Astrophysical Plasma and Stellar Interiors

OE3

Institute of Theoretical Astrophysics

Contents

Introduction

Computer modelling is a way we can better understand the world around us. Specifically, in science, it has helped us make great strides, specially in those branches where analytical solutions are hard to find, or even impossible, such as the Navier-Stokes equations. This is why computer simulations are relevant in astrophysics, as they can shed light on different phenomena which would be impossible to treat otherwise. In this project, we try to better understand how convection works within a star using a computer simulation of the hydrodynamics inside a cross section of the Sun. To do so, we will take the momentum, continuity, and energy equations and use them to simulate the ecosystem that governs the sun's convective motions.

1 Exercise 1: Proving equations for the simulation

1.1 Continuity equation

Let's start by taking the continuity equation as stated in the lecture notes,

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

To be consistent with the notation used in the project description, we use $\mathbf{u} = (u, w)$ for the velocity and, using $\nabla \cdot \mathbf{F} = \frac{\partial F_x}{\partial x} + \frac{\partial F_y}{\partial y}$ and rearranging, we get equation (14) from the project description:

$$\frac{\partial \rho}{\partial t} + \frac{\partial (\rho u)}{\partial x} + \frac{\partial (\rho w)}{\partial y} = 0 \Rightarrow \frac{\partial \rho}{\partial t} = -\frac{\partial (\rho u)}{\partial x} - \frac{\partial (\rho w)}{\partial y}. \tag{2}$$

1.2 Momentum equations

Now, we will start with the general momentum equation

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

Decomposing the exterior product $u \otimes u$ in the matrix representation of a 2-tensor,

$$\mathbf{u} \otimes \mathbf{u} = \begin{pmatrix} u \cdot u & u \cdot w \\ w \cdot u & w \cdot w \end{pmatrix} = \begin{pmatrix} u^2 & uw \\ wu & w^2 \end{pmatrix}. \tag{4}$$

And applying the ∇ operator to this matrix, we obtain

$$\nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u}) \hat{=} \begin{pmatrix} \frac{\partial}{\partial x} & \frac{\partial}{\partial y} \end{pmatrix} \rho \begin{pmatrix} u^2 & uw \\ wu & w^2 \end{pmatrix} = \begin{pmatrix} \frac{\partial (\rho u^2)}{\partial x} + \frac{\partial (\rho u w)}{\partial y} \\ \frac{\partial (\rho w u)}{\partial x} + \frac{\partial (\rho w^2)}{\partial y} \end{pmatrix}. \tag{5}$$

With this in mind, equation (3) can be rewritten as

$$\frac{\partial \rho \mathbf{u}}{\partial t} + \begin{pmatrix} \frac{\partial (\rho u^2)}{\partial x} + \frac{\partial (\rho u w)}{\partial y} \\ \frac{\partial (\rho w u)}{\partial x} + \frac{\partial (\rho w^2)}{\partial y} \end{pmatrix} = -\nabla P + \rho \mathbf{g}. \tag{6}$$

Form this point we can separate our equation in the x and y components of the momentum, using the first and second row of our matrix for the horizontal and vertical components respectively. For the equation corresponding to the x coordinate, we have:

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

Rewriting $\nabla P = \frac{\partial P}{\partial x}$ and using the approximation that the x-component of the gravity $\mathbf{g} = (g_x, g_y)$ is $g_x = 0$ because tidal forces can be neglected (more on this later) we get:

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

which is precisely the horizontal momentum equation. For the equation on the other axis we sub $\nabla P = \frac{\partial P}{\partial y}$ and use the fact that the gravity g_y in this direction is nonzero, that is

$$\frac{\partial \rho w}{\partial t} + \frac{\partial(\rho w^2)}{\partial y} + \frac{\partial(\rho u w)}{\partial x} = -\frac{\partial P}{\partial y} + \rho g_y. \tag{9}$$

Rearranging, we obtain the verical momentum equation:

$$\frac{\partial \rho w}{\partial t} = -\frac{\partial(\rho w^2)}{\partial y} - \frac{\partial(\rho u w)}{\partial x} - \frac{\partial P}{\partial y} + \rho g_y. \tag{10}$$

1.3 Energy equation

Starting form the energy equation

$$\frac{\partial e}{\partial t} + \nabla \cdot (e\mathbf{u}) = -P\nabla \cdot \mathbf{u}, \tag{11}$$

expanding the terms involving the ∇ operator terms

$$\nabla \cdot (e\mathbf{u}) = \frac{\partial(eu)}{\partial x} + \frac{\partial(ew)}{\partial y} \quad \text{and} \quad \nabla \cdot \mathbf{u} = \frac{\partial u}{\partial x} + \frac{\partial w}{\partial y};$$

and including them in the energy equation, we arrive at

$$\frac{\partial e}{\partial t} + \frac{\partial(eu)}{\partial x} + \frac{\partial(ew)}{\partial y} = -P\left(\frac{\partial u}{\partial x} + \frac{\partial w}{\partial y}\right). \tag{12}$$

Performing a basic rearrangement, we arrive at the equation as presented in the project description

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

1.4 Why is gravity considered only in the y direction?

We are simulating solar convection spanning a region of only a few megameters (Mm) beneath the photosphere, it is reasonable to approximate the gravitational acceleration as constant and equal to $g = GM_\odot/R_\odot^2$. This approximation holds because the vertical extent of the simulation domain is much smaller than the solar radius $R_\odot \approx 696{,}340\,\mathrm{km} = 696.34\,\mathrm{Mm}$. Over a few Mm (i.e., a few thousand km), the relative change in radial distance from the solar centre is less than 1%, leading to negligible variation in the gravitational field. In other words, at that height with such an small spanning of the region in the x compared to the radius, the tidal forces are negligible. Thus, treating gravity as uniform and equal to its value at the solar surface introduces minimal error while simplifying the numerical treatment of the system.

