{uu} = \begin{bmatrix} u^2 & uw \\ uw & w^2 \end{bmatrix}$$

multiplying with rho and then taking the dot product of this matrix with the nabla operator gives

$$\nabla \cdot (\rho \mathbf{u}\mathbf{u}) = \begin{bmatrix} \frac{\partial}{\partial x} & \frac{\partial}{\partial y} \end{bmatrix} \cdot \begin{bmatrix} \rho u^2 & \rho u w \\ \rho u w & \rho w^2 \end{bmatrix}$$
$$= \begin{bmatrix} \frac{\partial \rho u^2}{\partial x} + \frac{\partial \rho u w}{\partial y} & \frac{\partial \rho u w}{\partial x} + \frac{\partial \rho w^2}{\partial y} \end{bmatrix}$$

We can now write the horizontal component of the momentum equation as

$$\frac{\partial \rho u}{\partial t} = -\frac{\partial \rho u^2}{\partial x} - \frac{\partial \rho u w}{\partial y} - \frac{\partial P}{\partial x} \tag{7}$$

and similarly the vertical component can be written as

$$\frac{\partial \rho w}{\partial t} = -\frac{\partial \rho w^2}{\partial y} - \frac{\partial \rho u w}{\partial x} - \frac{\partial P}{\partial y} + \rho g_y \tag{8}$$

Here we have assumed that gravity works in the y-direction only. We can assume this since the slice of the sun we are simulating is located far enough from the centre of the sun that the vectors pointing to the centre (the direction gravity is pulling) are roughly parallel for all the points in our slice. Now, moving around the terms in the energy equation (3) we get

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

which gives the rewritten equation

$$\frac{\partial e}{\partial t} = -\frac{\partial e u}{\partial x} - \frac{\partial e w}{\partial y} - P\left(\frac{\partial u}{\partial x} + \frac{\partial w}{\partial y}\right) \tag{9}$$

To solve these equations numerically we need to discretise them. In the project text we are given the discrete versions of the continuity equation and the horizontal momentum equation. We are therefore left with discretising the vertical momentum equation to find the vertical velocity w_{ij}^{n+1} and the energy equation to find e_{ij}^{n+1}. Let's start with the vertical momentum equation; we can split (8) into additional terms by use of the product rule for derivatives to get

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

which on discretised form is

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

where we have written the vertical gravitational acceleration as $g_y = -g$ with $g > 0$. We write the derivatives as

$$\left[\frac{\partial u}{\partial x}\right]_{i,j}^n \approx \begin{cases} \frac{u_{i,j}^n - u_{i-1,j}^n}{\Delta x} & \text{if } w_{i,j}^n \geq 0 \\ \frac{u_{i+1,j}^n - u_{i,j}^n}{\Delta x} & \text{if } w_{i,j}^n < 0 \end{cases}$$
$$\left[\frac{\partial w}{\partial y}\right]_{i,j}^n \approx \begin{cases} \frac{w_{i,j}^n - w_{i,j-1}^n}{\Delta y} & \text{if } w_{i,j}^n \geq 0 \\ \frac{w_{i,j+1}^n - w_{i,j}^n}{\Delta y} & \text{if } w_{i,j}^n < 0 \end{cases}$$
$$\left[\frac{\partial \rho w}{\partial x}\right]_{i,j}^n \approx \begin{cases} \frac{[\rho w]_{i,j}^n - [\rho w]_{i-1,j}^n}{\Delta x} & \text{if } u_{i,j}^n \geq 0 \\ \frac{[\rho w]_{i+1,j}^n - [\rho w]_{i,j}^n}{\Delta x} & \text{if } u_{i,j}^n < 0 \end{cases}$$
$$\left[\frac{\partial \rho w}{\partial y}\right]_{i,j}^n \approx \begin{cases} \frac{[\rho w]_{i,j}^n - [\rho w]_{i,j-1}^n}{\Delta y} & \text{if } w_{i,j}^n \geq 0 \\ \frac{[\rho w]_{i,j+1}^n - [\rho w]_{i,j}^n}{\Delta y} & \text{if } w_{i,j}^n < 0 \end{cases}$$
$$\left[\frac{\partial P}{\partial y}\right]_{i,j}^n \approx \frac{P_{i,j+1}^n - P_{i,j-1}^n}{2\Delta y}$$

