Stellar convection

Candidate number 24
(Dated: May 26, 2024)

I. INTRODUCTION

In this report we will study how to model 2D convection in the upper layers of a star by solving the three equations of hydrodynamics numerically: the continuity equation, the momentum equation and the internal energy equation. We will go through how we discretize the equations, which numerical schemes we use, how we calculate the initial conditions, which boundary conditions we use and how we calculate the time step length. Firstly, we study how and if we can make the system stay in hydrostatic equilibrium. Then we will add a Gaussian temperature perturbation to the initial temperature to get convective motion. Using simulations of temperature, velocity and energy flux we will study how the system changes with time.

To make our convection-model a bit simpler, we will assume that the gravitational acceleration is constant, that the plasma is an ideal gas with constant molecular weight $\mu = 0.61$ and neglect turbulence and internal friction.

II. METHOD

A. Governing equations

To model the convection process in the star we will use the three hydrodynamic equations

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

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

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

The first equation 1 is the continuity equation without any sources or sinks. The second equation 2 is the momentum equation where we have excluded the stress tensor, but included gravity. Lastly, we have equation 3 which is the energy conservation equation [1]. In the equations above, ρ is the mass density, P is the pressure of the gas, $\boldsymbol{g}$ is the gravitational acceleration, e is the internal energy and $\mathbf{u} = (u, w)$ is the velocity in x and y direction respectively. The three equations are commonly used in hydrodynamics to describe the movement of a liquid. We will use them to describe the convection happening in the star.

We assume the gravitational acceleration to be constant inside our computational area and set the value to be $|\boldsymbol{g}| = GM_\odot/R_\odot^2$. The flow $\mathbf{u} = (u, w)$ is however, assumed to be non-constant [1].

B. Separating equations in x- and y-direction

Before we can solve the hydrodynamic equations numerically, we first need to separate them into x- and y-direction. We let x be the horizontal position along the surface of the star and y the vertical direction pointing in the radial direction. We use Cartesian coordinates instead of radial coordinates since our computational volume is small compared to the entire star. We start with the continuity equation. This equation, given by equation 1, can be separated into x- and y-direction by using that $\mathbf{u} = (u, w)$ where u and w is the velocity in x- and y-direction respectively and that $\nabla = \left(\frac{\partial}{\partial x}, \frac{\partial}{\partial y}\right)$. Calculating the dot product gives:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0 \tag{4}$$

$$\frac{\partial \rho}{\partial t} = -\frac{\partial \rho u}{\partial x} - \frac{\partial \rho w}{\partial y} \tag{5}$$

Next, we rewrite the momentum equation, given in equation 2. We start by calculating the outer product

$$(\rho \mathbf{u} \otimes \mathbf{u}) = \rho \begin{pmatrix} u \\ w \end{pmatrix} \times \begin{pmatrix} u \\ w \end{pmatrix} = \begin{pmatrix} \rho u^2 & \rho uw \\ \rho uw & \rho w^2 \end{pmatrix} \tag{6}$$

and apply the ∇-operator:

$$\nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u}) = \begin{pmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \end{pmatrix} \begin{pmatrix} \rho u^2 & \rho uw \\ \rho uw & \rho w^2 \end{pmatrix} \tag{7}$$

$$= \begin{pmatrix} \frac{\partial}{\partial x}\rho u^2 + \frac{\partial}{\partial y}\rho uw \\ \frac{\partial}{\partial x}\rho uw + \frac{\partial}{\partial y}\rho w^2 \end{pmatrix}. \tag{8}$$

We see that the momentum equation then separates into 2 equations, one in x-direction and one in y-direction:

$$\frac{\partial \rho u}{\partial t} = -\frac{\partial \rho u^2}{\partial x} - \frac{\partial \rho uw}{\partial y} - \frac{\partial P}{\partial x} \tag{9}$$

$$\frac{\partial \rho w}{\partial t} = -\frac{\partial \rho w^2}{\partial y} - \frac{\partial \rho uw}{\partial x} - \frac{\partial P}{\partial y} - \rho g. \tag{10}$$

Only the momentum equation in y-direction 10 has a gravitational contribution because gravity only works in one direction and we choose it to be in the vertical direction. Notice also that we set the gravitational contribution be negative such that it acts from the

surface of the star towards the center.

Lastly, we rewrite the internal energy equation which is given by equation 3. We separate into x- and y-direction by again calculating the dot product, and using that $\mathbf{u} = (u, w)$ as we did for the continuity equation. We then get that

$$\frac{\partial e}{\partial t} = -\frac{\partial eu}{\partial x} - \frac{\partial ew}{\partial y} - P\left(\frac{\partial u}{\partial x} + \frac{\partial w}{\partial y}\right). \tag{11}$$

C. Numerical schemes

In order to solve the hydrodynamic equations numerically, we need to approximate the derivatives. This is done by finite difference methods. These methods introduce a discontinuity in the differentials and we need to carefully choose the numerical schemes which makes our system the most stable for our simulation to be able to run longer without significant numerical errors.

Because we have two different types of spatial terms, flow-dependent and flow-independent, we implement two different differencing schemes to make our model more stable. The flow dependent terms are the terms where the spatial derivative is taken in the same direction as the flow it is multiplied with (either u or w), while the flow-independent terms are the other ones. For the flow-dependent terms we use a first order upwind scheme which takes the direction of the flow in the current cell into account making these terms more stable. If the flow is negative, we use the backwards differencing method, and if the flow is positive, we use the forward differencing method. For the flow dependent terms in x-direction, the upwind scheme looks like

$$\left[\frac{\partial \phi}{\partial x}\right]^n_{i,j} \approx \begin{cases} \frac{\phi_{i,j} - \phi_{i-1,j}}{\Delta x} & \text{if } u^n_{i,j} \geq 0 \\ \frac{\phi_{i+1,j} - \phi_{i,j}}{\Delta x} & \text{if } u^n_{i,j} < 0 \end{cases} \tag{12}$$

where ϕ represents some variable, n represents the current time step and i, j represents the x- and y-position respectively in the xy-grid. We notice that the type of differencing method depends on the direction of the horizontal flow $u^n_{i,j}$ that the derivative is multiplied with. The scheme looks similar for flow-dependent terms in the y-direction, but the type of differencing method is then determined by the direction of $w^n_{i,j}$ instead. For the flow-independent terms, which are more stable than the flow-dependent terms, we can use the central differencing scheme as approximation which is a bit less stable than the upwind scheme:

$$\left[\frac{\partial \phi}{\partial x}\right]^n_{i,j} \approx \frac{\phi_{i+1,j} - \phi_{i-1,j}}{2\Delta x} \tag{13}$$