2 Exercise 2: Deriving numerical algorithms for w and e

In order to construct the algorithms for both quantities $w_{i,j}^{n+1}$ and $e_{i,j}^{n+1}$, we will follow the same procedure as explained in the project description. This consists of writing the time derivatives of the left-hand side of the equation governing each quantity as forward differencing, and the spatial derivatives in the right hand side as a mixture of central differences and/or upwind derivatives. Even though the procedure is really similar, we have to outline it as the derived formulas are central to the simulation.

2.1 Vertical momentum equation $w_{i,j}^{n+1}$

The objective is to derive a numerical algorithm that relates $w_{i,j}^{n}$ to $w_{i,j}^{n+1}$. This is done by first writing equation (10), that is,

$$\frac{\partial(\rho w)}{\partial t} = -\frac{(\partial \rho w^2)}{\partial y} - \frac{\partial(\rho u \omega)}{\partial x} - \frac{\partial P}{\partial y} + \rho g_y, \tag{14}$$

and expand the first two terms of the right side $\frac{\partial(\rho w^2)}{\partial y}$ and $\frac{\partial(\rho u w)}{\partial x}$:

$$\frac{\partial(\rho w^2)}{\partial y} = \rho w \frac{\partial(w)}{\partial y} + w \frac{\partial(\rho w)}{\partial y}, \qquad \frac{\partial(\rho u w)}{\partial x} = \rho w \frac{\partial(u)}{\partial x} + u \frac{\partial(\rho w)}{\partial x}. \tag{15}$$

Using this, we can rewrite the equation (10) as

$$\frac{\partial(\rho w)}{\partial t} = -\left(\rho w \frac{\partial(w)}{\partial y} + w \frac{\partial(\rho w)}{\partial y}\right) - \left(\rho w \frac{\partial(u)}{\partial x} + u \frac{\partial(\rho w)}{\partial x}\right) - \frac{\partial P}{\partial y} + \rho g_y \tag{16}$$

$$= -\rho w \left(\frac{\partial(w)}{\partial y} + \frac{\partial(u)}{\partial x}\right) - w \frac{\partial(\rho w)}{\partial y} - u \frac{\partial(\rho w)}{\partial x} - \frac{\partial P}{\partial y} + \rho g_y \tag{17}$$

Now, we can discretize this equation in a set of points indexed by i and j, in the same way as was done in the project description,

$$\begin{aligned} \left[\frac{\partial(\rho w)}{\partial t}\right]_{i,j}^n = &-[\rho w]_{i,j}^n \left(\left[\frac{\partial w}{\partial y}\right]_{i,j}^n - \left[\frac{\partial u}{\partial x}\right]_{i,j}^n\right) \\ &- [w]_{i,j}^n \left[\frac{\partial(\rho w)}{\partial y}\right]_{i,j}^n - [u]_{i,j}^n \left[\frac{\partial(\rho w)}{\partial x}\right]_{i,j}^n - \left[\frac{\partial P}{\partial y}\right]_{i,j}^n + [\rho g_y]_{i,j}^n . \end{aligned} \tag{18}$$

This is out final analytical form of the discretized vertical momentum equation. The temporal derivative of the left hand side of (17) can be expressed as forward differencing, which will be used to advance the simulation.

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

Solving for $[\rho w]_{i,j}^{n+1}$ and dividing by $\rho_{i,j}^{n+1}$, which at this point in the algorithm we have already calculated, we get

$$w_{i,j}^{n+1} = \frac{1}{\rho_{i,j}^{n+1}}\left((\rho w)_{i,j}^{n} + \Delta t\left[\frac{\partial(\rho w)}{\partial t}\right]_{i,j}^{n}\right), \tag{20}$$

where we have to use equation (18) for the time derivative. In equation (18), each spatial derivative is discretized in the following way:

$$\left[\frac{\partial w}{\partial y}\right]_{i,j}^{n} \approx \begin{cases} \dfrac{w_{i,j}^{n} - w_{i-1,j}^{n}}{2\Delta y} & \text{if } w_{i,j} \geq 0, \\ \dfrac{w_{i+1,j}^{n} - w_{i,j}^{n}}{2\Delta y} & \text{if } w_{i,j} < 0, \end{cases} \tag{21}$$

$$\left[\frac{\partial u}{\partial x}\right]_{i,j}^{n} \approx \frac{u_{i,j+1}^{n} - u_{i,j-1}^{n}}{2\Delta x}, \tag{22}$$

$$\left[\frac{\partial(\rho w)}{\partial y}\right]_{i,j}^{n} \approx \begin{cases} \dfrac{(\rho w)_{i,j} - (\rho w)_{i,j-1}}{\Delta y} & \text{if } w_{i,j} \geq 0, \\ \dfrac{(\rho w)_{i,j+1} - (\rho w)_{i,j}}{\Delta y} & \text{if } w_{i,j} < 0, \end{cases} \tag{23}$$

