Fakultät für Physik und Astronomie

Ruprecht-Karls-Universität Heidelberg

Masterarbeit

Im Studiengang Physik

vorgelegt von

Alberto Botto Poala

geboren in Biella, Italy

2017

Hydrodynamische Simulationen

von

Mischungsprozessen über Convective Grenzflächen

Die Masterarbeit wurde von Alberto Botto Poala

ausgeführt am

Heidelberger Institut für Theoretische Studien

unter der Betreuung von

Prof. Friedrich Röpke

Department of Physics and Astronomy

University of Heidelberg

Master thesis

in Physics

submitted by

Alberto Botto Poala

born in Biella, Italy

2017

Hydrodynamic Simulations

of

Convective Boundary Mixing

This Master thesis has been carried out by Alberto Botto Poala

at the

Heidelberg Institute for Theoretical Studies

under the supervision of

Prof. Friedrich Röpke

Hydrodynamische Simulationen von Mischungsprozessen über Convective Grenzflächen:

Sogar mit den heute verfügbaren Rechenkapazitäten sind 1D Simulationen, das heißt sphärisch symmetrische Modelle, die einzige Möglichkeit die gesamte Lebensdauer eines Sterns zu simulieren. Obwohl diese Herangehensweise für machne Aspekte extrem erfolgreich war, können einige fundamentale Prozesse wie Konvektion und differentielle Rotation nicht aufgelöst werden. In gewissen Phasen der Sternentwicklung spielt Konvektion eine essentielle Rolle, da sie für die Durchmischung der chemischen Elemente und den Energietransport im Inneren eines Sterns verantwortlich ist. Momentan besteht der einzige Weg, die makroskopischen Effekte von Konvektion in 1D Sternentwicklungsprogrammen zu erfassen, in der Verwendung von parametrisierten Modellen, welche vom Aufbau der Schichten des Sterns abhängen. Es würde einen Meilenstein der Sternentwicklung darstellen, wenn man eine konvektive Grenzschicht nur mit Hilfe von Hamilton'schen und hydrodynamischen Variablen charakterisieren und ihr zeitliches Verhalten vorhersehen könnte. In dieser Arbeit wurde eine differentielle Studie des Problems der Durchmischung an konvektiven Grenzschichten anhand von mehrdimensionalen hydrodynamischen Simulationen durchgeführt. Es wurden verschiedene Schichtungen des Fluids sowie konvektive Geschwindigkeiten getestet und die daraus folgende Durchmischung an der Grenzschicht ausgewertet.

Hydrodynamic Simulations of Convective Boundary Mixing:

The only way to simulate the entire lifespan of a star, even with the current computational resources, is through 1D simulations, i. e. by assuming spherical symmetry. Although this approach has been for some aspects extremely successful, fundamental processes as convection and differential rotation are not resolved. Convection plays a leading role in certain evolutionary stages of stars, being one of the processes responsible for chemical mixing and energy transport in the stellar interiors. Currently, the only way to map the macroscopic effects of convection in 1D stellar evolution codes is through parametrized models that are stratification-dependent. A milestone in stellar evolution would be to characterize a general convective boundary by means of Hamiltonian and thermodynamical variables only and predict its behavior over time. In this work a differential study of the convective boundary mixing problem (CBM) was carried out, by means of multidimensional hydrodynamic simulations. Different fluid stratifications and convective velocities were tested, and the consequent boundary mixing were measured.

Contents

Chapter 1

Introduction

Nature is known for having three mechanisms to transport heat.

The first one is through **heat conduction**. Suppose we have two identical bricks of iron at two different temperatures and we could look at the atomic scale, the only difference we would notice is that the atoms of the warmer brick shake more. To say it in a more rigorous way, their mean kinetic energy (with a random velocity) is higher, hence so is the thermal energy. Imagine we put the two bricks in contact, the atoms of the warmer would start to hit and bounce off the atoms of the colder, transferring thermal energy (or heat) from the one to the other, until they reach a situation of equilibrium. This is of course the **Zero Law of Thermodynamics**.

The second mechanism is **radiation**. Every object with a non-zero absolute temperature emits a black body radiation, as long as atomic bounding configurations are complex enough in order to allow the application of the Boltzmann statistics for the energetic distribution of electrons (this is true for objects that have a number of atoms on the order of the Avogadro Number, and even far less). This allows a body to lose thermal energy over time through emission of electromagnetic radiation. This is the mechanism through which Earth and other planets re-emit the radiation received by the Sun and reach thermodynamic equilibrium (it cannot happen through conduction, since this would require that molecules in the atmosphere transfer their thermal energy to molecules or atoms in space, which are too rarefied to allow this).

The third mechanism, which is the topic of this thesis, is **convection**. If we consider a fluid stratification like in the atmosphere in which certain thermodynamic conditions are fulfilled, the medium might become unstable in certain regions and give rise to macroscopic ascending and descending blobs which later decay to smaller scale blobs and finally into turbulence. This enables a very efficient transfer of quantities such as heat and chemical composition through different layers of the fluid stratification.

A *Cumulonimbus*, the typical cloud of thunderstorms, is a perfect example of such a phenomenon. They seldom reach an altitude higher that ten thousand meters because thermodynamical conditions for convection are generally prohibitive in these regions of the atmosphere. Nevertheless when a convective blob hits the stable layer, two phenomena happen. First its turbulent motion erodes mass from it and over time the convective boundary moves upward. Second the hit of the blob imprints an

internal mode in the stable layer. These two are the reasons for which planes avoid flying not only into thunderstorms but also above them, because they expect to find turbulence.

These phenomena happen also in stellar inertia. A convective region over time entrains mass from the stable one, mixing energy and chemical composition (and hence affecting nuclear reaction rates) and imprints internal modes that can be observed on stellar surfaces (the field that studies this phenomenon is asteroseismology, which relies on data of space observatories such as *KEPLER*). It is therefore of fundamental importance to model the dynamics of the convective boundary and the **Convective Boundary Mixing** (CBM) problem, in order to properly understand stellar evolution. This is the ultimate goal of this work.

Chapter 2

Underlying Physics

2.1 Hydrodynamics

As with all the most beautiful and successful laws of Physics, Hydrodynamics is derived from some conserved quantities.

The following derivation of the equations of hydrodynamics, which are the foundation for our simulations, is inspired to the one presented by M. Bartelmann in *Theoretical Astrophysics* (Bartelmann (1994)).

2.1.1 Derivation of the Equations of Ideal Hydrodynamics

Let's consider for instance a monoatomic gas in a box. A fundamental assumption in ideal hydrodynamics is that the mean free path of particles is infinitesimal. Particles might be spatially equally distributed (like in a bottle) or might not (maybe because the box is so big that we get a density stratification like in the atmosphere). The same concept holds for the velocity distribution: it might be Maxwellian or it might not be. In any case we can define the **distribution function in phase space** $f(\vec{x^\mu}, \vec{p^\mu})$ such that

$$N = \int d^4x \; d^4p \; f(x^\mu, p^\mu) \; \delta_D[(p^0)^2 - \mathbf{p}^2 + m^2c^2] \; \Theta(p^0)$$

, where N is the total number of particles, δ_D is the Dirac δ function that selects in the integrations only the hypersurfaces physically allowed by the energy-momentum relation of General Relativity, and finally Θ makes sure that we are taking into account only positive momenta. Keeping this in mind, we can define the first and second momenta of our distribution function

$$J^\alpha = c \int \frac{dp^3}{E} \; f(x^\mu, p^\mu) p^\alpha \qquad\qquad T^{\alpha\beta} = c \int \frac{dp^3}{E} \; f(x^\mu, p^\mu) p^\alpha p^\beta, \tag{2.1.1}$$

namely the **current density** J and the **energy-momentum tensor** T. Carrying out the calculation the components read

$$J = \frac{n(t,\mathbf{x})}{c}\begin{pmatrix} c \\ \langle \dot{x} \rangle \\ \langle \dot{y} \rangle \\ \langle \dot{z} \rangle \end{pmatrix} \qquad T = \rho(t,\mathbf{x})\begin{pmatrix} c^2\langle \gamma \rangle & c\langle \gamma \dot{x} \rangle & c\langle \gamma \dot{y} \rangle & c\langle \gamma \dot{z} \rangle \\ c\langle \gamma \dot{x} \rangle & \langle \gamma \dot{x}^2 \rangle & \langle \gamma \dot{x}\dot{y} \rangle & \langle \gamma \dot{x}\dot{z} \rangle \\ c\langle \gamma \dot{y} \rangle & \langle \gamma \dot{y}\dot{x} \rangle & \langle \gamma \dot{y}^2 \rangle & \langle \gamma \dot{y}\dot{z} \rangle \\ c\langle \gamma \dot{z} \rangle & \langle \gamma \dot{z}\dot{x} \rangle & \langle \gamma \dot{z}\dot{y} \rangle & \langle \gamma \dot{z}^2 \rangle \end{pmatrix}$$

where $n(t,\mathbf{x})$ is the number density, c the speed of light, $\langle a(t,\mathbf{x}) \rangle$ means the average of the quantity a over time and space in a neighborhood of $(t,\mathbf{x})$, γ is the well known relativistic parameter.

By integrating the first and second momentum of Boltzmann Equation, one can show that J and T are divergence free, meaning in the non-relativistic case

$$\frac{\partial}{\partial^\mu} J^\mu = 0 \qquad\qquad \frac{\partial}{\partial^\mu} T^{\mu\nu} = 0 \;\; \forall \nu = 0, .., 3. \tag{2.1.2}$$

This means that if we apply the operator $(c^{-1}\partial_t, \partial_x, \partial_y, \partial_z)$ to J and T we obtain zero. Doing so with the density current we obtain the **continuity equation**

$$\partial_t \rho + \nabla \cdot (\rho \mathbf{v}) = 0. \tag{2.1.3}$$

We then apply the same procedure to the energy-momentum tensor.

We might choose to do it in the first column ($\nu = 0$), and that would lead us to the **energy conservation equation**

$$\partial_t \epsilon + \nabla \cdot (\epsilon \mathbf{v}) + P \nabla \cdot \mathbf{v} = 0, \tag{2.1.4}$$

where ϵ is the internal energy and P the pressure. These two variables, that were not explicitly included in T, arise naturally by splitting up the microscopic velocity of the fluid $\langle \dot{x} \rangle$ into a mean macroscopic velocity $\mathbf{v}$ and a random velocity $\mathbf{u}$ in the neighborhood of $\mathbf{v}$.

$$\langle \dot{x} \rangle = \mathbf{v} + \mathbf{u}$$

, and recalling that for a monoatomic gas

$$\epsilon = \frac{\rho}{2}\langle u^2 \rangle = \frac{3}{2} n k_B T = \frac{P}{2}$$

. When we operate on the other three columns of T, we obtain the **momentum conservation equations** for the three spatial dimensions.

$$\partial_t \mathbf{v} + (\mathbf{v} \cdot \nabla)\mathbf{v} + \frac{\nabla P}{\rho} = 0. \tag{2.1.5}$$

These last three are often called **Euler equations**.

Note that the operator on the left hand side is nothing but the total time derivative

$$\partial_t + \mathbf{v} \cdot \nabla = \frac{d}{dt},$$

so what we are actually writing is Newton's equation per unit volume

$$\rho \mathbf{a} = \nabla P.$$

In case other macroscopic forces like gravity are present, we simply add them on the right hand side as we would do in classical mechanics.

So far we have obtained 5 equations in 6 unknowns, namely the internal energy ϵ, the momentum $\mathbf{p}$, the pressure P and the density ρ. If we want to have at least a chance of integrating the system we're missing one equation, the **equation of state**, that relates the thermodynamic variables.

2.1.2 Viscous Hydrodynamics

As stated at the beginning of the previous section, one fundamental assumption of ideal hydrodynamics is that particles have an infinitesimal mean free path. What actually happens in nature might be very different. Particle have a finite mean free path, and hence they can transport local properties all through the fluid by diffusion: transport of energy generates heat conduction, transport of momentum friction. The consequence is that we end up with one additional term in the energy-momentum tensor representing the diffusive processes

$$T^{ij} \to T^{ij} + T^{ij}_d.$$

The density current remains unchanged (hence continuity equation) because mass is always conserved. As in the previous subsection, the new energy-momentum tensor needs to be operated on with ∇ and set equal to zero in order to obtain our modified equations in the viscous case. We will not carry out all the calculations, rather simply quote the result for the viscous versions (2.1.4) and (2.1.5).

The new energy balance equation reads

$$\partial_t \epsilon + \nabla \cdot (\epsilon \mathbf{v}) + P \nabla \cdot \mathbf{v} = \nabla \cdot (k \vec{\nabla} T) + v_{ij} T^{ij}_d,$$

where v_{ij} is the symmetrized velocity gradient tensor

$$v_{ij} = \frac{1}{2}(\partial_i v_j + \partial_j v_i).$$

This equation shows that the internal energy in a certain region of the fluid can change either if there is a temperature gradient or if the fluid is moving with a velocity field with a non vanishing symmetrized gradient (non solid-body rotation) because of friction. The new momentum conservation equations read

$$\rho \left(\partial_t + \vec{v} \cdot \vec{\nabla} \right) v^j + \partial^j P = \eta \vec{\nabla}^2 v^j + \left(\xi + \frac{\eta}{3} \right) \partial^j \vec{\nabla} \cdot \vec{v}.$$

These are often called **Navier-Stokes Equations**.

2.1.3 Hydrostatic Equilibrium in Stars

We shall now consider the static configuration of 2.1.5 with gravitational potential. It reads

$$\vec{\nabla}P = -\rho\vec{\nabla}\Phi. \tag{2.1.6}$$

This shows that the pressure stratification is adjusted only by the shape of the gravitational field. We can say more by taking the curl of this equation and we get

$$\vec{\nabla}\rho \times \vec{\nabla}\Phi = 0,$$

which shows that the gradient of the gravitational potential and of the density are parallel. Surfaces of constant density are surfaces of constant gravitational potential.

