

# AST3310 Project 3: 2D simulation of stellar convection

(Dated: May 25, 2025)

In this final project, we develop a 2D hydrodynamic model to simulate plasma convection in the Sun's upper layers, solving discretized continuity, momentum, and energy equations. The model tested convective instability and animated the convection for three temperature gradients  $\nabla^* = [0.41, 4, 8]$ , revealing stronger and more chaotic motions at higher  $\nabla^*$ . While the simulations qualitatively reproduce expected convective patterns, quantitative discrepancies (such as unrealistic convective speeds ( $\sim 10^8$  m/s) and mass non-conservation) highlight limitations of the model and its assumptions and idealized boundary conditions. The results also demonstrate the sensitivity of the solutions of the hydrodynamic equations to the initial conditions, emphasizing the need for future refinements in numerical methods and physical assumptions.

## I. INTRODUCTION

For our third and final project of AST3310 we had to construct a model to simulate the convective motion of plasma in the upper layers of the Sun. In this report, I will present the three hydrodynamic equations used for the simulation and their respective discretisations used in the model as well as how I perturb the initial temperature to provoke convective motion. In addition, I will take a look at how the model performs for three different temperature gradients  $\nabla^* = [0.41, 4, 8]$ .

The developed model simulates the convection inside a two-dimensional “box” of side lengths  $L_x = 12$  Mm in the horizontal direction and  $L_y = 4$  Mm in the vertical direction, with the upper boundary being the solar surface. The box is subdivided into a grid of  $N_x \times N_y$  cells of side lengths  $\Delta x = L_x/N_x$  and  $\Delta y = L_y/N_y$ , where we set  $N_x = 300$  and  $N_y = 100$ .

We assume a monatomic ideal gas with mean molecular weight  $\mu = 0.61$  and degrees of freedom parameter  $\gamma = 5/3$ . The ideal gas has internal energy given by

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

where  $k_B$  is Boltzmann's constant and  $m_u$  is the atomic mass unit.  $\rho$  and  $T$  are the density and temperature of the plasma. The equation of state relates the pressure  $P$  and internal energy through the equation

$$P = (\gamma - 1)e \quad (2)$$

Lastly, the model neglects the effect of the viscous stress tensor on the momentum equation for the plasma but includes the effect of gravity. Due the small size of the box (see fig. 1), we will take the gravitational acceleration  $\mathbf{g} = -\frac{GM_\odot}{R_\odot^2}\hat{\mathbf{y}} \equiv -g\hat{\mathbf{y}}$  to be constant at all points inside the box. Here  $G$  is the gravitational constant,  $M_\odot = 1.989 \times 10^{30}$  kg is the solar mass, and  $R_\odot = 6.96 \times 10^8$  m is the solar radius. This approximation is justifiable since the angular variation across the box is minuscule ( $\Delta\theta \sim 0.988^\circ$ ) making the vertical direction effectively radial, i.e.  $\hat{\mathbf{y}} \parallel \hat{\mathbf{r}}$ . The height of the box is also negligible compared to  $R_\odot$ , so the magnitude of the gravitational acceleration varies very little with the vertical position.

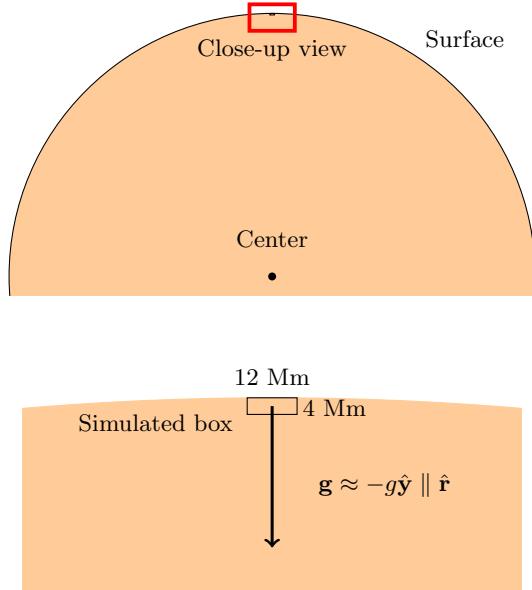


Figure 1. Top: A close-up view of the Sun highlighted by the red box, shown to scale with the full solar disk. Bottom: The simulated region ( $12\text{Mm} \times 4\text{Mm}$ ) in comparison to the close-up view. At this scale, gravitational acceleration is approximately aligned with the  $y$ -axis, as the angular variation across the box is negligible ( $\Delta\theta \sim 0.988^\circ$ ).

## II. THEORY

### A. The governing equations

Given the density  $\rho$  and velocity (flow)  $\mathbf{u} \equiv (u, w)$  of the plasma at each point in the box, our convection model must solve three hydrodynamic equations, namely the continuity equation, the momentum equation for both  $x$  and  $y$ , and the energy equation, which take these forms:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0 \quad (3)$$

$$\frac{\partial \rho \mathbf{u}}{\partial t} + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u}) = -\nabla P + \rho \mathbf{g} \quad (4)$$

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

The continuity equation (3) is formulated assuming no sources and no sinks, i.e. we have mass conservation inside the box. In this case, we can use the product rule to rewrite the equation as

$$\begin{aligned} \frac{\partial \rho}{\partial t} + \mathbf{u} \cdot \nabla \rho + \rho \nabla \cdot \mathbf{u} &= 0 \\ \Leftrightarrow \frac{\partial \rho}{\partial t} &= -u \frac{\partial \rho}{\partial x} - w \frac{\partial \rho}{\partial y} - \rho \left( \frac{\partial u}{\partial x} + \frac{\partial w}{\partial y} \right) \\ &= - \left( \rho \frac{\partial u}{\partial x} + u \frac{\partial \rho}{\partial x} \right) - \left( \rho \frac{\partial w}{\partial y} + w \frac{\partial \rho}{\partial y} \right) \\ \therefore \frac{\partial \rho}{\partial t} &= - \frac{\partial \rho u}{\partial x} - \frac{\partial \rho w}{\partial y} \end{aligned} \quad (6)$$