$$\left[\frac{\partial(\rho w)}{\partial x}\right]_{i,j}^{n} \approx \begin{cases} \dfrac{(\rho w)_{i,j} - (\rho w)_{i-1,j}}{\Delta x} & \text{if } u_{i,j} \geq 0, \\ \dfrac{(\rho w)_{i+1,j} - (\rho w)_{i,j}}{\Delta x} & \text{if } u_{i,j} < 0, \end{cases} \tag{24}$$

$$\left[\frac{\partial P}{\partial y}\right]_{i,j}^{n} \approx \frac{P_{i,j+1}^{n} - P_{i,j-1}^{n}}{2\Delta y}. \tag{25}$$

Note now that the convective terms are computed using the upwind differencing scheme, but now, the derivative $\left[\frac{\partial w}{\partial y}\right]_{i,j}^{n}$ is upwind and $\left[\frac{\partial u}{\partial x}\right]_{i,j}^{n}$ is central.

2.2 Internal energy $e_{i,j}^{n+1}$

Now, the objective is to derive a numerical algorithm that relates $e_{i,j}^{n}$ to $e_{i,j}^{n+1}$. This is done by first writing the equation (26) for e:

$$\frac{\partial e}{\partial t} = -\frac{\partial(eu)}{\partial x} - \frac{\partial(ew)}{\partial y} - P\left(\frac{\partial u}{\partial x} + \frac{\partial w}{\partial y}\right). \tag{26}$$

We expand the first two terms of the right side, $\frac{\partial(eu)}{\partial x}$ and $\frac{\partial(ew)}{\partial y}$:

$$\frac{\partial(eu)}{\partial x} = u\frac{\partial e}{\partial x} + e\frac{\partial u}{\partial x}, \qquad \frac{\partial(ew)}{\partial y} = w\frac{\partial e}{\partial y} + e\frac{\partial w}{\partial y}. \tag{27}$$

Using this, we can rewrite equation (26) as:

$$\frac{\partial e}{\partial t} = -\left(u\frac{\partial e}{\partial x} + e\frac{\partial u}{\partial x}\right) - \left(w\frac{\partial e}{\partial y} + e\frac{\partial w}{\partial y}\right) - P\left(\frac{\partial u}{\partial x} + \frac{\partial w}{\partial y}\right) \tag{28}$$

$$= -\left(u\frac{\partial e}{\partial x} + w\frac{\partial e}{\partial y}\right) - e\left(\frac{\partial u}{\partial x} + \frac{\partial w}{\partial y}\right) - P\left(\frac{\partial u}{\partial x} + \frac{\partial w}{\partial y}\right) \tag{29}$$

$$= -\left(u\frac{\partial e}{\partial x} + w\frac{\partial e}{\partial y}\right) - (e+P)\left(\frac{\partial u}{\partial x} + \frac{\partial w}{\partial y}\right). \tag{30}$$

Now, we can discretize this equation in a set of points indexed by i (for the y-direction) and j (for the x-direction):

$$\left[\frac{\partial e}{\partial t}\right]^n_{i,j} = -[u]^n_{i,j}\left[\frac{\partial e}{\partial x}\right]^n_{i,j} - [w]^n_{i,j}\left[\frac{\partial e}{\partial y}\right]^n_{i,j} - \left([e]^n_{i,j} + [P]^n_{i,j}\right)\left(\left[\frac{\partial u}{\partial x}\right]^n_{i,j} + \left[\frac{\partial w}{\partial y}\right]^n_{i,j}\right). \tag{31}$$

This is our final analytical form of the discretized energy equation before specifying the derivative approximations. The temporal derivative on the left-hand side of (30) is expressed as forward differencing:

$$\left[\frac{\partial e}{\partial t}\right]^n_{i,j} = \frac{e^{n+1}_{i,j} - e^n_{i,j}}{\Delta t}. \tag{32}$$

Solving for $e^{n+1}_{i,j}$, we get:

$$e^{n+1}_{i,j} = e^n_{i,j} + \Delta t\left[\frac{\partial e}{\partial t}\right]^n_{i,j}, \tag{33}$$

where we use equation (31) for the time derivative term $\left[\frac{\partial e}{\partial t}\right]^n_{i,j}$. In equation (31), terms $u\frac{\partial e}{\partial x}$ and $w\frac{\partial e}{\partial y}$ use upwind differencing for the derivatives of e, while terms involving $\frac{\partial u}{\partial x}$ and $\frac{\partial w}{\partial y}$ in the $(e+P)$ factor use central differencing. The specific discretizations are:

$$\left[\frac{\partial e}{\partial x}\right]^n_{i,j} \approx \begin{cases} \dfrac{e^n_{i,j} - e^n_{i,j-1}}{\Delta x} & \text{if } u^n_{i,j} \geq 0, \\ \dfrac{e^n_{i,j+1} - e^n_{i,j}}{\Delta x} & \text{if } u^n_{i,j} < 0, \end{cases} \tag{34}$$

$$\left[\frac{\partial e}{\partial y}\right]^n_{i,j} \approx \begin{cases} \dfrac{e^n_{i,j} - e^n_{i-1,j}}{\Delta y} & \text{if } w^n_{i,j} \geq 0, \\ \dfrac{e^n_{i+1,j} - e^n_{i,j}}{\Delta y} & \text{if } w^n_{i,j} < 0, \end{cases} \tag{35}$$

$$\left[\frac{\partial u}{\partial x}\right]^n_{i,j} \approx \frac{u^n_{i,j+1} - u^n_{i,j-1}}{2\Delta x}, \tag{36}$$

$$\left[\frac{\partial w}{\partial y}\right]^n_{i,j} \approx \frac{w^n_{i+1,j} - w^n_{i-1,j}}{2\Delta y}. \tag{37}$$