It is known from classical mechanics that if we are given a spherically symmetric matter distribution, the gravitational acceleration is $g = Gm/r^2$ which leads to

$$\frac{\partial P}{\partial r} = -\frac{Gm}{r^2}\rho. \tag{2.1.7}$$

The general approach to find the relation between the pressure and the density consists in taking the divergence of 2.1.6 and substituting the Poisson equation

$$\vec{\nabla} \cdot \left(\frac{\vec{\nabla}P}{\rho}\right) = -4\pi G\rho.$$

This equation can be solved only if an equation of state (EoS) is provided. For instance if we choose a polytropic equation of state (which means that the pressure is a function of the density only and not of the temperature) we obtain the **Lane-Emden Equation** that allows stable solutions for adiabatic indices up to 4/3 only. This EoS is used for instance for modeling white dwarfs, where the pressure doesn't depend on the temperature because of the degeneracy of the gas.

2.2 Energy Transfer

The present chapter about energy transfer and the following one about convection are inspired to the work of Kippenhahn, Weigert and Weiss *Stellar Structure and Evolution* Kippenhahn et al. (2012).

2.2.1 Transport of Energy by Radiation

In the deep interiors of stars energy is generated through nuclear reactions. In the Sun for instance all the energy is provided by burning hydrogen into helium (the so-called p-p chain), while just a little fraction by virialization. Because temperature is of the order of 10^9K, thermal motions allow protons to overcome the electrostatic potential barriers and interact through the nuclear force, giving rise to the reaction

$$6^1\mathrm{H} \rightarrow^4_2 \mathrm{He} + 2^1\mathrm{H} + 2\gamma + 2\nu, \tag{2.2.1}$$

where on the right hand side we have photons and neutrinos, that carry energy away.

Neutrinos, interacting only via the weak force, have a very low cross section and hence leave the core of the sun without scattering, draining energy very efficiently. We need to take into consideration very massive systems like core collapse supernovae and neutron stars in order to see a tangible interaction between the neutrino flux and matter. These astronomic objects are actually excellent neutrino laboratories provided by nature.

Photons instead, interacting electromagnetically, scatter multiple times before reaching the *photosphere*, the region where statistically they scatter for the last time before ending up in our eyes or our telescopes.

Let's now make a rough estimate of the mean free path of a photon in the sun.

$$l_{p,p} = \frac{1}{k\rho}, \tag{2.2.2}$$

where k is the mean cross section (averaged over all frequencies) per unit mass. For ionized hydrogen in stellar medium a rough average is 1 cm^2/g. The average density of the sun is $\bar{\rho}_\odot = 3M_\odot/4\pi R^3_\odot = 1.4\mathrm{g}/\mathrm{cm}^3$.

Plugging in the numbers we get $\bar{l}_{p,p\odot} = 2$ cm.

At this point we need to answer a fundamental question: do two sequential scatterings happen at thermodynamic equilibrium? In other words when one photon scatters, and then scatters again a few millimeters or centimeters away, does it encounter the same thermodynamic conditions? In order to evaluate that, let's consider the mean temperature gradient between the center of the sun and the photosphere

$$\frac{\Delta T}{\Delta r} = \frac{10^7 \text{ K} - 10^4 \text{ K}}{R_\odot} \simeq 1.4 \times 10^{-4} \text{ K cm}^{-1},$$

which means that after 2cm on the radial direction a photon sees a difference in temperature of $\Delta T = l_{p,p} \times 1.4 \times 10^{-4}$ K cm^{-1} $= 3 \times 10^{-4}$ K. The relative difference of radiation energy density can be easily computed since $u \sim T^4$, hence in the center

of the sun $\Delta u/u = 4\Delta T/T \sim 10^{-10}$: between two scatterings a photon is in **Local Thermodynamic Equilibrium** (LTE). This little anisotropy that we neglect is the actual cause of the luminosity of the sun, because it generates a non-vanishing net flux.

2.2.2 Diffusion of radiative energy

We have seen that the mean free path $l_{p,p}$ of photons is much smaller than the characteristic length of our physical system, hence we are allowed to treat the scattering processes statistically. This leads us to describe energy transfer as a diffusion process.

The diffusive flux j of a certain species of particles (number of particles migrating per unit area and unit time) is given by

$$\vec{j} = -D\nabla n, \tag{2.2.3}$$

where n is the species density. D is the diffusion coefficient

$$D = \frac{1}{3} v l_p, \tag{2.2.4}$$

which is a function of the mean free path and of the mean velocity. What we are actually interested in is not the diffusive flux of the particles j, but rather in the diffusive flux of their energy $\vec{F}$. The density needs to be substituted with the energy density $u = aT^4$, $\bar{v}$ with c and l_p with $l_{p,p}$. a is the radiation density constant. If we assume spherical symmetry the gradient of u is simply the radial component

$$\frac{\partial u}{\partial r} = 4aT^3 \frac{\partial T}{\partial r}. \tag{2.2.5}$$

So recalling (2.2.3) we have immediately that

$$F = -\frac{4ac}{3} \frac{T^3}{k\rho} \frac{\partial T}{\partial r}, \tag{2.2.6}$$

which can be written down in a more compact way as

$$\vec{F} = -k_{rad} \nabla T, \tag{2.2.7}$$

after having properly defined k_{rad} as the conduction coefficient. Now we use the well known relation between luminosity and flux and solve it for the temperature gradient

$$\frac{\partial T}{\partial r} = -\frac{3}{16\pi ac} \frac{k\rho l}{r^2 T^3}, \tag{2.2.8}$$

which in Lagrangian coordinate reads

$$\frac{\partial T}{\partial m} = -\frac{3}{64\pi^2 ac} \frac{kl}{r^4 T^3}. \tag{2.2.9}$$

We should never forget the assumptions we made when deriving this equation, namely the LTE. When we approach the photosphere, for instance, the density decreases,

the mean free path increases, and two scattering events happen so far away that LTE doesn't hold any longer. In this case radiation transport is no more a diffusive process.

Another form of heat transfer is conduction, but in stellar physics it becomes relevant only in cores of red giants or white dwarfs and it is fully neglected in our simulations.

2.2.3 The astrophysical ∇

Assuming hydrostatic equilibrium we can now divide 2.2.9 by 2.1.7 in Lagrangian coordinates and obtain

$$\frac{\partial T/\partial m}{\partial P/\partial m} = \frac{3}{16\pi acG}\frac{kl}{mT^3}. \tag{2.2.10}$$

This is nothing but the derivative of the temperature with respect to the pressure, which increasing monotonically upward, is representative of the radius. We furthermore rewrite this quantity as

$$\nabla_{\rm rad} = \left(\frac{d\ln T}{d\ln P}\right)_{\rm rad} = \frac{3}{16\pi acG}\frac{klP}{mT^4}. \tag{2.2.11}$$

This quantity will be of fundamental importance when we will discuss convective stability. Recall that we took into account only radiative transfer, but defining k properly we could take into account also heat conduction.

2.3 Fundamental Equations of Stellar Structure

At this point we are able to write down a system of equations that we will call **fundamental equations of stellar structure**, which describe a simplified but still very useful model. This will hold in a static, stable and spherically symmetric case.

Mass continuity

Here $m(r)$ is the mass contained inside the sphere of radius r

$$\frac{dm}{dr} = 4\pi r^2 \rho. \tag{2.3.1}$$

Hydrostatic equilibrium

We have already encountered the hydrostatic equilibrium equation

$$\frac{dP}{dr} = -\frac{Gm(r)}{r^2}\rho. \tag{2.3.2}$$

Equation of State

So far we have written down two equations in three variables, namely m, ρ, and P. If we want to have at least a chance to solve them we need another equation, namely the equation of state. Possible choices are:

- $\rho = const$ this would be the case for liquids.
- $P \propto \rho^\gamma$ i. e. the density and pressure are not a function of the temperature. This is the case, as already stated in the previous section for white dwarfs and leads to the Lane-Edmen equation and its solutions.
- $P = \frac{\mu}{R}\frac{P}{T}$ the equation of state for perfect gases and this is the case for the sun. It has the inconvenience that it introduces another variable, namely the temperature T, and hence it doesn't solve our original problem. We need at least another equation to solve the system.

Energy conservation

In case we have an energy source (such as nuclear reactions), this needs to be conserved

$$\frac{dL}{dr} = 4\pi r^2 \epsilon. \tag{2.3.3}$$

Temperature gradient

Of crucial importance to understand convection will be the following equation for the astrophysical nabla

$$\frac{dT}{dr} = -\frac{T}{P} Gm\rho\nabla. \tag{2.3.4}$$

2.4 Convection

Until now we have assumed a perfect spherical symmetry, meaning that all dynamical and thermodynamical quantities were a function of the radius only. Obviously this is not a realistic case. For a number of reasons stars have small perturbations, that may eventually grow and give rise to macroscopic phenomena. A classic example is convection.

We are relaxing the spherically symmetric case, but this does not mean that our fundamental equations of stellar structure are useless. As explained in the introduction, if we could section stars we would see that convection appears in concentric regions, hence we can still treat the problem as spherically symmetric, defining dynamic and thermodynamical quantities as averages on a proper region.

We shall now understand when, given thermodynamic variables, we are dealing with a dynamically stable or unstable layer.

2.4.1 Dynamical instability

Let us consider a perfect density stratification like the one in a star or in the atmosphere and let us break the symmetry of the system by adding a little perturbation in the thermodynamic variables. For any given quantity $A(r, \theta)$ (from now on A_e, because it is a function of the mass element we are considering), we compute A_s (which is an average at given r on the surrounding material). We can assume that the fluid moves adiabatically, thus the timescale for heating transfer is much smaller than the timescale for convection turnover.

We define a local property of the fluid

$$DA = A_e - A_s.$$

For instance we could imagine that a little region of a star is slightly hotter than the surroundings, hence we have in that region $DT > 0$. Note that because of the assumption we have made $DP = 0$ always, because when there is a pressure spike, the gas expands at the sound velocity c_s which is way higher than the low hydro regime we observe in stellar inertia.

If we have a $DT > 0$, since in a perfect gas the equation of state reads $\rho \sim P/T$, we obtain $D\rho < 0$. With a lower density, that mass element will be lifted by buoyancy. In a non-adiabatic system it might happen that heat is exchanged so quickly that temperature differences vanish immediately, but we assumed adiabatic processes. The question we are trying to answer is if the mass element, after a little upward movement, will still be buoyant and give rise to macroscopic convection, or if it will bounce back. The answer lies obviously in the temperature gradient, i.e. if once lifted a little bit the new DT will still be lifted by buoyancy it to the next layer, and so on.

Let's approach the problem from another point of view. Let's assume that we have a stable layer without perturbations, and we lift a mass element upward of Δr.

The density difference now is

$$D\rho = \left[\left(\frac{d\rho}{dr}\right)_e - \left(\frac{d\rho}{dr}\right)_s\right]\Delta r, \tag{2.4.1}$$

where the first derivative tells us how much the density of the mass element changes when lifted, the second one tells us how the surrounding density changes along the radial direction.

We denote the buoyancy force per unit volume

$$f_b = -g\ D\rho,$$

which points upward if $D\rho < 0$, which is the unstable configuration. If instead $D\rho > 0$ the mass element sinks back to its original position and no macroscopic motion appears. As a consequence $D\rho < 0$ is our **condition for stability**.

The problem with this criterion is that very often the density gradient is not known, since it does not appear in the fundamental equations for stellar structure. In order to proceed let us turn our gradient in spatial coordinates into a gradient in thermodynamic coordinates. As previously stated, our transformations are adiabatic, hence no exchange of energy occurs. This is very close to reality for stellar inertia. To begin let us write down the parametrizedequation of state $\rho = \rho(P, T, \mu)$ in differential form

$$\frac{d\rho}{\rho} = \alpha\frac{dP}{P} - \delta\frac{dT}{T} + \phi\frac{d\mu}{\mu}, \tag{2.4.2}$$

where $d\mu$ represents the change in the chemical composition, including ionization processes.

Substituting 2.4.1 in 2.4.2 we obtain

$$\left(\frac{\alpha}{P}\frac{dp}{dr}\right)_e - \left(\frac{\delta}{T}\frac{dT}{dr}\right)_e - \left(\frac{\alpha}{P}\frac{dp}{dr}\right)_s + \left(\frac{\delta}{T}\frac{dT}{dr}\right)_s - \left(\frac{\phi}{\mu}\frac{d\mu}{\mu dr}\right)_s > 0. \tag{2.4.3}$$

The two terms containing the pressure gradient cancel because as previously stated $DP = 0$. Let us multiply the remaining terms by the so called **pressure scale height** H_P

$$H_P = -\frac{dr}{d\ln P} = -P\frac{dr}{dP}. \tag{2.4.4}$$

H_P has the dimension of a length, being the characteristic distance over which the pressure changes. We finally obtain our condition for stability

$$\left(\frac{d\ln T}{d\ln P}\right)_s < \left(\frac{d\ln T}{d\ln P}\right)_e + \frac{\phi}{\mu}\left(\frac{d\ln \mu}{d\ln P}\right)_s. \tag{2.4.5}$$

We have already encountered the first term which is ∇_{rad} of 2.2.11. Let's define two new quantities

$$\nabla_e = \left(\frac{d\ln T}{d\ln P}\right)_e \quad \nabla_\mu = \left(\frac{d\ln \mu}{d\ln P}\right)_s. \tag{2.4.6}$$

Recall that since pressure is always decreasing at increasing radius we can measure the temperature as a function of the pressure, and hence its gradient. Furthermore, since our mass element buoys adiabatically, we call $\nabla_e = \nabla_{\rm ad}$. With this new notation we can write down our stability criterion in a more compact way as

$$\nabla_{\rm rad} < \nabla_{\rm ad} + \frac{\phi}{\delta}\nabla_\mu, \tag{2.4.7}$$

and obtain the **Ledoux criterion**.

In our simulations we use a monoatomic ideal gas equation of state, hence $\nabla_\mu = 0$. What we obtain is the **Schwarzschild criterion** for dynamic stability

$$\nabla_{\rm rad} < \nabla_{\rm ad}. \tag{2.4.8}$$

If the left hand side is bigger than the right hand side, it means that our stratification is not able to transport all the energy generated only through radiation, and it is hence required to rely on convection.

Note that up until now we simply have found a stability criterion, but we do not know anything about the convection that will arise. In the next section we will try to quantify it, especially since we need to know how much energy is transported from one layer to the other in order to map this result in 1D simulations of stellar evolution.

