Università degli Studi di Padova

Facoltà di Scienze MM.FF.NN.

Dipartimento di Astronomia

Halo-matter cross-correlation in cosmological simulations

Relatore: Prof. Giuseppe Tormen
Correlatore: Prof. Ravi Sheth

Laureando: Brunetto Marco Ziosi

Anno Accademico 2010-2011

“Try again. Fail again. Fail better.”
(Samuel Beckett)

“Stay hungry, stay foolish.”
(Steve Jobs)

Contents

Introduction

Gravitational lensing is a quite new and very powerful way of investigation of the large scale distribution of matter in the universe. Distant galaxies sometimes appear distorted, in a correlated way, because the foreground mass change the trajectory of the light reaching us. These distortions can be useful to map the mass distribution on large scales (Hayashi and White 2008). One of the methods to make use of this is to compute the cross-correlation between the cosmological structures and sub-structures with themselves or with the mass, and compare the resulting structure knowledge with the lensing observations.

The goal of this work is to compute the correlation function between the dark matter (DM) sub-haloes and the DM mass particles in the coordinate space. This is useful because it permits to check the theoretical prevision (Giocoli et al. 2010) of these quantities. Moreover the cross-correlation permits to compare different models of aggregation for the matter and gives a statistical tool to characterize a distribution of matter or the structures formed.

In this work the cross-correlation will be calculated on the result of some simulation: the GIF and GIF2, the Millennium Simulation, the Millimillennium Simulation and Millennium II Simulation. There are two ways to calculate the cross-correlation, one in the Fourier space, the other in the configuration space. In the Fourier space one has to create a grid over the particles distribution and then calculate the power-spectrum (cross-spectrum), from which he can obtain the CC. This method is subject to *shot-noise*. In case of a *sparse* distribution of matter in a big space (box) the shot-noise could be big enough to make the results useless. This happens especially in the case that we want to investigate small scales: we would need a fine grid, which dimension make its representation in memory difficult, almost empty because of the small number of particles to check. Moreover to obtain the correlation from the spectrum it is necessary to deconvolve and anti-transform it and this operation lead to a loss of information, especially on the small scales. The second way, that we will follow is to calculate the correlation in the configuration space counting the pairs at each distance.

Some considerations and tests lead us to choose Python as programming language and a binary tree as data structure. Some optimization was done over the original library, in particular for what regards the inclusion or exclusion of nodes with some characteristics, based on the cache statistic and in case of auto-correlation. In addition the slow distance calculation was substituted with a faster function in Fortran imported as a shared library with *f2py*.

Some tests was made both for the speed of the code and for the correctness of the results and some tuning was done to understand the best parameters (leafsize, strategy, . . .) to be used.

This dissertation is structured as follow: Chapter 1 introduce the standard model of cosmological structure formation. In Chapter 2 we will treat the two point correlation function, why it is important, how to calculate it and its connection with the power spectrum. The following chapters are dedicated to the presentation of the *halo model* (Chapter 3) and to the cosmological simulations (Chapter 4). In chapter 5 we will introduce and explain what *kd*-tree are and how they were developed. Chapter 6 treats the development of the main code used in this work. Chapter 7 is dedicated the results of our work. In Chapter 8 we presents some conclusions and propose further works.

Chapter 1

The Standard Model of Structure Formation

A little introduction on why the universe is as we see it.

1.1 Cosmological Principle

As of today, *Standard Cosmology* is based on the so called *Cosmological Principle*. This principle was introduced in the 1920's by Einstein, while he was developing his General Theory of Relativity, and before any astronomical observation could either confirm or refute it. The Cosmological Principle states that, on scales large enough, the universe is spatially homogeneous and isotropic, i.e. it has the maximum number of symmetries[1]. This assumption translates into powerful constraints on the viable cosmological models. In this way we can develop simplified models of the Universe starting from Einstein's equations, otherwise too difficult to be solved for an arbitrary matter distribution. The hypothesis of a homogeneous and isotropic Universe was confirmed by Penzias and Wilson in 1965 with the observation of the Cosmic Microwave Background (CMB), the

[1] In a more formal way, the Cosmological Principle states that the space-time has a maximally symmetric tridimensional sub-space.

remnants of the hot and dense past of the Universe. Its spatial features convinced the astronomers that the Cosmological Principle was valid.

To build a Universe model upon this idea and with tools offered by General Relativity we have now to consider the geometrical properties of the Universe as a fluid. The geometrical properties of a space are described by its metric, and the more general one for an homogeneous and isotropic space is the Robertson-Walker metric:

$$\mathrm{d}s^2 = (c\,\mathrm{d}t)^2 - a^2(t)\left[\frac{\mathrm{d}r^2}{1-kr^2} + r^2\left(\mathrm{d}\theta^2 + \sin^2\theta\,\mathrm{d}\varphi^2\right)\right] \tag{1.1}$$

where r is the comoving coordinate and $a(t)$ is the *scale factor*. Einstein's equations

$$R_{\mu\nu} - \frac{1}{2}g_{\mu\nu}R = \frac{8\pi G}{c^4}T_{\mu\nu} \tag{1.2}$$

with the energy-momentum tensor

$$T_{\mu\nu} = \left(p + \rho c^2\right)U_\mu U_\nu - p g_{\mu\nu} \tag{1.3}$$

link the geometric properties of space-time with the energy momentum tensor that describes the content of the universe.

In a space described by the Robertson-Walker metric, with mass-energy density at rest given by ρc^2 and pressure p the equations (1.2) reduce to the Friedmann's equations:

$$\ddot{a} = -\frac{4\pi G}{3}\left(\rho + 3\frac{p}{c^2}\right) \tag{1.4}$$

$$\dot{a}^2 + kc^2 = \frac{8\pi G}{3}\rho a^2 \tag{1.5}$$

where $a = a(t)$ is the scale factor and t is the proper time. From Friedmann's equations we can extract the *spatial curvature*

$$k = \left(\frac{\dot{a}}{c}\right)^2[\Omega(t) - 1] \tag{1.6}$$

with the *critical density* given by

$$\rho_c = \frac{3}{8\pi G}\left(\frac{\dot{a}}{a}\right)^2 \tag{1.7}$$

and *density parameter*

$$\Omega(t) = \frac{\rho}{\rho_c}. \tag{1.8}$$

CHAPTER 2

The Two Point Correlation Function

Here we present the two point correlation function, the motivation on why to use it and the best way to estimate it.

The modern technologies permits huge observations (big surveys, deep observations, . . .) as well as numerical simulations greater than ever before and so we have now a lot of data with the positions of the point representing DM particles, galaxies or halo centers or whatever is needed. What we need is a way to characterize this distribution in order to know the interesting properties of it and to compare this distribution with another, or with observative data or with analytic models. If we are interested in the clustering properties of our data, one of the most useful and most widely used tools is the *spatial correlation function*, in this case, the *two-point correlation function - TPCF* $\xi(r)$. In short, it measures the probability of finding a pair of objects (mass points, halos, sub-halos, galaxies) separated by a certain distance r. In a discrete case, here we have a discrete distribution of points, we have on average n points per volume unit. Following Peebles 1980 the probability to find one point in the infinitesimal volume dV is

$$\mathrm{d}P = n\,\mathrm{d}V. \tag{2.1}$$

If $\mathrm{d}V$ doubles, $\mathrm{d}P$ doubles too.

