Galaxy Evolution, Cosmology and HPC

Clustering Studies applied to Astronomy

Presented by:
Israel R Tshililo

Prepared for:
Prof Catherine Cress
Centre for High Perfomance Computing

and

Dr Simon Winberg
Dept. of Electrical and Electronics Engineering
University of Cape Town

A dissertation submitted to the Department of Electrical Engineering at the University of Cape Town in fulfilment of the academic requirements for a Master of Science in Engineering

May 20, 2016

Declaration

1. I know that plagiarism is wrong. Plagiarism is to use another's work and pretend that it is one's own.
2. I have used the IEEE convention for citation and referencing. Each contribution to, and quotation in, this report from the work(s) of other people has been attributed, and has been cited and referenced.
3. This report is my own work.
4. I have not allowed, and will not allow, anyone to copy my work with the intention of passing it off as their own work or part thereof.

Signature:.. Signed by candidate

Israel R Tshililo

Date:..... May 20, 2016

Acknowledgments

I would like to express my gratitude to the following people and institutions who helped me to in conducting this research dissertation.

Prof Catherine Cress for providing me the opportunity to study through the CHPC studentship program, for exposing me to the field of astronomy and the interesting scientific questions. I am grateful for the supervision and assistance in producing this dissertation

Dr Simon Winberg for his supervision, availability and constructive feedback during the research of this thesis.

My family for all their support and prayers. Especially my mom Mrs R Tshililo for her consistent support, encouragement and motivation.

Dr Sean February for his valuable assistance and encouragement in the field of astronomy

My colleagues in the Acelab for their inspiration and encouragement to complete this dissertation.

This dissertation was funded through a studentship granted by the Human Capital Development Programme of the CHPC, an initiative of the Department of Science and Technology (DST) and managed by the Meraka Institute of the Council for Scientific and Industrial Research (CSIR).

Abstract

Tools to measure clustering are essential for analysis of Astronomical datasets and can potentially be used in other fields for data mining. The Two-point Correlation Function (TPCF), in particular, is used to characterize the distribution of matter and objects such as galaxies in the Universe. However, it's computational time will be restrictively slow given the significant increase in the size of datasets expected from surveys in the future. Thus, new computational techniques are necessary in order to measure clustering efficiently.

The objective of this research was to investigate methods to accelerate the computation of the TPCF and to use the TPCF to probe an interesting scientific question dealing with the masses of galaxy clusters measured using data from the Planck satellite.

An investigation was conducted to explore different techniques and architectures that can be used to accelerate the computation of the TPCF. The code CUTE, was selected in particular to test shared-memory systems using OpenMP and GPU acceleration using CUDA. Modification were then made to the code, to improve the nearest neighbour boxing technique. The results show that the modified code offers a significant improved performance.

Additionally, a particularly effective implementation was used to measure the clustering of galaxy clusters detected by the Planck satellite: our results indicated that the clusters were more massive than had been inferred in previous work, providing an explanation for apparent inconsistencies in the Planck data.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background to the study

In cosmology today, the Universe is understood to be comprised of only 4.9% observable matter (galaxies, stars, and planets, etc), with the remaining 26.8% attributed to dark matter(DM) and 68.3% to dark energy(DE)[3] (*see section 2.1.2 for details*). These estimates are based on observations of the universe and the well accepted ΛCDM cosmological model, also known as the standard model for cosmology. Dark Matter is a hypothetical form of matter required to explain observations on scales of galaxies and larger. Dark energy is a hypothetical form of energy responsible for the acceleration in the rate of expansion of the Universe and the work that led to this discovery was awarded the Noble Prize in Physics in 2011[37]. However, the physical properties of DE & DM remain largely unknown. This has led to increased efforts worldwide to conduct very large wide and deep surveys (including large areas of the sky and very faint objects).

Ongoing surveys, such as DES[41], BigBOSS[66] or Euclid[27], are producing catalogs with hundreds of millions of objects. Now, with the development of the Square Kilometer Array (SKA) radio telescope project, the number of objects is expected to exceed tens of billions of objects in the future. It is understood that the spatial distribution of these objects contains invaluable information that can be vital in answering some of the open problems in cosmology, such as the nature of dark matter and dark energy.

One of the simplest observables that has been use to quantify the clustering of matter on different scales in the universe is the Two Point Correlation Function(TPCF). It's estimation is based on counting pairs of objects separated by a given distance. It's computational time grows with the square of the number of objects in the catalogs.

It has been used in astronomy since the 1980s[50], when catalogs were relatively small containing only hundreds to few a thousand objects and simple serial codes were sufficient for computation. However, given the increase in the size of catalogs, there is a growing need to investigate efficient computational techniques or platforms for performing the TPCF.

The Centre for High Performance Computing (CHPC) is the largest HPC facility in Africa and provides a great opportunity to test new engineering solutions within it's Advanced Engineering (ACE) Lab. This provides the necessary environment and infrastructure for conducting the research on this topic. Also, the CHPC will host data from the MeerKAT telescope which will make up the first 64 of the 200 dishes planned for phase 1 of the SKA in South Africa. Providing tools to analyse this data will ensure rapid delivery of scientific results: a tool to measure clustering is one of many we could provide.

1.2 Objectives of this study

This dissertation presents the research conducted to investigate and develop efficient computation techniques required to compute the Two-point correlation function (TPCF) in astronomy. The research involves looking at the algorithms, computational platforms and accelerators that can be applied in measuring the TPCF. In addition, it is important to demonstrate the usefulness of the TPCF tools by using them to answer an interesting scientific question in astronomy.

1.2.1 Problems to be investigated

The main problem is that the size of the current cosmological datasets/catalogs has increased and continues to increase, such that we need to investigate new computational techniques in order to compute the TPCF within a reasonable time frame. Also, there are apparent inconsistencies in astronomical observations made by the Planck satellite [45] which I use the TPCF to explore. A number of questions have been developed to help facilitate the investigation of these problems.