2.4.2 Mixing length theory

A fundamental problem in stellar astrophysics is to understand how much energy is transported by convection. Before the era of computational physics, when only analytic solutions were available, the most common tool used to model convective energy transport was the so called **mixing length theory**, introduced more than half a century ago by Böhm-Vitense (1958).

The total energy flux $l/4\pi r^2$ at given radius is given by the radiative flux $F_{\rm rad}$ (which may also include the conductive flux), and the convective flux $F_{\rm con}$. Their sum defines the ∇ that would be necessary to transport away all the energy generated in the center of the star

$$F_{\rm rad} + F_{\rm con} = \frac{4acGT^4m}{3kPr^2}\nabla. \tag{2.4.9}$$

Part of it is ultimately transported by convection, at the most $\nabla_{\rm rad}$ can equal ∇

$$F_{\rm rad} \leq \frac{4acGT^4m}{3kPr^2}\nabla. \tag{2.4.10}$$

In order to quantify then $F_{\rm con}$, let us consider a blob with a positive DT over its surroundings. Its motion will be mainly radial with a velocity v and with $DP = 0$ for reasons already explained. We can write down the energy flux as

$$F_{\rm con} = \rho v c_P DT. \tag{2.4.11}$$

Of course we should average v and DT over an imaginary sphere and hence obtain an equation that holds statistically over the solid angle at a given radius and not locally.

Blobs have an initial velocity that equals zero, and the same holds for DT. Over time it will migrate a distance l_m, namely the *mixing length* before dissolving via turbulent cascade. This means that on average a blob will move a distance $l_m/2$ where it accelerates, and another $l_m/2$ where it decelerates the same amount to stop at the end of the convective zone. Blobs will definitely also change shape during the migration, which makes it pretty difficult to strictly define temperature and velocity, but in a very rough approximation we claim that

$$\frac{DT}{T} = \frac{1}{T}\frac{\partial(DT)}{\partial r}\frac{l_m}{2} = (\nabla - \nabla_e)\frac{l_m}{2}\frac{1}{H_P}, \tag{2.4.12}$$

which is simply a Taylor expansion to determine DT after $l_m/2$. Recall that the buoyancy force per unit mass is $f_b = -g \cdot D\rho/\rho$ and that $D\rho/\rho - \delta DT/T$. Let us pretend on average that half of this force acting on the blob during its migration of l_m, hence the work done is

$$\frac{1}{2}k_r\frac{l_m}{2} = g\delta(\nabla - \nabla_e)\frac{l_m^2}{8H_P}. \tag{2.4.13}$$

Let us furthermore suppose that only half of this work ends up in kinetic energy, because the other half has to go in the work necessary for moving the surroundings. Then the average velocity of an element half way is

$$v^2 = g\delta(\nabla - \nabla_e)\frac{l_m^2}{8H_p}, \tag{2.4.14}$$

and plugging this result into (2.4.11)

$$F_{\text{con}} = \rho c_P T\sqrt{g\delta}\frac{L_m^2}{4\sqrt{2}}H_P^{-3/2}(\nabla - \nabla_e)^{3/2}. \tag{2.4.15}$$

Now let us consider a sphere of diameter d slightly hotter than the surrounding and moving radially with velocity v. It will lose energy because of radiative processes and because of the adiabatic expansion. The energy lost in time due to the first process is

$$\left(\frac{dE}{dt}\right)_{\text{rad}} = \frac{8acT^3}{3k\rho}DT\frac{S}{D}, \tag{2.4.16}$$

hence the drop in temperature per unit length radially, taking into account both processes is

$$\left(\frac{dT}{dr}\right)_e = \left(\frac{dT}{dr}\right)_{\text{rad}} - \frac{(dE/dt)_{\text{rad}}}{\rho V c_P v}. \tag{2.4.17}$$

Which multiplied by H_P/T gives

$$\nabla_e - \nabla_{\text{ad}} = \frac{(dE/dt)_{\text{rad}}H_P}{\rho V c_P v T}. \tag{2.4.18}$$

Plugging in the expression that we derived for $(dE/dt)_{\text{rad}}$ and taking an average for DT, we are left with an equation that contains a "form factor" $l_m S/Vd$ that for a

sphere should of course be $6/l_m$. In the literature $9/2l_m$ it is often used instead and this brings us to

$$\frac{\nabla_e - \nabla_{\rm ad}}{\nabla - \nabla_e} = \frac{6acT^3}{k\rho c_P l_m v}. \tag{2.4.19}$$

We finally obtained five equations(2.4.9, 2.4.10, 2.4.13, 2.4.17, 2.4.19) in five variables ($F_{\rm rad}$, $F_{\rm con}$, v, ∇_e, ∇). Note that the mixing length l_m is not one of those variables, and hence is a free parameter for which we have to assume a value.

Although the mixing length theory has been used extensively in 1D stellar evolution, in many occasions it has been proven unsuccessful, especially because of the uncertainties about the parameters Viallet et al. (2015). Thanks to the rise of computational resources over the last decade, atmospheric scientists and astrophysicists were able to run 2D and 3D hydro simulations and approach the problem more heuristically.

2.4.3 Bulk-Richardson number

So far we got to the point where we have a convective layer and a stable layer above or beneath it. What happens over time is that the convective boundary entrains mass from the stable layer by ingestion, because turbulent eddies capture stable fluid that is consequentially dragged and entrained in the turbulent layer. A fundamental problem in stellar evolution is to understand the dynamics of this boundary: what is its velocity (if we are in an Eulerian workframe), or how much mass from the stable layer is entrained in unit time (in a Lagrangian workframe).

One of the methods proposed to predict the mass entrainment rate is the **bulk-Richardson number** (Meakin & Arnett (2007), Viallet et al. (2015), Cristini et al. (2016)):

$$\mathrm{Ri}_B = \frac{\Delta b L}{\sigma_{\rm T}^2}, \tag{2.4.20}$$

which is a dimensionless parameter developed originally in atmospheric sciences (Richardson (1922), or for a more updated review Stull (1988)) to study the boundary layer problem in the earth atmosphere. It compares the stiffness of the boundary to the strength of the turbulence. L is the length scale of the turbulence (which is of course model dependent), $\sigma_{\rm T}$ is the standard deviation of its velocity, Δb is the buoyancy jump between unstable and stable layer.

In our case we define the length scale of the turbulence L as follows. Let's consider the autocorrelation function

$$a(\Delta r) = \langle f(r) f(r + \Delta r) \rangle, \tag{2.4.21}$$

where the brackets mean average over all possible r, but at fixed correlation length Δr. Recall that such a function always has a maximum for $\Delta r = 0$, (because every function f is perfectly equal to itself), then $a(\Delta r)$ decreases with increasing Δr (because $f(r + \Delta r)$ resembles less and less $f(r)$ as Δr increases). If f is periodic with period T, then the autocorrelation function a has maximum with periodicity T. In order to define L we autocorrelate the Mach number in the horizontal direction of the

convective region and take the length at which a drops to a half of its maximum, i. e. $a(L)/a(0) = 0.5$.

The buoyancy jump is defined by considering the **Brunt-Väisälä frequency**

$$N^2(r) = -g\left(\frac{\partial \ln \rho}{\partial r} - \frac{\partial \ln \rho}{\partial r}\Big|_s\right), \tag{2.4.22}$$

and integrating it over the boundary

$$\Delta b = \int_{r_i}^{r_f} N^2 dr. \tag{2.4.23}$$

Of course we need a proper definition of the boundary limits r_i and r_f. In our case we defined them by initializing a passive scalar in the convective region, which is over time advected in the unstable layer. By looking at the gradient of the concentration of the passive scalar in the vertical direction, one notices two spikes, which represent the boundary positions at given horizontal position. By averaging over all the horizontal positions and by computing the standard deviation, we are able to define the boundary position and its width.

Let us furthermore define the **entrainment coefficient** E as the boundary migration speed $u_{\rm E}$ over the turbulence standard deviation $\sigma_{\rm T}$. Previous works have found that

$$E = ARi_{\rm B}^{-n}. \tag{2.4.24}$$

Where A and n are coefficients that need to be determined, and this will be the primary goal of our simulations.

In a Lagrangian workframe this transforms to

$$\dot{M}_{\rm E} = \frac{\partial M}{\partial r} u_{\rm E}. \tag{2.4.25}$$

If this relation is proven to be true, it would be possible to know the boundary migration speed $u_{\rm E}$ or the mass entrainment rate $\dot{M}_{\rm E}$ given only L, Δb and $\sigma_{\rm T}$, and consequentially map this phenomenon into 1D stellar models. In the hope that these two free parameters exist and are universal (i.e. not system-, code- or resolution-dependent), the goal of this work is to determine them and to test their universality.

In order to do that, we will perform a *differential* study of the entrainment rate. This implies to simulate a convective boundary varying differentially two parameters. The buoyancy jump Δb is a function of the Brunt-Väisälä frequency, which is exactly one of the parameters that we set in our config file to initialize the stratification, hence we can control it very easily. There is no way, on the contrary, to directly control the turbulent Mach number $\sigma_{\rm T}$. As we will explain in the chapter dedicated to the physical setup, we implemented a heating function of which we can control the heating rate. This will help us generate a stronger or weaker turbulence regime when required. Our goal will be then to look for a relation between the entrainment rate and the bulk-Richardson number. Specifically we will investigate if the relation (2.4.24) holds for different values of $Ri_{\rm B}$

Chapter 3

Previous works

The only way to study thoroughly the macroscopic effects of convection like the boundary mixing problem is through multidimensional simulations. Only in the last decade computational power allowed astrophysics groups to run 3D simulations with a satisfying resolution. We report here what we believe are the most significant works carried out in the past.

Turbulent Convection in Stellar Interiors I. Hydrodynamic Simulation. Meakin & Arnett (2007) The first extensive work was carried out by C. Meakin and D. Arnett who simulated the oxygen shell burning and the hydrogen core burning in a $23M_{\odot}$ star both in 2D and in 3D, using the "box in a star" approach. In the core-burning simulation they had only one boundary to analyze, during the shell burning one they had both the upper and lower boundary. This group used the 1D stellar evolution code TYCHO that was then mapped into PROMPI, a multidimensional parallelized hydro code that solves Euler equations implementing PPM (piecewise parabolic method) with a nuclear reaction network. They reported $\log A = 0.027 \pm 0.38$ and $n = 1.05 \pm 0.21$ for the three dimensional case.

As we have seen in the previous section, the definition of the bulk-Richardson number is model dependent. In this case in order to define the boundary and its topology, they injected a passive scalar in the convective region. The biggest drop in its concentration defined the boundary topology. They then computed the mean value of the boundary coordinate and its standard deviation, which delimited the initial and final position of the convective boundary over which the Brunt-Väisäla frequency is integrated in order to obtain the buoyancy jump. They further defined the length scale of the turbulence as the length where the autocorrelation of the Mach number drops below 0.5. This is a simple but effective set of definitions and procedures that I used in my data analysis too.

3D Hydrodynamic Simulations of Carbon Burning in Massive Stars. Cristini et al. (2016) Another recent study has been carried out by A. Cristini et al.. They simulated a carbon burning shell in a $15M_{\odot}$ star (box in a star method) spanning around 3000s and 2×10^9 cm in four runs with different resolutions (from 128^3 to 1024^3). Cristini et al. used the same code PROMPI with self gravity in the Crowling

approximation necessary to describe deep interiors of stars, and they used a very similar procedure to get the entrainment rate coefficient. After analyzing the convective structure with help of the Reynolds-averaged Navier-Stokes equations (RANS), they reported parametric values of $A = 0.06(+0.27/-0.04)$ and $n = 0.81(+0.38/-0.28)$. Although the second value agrees well with the previous study, the first one is not even in the same order of magnitude and affected by huge uncertainty, hence the motivation for this work.

Chapter 4

Code Description

4.1 Code Overview

The code we used is called Seven League Hydro (SLH). It was originally developed by Dr. Fabian Miczek during his PhD thesis (Miczek (2013)). SLH is a finite-volume multidimensional code that solves the Euler equations (ideal hydrodynamic) using explicit or implicit time discretization.

The code automatically decomposes the domain to parallelize the computation on the desired number of processors on distributed memory (MPI) and shared memory enviroments (OpenMP). A wide choice of boundary conditions are implemented.

Furthermore SLH implements a general equation of state, a nuclear reaction network, passive and active scalars which are fundamental for the study of the chemical composition of the fluid, radiation in the diffusion limit and thermal conduction.

Over the years SHL has undergone several scaling tests, the most remarkable of which at the Jülich Supercomputing Center (JSC) in 2016 where it scaled over all the 458752 cores of *JUQUEEN* (a Blue Gene machine, currently the latest generation of IBM supercomputers).

4.2 Finite Volume Discretization

This chapter follows section 3.3 of Miczek (2013).

In order to discretize the physical domain we employ a set of curvilinear coordinates ξ, η, ζ. This set is partitioned by a regular, equidistant grid with $N_\xi \times N_\eta \times N_\zeta$ cells. Without loss of generality, we define the domain extent such that

$$\begin{aligned} \xi &\in [1/2, N_\xi + 1/2] \\ \eta &\in [1/2, N_\eta + 1/2] \\ \zeta &\in [1/2, N_\zeta + 1/2] \end{aligned} \tag{4.2.1}$$

Here we impose that the cell volume is constant, i. e. $\Delta\xi = \Delta\eta = \Delta\zeta = 1$ in order to simplify some calculations. This may, in general, not be the case, since SLH allows the user to call a general curvilinear grid. In this particular with this notation integer values of coordinates refer to the centers of the cells, half integer to cell faces, e. g. by $(\xi, \eta, \zeta) = (i, j + 1/2, k)$ we refer to the interface at the right-hand side of cell (i, j, k).