The mean number of points in a finite volume V is

$$N = nV. \tag{2.2}$$

The two point correlation function of this distribution is defined by the joint probability of find two point with separation r:

$$\mathrm{d}^2P = n^2\,\mathrm{d}V_1\,\mathrm{d}V_2\left[1 + \xi\left(r\right)\right]. \tag{2.3}$$

Because we are dealing with an homogeneous and isotropic universe, ξ depends only upon r. The probability is proportional to $\mathrm{d}V_1$ and $\mathrm{d}V_2$ because if one of these doubles, it doubles also the probability. In case of Poissonian process on a uniform random distribution the two probabilities are independent, so

$$\mathrm{d}^2P = n^2\,\mathrm{d}V_1\,\mathrm{d}V_2 \tag{2.4}$$

that is $\xi \equiv 0$. Then we can see that $\xi\left(r\right)$ represents the excess (or defect) of clustering of a distribution compared to a uniform Poissonian one. If $\xi\left(r\right) > 0$ we have a positive correlation, in case of $-1 < \xi\left(r\right) < 0$ we have anti-correlation. In any case it must be $\xi\left(r\right) > -1$.

2.1 Correlation estimators

While the role of the TPCF is central, estimators for extracting it from a set of spatial points are confusingly abundant in literature. The reason is partly due to the lack of a clear criterion to distinguish between the estimators. As described in Szapudi and Szalay 1998 the simple estimator

$$\xi\left(r\right) = \frac{DD}{RR} - 1 \tag{2.5}$$

was widely used, where DD denotes the number of pairs at a given range of separations and RR the number of random pairs, with the same separation, generated over a similar area as the data. In Landy and Szalay 1993 a new estimator is proposed:

$$\xi_{LS}\left(r\right) = \frac{DD - 2DR + RR}{RR} \tag{2.6}$$

CHAPTER 3

HALO MODEL

Here we learn about the winning method in the last years to describe the clustering of the dark matter structures and of its extension by Giocoli et al. 2010.

As presented in Cooray and Sheth 2002, the Halo Model (HM) is one of the possible approach the description of the non-linear gravitational clustering of the DM, characterized by having all the mass associated with virialized dark matter haloes. The HM has its origins in a paper by Neyman, Scott *et al.*, interested in describing the spatial distribution of the galaxies. They thought that it would be useful to think of the galaxies distribution as being made up of distinct clusters with different sizes. Galaxies are discrete objects so the initial work described the statistical properties of a distribution of discrete points. This description require to know some parameters: the distributions of cluster sizes, the distribution of points around the cluster center and the description of the clustering of the clusters but none of this elements were known at that time.

Now we know that much of the mass of the Universe is made of DM, initially rather smoothly distributed and that galaxies are biased tracers of the DM distribution.

The initial fluctuation field was very close to a Gaussian random field and there were developed a perturbation theories from linear regime to higher order to describe the gravitational clustering from those initial fluctuation. These describe the evolution and mildly non-linear clustering of the DM but they fail when the clustering became highly non-linear (typically on scales smaller than a few Mpc). In perturbation theory we also don't have a rigorous framework for describing the differences between the clustering of the galaxies and that of the DM.

Numerical simulations of the large scale structure clustering were developed to study the non-linear evolution of the the DM distribution. These show that initially smooth DM distribution evolves into sheets, filaments and knots. The denser knots are called DM haloes. Simulations also show that the halo abundance, spatial distribution and internal density profiles are closely related to the properties of the initial fluctuation field. If we consider these haloes as the Neyman&Scott's clusters, the formalism would provide a way to describe the spatial statistics of the dark matter density field from the linear to the non linear regimes.

3.1 Halo structures and galaxies

As stated in Giocoli et al. 2010, for example, to know the non-linear dark-matter power spectrum ($P_{DM;NL}(k)$) is important for understanding the large scale structure of the Universe. Giocoli *et al.* analytically model the dark-matter power spectrum ($P_{DM}(k)$) and the cross-power spectrum (CPS) of the DM with the DM haloes. This is an extension to the halo model formalism and includes realistic substructure population within individual DM haloes and the scatter of the concentration parameter at fixed halo mass. This extension increases the predicted power on the small scales and it is crucial for proper modeling the cosmological weak-lensing signal due to low-mass haloes. The halo model approach can be improved in accuracy increasing the number of ingredients calibrated from the n-body simulations and from large galaxy redshift surveys which made possible accurate studies of the large-scale cosmic structures.

Galaxies are believed to form and reside in DM haloes that extend much beyond their observable radii. According to the standard scenario of structure formation galaxies with dissimilar features reside in different DM haloes and have experienced different formation histories. Some of them are located at

White 2008). The cross-correlation between halo centers and mass ξ_{hm} is the spherically averaged halo density profile averaged over all the haloes in the dataset. Its shape is composed by two parts, the one-halo and two-halo terms presented in the previous sections. They are dominated by the particles within the same halo or in different haloes respectively.

It can be also easily shown that ξ_{hm} on large scales follows closely the mass auto-correlation function with a mass-dependent offset in amplitude, the *halo bias factor* $b\left(M\right)$.

So we can write:

$$\xi_{hm}\left(r;M\right)=\begin{cases}\xi_{1h}\left(r\right) & \text{if}\,\xi_{1h}\left(r\right)\gg\xi_{2h}\left(r\right)\\ \xi_{2h}\left(r\right) & \text{if}\,\xi_{1h}\left(r\right)\ll\xi_{2h}\left(r\right)\end{cases} \tag{3.11}$$

with

$$\xi_{1h}\left(r\right)=\frac{\rho_{halo}\left(r;M\right)-\bar{\rho}_m}{\bar{\rho}_m} \tag{3.12}$$

$$\xi_{2h}\left(r\right)=b\left(M\right)\xi_{lin}\left(r\right). \tag{3.13}$$

The one-halo term has been studied extensively and it is well fitted by the NFW profile or some modified forms of it.

As presented before, regarding the substructures-matter cross correlations, the contributions are:

- for the 1H term the correlation with the substructure mass, the mass in other substructures of the same halo and the correlation with the smooth component
- for the 2H term we have only two contributions, the one from the correlation with the smooth component and the one from the correlation with the clump one.

In this work we do not distinguish the contributions from the smooth and the clump component, they appear as a single contribution.