Our model neglects the effects of the viscosity of the plasma on its momentum, so the momentum equation (4) excludes the viscous stress tensor term. However, we still include the effect of the gravitational acceleration  $\mathbf{g} = -\frac{GM_\odot}{R_\odot^2} \hat{\mathbf{y}} \equiv -g \hat{\mathbf{y}}$ . To get the explicit momentum equations we must first write out the second term on the right hand side using the outer product

$$\begin{aligned} \mathbf{u} \otimes \mathbf{u} &= \begin{bmatrix} u^2 & uw \\ uw & w^2 \end{bmatrix} \\ \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u}) &= \begin{bmatrix} \frac{\partial}{\partial x} & \frac{\partial}{\partial y} \end{bmatrix} \begin{bmatrix} \rho u^2 & \rho uw \\ \rho uw & \rho w^2 \end{bmatrix} \\ &= \left[ \left( \frac{\partial \rho u^2}{\partial x} + \frac{\partial \rho uw}{\partial y} \right), \left( \frac{\partial \rho uw}{\partial x} + \frac{\partial \rho w^2}{\partial y} \right) \right] \end{aligned}$$

So in the  $x$ -direction, we get

$$\frac{\partial \rho u}{\partial t} = -\frac{\partial \rho u^2}{\partial x} - \frac{\partial \rho uw}{\partial y} - \frac{\partial P}{\partial x} \quad (7)$$

while in the  $y$ -direction we must also include the gravity term  $-\rho g \hat{\mathbf{y}}$  such that the vertical momentum equation reads

$$\frac{\partial \rho w}{\partial t} = -\frac{\partial \rho w^2}{\partial y} - \frac{\partial \rho uw}{\partial x} - \frac{\partial P}{\partial y} - \rho g \quad (8)$$

Finally, the internal energy equation (5) can be written out in a similar way to the continuity equation, just adding the pressure term  $-P \nabla \cdot \mathbf{u}$

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

## B. The initial conditions

Initially, we let the plasma in the simulation box be in hydrostatic equilibrium with zero velocity everywhere,

while the surface temperature  $T_\odot = 5778$  K and pressure  $P_\odot = 1.8 \times 10^4$  Pa are set to match the values of the solar photosphere. To find the initial values for  $T$  and  $P$  inside the rest box, we also need the equation of state

$$P = (\gamma - 1)e = \frac{\rho k_B}{\mu m_u} T \quad (10)$$

and the temperature gradient of the star

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

which we will let assume values between  $2/5$  and  $10$  in this project. From these two expressions we can formulate a differential equation for  $T(y)$  by evoking the chain rule and the hydrostatic equilibrium condition

$$-\nabla P + \rho \mathbf{g} = 0 \Rightarrow \frac{\partial P}{\partial y} = -\rho g \quad (11)$$

This gives

$$\begin{aligned} \nabla^* &= \frac{P}{T} \frac{\partial T}{\partial y} \frac{\partial y}{\partial P} \\ &= -\frac{\rho k_B}{\mu m_u} \frac{\partial T}{\partial y} \frac{1}{\rho g} \\ \Leftrightarrow \frac{\partial T}{\partial y} &= -\frac{\mu m_u g}{k_B} \nabla^* \end{aligned}$$

Separating the variables and integrating both sides from  $y$  to the height  $H$  of the box we get

$$\begin{aligned} \int_{T(y)}^{T_\odot} dT' &= -\frac{\mu m_u g}{k_B} \nabla^* \int_y^H dy' \\ T(y) &= T_\odot + \frac{\mu m_u g}{k_B} \nabla^* (H - y) \\ T(y) &= T_\odot + \frac{\mu m_u g}{k_B} \nabla^* (H - y) \end{aligned} \quad (12)$$

From the initial temperature, we can now find the initial pressure through

$$\begin{aligned} \nabla^* &= \frac{P}{T} \frac{\partial T}{\partial P} \\ \Rightarrow \frac{dP}{P} &= \frac{1}{\nabla^*} \frac{dT}{T} \end{aligned}$$

Integrating both sides up to the surface values, we get

$$\ln \frac{P_\odot}{P} = \frac{1}{\nabla^*} \ln \frac{T_\odot}{T}$$

$$P(y) = P_\odot \left[ \frac{T(y)}{T_\odot} \right]^{1/\nabla^*} \quad (13)$$

Once our model has calculated the initial temperature and pressure values, we can calculate the initial internal energy and density from the equation of state, i.e.

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

### C. The boundary conditions

To successfully simulate the convection inside the box we must impose appropriate boundary conditions. To discuss these conditions, let's first introduce the notation used in the project description. Let  $\phi_{i,j}^n$  refer to some variable  $\phi$  at time step  $n$  and spatial coordinates  $(i, j)$  in the grid of cells with  $i \in [0, N_x - 1]$  and  $j \in [0, N_y - 1]$ .

The simplest condition we impose is that the primary variables are periodic in the horizontal direction (along the  $x$ -axis), i.e.

$$\phi_{-1,j}^n = \phi_{N_x-1,j}^n \quad \text{and} \quad \phi_{N_x+1,j}^n = \phi_{0,j}^n$$

for all primary variables  $\phi$  and at all vertical positions  $y_j$  in the box. This can be understood as placing identical boxes to the left and right of our simulated box.

Next, we ensure that the vertical components of the velocity are zero at the surface and at the bottom of the box:  $w_{i,N_y-1}^n = w_{i,0}^n = 0$ . We also impose the condition that the vertical gradient of the horizontal component of the velocity should be zero at the boundaries. This is achieved by using the 3-point forward difference approximation

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

along the bottom of the box and the 3-point *backward* difference approximation

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

at the surface.