The left hand side can be discretised to

$$\left[\frac{\partial \rho w}{\partial t}\right]_{i,j}^n \approx \frac{[\rho w]_{i,j}^{n+1} - [\rho w]_{i,j}^n}{\Delta t}$$

which gives an expression for $w_{i,j}^{n+1}$:

$$w_{i,j}^{n+1} = \frac{[\rho w]_{i,j}^n + \left[\frac{\partial \rho w}{\partial t}\right]_{i,j}^n \Delta t}{\rho_{i,j}^{n+1}} \tag{10}$$

Finding the discrete version of the energy equation we can start with splitting some of the terms in (9) using the product rule for derivatives again:

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

which on discretised form is

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

where we have used expression for the equation of state of an ideal gas (5) to see that $e + P = \gamma e$. The derivatives on the right hand side can be discretised as

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

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

$$\left[\frac{\partial u}{\partial x}\right]^n_{i,j} \approx \frac{u^n_{i+1,j} - u^n_{i-1,j}}{2\Delta x}$$

$$\left[\frac{\partial w}{\partial y}\right]^n_{i,j} \approx \frac{w^n_{i,j+1} - w^n_{i,j-1}}{2\Delta y}$$

The two energy gradients are calculated using upwind differencing because they are both connected to a flow, i.e. the horizontal- and vertical energy gradients are multiplied with the horizontal velocity $u^n_{i,j}$ and the vertical velocity $w^n_{i,j}$ respectively. The left hand side of the energy equation can be discretised as

$$\left[\frac{\partial e}{\partial t}\right]^n_{i,j} \approx \frac{e^{n+1}_{i,j} - e^n_{i,j}}{\Delta t}$$

which gives an expression for $e^{n+1}_{i,j}$:

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

Initial conditions

The project text gives two criteria for the initial conditions of our simulation:

1. The gas needs to be in hydrostatic equilibrium
2. The double logarithmic gradient
$$\nabla = \frac{\partial \ln T}{\partial \ln P} \tag{12}$$
must be slightly larger than 2/5

A gas is in hydrostatic equilibrium when the gas' pressure gradient force cancels out any external forces acting on it. In our case the external force is gravity and as a consequence the pressure gradient can be written as

$$\frac{\partial P}{\partial y} = -\rho g \tag{13}$$

where the gradient is in the y-direction since this is the direction gravity is pointing. Rewriting the double logarithmic gradient (12) we get

$$\nabla = \frac{P}{T}\frac{\partial T}{\partial P} \tag{14}$$

which can also be written as

$$\nabla = \frac{P}{T}\frac{\partial T/\partial y}{\partial P/\partial y}.$$

Since we are assuming an ideal gas we know that $\rho = \mu m_u P/k_B T$. Inserting (13) into the expression for the double logarithmic gradient above we get

$$\nabla = \frac{P}{T}\frac{\partial T/\partial y}{-\rho g} = \frac{P}{T}\frac{\partial T/\partial y}{-\frac{\mu m_u g}{k_B T}P} = \frac{\partial T/\partial y}{-\frac{\mu m_u g}{k_B}}$$

Rearranging we have

$$\partial T = -\frac{\mu m_u g}{k_B}\nabla \partial y,$$

and taking the integral we get an expression for the temperature as a function of height

$$T = T_0 - \frac{\mu m_u g}{k_B}\nabla(y - y_0) \tag{15}$$

where we assume the gravitational acceleration to be constant $g = GM_\odot/R_\odot^2$ in our simulation box. T_0 is taken to be the temperature of the sun's photosphere and we set $y_0 = R_\odot$, i.e. the sun's radius. To find the pressure we rearrange equation (14) to find that

$$\frac{dP}{P} = \frac{1}{\nabla}\frac{dT}{T}$$
$$\int_{P_0}^{P}\frac{dP}{P} = \frac{1}{\nabla}\int_{T_0}^{T}\frac{dP}{P}$$
$$\ln P/P_0 = \frac{1}{\nabla}\ln T/T_0$$

Taking Euler's number to the power of this we get

$$P = P_0(T/T_0)^{1/\nabla} \tag{16}$$

where P_0 is the pressure at the solar photosphere.