The scheme looks similar for the spatial derivatives in y-direction. To approximate the time derivatives we use forward differencing.

$$\left[\frac{\partial \phi}{\partial t}\right]^n_{i,j} = \frac{\phi^{n+1}_{i,j} - \phi^n_{i,j}}{\Delta t} \tag{14}$$

[1]. This is not a very stable scheme, but that is fine as we will only use this expression to find an expression for advancing the primary variables ϕ in time, and calculate the time derivative itself using methods of the next section.

D. Discretization

Now that we know which numerical scheme to use for each term, what remains before being able to solve the governing equations is to discretize the equations and each partial derivative. This is done using the finite difference methods from the previous section. For each of the governing equations we use the chain rule to split up into additional terms of the primary variables. Then we use different numerical schemes to approximate the quantities at each grid point. Using the forward differencing scheme for the time derivative we find an expression for how the primary variables advance in time. Since the discretizations and algorithms for the continuity equation and horizontal momentum equation are already given in the project description, we will only mention the discretizations for the vertical momentum equation and energy equation and algorithms for $w^{n+1}_{i,j}$ and $e^{n+1}_{i,j}$ here.

When discretizing the momentum equation in vertical direction, we let the primary variables be ρ, ρu, ρw and e to simplify, so that when using the chain rule we get

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

which in discritized form is written as

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

The derivatives are discretized as follows, using the central and upwind differencing schemes as explained in the

previous section:

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

The expression for how the vertical velocity advances in time is found from the forward differencing scheme of the time derivative.

$$\left[\frac{\partial \rho w}{\partial t}\right]^n_{i,j} \approx \frac{[\rho w]^{n+1}_{i,j} - [\rho w]^n_{ij}}{\Delta t} \tag{15}$$
$$\Rightarrow w^{n+1}_{ij} = \frac{[\rho w]^n_{i,j} + \left[\frac{\partial \rho w}{\partial t}\right]^n_{i,j} \Delta t}{\rho^{n+1}_{i,j}} \tag{16}$$

For the energy equation, we again use the chain rule to split into additional terms and the internal energy equation then becomes

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

In discretized form this equation can be written as

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

The derivatives are approximated using the central and upwind differencing schemes as follows:

$$\left[\frac{\partial u}{\partial x}\right]^n_{i,j} \approx \frac{u^n_{i+1,j} - u^n_{i-1,j}}{2\Delta x}$$
$$\left[\frac{\partial w}{\partial y}\right]^n_{i,j} \approx \frac{w^n_{i,j+1} - w^n_{i,j-1}}{2\Delta y}$$
$$\left[\frac{\partial e}{\partial x}\right]^n_{i,j} \approx \begin{cases} \frac{e^n_{i,j} - e^n_{i-1,j}}{\Delta x} & \text{if } u^n_{i,j} \geq 0 \\ \frac{e^n_{i+1,j} - e^n_{i,j}}{\Delta x} & \text{if } u^n_{i,j} < 0 \end{cases}$$
$$\left[\frac{\partial e}{\partial y}\right]^n_{i,j} \approx \begin{cases} \frac{e^n_{i,j} - e^n_{i,j-1}}{\Delta y} & \text{if } w^n_{i,j} \geq 0 \\ \frac{e^n_{i,j+1} - e^n_{i,j}}{\Delta y} & \text{if } w^n_{i,j} < 0 \end{cases}$$

Again we use a forward differencing scheme to approximate the derivative with respect to time and find an expression for how the energy advances in time.

$$\left[\frac{\partial e}{\partial t}\right]^n_{i,j} \approx \frac{e^{n+1}_{i,j} - e^n_{i,j}}{\Delta t} \tag{17}$$
$$\Rightarrow e^{n+1}_{i,j} = e^n_{i,j} + \left[\frac{\partial e}{\partial t}\right]^n_{i,j} \Delta t \tag{18}$$

E. The model

Before we explain how the initial conditions and boundary conditions are calculated, we need to know a little bit more about the model we want to make. The model will simulate convection in a rectangular region in the upper layers of the star. This region has dimensions of 12 Mm in the horizontal x-direction and 4 Mm in the vertical y-direction. The top of this rectangular region is located in the photosphere of the star and only covers a very small part of the star.

We split the rectangular volume into cubic cells. There are $N_x = 300$ cells in the x-direction and $N_y = 100$ cells in the y-direction. All cells are identical with a size of $\Delta x \times \Delta y$ where $\Delta x = 12$ Mm/300 and $\Delta y = 4$ Mm/100, making the spatial resolution 0.04 Mm in both x- and y-direction. Both space and time is divided into discrete positions and instants

$$x_i = x_0 + i\Delta x \quad y_i = y_0 + j\Delta y \quad t_n = t_0 + n\Delta t \tag{19}$$

where n represents the current time step and i, j is the x- and y-position respectively. Each cell is then located at $(x, y) = (i\Delta x, \Delta y)$ where $i \in [0, N_x - 1]$ and $j \in [0, N_y - 1]$.

