Simulating the Convective Zone of a Sun-like Star

Kandidatnr: 18

May 2023

Introduction

In this paper, we will simulate a segment of the convection zone in a star that is in hydrostatic equilibrium with the x direction being horizontal and y being in the radial direction, so one can use y and r interchangeably. Stars are of course made of gas, incredibly dense gas at that, which we can assume to be ideal. At the scale we will be looking at for this simulation, the mean free path of particles in the gas will be so minuscule that it will be safe to assume that we can treat this dense gas as a liquid. In order to accomplish our simulation, we need three absolutely essential equations in hydrodynamics. The first is the continuity equation with no energy sources or sinks (for example, fusions and radiation respectively), which is as follows:

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

Where $\mathbf{u} = (u, w)^2$ is the flow or the movement in the x and y direction and ρ is the mass density. We also need the momentum equation for both the x and y direction, excluding the viscous stress tensor:

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

Where $\otimes$ is the outer product, P is the pressure, and $\mathbf{g}$ is the gravity vector. Lastly, we need the energy equation:

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

Where e is the internal energy.

Although these equations are quite compact, we can extract a good deal of information from them, and they alone are what we need for our simulation. Once we have our simulation working in hydrostatic equilibrium, we will add a bubble of hot gas in the form of a 2D Gaussian.

Method

In order to start the algorithm we must first solve for initial conditions, starting with temperature T. We assume hydrostatic equilibrium, meaning that the pressure and the gravitational force are balancing each other out. The radial part of the force, or in our case, the force in the y-direction then becomes:

$$\frac{\partial P}{\partial y} = -\frac{Gm\rho}{y^2} \tag{4}$$

We also have use for the logarithmic temperature gradient:

$$\nabla = \frac{\partial lnT}{\partial lnP} = \frac{P}{T}\frac{\partial T}{\partial P} \quad , \quad \frac{\partial T}{\partial P} = \frac{T}{P}\nabla$$

Solving for the temperature gradient with respect to y:

$$\frac{\partial T}{\partial y} = \frac{\partial T}{\partial P}\frac{\partial P}{\partial y} = -\frac{T}{P}\nabla\frac{Gm\rho}{y^2}$$

Using the ideal gas law, P can be defined as:

$$P = \frac{\rho}{\mu m_u}k_bT$$

Which we then can set in:

$$\frac{\partial T}{\partial y} = -\frac{T}{\frac{\rho}{\mu m_u}k_bT}\nabla\frac{Gm\rho}{y^2} \quad \rightarrow \quad \partial T = -\nabla\frac{\mu m_u Gm}{k_b}\frac{1}{y^2}\partial y$$

We now integrate both sides:

$$\int \partial T = -\nabla\frac{\mu m_u Gm}{k_b}\int\frac{1}{y^2}\partial y \quad \rightarrow \quad T - T_0 = \nabla\frac{\mu m_u Gm}{k_b y}$$

Using $g = \frac{Gm}{r^2}$, we multiply and divide the right side by y and move T_0 to the right side as well:

$$T = \nabla\frac{\mu m_u Gm}{k_b y}\frac{y}{y} + T_0 = \nabla\frac{\mu m_u g}{k_b}y + T_0 \tag{5}$$

We know need an expression for P, which we can find in a similar manner to how we found T:

$$\nabla = \frac{\partial lnT}{\partial lnP} \quad \rightarrow \quad \partial lnP = \frac{1}{\nabla}\partial lnT \quad \rightarrow \quad \int lnP = \frac{1}{\nabla}\int \partial lnT$$

$$\rightarrow lnP - lnP_0 = \frac{1}{\nabla}(lnT - lnT_0) \quad \rightarrow \quad lnP = \frac{1}{\nabla}ln(\frac{T}{T_0}) + lnP_0$$

$$\rightarrow \quad e^{lnP} = e^{\frac{1}{\nabla}ln(\frac{T}{T_0})} + e^{lnP_0}$$

$$P = P_0 \left(\frac{T}{T_0}\right)^{1/\nabla} \tag{6}$$

In order to calculate the next iteration, we first need to "massage" our formulas into their 2D form. Starting from the continuity equation (1) we get:

$$\frac{\partial\rho}{\partial t} = -\nabla\cdot(\rho\mathbf{u}) = -\left(\frac{\partial}{\partial x} + \frac{\partial}{\partial y}\right)\cdot(\rho u, \rho w) = -\frac{\partial\rho u}{\partial x} - \frac{\partial\rho w}{\partial y} \tag{7}$$