Therefore, on completion of this dissertation, the following questions will need to be addressed:

1. **What is the two-point correlation function (TPCF) and what are the**

current computational techniques? *This question concerns algorithms used to compute the two-point correlation statistic and how it is applied in the field of astronomy. It is important to ensure that the algorithm adopted is appropriate, as the TPCF is applicable to other fields outside that of astronomy.*

2. **What methods/techniques are available to accelerate the computation of the TPFC and how well do these techniques scale with different sizes of datasets/catalogs?** *This question also seeks to identify ways in which the TPCF is currently being implemented. It helps with evaluating the scalability of the proposed solution.*

3. **How can we use the TPCF tools to answer some of the problems/questions in cosmology/astronomy.?** *We need to demonstrate that the TPCF is usable in addressing practical problems, such as probing the masses of galaxy clusters using their clustering signature.*

1.3 Scope and Limitations

The scope of this study is to investigate and develop an efficient computational tool for implementing the TPCF on large datasets. In addition, use the TPCF in probing the relationship of the mass of the galaxy clusters and their clustering signature.

The scope of the study is limited by the following factors:

- **Available Catalogs:** In testing the code, we limited the scope to three datasets presented in section 3.2.4.
- **CUTE:** Short for "Correlation Utilities and Two-Point Estimatio" (*see section 2.6.1*). This will be the only parallel code to be explored in this investigation, since it offers the advantage of testing both shared-memory platforms as well as GPU acceleration and it's publicly available on a free license.
- **Hardware availability:** The testing of the code will be limited to the platforms available within the ACELab at the CHPC. The details of the hardware is provided in section 3.2.1.
- **3D correlation analyses:** The solution was developed primarily to solve the spatial 3D correlation function. The angular correlation function was only tested as a validation tool for the initial prototype solution.

1.4 Plan of development

This section describes the organisation of the dissertation.

Chapter 2 presents a brief overview on the subject of physical cosmology, current research on cosmology from the Planck satellite observations and some theories of correlation analysis used to address questions in the field. The chapter also explores parallel computing paradigms and provides a detailed review of relevant parallel codes to be explored in this dissertation.

Chapter 3 follows with a concise description of the research methodology followed throughout this investigation. It includes a presentation of the research environment, along with a selection of the catalogs to be used in this study. Then, the chapter ends with a definition of experiments, specifically designed to address each research question in this dissertation.

Chapter 4 provides a detailed description of prototype scripts and code algorithms developed in this dissertation. This involves the data preprocessing method developed for extracting information required from the data catalog and the python prototype scripts developed to test the TCPF concepts from literature. A detailed analysis of CUTE's 3D boxing technique is also provided.

Chapter 5, then presents the results from our investigation aimed at answering the first question in sesction 1.2.1. This involves comparisons of clustering measurements from previous work with those generated with our own code.

In Chapter 6 we investigate acceleration of the TPCF codes. We then use the code in Chapter 7 to study the clustering of galaxy clusters, thus providing a probe of the mass of the clusters detected by the Planck satellite. In chapter 8, we conclude and provide some recommendations.

Chapter 2

Literature Review

This chapter presents a background and review of reference materials that were consulted to formulate the theoretical basis of this research. The chapter is structured such that it covers the general-to-specific aspects of the theory, which can be illustrated by the diagram in figure 2.1:

Firstly section 2.1 provides brief background on the theory of Physical cosmology. This includes the standard model of cosmology, also commonly referred to as the ΛCDM model. The section also looks at historical observations of "large scale structures", highlighting the improvements made in conducting cosmological surveys and the increasing number of objects being observed over the past years.

Subsequently, section 2.2 provides the scientific context for this research. This section describes the tension from recent results reported by Planck Collaboration (XX & $XXIV$) regarding observations of galaxy clusters and the base ΛCDM model. Results from other studies (gravitational lensing) are also presented, in order to expose the possible causes for this tension.

In section 2.3 a brief overview of correlation analysis in cosmology, specifically the Two-point correlation function (TPCF) is presented. This involves the definition of the TPCF and methods for measurements. In addition, the computational complexity for calculating the TPCF is also explained. Then, section 2.4 describes distance measures in astronomy, including the relevant coordinates systems and distance conversions employed.

Section 2.5 provides the engineering context for this research. A broad overview on parallel computing and history of high performance computing is presented. Then the hardware classification of parallel computers is provided, followed by a descriptions on

some of the parallel programming models used. A description of the Centre for High Performance Computing (CHPC) is also presented.

A review of relevant parallel codes for computing the TPCF is presented in section 2.6. This includes a more detailed description of the CUTE (Correlation Utilities and Two-point Estimation) code, and a brief presentation of results from the GP2PCF study. Finally, section 2.6.3 explains how this chapter is applied to the rest of the dissertation.