Both for analytic and numerical study of Euler equations, it is useful to switch to a dimensionless workframe. This is achieved by decomposing state variables into two parts, namely a reference value (denoted with an r) and a dimensionless number (denoted with a hat), e.g. $\rho = \rho_r \cdot \hat{\rho}$. This leads to equations for the state vector

$$\frac{\partial \hat{\mathbf{U}}}{\partial \hat{t}} + \frac{\partial \hat{\mathbf{F}}_x}{\partial \hat{x}} + \frac{\partial \hat{\mathbf{F}}_y}{\partial \hat{y}} + \frac{\partial \hat{\mathbf{F}}_z}{\partial \hat{z}} = \hat{\mathbf{S}}, \tag{4.2.2}$$

where the quantities $\hat{\mathbf{U}}$, $\hat{\mathbf{F}}_x$, $\hat{\mathbf{F}}_y$ and $\hat{\mathbf{F}}_z$ are defined by the stress-energy tensor as discussed in the previous chapter. From now on for simplicity we will drop the hat notation. We discretize the spatial component of in space Euler equations as

$$J^{-1}\frac{\partial \mathbf{U}}{\partial t} + \nabla_{\xi,\eta,\zeta} \cdot \mathbf{F} = J^{-1}\mathbf{S}, \tag{4.2.3}$$

where J is the determinant of the Jacobian of the transformation from the physical domain to the computational grid.

These equations are integrated over the volume $\Omega_{i,j,k} = \Delta\xi \times \Delta\eta \times \Delta\zeta$ of the cell i, j, k, which in our assumptions is constant for every cell (differentially $d\Omega = d\xi \times d\eta \times d\zeta$) of the cell (i, j, k). The integrals read

$$\int_{\Omega_{i,j,k}} J^{-1}\frac{\partial \mathbf{U}}{\partial t} d\Omega + \int_{\Omega_{i,j,k}} \nabla_{\xi\eta\zeta} \cdot \mathbf{F} d\Omega = \int_{\Omega_{i,j,k}} J^{-1}\mathbf{S} d\Omega. \tag{4.2.4}$$

By exchanging the derivatives with the integrals, it is clear that these are nothing but the averages of the conserved quantities in the cell times the volume of it. We therefore introduce some new cell-averaged quantities such as

$$\mathbf{U}_{i,j,k} = \int_{\Omega_{i,j,k}} J^{-1}\mathbf{U} d\Omega \qquad \mathbf{S}_{i,j,k} = \int_{\Omega_{i,j,k}} J^{-1}\mathbf{S} d\Omega. \tag{4.2.5}$$

In addition to this, we can apply Gauss theorem on the second term on the left-hand side and obtain the following expression

$$\frac{\partial \mathbf{U}_{i,j,k}}{\partial t} + \frac{1}{V_{i,j,k}} \oint_{\partial\Omega} \mathbf{F} \cdot d\mathbf{s} = \mathbf{S}_{i,j,k}. \tag{4.2.6}$$

If we suppose to know the integral fluxes through the six interfaces, we can further decompose the second term into six numerical fluxes as follows

$$\begin{aligned} \mathbf{F}_{i+1/2,j,k} &= \oint_{\partial\Omega_{(i+1/2,j,k)}} \mathbf{F}_\xi d\xi \\ \mathbf{F}_{i,j+1/2,k} &= \oint_{\partial\Omega_{(i,j+1/2,k)}} \mathbf{F}_\eta d\eta \\ \mathbf{F}_{i,j,k+1/2} &= \oint_{\partial\Omega_{(i,j,k+1/2)}} \mathbf{F}_\zeta d\zeta \end{aligned} \tag{4.2.7}$$

This allows us to write down the discretized Euler equations as

$$\begin{aligned} \frac{\partial \mathbf{U}_{i,j,k}}{\partial t} + \frac{1}{V_{i,j,k}} (&\mathbf{F}_{i+1/2,j,k} + \mathbf{F}_{i-1/2,j,k} + \mathbf{F}_{i,j+1/2,k} + \\ &\mathbf{F}_{i,j-1/2,k} + \mathbf{F}_{i,j,k+1/2} + \mathbf{F}_{i,j,k-1/2}) = \mathbf{S}_{i,j,k}. \end{aligned} \tag{4.2.8}$$

What is stored in the memory are the cell-averaged quantities. Their value changes only by numerical fluxes from the neighbor cells or by source terms. This implies that if there are no source terms, the sum of cell-averaged variables cannot change (except for fluxes across the boundaries). In this way, the finite volume discretization perfectly reflects the conservative nature of hydrodynamic. We would like to emphasize that the numerical fluxes still remain unknown, and should be reconstructed for the cell (i, j, k) from the surrounding interfaces.

In addition to that, what we performed was a *partial discretization*, since we still have a continuous time derivative and no finite difference. This method, through which we discretize first the spatial dimension and then separately the temporal one, is known as the **method of lines**. It is a very effective method to reduce partial differential equations to a set of coupled ordinary differential equations such as

$$\frac{\partial \mathbf{U}_{i,j,k}}{\partial t} + \mathbf{R}_{i,j,k} = 0, \tag{4.2.9}$$

where the vector $\mathbf{R}_{i,j,k}$ is known as *spatial residual* of the cell (i, j, k). As previously mentioned, this is a function with the numerical fluxes of the surrounding cells, and suitable numerical methods are needed in order to correctly compute them.

4.3 Explicit vs. Implicit Time Stepping

This follows sections 5.1 and 5.2 of Miczek (2013).

The simplest way to discretize in time equation 4.2.9 is the so called **forward Euler method** [1]:

$$\frac{\mathbf{U}_{i,j,k}^{n+1} - \mathbf{U}_{i,j,k}^{n}}{\Delta t} + \mathbf{R}_{i,j,k}(\mathbf{U}^n) = 0. \tag{4.3.1}$$

This method simply approximates the time derivative with a finite difference at each grid point, and a spacial residual is evaluated at the old time step. Notice that equation 4.3.1 can be solved analytically for the conserved quantity $\mathbf{U}_{i,j,k}^{n+1}$. This means that we are dealing with an *explicit time stepping*. Although equation 4.3.1 has a very direct and simple implementation, this method is accurate to first order, i.e. it has an approximation error that is $\mathcal{O}(\Delta t)$.

A considerable amount of higher-order explicit time stepping schemes are available. In SLH the a third-order Runge-Kutta scheme (RK3) developed by Shu and Osher (1988) is implemented.

There are essentially two advantages of an explicit time stepping method: they are easy to implement and they require little resources memory-wise on the hardware. The biggest disadvantage is that one always has to choose the time step lower than a maximum value δt. In case this condition is not fulfilled, numerical errors start to grow over time and dominate over the physical solution. In the case of Euler equations, in order to determine δt, the CFL criterion is generally used:

$$\Delta t = \frac{\mathrm{CFL}}{N_{\mathrm{dim}}} \min\left(\frac{\Delta x}{|q_n| + c/M}\right). \tag{4.3.2}$$

where N_{dim} is the number of dimensions, Δx is the cell width, the quantity in denominator is the wave velocity and the minimum has to be evaluated over all the computational grid. At last CFL is a dimensionless coefficient that determines how many cells a wave can travel for time step. For all the explicit time steppers implemented in SLH, CFL is always smaller than unity. One can see that, roughly, $\Delta t \propto M$, hence the lower the mach number, the lower the maximum allowed time step. This criterion shows why the use of explicit time steppers for low mach flow is computationally impracticable. For this reason we rely on a new discretization method, namely the **backward Euler method**. According to this discretization the conserved quantities at the time step $n + 1$ are given by

$$\frac{\mathbf{U}_{i,j,k}^{n+1} - \mathbf{U}_{i,j,k}^{n}}{\Delta t} + \mathbf{R}_{i,j,k}(\mathbf{U}^{n+1}) = 0. \tag{4.3.3}$$

The only difference compared to the forward Euler method is that the residual is evaluated at the time step $n+1$ instead of n. An analytic solution for this equations is, in general, not possible, and sophisticated numerical solvers need to be employed

[1] The forward Euler method can be applied to resolve general PDEs, not only Euler equations. It is a coincidence that both the fluid dynamics equations and the numerical scheme that we use to solve them were discovered by Swiss mathematician Leonhard Euler.

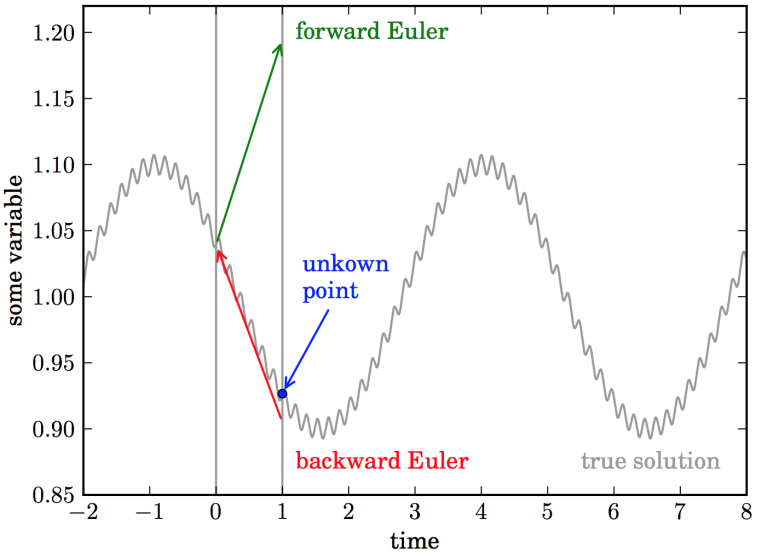


Figure 4.1: Conceptual difference between explicit and implicit integration. Picture from Miczek (2013).

in order to find an approximate solution (in our case we implement the so-called **Newton-Raphson method**, see next section). The advantage of integrating implicitly is very well explained by figure 4.1. What is shown is a slow advection wave with a relatively low frequency, with a high frequency acoustic wave on it. Suppose we start integrating from $t = 0$ explicitly with the forward Euler method. Since the derivative is evaluated at $t = 0$, the risk is that the acoustic wave completely dominates the result, leading to a corrupted value for the conserved quantity. Integrating implicitly instead, the derivative is evaluated over the entire time step. In other words we extrapolate the derivative starting from the final point backward to the initial one, ensuring that the finite difference method resolves the phenomena at the time scale we are interested in. Acoustic waves in the low mach regime in fact decouple from the advective solution, hence we don't need to resolve them. The backward Euler method again generates an error which is $\mathcal{O}(\Delta t)$, but more refined methods are available in the literature and are implemented in SLH.

The question we need to answer when choosing the integration technique is if the rise in efficiency given by implicit methods compensate the loss, given by the expensive numerical solvers needed to approximate the solution of the implicit equation. In the low mach regime (from $M \simeq 10^{-4}$ to 10^{-3}) an implicit integration with SLH is always more convenient.

4.4 Newton-Raphson Method

This chapter follows section 5.4 of Miczek (2013). In this section we analyze the so-called **Newton-Raphson method**, which is the iterative solver implemented in SLH in order to find an approximate solution to the implicit equation 4.3.3.

The first step we need to take is to move every term of our discretized equation to the left-hand side and define a quantity $\mathbf{D}_{i,j,k}(\mathbf{U}^{n+1})$ called the *defect*:

$$\mathbf{D}_{i,j,k}(\mathbf{U}^{n+1}) = \mathbf{R}_{i,j,k}(\mathbf{U}^{n+1}) + \frac{\mathbf{U}^{n+1}_{i,j,k}}{\Delta t} - \frac{\mathbf{U}^{n}_{i,j,k}}{\Delta t} = 0. \tag{4.4.1}$$

Obviously the equations are solved if the defect is zero. This algorithm starts from an initial guess $\mathbf{U}^0$, which is iteratively refined by calling

$$\mathbf{U}^{k+1} = \mathbf{U}^k - \left(\lambda \frac{\partial \mathbf{D}^k}{\partial \mathbf{U}}\right)^{-1} \mathbf{D}^k. \tag{4.4.2}$$

Where λ is a convergence parameter that in generally set to one (in pathological cases it might be set $\lambda > 1$ to dump the convergence ratio). The more the iterations, the closer $\mathbf{U}^k$ gets to the root of $\mathbf{D}$.

In order to compute the right-hand side, one should invert the Jacobian of the defect, but this is computationally and memory-wise very expensive. It is instead easier to rewrite 4.4.2 as

$$\lambda \frac{\partial \mathbf{D}^k}{\partial \mathbf{U}} \Delta \mathbf{U} = -\mathbf{D}^k, \tag{4.4.3}$$

where $\Delta \mathbf{U} = \mathbf{U}^{k+1} - \mathbf{U}^k$. This is a very effective method to transform a system of non-linear equations to a set of systems of linear equations.

The Newton-Raphson method generally converges quadratically to the solution, i.e. the number of significant digits doubles at every iteration. In particular cases like a double root (namely $f'(\alpha)=0$, where α is the defect root) the convergence is linear. Obviously the first and second derivative should be non-zero over the iterative domain and to improve the efficiency the initial guess should be as close as possible to the root of the defect. In SLH we choose as $\mathbf{U}^0$ the solution of the previous time step.

SLH prints in the log file information about every iteration of the Newton-Raphson method for every variable. Specifically it prints the L2 norm of the defect. For the density ρ for instance

$$||\mathbf{D}||^\rho = \sqrt{\sum_{i,j,k} \left(\mathbf{D}^\rho_{i,j,k}\right)^2}, \tag{4.4.4}$$

which is the quantity that decreases quadratically.

In equation 4.4.1 the very last term is a constant vector, since it's the state vector at the previous time step ($t = n$). Only the state vector at the current time step ($t = n + 1$) changes at every iteration of the Newton-Raphson method. These sums are of course cause of numerical cancellation errors. In SLH the L2 norm of the

defect is scaled by the L2 norm of the constant vector, because different variables have numerically different orders of magnitude.

$$\frac{||\mathbf{D}||^{\rho}}{||\mathbf{U}^{n}/\Delta t||^{\rho}}. \tag{4.4.5}$$

This is done in order to monitor the convergence in a dimensionless way and relative to machine precision [2] Note that reaching machine precision when converging is actually rarely possible because of

- round-off errors in the discretization in the flux functions and in the linear solvers
- approximations in the defect Jacobian and in the linear systems

Moreover, the process of discretization of Euler equations generates errors which depend on the numerical scheme called and the grid chosen. As a consequence, converging to machine precision is often a waste of computational resources.

[2]Recall that for double precision floating point value, machine precision is $\simeq 10^{-16}$.