Lastly, we must find the boundary conditions for the internal energy and the density at the boundaries. We require that hydrostatic equilibrium (11) must be fulfilled at the boundaries. From the equation of state (10), we get

$$\frac{\partial e}{\partial y} = \frac{1}{\gamma - 1} \frac{\partial P}{\partial y} = -\frac{\rho g}{\gamma - 1} = -\frac{\mu m_u g}{k_B T(\gamma - 1)} e$$

Using the 3-point forward/backward difference approximations for  $\partial e / \partial y$ , we get the boundary conditions

$$\begin{aligned} e_{i,0}^n &= \frac{4e_{i,1}^n - e_{i,2}^n}{3 - 2\Delta y \frac{\mu m_u g}{k_B T_{i,0}^n}} \\ e_{i,N_y-1}^n &= \frac{4e_{i,N_y-2}^n - e_{i,N_y-3}^n}{3 - 2\Delta y \frac{\mu m_u g}{k_B T_{i,N_y-1}^n}} \end{aligned}$$

From these values at the boundary, we can also calculate the density as

$$\begin{aligned} \rho_{i,0}^n &= (\gamma - 1) \frac{\mu m_u e_{i,0}^n}{k_B T_{i,0}^n} \\ \rho_{i,N_y-1}^n &= (\gamma - 1) \frac{\mu m_u e_{i,N_y-1}^n}{k_B T_{i,N_y-1}^n} \end{aligned}$$

## III. METHOD

### A. Discretisations of the governing equations

#### 1. Numerical differentiation schemes

We need to solve the hydrodynamic equations (6-9) numerically and thus need explicit discretisations of said equations. To calculate the derivatives of the primary variables, we will employ two different numerical schemes: first order upwind differencing and Forward Time Centered Space (FTCS).

The upwind differencing scheme is used for advective terms which are terms that consist of a flow component ( $u$  or  $w$ ) multiplied with a derivative in its corresponding direction. In this scheme, the derivatives of a variable  $\phi$  will be calculated as

$$\left[ \frac{\partial \phi}{\partial x} \right]_{i,j}^n \approx \begin{cases} \frac{\phi_{i,j}^n - \phi_{i-1,j}^n}{\Delta x} & \text{if } u_{i,j}^n \geq 0 \\ \frac{\phi_{i+1,j}^n - \phi_{i,j}^n}{\Delta x} & \text{if } u_{i,j}^n < 0 \end{cases} \quad (14)$$

$$\left[ \frac{\partial \phi}{\partial y} \right]_{i,j}^n \approx \begin{cases} \frac{\phi_{i,j}^n - \phi_{i,j-1}^n}{\Delta y} & \text{if } w_{i,j}^n \geq 0 \\ \frac{\phi_{i,j+1}^n - \phi_{i,j}^n}{\Delta y} & \text{if } w_{i,j}^n < 0 \end{cases} \quad (15)$$

An intuitive explanation for why we use this scheme is that the upwind scheme accounts for the direction of the flow when calculating derivatives. Since the information follows the flow of the plasma, the most accurate way to approximate a derivative at any point is to use data from the upwind direction, i.e. where the flow is coming from. Say for example that the plasma is moving to the right at a certain grid point (see fig. 2). We assume that the cells are sufficiently small and that the flow of the plasma is well-behaved such that there are no discontinuous variations in the flow. Then its behavior is mostly influenced by the left neighboring cell (the upwind source), and so a left-sided derivative is more physically meaningful [2].

The FTCS scheme is used for the other non-advective terms. In this scheme, the derivatives of  $\phi$  are simply calculated as

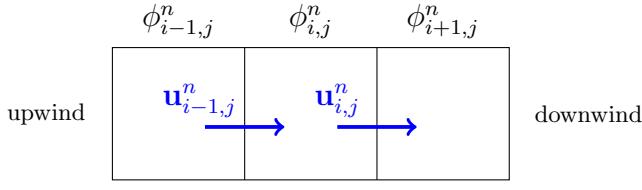


Figure 2. Example of upwind differencing. The flow towards the right means that the value of  $\phi$  from the left neighboring cell must be used when calculating  $\partial\phi/\partial x$  at the point  $(i, j)$  since this is the value that “flows into the cell”.

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

## 2. The internal energy: an example of how to implement the discretisation

As an example, the equation for the internal energy  $e$  (9), which can be rewritten using the chain rule as

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

has the following discretisation in the case where  $u, w \geq 0$

$$\left[\frac{\partial e}{\partial t}\right]_{i,j}^n = -(e_{i,j}^n + P_{i,j}^n) \left[ \underbrace{\frac{u_{i+1,j}^n - u_{i-1,j}^n}{2\Delta x}}_{\text{FTCS}} + \underbrace{\frac{w_{i,j+1}^n - w_{i,j-1}^n}{2\Delta y}}_{\text{FTCS}} \right] - u_{i,j}^n \underbrace{\frac{e_{i,j}^n - e_{i-1,j}^n}{\Delta x}}_{\text{upwind}} - w_{i,j}^n \underbrace{\frac{e_{i,j}^n - e_{i,j-1}^n}{\Delta y}}_{\text{upwind}}$$

In the general case, the two last terms will be calculated following the direction of the flow as described in eqs. (14) & (15). The left hand side of the equation can be approximated using forward time differentiation

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

This gives us an algorithm for advancing the internal energy in time

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

Similarly, we use the discretisation and time advancement algorithm for the density  $\rho$  proposed in eqs. (19) & (20) in the project description.

## 3. The momentum equations

To advance the velocity values in time we must first rewrite the momentum eqs. (7) & (8) using the product rule

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

which accounts for the change in velocity  $(u, w)$ . The first of these equations has a proposed discretisation given in eq. (22) of the project description

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

where I have indicated which differentiation method is used under each derivative. Even though  $\partial w/\partial y$  is multiplied with  $\rho u$ , we do not use the upwind differencing scheme since the velocity component  $u$  is orthogonal to the direction of the derivative ( $y$ -direction). Following the same method for discretizing the second momentum equation we get

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

Once we have advanced  $\rho u$  and  $\rho w$  in time, we can find the algorithm for  $u_{i,j}^{n+1}$  and  $w_{i,j}^{n+1}$

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

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

#### 4. Time step length

To choose an optimal time step length  $\Delta t$  that is small enough for the model to be stable while still being sufficiently big to minimize the run time, we will implement a variable time step length. This is achieved by selecting the biggest  $\Delta t$  that ensures that the relative changes of the primary variables ( $\rho$ ,  $e$ ,  $u$  and  $w$ ) never exceeds some percentage  $p$  for any point inside the box. From the time derivatives of the primary variables  $\rho$  and  $e$  we can calculate the relative change  $\Delta\phi/\phi$  of each variable  $\phi$  for each time step  $\Delta t$ :

$$\text{rel}(\phi) = \frac{\Delta\phi/\phi}{\Delta t} \approx \left| \frac{1}{\phi} \frac{\partial \phi}{\partial t} \right|$$

