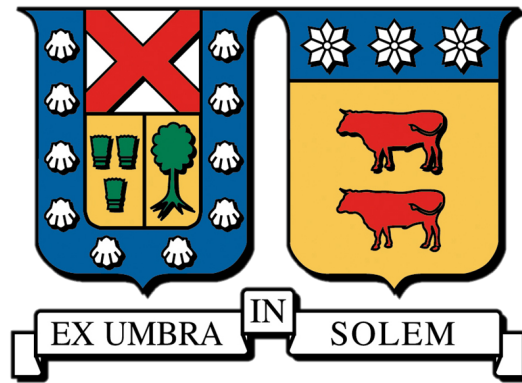


UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA

DEPARTAMENTO DE FÍSICA



Licenciatura en Astrofísica

**Running Non Zoom and Zoom Cosmological Simulations Using
MUSIC and RAMSES Codes**

Gonzalo Aichele Ramos

PROFESSOR: Rory Smith

January 11, 2023

Contents

1	Introduction	1
1.1	Environments and Their Effects on Galaxies	1
1.2	RAM Pressure Stripping in Void Walls	3
1.3	Simulations	6
2	Simulation Software Packages	7
2.1	MUSIC	7
2.1.1	Input file	8
2.2	RAMSES	11
2.2.1	Input File	13
2.3	Tools to Analyze Cosmological Simulations	16
3	Simulations Run	17
3.1	Non Zoom Simulation	17
3.2	Zoom Simulation	21
4	Conclusion	24

Chapter 1

Introduction

Whenever astrophysicists face problems regarding the large scale structure of our universe (LSS), the main challenge that raises is the limitation of the instruments used to measure, even more when used to observe at high redshifts, where observations get biased due to low luminosity galaxies for which the current resolution of the instruments is not enough. This is where cosmological simulations come in handy, these simulations require the application of the whole theory on which our whole understanding of the universe is based, this implies that if a simulation is correctly constructed we can predict the observations and if these match our predictions then the theory is confirmed. Cosmological simulations duplicate the universe into a box that has the same conditions and interactions between particles, this box will allow us to study multiple interactions present in the intergalactic medium.

There are two main benefits that simulations have over the observations, in the first place the 3-dimensional detail they give us. When observing the universe, astronomers get to see a 2D image and have to employ different tricks to interpret the differences present in the line of sight, with simulations we can study a controlled box which we can study from any angle required. The second and most important advantage is the ability to study the time evolution of specific objects. When observing deep fields, it is impossible to study the time evolution of the objects due to the large time-scales on which the processes involving significant changes in structures and properties take place, cosmological simulations give us the ability to follow objects such as dark matter halos in time. The latter fact is important for this particular project, because the long term goal is to study a certain environmental effect that might be causing a specific galaxy morphology, one possibility for studying this phenomenon is to follow these galaxies through time to confirm that these morphologies can be caused by this effect, without time evolution it is not possible to identify the cause.

1.1 Environments and Their Effects on Galaxies

The LSS of the universe can be divided in substructures called environments that present different morphologies and densities, the latter property is responsible for multiple effects caused on the galaxies that conform each environment. There are 6 environments in the LSS, voids, walls, filaments, groups, cluster and super-clusters, ordered by increasing density, they are shown in figure 1.1, walls are not shown because they are hard to identify, even more in a 2D image. Walls

are pseudo plane structures that form all the other structures except for voids which they limit, two intersecting walls form a filament, within the filaments there are groups, intersecting filaments form clusters and a collection of clusters is called a super cluster. Voids are regions where matter has been almost completely stripped by the other structures due to gravitational effects and they are therefore the least dense environments.

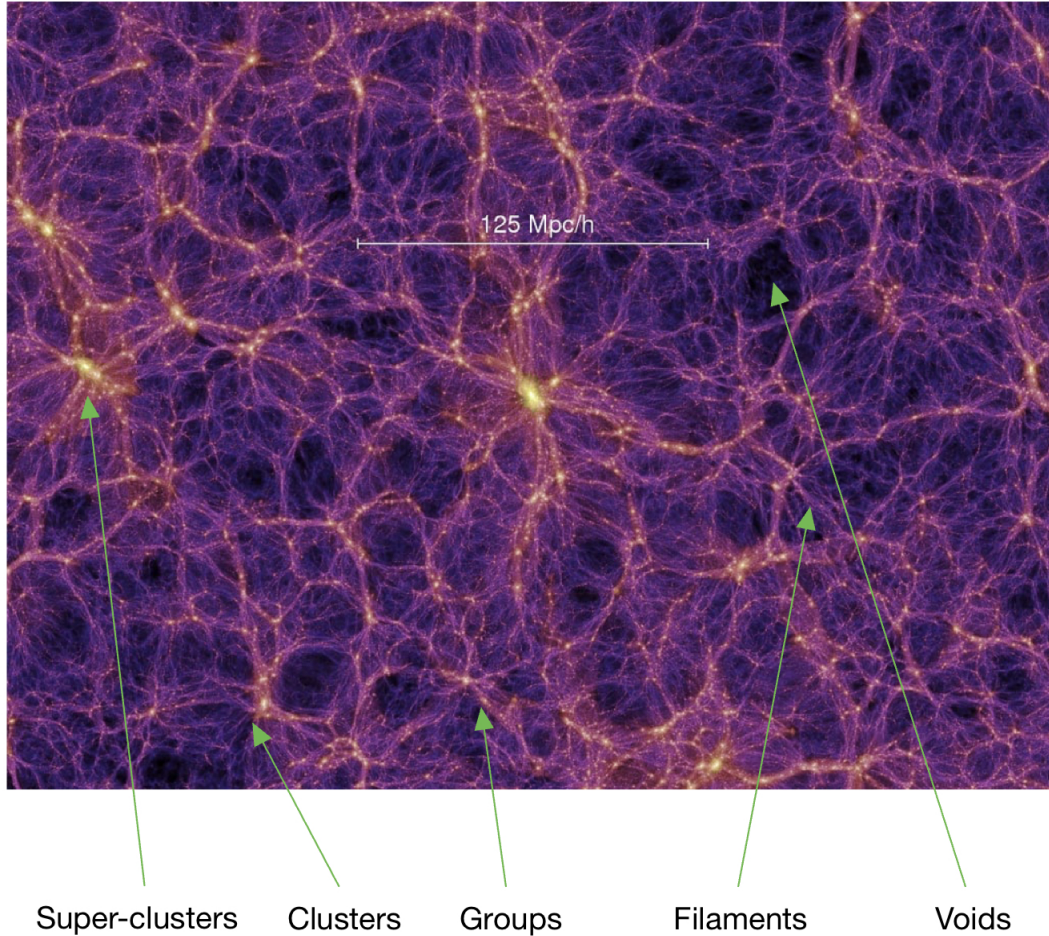


Figure 1.1: LSS of the universe identifying the substructures present.

These definitions are often times ambiguous and mainly serve the purpose of a qualitative description of the LSS, it is more useful for scientific goals to talk about environments in terms of density, which is strongly correlated to the effects the environment have on galaxies. These effects also depend on the galaxy mass, which is the most fundamental property of galaxies, it is correlated with all the other galaxy properties, therefore a good way to understand the effect environments have on galaxies is to study the effect they have on galaxy mass, Peng et al. 2010[4] show how environmental density affect galaxy properties by studying galaxy star formation rate(SFR) at different masses and environmental densities, they found that galaxies with high stellar mass(greater than $10^{11} M_{\odot}$) are not affected by environmental effects and keep their low SFR no matter the environmental density. Whereas for low stellar mass(dwarf) galaxies(under $10^{10} M_{\odot}$), at a fixed mass their SFR depends on the environmental density, this means that low

mass galaxies are susceptible to being affected by environmental effects.

Since environmental effects depend on the density of the environment, higher density environments are more likely to have an effect on galaxies. There are multiple environmental effects that a galaxy can go through, some of them are:

- Tidal Encounters: Galaxies encountering each other produces a tidal force between them.
- Tidal Stripping: The effect of loosing mass due to tidal encounters.
- Galaxy Mergers: Collisions of galaxies that end up in one resulting galaxy.
- Gas Accretion: Accretion of environmental gas.
- RAM Pressure Stripping: Loss of gas due to a pressure proportional to velocity and environmental density.

Of all these effects, RAM pressure stripping is the one of interest for this study, the reasons for this interest will be discussed on section 1.2.

1.2 RAM Pressure Stripping in Void Walls

As discussed by Gunn & Gott 1972[1], ram pressure stripping is the pressure that the interstellar material of a galaxy feels due to the relative velocity v_r between the galaxy and the environmental gas surrounding the galaxy. This pressure is proportional to the environment gas density ρ_e and is determined by equation (1.1).

$$P_r \simeq \rho_e v_r^2 \quad (1.1)$$

Whether the galaxy is effectively stripped, depends also on the binding of the interstellar gas to the galaxy, this is why typically on spiral galaxies the gas is stripped from the outer regions of the galactic disk where the gas is more weakly bound due to the separation from the center of the galaxy and the reduced thickness of the disk at higher radii.