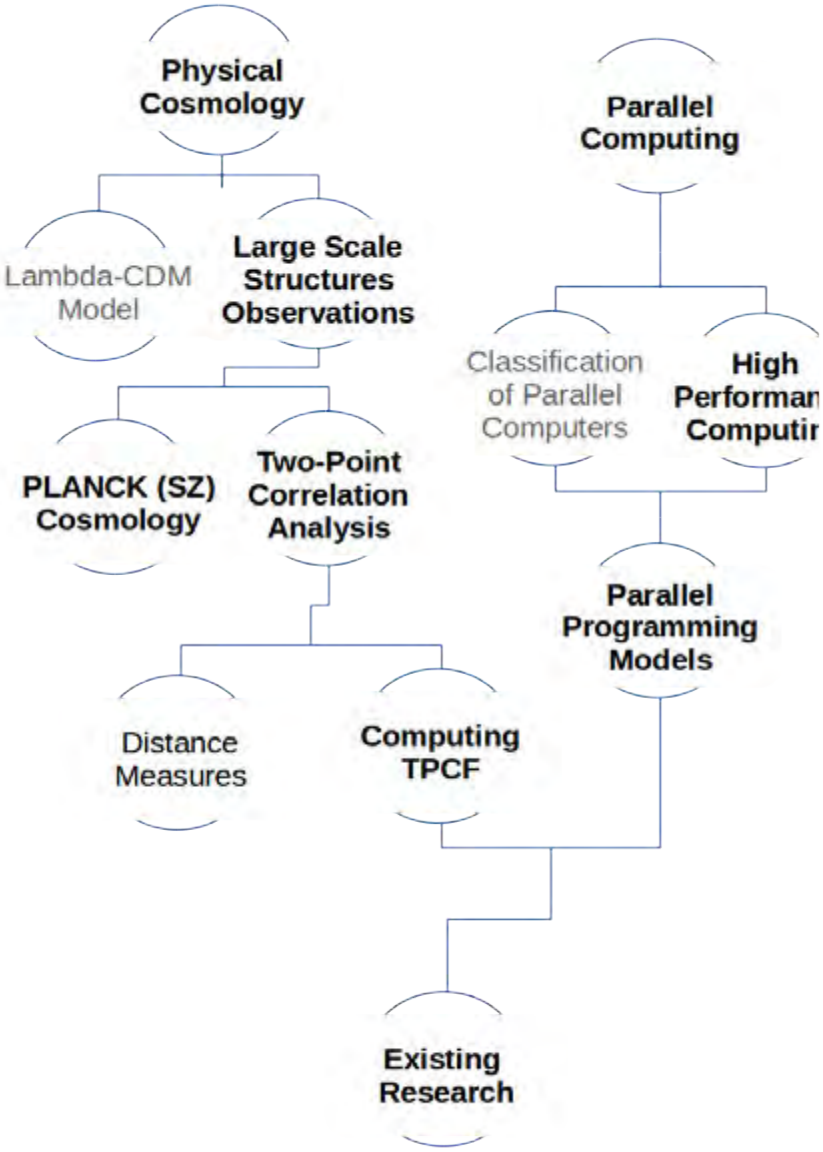


Figure 2.1: Structure of the Literature Review.

2.1 Background to Physical Cosmology

Physical cosmology refers to the study of the origins, structure and evolution of our Universe, governed by both theoretical principles and observational evidence. The key concepts and observations are outlined below.

2.1.1 The Big Bang Theory

The Big Bang Theory is the prevailing cosmological model that describes the earliest known periods of our universe and it's evolution over time. It states that the universe started in a singularity- (very hot and dense state) and then expanded after the Big Bang event illustrated in Fig.2.2. The Big Bang is believed to be a single event that occurred 13.7 $Gyr(1Gyr = 10^9)$ years ago. The universe then grew exponentially in the phase commonly referred to as the cosmic inflation[68]. After approximately one second, the universe went through a phase referred to as Big Bang nucleosynthesis. In this phase the universe cooled sufficiently to allow the formation of primordial elements such as Helium and Lithium. After about 350000 years, the universe had expanded and cooled down so much that photons began to free-stream, creating the Cosmic Microwave Background (CMB). Electrons combined with nuclei to form atoms. This gas coalesced through gravity forming structures such stars, galaxies and other astronomical structures that we are able to observe today.

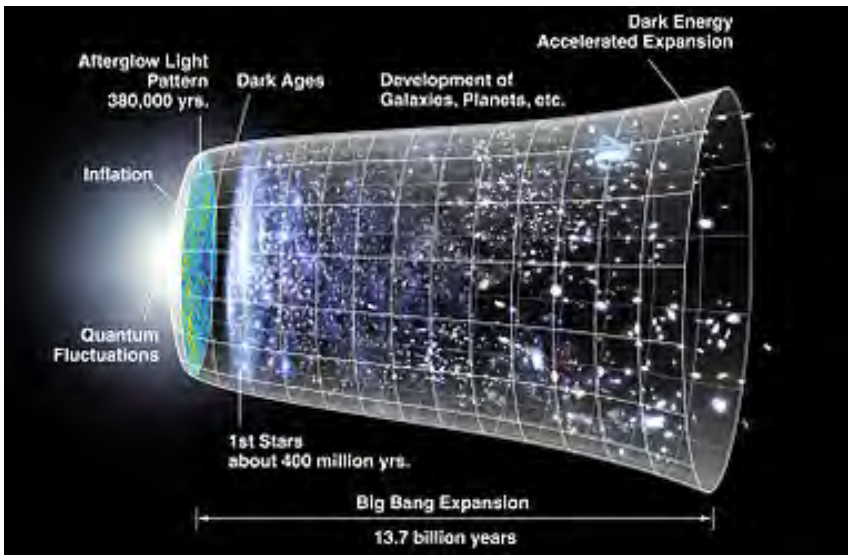


Figure 2.2: A visualization illustrating the expanding universe through different epochs, with the initial inflation epoch starting on the left towards the current accelerated expansion on the right[1]

The theory was first proposed in 1927 by Georges Lamaitre. Later in 1929, Edwin Hubble

concluded that galaxies are moving away from each other using his analysis of galactic observations [19], which further strengthened the hypothesis of an expanding universe. In 1964, the cosmic microwave background (CMB) radiation was discovered, providing strong evidence for an expanding universe and resolving some divided opinions within the scientific community regarding the Big Bang model. This discovery was awarded a Nobel prize in physics in 1978[77, 53]. The most recent discovery observations of supernovae shows that the expansion of the universe is accelerating.