An additional condition is that a fluid particle's velocity should not be so high that it skips over an entire grid point within a single time step. We must therefore also calculate the relative change in position at each point:

$$\text{rel}(x) = \left| \frac{1}{\Delta x} \underbrace{\frac{\partial x}{\partial t}}_u \right| = \left| \frac{u}{\Delta x} \right|$$

$$\text{rel}(y) = \left| \frac{1}{\Delta y} \underbrace{\frac{\partial y}{\partial t}}_w \right| = \left| \frac{w}{\Delta y} \right|$$

Given these relative changes of each primary variable, we must compare their respective maxima inside the box and choose the maximum of these:

$$\delta \equiv \max \left\{ \max [\text{rel}(\phi)], \forall \phi \right\}$$

The value of  $\delta$  will then be the biggest relative change of any variable inside the box. We then choose  $\Delta t$  such that

$$\delta \cdot \Delta t = p \Rightarrow \Delta t = p/\delta$$

For my model, I ended up using  $p = 0.01$ . I also implemented an upper bound of  $\Delta t = 0.1$  s for the time step since for conditions close to hydrostatic equilibrium or with tiny perturbations, the relative changes were so small that they yielded giant time steps, often bigger than the intended total simulation time.

## B. Perturbing the initial temperature gradient

After initialising the plasma in the simulation box according the initial conditions presented in section II B, the plasma will be in hydrostatic equilibrium with zero velocity everywhere by definition. So to provoke the gas to be convectively unstable we can perturb the initial temperature (12) by adding some regions of hotter or colder plasma.

This was achieved by adding one or more 2D Gaussian perturbation(s) to the temperature after calculating the initial temperature in the box. The 2D Gaussian has the form

$$\delta T(x, y) = T_{\text{pert}} \exp \left[ -\frac{(x - x_0)^2}{2\sigma_x^2} - \frac{(y - y_0)^2}{2\sigma_y^2} \right]$$

where  $T_{\text{pert}}$  is the amplitude of the perturbation,  $x_0$  and  $y_0$  are the coordinates of the center of the Gaussian, and  $\sigma_x$  and  $\sigma_y$  are the standard deviations in the  $x$ - and  $y$ -directions.

## C. Model structure

The python-script used to simulate the convection inside the box has this general structure:

1. Initialise the box and set the initial values of  $u$ ,  $w$ ,  $T$ ,  $P$  at each point in the grid according to the conditions described in section II B.

2. Perturb the original temperature  $T_i$  (12) by adding  $N$  2D Gaussians centered at different points in the box, i.e. let

$$T_i \rightarrow T(x, y) = T_i + \sum_{n=1}^N \delta T_n(x, y)$$

3. Calculate the initial values of  $e$  and  $\rho$  given the perturbed initial temperature using the equation of state. This is key since the variations in the density gives rise to buoyant forces and thus ensures that the plasma is provoked to become convectively unstable.

4. At each time step, calculate the derivatives of the four primary variables using the discretisations discussed in section III A and listed in appendix A 1

5. Find the optimal time step length  $\Delta t$  and evolve the four primary variables in time using the Forward Euler method as listed in appendix A 2.

6. Apply the boundary conditions described in section II C.

Parameter	Case nr. 1	Case nr. 2	Case nr. 3
$\nabla^*$	0.41	4	8
$T_{\text{pert}}$	$1.94 \times 10^4$ K	$1.64 \times 10^5$ K	$\pm 3.25 \times 10^5$ K
$x_0$	[3,6,9] Mm	[3,6,9] Mm	$\in [4,8]$ Mm
$y_0$	0	0	$\in [0,4]$ Mm
$\sigma_x$	0.3 Mm	0.3 Mm	$\in [0,0.5]$ Mm
$\sigma_y$	1 Mm	1 Mm	$\in [0,0.5]$ Mm

Table I. Parameters for the three cases convection cases shown in figs. 5–7. The values for the third case were chosen at random from a uniform distribution over the given intervals.

7. Update the temperature  $T$  and the pressure  $P$  after the boundary conditions are imposed:

$$P_{i,j}^{n+1} = (\gamma - 1)e_{i,j}^{n+1}$$

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

8. Repeat steps 4 to 7 for each time step until the total simulation time is reached.

## IV. RESULTS

In figs. 5–7, I show snapshots from 1000 s of simulation from three different sets of initial conditions. The parameters of the three cases are listed in tab. I. For the two first cases, three equally spaced perturbations with amplitude  $T_{\text{pert}} = \max(T_i)/2$ . The only difference between case nr. 1 and 2 is the temperature gradient;  $\nabla^* = 0.41$  in the first case, which is just slightly higher than the adiabatic value of  $\nabla_{\text{ad}} = 2/5 = 0.4$ , while  $\nabla^* = 4$  in the second case. Fig. 5 shows the temperature  $T(\mathbf{r}, t)$  in the box together with the velocities  $\mathbf{u}(\mathbf{r}, t)$  plotted as a vector field. Fig. 6 shows the density  $\rho(\mathbf{r}, t)$  instead of the temperature.

The third case introduces three random perturbations with the same amplitude. The temperature gradient is also the highest out of the three cases with  $\nabla^* = 8$ . This set of initial conditions was chosen mostly to highlight the chaotic nature of the solutions of the hydrodynamic equations, as well as serving as a test of the model's stability for high  $\nabla^*$  values.

In fig. 3, we can observe the average speed  $v = \sqrt{u(\mathbf{r}, t)^2 + w(\mathbf{r}, t)^2}$  inside the box as a function of time for the three cases. I have also plotted the relative change in average density  $\Delta\rho(\mathbf{r}, t)/\rho(\mathbf{r}, 0)$  inside each of the boxes in fig. 4.

## V. DISCUSSION

### A. Sanity check

As a “sanity check” for our convection model we had to simulate the box in hydrostatic equilibrium with a

temperature gradient of  $\nabla^* = \nabla_{\text{ad}} = 2/5$  and without perturbing the temperature. In the resulting video from simulating 60 s of this box, we do not see any noticeable changes to the temperature nor the velocity field, so the model passes the sanity check qualitatively.