Since we assume the plasma to be an ideal gas, we calculate the internal energy in the model as

$$e = \frac{k_B T \rho}{\mu m_u (\gamma - 1)} \tag{20}$$

in units of J/m^3, where k_B is the Boltzmann constant in J/K, T is the temperature of the gas in K, μ is the mean molecular mass which we approximate to be constant equal to $\mu = 0.61$, m_u is the atomic mass unit in kg, ρ is the density of the gas in kg/m^3 and γ is the adiabatic index for a point particle gas which in our case is $\gamma = (f + 2)/f = 5/3$ where $f = 3$ is the degrees of freedom for the particles in the gas. The equation of state for an ideal gas is then given by

$$P = (\gamma - 1)e. \tag{21}$$

As mentioned, we assume gravitational acceleration to be constant inside the computational box and set the value to be $|\boldsymbol{g}| = GM_\odot/R^2_\odot$. Inserting equation 20 for the

internal energy in equation 21 for pressure and solving for the density ρ, gives the expression

$$\rho = \frac{\mu m_u}{k_B T} P \tag{22}$$

which we will use several times later.

F. Initial conditions

Now we are ready to find the initial conditions for the model. The initial conditions for the temperature T, pressure P, density ρ and energy e are found so that they satisfy the two requirements:

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

The last requirement ensures that our system always transports energy by convection and we set $\nabla = 2/5 + 10^{-4}$. We find the initial condition for temperature by first rewriting the logarithmic gradient as

$$\nabla = \frac{P}{T}\frac{\partial T}{\partial P} = \frac{P}{T}\frac{\partial T}{\partial y}\frac{\partial y}{\partial P}. \tag{23}$$

Since our system is required to be in hydrostatic equilibrium we know that $\partial P/\partial y = -\rho g$ because the compression due to gravity is balanced by the pressure gradient. The pressure gradient can then be expressed as

$$\frac{\partial P}{\partial y} = -\rho g = -\frac{\mu m_u}{k_B T} P g \tag{24}$$

where we used equation 22 for the density ρ. Inserting this into the expression for the logarithmic gradient and solving for the temperature we get

$$\begin{aligned} \nabla &= \frac{P}{T}\frac{\partial T}{\partial y}\left(-\frac{k_B T}{\mu m_u P g}\right) \\ \Rightarrow \frac{\partial T}{\partial y} &= -\nabla \frac{\mu m_u g}{k_B}. \end{aligned}$$

We integrate from the bottom of the box to the top to get the final expression for temperature

$$\int_T^{T_0} \partial T' = -\int_y^{R_0} \frac{\mu m_u g}{k_B} \partial y' \tag{25}$$

$$T(y) = T_0 - (y - R_0)\frac{\mu m_u g}{k_B}\nabla \tag{26}$$

were $T_0 = 5778$ K and $R_0 = 4 \cdot 10^6$ Mm is the temperature and radius at the top of the box, in the photosphere, and T and y is the temperature and radius at the bottom of the box, meaning 4 Mm further into the Sun. We integrate from bottom to top because we set up our computational grid with $(x, y) = (0, 0)$ in the bottom left corner. We then use the equation we found for the temperature to find the initial condition of the pressure. We start by inserting equation 26 for the temperature into equation 24 for the pressure gradient

$$\begin{aligned} \frac{\partial P}{\partial y} &= -\frac{\mu m_u g P}{k_B\left(T_0 + (R_0 - y)\frac{\nabla \mu m_u g}{k_B}\right)} \\ \Rightarrow \frac{\partial P}{P} &= -\frac{\partial y\,(1/\nabla)}{\frac{k_B T_0}{\mu m_u g \nabla} + (R_0 - y)}. \end{aligned}$$

We then substitute $y' = (R_0 - y) + \frac{k_B T_0}{\mu m_u g \nabla}$ and integrate from bottom to top of the computational box, where $P_0 = 1.8 \cdot 10^4$ Pa, which gives us

$$\begin{aligned} \frac{\partial P}{P} = \frac{1}{\nabla}\frac{\partial y'}{y'} \Rightarrow \int_P^{P_0} \frac{\partial P'}{P'} &= \frac{1}{\nabla}\int_{y'}^{y'_0} \frac{\partial y''}{y''} \\ \Rightarrow \ln\frac{P}{P_0} &= \frac{1}{\nabla}\ln\frac{y'}{y'_0}. \end{aligned}$$

We solve this equation for pressure and use the expression for y' to find y'_0 which is the value at the top of the box such that $y = R_0$ which gives $y'_0 = \frac{k_B T_0}{\mu m_u g \nabla}$. The expression for the initial condition for pressure then becomes

$$\frac{P(y)}{P_0} = \left(\frac{y'}{y'_0}\right)^{1/\nabla} = \left(\frac{R_0 - y + \frac{k_B T_0}{\mu m_u g \nabla}}{\frac{k_B T_0}{\mu m_u g \nabla}}\right)^{1/\nabla} \tag{27}$$

$$P(y) = P_0\left(\frac{T}{T_0}\right)^{1/\nabla}. \tag{28}$$

We find the initial condition for energy by rearranging the terms in the equation of state, equation 21, and to find the initial condition for the density we use equation 22. This gives us

$$e(y) = P/(\gamma - 1) \tag{29}$$

$$\rho(y) = \mu m_u P/(k_B T) \tag{30}$$

where we use the expression for T and P that we derived above. The initial condition for velocity is zero everywhere.

G. Boundary conditions

When solving the governing equations numerically, we need to apply boundary conditions since the equations can not be solved at the end points of the numerical grid. The end points are where $i = N_x - 1$, $j = N_y - 1$ and $i = 0, , j = 0$. We therefore need to set values for $\phi^n_{-1,j}$, $\phi^n_{N_x,j}$, $\phi^n_{i,-1}$, ϕ^n_{i,N_y} and the corresponding derivatives.

We start with the horizontal boundaries. To make sure that we stay inside the box when reaching the endpoints and not start calculating values outside of it,

we apply periodic horizontal boundaries. This means that we set $\phi^n_{-1,j} = \phi^n_{N_x-1,j}$ and $\phi^n_{N_x,j} = \phi^n_{0,j}$. We also do this for the derivatives of ϕ.