2.3 Upwind method: when to use it?

with the convective terms such as $\frac{\partial(\rho u)}{\partial x}$, $\frac{\partial(\rho u)}{\partial y}$
and $\frac{\partial(ew)}{\partial x}$,$\frac{\partial(ew)}{\partial y}$ this is because the upwind method respects the physics of the convective transport, it makes sure that the discretization works with the natural laws. With this method the physical

information comes from the same direction than the flux, which makes a lot of sense. On the other hand if we would have used the central differences discretization method, it will use symmetrical (left and right) data, which is physically incorrect for convection violating the physical causality. So applying the Upwind method is the more convenient way to solve it.

3 Exercise 3 : Initial and boundary conditions.

3.1 Initial conditions

The problem we are trying to solve is a Cauchy problem, that is, we need to evolve the state of the system forward in time. We need then to obtain a set of initial values for all the principal variables u, w, e and ρ. To do so, we are going to suppose that the system starts in hydrostatic equilibrium and that the temperature in the top boundary y_{ph} are those of the photosphere, $T_{\text{ph}} = 5778.0\,\text{K}$ and $P_{\text{ph}} = 1.8 \times 10^4\,\text{Pa}$. We then find P and T in all our domain by imposing hydrostatic equilibrium and supposing that $\nabla = \frac{\partial \ln T}{\partial \ln P} = \text{const.}$, which implies

$$\nabla = \frac{\partial \ln T}{\partial P}\frac{\partial P}{\partial \ln P} = \frac{\partial \ln T}{\partial P}\left(\frac{\partial \ln P}{\partial P}\right)^{-1} = \frac{\partial \ln T}{\partial P}\left(\frac{1}{P}\right)^{-1} = \frac{\partial \ln T}{\partial P}P. \tag{38}$$

Now we can integrate both sides of the equation with respect to P:

$$\frac{\nabla}{P} = \frac{\partial \ln T}{\partial P} \Rightarrow \int_{P_0}^{P} \frac{\nabla}{P} dP = \int_{\ln T_0}^{\ln P} d\ln T \tag{39}$$

And computing those integrals we have

$$\nabla(\ln P - \ln P_0) = \ln T - lnT_0 \Rightarrow \nabla\Big(\ln\frac{P}{P_0}\Big) = \ln\frac{T}{T_0} \Rightarrow \Big(\frac{P}{P_0}\Big)^{\nabla} = \frac{T}{T_0} \Rightarrow T(P) = T_0\Big(\frac{P}{P_0}\Big)^{\nabla}. \tag{40}$$

Note that this expression is easily invertible to obtain $P(T)$. Now in order to obtain the specific identity with dependence on the height y, we suppose hydrostatic equilibrium,

$$\frac{dP}{dy} = -\rho g. \tag{41}$$

Substituting first the equation of state $\rho = \frac{\mu m_u}{k_B T}P$ and second the previously obtained $T(P)$ expression, we obtain

$$\frac{dP}{dy} = -\frac{P\mu m_u}{k_B T}g = -\frac{\mu m_u g}{k_B}\frac{P}{T} = -\frac{\mu m_u g}{k_B T_{ph}}\frac{P}{P_{ph}^{-\nabla}P^{\nabla}} = -\frac{\mu m_u g}{k_B T_{ph}}P_{Ph}^{\nabla}\cdot P^{1-\nabla} \tag{42}$$

Rewriting the equation and integrating along P

$$\int_{P_{ph}}^{P} P^{\nabla-1}dP = -\frac{\mu m_u g}{k_B T_{ph}}P_{Ph}^{\nabla}\int_{y_{ph}}^{y} dy \Rightarrow \frac{P^{V} - P_{ph}^{V}}{\nabla} = -\frac{\mu m_u g}{k_B T_{ph}}P_{Ph}^{\nabla}(y - y_{ph)}. \tag{43}$$

Rearranging, we achieve an expression for $P(y)$:

$$P^{\nabla} = P_{ph}^{\nabla}\Big(1 - \frac{\mu m_u g}{k_B T_{ph}}P_{Ph}^{\nabla}(y - y_{ph)}\Big) \Rightarrow P = P_{ph}\Big(1 - \frac{\mu m_u g}{k_B T_{ph}}P_{Ph}^{\nabla}(y - y_{ph)}\Big)^{1/\nabla}. \tag{44}$$

Finally, substituting expression (44) for $P(y)$ in the $T(y)$ relation, that is, expression (40), we have the initial temperature in terms of the depth from the photosphere,

$$T(y) = T_{ph}\Big(1 - \frac{\mu m_u g}{k_B T_{ph}} P_{Ph}^{\nabla}(y - y_{ph)}\Big). \tag{45}$$

Obtaining initial data for ρ and e is a matter of applying the equation of state,

$$\rho(y) = \frac{\mu m_u}{k_B T(y)} P(y), \qquad e(y) = \frac{P(y)}{(\gamma - 1)}. \tag{46}$$

Finally, since we are parting from a state of equilibrium, we suppose $\mathbf{u} = (u, w) = (0, 0)$ everywhere.