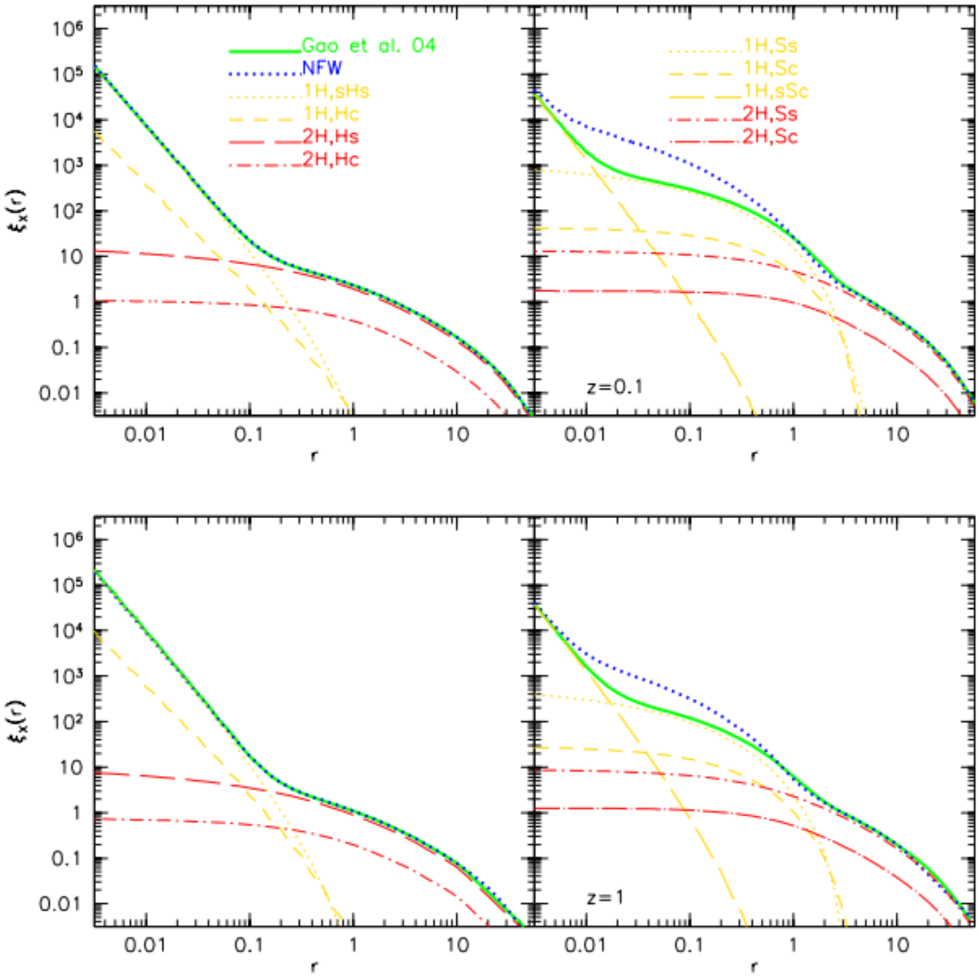

Figure 3.1: Halo and subhalo-mass cross-correlation at redshift $z = 0.1$ (top) and $z = 1$ (bottom) from Giocoli et al. 2010. In each panel can be seen the contribution of haloes and substructures and mass, respectively.

Chapter 4

Simulations

In this chapter we will take confidence with the concept of simulation and we would have a look to one of the most important simulation of the astrophysics done till now.

A good introduction on what is a simulation and its implication on the modern scientist trade is provided by Karniadakis and Kirby 2003. In particular the focus on how the new technologies changed the scientist workflow and skills. Indeed, although until not so many years ago the work of the scientists was connected with the observations in a laboratory (or at the telescope) and/or with paper and pencil. Some rudimentary machines were available to help the counts sometimes. In few years those machines became powerful enough not only to became a fundamental part of the analysis but also to permit the scientist to recreate a numerical model of the system under study and let it evolve into the computer.

The modern scientist then often spend more and more time in front of a laptop, a workstation, or a parallel supercomputer and less and less time in the physical laboratory or in the workshop. Sometimes the old approach of "cut-and-try" has been replaced by "simulate-and-analyse in several disciplines, from the astrophysics, to particle physics, to biology. As a side effect, the modern scientist must be able to use the new tools, so he have to join together the knowledge in his field of research with the computer programming.

In the classical scientific approach, the physical system is first simplified and set in a form that suggests what type of phenomena and processes may be important, and correspondingly what experiments are to be conducted. In the absence of any known-type governing equations, dimensional inter-dependence between physical parameters can guide laboratory experiments in identifying key parametric studies. The database produced in the laboratory is then used to construct a simplified "engineering" model which after field-test validation will be used in other research, product development, design, and possibly lead to new technological applications. This approach has been used almost invariably in every scientific discipline, i.e., engineering, physics, chemistry, biology, etc.

The simulation approach follows a parallel path but with some significant differences. First, the phase of the physical model analysis is more elaborate: the physical system is cast in a form governed by a set of partial differential equations, which represent continuum approximations to microscopic models. Such approximations are not possible for all systems, and sometimes the microscopic model should be used directly. Second, the laboratory experiment is replaced by simulation, i.e., a numerical experiment based on a discrete model. Such a model may represent a discrete approximation of the continuum partial differential equations, or it may simply represent a statistical representation of the microscopic model. Finite difference approximations on a grid are examples of the first case, and Monte Carlo methods are examples of the second case. In either case, these algorithms have to be converted to software using an appropriate computer language, debugged, and run on a workstation or a parallel supercomputer. The output is usually a large number of files of a few Megabytes to hundreds of Gigabytes, being especially large for simulations of time-dependent phenomena. To be useful, this numerical database needs to be put into graphical form using various visualization tools. Visualization can be especially useful during simulations where interactivity is required as the grid may be changing or the number of molecules may be increasing.

The question is if this is a new science, and how one could formally obtain such skills. Moreover, does this constitute fundamental new knowledge or is it a "mechanical procedure" an ordinary skill that a chemist, a biologist or an en-

Chapter 5

Trees

In this chapter we will understand what a kd-tree is, why it was created and the basics of the kd-tree optimization.

Pairwise distance computations are of fundamental importance in many fields, not only in astronomy and cosmology. They are often used in machine learning, graphics computation and so on. A particular class of this type of problems is called *all-query*, *all-point-pairs* or *N-body*-like problems (Ram et al. 2009, Gray and Moore 2001). In particular, as we stated above, the two-point correlation function can be thought of roughly as a measure of the clumpiness of a set of points and it is easily defined as the number of pairs of points in a dataset which lie within a given radius r of each other.In this type of problems we have the interaction between a reference of two sets of points of size $O(N)$. The direct solution requires a total running time of $O\left(N^2\right)$. This type of dependence is often too high to permit the computation. The general approach usually followed to solve this problem is to reduce the number of distance computation. One of the most famous problem that need this approach is the *nearest neighbours* problem (NNP). Indeed one of the first appearance of *kd*-tree was in 1977 by Freidman, Bentley, and Finkel 1977 to solve the "best matches", or nearest neighbours problem. The problem is to find, among big datasets, the m closest matches or nearest neighbours to a given query record according to some dissimilarity or distance measure. Formally, given a dataset of N data-points (each described by K keys) and a dissimilarity measure or distance D we have to find the m matches closest to a query point (non necessary in the dataset). The most famous problem of this kind is the so called "post-office problem"[1]. The solution proposed by Freidman, Bentley, and Finkel 1977 require a computation proportional to $kN \log N$. The expected number of points examined in each search is independent of the dataset size and the expected computation to perform each search is proportional to $\log N$.

Structures use for associative searching

The first, rough, technique they present for solving the NNP is the *cell* methods. The k-dimensional key space is divided into small identically sized cells. A spiral search of the cells from any query record will find the best matches of that record. This is extremely costly in terms of space and time, especially when the dimensionality of the space is large.

The key point, in every strategy, is to minimized the number of record examinated. In Freidman, Bentley, and Finkel 1977 the authors quickly presents some of the strategies, including clustering techniques (using the triangle inequality), the formation of a projection of the records onto one or more keys keeping a linear list of these keys (this does not require to satisfy the triangle inequality). The expected computation required for the last strategy is proportional to $km^{1/k}N^{1-1/k}$. Other ways use binary keys with the Hamming distance[2] or the Voronoi diagram[3] in the special case of only two keys and Euclidean space. The last case can search for best matches in worst case of $O\left[(logN)^2\right]$ after a dataset

[1] The post-office problem is a problem presented by Donald Knuth in The Art of Computer Programming in 1973. The problem is about assigning to a residence the nearest post office among all.