The vertical boundaries are a bit more complicated to figure out. We will use the 3-point forward difference approximation

$$\left[\frac{\partial \phi}{\partial y}\right]^n_{i,j} \approx \frac{-\phi^n_{i,j+2} + 4\phi^n_{i,j+1} - 3\phi^n_{i,j}}{2\Delta y} \tag{31}$$

for the lower vertical boundary conditions and the 3-point backward difference approximation

$$\left[\frac{\partial \phi}{\partial y}\right]^n_{i,j} \approx \frac{3\phi^n_{i,j} - 4\phi^n_{i,j-1} + \phi^n_{i,j-2}}{2\Delta y} \tag{32}$$

for the upper vertical boundary conditions. These schemes provide more stability as they are schemes of second order [1]. We start with the velocity. We want the vertical component of the velocity to be 0 at both the upper and lower boundary because we do not want the gas to move outside of the box at the top or bottom. This boundary condition is simply described as

$$w^n_{i,-1} = w^n_{i,0} = 0. \tag{33}$$

In addition, we want the vertical gradient of the horizontal component of the velocity to be 0 at the boundary as we do not want the horizontal velocity to be influenced by the surroundings outside of the box. To find an expression for the upper boundary condition for the horizontal velocity component, we use the 3-point backward difference approximation and get

$$\left[\frac{\partial u}{\partial y}\right]^n_{i,-1} = \frac{3u^n_{i,-1} - 4u^n_{i,-2} + u^n_{i,-3}}{2\Delta y} = 0 \tag{34}$$

$$\Rightarrow u^n_{i,-1} = \frac{1}{3}\left(4u^n_{i,-2} - u_{i,-3}\right). \tag{35}$$

For the lower boundary we use the 3-point forward approximation and get

$$\left[\frac{\partial u}{\partial y}\right]^n_{i,0} = \frac{-u^n_{i,2} + 4u^n_{i,1} - 3u^n_{i,0}}{2\Delta y} = 0 \tag{36}$$

$$\Rightarrow u^n_{i,0} = \frac{1}{3}\left(4u^n_{i,1} - u_{i,2}\right). \tag{37}$$

To find the boundary conditions for the density and energy we need to fulfill the requirement of hydrostatic equilibrium. We also need to be careful as the density and energy are coupled from the equation of state, equation 21. First, we find an expression for the derivative of energy in vertical direction:

$$\frac{\partial e}{\partial y} = \frac{1}{\gamma - 1}\frac{\partial P}{\partial y} = -\frac{\rho g}{\gamma - 1} \tag{38}$$

$$= -\frac{\mu m_u P g}{k_B T(\gamma - 1)} = -\frac{\mu m_u g}{k_B T} e \tag{39}$$

where we have used the expression of the pressure gradient from equation 24, then equation 22 for the density and in the last step equation 21 for the pressure. We then use the 3-point backward approximation for the upper boundary together with the expression found above and get

$$\left[\frac{\partial e}{\partial y}\right]^n_{i,-1} = \frac{3e^n_{i,-1} - 4e^n_{i,-2} + e^n_{i,-3}}{2\Delta y} = -\frac{\mu m_u g}{k_B T^n_{i,-1}} e^n_{i,-1}$$

$$\frac{e^n_{i,-2} - 4e^n_{i,-3}}{2\Delta y} = e^n_{i,-1}\left(-\frac{3}{2\Delta y} - \frac{\mu m_u g}{k_B T^n_{i,-1}}\right)$$

$$e^n_{i,-1} = \frac{4e^n_{i-2} - e^n_{i-3}}{3 + \frac{2\Delta y \mu m_u g}{k_B T^n_{i,-1}}}.$$

For the lower boundary we use the 3-point forward approximation and find

$$\left[\frac{\partial e}{\partial y}\right]^n_{i,0} = \frac{-e^n_{i,2} + 4e^n_{i,1} - 3e^n_{i,0}}{2\Delta y} = -\frac{\mu m_u g}{k_B T^n_{i,0}} e^n_{i,0}$$

$$\frac{-e^n_{i,2} + 4e^n_{i,1}}{2\Delta y} = e^n_{i,0}\left(\frac{3}{2\Delta y} - \frac{\mu m_u g}{k_B T^n_{i,0}}\right)$$

$$e^n_{i,0} = \frac{4e^n_{i,1} - e^n_{i,2}}{3 - \frac{2\Delta y \mu m_u g}{k_B T^n_{i,0}}}.$$

To find an expression for the vertical boundary conditions for density we insert the boundary conditions for the internal energy in equation 22 for density, which gives the boundaries

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

$$\rho^n_{i,0} = \frac{\mu m_u(\gamma - 1)}{k_B T^n_{i,0}} e^n_{i,0}.$$

H. Time step

We calculate the optimal length of each time step based on 2 conditions. The first condition is that none of the primary variables $\rho, e, \rho u, \rho w$ get to change by more than a certain percentage. The other condition is that none of the fluid particles should be able to move so fast that they move past a whole grid point when multiplied with the time step. To satisfy the first condition, we calculate the relative change $\Delta\phi/\phi$ per time step Δt for every grid point and for each primary variable ϕ:

$$\text{rel}(\phi) = \frac{\Delta\phi}{\phi} \cdot \frac{1}{\Delta t} = \left|\frac{\partial \phi}{\partial t} \cdot \frac{1}{\phi}\right|. \tag{40}$$

To satisfy the second condition we calculate the relative change in position for both x- and y-direction and for

every grid point:

$$\text{rel}(x) = \left| \frac{\partial x}{\partial t} \cdot \frac{1}{\Delta x} \right| = \left| \frac{u}{\Delta x} \right| \tag{41}$$

$$\text{rel}(y) = \left| \frac{\partial y}{\partial t} \cdot \frac{1}{\Delta y} \right| = \left| \frac{w}{\Delta y} \right|. \tag{42}$$

Next, we find the maximum value of the relative changes for all the primary variables and position. Of the six maximum values we are left with, we choose the largest one again, giving us the largest relative change δ for any quantity at any point in the grid. The length of the time step Δt is then determined by letting the maximum relative change δ during the time step be equal a small constant number p

$$\delta \cdot \Delta t = p \Rightarrow \Delta t = \frac{p}{\delta} \tag{43}$$

which ensures that the 2 conditions are satisfied. This constraint is known as the Courant-Friedrichs-Lewy condition [1]. Notice from the expression above that δ can not be zero, and since the velocity fields might have stationary points we choose to exclude all the points where $\delta = 0$. To not get unnecessarily small time steps we also set a lower boundary for the time step length of $\Delta t = 0.01$ and to avoid too large time step lengths we set an upper boundary for the time step length of $\Delta t = 0.1$. We let $p = 0.1$.