2.1.2 The Standard Model

The Big Bang Theory has been parameterized into a mathematical model known as ΛCDM and it serves as the framework for current investigations of theoretical cosmology. The letter Λ stands for the cosmological constant associated with dark energy and the CDM term is an abbreviation for cold dark matter. The model is based on the cosmological principle, which states that our location in the Universe is not significant, because when viewed from a sufficiently large scale the Universe looks the same in all directions (isotropy) and from every location (homogeneity) [53]. This model is also commonly referred to as the standard model of the Big Bang Cosmology, because it offers a reasonably good account of various observations, such as[67]:

- the existence of structures in the cosmic microwave background (CMB)
- the large scale structures in the distribution of galaxies
- the accelerating expansion of the universe observed in the light from distant galaxies and supernovae

The CMB is a thermal radiation left over after the Big Bang. It is considered the oldest light in the Universe, dating back to the epoch of recombination [3]. It is observable through radio telescopes, appearing as a faint background glow strongest in the microwave region of the radio spectrum. Precise measurements of the CMB are very critical for cosmology and observations show that the CMB has a thermal black body spectrum at a temperature of $2.72548 \pm 0.00057K$[89]. A map representing the CMB is presented in Fig2.3. The glow of CMB is measured to be nearly uniform in all directions, with only a slight residual variation observed consisting of a very specific pattern. The initial fluctuations are thought to be generated by quantum fluctuations, which occurred in the very early universe. As the universe expanded over the cosmic time scale, the matter coalesced through gravity forming large scale structures of matter we can observe currently.

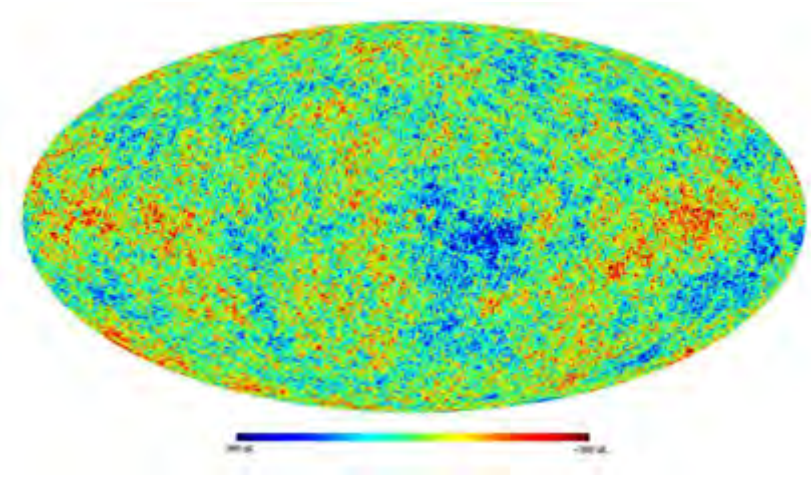


Figure 2.3: The "all-sky map" from the 9year WMAP Cosmic Microwave Background survey. The positive and negative variations in measured temperature are displayed as red and blue respectively while the mean is represented as cyan[2] .

Observations of galaxies, reveal strong clustering of galaxies, gravitational lensing of light by galaxy clusters and flat rotational curves of galaxies[31][32]. These observations cannot be explained if all the mass is attributed to luminous matter. This led to the proposal of the hypothetical form of matter referred to as dark matter. Dark matter is describes as being cold, hence abbreviated CMD because it's velocity is much slower than the speed of light [57]. The cold dark matter theory predicts that structure grows hierarchically, with small objects collapsing under their self-gravity first and merging in a continuous hierarchy to form larger and more massive objects, which is generally consistent with observations.

Observations since the 1990s have shown that the universe is expanding at an accelerated rate. This accelerated expansion as been attributed to a hypothetical form of energy that permeates all space known as dark energy. Similarly to dark matter, dark energy does not directly interact with ordinary matter. However, dark energy has a non-standard gravitational interaction. The components of the ΛCDM model are summarized in figure 2.4.

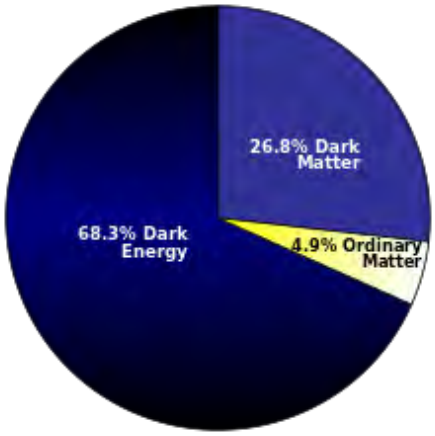


Figure 2.4: A diagram of the estimated distribution of mass-energy/matter in the Universe[3]

2.1.3 Historical Observations of large scale structures

Large-scale structure is defined as the structure or the inhomogeneity of the Universe on scales greater than that of a galaxy[9]. In 1926, Edwin Hubble used his catalog of 400 extragalactic nebulae to test the homogeneity of the Universe and found it to be generally uniform on large scales[65]. However, when the Shapley-Ames catalog of bright galaxies was published in 1932, the authors found an irregularity in the distribution of that galaxies projected onto the sky. Then, two year later (in 1934) Edwin Hubble used a larger statistical sample of the same catalog and found that on angular scales less than $\sim 10^{\circ}$ there was an excess in the number counts of galaxies above what would be expected for a random Poisson distribution[50]. Therefore, although on the largest scales the Universe appears to be homogeneous, on smaller scales it is clearly clumpy.

In 1967, measurements of large scale structures had improved significantly with the Lick galaxy catalog. The catalog contained information on approximately a million galaxies, obtained using photographic plates at the 0.5 refractor at Lick Observatory [58]. This catalog was then used to produce maps of the counts of galaxies in angular cells across the sky shown in figure 2.5[4]. The maps revealed in greater detail the non-uniformity in the distribution of galaxies projected onto the plane of the sky. The map show a foam-like pattern made up of long strands of galaxies forming filaments, clumps of galaxies and empty regions. The clustering of the galaxies in the catalog was measured using the angular two-point correlation function (TPCF) (*see section 2.3.1*). On scales of $\sim 0.1^{\circ} - 5^{\circ}$, the TPCF is well fit by a power law[46]. It was also discovered that the clustering amplitude is lower for fainter galaxies and it was attributed to projection effects along the line of sight. This led to a need for better surveys which included 3-