Chapter 5

Hydrodynamic Simulations

5.1 2D Simulations

The physics of 2D hydrodynamics is qualitatively very similar but quantitatively very different to the 3D case. Concepts such as hydrostatic equilibrium or the Schwarzschild criterion can be applied in 2D as well as in 3D, but the resulting flows will have different properties. Turbulence for instance in the 2D case transfers kinetic energy from smaller eddies to larger eddies, while in the 3D case the opposite happens (for an introduction in the subject see Boffetta & Ecke (2012)). We expect hence the entrainment mass ratio to be different between the 2D and 3D cases.

Nevertheless, since 2D simulations are computationally so cheap, they are a very useful tool to familiarize with the code and to run tests in order to setup bigger and more expensive 3D runs. In our case we needed a rough estimate of the entrainment rate, of the boundary migration velocity and of the convective turnover timescale. Specifically, it is necessary to make sure that convection reaches a stable regime over a fairly large time domain (in our case at least 5 convective turnover times), to correctly collect the data that characterizes the boundaries.

5.1.1 Simulations setup

The physical setup used for our simulations is the so called "box in a star" method, meaning that we simulate some relatively very small internal region of a star.

In our case for the 2D runs it will be a box of 2.50×10^9 cm on the x axis and 1.25×10^9 cm on the y axis. Gravity is constantly pointing downward on the y axis with a magnitude of 10^3 cm/s^2. This generates a pressure stratification in the fluid that covers approximately three pressure scale heights.

The controlling parameter to initialize the fluid stratification is the temperature gradient. In our case we divided the simulated region in three parts. The bottom region (labeled as 1) starts at the lower boundary and reaches 1/3 of the domain, the central (labeled as 2) proceeds until the middle, and the upper one (labeled as 3) reaches the upper boundary, as shown in Figure (5.1). Convection will be generated in the central region. We chose to initialize a larger stable layer above rather than

a symmetric setup because, as the first tests showed, the upper boundary is always softer and hence entrains more material. This is in agreement with what found by Meakin & Arnett (2007). We define a parameter α_i ($i = 1, 2, 3$) which is nothing but the fraction of the ∇ over the $\nabla_{\rm ad}$. As seen in previous sections $\alpha_i < 1$ implies stability in the region i, instability otherwise.

We always initialized the setup with $\alpha_1 = \alpha_3$ much smaller than 1; and $\alpha_2 = 0.99$, which means a very precarious situation in terms of stability in the second region.

A heating function furthermore heats the second region with a gaussian profile (see Figure 5.1) to stimulate convection. Without it the system would never become convective in the first place, since hydrostatic equilibrium is fulfilled everywhere. Furthermore the heating function keeps the system convectively unstable over time. In fact without the constant energy generation because of heat mixing and numerical viscosity, the system would soon reach again a stable configuration of hydrostatic equilibrium. The heating mimics the energy generation in deep stellar interiors due to nuclear burning. The choice of a Gaussian profile is a common practice in stellar physics. On the one hand we require that the heating is peaked deep inside the convective region and that it does not influence the stable layers, on the other hand it is preferable not to use functions with discontinuity, because this might generate shock waves that propagate into the system.

The values of α_1 and α_3 are controlling parameters for the bulk-Richardson number, since they are proportional to Δb. The advantage of simulating a strip of convection between two stable layers (miming a Shell convection) instead of a convective region at the bottom that grows upward (core convection-like) is twofold. It gives us two convective boundaries to study per run instead of one, and we avoid to insert a wall boundary (see next paragraphs) in contact with convective motions, which is a very unphysical setup.

One of the hardest tasks has been to generate the correct amount of heating such that the turbulence standard deviation *at the interface* was constant during the run (which is one of the ingredients of the bulk-Richardson number, hence one of the parameters we want to control in order to perform a differential study). In order to do that, the heating rate decreases over time in an hyperbolic fashion such that, at $t = 10^5$ s, the total heat generation is reduced to 50% of the original amount.

At the bottom of the convective region small perturbations in temperature and Mach number had been imprinted on the stratification in order to break the initial symmetry of the system.

A wide range of boundary conditions have been tested for this problem. For the vertical direction we used wall boundaries. As previously stated, in our first setup we used to stimulate convection at the bottom of the simulated region, but implementing periodic boundaries in the horizontal direction (that definitely provide the most physical situation) a shear flow appeared in both the stable and convective region at the onset of convection. We then tested reflective boundaries and wall boundaries. In the first case we observed a mass loss (very likely because of a bug in the code). In the second case after a few tens of minutes of the onset of convection, a spike in the Mach number at the boundary appeared, and the code crashed. Interestingly enough, the spike appeared always at the left boundary. For this reason we initially suspected

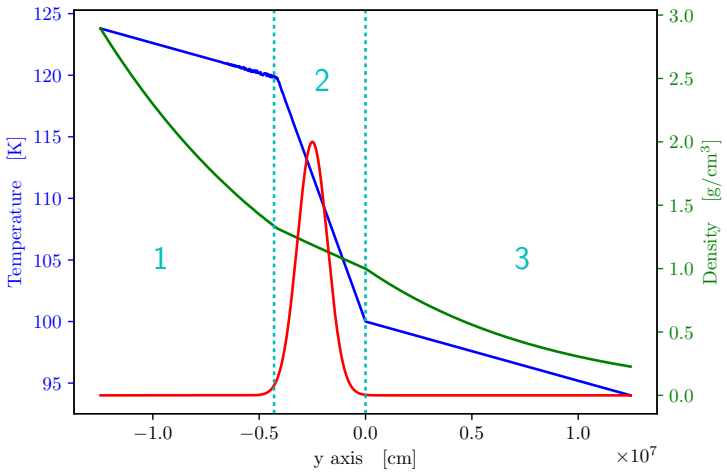


Figure 5.1: Example of initial temperature (blue) and pressure (green) profile along the y-axis. The red curve represents the shape of the heating function in arbitrary unit. Notice the little perturbations in temperature between the first and second region (see text).

another bug in the code. What actually used to happen is that some fluid stagnated right above the convective boundary (specifically the first column of cells on the left) and over time a massive temperature gradient was established. We suppose that this computational artifact appeared only on one side of the grid because the iterative linear solvers implemented in SLH are not perfectly symmetric. This is generally not a source of problems unless in specific pathologic cases as the one we had to face. We then tested a new setup, where convection was stimulated in the central region of the domain. By implementing periodic boundaries the same shear flow appeared, but only after a few tens of hours of simulated time, which allowed us to collect enough data for the analysis. As we will explain this is not a perfectly physical solution, since internal modes can bounce on them and be reflected.

As briefly explained in the previous sections, we initialized two different passive scalars inside the regions one and three. Recall that passive scalars are just like dyes in the fluid which in no way influence the dynamics of the system, nor do they diffuse. The passive scalar one and three (initialized in the regions 1 and 3) will be used to compute the entrained mass, i. e. for every time step we integrate the passive scalar densities in the convective region and obtain the total entrained mass at each boundary. The two passive scalars are also used in order to define the boundary topology (i. e. position and width), by looking at the biggest drop in passive scalar density.

As previously stated our goal is to perform a *differential* study of the bulk-Richardson number and the CBM problem. This implies running simulations with different values of Δb and σ_{T} at different resolutions. Because 2D simulations are computationally so cheap, we managed to run a sufficient amount of them in a short time. With the code 2d0.10-100 we will refer to a 2-dimensional run with $\alpha_1 = \alpha_3 = 0.1$

and 100% of the heating referring to the fiducial value of 3.6×10^{15} erg/s.

We run in 2D on a 2048×1024 uniform Cartesian grid. It is worth remarking that when doing CFD with a higher resolution, one not only better resolves the features of the system, but also decreases the numerical viscosity (increases the Reynolds number). This is the reason why we chose this grid setup: we want to keep cells squared and keep viscosity a scalar quantity, and prevent it from becoming a tensorial one. In the next section we will perform a convergence study, in order to understand which roles the resolution and the viscosity play in the phenomenon of convective entrainment.

We performed the runs on our local cluster at HITS. Every job required about 4.5×10^4 cpu hours on bridge architecture, and roughly the half of it on Haswell architecture. We saved one output file every 100 time steps, in order to both save memory and to avoid wasting too much computing time in I/O processing.

5.1.2 Evolution of a 2D run

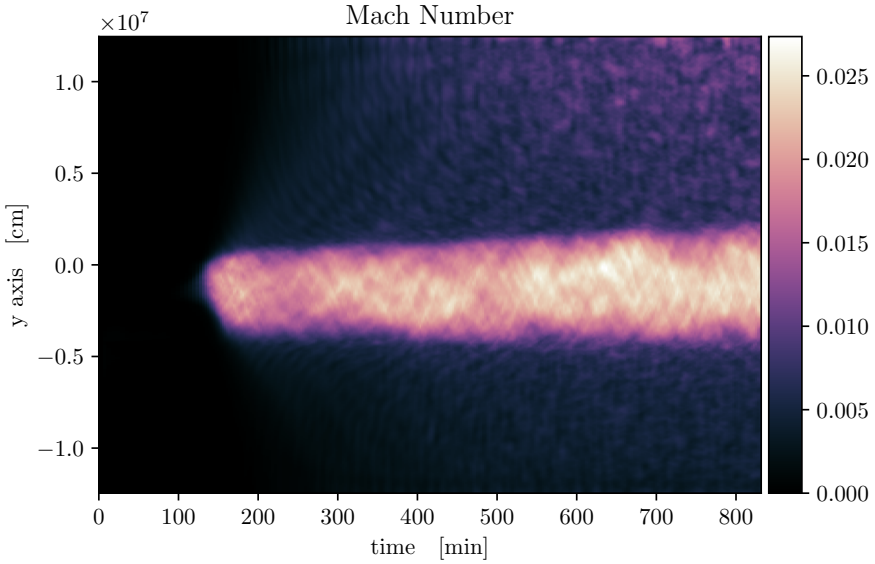
Let us consider the setup 2d0.1-100.

Convection starts at $t \simeq 150$ min. Because of the already mentioned implicit time stepping, the lower the Mach number, the bigger the time step, allowing us to save a huge amount of computational resources before the rise of convection. A remarkable difference in computational efficiency compared to an explicit time stepping was also observed during the convective regime, as long as the Mach number was below 10%, which has always been our case.

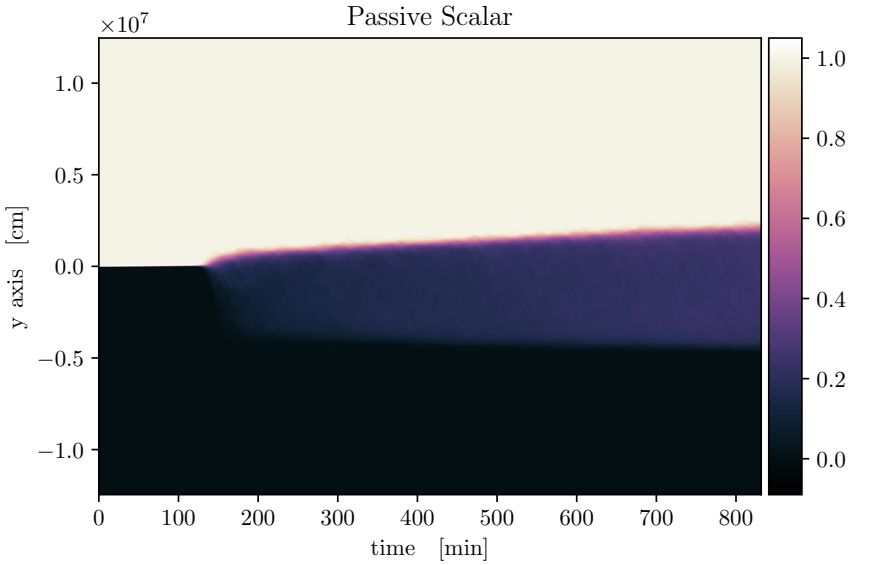
In Figure (**??** we plot the profile of the Mach number over the simulated time. Convection starts at around $t = 150$min in the central region (region 2), while the upper and lower region remain stable. Some time is needed for convection to develop, because the fluid needs to be heated to an unstable configuration. One can also observe that the convective region expands over time, i. e. the upper boundary moves upward and the lower boundary downward. Two remarks need to be made.

First of all the Mach number is stable over time, thanks to the heating function that we implemented, featuring a decrease of heat generation as previously explained. Second it is clear that some internal modes are excited by the convective blobs when they hit the stable layers and they propagate through them. They appear more significant in the upper region and to a certain extent this is true, but mainly this is due to the fact that the speed of sound there is lower. The difference is not so dramatic when plotting the absolute velocity. Two interesting questions remain without answer. First of all it is impossible to tell to which extent the dynamic of the boundary is influenced by these modes. Second of all it is possible that the chemical mixing due to these modes in the stable stratification might have a significant impact on the evolution of a star, which is obviously not considered in 1D simulations. We believe that further investigations of this hypothesis are needed.

In Figure (5.2b) we plot the upper passive scalar (initialized in the region 3), which over time is entrained by convection. We clearly see the motion of the boundaries that over the 800 min of simulated time move gradually (and roughly linearly in time) upward and downward. As previously mentioned the first passive scalar was initialized



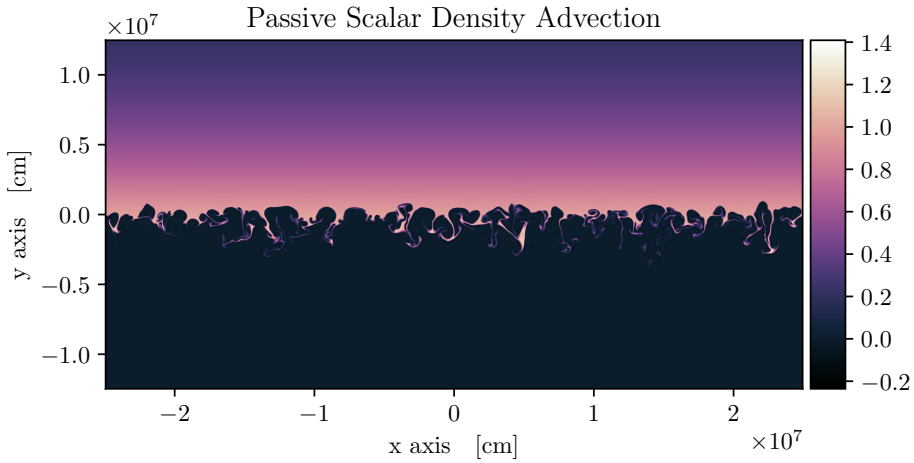
(a) Mach number profile over time.