[2] In information theory, the Hamming distance between two strings of equal length is the number of positions at which the corresponding symbols are different. Put another way, it measures the minimum number of substitutions required to change one string into the other, or the number of errors that transformed one string into the other. For example the Hamming distance between "toned" and "roses" is 3.

[3] In mathematics, a Voronoi diagram is a special kind of decomposition of a metric space determined by distances to a specified discrete set of objects in the space, e.g., by a discrete set of points.

Chapter 6

Developing a Python correlation code

The goal of this work, as we previously stated is to obtain the cross-correlation between the halo centers and the DM, on scales from 10 kpc/h up to 5 Mpc/h. To achieve this target we had to build some tools not available, for example an easy way to count the couples as fast as possible and some functions to read the initial data and to store the data read in a comfortable format.

The first thing to do was to decide which language to use. The candidates were

- well-known Fortran, compiled, probably the most tested and the fastest language for scientific computations
- C/C++, compiled, maybe the most used language in the world, fast, low level, and with a not-so-comfortable syntax
- Python, interpreted, a modern, flexible and widely used language by Guido Van Rossum

We have decided to use Python.

6.1 Python

The first objections one can do is that I needed a fast way to do computation, and Python is not famous for its speed. It is interpreted, high level and most of the existing code in astrophysics is Fortran or, at least, C. A `for` loop in Python takes order of magnitude more time respect to C or Fortran. Despite of all of these considerations and other, more or less technical, the use of Python is nevertheless growing fast in the scientific community. Why?

Flexibility

Python is a general purpose, object and aspect oriented language, so it permits a lot of different styles of programming and to implement own code following many different design patterns. It doesn't have the strict and innatural syntax of C/C++ nor the obsolete style of Fortran. In Python everything is an object, with its methods and attributes. In Python its easy to cover the entire workflow, from the data analysis to the plot of the results with only one language.Because of its modularity it permits an easy integration, improvement and reuse of the code.

Fast developing

It is modern and flexible syntax and design permits a fast develop and debug because one can test in the interpreter the pieces of code. The libraries are self-documenting and the *introspection* permits to easy understand what is happening or how a tool works. Introspection means that you can explore an object interactively, it is possible to ask the properties, the values or any important information about it.In Python no `segmentation fault` is possible: every error (in Python you call them *exception*) is managed automatically or following the user instruction.

Libraries

Python itself is usually not fast enough for a massive computational application, both because it is interpreted and because of its design. For some things (list comprehension, . . .) is quite fast, usually enough for the average user needed. It is really not fast enough for the purpose of this work. Because it is a general

- some radii are greater than the maximum distance between the nodes: for these radii the total number of possible couples (including all the subtrees inside the nodes) are added to the corresponding cells in the array containing the counts as function of the radius and then these radii are dropped

- some radii are shorter than the minimum distance: this radii are dropped

- some radii intersect the nodes: for these radii the nodes are opened

Now we have some radii for which the two nodes must be opened. If the two nodes are inner nodes, each node is divided and the counting routine is recursively called on any combination of the sub-nodes. If instead the two nodes are leaves, the code compute all the possible distances between the nodes (taking care of saving computation in the self-correlation case). Then the distances are sorted and using `numpy.searchsorted()` the code calculates for each radius how many couples are separated by a distance shorter than that radius and add them to the result array.

Counting strategies

In order to achieve better performance we try different approaches to open the leaves. I particular, the differences between the strategies tested were if to maintain or not the square roots, if to use a linear or logarithmic scale for the radii and if to use the sort or another method for counting the distances. The winning strategy is to drop the square roots, to use a linear scale (but, the radii are logarithmically equally spaced) and to maintain the distance counting using `numpy.sort()` and `numpy.searchsorted`.

Searchsort and alternate distance counting

The original version of the code calculates the distances between all the possible couples between the two nodes and sort the resulting array. Then, the `numpy.searchsorted` function identifies the index where to insert each radius to maintain the array ordered, that is the number of distances shorter than the radius. The other way consists in to divide (using an integer division) the distances

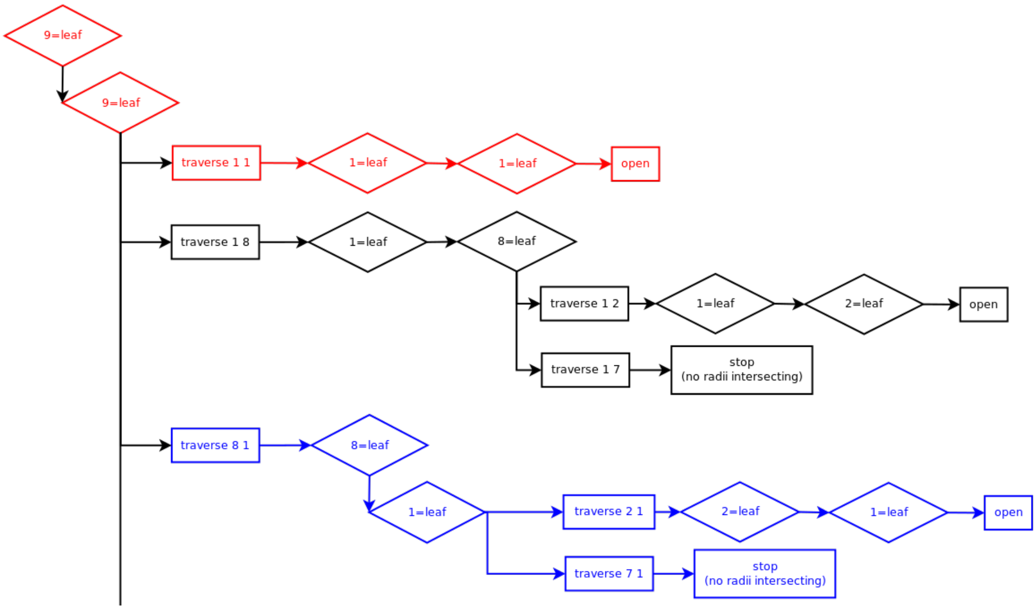

Figure 6.3: A simplified example of how an autocorrelation traverse works. This is the beginning of a traverse. In this case the two trees are traversed up-down, starting from the root node. The rhombs represents an `if` statement, for example when the code check if a node is a leaf. The square are the action to be taken, e.g. start a traverse or stop. The red branches are the non redundant actions, the black and the blue are the redundant actions that the code recognize and perform only once.

array by the bin length (the bins must be all equals): the result represents how many times the distance is greater than the bin spacing, thus it is the index of the bin to which we have to add 1.

Distances

The two most time consuming part of the code are traversing the tree and open the leaves. In both cases is necessary to calculate distances, and in fact timing the code shows that most of the time is spent in the routine. After some tests the solution was to create an ad-hoc Fortran library containing few lines that

Chapter 7

Results

7.1 GIF galaxy-galaxy auto-correlation

The first test we made to check the code was the computation of the galaxy-galaxy correlation on the GIF data. The data was downloaded from the GIF site `http://www.mpa-garching.mpg.de/Virgo/data_download.html` and the original `ASCII` data were converted to the `HDF5` format to be compatible with the code and the other data. The distance range we sample was between 10 kpc/h and 10 Mpc/h. The lower limit is because of the softening length, the upper limit was enough to sample the 2H term. It took about 16 seconds to compute the auto-correlation between 15445 galaxy centers on Uno. Because of the geometry and clustering properties of the dataset we chose to use the same number of random as the data particles. There was no need to sample very small scales. This particular data was preferred because of the small computational effort needed and the big literature comparison available. A χ^2 test on the result shows that our result is compatible with the literature. We can compare these results with the results obtained by Diaferio et al. presented in figure 7.2. We can see that the results of our code are in good agreement with the Diaferio, White, and Kauffmann 1999 results. We would expect a slope of about -1.8 and we find -1.7 with an unweighted χ^2 fit.