Boundary conditions

The project text gives us that the vertical boundary condition for the horizontal velocity is that the vertical gradient should be zero:

$$\left[\frac{\partial u}{\partial y}\right]^n_{i,0} = \left[\frac{\partial u}{\partial y}\right]^n_{i,-1} = 0 \tag{17}$$

where we take the index $j = 0$ to be the bottom of the simulation box, while the index $j = -1$ is the top. We can find expressions for the gradients by utilising the 3-point forward difference approximation

$$\left[\frac{\partial \phi}{\partial y}\right]_{i,j}^{n} = \frac{-\phi_{i,j+2}^{n} + 4\phi_{i,j+1}^{n} - 3\phi_{i,j}^{n}}{2\Delta y} \tag{18}$$

and the 3-point backward difference approximation

$$\left[\frac{\partial \phi}{\partial y}\right]_{i,j}^{n} = \frac{3\phi_{i,j}^{n} - 4\phi_{i,j-1}^{n} + \phi_{i,j-2}^{n}}{2\Delta y} \tag{19}$$

Inserting (17) into (18) and (19) we get

$$u_{i,0}^{n} = \frac{1}{3}(4u_{i,1}^{n} - u_{i,2}^{n}) \tag{20}$$

$$u_{i,-1}^{n} = \frac{1}{3}(4u_{i,-2}^{n} - u_{i,-3}^{n}) \tag{21}$$

The vertical boundary conditions for energy and density require hydrostatic equilibrium, which means the pressure gradient is the one given in equation (13). Using $\rho = \mu m_u P / k_B T$ and the equation of state $P = (\gamma - 1)e$ we have the condition

$$\frac{\partial e}{\partial y} = -\frac{\mu m_u g}{k_B T} e \tag{22}$$

for the vertical boundaries. Utilising the 3-point difference approximations like we did for the horizontal velocity we get the following expressions for the energy at the boundaries:

$$e_{i,0}^{n} = \frac{4e_{i,1}^{n} - e_{i,2}^{n}}{3 - \frac{\mu m_u g}{k_B T_{i,0}^{n}} 2\Delta y} \tag{23}$$

$$e_{i,-1}^{n} = \frac{4e_{i,-2}^{n} - e_{i,-3}^{n}}{3 + \frac{\mu m_u g}{k_B T_{i,-1}^{n}} 2\Delta y} \tag{24}$$

We see that we need the temperature at the boundaries in order to find the energy. We can find the temperature gradient for hydrostatic equilibrium from the expression we found for the temperature (15) (or by rewriting (16)), and we find that

$$\frac{\partial T}{\partial y} = -\frac{\mu m_u g}{k_B} \nabla \tag{25}$$

Utilising the 3-point difference approximations again like for u and e we find

$$T_{i,0}^{n} = \frac{1}{3}\left(4T_{i,1}^{n} - T_{i,2}^{n} + \frac{\mu m_u g}{k_B} \nabla 2\Delta y\right) \tag{26}$$

$$T_{i,-1}^{n} = \frac{1}{3}\left(4T_{i,-2}^{n} - T_{i,-3}^{n} - \frac{\mu m_u g}{k_B} \nabla 2\Delta y\right) \tag{27}$$

With the temperature and the energy at the boundaries we can calculate the pressure and density from the formulas for ideal gas (4,5).

Temperature perturbation

Having the initial conditions which will produce a system in hydrostatic equilibrium, we can add a temperature perturbation to introduce some movement. This perturbation will take to form of a Gaussian distribution, which in 2D can be written as

$$G(x, y) = A \exp\left(-\frac{(x - x_0)^2 + (y - y_0)^2}{2\sigma^2}\right) \tag{28}$$

where A is the amplitude, σ is the standard deviation and (x_0, y_0) is the position of the Gaussian. With A being some temperature, we can introduce a temperature perturbation by adding (28) to our initial temperature (15). The consequences of adding this perturbation depends on how we calculate the other initial conditions as well; we can for choose whether we want to update the pressure by using (16) after the perturbation or leave as it was for the hydrostatic case.