I. Perturbation

After initializing the class to be in hydrostatic equilibrium, we provoke the gas to become convectively unstable. This is done by adding a two-dimensional Gaussian perturbation to the initial temperature distribution. The perturbation we use is given as

$$T = A \exp\left(- \left(\frac{(x - x_0)^2}{2\sigma_x^2} + \frac{(y - y_0)^2}{2\sigma_y^2} \right) \right) \tag{44}$$

where x_0, y_0 represent the central position of the perturbation, A is the amplitude and σ_x, σ_y the standard deviation in x- and y-direction respectively.

We let the temperature perturbation have amplitude $A = 0.9\,T_0$, where T_0 is the solar temperature of the photosphere and place it in the center of the box at $(x_0, y_0) = (6, 2)$ Mm with standard deviations $\sigma_x = \sigma_y = 10^6$ Mm.

J. Simulation/Evolving in time

The algorithm for solving and simulating our system is as following. First we initialize the primary parameters using equation 26 for initializing temperature, equation 28 for initializing pressure, equation 29 for initializing energy and equation 30 for initializing the density. To advance in time we use the `FVis`-module. This module requires a method for advancing the simulation by a time step and updating the variables. We name this method `hydro_solver()` which solves the governing equations described in section II A by implementing everything we have presented so far. The way this method works is that we first calculate the spatial derivatives and use them to calculate the time derivatives of all the primary variables at the current time step as described in section II D. We then update the timestep by calling the method `timestep()` which calculates Δt as described in section II H. Note that when solving the governing equations, we solve them everywhere except for the first and the last vertical indexes of the computational grid as these are the vertical boundaries which we set separately. After updating the time step we advance all the primary parameters in time, from the current time step n to the next time step $n + 1$. We first advance the density

$$\rho_{i,j}^{n+1} = \rho_{i,j}^{n} + \left[\frac{\partial \rho}{\partial t} \right]_{i,j}^{n} \Delta t \tag{45}$$

[1] as the density is needed to advance both the vertical and horizontal velocity in time. We then advance the energy

$$e_{i,j}^{n+1} = e_{i,j}^{n} + \left[\frac{\partial e}{\partial t} \right]_{i,j}^{n} \Delta t. \tag{46}$$

(This expression was derived in section II D.) Using the updated value for density we advance the velocities as following:

$$u_{i,j}^{n+1} = \frac{[\rho u]_{i,j}^{n} + \left[\frac{\partial \rho u}{\partial t} \right]_{i,j}^{n} \Delta t}{\rho_{i,j}^{n+1}} \tag{47}$$

[1] and

$$w_{i,j}^{n+1} = \frac{[\rho w]_{i,j}^{n} + \left[\frac{\partial \rho w}{\partial t} \right]_{i,j}^{n} \Delta t}{\rho_{i,j}^{n+1}}. \tag{48}$$

(The last expression was also derived in section II D.) Before we advance the pressure and temperature in time, we set the boundary conditions by the method `boundary_conditions()`. Since both the pressure and temperature are secondary variables dependent on the primary variables which we have already advanced in time, we need to update the boundary conditions (on the lower and upper boundary) after updating the primary variables. The pressure and temperature is then updated as follows

$$P_{i,j}^{n+1} = (\gamma - 1) e_{i,j}^{n+1} \tag{49}$$

$$T_{i,j}^{n+1} = \frac{P_{i,j}^{n+1} \mu m_u}{\rho_{i,j}^{n+1} k_B} \tag{50}$$

where we used equation 29 to find the expression for pressure and equation 30 to find the expression for temperature.

Now that we have explained every step in the `hydro_solver()` method, we can then use the `FVis`-module to advance in time. We first use the module to make a 60 second long 2D video for each primary variable to see that the system stays in hydrostatic equilibrium. If that is the case, all variables are 0 or close to 0 during the whole video. After confirming this, we add the temperature perturbation to the initial temperature distribution. We add the perturbation in the `Initialise()` method, before the parameters are evolved in time and before the initial density and initial internal energy is calculated. We need to add the perturbation before density is calculated or else the temperature perturbation is overwritten in the next step. We then make a 300 second 2D video of the convective motion showing temperature as a contour plot and the velocity as a quiver plot. We also make a 1D video of the horizontally averaged vertical energy flux to see how much energy is being transported at different vertical positions inside our computational box for the same amount of time. To see how the energy flux evolves over a longer period of time, we also create a 1D plot of the time evolution of the average vertical velocity flux for 1200 seconds.

III. RESULTS

A. Temperature

Snapshots of the temperature changes in the simulation for times 0 sec, 1 min 15 sec, 2 min 50 sec and 4 min can be seen in figure 2, 3, 4 and 5 respectively. Snapshots from the animation of the horizontally averaged vertical energy flux can be seen in figure 6. Each snapshot of the flux is taken at the same time step as the temperature snapshots.

The time evolution of the average vertical energy flux for a longer period of time of 1200 seconds is shown in figure 1.