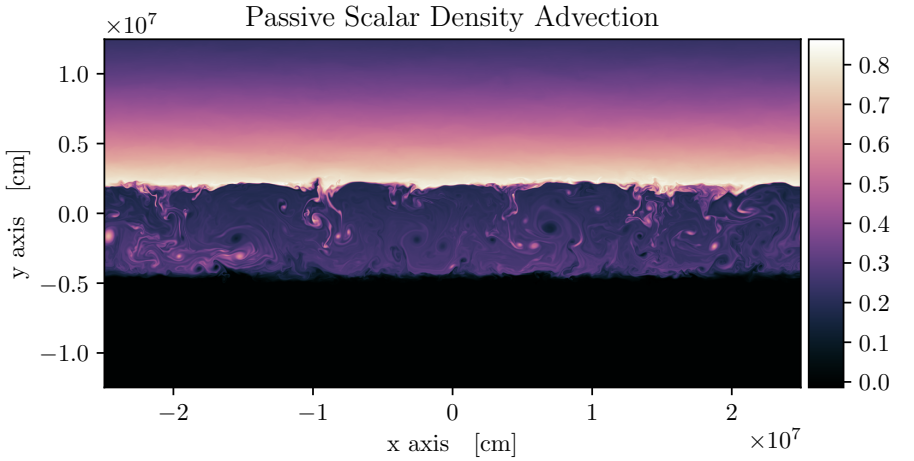
(b) Entrainment of the third passive scalar from the upper stable region (region 3) in the convective region.

Figure 5.2: Time evolution of a 2D run for the Mach number and the third passive scalar (average on the horizontal direction).

in the lower region (region 1), and consequently advected into the convective region. A Lagrangian study of entrainment consists in quantifying the amount of passive scalar ingested at each boundary over time, i. e. integrating each passive scalar density over the convective region at every time step, and analyze the growth of this quantity over time.



(a) Advection of the upper passive scalar (from region 3) at the onset of convective motions, at $t \simeq 150$ min.



(b) The same passive scalar of above, advected at $t \simeq 800$ min. This defines the topology of the boundary.

Figure 5.3: Advection of the passive scalar initialized in the central region over the simulated time

As previously mentioned the two passive scalars are also used to define the boundary topology, position, and width. In Figure (5.3a) we show the upper passive scalar at the onset of convection. The rise of the first convective blobs is clearly visible. In the next 650min part of this dye will be entrained from the upper stable layer in the convective region as it can be clearly seen in Figure (5.3b). We define the boundary position as the surface where the gradient of the concentration of the upper passive scalar changes the most. This holds for both the upper and lower boundary.

For the runs with lower resolution this method worked perfectly, but when moving to a higher resolution a problem appeared. In Figure (5.4) in the middle of the convective layer some blobs of entrained material are visible, that even when fully in a convective regime still contain a very high passive scalar density. This is due to the fact that the higher the resolution, the lower the numerical diffusion.

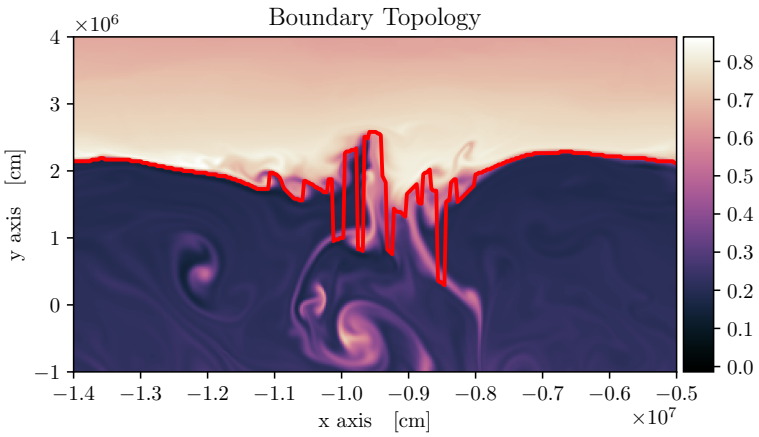


Figure 5.4: The detected upper boundary topology over the passive scalar density (see text).

As a result, the topology is characterized by multiple analytical discontinuities. This suggests that our prescription is only able to provide mean values for the boundary position and width, and not local values. Hence these discontinuities should not be necessarily interpreted as a failure of our method, rather as phenomenon which encapsulates the more turbulent and stochastic regime that we approach by enhancing the resolution, that consequentially can only be treated from a statistical standpoint.

For the 3D runs this problem does not appear, because of the different turbulent structure of convection. In fact in the 3D case, turbulent blobs decay in smaller and smaller blobs (contrarily to what happens in the 2D case), and this natural process helps brake down these spikes (at least at the resolutions we performed our runs).

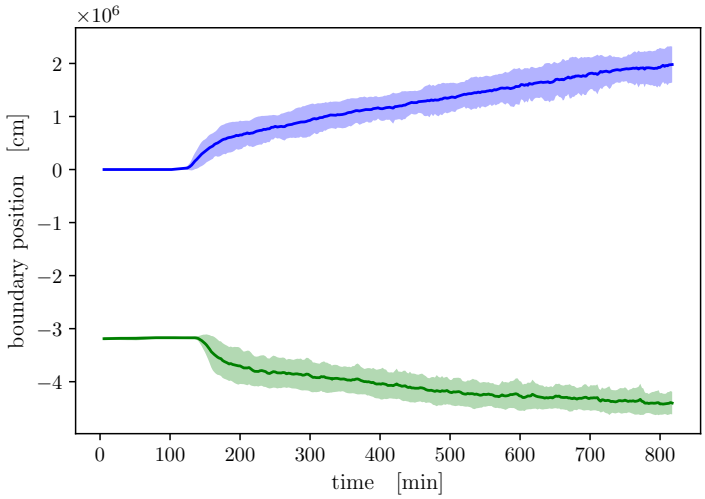
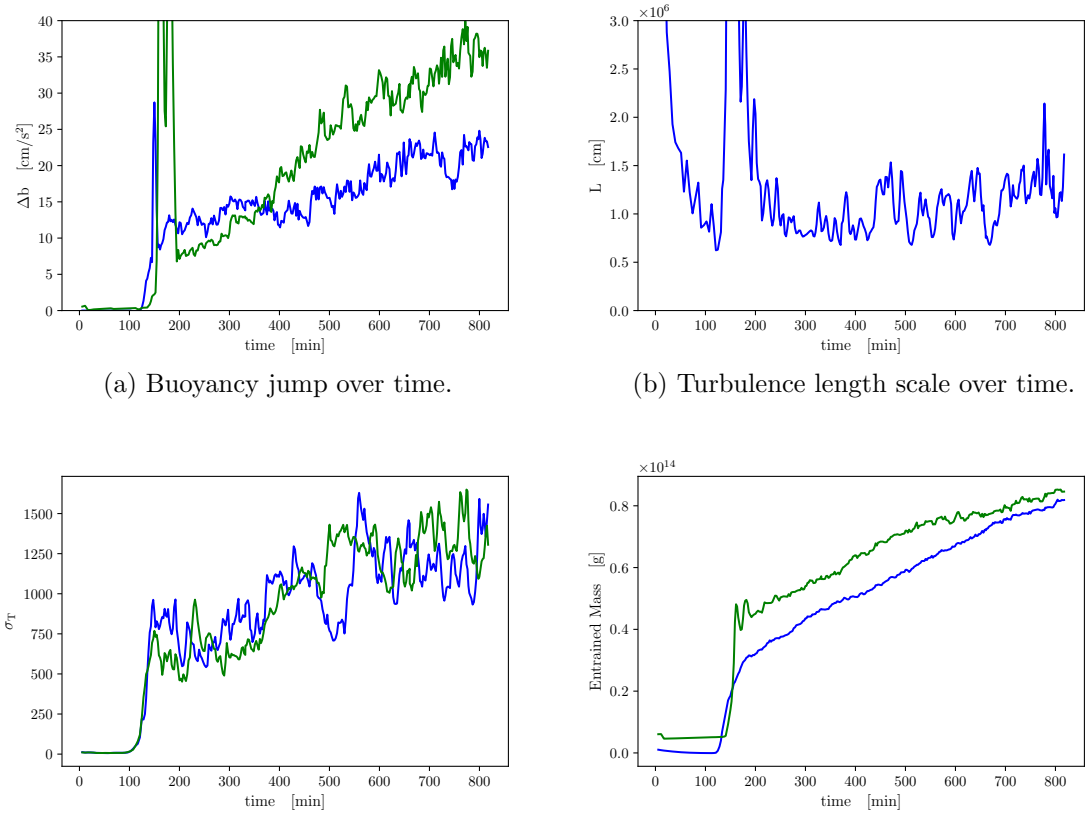


Figure 5.5: Boundaries positions over time. Shaded regions represent the boundary thickness (see text). Blue lines represent the upper boundary, green lines the lower boundary.



(a) Buoyancy jump over time.

(b) Turbulence length scale over time.

(c) Turbulence standard deviation over time.

(d) Entrained mass over time.

Figure 5.6: Time evolution of the parameters needed to compute the bulk-Richardson number and of the entrained mass at the boundaries. Blue lines represent the upper boundary, green lines the lower boundary.

We plot in Figure (5.5) the boundaries positions over time. Shaded regions represent the boundary thickness (calculated as the boundary position standard deviation). Convection clearly starts at about $t = 150\,\text{min}$. After a transition period of about 50min we can clearly see that there is a migration of the upper boundary upward and of the lower boundary downward, which is fairly stable over the run. Specifically, in the upper case it starts in the middle of the simulated region and ends up at $\simeq 2.0 \times 10^6$ cm, for the lower case it starts at $\simeq -3.1 \times 10^6$ cm and moves downwards to $\simeq -4.5 \times 10^6$ cm. The boundaries thickness is also overall stable. As previously mentioned an Eulerian study is impracticable for this phenomenon, because it is hard to quantify how much of the boundary migration in an Eulerian frame is due to the entrainment and how much due to the adiabatic expansion of the fluid. We observe a decrease in density in the central region of a few percent, and we presume the volume expands consequently.

In Figure (5.6) we plot the parameters relevant to our analysis. The first ingredient of the bulk-Richardson number is the buoyancy jump Δb, that we plot in figure 5.6a. In both cases there is an increasing trend, more significant in the lower boundary. This

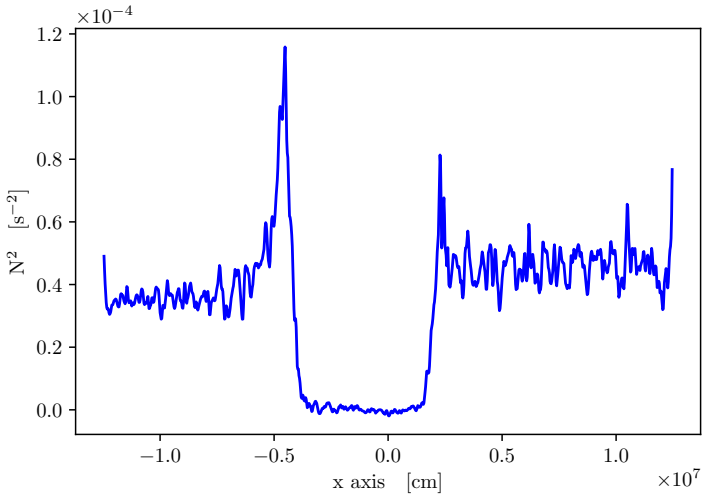


Figure 5.7: Horizontal average of the Brunt-Väisälä frequency at about $t = 500$ min.

plot shows one of the biggest challenges of running simulations to study differentially the bulk-Richardson number: it is extremely hard to obtain a run where all the parameters (Δb, L, σ_{T} and M_{E}) are constant over time. Hence one finds itself in the inconvenient situation of having to average over values that span a full order of magnitude with a strong increasing or decreasing trend, and the quality of the analysis is consequentially affected by this uncertainty.

Notice that the lower boundary is stiffer than the upper one (the buoyancy jump is higher, hence the bulk-Richardson number) even if in the initial setup the Brunt-Väisälä frequency is the same for the two stable regions. Recall that the Brunt-Väisälä frequency N^2 is the quantity we integrate in equation (2.4.23) over the boundary thickness in order to calculate Δb. Looking at Figure (5.7) it is clear that at the convective boundaries spikes arise during the simulation in the Brunt-Väisälä frequency, and in the lower case it is more pronounced. Obviously when integrated over the interface, this gives rise to a bigger buoyancy jump. These spikes in the Brunt-Väisälä frequency at the interfaces were also found by Viallet et al. (2013), Arnett et al. (2015), Cristini et al. (2016).

In Figure (5.6b) we plot the turbulent length scale L over time. In this case we only calculated L at the center of the convective layer, so this value will be used both for the upper and lower boundary analysis. The length scale at the interfaces has a value which is overall comparable, but it is affected by some spikes, reason for which we decided to calculate it in a fully convective region. The turbulent length scale is the parameter that takes more time to stabilize, becoming ultimately stable at around $t = 300\,\mathrm{min}$.

In Figure (5.6c) we plot the standard deviation of the turbulence absolute Mach number at the convective interfaces over time. It is overall constant, with a slight trend to increase. This result has been obtained after extensive tests to correctly tune the heating function. It is in fact extremely hard to forecast the turbulence velocity by the heating generation and its time evolution. This problem perfectly shows the

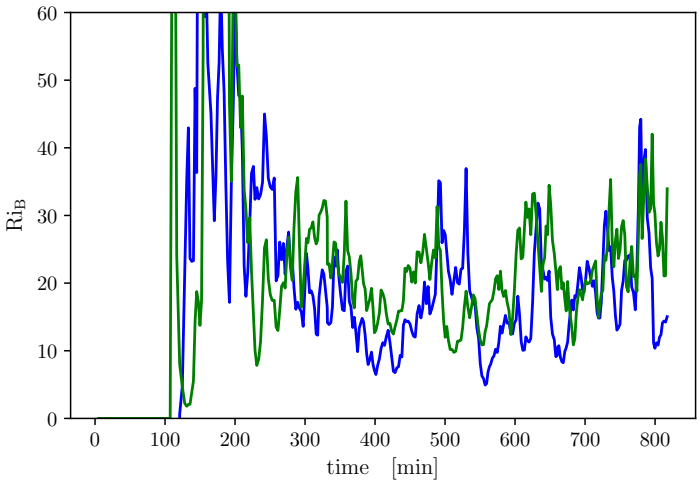


Figure 5.8: Time evolution of the bulk-Richardson number. The blue line represents the upper boundary, the green line the lower one.

non-linearity of hydrodynamics as a theory.