Ram pressure stripping not only changes the morphology of the galaxy, it also affects its properties due to the loss of most if not all of the galaxy's gas, this loss (B. Vulcani et al. 2022[10]) has observational implications, the main one is the tail of gas left behind the galaxy path which in galaxies undergoing ram pressure stripping have been called “jellyfish galaxies” (Figure 1.2) due to their morphologies.



Figure 1.2: Spiral galaxy ESO 137-001, example of a galaxy experiencing ram pressure stripping presenting a “jellyfish” morphology. Image taken from NASA’s website https://images.nasa.gov/details-GSFC_20171208_Archive_e001178

As mentioned on section 1.1, the impact of environmental effects on galaxies depend on the environmental density, in the case of RAM pressure stripping this is also true. Higher density environments such as clusters present galaxy velocities of around $v_{gal} \sim 1000 \text{ km/s}$, these environments can be treated as static so $v_r \simeq v_{gal}$, this velocity yields a high RAM pressure that can very effectively strip dwarf galaxies. For the case of less dense environments such as void walls, the environment gas is falling in the same direction and velocity as the galaxies, this means that for these galaxies the velocity $v_r = 0$ and therefore RAM pressure stripping is not expected to happen in these cases and it is often not studied. This doesn’t mean that RAM pressure stripping can’t occur on low density environments, B. B. Thompson et al. 2022[8] studied environmental gas accretion in low density environments such as voids and void walls, they classified two types of halos present in these two environments, the fast flow halos and the slow flow halos which satisfy the following conditions:

$$\text{Fast flow : } v_r \geq 1.5 \times v_{esc}$$

$$\text{Slow flow : } v_r < 0.5 \times v_{esc}$$

Where v_r is the velocity from equation (1.1), and v_{esc} is the escape velocity of the halos. They found that slow flow dominated the halo population in every environment of the voids and were present even inside the void walls, whereas fast flow halos represented only 12% of the total, and they were concentrated within $1.0 \text{ Mpc } h^{-1}$ from the walls. In order for these halos to be fast flow,

the only way is for them to have passed the void wall after falling towards it and be opposed to the infalling gas from the other side of the wall as can be appreciated by the orange arrows in figure 1.3, this means these halos have velocities of $v_r = 2v_{gal}$ and since v_r is squared in equation (1.1), at sufficiently large v_{gal} it might be possible to compensate the low density of the environment ρ_e and have RAM pressure stripping occur to these galaxies and present jellyfish morphologies.

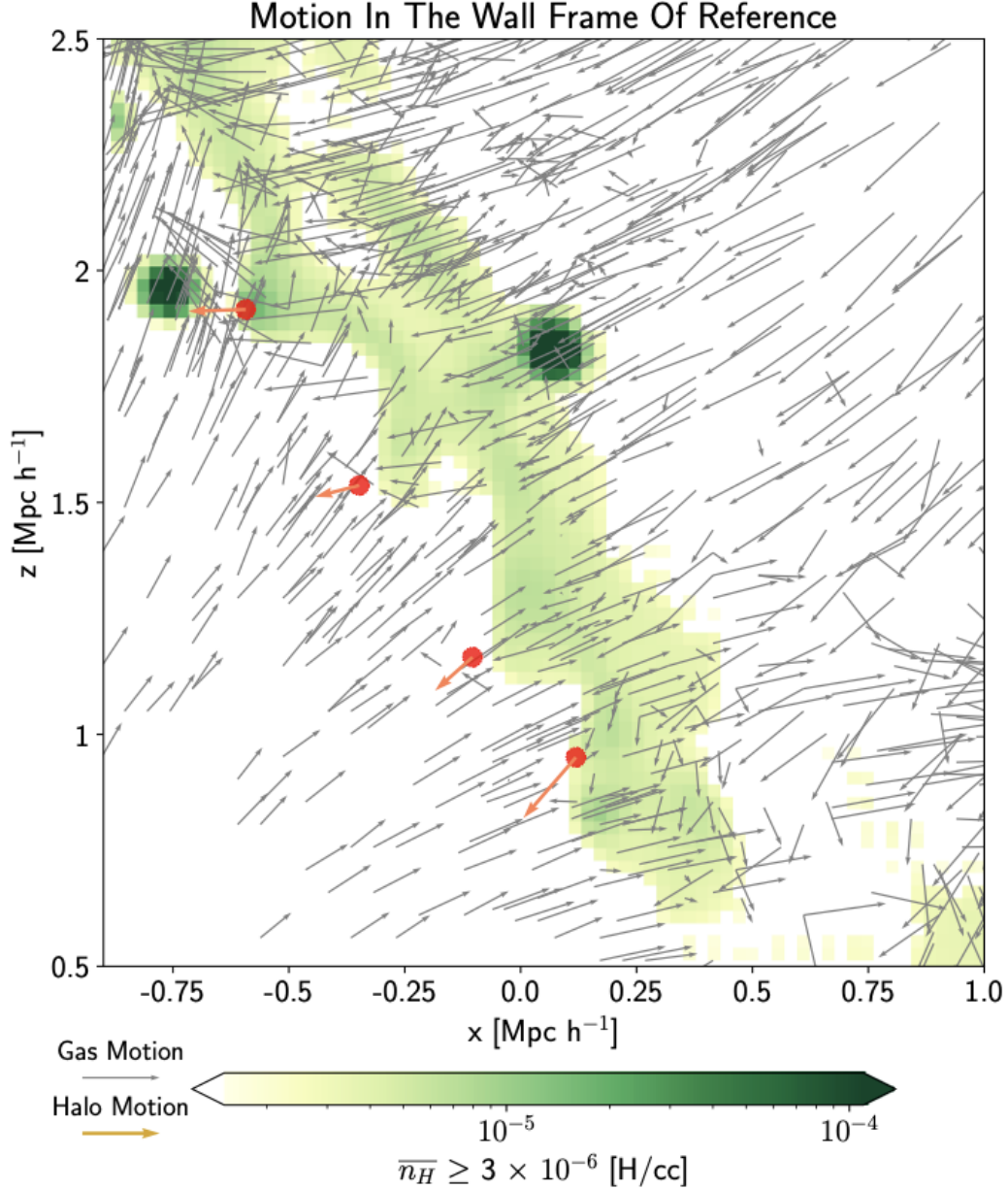


Figure 1.3: Section of a void wall with it's gas density shown in green and the infalling gas shown with grey arrows proportional to the falling velocity. The whole plot is in the reference frame of the wall. The red dots are four halos with their respective velocities in orange arrows which have passed through the wall and came out from the other side and currently oppose the gas velocity. Plot taken from B. B. Thompson et al. 2022[8].

1.3 Simulations

The long term goal of this study is to simulate the evolution with time of individual fast flow halos coming out of void walls which are jellyfish galaxy candidates, and confirm if it is possible for RAM pressure stripping to occur in these halos. To achieve this, it is important to first understand how to run cosmological zoom simulations which have high resolution in specific regions.

As mentioned in section 1.2, B. B. Thompson et al. 2022[8] studied gas accretion in cosmological voids, for this they used two codes to run their cosmological simulations, the Multi-Scale Initial Conditions(MUSIC; Hahn & Abel[3]) software package to set the initial conditions, and the adaptive mesh refinement(AMR) code Ramses(R. Teyssier[7]) to evolve the initial conditions up to redshift $z = 0$, in their simulations, the focus was to increase the resolution(zoom) on the whole void to inspect the environment without focusing on the specific halos, this limits the individual properties that can be studied.

To run a cosmological zoom simulation, it is important to understand the goal of the simulation because it is not possible to run high resolutions in the whole box due to the intensive computing power and time required. If we want to increase resolution, the region in which the resolution is increased must be smaller to compensate, this last fact is convenient for this study, since we want to study small regions the resolution can be increased much more. The common method used to run zoom simulations is to first run a non zoom simulation to identify the regions of interest and then run the simulation again but increasing the resolution in those regions.

The purpose of the present work is to understand how cosmological simulations work and are set up in order to increase the resolution in specific regions of interest of the cosmological box, with the motivation of eventually running simulations focused on individual DM halos in void walls. The second goal of this work is to learn how to use tools for the analysis of cosmological simulation to be able to do science after running them.

This work is structured as follows. In chapter 2, the way both MUSIC and RAMSES work is discussed and also the methodology to run them for non zoom and zoom cosmological simulations. At the end of the same chapter the tools for the simulation analysis are briefly mentioned. In chapter 3, I talk about the simulations run to understand how the codes work, and discuss if those simulations were correctly run. Finally in chapter 4, I present the conclusion and the plans for the future.

Chapter 2

Simulation Software Packages

This section will be dedicated to understanding the methodology to run cosmological zoom and non zoom simulations using MUSIC[3] and RAMSES[7], plus the mention of the tools required to analyze the resulting simulations.

The first step to run a cosmological simulation is to set the initial conditions, in this case this is done using the Multi-Scale Initial Conditions code(MUSIC). This software package sets a box with a defined number of particles which may or may not have the same mass depending of the refinement, and also may be baryons, DM particles or both, this particles are the set with an initial perturbation. Once the initial conditions are ready, they can be used to run simulations on various codes but the one selected for this study is the adaptive mesh refinement(AMR) code called RAMSES, which receives the initial conditions and evolves them from a certain redshift up to $z = 0$. Finally, the outputs are analyzed using AHF, which identifies the DM halos from each RAMSES output. In sections 2.1, 2.2 of this chapter, each code will be described individually, and the tools to analyze cosmological simulations will be presented in section 2.3.