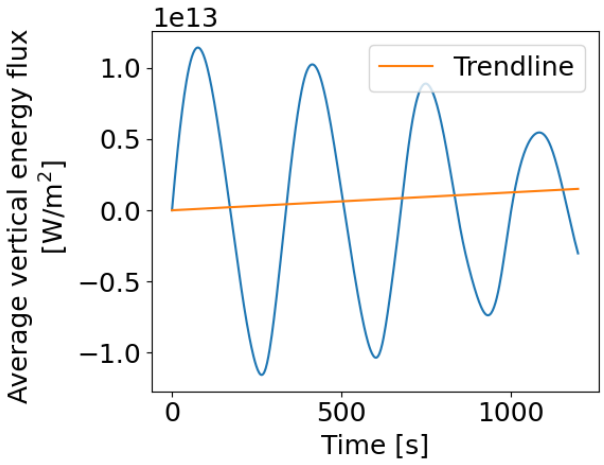

Figure 1. Time evolution of the average vertical energy flux for 1200 seconds. The trendline shows a simple linear trend for the time evolution.

IV. DISCUSSION

A. Hydrostatic equilibrium

Before we implemented the perturbation, we let our simulation run for 60 seconds to verify that our system was in hydrostatic equilibrium. The computational box showed no changes except for the vertical velocity w. The vertical velocity reached a max velocity of 80 m/s when it should have been 0. This is however quite small when comparing to the perturbation simulations where the vertical velocity reaches a value of over 3000 m/s. To check how much this affects the system we also calculated $|\frac{\partial P}{\partial y} - \rho g|$, which should be 0 when the system is in hydrostatic equilibrium as the gravity and pressure gradient balance each other. We got an average value of 1.45, which we decided to be close enough for our purposes, considering that our system is quite unstable, which makes it almost impossible to get it perfectly in hydrostatic equilibrium.

B. System changes in velocity field and temperature

In figure 2 we see the system at 0 s. The temperature perturbation is visible as a warmer region in the middle of the box and we see that the system starts at rest as all the arrows are approximately 0 in size. Since the arrows represents the velocity field, the velocity (u, w) is 0, which it should be in the first time step.

In the second snapshot after 1 min and 45 seconds, shown in figure 3, we see that the gas starts to move upwards as indicated by the velocity field. We see that the arrows point straight up in the middle of the box and increase in length at the upper part of the box. The reason for this behaviour is that the parcel of hot gas has a lower density than the surroundings, and therefore rises to the surface. This results in a positive acceleration upwards shown by the velocity field with arrows increasing in length.

After 2 minutes and 50 seconds, we see from figure 4 that the convection cycle is almost complete. The top of the parcel has reached the surface and cooler gas is starting to fall back down. The velocity field at the top

now points in the horizontal direction due to hot gas rising from below causing the cooler gas at the surface to be pushed away to the sides. Since the cooler gas has a higher density than the hot gas, we see it starting to fall back down.

After 4 minutes the convection cycle is completed as we see in figure 5, although it was probably complete some time before this. We see that the cooler gas falls all the way down as it has a higher density. At the bottom it is warmed back up again and pushed into the middle of the box by the cool gas falling down from above. At the bottom, the gas gets warmed up again, starts rising and thus completing the cycle.

In only 4 minutes, we do not see very prominent changes in the temperature distribution. However, when comparing figure 2 to figure 5 we see that there has been a slight increase of temperature in the middle top part of the box. We could have simulated for a longer period to study the temperature changes further, but since our system is quite unstable, our simulation time is limited and will eventually become unphysical due to numerical errors. Had we used even more stable schemes and decreased the time step length Δt more, we could perhaps have simulated long enough to study the temperature too.

C. System changes in energy flux

The horizontally averaged energy flux in the upper left plot in figure 6 shows that the flux is on average close to 0 initially, before the gas starts to move. This is consistent with what we saw for the velocity field in figure 2 where the arrows are close to 0 meaning no gas, and no energy is being transported.

In the upper right plot of figure 6 we see that the energy flux reaches a maximum value of about $6 \cdot 10^8$ W/m^2. We also see that this value is obtained at a vertical position a bit below the middle. This is consistent with what we see in the temperature plot in figure 3 at the same time step. Almost all the vectors of the velocity field point in the vertical direction causing the energy flux to have a high value here. We also see that the temperature is higher at the bottom part of the box and since energy and temperature is proportional through equation 20 this explains why the maximum in energy flux is shifted down a bit.

In the bottom left plot of figure 6, showing the simulation after 2 minutes and 50 seconds, we see that the energy flux being transported is nearly 0 on average. This can be explained by looking at figure 4 for the time step. As previously mentioned, the convection cycle is almost complete at this time step, but we see that the velocity field transporting gas upwards in the middle is stronger than the velocity field transporting gas downwards at the sides. It therefore seems like the amount of gas being transported upwards is the same as the total amount that is being transported downwards, the averaged energy flux transported will therefore be small.

The bottom right plot of figure 6 shows the average energy flux after 4 minutes. We see that the energy flux is again maximized, but this time at -$6 \cdot 10^8$ W/m^2. When studying figure 5 we see that the velocity field seems to have about the same strength both in the middle and at the sides. However, since the gas falls down on both sides, but is only rises in the middle, more gas is transported downwards than upwards. This explains why the average energy flux again reaches a maximum, but with a negative sign.

If we simulate the convective gas for longer than 4 minutes we assume we will see a pattern where the average energy flux oscillates between positive and negative max values. To see if this was indeed the case we plotted the time evolution of the average vertical flux for 1200 seconds which can be seen in figure 1. We see that the average vertical energy flux does oscillate between maximum values with opposite sign. We also see that the maxima become smaller as time passes. The reason for this is not that the star reduces its flux of energy, but as mentioned before, our system is quite unstable and due to numerical errors we get that the maxima decrease in value.

V. CONCLUSION

To conclude, we have found that by solving the three equations of hydrodynamics numerically and implementing a gaussian temperature perturbation, we were able to simulate convective motions in the upper layer of a star. We found that before we added the perturbation, the system stayed in hydrostatic equilibrium, with only the vertical component of the velocity having a value bigger than zero, but still relatively small. When studying the convective motions we found that the hot gas rises to the top, cools, falls back down and reheats in a circular motion as expected. We also observed that the vertical energy flux oscillates between maximum values of $\pm 6 \cdot 10^8$ W/m^2, but because our system is quite unstable the energy flux decreases over time.