We split the momentum equation (2) into its respective x and y parts:

$$\frac{\partial\rho\mathbf{u}}{\partial t} + \nabla\cdot\begin{bmatrix}\rho u^2 & \rho uw\\ \rho wu & \rho w^2\end{bmatrix} = -\nabla P + \rho\mathbf{g}$$

$$\rightarrow \frac{\partial\rho\mathbf{u}}{\partial t} + \begin{bmatrix}\left(\frac{\partial\rho u^2}{\partial x} + \frac{\partial\rho uw}{\partial y}\right)\\ \left(\frac{\partial\rho wu}{\partial x} + \frac{\partial\rho w^2}{\partial y}\right)\end{bmatrix} = \left(-\frac{\partial P}{\partial x}, -\frac{\partial P}{\partial y}\right) + \rho\mathbf{g}$$

Which can then be split up into both x, and y components, starting with x:

$$\frac{\partial\rho u}{\partial t} + \frac{\partial\rho u^2}{\partial x} + \frac{\partial\rho uw}{\partial y} = -\frac{\partial P}{\partial x} - \rho g_x \quad , \quad g_x = 0$$

$$\rightarrow \quad \frac{\partial\rho u}{\partial t} = -\frac{\partial\rho u^2}{\partial x} - \frac{\partial\rho uw}{\partial y} - \frac{\partial P}{\partial x} \tag{8}$$

And y:

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

$$\rightarrow \frac{\partial\rho w}{\partial t} = -\frac{\partial\rho wu}{\partial x} - \frac{\partial\rho w^2}{\partial y} - \frac{\partial P}{\partial y} - \rho g_y \tag{9}$$

Lastly the energy equation (3) can be expressed as:

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

$$\rightarrow \quad \frac{\partial e}{\partial t} = -\left(\frac{\partial eu}{\partial x} + \frac{\partial ew}{\partial y}\right) - P\left(\frac{\partial u}{\partial x} + \frac{\partial w}{\partial y}\right) \tag{10}$$

Now in order to implement these equations into our script, we must first discretize and change them into an algorithmic format. We must first find the vertical component of the momentum equation. We can start with equation (9):

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

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

It then follows that equation (8) can be expressed as:

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

So, the value in the next iteration for both u and w respectively becomes:

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

Using the energy equation (10), we have:

$$e^{n+1}_{i,j}=e^n_{i,j}+\frac{\partial e}{\partial t}\Delta t=e^n_{i,j}+\left[-\left[\frac{\partial eu}{\partial x}\right]^n_{i,j}-\left[\frac{\partial ew}{\partial y}\right]^n_{i,j}-P\left(\left[\frac{\partial u}{\partial x}\right]^n_{i,j}+\left[\frac{\partial w}{\partial y}\right]^n_{i,j}\right)\right]\Delta t$$

Where we use the product rule on both on both $\left[\frac{\partial eu}{\partial x}\right]^n_{i,j}$ and $\left[\frac{\partial ew}{\partial y}\right]^n_{i,j}$:

$$\left[\frac{\partial eu}{\partial x}\right]^n_{i,j}=u\frac{\partial e}{\partial x}+e\frac{\partial u}{\partial x} \quad , \quad \left[\frac{\partial ew}{\partial y}\right]^n_{i,j}=w\frac{\partial e}{\partial y}+e\frac{\partial w}{\partial y}$$

We then set those in and get the following:

$$\begin{aligned}e^{n+1}_{i,j}=e^n_{i,j}+&\left[-\left(u\left[\frac{\partial e}{\partial x}\right]^n_{i,j}+e\left[\frac{\partial u}{\partial x}\right]^n_{i,j}\right)\right.\\ &\left.-\left(w\left[\frac{\partial e}{\partial y}\right]^n_{i,j}+e\left[\frac{\partial w}{\partial y}\right]^n_{i,j}\right)-P\left(\left[\frac{\partial u}{\partial x}\right]^n_{i,j}+\left[\frac{\partial w}{\partial y}\right]^n_{i,j}\right)\right]\Delta t\end{aligned}$$

We still need Δt in order to calculate the next iteration, which we find using dynamic step size. For each primary variable ϕ(excluding u and w since the derivative is often 0), and for x and y we calculate the relative change at each grid point. We find the maximum change for each variable and pick the maximum out of those. We do this to make sure that we don't change by too big of a percentage for a single variable. We then divide this largest relative change δ by a small number p.