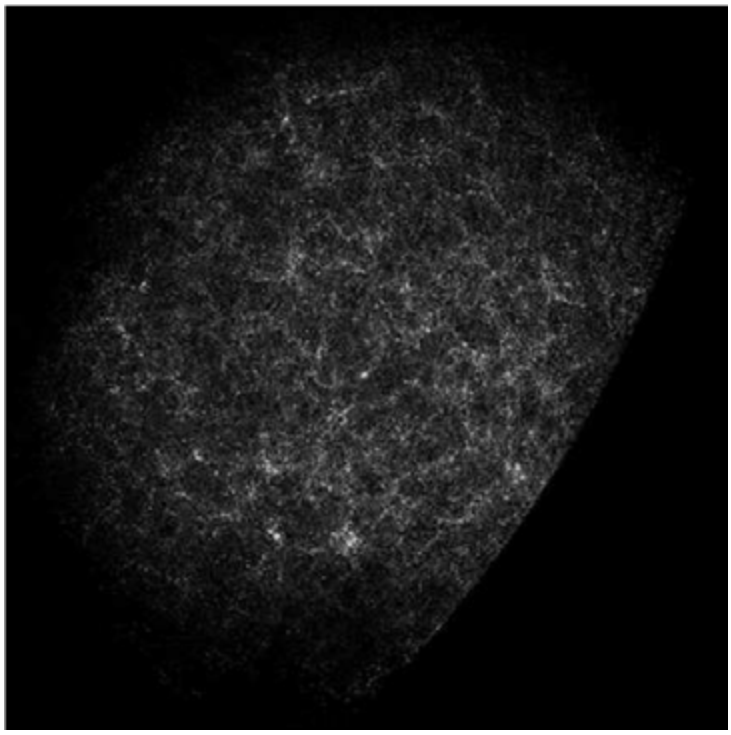


Figure 2.5: Angular distribution of counts of galaxies brighter than B $\sim$ 19 on the plane of the sky, reconstructed from the Lick galaxy catalog (from Seldner et al. 1977) This image shows the number of galaxies observed in $10' \times 10'$ cells across the northern galactic hemisphere, where brighter cells contain more galaxies. The northern galactic pole is at the center, with the galactic equator at the edge. The distribution of galaxies is clearly not uniform; clumps of galaxies are seen in white, with very few galaxies observed in the dark regions between [4].

dimensional (3D) information. In the standard model of cosmology, light from galaxies is redshifted by an amount that corresponds to their distance so optical spectra provide distances to galaxies (see appendix A.2)

In the late 1970s, the first large scale redshift surveys were carried out. These included optical spectra of individual galaxies used to measure their distance, providing the spatial distribution of large galaxy samples. The initial work mapped the three dimensional spatial distribution of 238 galaxies around and toward the Coma/Abell 1367 super-cluster. In [31], the authors noticed large regions of greater $20h^{-1}Mpc$ without galaxies refered to as 'voids'. They used the redshift information to show that galaxies are clearly clustered in three dimensions, forming chains of galaxies known as filaments[54]. Other additional redshift surveys conducted at that time were the KOS (Kirshner, Oemler, Schechter)[52] and the CfA (Centre for Astrophysics)[51] surveys.

The KOS survey was completed in 1978, containing measures of 164 galaxies brighter than magnitude 15 in eight separate fields on the sky, covering a total of 15 deg^2. The CfA survey ran in two phases, the first phase completed in 1982, containing 2400 galaxies

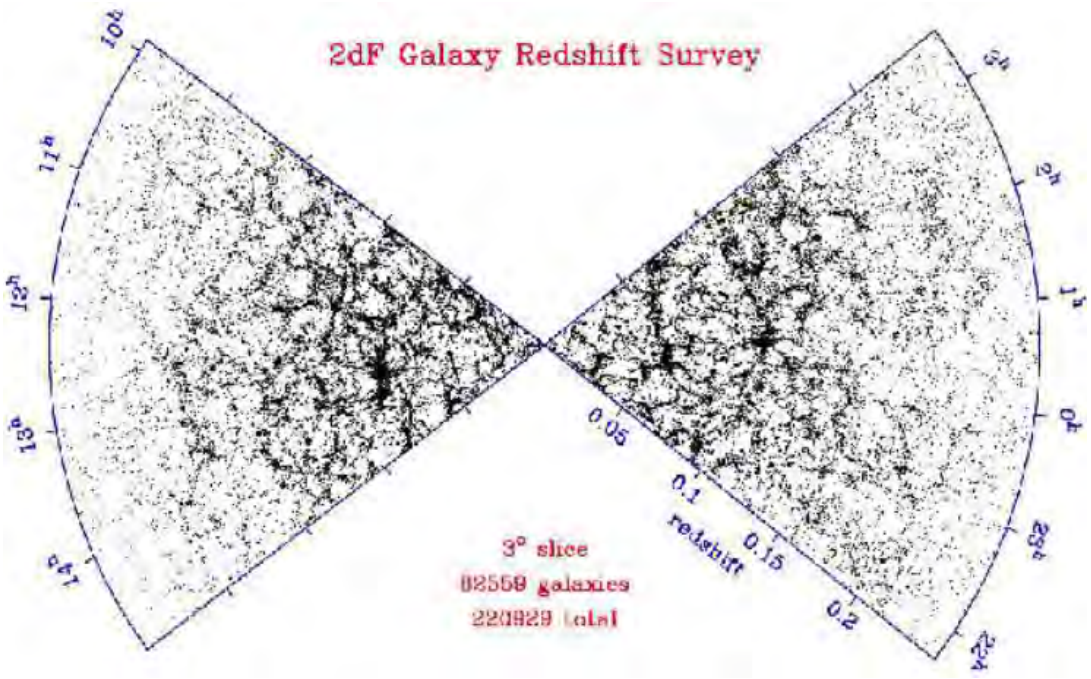


Figure 2.6: The spatial distribution of galaxies as a function of redshift and right ascension (projected through 3^o in declination) from the 2dF Galaxy Redshift Survey [5].

brighter than magnitude 14.5 across the north and south galactic poles covering a total of 2.7 steradians. The second CfA survey ran from 1985 to 1995, containing spectra of about 5400 galaxies and led to the discovery of the "Great Wall" [1] [51] (see Figure 2.6), with large under-dense voids. These discoveries further demonstrated how matter and galaxies is clustered throughout the Universe, also paving a way for studies in theoretical models of structure formation.