VI. REFLECTION

I personally found this project to be the most fun as it gave me a deeper understanding of the convective zone in a star, building on the knowledge I already have about the convective zone in the Sun from AST2210. I also struggled a lot during the project with trying to

make the vertical velocity component 0 or smaller than 80 m/s for the sanity check, but decided that this would have to do as it is not a very big value, but I would like to investigate further how i could have reduced this even more. I also spent quite some time waiting for the simulation to make videos as I had set the lower boundary for the time step much lower than 0.01, as I eventually set it to. I realize that by increasing this lower boundary the simulation becomes more unstable in a shorter time period, but as the simulation time becomes a lot shorter and the main physical ideas are still visible I found 0.01 to be the best value for the lower boundary. To increase the numerical stability I could have implemented higher order schemes. It could have been fun to study how much this would affect our results.

It would also have been cool to visualize the simulation from above, looking down at the surface to see the granulation pattern form. We see clearly from the simulation that such a pattern would form.

---

[1] B. Gudriksen, *Stellar convection, Term project 3* (Institute of Theoretical Astrophysics, University of Oslo, 2024).

Appendix A: Figures

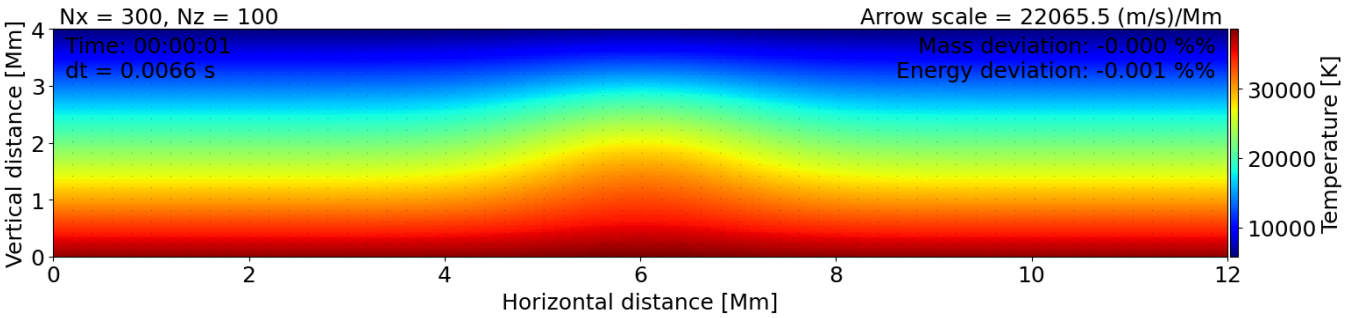

Figure 2. Snapshot of convection simulation after 0 seconds. The figure shows temperature in color and the velocity field is visualized by arrows. At this time step we see the initial perturbation as an increase in temperature in the middle.

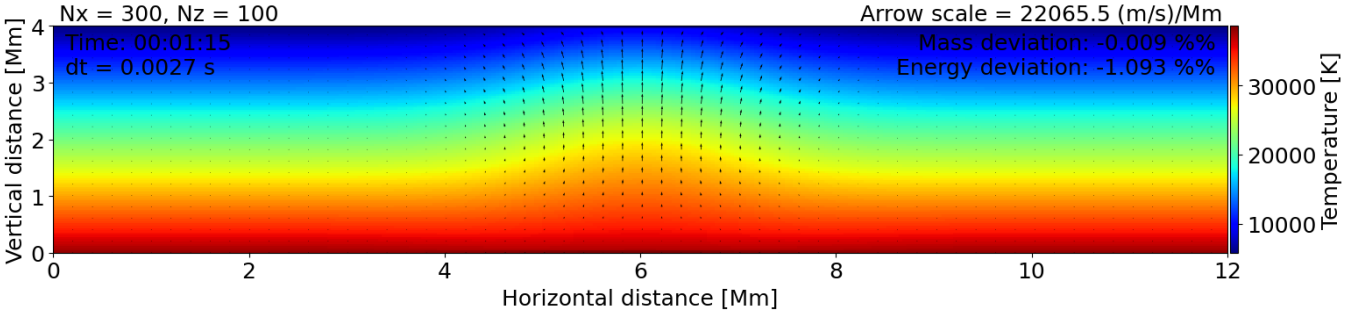

Figure 3. Snapshot of convection simulation after 1 minute and 15 seconds. The figure shows temperature in color and the velocity field is visualized by arrows. We see the hot gas starts to move upwards as indicated by the velocity field.

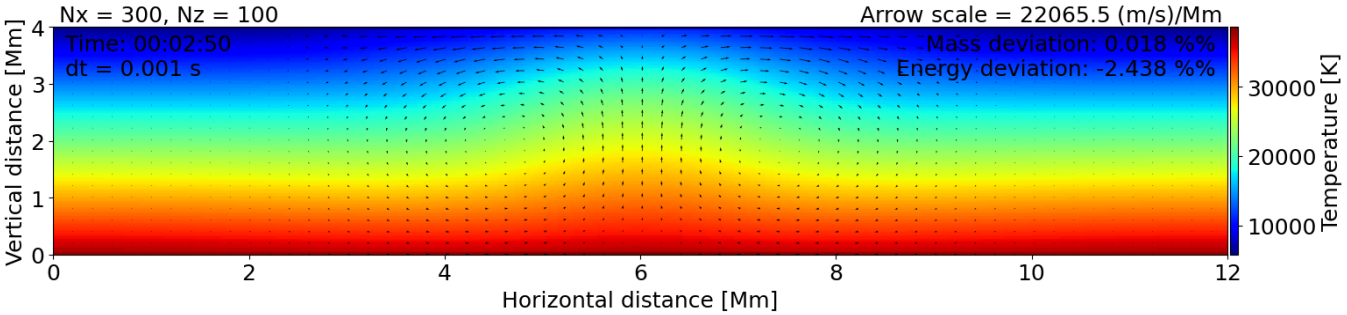

Figure 4. Snapshot of convection simulation after 2 minutes and 50 seconds. The figure shows temperature in color and the velocity field is visualized by arrows. We see the cooler gas starts to fall back down.