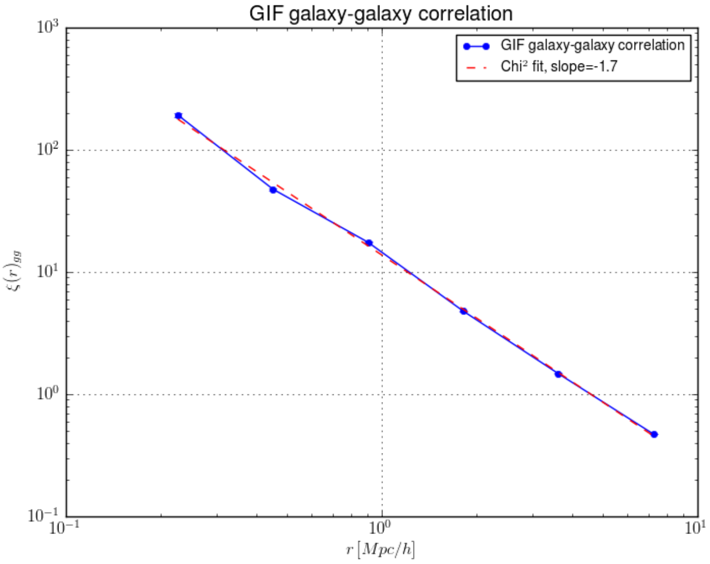

Figure 7.1: GIF galaxy-galaxy correlation.

7.2 GIF2 matter-matter auto-correlation

Because of the success of the first test we decided to test the code on the matter-matter two point correlation function. In this case we chose the GIF2 simulation data to have better mass and force resolution. The GIF2 particles data were already available on our servers in Fortran binary files. We convert the original data in a set of 22 `HDF5` files containing the particles positions sorted on the x coordinate. Every file contains 5 Mpc/h in x and about 2.5×10^6 particles. The computation was performed in parallel using 22 CPUs on Pico and correlating each file with itself and the two following to reach the 10 Mpc/h of distance for the correlation. The minimum distance for the correlation was fixed at 10 kpc/h because of the softening distance of 6.6 kpc/h. The computation needed 63 processes and a total CPU time of 88 days, with a minimum of about 10 hours and a maximum of 120 hours for a single process. In figure 7.3 is shown the results in a logarithmic scale (the last two distance bin are not visible because they have negative values.). In this case we compare our results with the results obtained by Hayashi and White in Hayashi and White 2008 in figure 7.4 on the

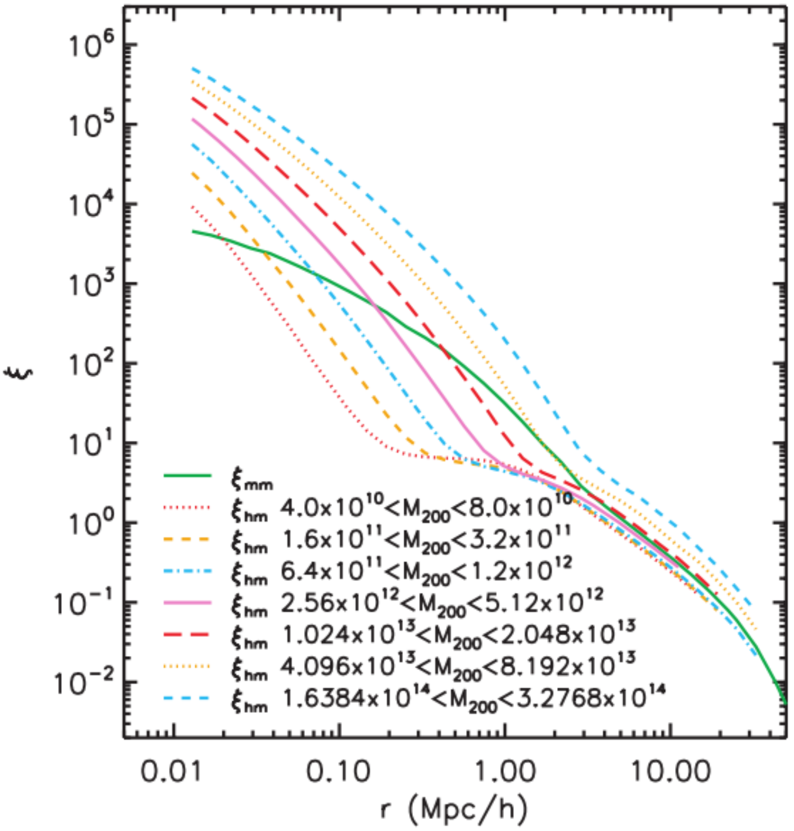

Figure 7.4: Millennium matter auto-correlation and halo-matter cross-correlation from Hayashi and White 2008.

The computation for the small scales was performed on 20 CPUs on Pico and required about 26 hours and it was splitted on 1091 processes with a minimum of 48 seconds and a maximum of 234 seconds. For the larger scales the random was as many as the data. This takes 46 hours with a minimum of 56 seconds and a maximum of 466 seconds. In this case the cross-correlation profile is obtained by virtually sitting on each halo center and counting how many matter particles there are at each radius.

The results, in figure 7.5, can be compared with the results by Hayashi and White 2008 in figure 7.4 and with the theoretical predictions by Giocoli et al. 2010 in figure 7.6.

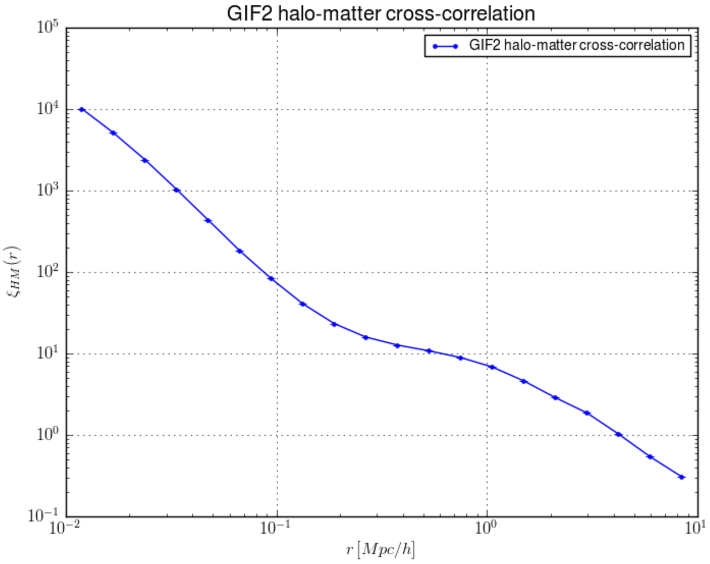

Figure 7.5: GIF2 halo-matter correlation.