The development of multiple object spectrographs and larger telescopes has improved measures of redshift surveys immensely. This has enabled simultaneous observations of hundreds of galaxies and conduction of deeper surveys of both lower luminosity nearby galaxies and distant luminous galaxies. The best examples of this type of redshift surveys are the Two Degree Field (2dF) Sky Survey[5] and the Sloan Digital Sky Survey (SDSS)[63], containing spectroscopic redshifts of approximately 220000 and a million galaxies respectively. These surveys currently provide the best maps of large scale structures in the Universe today (see Figure 2.6). Measurements of the spatial distribution of these large scale structures provide great insight into theoretical models of structure formation.

[1] Is an immense galaxy filament or a supercluster of galaxies, also called the Coma Wall. It is one of the largest known superstructures in the observable universe.

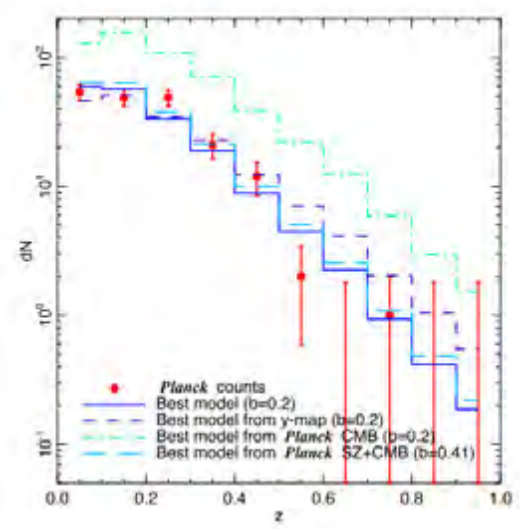
2.2 Planck Cosmology

The Planck satellite was launched in 2009 and made measurements of emission in nine frequency bands between 30 and 857 GHz, across the whole sky. The goal was to study the primary CMB and distortions to the primary CMB caused by large-scale structure viewed along the line of sight to the CMB. Clusters of galaxies are gravitationally bound groups of galaxies with about $100 - 1000$ members and are an example of large-scale structure that distorts the primary CMB. Clusters contain vast amounts of hot ($\sim 10^7 K$) gas in the intra-cluster medium. When the CMB streams through the cluster, some CMB photons are up-scattered to higher frequencies via the Inverse Compton effect. At frequencies below ~ 218 GHz, clusters appears as 'holes' in the CMB. At higher frequencies, clusters appear as bright patches. This distortion to the primary CMB is known as the thermal Sunyaev-Zeldovich effect.

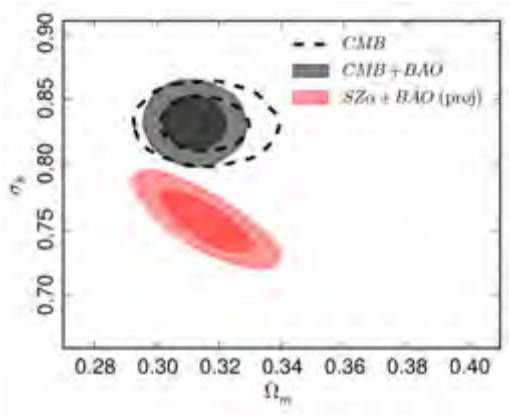
Results from the study by the Planck collaboration shown in Figure 2.7a, found fewer clusters than predicted when combining the ΛCDM model with the results from their primary CMB observations[18]. These results were also expressed as a tension between (Ω_m, σ_8) constraints from primary CMB versus those from SZ-cluster counts (where Ω_m is the matter density in units of the critical density), as shown in figure 2.7b. The σ_8 parameter represents the rms density fluctuations within a sphere of radius $r \sim 8h^{-1}Mpc$ and is used to describe the amplitude of the power spectrum on scales where it becomes non-linear (where h is a parameter in the range [0.5, 1] reflecting the uncertainty in the value of the Hubble constant H for the rate of expansion of the universe: $h = \frac{H}{100(km/s)/Mpc}$) This conflict could indicate something is wrong with the ΛCDM model or that the masses of clusters inferred from SZ signatures are not correct[45].

An alternative method that can be used to estimate the mass of clusters is to use gravitational lensing. Figure 2.8 illustrates the difference between observation made using the Sunyaev-Zeldovich effect and gravitational lensing. Lensing studies such as Weighing the Gaints (WtG)[35] show on average $\sim 30 - 40\%$ higher mass than estimated through the SZ effect[7]. This indicates that the problem with the Planck cluster count results, is in the calibration of the mass estimated through the SZ effect.

Attempting to measure masses of SZ-clusters using other methods is an active area of research, studies in [42], [70][71] and [38] compare weak lensing measurements of cluster masses with clusters detected using Planck and other telescopes such as the Atacama Cosmology Telescope and the South Pole Telescope. Estimates of the mass can also be obtained from X-ray observations (eg.[42]) and from optical spectroscopy (eg.[91]). In this thesis, we probe the masses of the Planck SZ clusters using their clustering signature.