2.1 MUSIC

The starting point from which cosmological simulations run are the initial conditions, these are the set of positions and velocities of the particles that constitute our cosmological box that is then evolved. Setting these initial conditions(ICs) is not a trivial task at all, MUSIC handles the complex cosmology behind the set up to allow the users to generate their own ICs by giving the program an input file containing the desired parameters, all these parameters were studied from the MUSIC user's guide[2].

As the name suggests, MUSIC allows for Multi-Scale ICs, this means that it sets the initial conditions for different levels of mass refinement for the particles used, in order to do this without having problems of high mass particles interacting strongly with low mass particles, it has to gradually decrease the mass of the particles by levels. This feature is particularly useful for zoom simulations since the goal is to have high resolution in specific regions without having to compute the whole box with high resolution. Together with this last fact, MUSIC centers the region of interest using periodic boundary conditions(which will be discussed on section 2.2), this allows us to run simulations without zoom to identify a region of interest and then feed MUSIC this region to increase its resolution and account for the shift to the center to then run more simulations changing

the focus region but keeping almost the same cosmological box. Note that it is impossible to keep the exact same box as changing the particle masses may lead to slightly different final positions of DM halos or even generate multiple halos from one at low resolution.

Once the ICs are ready, an output file will be generated, which is then used to evolve the cosmological box using other code, in this case RAMSES. In the following subsection, the input file parameters used to set the ICs are discussed.

2.1.1 Input file

The input file that MUSIC receives contains 5 sections, setup, cosmology, output, random and poisson, of these 5 sections only the first 3 were modified and studied for this project. In this section the relevant parameters used for this projects will be discussed.

[Setup]

The setup section is where the general parameters to set the box and the refinement, an example of this section is the following.

[setup]	
boxlength	= 100
zstart	= 50
region	= ellipsoid
levelmin	= 7
levelmax	= 12
padding	= 8
overlap	= 4
baryons	= no
region_point_file	= /pth
region_point_shift	= 55, -51, -48
region_point_levelmin	= 7

It is worth noting that these are not all the parameters that can be used for the setup section, but they are the most relevant ones for the context of this project.

The first parameter sets up the length of the sides of the box in comoving Mpc/h, it is important to set the boxlength big enough for the cosmological principle to hold. Zstart fixes the initial redshift at which the simulation starts, this parameter should be set big enough for LSS to form. Then there are the region parameters, these allow the user to select the region of interest by entering a file path which contains the position of the particles that are to be refined, specifying the previous coordinate shift and the minimum level of the particles, this generates a region as close to a 3D ellipse as possible surrounding the particles. This last tool is really useful in the context of this project because it helps increasing the resolution of individual halos without sacrificing computer processing power in regions that are not of interest. Finally there are the level parameters, these

determine the number of particles, each level has a number of particles determined by

$$n_{\text{part}} = 2^{3L}$$

Where L is the level, the factor 3 accounts for the three dimensions. This section is where the zoom simulation is mainly configured, it is possible to use the same value for levelmin and levelmax, doing this would lead to a non zoom simulation with the same particle mass throughout the entire box. In order to do a zoom simulation, it is necessary to have $\text{levelmax} \geq \text{levelmin} + 1$, this means having regions where the number of particles is increased by a factor of 2^3 , and the particle mass is decreased by a factor of 2^{-3} for each increased level.

As mentioned at the beginning of this section, the refinement has to be gradual if one is to refine from a level L to a level greater than $L + 1$. This is done by MUSIC using the parameter padding, this parameter gradually increases the resolution from levelmin up to levelmax by assigning N_{padding} grid cells in for each level between levelmin and levelmax.

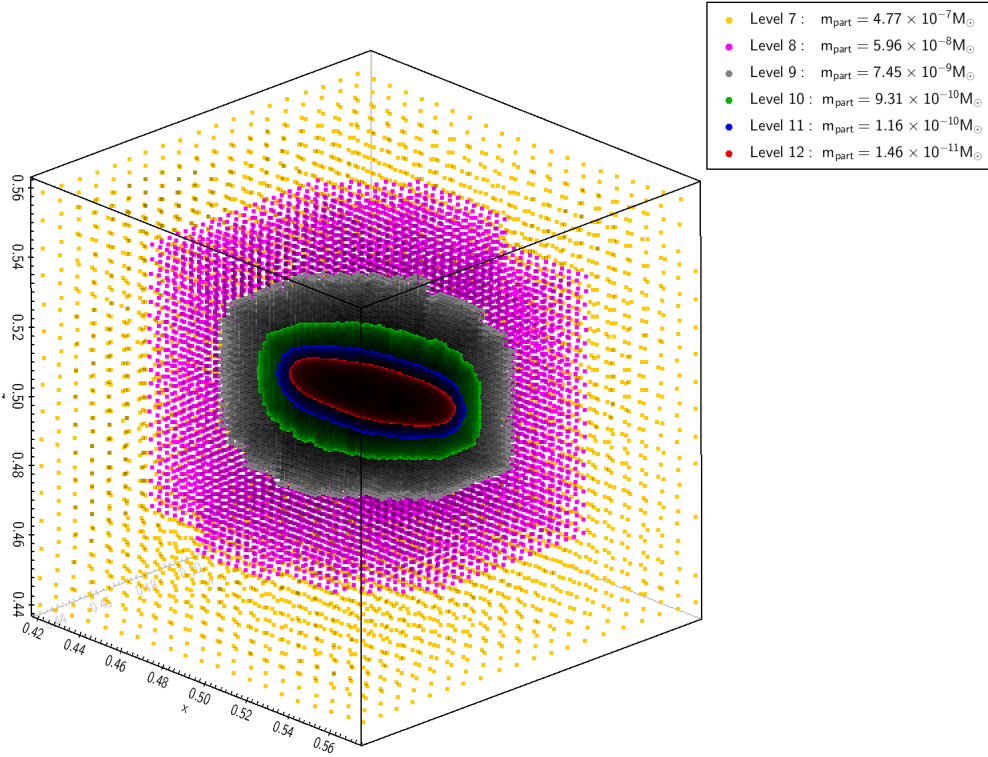


Figure 2.1: Initial conditions set in a box at redshift $z = 50$, where it is possible to appreciate the increase in resolution from level 7 up to level 12 using padding. On the legend, the masses of the particles at each level is specified in solar masses $M_{\odot}$.

Figure 2.1 shows how the mass refinement occurs at different levels for the initial conditions, figure 2.2 is a zoom for level 11 where the 8 grid cells conforming this level can be appreciated. In the same figure it is also possible to see the perturbations in the positions of the particles. It is worth noting that for each level, the padding has the same number of grids, but the padding

regions are bigger at lower levels because there are less particles and therefore the size of those grids is bigger.

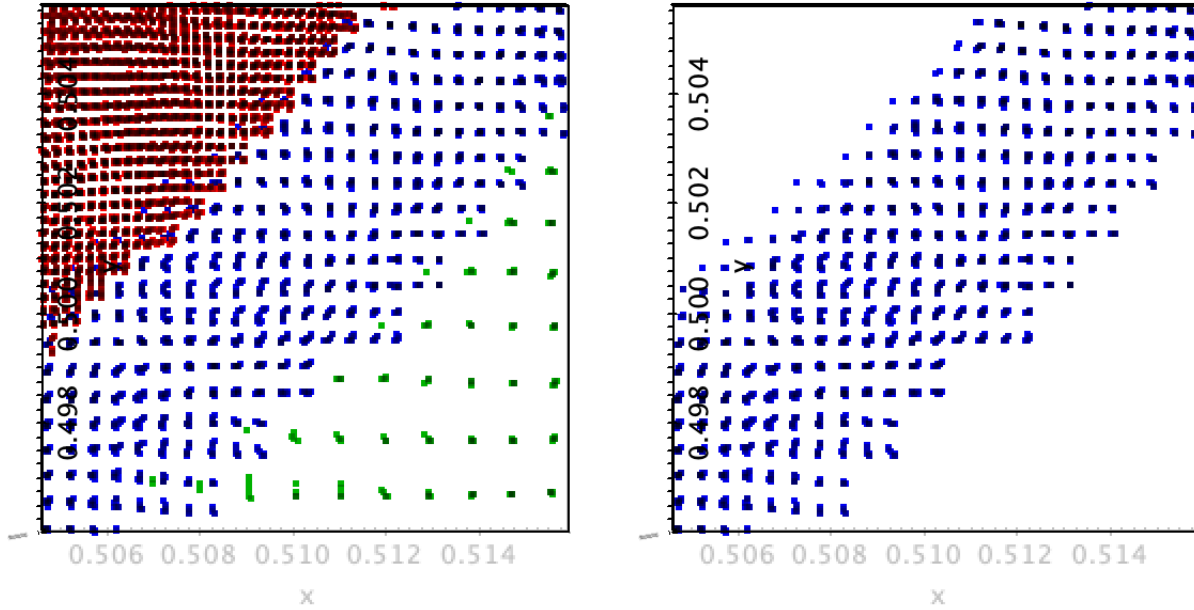


Figure 2.2: Zoom in levels 12, 11 and 10 at redshift $z = 0$ keeping the same color coding as in figure 2.1 to appreciate the padding between levels. The upper left corner are the particles at level 12, the lower right corner are the particles at level 10. On the left panel the three levels appear to show the different layer, on the right panel only particles at level 11 are shown.