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

Now that we have the next iteration for our primary variables, we need to find T and P based on those values. The internal energy of an ideal gas can be expressed as:

$$e = \frac{1}{(\gamma - 1)} \frac{\rho}{\mu m_u} k_B T = \frac{P}{(\gamma - 1)} \tag{11}$$

Where $\gamma = 5/3$ for an ideal gas. Using this and our newly calculated $e_{i,j}^{n+1}$, we can calculate $P_{i,j}^{n+1}$:

$$P_{i,j}^{n+1} = \frac{2}{3} e_{i,j}^{n+1}$$

Using that, $\rho_{i,j}^{n+1}$, and ideal gas, we can calculate $T_{i,j}^{n+1}$:

$$T_{i,j}^{n+1} = \frac{P_{i,j}^{n+1}}{\rho_{i,j}^{n+1}} \frac{\mu m_u}{k_B}$$

In order to keep our simulation running correctly, we had to set horizontal and vertical boundary conditions. We used the 3-point forward and backward difference approximation:

$$\left[\frac{\partial \phi}{\partial y}\right]_{i,0}^{n} = \frac{-\phi_{i,2}^{n} + 4\phi_{i,1}^{n} - 3\phi_{i,0}^{n}}{2\Delta y} \quad , \quad \left[\frac{\partial \phi}{\partial y}\right]_{i,N_y-1}^{n} = \frac{3\phi_{i,N_y-1}^{n} - 4\phi_{i,N_y-2}^{n} + \phi_{i,N_y-3}^{n}}{2\Delta y}$$

We are given the fact that the vertical gradient of the u should be 0 at the boundaries and therefore:

$$0 = \frac{-\phi_{i,2}^{n} + 4\phi_{i,1}^{n} - 3\phi_{i,0}^{n}}{2\Delta y} \quad , \quad 0 = \frac{3\phi_{i,N_y-1}^{n} - 4\phi_{i,N_y-2}^{n} + \phi_{i,N_y-3}^{n}}{2\Delta y}$$

After some simple algebra, we get:

$$u_{i,0}^{n} = \frac{4u_{i,1}^{n} - u_{i,2}^{n}}{3} \quad , \quad u_{i,N_y-1}^{n} = \frac{4u_{i,N_y-2}^{n} - u_{i,N_y-3}^{n}}{3}$$

We find the boundary conditions for energy very similarly but first, we need an expression for the vertical gradient. Using equation (11), along with our condition of hydrostatic equilibrium (4), we have:

$$\frac{\partial e}{\partial y} = \frac{3}{2} \frac{\partial P}{\partial y} = -\frac{3}{2} g\rho$$

Then setting the middle expression for the internal energy into ρ, we get:

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

We then set that into the 3-point expression and get:

$$\left[\frac{\mu m_u g}{k_B}\frac{1}{T_{i,0}^n} + \frac{3}{2\Delta y}\right] e_{i,0}^n = \frac{-\phi_{i,2}^n + 4\phi_{i,1}^n}{2\Delta y}$$

$$\rightarrow \quad e_{i,0}^n = \frac{\left(-\phi_{i,2}^n + 4\phi_{i,1}^n\right)}{\left(3 + \frac{\mu m_u g}{k_B}\frac{2\Delta y}{T_{i,0}^n}\right)} \quad , \quad e_{i,N_y-1}^n = \frac{\left(-\phi_{i,N_y-3}^n + 4\phi_{i,N_y-2}^n\right)}{\left(3 - \frac{\mu m_u g}{k_B}\frac{2\Delta y}{T_{i,N_y-1}^n}\right)}$$

Once again, we can use the ideal gas law and internal energy of an ideal gas to find the boundary conditions for ρ:

$$\rho_{i,j}^n = \frac{2}{3}\frac{\mu m_u}{k_B}\frac{e_{i,j}^n}{T_{i,j}^n}$$

Lastly, to add our Gaussian perturbations p_g into the system, we use the formula for a 2D Gaussian:

$$p_g = e^{\frac{-((x-\mu_x)^2+(y-\mu_y)^2)}{2\sigma^2}}$$

Where μ_x and μ_y are the means in each dimension and σ being the standard deviation. The Gaussian is added to our temperature, and since everything is dependent on temperature in some way, it affects the whole system.