(a) Cluster count



(b) Primordial CMB constraints vs constraints from SZ detections

Figure 2.7: Results from Planck's cluster counts compared to predictions of the ΛCDM model. [6]

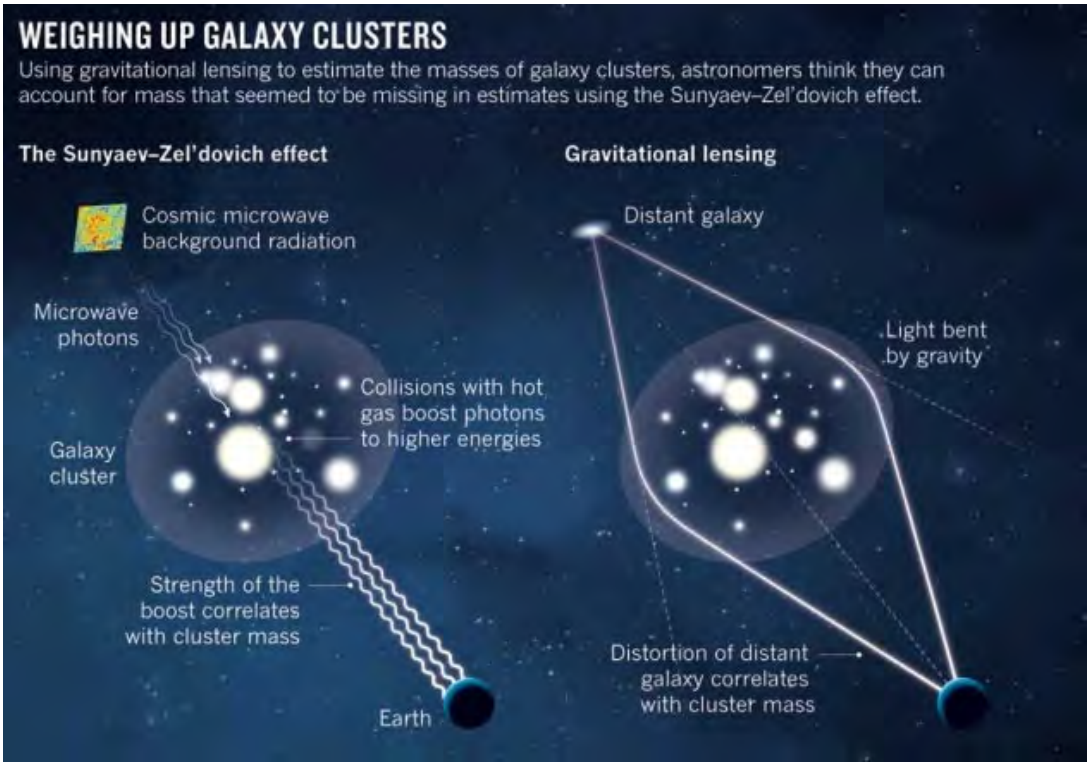


Figure 2.8: An illustration demonstrating the concepts of The Sunyaev-Zeldovich and gravitational lensing effects. [7]

2.3 Correlation Analysis in Cosmology

Correlation analysis is a widely used tool in a variety of fields such as genetics, geology, finance, etc. In the field of astronomy it used to quantify the clustering of matter on different scales in the Universe and it is commonly implemented as the Two-point correlation function (TPCF). In this section we look at the definition of the TPCF, the different estimators used in calculating the function and the relevant types of correlation analysis available in literature.

2.3.1 The Two-point Correlation Function

The Two-point Correlation Function (TPCF) is a function of one variable (distance), which describes the probability that two galaxies are separated by this particular distance. Thus, the three-dimensional TPCF $\xi(r)$ is defined as the measure of the excess probability dP of finding a galaxy in a small volume element dV_1 at a given distance separation r_{12} from another galaxy in dV_2 (see Fig:2.9), with respect to an expected unclustered random Poisson distribution

$$dP = n[1 + \xi(r)]dV_1 dV_2 \tag{2.1}$$

where n is the mean number density of the object sample in question[90]. The measurements of $\xi(r)$ are generally in comoving space(see appendix A.3), with r expressed in units of $h^{-1}Mpc$ [9].

The Fourier power spectrum $P(k)$ is also often used to describe clustering. The spatial correlation function $\xi(r)$ is related to $P(k)$, by:

$$\xi(r) = \frac{1}{2\pi^2} \int dk\, k^2 P(k)\, \frac{\sin(kr)}{kr} \tag{2.2}$$

where the scale (λ) of fluctuation is related to the wavenumber k by $k = 2\pi/\lambda$.The power spectrum is the quantity predicted directly by theories for the formation of large scale structure. In the case of a density field in which the fluctuations are drawn from a Gaussian distribution, the power spectrum gives a complete statistical description of the density fluctuations observed in the CMB[9].

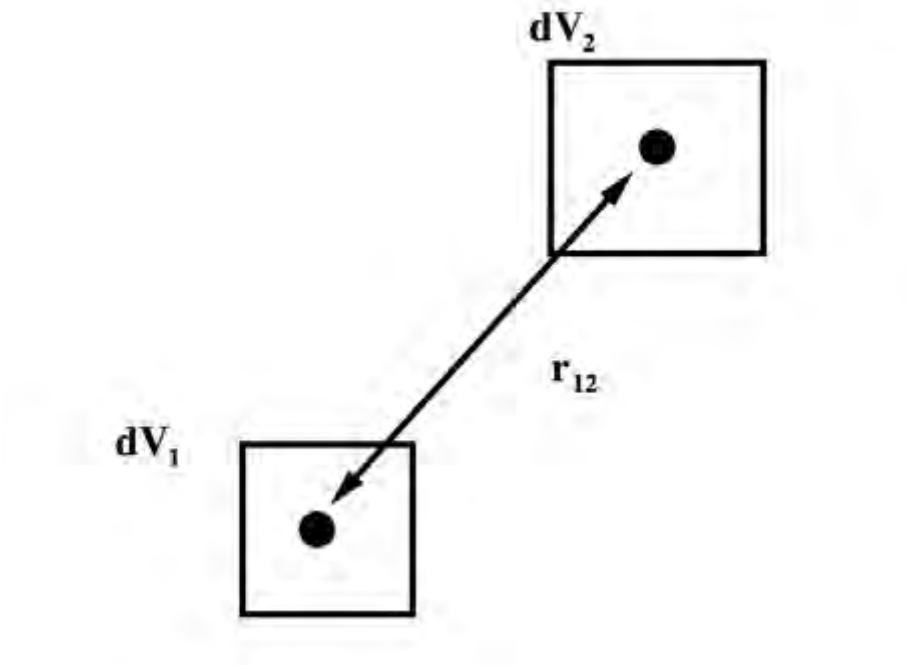


Figure 2.9: The two-point correlation function describes the excess probability, compared with a random distribution of galaxies, of finding a galaxy in an element of volume dV_2 at distance r_{12} away from a galaxy in dV_1. [8].