However, when I checked the maximum value of the vertical component  $w$  in the box, it reached massive values of  $|w| \sim 10^6$  m/s which I was not able to explain. Nevertheless, the qualitative behavior of the hydrostatic equilibrium case and the subsequent convectively unstable cases shown in the videos and snapshots seemed realistic despite this strange problem.

### B. The effect of larger temperature gradients

The most obvious effect of increasing the temperature gradient  $\nabla^*$  is that the plasma becomes more and more convectively unstable. In fact, if we look at the average speed in the box in the three cases (fig. 3), we see that the peak increases for higher values of  $\nabla^*$ ; going from  $\bar{v}_{\text{max}} \sim 2 \times 10^8$  m/s for  $\nabla^* = 0.41$ , to  $\bar{v}_{\text{max}} \sim 8 \times 10^8$  m/s for  $\nabla^* = 8$ . This means the convective motion is more dramatic for large values of  $\nabla^*$ . The plasma also takes longer to “settle down” after the initial perturbations for the large gradients, which is both reflected in the average speed graphs (fig. 3) and in the snapshots in figs. 5–7 where we see that there is little movement at the end of the simulation for the first case while there is still a lot of convective motion in the two other cases.

This connection between between large temperature gradients and convective instability is consistent with what we would expect from the theory presented in the lecture notes and in project 2 of AST3310. Indeed, since  $\nabla^* = 0.41$  is just slightly higher than the adiabatic value  $\nabla_{\text{ad}} = 2/5$ , the instability criterion  $\nabla^* > \nabla_{\text{ad}}$  is barely satisfied in the first case, which explains the undramatic convective motion.

We notice another effect of larger gradients when looking at the relative change in density for each of the three cases (fig. 4). In the first case, average density in the box stays approximately constant, only increasing by about  $\sim 2\%$  after 1000 s. This indicates satisfactory numerical stability since the mass inside the box is more or less conserved.

However, this variation increases with larger gradients. For the box with  $\nabla^* = 8$ , the average density has increased by well over 400% compared to the initial value. This means that we are not even close to having mass conservation and that the model is not numerically stable for large temperature gradients.

### C. Chaos

The third set of initial conditions serves as a great demonstration of the chaotic nature of the solutions to

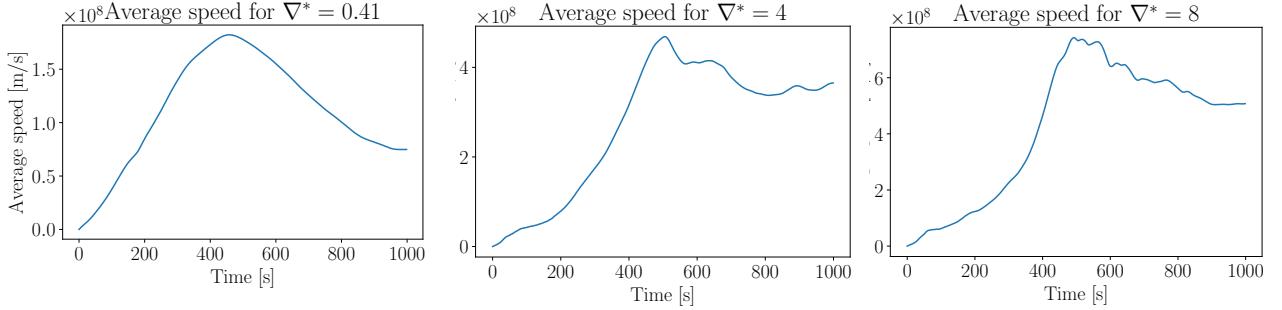


Figure 3. The average speed  $v = \sqrt{u(\mathbf{r}, t)^2 + w(\mathbf{r}, t)^2}$  as a function of time plotted for the three cases with their respective temperature gradients  $\nabla^* = [0.41, 4, 8]$ .

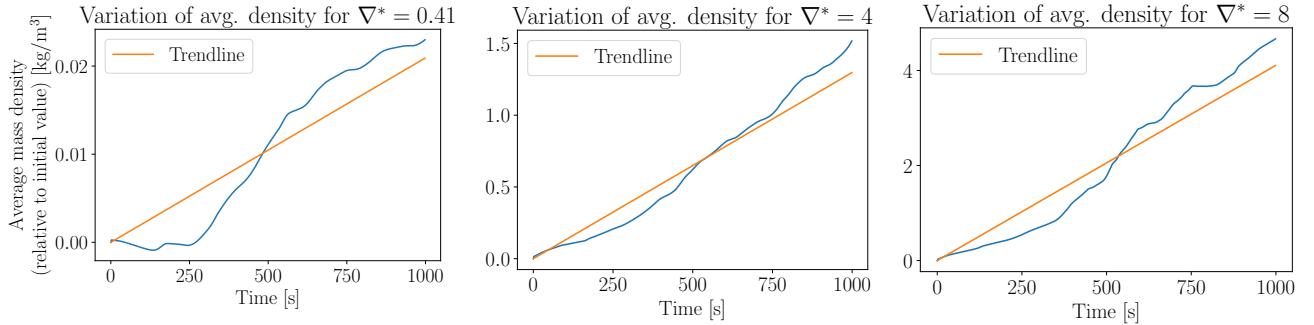


Figure 4. The relative change in average density  $\Delta\rho(\mathbf{r}, t)/\rho(\mathbf{r}, 0)$  as a function of time for the three cases with their respective temperature gradients  $\nabla^* = [0.41, 4, 8]$ . The orange line indicates the general trends.

the hydrodynamic equations. Taking a look at the bottom snapshot in fig. 7, we see that when the initial perturbations are not symmetrically distributed in the box, the resulting pattern is chaotic and hard to predict. In fact, the system of equations we solve in this model will be really sensitive to the initial conditions, with small differences in the initial conditions giving rise to big differences later in the simulation.