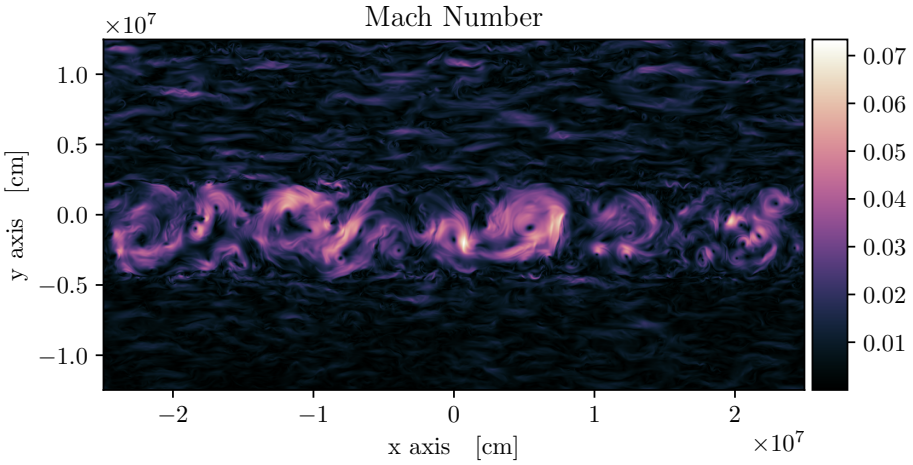
In Figure (5.6d) the entrained mass over time. As already mentioned we will perform a Lagrangian study of entrainment, because it is impracticable to quantify how much of the boundary migration is due to the entrained mass or to the adiabatic expansion of the fluid. After an initial spike of entrained mass, we notice that also the entrainment rate stabilizes. We calculate the entrainment rate by fitting the entrained mass over time between $t = 300\,\mathrm{min}$ until the end of the simulation, and by taking the angular coefficient of the resulting fit.

We show in Figure (5.8) the bulk-Richardson number over time. Blue lines represent the upper boundary, green lines the lower boundary. At around $t = 300\,\mathrm{min}$, when the entrainment becomes linear and we start our analysis, the bulk-Richardson number still oscillates over half an order of magnitude. This obviously deeply affects our data analysis, making necessary a large amount of runs to collect as much data as possible. This also makes a differential study of the entrainment over time impracticable.

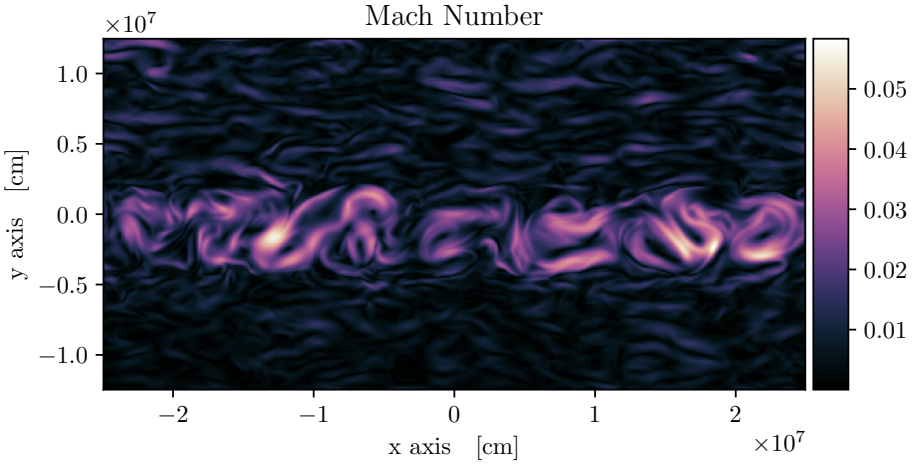
For every run we will collect the relevant parameters and present them in the

Table 5.1: Boundaries parameters for the 2d0.10-100 run, U stands for the upper boundary, L for the lower boundary. The errors are the standard deviations on the mean value.

Run	Δb $(\mathrm{cm\,s^{-2}})$	σ_T $(\mathrm{cm\,s^{-1}})$	L $(10^5\,\mathrm{cm})$	Ri_B	$\dot{M}_\mathrm{E}$ $(10^9\,\mathrm{g\,s^{-1}})$
2d0.7-0.01U	18.17 ± 3.62	1118 ± 202	10.88 ± 2.56	17.07 ± 7.41	1.287 ± 0.006
2d0.7-0.01L	27.68 ± 7.18	1190 ± 218	10.88 ± 2.56	21.62 ± 6.54	0.913 ± 0.002



(a) Mach number profile for the 2048×1024 run.



(b) Mach number profile for the 512×256 run.

Figure 5.9: Turbulent structures at different resolutions at about $t = 500$ min. Notice the higher Mach number in the 2048×1024 run due to lower numerical viscosity (see text).

layout of Table (5.1).

5.1.3 Resolution study

In the last section we analyzed the run 2d0.10-100. In this section we will compare the previous results with the ones obtained by running the same setup on coarser resolutions, in order to understand if we are properly resolving the system.

The same setup was run on a 1024×512 and 512×256. As expected, the finer resolution run shows smaller and more refined turbulent structures in the convective region, while in the coarser run we observe fewer eddies but of bigger size (see figure 5.9).

We report in Table (5.2) the relevant parameters of the upper boundary for the different resolution runs.

Table 5.2: Boundaries parameters for the high, medium, and low resolution runs.

Run	Δb ($\mathrm{cm\,s^{-2}}$)	σ_{T} ($\mathrm{cm\,s^{-1}}$)	L ($10^5\,\mathrm{cm}$)	Ri_{B}	$\dot{M}_{\mathrm{E}}$ ($10^9\,\mathrm{g\,s^{-1}}$)
2048×1024 U	18.17 ± 7.41	1118 ± 202	10.88 ± 2.56	17.07 ± 7.41	1.287 ± 0.006
1024×512 U	15.48 ± 2.31	1020 ± 196	11.07 ± 3.30	17.69 ± 8.40	1.255 ± 0.013
512×256 U	12.82 ± 2.40	970 ± 233	11.50 ± 1.71	17.60 ± 7.66	1.174 ± 0.016
2048×1024 L	27.62 ± 6.54	1190 ± 202	10.88 ± 2.56	21.62 ± 6.54	0.926 ± 0.002
1024×512 L	25.85 ± 6.48	985 ± 179	11.07 ± 3.30	31.93 ± 17.72	1.006 ± 0.017
512×256 L	25.46 ± 6.15	977 ± 143	11.50 ± 1.71	33.33 ± 17.72	1.026 ± 0.031

We observe that all the parameters are overall constant, within a 10% oscillation range. The only exception is Δb. We initially suspected that this increase in the buoyancy jump was due to the boundary detection problem mentioned in the previous section, but the boundary thickness is the same over the three runs. What does not converge are the spikes described in Figure (5.7), that appear to be very sensitive to the resolution.

The turbulence length scale (which is the same for upper and lower boundary because it is computed in the middle of the convective region) slightly decreases with the resolution. This is what one would intuitively expect, since more refined structure become uncorrelated after lower distances.

Woodward et al. (2015) found that the entrainment rate decreases exponentially with the enhancement of the resolution, before converging to the physical value. In our case we observe a constant entrainment (with a slight increasing trend, we presume due to second order phenomena), hence we can safely assume that even in the lowest resolution case we are properly resolving the system.

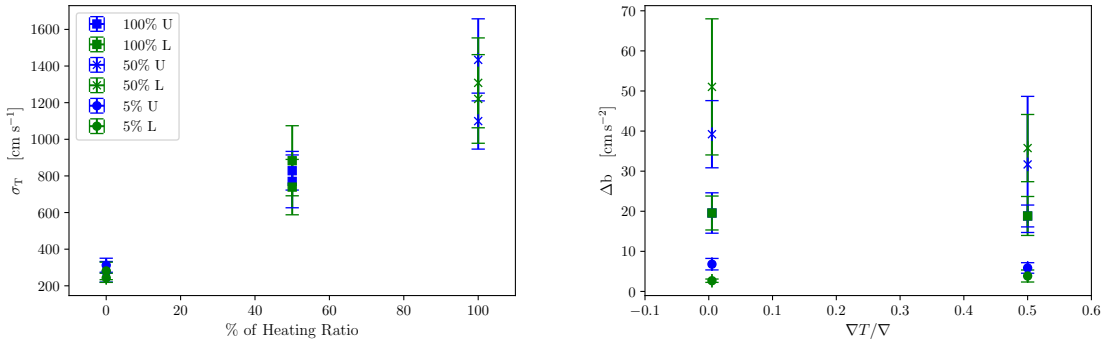
Notice that the turbulence standard deviation increases with the resolution as expected, because of the higher Reynolds number due to lower numerical viscosity. One could show that the relation between the number of cells and the Reynolds number is $N \propto \mathrm{Re}^{9/4}$ (see for instance Coleman & Sandberg (2010)).

One could still point out that the convergence reached in a 2D setup does not imply a convergence in 3D, but because 3D simulations are computationally so expensive, it is impracticable for us to test this hypothesis.

Even at the lower boundary all the parameters oscillate within a 10% range, including Δb. The only exception is the bulk-Richardson number but this is affected by a significant error in the lower resolution runs. Again in the higher resolution case the turbulence standard deviation is higher, as the entrainment rate.

5.1.4 Differential 2D study

As mentioned more than once in the previous chapters, the goal of this work is to perform a *differential* study of the CBM problem. To achieve this, we run four



(a) Dependency of the Mach number on the heating rate (see text).

(b) Dependency of the buoyancy jump on the initial temperature gradient.

Figure 5.10: Dependency of the parameters needed to compute the bulk-Richardson number on the initial simulation setup. Blue dots represent the upper boundary, green dots the lower boundary.

simulations (which provided us 8 boundaries to analyze) varying differentially the stratification and the turbulent regime.

Following the nomenclature we used in the previous section we hence run the six following simulations: 2d0.5-100 (where $\nabla T/\nabla = 0.5$ and the heating ratio is 3.6×10^{15}erg/s), 2d0.005-100, 2d0.5-050, 2d0.005-050, 2d0.5-005 and 2d0.005-005.

We have no direct control over the turbulent Mach number, but we have it on the heating function. We initialized three different heating ratios, specifically 100 %, 50 % and 5 % of the reference value of 3.6×10^{15}erg/s. In Figure (5.10a) we show the linear dependency of the turbulent Mach number as a function of the heating ratio. This result is a powerful tool to forecast a priori in future simulations the resulting bulk-Richardson numbers.

In order to vary differentially the fluid stratification (hence Δb), we initialized our simulation setups with three different values for the temperature gradient, namely $\nabla T/\nabla = 0.5$, 0.50 and 0.005. In Figure (5.10b) we plot the buoyancy jump measured in our simulations over the fraction of the initialized temperature gradient. It seems that the data collected for the most stable stratification ($\nabla T/\nabla = 0.005$) shows a higher value for Δb compared to the least stable one, but this definitely does not reflect the change of two orders of magnitude of the initialized values. We believe that this the consequence of two factors.

The first cause for this discrepancy might be the spikes in the Brunt-Väisälä frequency analyzed in section (5.1.2). It is unclear if the magnitude of these spikes exhibit a non-linear behavior, and a further investigation of such a phenomenon is necessary.

The second reason is shown in Figure (5.11). As already mentioned in multiple occasions, the buoyancy jump Δb is the thermodynamical term in the bulk-Richardson number that encapsulates the information about the stratification, in contrast to the hamiltonian term σ_T. We found instead a clear relation between Δb and σ_T. We im-

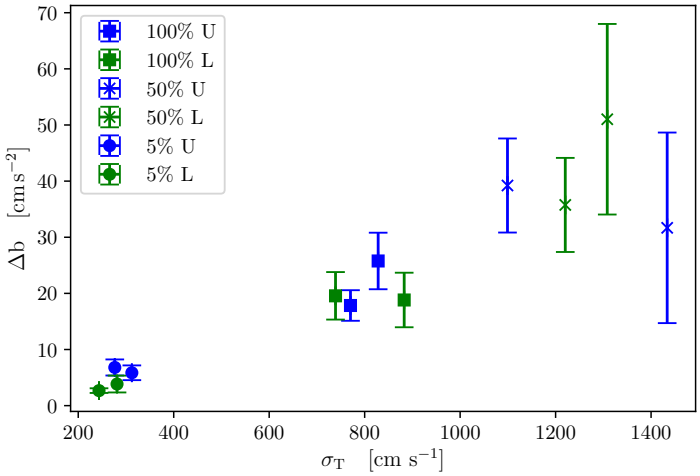


Figure 5.11: Buoyancy jump over turbulent Mach number. The error bars are one standard deviation on the mean values.

pute this correlation to the deformations imprinted by stronger turbulent convective blobs to the boundary. This in fact leads to a wider boundary thickness, which is the domain over which we integrate Equation (2.4.23). The data collected by us on the boundary thickness as a function of the turbulent regime confirms this hypothesis.

The turbulent length scale L also shows an increasing trend with the Mach number. Equation (2.4.20) can be rewritten as

$$\log E = \log A - n \log Ri_{\mathrm{B}}. \tag{5.1.1}$$

Recall that on the left-hand side $E = u_{\mathrm{E}}/\sigma_{\mathrm{T}}$ is the boundary migration speed normalized by the turbulent Mach number, which we calculate by mean of the Lagrangian study in our simulations. On the right-hand side we find the bulk-Richardson number, that we can also calculate for every simulation we run. We plot in Figure (5.12) the experimental value for $\log E$ over Ri_{B}. As usual color blue stands for the upper boundary and green for the lower one. Dots represent the four runs with 5% of the heating function, crosses the two ones with 100%.