3.2 Vertical boundary conditions for u, ρ and e

Regarding the vertical boundary conditions for the velocity components, let's consider the 3-point forward and backwards difference approximation

$$\left[\frac{\partial \phi}{\partial y}\right]_{i,j}^n \approx \frac{-\phi_{i,j-2}^n + 4\phi_{i,j-1}^n - 3\phi_{i,j}^n}{2\Delta y}, \qquad \left[\frac{\partial \phi}{\partial y}\right]_{i,j}^n \approx \frac{3\phi_{i,j}^n - 4\phi_{i,j+1}^n + \phi_{i,j+2}^n}{2\Delta y}. \tag{47}$$

Supposing that the vertical gradient of the horizontal component of the velocity should be zero at each of the vertical boundaries and using (47), we obtain

$$0 = \left[\frac{\partial u}{\partial y}\right]_{i,N_y-1}^n \approx \frac{-u_{i,N_y-3}^n + 4u_{i,N_y-2}^n - 3u_{i,N_y-1}^n}{2\Delta y} \Rightarrow u_{i,N_y-1}^n = \frac{-u_{i,N_y-3}^n + 4u_{i,N_y-2}^n}{3} \tag{48}$$

$$0 = \left[\frac{\partial u}{\partial y}\right]_{i,0}^n \approx \frac{3u_{i,0}^n - 4u_{i,1}^n + u_{i,2}^n}{2\Delta y} \Rightarrow u_{i,0}^n = \frac{3u_{i,0}^n - 4u_{i,1}^n}{3}. \tag{49}$$

Obtaining the vertical boundary conditions for ρ and e is a bit trickier since the equations of state couple this two quantities and we need to ensure that hydrostatic equilibrium is fulfilled at the boundaries, that is $\frac{dP}{dy} = -\rho g = \rho|g|$. For these boundaries, it is more convenient to work with second order forward and backwards differencing. We will derive in detail the condition for the $j = 0$ boundary, but the $j = N_y - 1$ is analogous. For instance, consider

$$\left[\frac{dP}{dy}\right]_{i,0}^n \approx \frac{P_{i,1}^n - P_{i,0}^n}{\Delta y} = \frac{\rho_{i,1}^n + \rho_{i,0}^n}{2} g \Rightarrow P_{i,0}^n = P_{i,1}^n - \frac{\Delta y g}{2}(\rho_{i,1}^n + \rho_{i,0}^n). \tag{50}$$

Nore that we have used a suitable average $\bar{\rho}$ for the value of ρ between the two cells. Now, we can use the equation of state to relate $\rho_{i,0}^n = \frac{P_i^n{,0}\mu m_u}{k_B T_{i,0}}$, which enables us to write

$$P_{i,0}^n = P_{i,1}^n + \frac{\Delta y g}{2}\left(\rho_{i,1}^n - \frac{P_{i,0}^n \mu m_u}{k_B T_{i,0}}\right) \Rightarrow P_{i,0}^n = \frac{P_{i,1}^n + \Delta y g \rho_{i,1}^n/2}{1 - \Delta y g \mu m_u / 2k_B T_{i,0}}. \tag{51}$$

This provides us with an expression that *almost* relates values on the boundary with values on the interior, except for the $T_{i,0}^n$ term. To solve this, we impose $T_{i,0}^n = T_{i,1}^n$, that is, no temperature gradient exists on the boundary. With this assumption, we can write the final expression for $P_{i,0}^n$ as

$$P_{i,0}^n = \frac{P_{i,1}^n + \Delta y g \rho_{i,1}^n/2}{1 - \Delta y g \mu m_u / 2k_B T_{i,1}}, \quad \rho_{i,0}^n = \frac{P_{i,0}^n}{k_B T_{i,0}}, \qquad e_{i,0}^n = \frac{P_{i,0}^n}{(\gamma - 1)}. \tag{52}$$

Obtaining the boundary conditions for $e^n_{i,0}$ and $\rho^n_{i,0}$ was straightforward, since they are all related to the pressure via the equation of state. For the $j = N_y - 1$ boundary, it's the same procedure, but with different signs due to approximating the derivative the other way around, obtaining

$$T^n_{i,N_y-1} = T^n_{i,N_y-2}, \quad P^n_{i,N_y-1} = \frac{P^n_{i,N_y-2} - \Delta y g \rho^n_{i,N_y-2}/2}{1 + \Delta y g \mu m_u / 2k_B T_{i,N_y-2}},$$
$$\rho^n_{i,N_y-1} = \frac{P^n_{i,N_y-1}}{k_B T_{i,N_y-1}}, \quad e^n_{i,N_y-1} = \frac{P^n_{i,N_y-1}}{(\gamma - 1)}.$$

In any case, the order of the computation for the boundary conditions must be T first, then P and finally ρ and e in any order.

4 Exercise 4 : Explanation of the code

In this section we make an overview of the code and explain the different methods we employ. For more detail, please consult the commented source code.