#### D. Validity of the model

As I have alluded to earlier in the discussion, the model performs well when it comes to recreating the qualitative behaviour we would expect from convective motion. The rendered animations of the three convectively unstable cases show motions similar to what we see in boiling water. Perhaps a more scientific consideration is that the second case (fig. 6) shows the qualitative connection between the temperature, density and convective instability we would expect. In the snapshot in the middle of fig. 6 we see that the regions of lowest density correspond to the perturbed regions of hot plasma. Inside these pockets of hot and light plasma, the vertical velocity increases due to the buoyant force that arises from the density differences and the pressure gradient.

Nevertheless, the model fails to give satisfactory quantitative results. One problem is the non-zero values of  $w$  that appear in the hydrostatic equilibrium case in the sanity check. Another is that the average speeds (fig. 3) are way too high if we compare them to the values observed in the granulation pattern of the Sun. The radial convective velocities in the granulation pattern of the Sun are only around  $\sim 10^2$  m/s [1]. I was not able to find any sources about the convective velocity in the tangential direction, but it seems unlikely that it would be high enough to make the average speed  $\bar{v} \sim 10^8$  m/s like our model predicts.

The numerical instability that manifests itself in the increase of the mass inside the box is yet another problem for large temperature gradients.

I suspect that a likely source for these problems are the approximations we make (ideal gas, gravity only vertical, no viscous stress tensor, etc.) and the somewhat unphysical boundary conditions we impose. This suspicion is reinforced when looking at the waves that appear in the middle snapshot of fig. 7 propagating from the lower boundary. These waves are an artifact of the vertical boundary conditions we impose (the plasma “bounces off the floor”) which would not happen in the Sun. Including the stress tensor in the momentum equation (4) should also reduce the velocities.

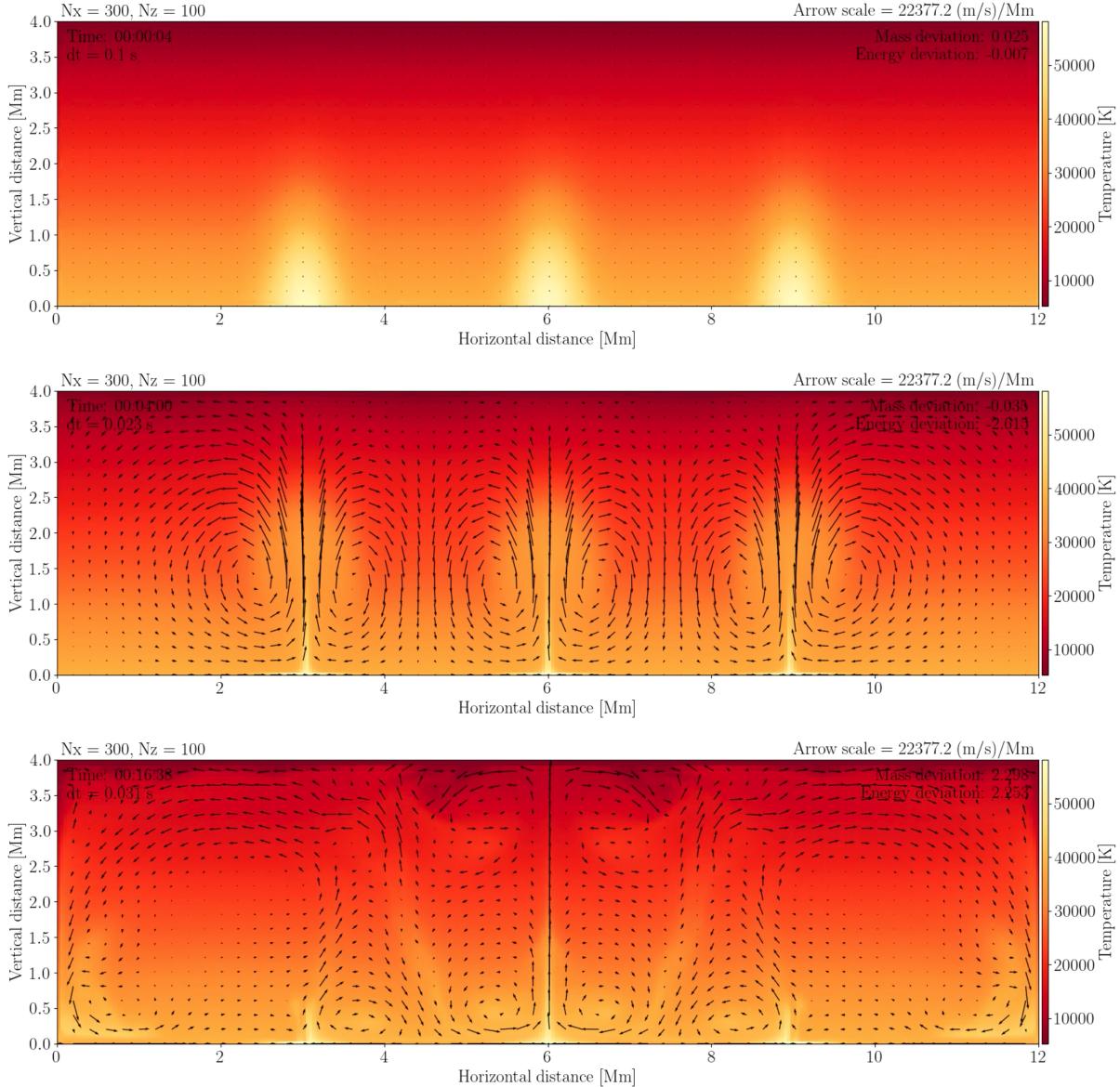


Figure 5. Case nr. 1: temperature gradient  $\nabla^* = 0.41$ , std.  $\sigma_x = 0.3$  Mm,  $\sigma_y = 1$  Mm, and perturbation amplitude  $T_{\text{pert}} = 1.94 \times 10^4$  K. The Gaussians were placed at  $x_0 = [3, 6, 9]$  Mm and  $y_0 = 0$ . *Top:* initial temperature  $T(\mathbf{r}, t = 4$  s) with the perturbations clearly visible and no movement. *Middle:* the temperature  $T(\mathbf{r}, t = 240$  s) shows the pockets of hot plasma rising. *Bottom:* at the end of the simulation time  $T(\mathbf{r}, t = 998$  s).