It is clear that the data collected by us definitely does not span a range large enough to perform a proper differential study. This problem is a consequence of two facts. As previously mentioned it is extremely difficult a priori to forecast, given initial parameters, what bulk-Richardson number will result from one given simulation. In addition to this, a higher bulk-Richardson number (hence a stronger boundary or a lower turbulent Mach number) requires more computational resources, which we could not afford by the time we realized the problem. In fact, simulating a stiffer boundary requires a better refinement of the grid. In the case of a lower entrainment rate instead, the system needs to be simulated for a longer time, because the boundary moves at a lower speed.

On the other side, reaching a lower bulk-Richardson number would not be relevant

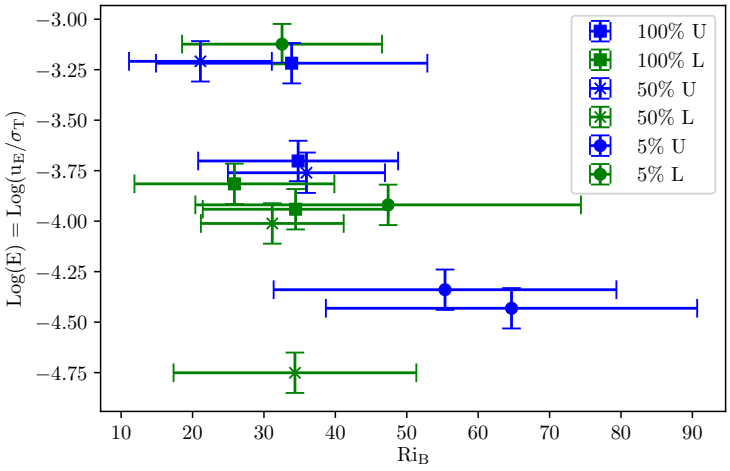


Figure 5.12: Normalized entrainment rate against bulk Richardson number.

for most applications.

As it can be seen at a first glance, there is no clear correlation between the bulk-Richardson number and the normalized migration velocity over time. It appears that a decreasing trend could be found, which would confirm that the n parameter should assume a negative value. This is what one would expect by a qualitative analysis of the problem. This implies in fact that the stiffer the boundary and the weaker the turbulence regime, the lower the entrained mass. The A parameter instead, which is essential to quantify the mass ingestion, cannot be reliably determined by the data. This is very similar to what was found by Cristini et al. (2016) in their carbon burning shell simulation, when compared their results to the ones of Meakin & Arnett (2007).

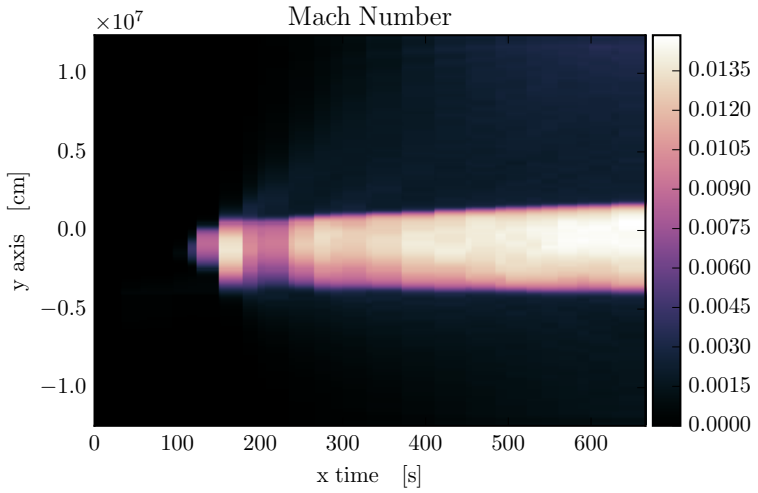
We report parametric values of $n = -0.028 \pm 0.007$ and $A = 0.071 \pm 0.291$. Both n and A are one order of magnitude lower than the 3D reference values that can be found in the literature. Nevertheless for no reason we would expect that the parameters extracted for a 2D setup would agree with the three-dimensional case.

The error values clearly show that the data range is not wide enough to perform an accurate differential study. Future runs that were not possible during the master project will reach bulk-Richardson numbers of one order of magnitude higher than the ones obtained for this work. This will be achieved primarily by using a non-uniform Cartesian grid which is implemented in SHL. This will help us better refine the convective boundary regions and avoid wasting computational resources to resolve the stable layers above and below.

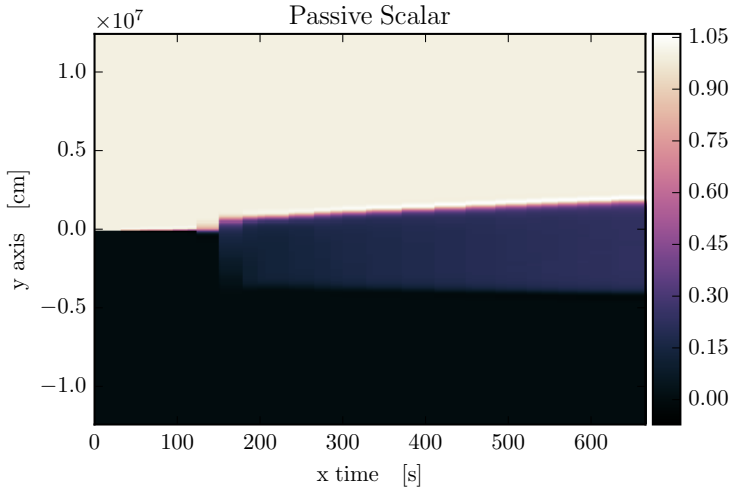
5.2 3D simulations

After having correctly tuned the heating function in a 2D setup, we switched to a 3D setup. Only one simulation was run: 3d0.10-100. In fact, because of the problems discussed in the differential study section, we want to make sure that we can successfully outdistance the bulk-Richardson numbers before running other computationally very expensive 3D runs.

The simulation was run on the JUQUEEN machine at Jülich Supercomputing Center (JSC) and scaled over one rack (16384 cores) for four days. Although SLH



(a) Mach number profile over time.



(b) Entrainment of passive scalar from the upper stable region into the convective region.

Figure 5.13: Time evolution of a 3D run for the Mach number and the upper passive scalar. In comparison to the 2D case less output files were saved, hence the lower resolution in time.

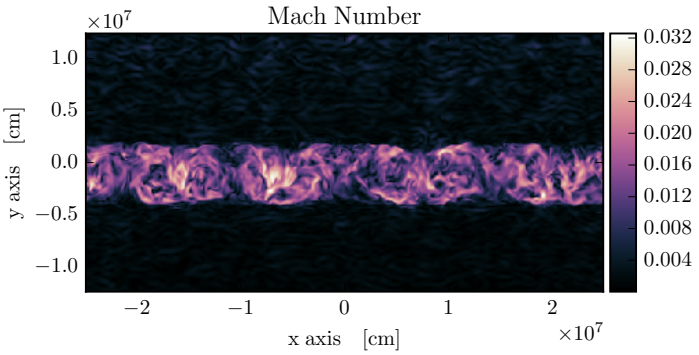


Figure 5.14: Mach number of a 3D run at about $t = 500\,\text{min}$.

scales very well even on tens of thousands of cores, on JUQUEEN a very high number of CPU hours was needed, primarily because the PowerPC® A2 processors (IBM) of JUQUEEN are optimized to obtain the highest efficiency in terms of computations per watt, rather that per core.

Only one output every 500 time steps was saved, hence the lower resolution in time for the plots compared to the 2D analysis. This was not necessary to reduce the output size (around 300 GB total), but to avoid to spend too many CPU hours in I/O processing.

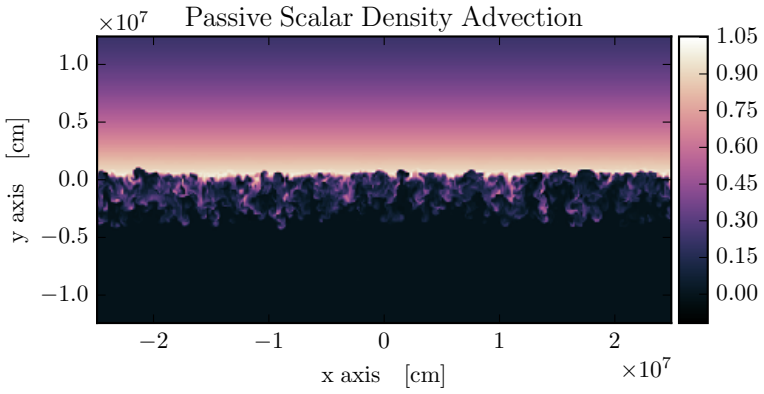
5.2.1 Evolution of a 3D run

The physical setup was mapped on a uniform Cartesian grid with $512 \times 256 \times 512$ cells. In analogy with the 2D setup, we implemented wall boundary conditions for the vertical direction, and periodic boundaries for the horizontal ones. The same usual time-dependent gaussian-shaped heating function was furthermore implemented, with the same heating rate.

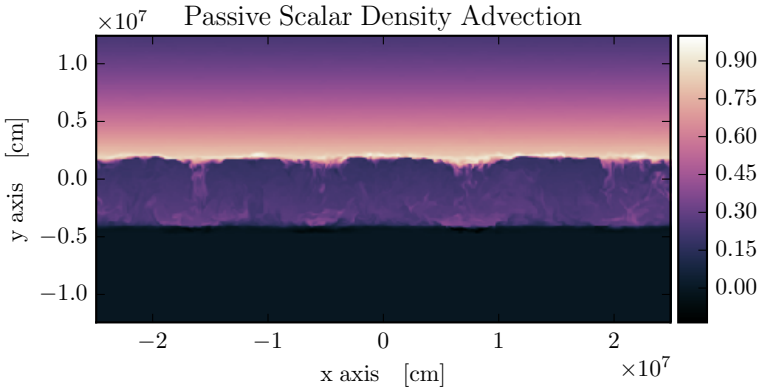
Similarly to the 2D case, convection starts at $t \simeq 150\,\text{min}$, and in Figure (5.13) the growth of the convective region by entrainment of stable medium is visible.

The 3D case shows also a Mach number of the turbulent flow constant in time over the run, as shown in Figure (5.13a), but compared to the 2D Mach number in Figure (5.2a) it is significantly lower. This result was expected because, as previously mentioned, three-dimensional turbulence has a different behavior compared to the two-dimensional one. In our cases the 3D turbulence standard deviation was always a factor of a few lower than in the 2D case, which is in agreement with what found by Meakin & Arnett (2007) in their core-convection simulation.

The entrainment plotted over time in Figure (5.13b) is also roughly linear and qualitatively resembles the 2D result, with the upper and lower boundaries that are migrating upward and downward respectively. The different turbulence configurations between the 2D and 3D case can be also better understood by comparing Figure (5.14) with Figure (5.9a). In Figure (5.9) we showed that the better the resolution, the more refined and complex the turbulent structure. When considering a 3D output, even by looking at a 512×256 section of the grid, the Mach number already shows a very complex and convoluted shape, comparable in complexity to the 2048×1024 2D case.



(a) Onset of convective blobs at $t \simeq 150\,\mathrm{min}$



(b) Passive scalar advection at $t \simeq 600\,\mathrm{min}$.

Figure 5.15: Passive scalar density advection for the 3D run. Notice the different shape of the convective blobs compared to Figure (5.3).

This obviously has an impact on the entrainment dynamics. Notice that also in this case the Mach number is roughly the half of the 2D case.

In Figure (5.15) we plot, as we did for Figure (5.3), the upper passive scalar density advection. In Figure (5.15a) the onset of convection is visible by the presence of the fist convective blobs, that over time fill the convective region as visible in Figure (5.13b). Notice that in the 3D case the passive scalar is advected fairly homogeneously in the convective layer, and the problem described in Figure(5.4) for the boundary detection did not appear. This is another consequence of the previously mentioned 3D turbulent cascade, that helps spread the passive scalar in the convective region better than in the 2D case.

Bibliography

Arnett W. D., Meakin C., Viallet M., Campbell S. W., Lattanzio J. C., Mocák M., 2015, ApJ, 809, 30

Bartelmann M., 1994, pp 127–160

Boffetta G., Ecke R. E., 2012, Annual Review of Fluid Mechanics, 44, 427

Böhm-Vitense E., 1958, ZAp, 46, 108

Coleman G., Sandberg R., 2010, pp 2–5

Cristini A., Meakin C., Hirschi R., Arnett D., Georgy C., Viallet M., 2016, ArXiv e-prints

Kippenhahn R., Weigert A., Weiss A., 2012, pp 47–52

Meakin C. A., Arnett D., 2007, ApJ, 667, 448

Miczek F., 2013, pp 30–62

Richardson L., 1922, Quarterly Journal of the Royal Meteorological Society, 48, 282

Stull R., 1988, pp 177–180

Viallet M., Meakin C., Arnett D., Mocák M., 2013, ApJ, 769, 1

Viallet M., Meakin C., Prat V., Arnett D., 2015, A&A, 580, A61

Woodward P. R., Herwig F., Lin P.-H., 2015, ApJ, 798, 49

Acknowledgements

At this point I need to thank my supervisor Fritz Röpke for welcoming me in the PSO group, suggesting such an interesting topic to me to work on and providing a powerful tool such as SLH. I need to thank every member of the group for creating such an inspiring and friendly atmosphere. When I walked into my office one year ago, I couldn't have expected any better. In particular I need to thank Philipp Edelmann who patiently introduced me in the computational universe and Samuel Jones for sharing his experience on the CBM problem.

I also feel I need to thank every member of the University of Heidelberg and of the Heidelberg Institute for Theoretical Studies who contributed over the past two years to my academic education.

Furthermore I need to thank the Jülich Supercomputing Center for providing part of the computational resources used for this work.

Last but not least, I need to thank the members of the open source community for developing and maintaining very valuable tools such as GNU/Linux, Python, Git and Vim. Without them this work would have not been possible.

Erklärung:

Ich versichere, dass ich diese Arbeit selbstständig verfasst habe und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Heidelberg, den 7. April 2017 .....