7.4 GIF2 halo-matter cross-correlation in mass bins

Once checked that the code was working well calculating the halo-matter cross-correlation giving a result in good agreement with the results in literature and with the theoretical previsions the next step is to investigate the contribution of the signal from haloes of different masses. To do this we first extracted the haloes belonging to the mass bins from the dataset. The bins were centered on $\frac{M^*}{64}$, $\frac{M^*}{16}$, $\frac{M^*}{4}$, 1, $4M^*$, $16M^*$ where M^* is the mass where $M^* = 8.9 \times 10^{12} M_\odot$ and correspond to 5171 simulation particles. The bins were from $\frac{1}{\sqrt{2}}$ to $\sqrt{2}$ respect to the center. In table 7.1 it is possible to have an idea of the selection in mass.

In this case we performed the analysis with 1099 processes for each mass bin, 6548 in total, for a total time of 30 hours, a minimum of 0.5 seconds and a maximum of 104 seconds. What we expected was to see the virial radius (that is where the transition between the 1H and the 2H terms happens) to move at bigger scales increasing the mass considered. This can be seen in figure 7.7 and

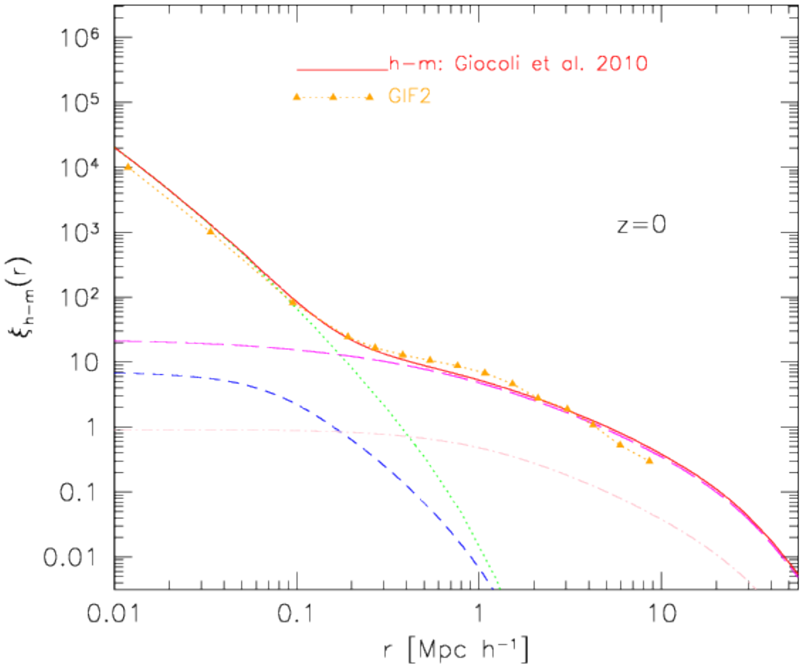

Figure 7.6: Giocoli et al. 2010 halo-matter correlation predictions. In this figure are clearly visible the contributions by the 1H and 2H terms and the sum of these two. The green line is the contribution of the correlation between the halo center with the smooth component of the halo, that is the particles belonging to the halo but not to the substructures, the blue is the correlation between the halo center and the clump component, that is the particles in substructures. The 2H term is the composition of the halo center with the smooth and clump components of other haloes. In addiction the profile calculated by our code is superposed to be easily compared.

the results can be compared with those by Hayashi and White 2008 obtained from the Millennium simulation. Another interesting analysis would be to fit each obtained profile with the NFW profile, as in Hayashi and White 2008 but due to the low resolution of the simulation and of the poor sampling of the data pairs with random pairs this test was not so significant. For example the two profiles corresponding to the higher masses do not cover the smaller scales. It

Bin center M^*	**bin center** $(M_\odot)$	**min** $(M_\odot)$	
1/64	1.4×10^{11}	1.0×10^{11}	2.0×10^{11}
1/16	5.6×10^{11}	3.0×10^{11}	7.9×10^{11}
1/4	2.2×10^{12}	1.6×10^{12}	3.1×10^{12}
1	8.9×10^{12}	6.3×10^{12}	12.6×10^{12}
4	3.6×10^{13}	2.5×10^{13}	5.1×10^{13}
16	1.6×10^{14}	1.1×10^{14}	2.3×10^{14}

Table 7.1: Mass bins for the halo-matter cross-correlation in units of M^* and $M_\odot$.

can be also noticed that lower masses haloes have steeper slope on small scale than the more massive haloes. This is because on average they are more centrally concentrated.

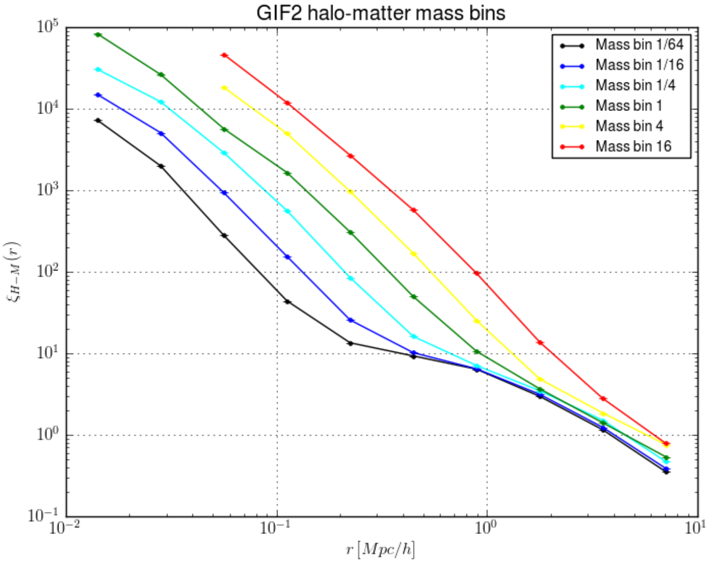

Figure 7.7: GIF2 halo-matter in mass bin correlation. in this figure can be seen how the transition between the 1H and 2H terms move to bigger scale increasing the mass of the haloes.

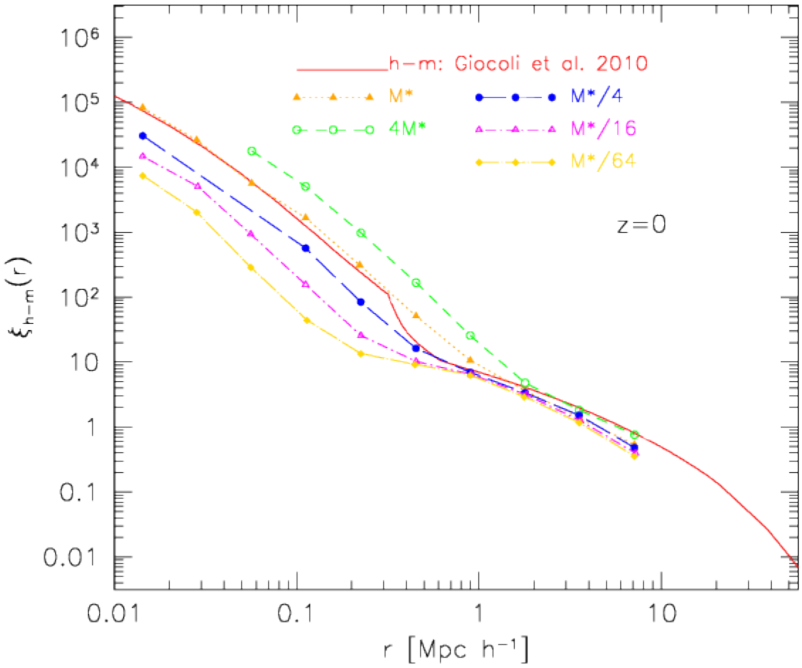

Figure 7.8: In figure you can see the GIF2 halo-matter cross-correlation per different mass bin computed by our code and the theoretical prediction by Giocoli *et al.* for a mass bin centered on M^*. The drop of the theoretical prediction at 2×10^{-1} is due to the mass cutoff in the Fourier space. A part from this the profiles are in good agreement.