[Cosmology]

As the name suggests, the cosmology section is where the cosmology of the box at redshift z_{start} is defined, as shown on the following example.

```
[cosmology]
Omega_m           = 0.28
Omega_L           = 0.72
Omega_b           = 0.045
H0                = 70.0
```

Here, the first three parameters define the matter, cosmological constant and baryon densities respectively, H_0 is the Hubble parameter at redshift zero in km/s/Mpc.

[Output]

Finally there is the output section, where the user defines the format required. There are multiple formats for many different cosmological simulation codes that require ICs, for the case of RAMSES,

the format used is “grafic2”. In this section we also tell MUSIC the filename to store the output, for each level of refinement, MUSIC stores a file containing the positions and velocities of the particles.

2.2 RAMSES

Once the ICs are set, we are left with a box full of perturbed particles as shown in figure 2.1. By some method we need these particles to evolve with time to form the LSS as we know it, there are multiple codes that receive ICs and evolve them in time, but the one studied in the present work is RAMSES[7], which is an adaptive mesh refinement(AMR) code. The way RAMSES works and is set up, as well as it’s output will be discussed in this section.

When attempting to evolve the ICs of the cosmological box, the problem that raises is the way in which gravitational interactions occur in each time step. Recall from section 2.1 that the number of particles in the box depends on the level of refinement and number of dimensions by 2^{3L} , this means that the average order of magnitude for the number of particles in a box is $\sim 10^6$, to compute the gravitational force exerted on each particle by the others at each time step would require enormous amounts of time. This is why people have came up with multiple solutions to this problem, one of them is the one used by RAMSES, which is the particle mesh(PM) method.

The PM method consists on setting a 3D Cartesian grid structure in the box containing the particles, naturally the particles will end up inside the cells of the grid. Then each cell of the grid is treated as a cubical shaped cloud of uniform density corresponding to the masses of the particles inside each cell, this method is called Cloud-in-Cell(CIC). Once the density distribution is computed for the whole mesh, RAMSES calculates the spatial solution for the Poisson equation to obtain the gravitational potential. The final step is to apply this potential to each particle depending on the cell it is occupying in order to compute the particle acceleration, then to advance in time step, RAMSES calculates the particle’s new velocity and position.

The mesh employed in RAMSES is not a static mesh, if that were the case, particles would agglomerate in single cells causing to treat multiple particles as one cell of homogeneous density, which defeats the purpose of having previously refined the masses with MUSIC by reducing the resolution at which the particles interact. This is why RAMSES uses adaptive mesh refinement(AMR), which means it also has levels of refinement that reduce the size of the cells of the grid. The way AMR works is shown in figure 2.3, which is a 2D representation of a RAMSES grid, initially the particles form a uniform distribution with perturbations(ICs) in a), once the CIC method is applied, the particles are shifted to their new positions in b). At this stage RAMSES verifies the number of particles in each cell, if it’s greater than the parameter m_{refine} (discussed in section 2.2.1), then RAMSES increases the level of the cell by dividing it in 2^{dim} cells, in the case of figure 2.3 the m_{refine} was considered to be 20 so no cells are refined in b). After checking the m_{refine} , RAMSES once again applies the CIC method to shift the particles to their new positions shown in c) and checks the refinement, since at this point there are cells with more than 20 particles, it increases the level for those cells as shown in d) where the corresponding cells are divided in red. Once this is done, the particles are shifted with the CIC method again, this process keeps on repeating until the normalized scale factor reaches $a = 1$ and the simulation ends, it is worth noting that there is a maximum level at which RAMSES refines the cells, this value will be discussed in section 2.2.1.

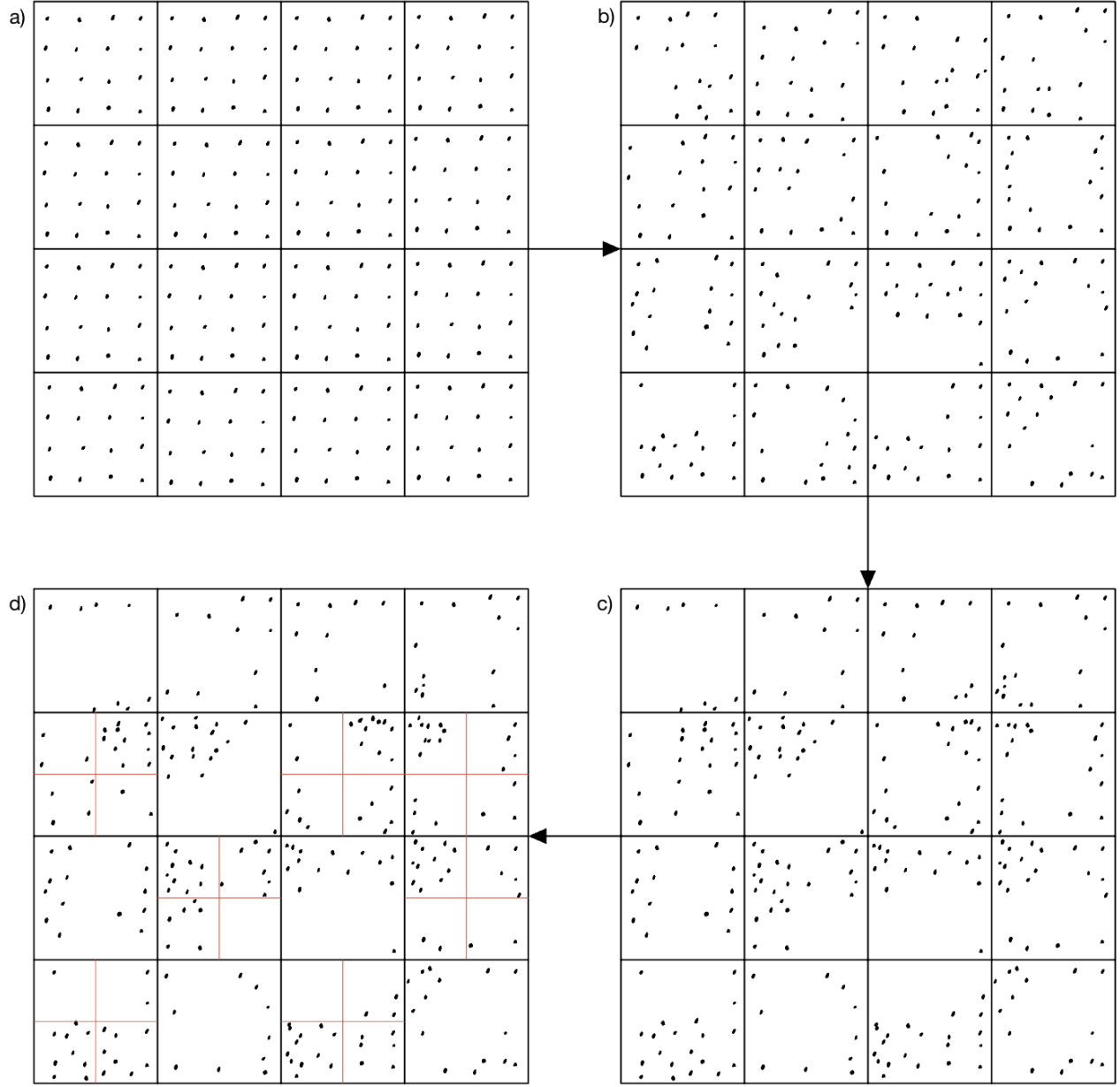


Figure 2.3: 2D representation of RAMSES AMR grid, a), b), c) and d) show the evolution of the grid containing the particles, and how the grid cells are divided due to the refinement.

Figure 2.3 represents the importance of the refinement in the resolution of the simulation, as seen in d), the refined cells are clearly not homogeneous and therefore treating them as an homogeneous density cloud would result in missing information of the gravitational potential in those regions.

One question that naturally comes up when looking at figure 2.3 is what happens to the particles on the borders of the box. If there are no cells at one side and a whole box of cells at the other, particles would tend to fall to the center of the box completely loosing the homogeneity of the cosmological box. The solution to this problem is the use of periodic boundary conditions which

mean that the box we are running our simulation in is interacting with itself on the edges, as is better illustrated in figure 2.4, the particles on an arbitrary “left” side feel the interactions with the corresponding “right” side. MUSIC(2.1) also takes advantage of periodic boundary conditions to center the zoomed region by shifting the particles in the ICs. The cosmological principle still has to hold in the box so the size of it has to be big enough which is why the *boxlength*(discussed in section 2.1.1) is typically set around 100 Mpc/h, this fact together with periodic boundary conditions ensures the homogeneity of the cosmological box.