4.1 Context and Domain

The main things that we need to know about where our project is located are:

- We are simulating a 2D section of a star. More specifically we are trying to model how hot gas moves inside a rectangular cross section of a star of size $12 \cdot 10^8$ meters x $4 \cdot 10^8$ meters.
- We discretize the domain in a grid of 300 cells (x-axis) $\times$ 100 cells (y-axis). If we divide the space between each cell we get $\frac{12 \cdot 10^6}{300} = 40 \cdot 10^3\,\text{m}$ and $\frac{4 \cdot 10^6}{100} = 40 \cdot 10^3\,\text{m}$ so each cell is $4 \cdot 10^4\,\text{m} \times 4 \cdot 10^4\,\text{m}$.
- We use $\nabla =$ const. between 2/5 and 10 and constants as provided in the project description and lecture notes. Recall that if $\nabla < 0.4$ convection will not happen due to high stability but on the other hand if $\nabla > 0.4$ the simulation we begin to be unstable allowing convection to happen.

4.2 Initial conditions

The initial conditions that kickstart our project are:

- For the top boundary, representing the photosphere, we use $T_{\text{ph}} = 5778.0\,\text{K}$ and $P_{\text{ph}} = 1.8 \cdot 10^4\,\text{Pa}$. We calculate the initial conditions as stated in §3.1.
- When the initial conditions are calculated, we can apply a Gaussian perturbation setting the `self.apply_perturbation` or `self.apply_isobaric_perturbation` flags to `True`. These two options will be explained in greater detail in §5. In any case, this perturbations are calculated over the hydrostatic background.
- The main difference between these two is that the isobaric calculates ρ in terms of the Gaussian-perturbed T, while in the other one ρ is kept constant and P is modified. We only present the results using the isobaric perturbation because the other way of perturbing the initial conditions yield a pressure perturbation that propagates outside as acoustic waves, generating a lot of noise that renders the simulation unstable.

4.3 Time evolution of the system

In order to evolve the system, we will apply the main equations that govern the system in their discretized form. Precisely, we use equations (19)-(20) and (22)-(24) of the project description and equations (31)-(37) and (37)-(31) from this same document. The only caveat is that we need to calculate $\rho_{i,j}^{n+1}$ before anything else, since the calculations for $u_{i,j}^{n+1}$ and $w_{i,j}^{n+1}$ depend on this result.

4.4 Boundary conditions

The boundary conditions function is called **once per time step, immediately after updating the main variables**. Boundaries are not applied during initialization as the initial state is designed to satisfy them. The boundary conditions are applied as explained in the project description and previous sections of the document.

4.5 Time step calculation

The time step is dynamically calculated using the Courant-Friedrichs-Lewy (CFL) condition, with a slightly different implementation with respect to the specification in the document. The implemented function computes the timestep by calculating the local sound speed $c_{i,j}^n = (\gamma P_{i,j}^n / \rho_{i,j}^n)^{1/2}$, where γ is the adiabatic index and $P_{i,j}^n$ and $\rho_{i,j}^n$ are the pressure and density in a cell at a given time. Then, we determine the maximum effective wave propagation speeds in the x and y directions by adding the local fluid velocities u and w to the sound speed and taking the maximum absolute values. These represent the fastest signals that could travel across the grid. The timestep is then computed as the minimum of $\Delta x / v_{\mathrm{max},x}$ and $\Delta y / v_{\mathrm{max},y}$, scaled by the CFL prefactor $p = 0.05$ (a more conservative value to account for the simplicity of the approach), which guarantees that information does not travel more than one cell per time step. The resulting time step dt is stored and returned.

It is worth noting that the implementation of the time step calculation as specified in the project description was done, but there were several numerical instabilities, while this simplified approach was more rebust.

4.6 Main program and plotting

The main function of the code uses the `FVis3.py` module as it is explained in the module's documentation. We call the `hydro_solver` method within the `FVis3.save_data` function to evolve the system and save certain snapshots of the data to a file. Then, we visualize the data with the `FVis3.animate_2D` function.