7.5 GIF2 subhalo-matter cross-correlation

As in the halo-matter cross correlation the subhalo-matter cross-correlation is the composition of more than one contribution. In this case, as presented in 3.1, the 1H term is the sum of the correlation with the smooth and clump component of the same sub-halo and with the matter in other substructures. The 2H term contain the correlation with the smooth component and clump component of other haloes. To select the sub-haloes we extracted the haloes with more than 200 particles, than for each halo we retrieve the list of the contained sub-haloes.

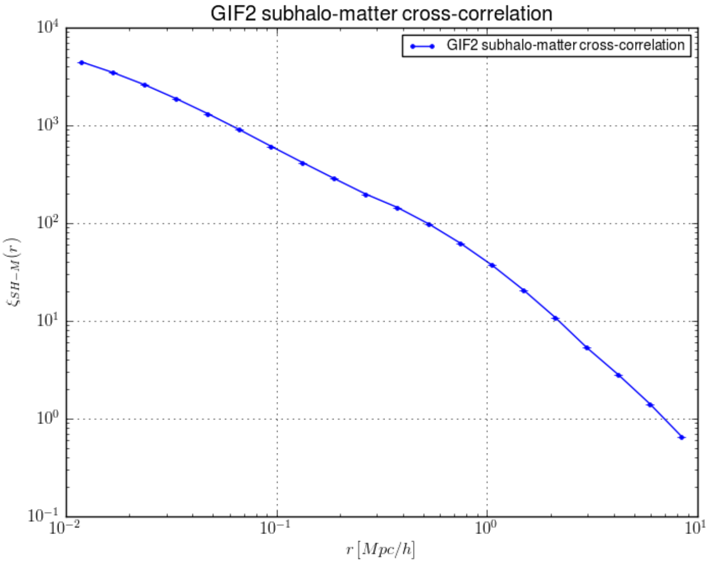

Figure 7.9: GIF2 subhalo-matter correlation.

For each sub-halo the coordinates were converted from those of the host halo center to the global ones. Some check were done to avoid problems related to the presence of void files.

It took a total CPU time of 35 hours and the computation was performed splitting it in 2184 different processes, with a minumim and maximum time needed of 23 and 272 seconds.

Subhalo-matter cross-correlation in host halo mass bins

In the same way we would computed the cross-correlation for the haloes belonging to different mass bins we were interested in investigate the signal due to the substructures hosted in haloes of different masses. On the base of the host haloes mass selection we extracted the corresponding substructures and computed the cross-correlation with all the particles.

As it was done for the halo-matter correlation we splitted the computation in many different processes, 1100, feeding many CPUs at the same time. The total CPU time is 43 hours and the minimum and the maximum are 5 and 91 seconds respectively.

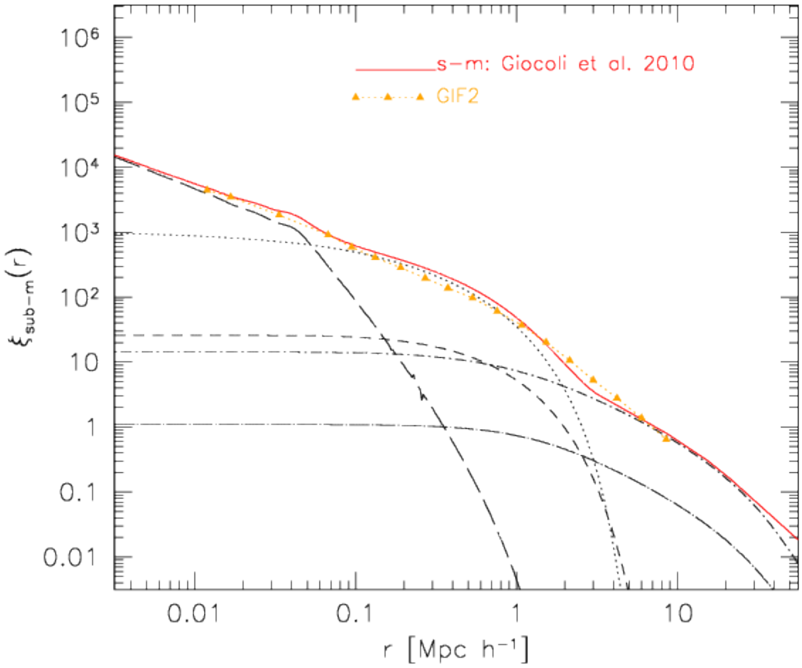

Figure 7.10: Subhalo-matter cross-correlation. The figure compares our results from the GIF2 analysis with the theoretical predictions from the model of Giocoli et al. 2010. It can be seen the good agreement between the two profiles. For the theoretical predictions are also shown the different contributions: the subhalo correlation with the subhalo matter, the smooth matter in the host halo, the matter in other subhaloes in the same host halo, the smooth and clump components of other haloes.

The result now depends on new quantities: at the smallest scales the density profile of the substructures itself; at intermediate scales the distributions of smooth matter around a sub-halo; at the largest scales the biased distribution of sub-haloes compared to matter. What we obtained shows some interesting features but it is limited by the low resolution at small scales but it is difficult to interpret. At small scales the sub-haloes hosted in smaller haloes present a steeper profile, probably due to their dynamical history inside the halo. At intermediate scales we can see that sub-haloes hosted in larger haloes live in denser regions, because they are located, on average, at smaller fraction of the host halo virial radius. At the largest scales we observe the subhalos-matter biased distribution.

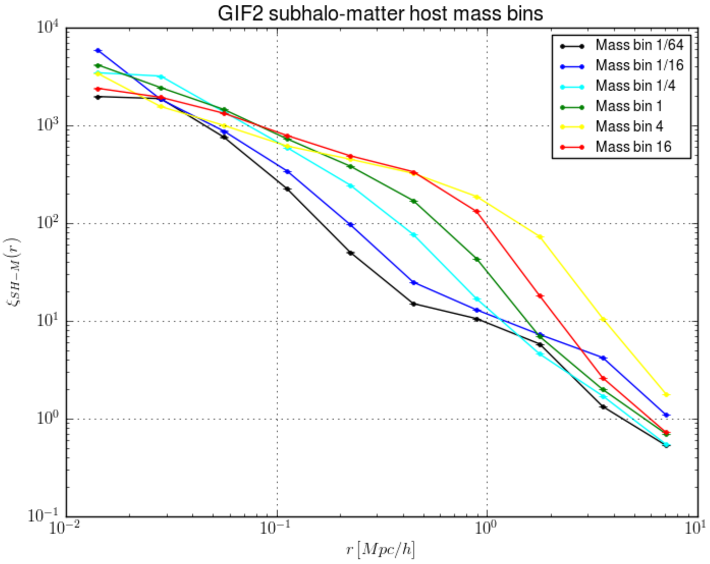

Figure 7.11: GIF2 subhalo-matter in mass bin correlation.