Another source of inaccuracies could be that the amplitudes of the temperature perturbations were too high, giving too dramatic changes in temperature for the resolution of the box (only  $300 \times 100$  cells). However, I sadly did not have the time to experiment with different perturbation amplitudes to check if this was the case.

## VI. CONCLUSION

This project modeled convective motion of plasma inside a rectangular box in the Sun's upper layers using

discretized hydrodynamic equations. The simulations demonstrated that larger temperature gradients  $\nabla^*$  led to stronger convective instability, but also revealed numerical instabilities at high  $\nabla^*$ 's, violating mass conservation. While the model qualitatively reproduced convective patterns, quantitative discrepancies – such as unrealistically high velocities – highlight limitations of the model and its approximations and boundary conditions. Future work could refine these assumptions and explore higher resolutions or alternative numerical schemes to improve accuracy.

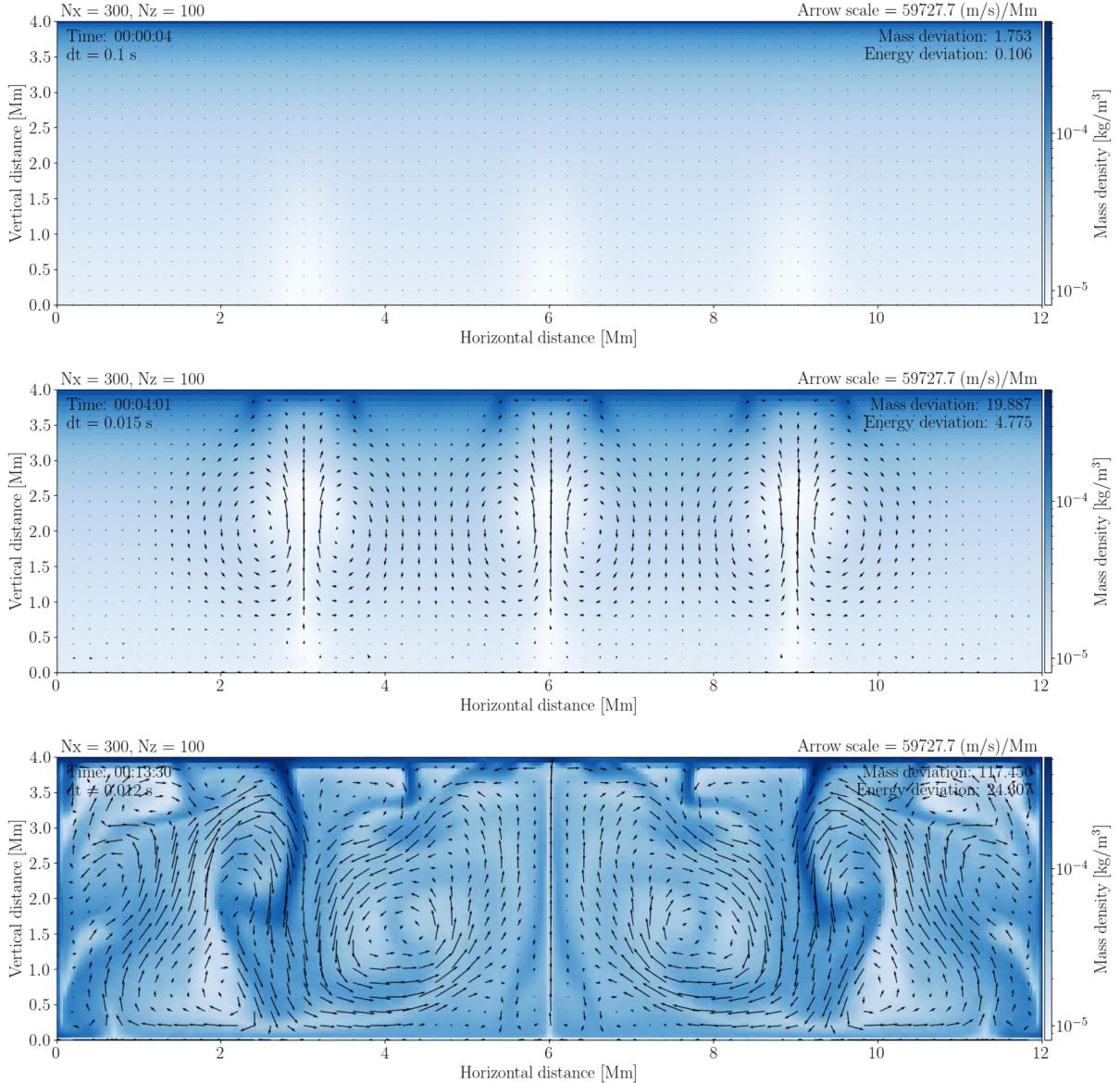


Figure 6. Case nr. 2: temperature gradient  $\nabla^* = 4$ , std.  $\sigma_x = 0.3$  Mm,  $\sigma_y = 1$  Mm, and perturbation amplitude  $T_{\text{pert}} = 1.64 \times 10^5$  K. The Gaussians were placed at  $x_0 = [3, 6, 9]$  Mm and  $y_0 = 0$ . Top: initial density  $\rho(\mathbf{r}, t = 4$  s) with the perturbations of less dense plasma being faintly visible. Middle: the density  $\rho(\mathbf{r}, t = 241$  s) shows the pockets of less dense plasma rising. Bottom: near the end of the simulation time  $\rho(\mathbf{r}, t = 810$  s).

## VII. REFLECTION

First of all, I would say that this project was quite challenging! The analytical work we had to do was fairly quick and painless, but the programming was frustrating. It took some time to learn how to use FVis3 and its limitations made me dive into the script and add some features here and there, which took some time (e.g. adding the possibility to get a logarithmic colormap for the den-

sity plots). The code was also terribly slow, often taking 10-20 min to simulate and render the animations!

It was really fun to try to explore different aspects of the solutions and try to explain the behaviour and the errors.

All in all, I am moderately satisfied with the performance of the model. I am still disappointed that I was not able to identify the problem with the non-zero  $w$  values in the sanity check!

At least I was able to produce some cool animations...