5 Gaussian perturbation and nabla variation Simulations

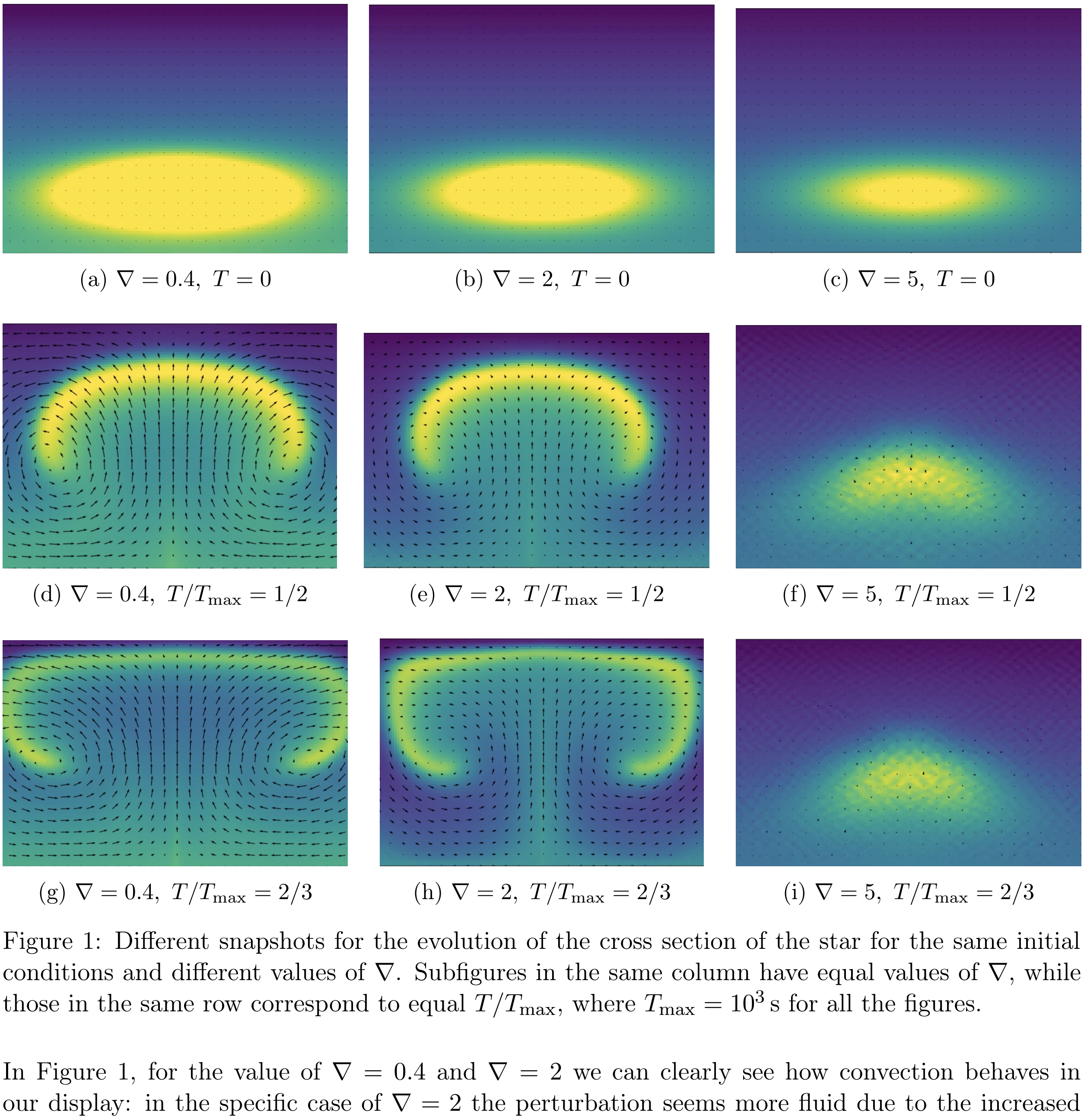
In order to check if our convection simulation works well, we have added a Gaussian perturbation in the centre of our display at the beginning of the simulation. The 2D Gaussian function that we have use is defined as

$$f(x,y) = A \cdot \exp\left[-\left(\frac{(x - x_0^2)}{2\sigma_x^2} + \frac{(y - y_0^2)}{2\sigma_y^2} \right) \right], \tag{53}$$

where:

- A is the amplitude proportional to the central temperature,
- (x_0, y_0) is the centre of the perturbation and its centred in the x-axis and situated in $1/4$ of the height of the y-axis,
- σ_x and σ_y defines the dispersion (10 percent of the box dimensions).

Deferents ∇ values were tested in order to check how our simulation will behave. We will comment down here some screenshots of the tree different values of ∇ that we are going to test and also we will ad the full videos in the exercise 5 folder.



(a) $\nabla = 0.4,\ T = 0$ (b) $\nabla = 2,\ T = 0$ (c) $\nabla = 5,\ T = 0$

(d) $\nabla = 0.4,\ T/T_{\max} = 1/2$ (e) $\nabla = 2,\ T/T_{\max} = 1/2$ (f) $\nabla = 5,\ T/T_{\max} = 1/2$

(g) $\nabla = 0.4,\ T/T_{\max} = 2/3$ (h) $\nabla = 2,\ T/T_{\max} = 2/3$ (i) $\nabla = 5,\ T/T_{\max} = 2/3$