7.6 Bias

We can consider the *bias* as an indicator of how two different distribution we are considering differs one from the other. It also tells us how one distribution can well represents the other. We have, in the previous chapter, shortly mention that the large scale halo-matter correlation follows, biased, the matter-matter correlation profile. If we divide the halo-matter profile by the matter-matter one on large scale we have a an idea of how well the halo-matter profile represents the matter-matter one. Our results were compared with those in Sheth and Tormen 1999 and there is no a perfect agreement as you can see in table 7.2. Because our profiles are in good agreement with those in literature we think that a possible explanation can be the scale we sampled, 3.5 Mpc/h because of the data available. It is possible that the same operation on bigger scales would yield a better

CHAPTER 8

CONCLUSIONS

The goal of this work was to investigate the cross-correlation between sub-haloes and dark matter using the Millennium II simulation data. This goal was not achieved because of some technical problems. First we had some problems to obtain the data: they arrived on an hard disk from MPA in Garching but due to a failure in the first hard disk we had to wait for a second shipping. Moreover the Millennium II data was in a "private" Fortran binary format and we can only access the particles positions from the hard disk and the FOF centers from the online database, but we were unable to reconstruct the information about the sub-haloes centers. We also had some problems with our servers and to gain access to some CPU time on other machines. Some of the machines furthermore had some compatibility problems with the code so we had to spend a lot of time on this.

Despite this problems we successfully tested the code on the GIF and GIF2 code and obtained the results presented in the previous chapters. Some test slices of the Millennium II are now under analysis and we will start with the subhalo-matter cross-correlation as soon as possible.

8.1 Further works

The code is now fully working and its results tested. There are however some possible improvements. First we have to do is to clean the code and to let it be more general. The second step is about optimization. Now the code has good times but it is possible that with some changes the times will be smaller. Some possible improvements are to rewrite some parts in Chyton or C/C++/Fortran, but we thinks that this way is not so promising. In the future we will try to convert some part of the calculation to be run on GPUs and may be some further modifications to the tree code such as more statistics on the nodes and less on the fly work will give us better execution times.

Appendix A

Better bugs

A.1 array2int

With the introduction of the Fortran version of the distance calculation the output of the distance between two nodes became a one-element array (instead of the float obtained from the previous, Python, version). Thinking about the previous conversion from the configuration file parsing I converted the output of the Fortran module to `int` instead of converting it to `float`. Converting the data coordinates in kpc/h give the correct results.The result was that for a defined `leafsize`, depending on the total number of particles, the differential data counts dropped at little radii.What actually happened was that the distances between nodes were truncated to `int` and so they were underestimated. In this way little radii that should be excluded or at least should open leaves included all the leaves, leading to higher cumulative counts. The subtraction of the pair of adjacent counts eventually gave lower differential counts. Random counts, data in kiloparsecs or bigger leaf give a smaller effect that was invisible, the same happened for bigger radii, with higher counts.

A.2 `.sort` instead of `.sort()`

Two trivial parenthesis forgotten in calling the `.sort()` method of the array containing the distances return the memory address of the method instead of sorting the array, according to the Python syntax. In this way no error was raised but the counting for opened leaves was wrong.

A.3 Random population

In case of self-correlation we thought that use two set of random with only one set of data will improve the error by overestimate the random. Bad idea!! This choice let the code counts $\frac{2n}{(n-1)} \sim 2$ times the random couples. To fix this one can simply use the same number of sets both for the data and for the random or divide the final counts. The first solution save a lot of computation.

Bibliography

[1] Sunil Arya et al. “An optimal algorithm for approximate nearest neighbor searching fixed dimensions”. In: *Journal of the ACM* 45.6 (1998), pp. 891–923. ISSN: 00045411. DOI: `10.1145/293347.293348`. URL: `http://portal.acm.org/citation.cfm?doid=293347.293348`.

[2] JR Bond et al. “Excursion set mass functions for hierarchical Gaussian fluctuations”. In: *The Astrophysical Journal* 379 (1991), pp. 440–460. URL: `http://adsabs.harvard.edu/full/1991ApJ...379..440B`.

[3] M. Boylan-Kolchin et al. “Resolving cosmic structure formation with the Millennium-II Simulation”. In: *Monthly Notices of the Royal Astronomical Society* 398.3 (2009), pp. 1150–1164. ISSN: 1365-2966. eprint: `arXiv:0903.3041v2`. URL: `http://onlinelibrary.wiley.com/doi/10.1111/j.1365-2966.2009.15191.x/full`.

[4] a Cooray and R Sheth. “Halo models of large scale structure”. In: *Physics Reports* 372.1 (Dec. 2002), pp. 1–129. ISSN: 03701573. DOI: `10.1016/S0370-1573(02)00276-4`. URL: `http://linkinghub.elsevier.com/retrieve/pii/S0370157302002764`.

[5] Antonaldo Diaferio, Simon D M White, and Guinevere Kauffmann. “Clustering of galaxies in a hierarchical universe I. Methods and results at z 0 ”. In: *Monthly Notices of the Royal Astronomical Society* 206 (1999), pp. 188–206.

[6] R A Finkel and J L Bentley. “Quad trees a data structure for retrieval on composite keys”. In: *Acta Informatica* 4.1 (1974), pp. 1–9. ISSN: 00015903. DOI: `10.1007/BF00288933`. URL: `http://www.springerlink.com/index/10.1007/BF00288933`.

[7] Jerome H. Freidman, Jon Louis Bentley, and Raphael Ari Finkel. "An Algorithm for Finding Best Matches in Logarithmic Expected Time". In: *ACM Transactions on Mathematical Software* 3.3 (Sept. 1977), pp. 209–226. ISSN: 00983500. DOI: 10.1145/355744.355745. URL: http://portal.acm.org/citation.cfm?doid=355744.355745.

[8] L Gao, SDM White, and A Jenkins. "The subhalo populations of ΛCDM dark haloes". In: *Monthly Notices of the* 000.February (2004). eprint: 0404589v3. URL: http://onlinelibrary.wiley.com/doi/10.1111/j.1365-2966.2004.08360.x/full.

[9] Carlo Giocoli et al. "Halo model description of the non-linear dark matter power spectrum at k 1 Mpc^-1". In: *Monthly Notices of the Royal Astronomical Society* 15.March (2010), pp. 1–15. URL: http://arxiv.org/abs/1003.4740.

[10] A.G. Gray and A.W. Moore. "N-Body'problems in statistical learning". In: *Advances in neural information processing systems* (2001), pp. 521–527. ISSN: 1049-5258. URL: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.28.7138\&amp;rep=rep1\&amp;type=pdf.

[11] Eric Hayashi and Simon D. M. White. "Understanding the halo-mass and galaxy-mass cross-correlation functions". In: *Monthly Notices of the Royal Astronomical Society* 388.1 (July 2008), pp. 2–14. ISSN: 00358711. DOI: 10.1111/j.1365-2966.2008.13371.x. URL: http://blackwell-synergy.com/doi/abs/10.1111/j.1365-2966.2008.13371.x.

[12] G. Karniadakis and R.M. Kirby. *Parallel scientific computing in C++ and MPI.* Cambridge University Press Cambridge, UK, 2003. URL: http://embedded.ufcg.edu.br/~ivocalado/ebooks/mpi/CambridgeUniversityPress-ParallelScientificComputinginC++andMpi.pdf.

[13] M. Kerscher, I. Szapudi, and A.S. Szalay. "A comparison of estimators for the two-point correlation function". In: *The Astrophysical Journal Letters* 535.1994 (2000), p. L13. URL: http://iopscience.iop.org/1538-4357/535/1/L13.