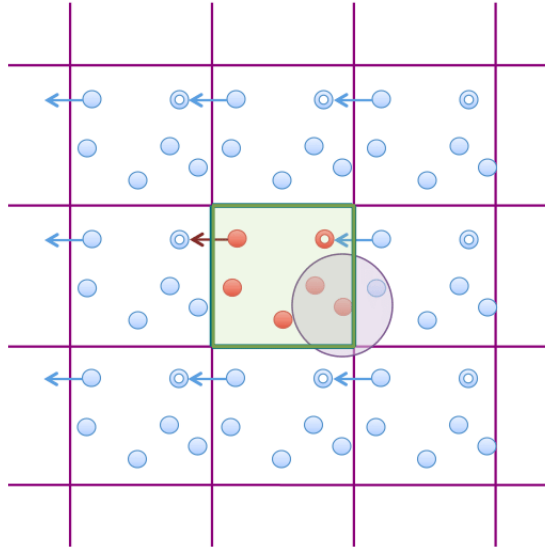


Figure 2.4: Illustration for the periodic boundary conditions. Image taken from F. Taslimi 2013[6].

So far I have discussed the way RAMSES treats the particles from the initial conditions, in section 2.2.1 I will discuss how this is implemented with the input file by the user to run a non zoom and a zoom simulation.

2.2.1 Input File

RAMSES receives an input file containing the instructions to run the cosmological simulation, this file contains 9 sections which are:

- Global Parameters
- Output Parameters
- Physical Parameters
- AMR Grid
- Boundary Conditions
- Poisson Solver
- Initial Conditions
- Hydrodynamics solver
- Refinement Strategy

Not all this sections were modified for the purpose of the present work, for the case of Hydrodynamics Solver and Physical Solver, the parameters involved control the hydrodynamical aspect of the simulation which involve baryons, the simulations ran so far contain only DM particles, therefore this parameters are not to be discussed now but will be relevant for the long term goal of this project when studying gas stripping of dwarf galaxies. The most relevant sections and parameters will now be discussed.

Global Parameters (&RUN_PARAMS)

This is where the parameters for the global control of the simulation are set. An important parameter to mention is *nrestart*, it takes integer values which tell RAMSES the number of output from which it has to start the simulation. This parameter is really useful because the user can choose the regions where the refinement should reach higher levels, *nrestart* is used when for whatever reason the simulation stops before finishing, then it is possible to continue from the last output generated. *nrestart* also could in principle allow the user to follow the region refining it as it evolves and restarting the simulation every time the refinement region is modified instead of having to refine the whole region from start to end positions.

AMR Grid (&AMR_PARAMS)

In this section, the global parameters of the AMR grid are set, these include the levels of refinement and the number of grids. The specific parameters for the AMR grid are set in the refinement strategy section.

The first relevant parameter for this section is *levelmin*, it takes an integer and it defines the base(coarse) grid for the whole simulation containing 2^{levelmin} octs. Octs are the unit of the Cartesian grid for RAMSES[7], the code recognizes one oct as 2^{dim} cells, it is implemented that way due to the type of data structure that RAMSES uses to manage the whole grid. The second parameter is *levelmax*, which defines the maximum level of refinement that the grid reaches, as was discussed for figure 2.3, it is important for this parameter to be large enough for the resolution not to be lost in the regions where there is a greater number of particles. If *levelmin*=*levelmax* we are left with a regular and static Cartesian grid.

Initial Conditions (&INIT_PARAMS)

Here the ICs generated from MUSIC are given to RAMSES through the *initfile*(#) parameter, it receives a path for each MUSIC level of refinement ICs file, the value replaced in # is the number of the file, in the case of no zoom from MUSIC this value is set to 1, but it takes as many files as MUSIC generates. It is possible to tell RAMSES the initial value of the redshift, but for MUSIC ICs it is not necessary since RAMSES reads it from the ICs files.

Output Parameters (&OUTPUT_PARAMS)

Every time RAMSES advances one time step it could generate an output (commonly called snapshots), if that were the case we would end up having too many outputs, this might be useful for some other studies such as generating DM halo merger trees. For this case that many outputs are not needed, with the output params we can tell RAMSES at which values of the normalized scale factor *a* to generate outputs with the parameter *aout*, and how many outputs are required with *noutput*. Also there is the *foutput* that tells RAMSES the frequency on which it should generate outputs, if this is set to 1 then it will generate 1 output for each time step. Other useful parameter of this section is *aend*, which tells RAMSES at what value of *a* it should end the simulation, this parameter could be useful to “pause” the simulation to change the refinement region to follow objects of interest.

Refinement Strategy (&REFINE_PARAMS))

In this section the specific refinement parameters of the grid are set. The most important parameter of this section is *m_refine*, as discussed for figure 2.3, this parameter defines the number of DM particles at which the cell is divided in 2^{dim} cells, for the case of simulations containing baryons this parameter is the mass inside a cell at which it is refined. It is important to note that this parameter is not necessarily fixed for every level, one can use a certain value for each level to refine higher levels with more or less particles. The second important aspect of the refinement parameters is the position, shape and size of the region to be refined, this is done by the parameters *x_refine*, *y_refine* and *z_refine* that set the center of the refinement region at every level of the grid, and *r_refine*, *a_refine* and *b_refine* that determine the size and shape of the refinement region. These last parameters are useful to refine specific regions such as individual halos, and follow them at each snapshot changing the refinement center.

As has been discussed in this section, RAMSES treats the cosmological box as a dynamic grid containing particles, that refines the sizes of the cells based on the number of particles or mass that each cell contains. Even though the most important aspect to increase the resolution in a region of the cosmological box is decreasing the mass of the particles using MUSIC ICs, it is also essential for zoom simulations, to correctly account for the distribution of those particles for them to correctly interact with each other and form the LSS, therefore it is important to set up carefully the AMR from RAMSES to have a representative cosmological simulation. The only way of doing this is by running many zoom simulations and keeping track of the amount of grids at each level of refinement, if there seems to be too many grids at the highest level then the *levelmax* parameter from RAMSES should be increased. It is important to get the *levelmax* right in zoom simulations, because otherwise the resolution would be lost in the region of interest. On the other hand, if *levelmax* is too high, the simulation can take an unnecessarily long amount of time to run.

In the case of a non zoom simulation, it is a matter of setting one level for the ICs and then let RAMSES refine the grid from the same level up to about double the levels of the initial grid and increase the *levelmax* if necessary. Since the non zoom simulation will not be deeply studied and it's purpose is to identify regions of interest, it is not crucial to get the refinement level exactly right.

2.3 Tools to Analyze Cosmological Simulations

After the simulations are done running, we are left with the RAMSES outputs, at this stage they are not ready to get plots because they are in binary format, this is why RAMSES comes with utility codes. The one used in the present work is **part2list**, which takes each RAMSES output and converts it in a list containing the properties of each particle in the cosmological box.

The properties obtained with this program are **part2list** are positions, velocities, masses and IDs. The IDs of the particles are essential for the purpose of this project because they allow us to follow individual particles through the different snapshots, which means that we can know at $a = 1$ which particles will form specific halos. This helps when defining the region that will be refined both in MUSIC and in RAMSES to run the zoom simulations focused on individual halos.

The second tool for analysis is AHF (Amiga's Halo Finder)[5], which as its name suggests, is a tool that identifies DM halos in cosmological simulations and for each snapshot finds the halos, calculates their properties and returns a table containing all the halos found. It does this by receiving the RAMSES outputs in a format called "gadget", the RAMSES outputs have to be converted to this format before giving it to AHF, this is done with a program called **ramses2gadget** that comes with AHF. For the purpose of this work, as the method for finding halos doesn't need to be modified for zoom simulations, AHF was used as a black box.

Once the information of the halos is obtained, an important theoretical analysis tool is the halo mass function, which is a histogram of the masses of the halos present in a simulation. The halo mass function plot allows to study the resolution of the cosmological simulation by giving the minimum mass halos present in the simulation, if the resolution is higher, then the masses of the particles are smaller and therefore smaller mass halos can be formed.

The last tool used for analysis is TOPCAT, which is a software for data analysis that can manage tables containing large amounts of rows and allows the user to plot, match tables and obtain subsets of data really fast and easily. With TOPCAT, one can visualize 3D plots of the positions of the particles to identify the regions of interest in the cosmological box, match tables based on particle IDs to follow individual halos in different snapshots, alter the tables to obtain subsets for specific positions to get an input file for MUSIC and plot halo mass functions.

Chapter 3

Simulations Run

After learning the way cosmological simulations work, two dark matter only simulations were run setting ICs with MUSIC and evolving them with RAMSES. The first simulation ran was a non zoom simulation, this was useful to get started and more familiarized with employing the codes, besides having an initial cosmological box to zoom in. The next step was to run a zoom simulation in that same box zooming in a big region, in both cases, the parameters used were the same as the ones ran by B. B. Thompson et al. 2022[8], therefore the first zoom was made in a void environment. In sections 3.1 and 3.2, the results from the non zoom and the zoom simulations respectively, will be presented and discussed.