Code

To perform the simulation we have written a code that can take one state of our box and find the next state, i.e the state after a small time dt. We do this using the discretisations we found for the continuity-, momentum- and energy equations (1,2,3) to find the mass densities, velocities and energy densities at the next step. Using the new mass densities and energy densities we can find the temperature and pressure from our assumption of ideal gas (4,5). Doing this for the points in the box that are not on the vertical boundaries, we can then find the values on the boundaries by the methods discussed in the Boundary conditions section, using the already calculated values for the points not on the boundary. When this new state is calculated we simply use the same method again to find the next one after that.

RESULTS

Exploring stellar convection we have performed three simulations. The first is simply the gas in hydrostatic equilibrium to demonstrate that the code is working. A snapshot of this is included in figure 1.

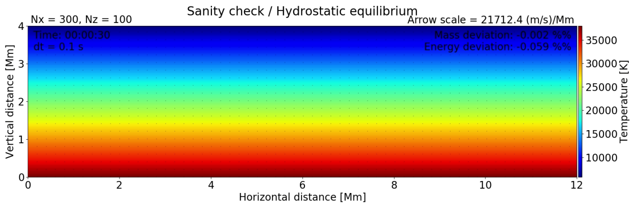

FIG. 1. Figure showing a snapshot of the gas in hydrostatic equilibrium without perturbations, at Time: 00:00:30. Temperature is shown in colour and velocity is shown as vectors.

The second simulation we call the convection simulation, which we get from adding a Gaussian perturbation to the temperature in the centre of the system with amplitude $A = 17000$ K and standard deviation $\sigma = 0.6$ Mm, and not updating the pressure after doing so. This simulation appears to produce convection, which is why that's what it's called. Snapshots from this simulation showing the system's temperature and velocity vectors is shown in figure 2 and 3. Figure 4, 5 and 6 show snapshots of the system's mass density. In figure 7 we have a snapshot with the system's vertical velocity represented in colour.

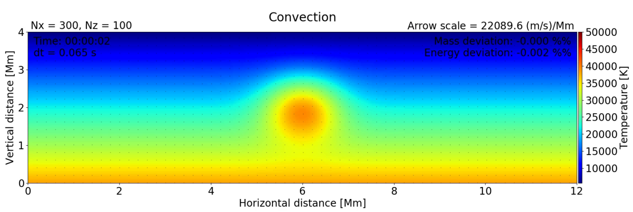

FIG. 2. Figure showing a snapshot of the convection simulation taken at start of simulation, with temperature shown in colour and velocity as vectors

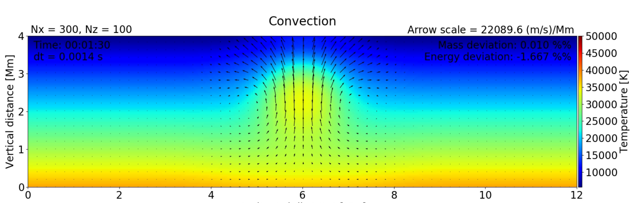

FIG. 3. Figure showing a snapshot of the convection simulation at Time: 00:01:30, with temperature shown in colour and velocity as vectors

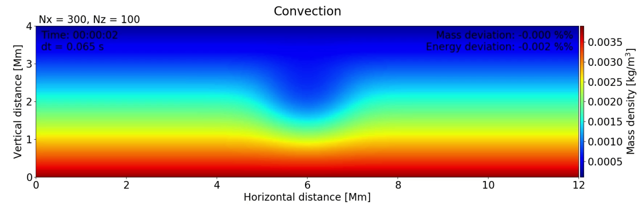

FIG. 4. Figure showing snapshot of the mass density at the start of the convection simulation.

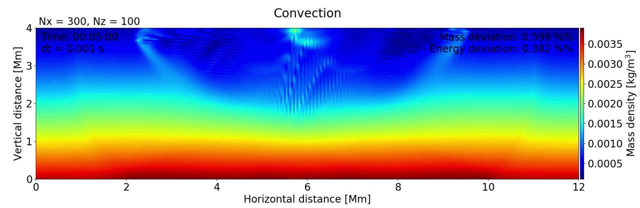

FIG. 5. Figure showing snapshot of the mass density of the convection simulation, at Time: 00:04:00.