We then began writing our code with all the analytical/numerical expressions we needed to achieve hydrostatic equilibrium in our star. We used a class-based script to simulate the $12 \times 4Mm$ box. We have a method for initializing all of our arrays and adding our perturbations when we want. When adding perturbations, we set our μs so that the bubble of hot gas starts in the middle of the x-axis and just under the middle of the y-axis so that we have some time to see how it develops. Both the μ values and σ were found from trial and error. We created a method *step*() which calculates all of the gradients for the four primary variables, our dt, and with those the next iteration of variables. Since both P and T both rely on the primary variables, we apply $boundary_conditions()$ before the calculation of these two in order to avoid any numerical errors, then with the corrected borders, we calculate P and T. We then use a visualization script to run this *step*() method for N frames, plot, and animate.

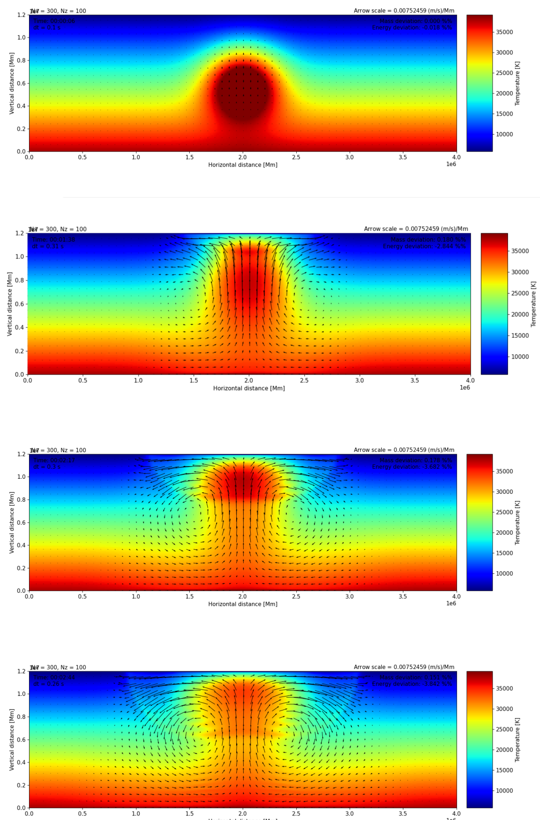

Figure 1: Snapshots from an animation of a 2D Gaussian perturbation moving upwards and dispersing on the surface.

Results

When it comes to adding the perturbations, we experimented with several different sigma values until we found one that looked good for us, which ended up being $0.75Mm$. In order to be able to detect the perturbation, we had to increase its amplitude significantly, which again took a bit of trial and error, but we ended up increasing it with a factor of $25,000$. With the perturbations added, they behave perfectly as expected (see Figure 1). The x-axis in Figure 1 goes from 0 to $12Mm$. The y-axis is technically flipped, it goes from 0 to $4Mm$ with the cooler surface being on the top.

Discussion

Unfortunately, as in all simulations, we couldn't make a perfect simulation based on our assumptions and did end up losing some energy in the simulation (a maximum of around 3.8%). Something we found to be very interesting was that as time progresses, the energy deviation increases until a point when it turns around, meaning that the system is losing some energy, and then as soon as the bubble disperses on the top of the box, the energy deviation starts decreasing. So the act of dispersion is somehow more "cost-efficient" in a way.

This project was quite particular when it came to achieving hydrostatic equilibrium. We struggled for a substantial amount of time with our perturbations moving in the wrong direction, only to find out that it was due to the fact that we had our y-array facing the wrong direction, and therefore the temperature was as well. Calculating the analytical expressions was never a problem, implementing them, however, seemed to be much more problematic. Throughout the process, there was always a combination of many small things that made it so that our simulation didn't work as wished. In the beginning, we were wondering why our simulation was moving from left to right on the screen. Little did we know that the *np.roll*() function we were using, which shifts an array by 1 index in the given direction, was affecting our program like that.

Conclusion

In this paper, we have created a 2D simulation of a closed $12 \times 4Mm$ box inside the convective layer of a sun-like star using a class-based Python script. We did this by looking at the star at such a large scale, allowing us to treat the gas as a liquid and use 3 main hydrodynamical equations. At the same time, we also assumed an ideal gas, as we physicists tend to do. From those 3 equations, we calculated the gradients for our primary variables (u, w, u, ρ), calculated Δt using dynamic step size, and with those, calculated the next iteration. This project was quite challenging, but most of the time, the best results come with just a splash of suffering. This was one of those cases.

Resources

[1] Gudiksen, B.V. “AST3310: Astrophysical Plasma and Stellar Interiors.” Institute of Theoretical Astrophysics, University of Oslo, 17 Nov. 2022.