3.1 Non Zoom Simulation

The non zoom simulation was ran using a boxlength of 100 Mpc/h, and with a starting redshift $z_{\text{start}} = 50$, as this is a DM only simulation the baryons are turned off, these last three parameters will remain the same for the zoom simulation. Since this is a non zoom simulation the *levelmin* parameter has to be equal to *levelmax*, the level for this simulation was chosen to be *levelmin* = 7, which means that MUSIC sets 2^{21} particles in the box initially. Once the ICs are ready, the file generated is given to ramses, where the *levelmin* of the grid matches the value of MUSIC's *levelmin*, and the maximum level of refinement of the grid is set to *levelmax* = 15, the particle refinement for the grid is set to be *m_refine* = 9*8.0, which means that for 9 levels of refinement from *levelmin*, the cells will refine when there are 8 DM particles inside them, the amount of levels can be set higher than the difference between *levelmin* and *levelmax* and RAMSES will ignore the refinement at levels that it doesn't reach. The output parameter for this simulation was *aout*=0.1, 0.25, 0.5, 0.75, 1.0, meaning that for each value of the normalized scale factor, an output was generated. Besides those outputs, RAMSES also generates an output containing the initial conditions at $a = 0.01$.

After running the simulation, obtaining the table with **part2list**, finding the halos with AHF and plotting the results using TOPCAT, 6 cosmological boxes were obtained, these are shown in figure 3.1. The cosmological boxes from figure 3.1 are presented in an $X - Y$ plane, for the Z coordinate, a sub-sample of each box was taken with $Z = 0.5 \pm 0.1$, so we are seeing "slices" of the boxes at the center, note that the output generated by RAMSES creates boxes whose coordinates are normalized, which is why the values of X, Y and Z go from 0 to 1.

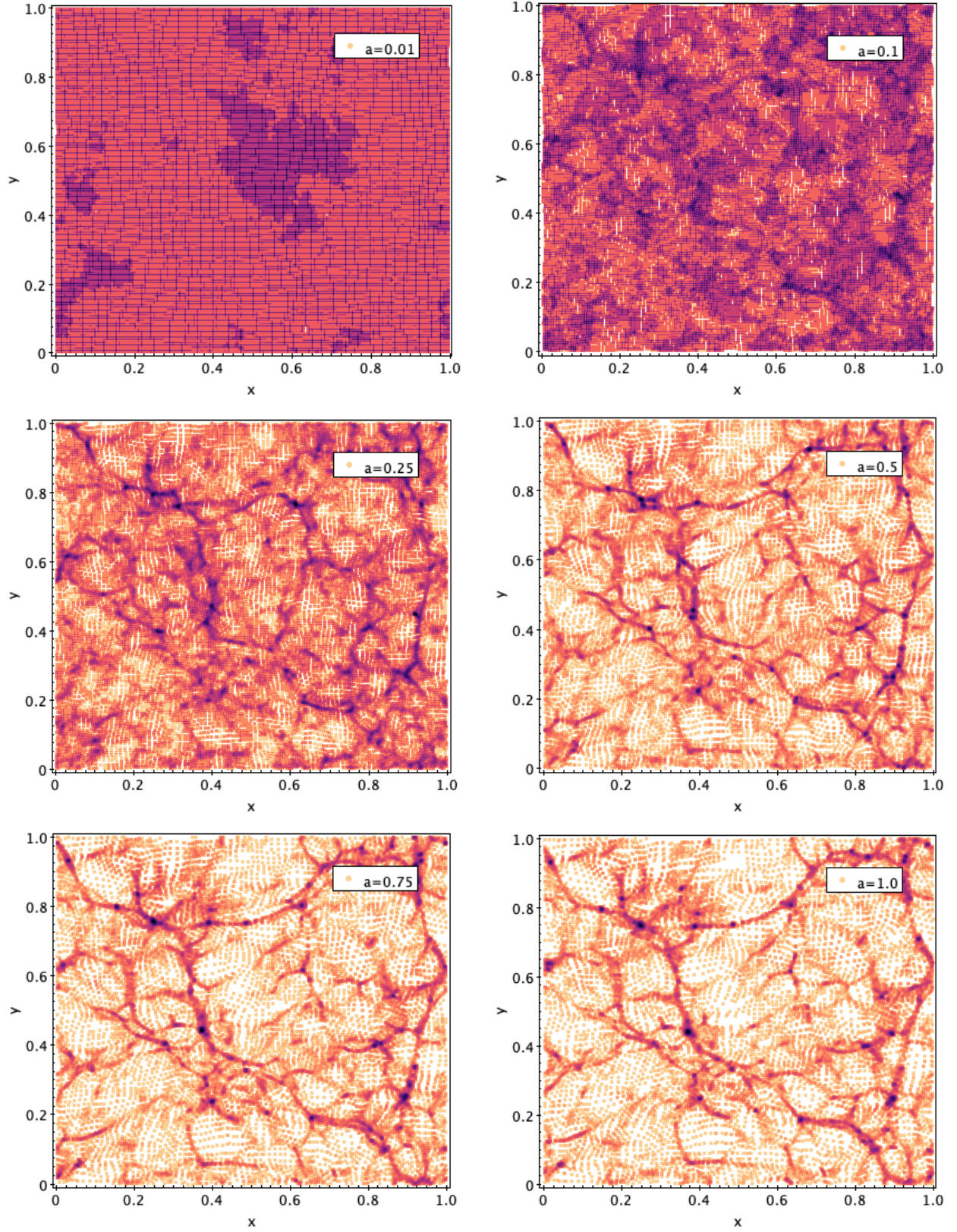


Figure 3.1: Cross section of the cosmological boxes at the coordinate $Z = 0.5 \pm 0.1$, the six density plots present show the evolution from $a = 0.01$ up to $a = 1.0$.

As the plots from figure 3.1 are density plots, the darker regions represent higher number density of DM particles, therefore it is possible to appreciate the LSS forming as a increases. For the first plot shown at $a = 0.01$, an almost uniform grid is presented from the ICs with some darker areas that show how MUSIC creates overdense regions when setting the ICs. As the box evolves, the LSS clearly forms, and by $a = 0.75$ it is possible to qualitatively see that the LSS has been almost completely formed. This last fact is backed by the halo mass function plotted in figure 3.2, from this plot we can see that by $a = 0.75$ almost all the final halos have already formed and only a few higher mass halos are missing.

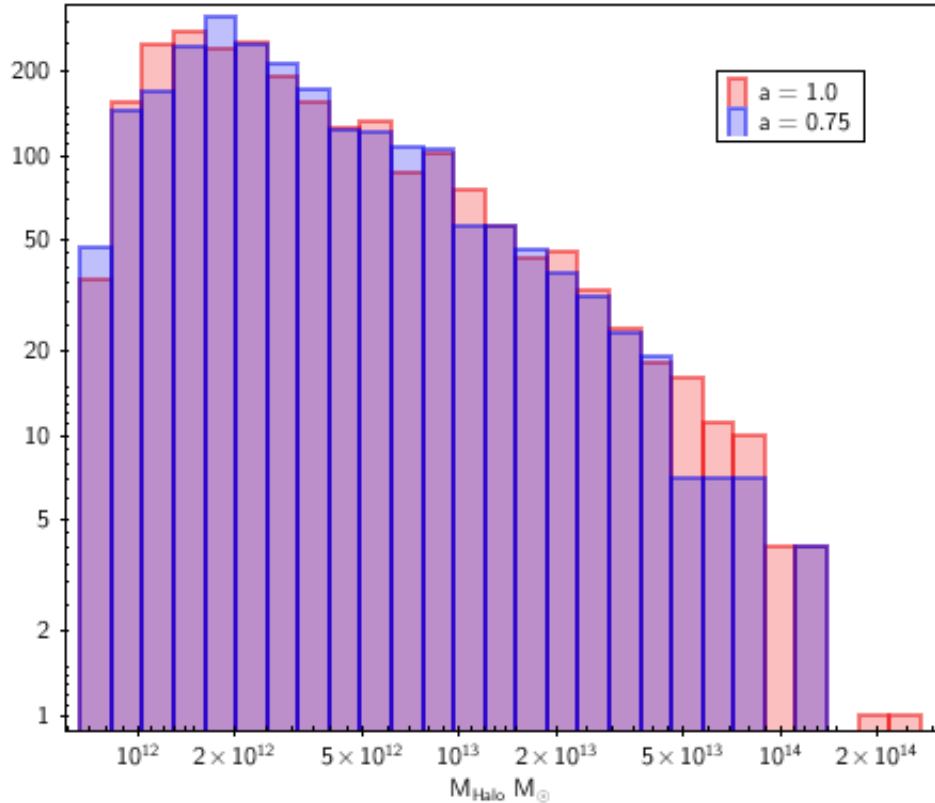


Figure 3.2: Halo mass function for the non zoom simulations at $a = 1.0$ in red and $a = 0.75$ in blue.

On the same line, we can plot the halo mass functions for the other values of a compared to $a = 1.0$ and obtain the plot from figure 3.3, where halo formation is clearly shown by first having low mass halos at $a = 0.25$, then increase the low mass halo numbers and start seeing higher mass halos by $a = 0.5$, which is exactly what is expected to happen. The first two snapshots are not plotted because at these values of a , AHF didn't recognize halos formed.