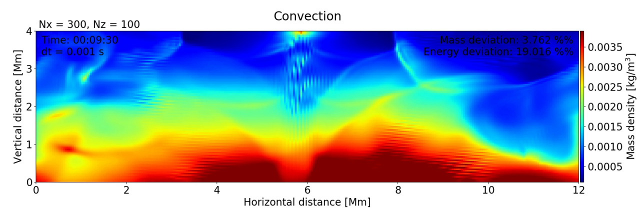

FIG. 6. Figure showing snapshot of the mass density of the convection simulation, at Time: 00:09:30.

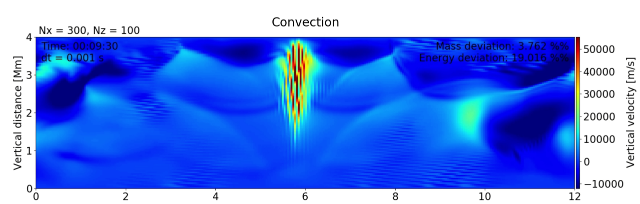

FIG. 7. Figure showing snapshot of the vertical velocity of the convection simulation, at Time: 00:09:30.

The third simulation we call the explosion simulation, in which the same perturbation is added as for the convection simulation, but the pressure is also updated. This results motion that resembles that of an explosion, which is why that's what it's called. Snapshots from this simulation showing the system's temperature and velocity vectors is shown in figure 8 and 9. Figure 10 and 11 show snapshots of the system's mass density.

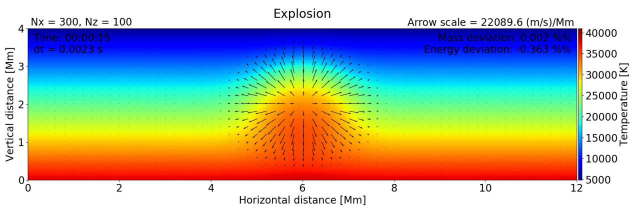

FIG. 8. Figure showing a snapshot of the explosion simulation at Time: 00:00:15, with temperature shown in colour and velocity as vectors

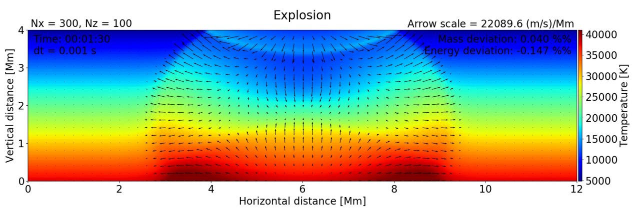

FIG. 9. Figure showing a snapshot of the explosion simulation at Time: 00:01:30, with temperature shown in colour and velocity as vectors

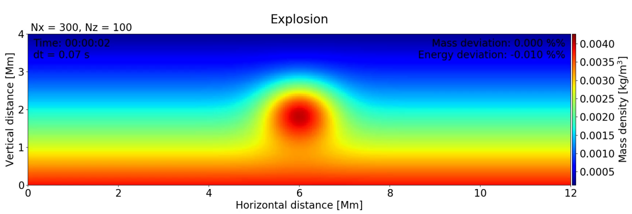

FIG. 10. Figure showing a snapshot of the mass density at the start of the explosion simulation

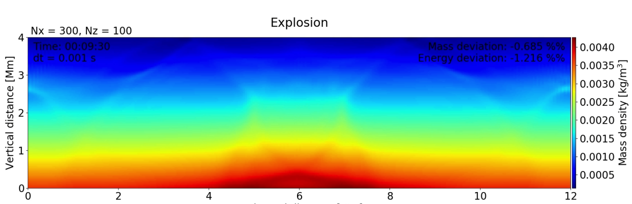

FIG. 11. Figure showing a snapshot of the mass density of the explosion simulation at Time: 00:09:30.