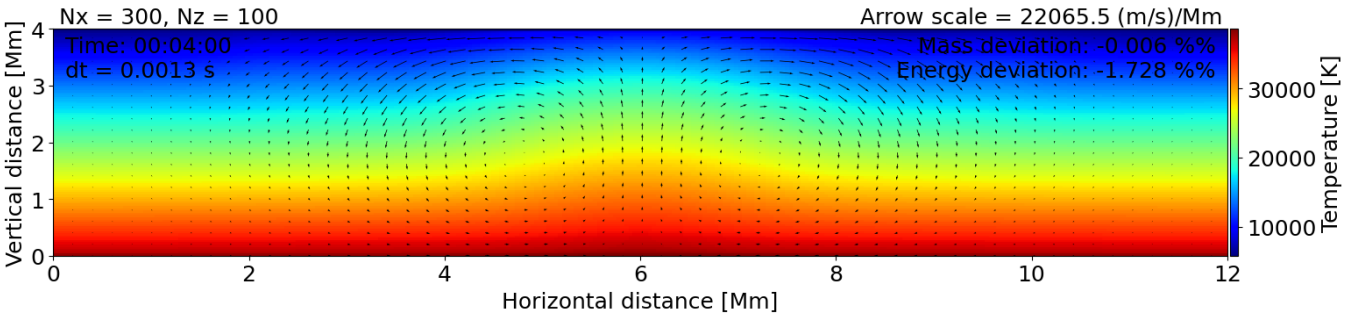

Figure 5. Snapshot of convection simulation after 4 minutes. The figure shows temperature in color and the velocity field is visualized by arrows. The convection cycle is complete, creating a circular motion on both sides of the upstream in the middle.

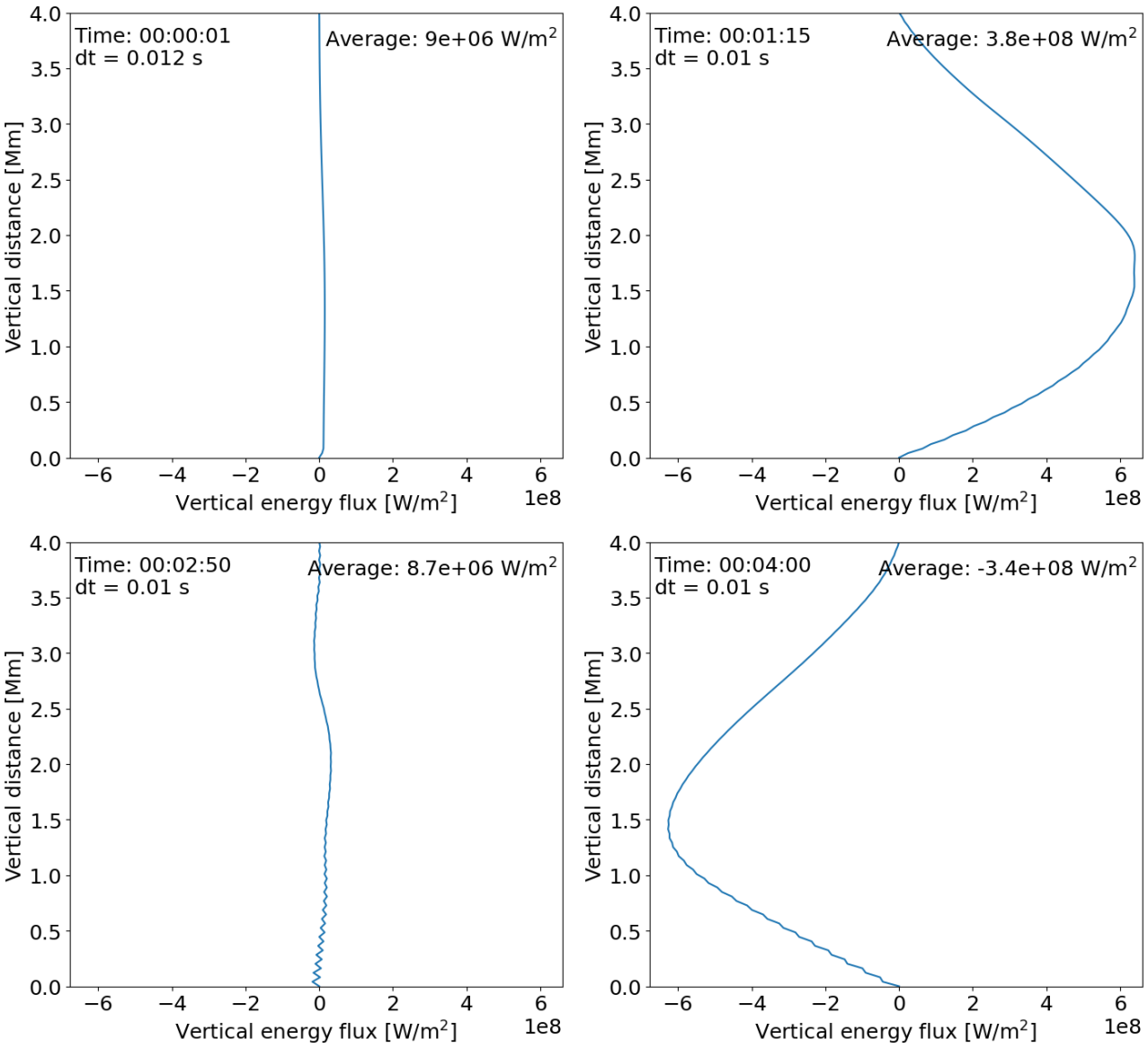

Figure 6. In each plot we see snapshots from an animation of the horizontally averaged vertical energy flux. In the top left plot we see the initial averaged energy flux at time 0 s. In the top right plot we see the average energy flux after 1 minute and 15 seconds. The bottom left plot shows the average energy flux after 2 minutes and 50 seconds and the bottom right plot shows the average energy flux after 4 minutes. We see the energy flux oscillating between maximum value in flux.