These last two plots were expected to present these behaviors in halo formation at low resolution, which is why we can say that the non zoom simulation was correctly run and it can now be used for a zoom simulation.

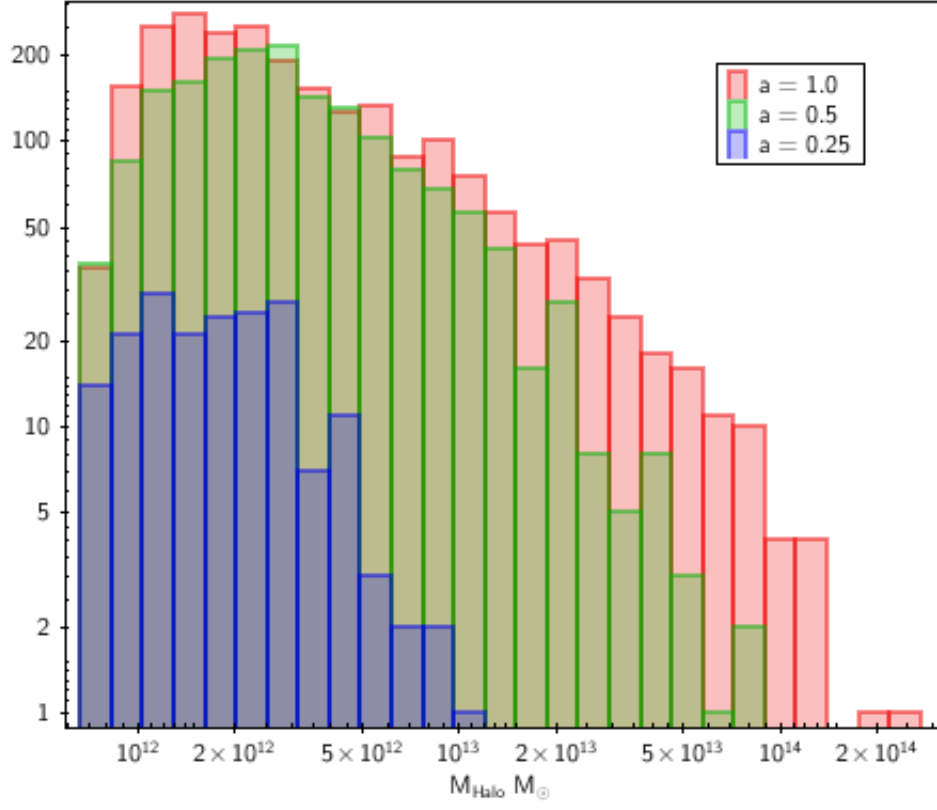


Figure 3.3: Halo mass function for the non zoom simulations at $a = 1.0$ in red, $a = 0.5$ in green and $a = 0.25$ in blue.

3.2 Zoom Simulation

For the zoom simulation, almost the same parameters as for the non zoom were used except for the *levelmax* which was set to 10 with a *padding* of 8, this results in a box of ICs containing 20353222 particles, which is around ten times the number of particles from the non zoom simulation. Then the RAMSES run uses the files generated with refinement from *levelmin*= 7 to *levelmax*= 16 and *m_refine*= $30 * 8.0$.

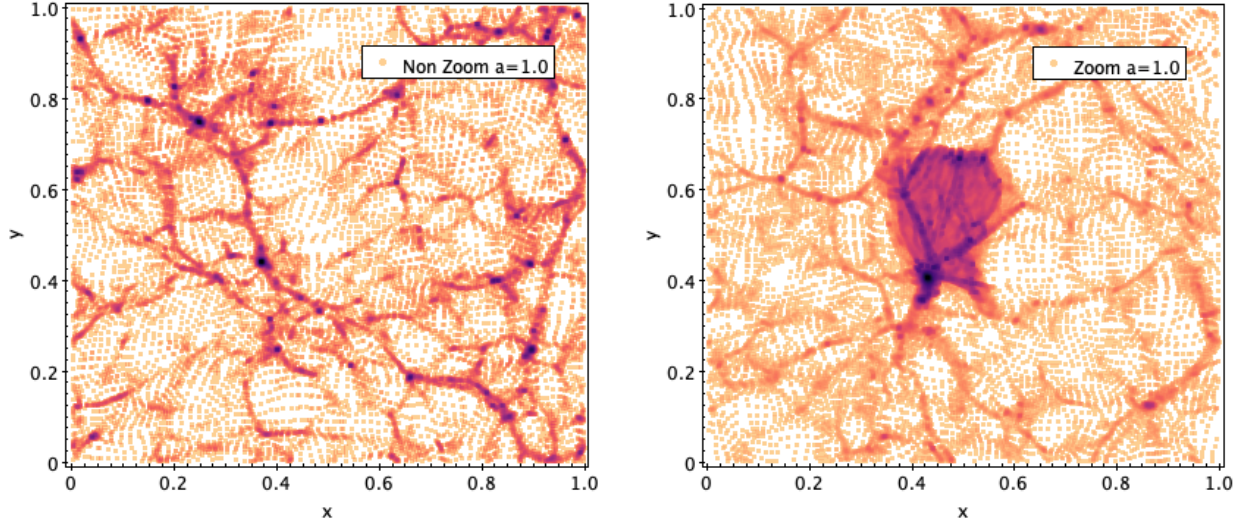


Figure 3.4: Cross section of the cosmological boxes at the coordinate $Z = 0.5 \pm 0.1$, the two density plots present show the non zoom simulation in the left, and the zoom simulation in the right, both at $a = 1.0$.

When comparing both the zoom and the non zoom simulations at $a = 1.0$ in figure 3.4, one can see clearly the zoom region where the number density of DM particles is higher than the rest of the box. To have an idea of refinement in a zoom simulation, in this box the refined region has a radius of refinement of 0.25 in units of the cosmological box, and this region contains 90% of the particles of the whole box, in other words only 10% of the particles in the box are level 7 mass particles. This also has implications in the halo mass function plot, since there are more particles with lower mass, it is possible to resolve lower mass halos that in the non zoom simulation would have been impossible to see since the masses of the particles themselves were higher than some of the halos formed in the zoom simulation. This last fact is clearly seen when comparing the halo mass functions of the non zoom and zoom simulations as shown in figure 3.5, where lower mass halos that doesn't exist in the non zoom simulation have formed in the zoomed one, this is a clear sign of the increased resolution in the zoom simulation. For future simulations to follow jelly fish candidates, the resolutions will be even higher, meaning that the halo mass functions will show halos of even lower masses.

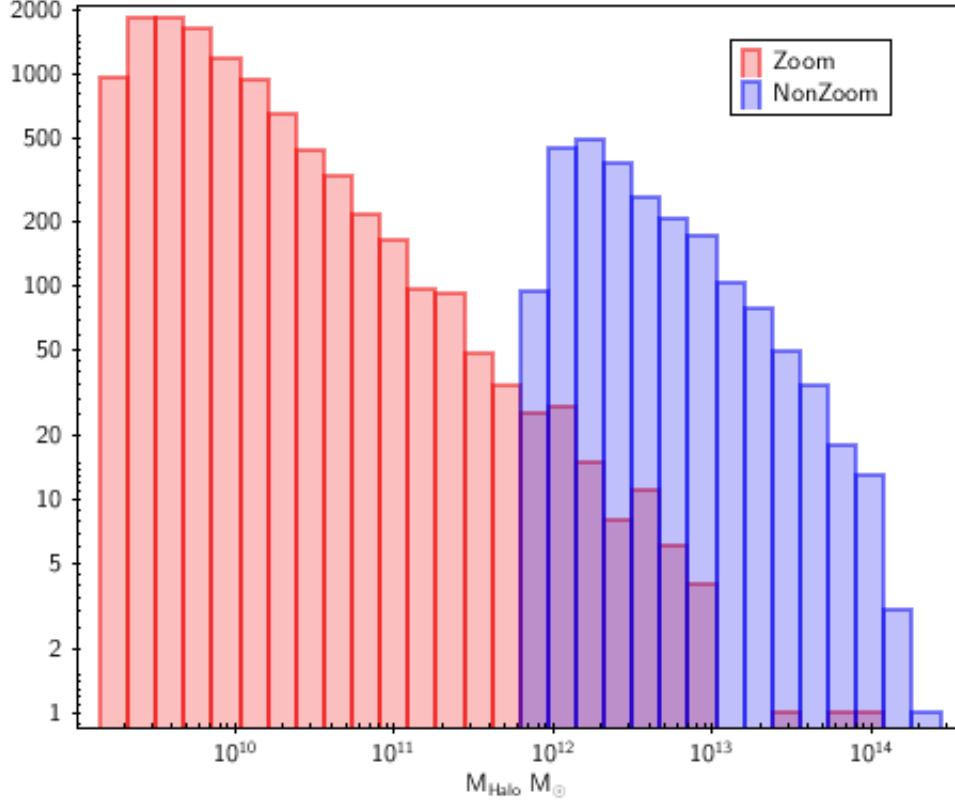


Figure 3.5: Halo mass function for the zoom simulation in red and non zoom simulation in blue.