DISCUSSION

In figure 1 we have a snapshot of the gas in hydrostatic equilibrium, showing that after half a minute there is no noticeable movement. This is as expected, since the pressure gradient balances out the external force, i.e. gravity. We do see some deviations in the mass and energy, as displayed on the right hand side of the figure. Small deviations are however to be expected from numerical errors, and the fact that they are very small here suggests that our code is working. For our convection simulation we can see from figure 2 that the gas starts out static, with a temperature perturbation in the middle. In figure 3 we have another snapshot taken one and a half minute later, where we see that the perturbation has caused movement in the gas, creating what looks like two vortexes in the wake of the rising bubble of hot gas (i.e. the perturbation). The bubble likely rises because the mass density in the perturbed area is smaller than outside it, as we see in figure 4, which makes the denser gas around the bubble push it upwards. The reason the mass density is lower comes from our assumption of ideal gas, which means that the density is inversely proportional to temperature. Since the pressure doesn't change the mass density must therefore be smaller than for the gas around it, which is in hydrostatic equilibrium. A few minutes after this we start getting more complex and detailed motion of the gas when the bubble seems to hit the upper vertical boundary. We can see thin, straight streaks of alternating high and low mass densities, shown in figure 5. These streaks are probably representing convection on smaller scales, and them being so straight is a consequence of the limited resolution of our simulation, since it seems pretty unphysical for a gas/fluid to move in this way. This seems supported by the fact that the streaks are the width of one data point / cell. Figure 5 also shows that the bubble hitting the "roof" of our simulation seems to cause pressure waves along the roof, outwards from the middle of the simulation box. The gas rising into the vertical boundary seems to cause high densities in the upper middle of the box, which pushes the gas out to the side, causing the pressure waves. As the process continues the system gets seemingly more and more chaotic, as we can see in figure 6 which is showing a snapshot of the mass density nearing the ten minute mark, i.e. the end of our simulation. Figure 7 shows the state of the convection simulation near the end in a clearer light; the gas seems to be rising in the middle of the box and sinking near the edges. Convection is still occurring, it is just happening in a more chaotic fashion than it was in figure 3. In any case, the system doesn't seem to want to return to the hydrostatic state, at least not in time spans short enough for us to simulate in this project. In our explosion simulation we have added a temperature perturbation equal to that of the convection simulation, only we have now updated the pressure after adding it. The result is that the gas in the perturbed area is swiftly expelled from the centre of the perturbation and generates a shock wave, as can be seen in figures 8 and 9. This is pretty different from our convection simulation, and comes from the fact that the pressure is no longer in hydrostatic equilibrium, meaning that the pressure gradient in the momentum equation (2) quickly sets the system into motion. In figure 10 we see that the mass density in the perturbed area is now higher than around it, which comes from the fact that for an ideal gas the density is proportional to pressure. It is also inversely proportional to temperature as discussed earlier, but due to our relation between pressure and temperature in equation (16) the contribution from the pressure outweighs that from the temperature. Though this simulation started off more active than the convection simulation, the pressure waves from the initial "explosion" just seems to bounce around the box for a while until the system almost seems to settle down, as seen in 11. This is in pretty stark contrast to the convection simulation which seemingly just gets more chaotic over time. Maybe this is because convective processes are the result of a gas being unstable and are as a consequence more chaotic, while the pressure waves in the explosion simulation are a lot more orderly as they just symmetrically bounce around the box. The mass- and energy deviations near the end of the convection simulation, as shown in figure 6 are certainly a lot higher than the mass- and energy deviations for the explosion simulation, shown in figure 11. Particularly the energy, which deviates by almost twenty percent for the convection simulation, while it is around one percent for the explosion simulation. The deviations are to be expected from numerical errors, but the fact that it is so much higher in one simulation might point to the fact that there is more complex motion going on.

CONCLUSION

We have modelled stellar gas in hydrostatic equilibrium and observed that by adding temperature perturbations we can get movements in the gas that resemble convection, as well as pressure waves resulting from an explosion. Most of the motions of the gas seem physical, but there are also some structures in our simulation that might result from limited resolution.

SUMMARY

Doing this project I have learned how to perform fluid simulations with python. I have also gotten better at vectorizing my code and dealing with 2D matrices in python. Most of the trouble I had on this project is related to the fact that the simulations take quite long to run. To begin with I didn't have a lower limit on how small one timestep could get, which resulted in a simulation I left on overnight taking over five hours. Even after setting a lower limit to the timesteps, the simulations can still take hours to complete. Finding out I forgot to save a variable that I wanted to visualise meant having to do a few hours long simulation again.

REFERENCES

1 Gudiksen, B.V., 2020, Lecture Note *AST3310: Astrophysical plasma and stellar interiors* in the course AST3310