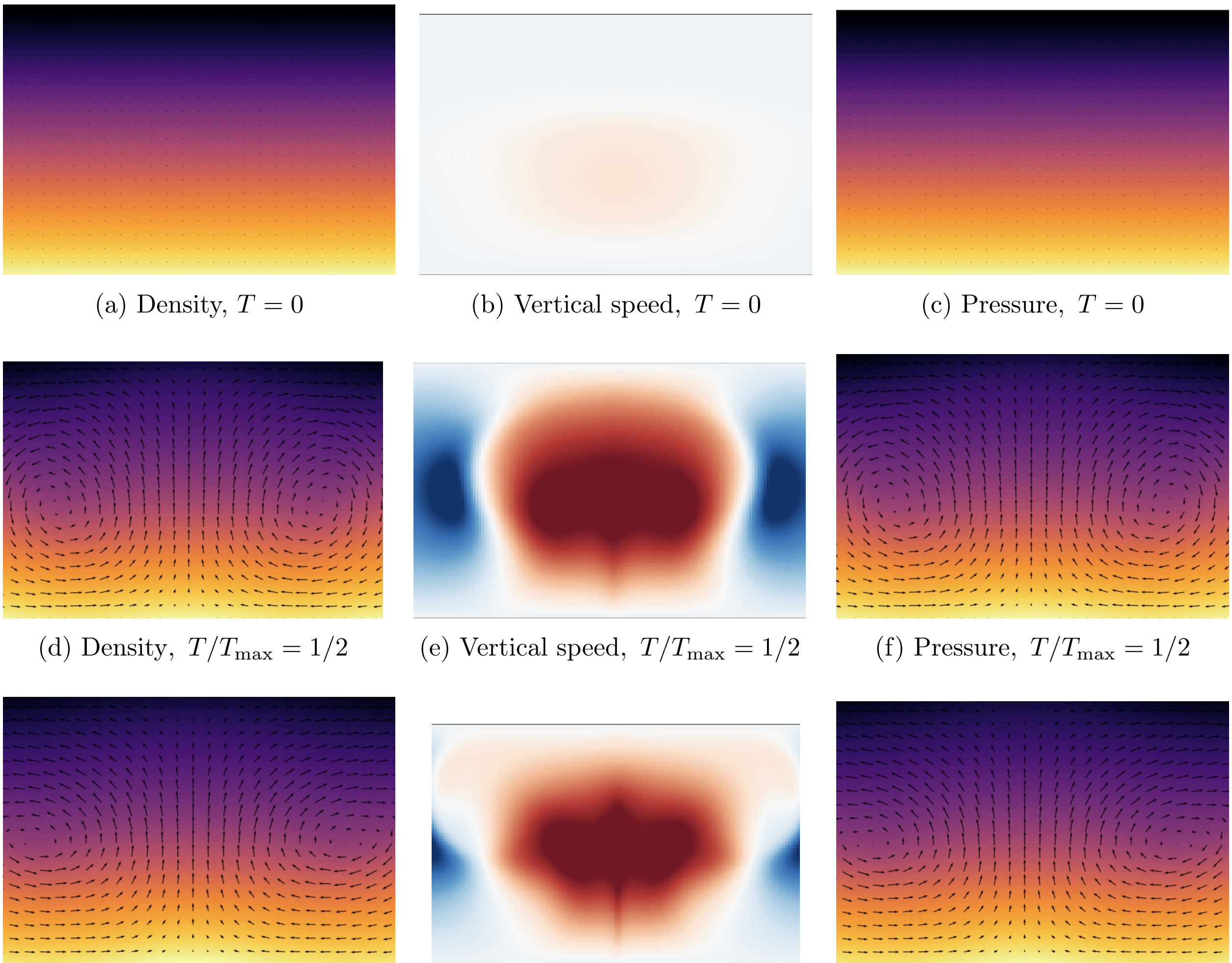
Figure 1: Different snapshots for the evolution of the cross section of the star for the same initial conditions and different values of ∇. Subfigures in the same column have equal values of ∇, while those in the same row correspond to equal $T/T_{\max}$, where $T_{\max} = 10^3$ s for all the figures.

In Figure 1, for the value of $\nabla = 0.4$ and $\nabla = 2$ we can clearly see how convection behaves in our display: in the specific case of $\nabla = 2$ the perturbation seems more fluid due to the increased inestability. Finally, in the simulation with $\nabla = 5$ (3^{rd} column) the instability is so high that our simulation is surrounded by a lot of noise due to disturbances.

6 Parameter variation simulations

For this exercise we will apply the stimulation for the same nablas $\nabla = 0.4$ than the previous one but this time we will plot Energy density, vertical velocity and pressure. To see this in greater detail, please refer to the videos attached with this document.

In Figure 2 we can see that first and the third columns are almost the same. That is because of the proportionality relation $e = P/(\gamma - 1)$. Although this is the only quantity we show in this document, to keep it as compact as possible, many more animations for different values of ∇ and different quantities have been computed.



(a) Density, $T = 0$ (b) Vertical speed, $T = 0$ (c) Pressure, $T = 0$

(d) Density, $T/T_{\max} = 1/2$ (e) Vertical speed, $T/T_{\max} = 1/2$ (f) Pressure, $T/T_{\max} = 1/2$

(g) Density, $T/T_{\max} = 2/3$ (h) Vertical speed,$T/T_{\max} = 2/3$ (i) Pressure, $T/T_{\max} = 2/3$

Figure 2: Different snapshots for the evolution of the cross section of the star for the same ∇ but for different parameters . Sub-figures in the same column have equal parameters, while those in the same row correspond to equal $T/T_{\max}$, where $T_{\max} = 10^3\,\mathrm{s}$ for all the figures.

7 Conclusion

In this project we have successfully modelled the main equations governing the convective motions of the Sun. By applying a Gaussian perturbation to our section of the sun, we have been able to appreciate how this perturbation evolves convectively, allowing us to understand that our simulation respects natural laws and imitates the behaviour of a star in a relatively accurately way.

8 Reflection

I could say that this project has been the one that has given me the greatest learning experience in the course, since having learned the entire computational basis in the previous projects, in this one I have been able to apply all the tools learned to build a quality simulation. The main problems I have faced are, first of all, understanding the FVis module, especially the time plotting, since for several days I was not able to get the simulation to update and it was due to a small comprehension error I had when reading the Requirements on your solver section. On the other hand, another of the main problems I have had to face are some waves produced when applying the Gaussian perturbation that filled the entire simulation with noise. This was because the density remained constant while T and P were perturbed. But we have finally been able to overcome these adversities and create a decent simulation.