The halo mass function is also plotted for the different values of a for the zoom simulation in figure 3.6, where the plots for $a = 0.25$, $a = 0.5$ and $a = 0.75$ were compared to the final $a = 1.0$, here we see that at lower a the low mass halos have higher number density than the final low mass halos, this differs from what is seen in figure 3.3 because we are looking at much lower masses. Comparing for masses of 10^{12} which is the lowest value in the non zoom simulation, the behavior of the halo mass function is the same for both simulations. Moreover, the low mass halos decrease in number when increasing a because they merge into other halos, leading to an increase in the high mass halos at $a = 1.0$.

The halo mass function plots show that lower mass halos are resolved for an increased resolution of the cosmological simulations as was expected. This means that the zoom has been successfully done to the cosmological box, therefore it is now possible to run zoom simulations on more specific regions of the box.

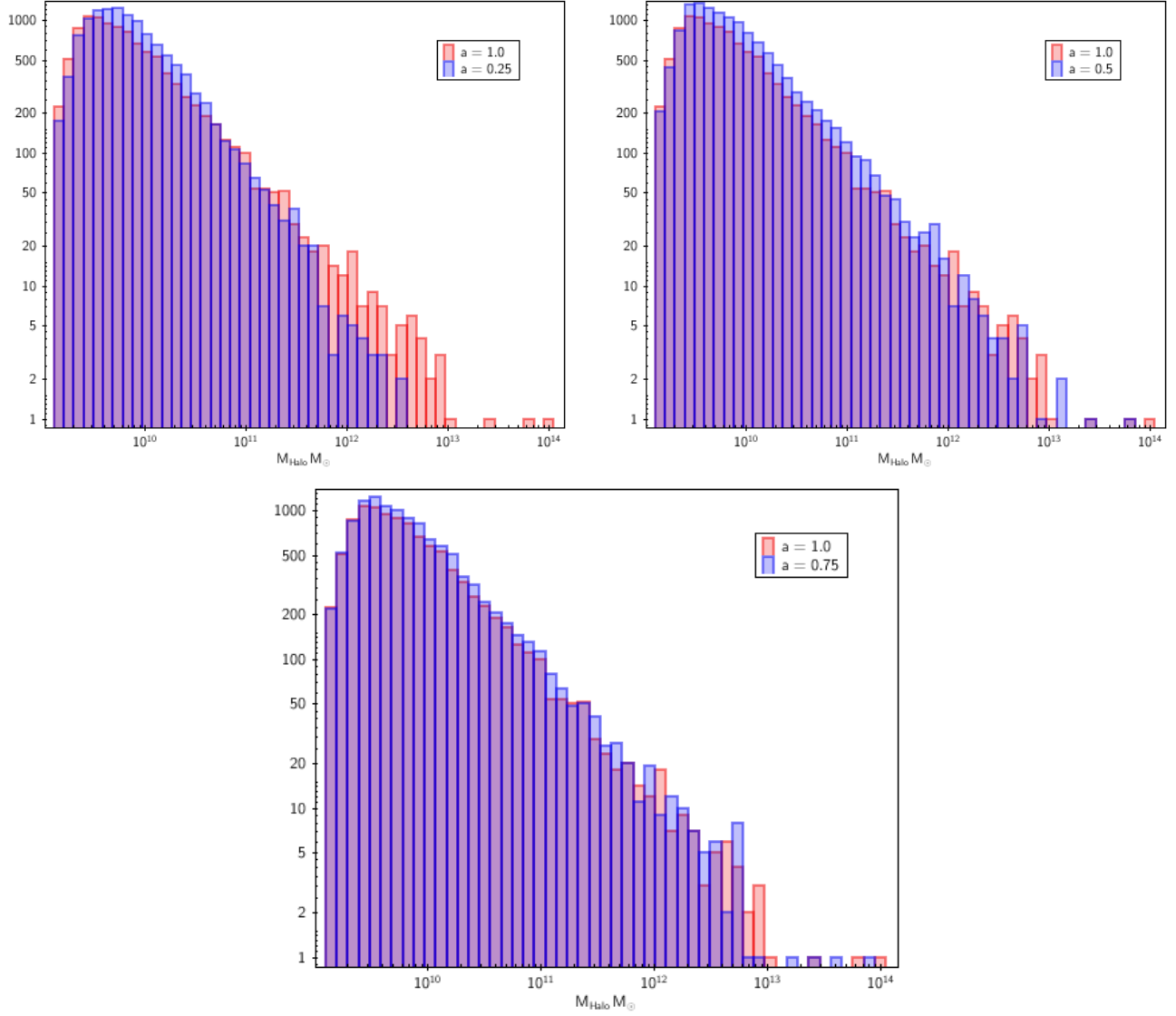


Figure 3.6: Halo mass function for the zoom simulation, comparing $a = 1.0$ in red, with $a = 0.25$, $a = 0.5$ and $a = 0.75$ all in blue.

Chapter 4

Conclusion

After studying, understanding, running and analyzing a non zoom and a zoom cosmological simulation, it has been verified by the halo mass functions that both simulations were run effectively and that the resolution was correctly increased, meaning that the zoom was accomplished and understood using the tools discussed in section [2.3](#).

Future Plans

The following step is to run DM zoom simulations focused on specific fast flow halos crossing void walls as discussed in chapter [1](#). Once those zooms are accomplished, it follows to understand how to run simulations with baryons and the RAMSES hydrodynamics solver turned on. After this, the more scientific long term goal motivated in section [1](#), is to run the simulations focused on the jelly fish candidates with gas and DM present to confirm if it is possible for RAM pressure stripping to occur near void walls, and form jelly fish galaxies.

Acknowledgements

I thank professor Rory Smith for the support and help he has given me to learn about cosmological simulations and the cosmological context on which these are being run. I also thank professor Antonio Montero for the discussions that helped me better understand the physics behind the initial conditions.

References

- [1] James E. Gunn and III Gott, J. Richard. On the Infall of Matter Into Clusters of Galaxies and Some Effects on Their Evolution. , 176:1, August 1972.
- [2] Oliver Hahn. *The MUSIC User’s Manual*. Kavli Institute for Particle Astrophysics and Cosmology.
- [3] Oliver Hahn and Tom Abel. Multi-scale initial conditions for cosmological simulations. *Monthly Notices of the Royal Astronomical Society*, 415(3):2101–2121, 08 2011.
- [4] Ying jie Peng, Simon J. Lilly, Katarina Kovač, Micol Bolzonella, Lucia Pozzetti, Alvio Renzini, Gianni Zamorani, Olivier Ilbert, Christian Knobel, Angela Iovino, Christian Maier, Olga Cucciati, Lidia Tasca, C. Marcella Carollo, John Silverman, Pawel Kampczyk, Loic de Ravel, David Sanders, Nicholas Scoville, Thierry Contini, Vincenzo Mainieri, Marco Scodeggio, Jean-Paul Kneib, Olivier Le Fèvre, Sandro Bardelli, Angela Bongiorno, Karina Caputi, Graziano Coppia, Sylvain de la Torre, Paolo Franzetti, Bianca Garilli, Fabrice Lamareille, Jean-Francois Le Borgne, Vincent Le Brun, Marco Mignoli, Enrique Perez Montero, Roser Pello, Elena Ricciardelli, Masayuki Tanaka, Laurence Tresse, Daniela Vergani, Niraj Welikala, Elena Zucca, Pascal Oesch, Umami Abbas, Luke Barnes, Rongmon Bordoloi, Dario Bottini, Alberto Cappi, Paolo Cassata, Andrea Cimatti, Marco Fumana, Gunther Hasinger, Anton Koekemoer, Alexei Leauthaud, Dario Maccagni, Christian Marinoni, Henry McCracken, Pierdomenico Memeo, Baptiste Meneux, Preethi Nair, Cristiano Porciani, Valentina Presotto, and Roberto Scaramella. Mass and environment as drivers of galaxy evolution in sdss and zcosmos and the origin of the schechter function*. *The Astrophysical Journal*, 721(1):193, aug 2010.
- [5] Steffen R. Knollmann and Alexander Knebe. AHF: Amiga’s Halo Finder. , 182(2):608–624, June 2009.
- [6] Farzaneh Taslimi. *End-functionalized rodlike colloid suspensions under shear flow*. PhD thesis, 10 2013.
- [7] R. Teyssier. Cosmological hydrodynamics with adaptive mesh refinement. A new high resolution code called RAMSES. , 385:337–364, April 2002.
- [8] B B Thompson, R Smith, and K Kraljic. Gas accretion and ram pressure stripping of haloes in void walls. *Monthly Notices of the Royal Astronomical Society*, oct 2022.
- [9] Frank C. van den Bosch. Theory of galaxy formation, 2020.

- [10] Benedetta Vulcani, Bianca M. Poggianti, Rory Smith, Alessia Moretti, Yara L. Jaffé, Marco Gullieuszik, Jacopo Fritz, and Callum Bellhouse. The relevance of ram pressure stripping for the evolution of blue cluster galaxies as seen at optical wavelengths. *The Astrophysical Journal*, 927(1):91, mar 2022.