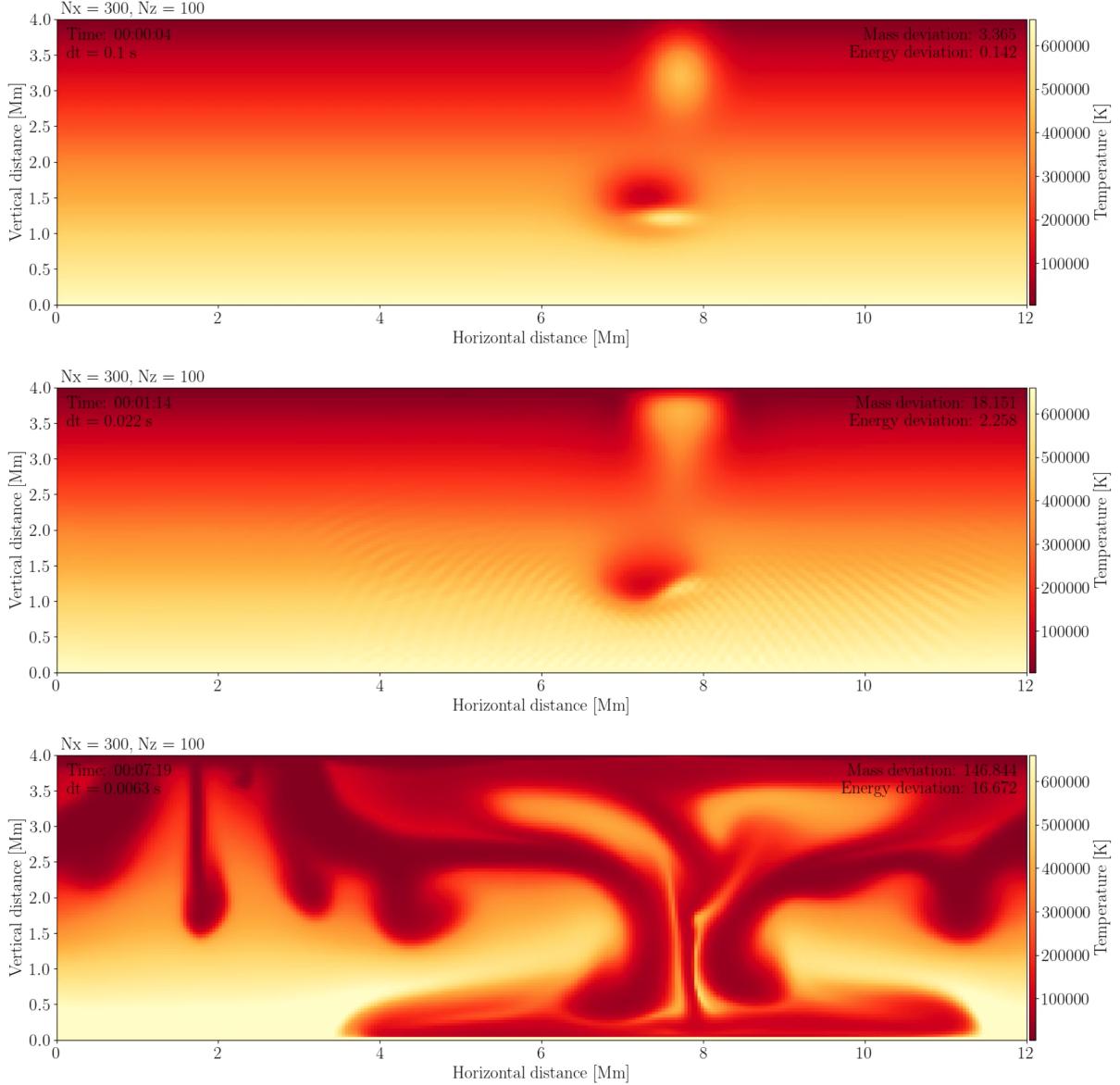


Figure 7. Case nr. 3: temperature gradient  $\nabla^* = 8$  and perturbation amplitude  $|T_{\text{pert}}| = 3.25 \times 10^5$  K. The Gaussians were placed at random points picked from a uniform distribution with  $x_0 \in [4, 8]$  Mm and  $y_0 \in [0, 4]$  Mm. The std.  $\sigma_x, \sigma_y$  were also chosen at random. *Top*: initial temperature  $T(\mathbf{r}, t = 4$  s) with the three perturbations of hot plasma being clearly visible. *Middle*: the temperature  $T(\mathbf{r}, t = 74$  s) shows temperature waves propagating from the lower boundary and interfering with each other. *Bottom*: for the temperature  $T(\mathbf{r}, t = 439$  s), we observe a chaotic pattern.

- 
- [1] NASA Marshall Space Flight Center. Surface flows. Solar Physics Group, NASA Marshall Space Flight Center, 2025. [Online; accessed 25.05.2025].
  - [2] Wikipedia contributors. Upwind scheme — Wikipedia, the free encyclopedia. [https://en.wikipedia.org/wiki/Upwind\\_scheme](https://en.wikipedia.org/wiki/Upwind_scheme), 2025. [Online; accessed 24.05.2025].

## Appendix A: Discretisations used for the differential equations

### 1. Derivatives

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

### 2. Time evolution

$$\begin{aligned}
\rho_{i,j}^{n+1} &= \rho_{i,j}^n + \left[ \frac{\partial \rho}{\partial t} \right]_{i,j}^n \Delta t \\
e_{i,j}^{n+1} &= e_{i,j}^n + \left[ \frac{\partial e}{\partial t} \right]_{i,j}^n \Delta t \\
u_{i,j}^{n+1} &= \frac{[\rho u]_{i,j}^{n+1}}{\rho_{i,j}^{n+1}} = \frac{[\rho u]_{i,j}^n + \left[ \frac{\partial \rho u}{\partial t} \right]_{i,j}^n \Delta t}{\rho_{i,j}^{n+1}} \\
w_{i,j}^{n+1} &= \frac{[\rho w]_{i,j}^{n+1}}{\rho_{i,j}^{n+1}} = \frac{[\rho w]_{i,j}^n + \left[ \frac{\partial \rho w}{\partial t} \right]_{i,j}^n \Delta t}{\rho_{i,j}^{n+1}}
\end{aligned}$$