2.3.2 The Angular Clustering

Redshift information is not always available for a given sample of galaxies, as it is observationally expensive to obtain spectra for large samples of galaxies. Therefore, the spatial distribution of galaxies can be measured in two dimenstions as a projection onto the plane of the sky. This is known as the projected angular correlation function $\omega(\theta)$, defined as the probability above Poisson of finding two galaxies with an angular separation θ:

$$dP = n[1 + \omega(\theta)]d\Omega \tag{2.3}$$

where n is the mean number of galaxies per steradian and $d\Omega$ is the solid angle of a second galaxy at a separation θ from a randomly chosen galaxy. Generally the projected two-point angular correlation function $\omega(\theta)$ can be fitted with a power law expressed as;

$$\omega(\theta) = A_\omega \theta^\delta \tag{2.4}$$

where A is the clustering amplitude at a given scale and δ is the slope of the correlation function. If the redshift distribution of the sources is well known, one can then infer the three dimensional TPCF $\xi(r)$. The $\xi(r)$ is also usually fit as power law

$$\xi(r) = (r/r_0)^{-\gamma} \tag{2.5}$$

where r_0 is the characteristic scaling length of the clustering, defined as the scale at which $\xi = 1$. The details of how to infer $\xi(r)$ from $\omega(\theta)$ is described in [9]. The methods is mostly employed in cases where it's not feasible to obtain the redshift of individual sources. In cases where the redshift of the of the sources is known, it is preferable to measure ξ directly.

2.3.3 Estimating the TPCF

To measure $\xi(r)$, one counts pairs of galaxies as a function of separation and divides by what is expected for an unclustered distribution. To do this, one must construct a "random catalog" with randomly distributed points within the identical three dimensional coverage of the data catalog, including the same sky coverage and a smoothed redshift distribution. An early estimator that has widely used since 1974 proposed by Peebles & Hauser is expressed as:

$$\xi_{PH}(r) = \frac{n_r(n_r-1)}{n_d(n_d-1)}\frac{DD}{RR} - 1 \tag{2.6}$$

where n_d and n_r are the number of points in the data & random catalog generated respectively and DD & RR are histograms containing the count of pairs of objects found separated by a given distance in each catalog. However this estimator is very sensitive to the size of the random catalog and doesn't handle edge corrections well.

Later in 1983, Davis & Peebles proposed an improved estimator to minimize statistical errors by making use of the cross-correlation of random and data objects,

$$\xi_{DP}(r) = \frac{n_r}{n_d(n_d-1)}\frac{DD}{DR} - 1 \tag{2.7}$$

where DR is the histogram containing the count of pairs of objects found separated by a given distance between the data and random catalog. Then in 1993, Hamilton proposed a much better estimator with smaller statistical errors,

$$\xi_{H}(r) = \frac{4n_r n_d}{(n_d-1)(n_r-1)}\frac{DD\times RR}{DR^2} - 1 \tag{2.8}$$

and within the same year Landy & Szalay(ξ_{LS}) proposed another estimator for the TPCF

$$\xi_{LS} = \frac{\frac{n_r(n_r-1)}{n_d(n_d-1)}DD - \frac{n_r-1}{n_d}DR + RR}{RR} \tag{2.9}$$

This is the most widely used estimator in cosmology. Also, various studies have been

conducted in [82] and their results showed that it presents the best properties in estimating the TPCF.

As evident from the estimators presented above, it is important to generate the random catalog correctly. This includes the background spatial distribution both in redshift and angular selection. Also the random catalog needs to be large enough to minimize Poisson errors in the estimator.

2.3.4 Computational Complexity of the estimators

In order to compute the $DD, RR, \& DR$ used in the estimators as described in the previous subsection, we autocorrelate and cross correlate each pair of catalogs. A simple serial algorithm to describe this computation involves looping over each catalog and performing three operations in each iteration; firstly calculating the distance between each pair of objects, followed by determining the bin corresponding to that calculated distance and then incrementing the histogram count in that particular bin. This can be illustrated in the pseudo-code below As can be seen from the code above, the nested loops

```
1  int histogram[nbins];
2  for(i=0;i<np1;i++) {
3    for(j=0;j<np2;j++) {
4      //Calculate distance between two objects
5      double dist=get_dist(x1[i],y1[i],z1[i],
6                           x2[j],y2[j],z2[j]);
7      //Calculate bin number
8      int ibin=bin_dist(dist);
9      //Increase histogram count
10     histogram[ibin]++;
11   }
12 }
```

Figure 2.10: A snippet of a serial code for computing the histogram counts for the catalog pair separation [9].

make this computation into an N^2 problem, for which the computational time increases rapidly as size of the catalogs increases. Now considering the exception of enormous datasets containing hundreds of millions of objects from new galaxy surveys, it has led to the necessity for investigating parallelization or some form of acceleration for this algorithm/calculation.

2.4 Distance Measures in Astronomy

Correlation analysis depends on distance estimation. In appendix A.1 we review celestial coordinates and the calculation of angular separations in the sky. In brief: positions in the sky are given in terms of right ascension (α) and declination (δ), where alpha is the projection of longitude into the sky and delta is the projection of latitude into the sky. The angular separation between objects is then given by:

$$\Delta\sigma = \arccos\left(\sin\delta_1 \sin\delta_2 + \cos\delta_1 \cos\delta_2 \cos\Delta\alpha\right) \tag{2.10}$$

To obtain 3-dimensional information in galaxy surveys, one needs to estimate the distance to the galaxy. This is obtained by measuring the redshift of known features in the spectra of galaxies (more detail in Appendix A.2). In the ΛCDM model the so-called comoving distance is related to redshift by:

$$\chi(z) = \frac{c}{H_0}\int_0^z [(1+z')^2(1+z'\Omega_{m,0}) - z'(2+z')\Omega_{\Lambda,o}]^{-\frac{1}{2}} dz' \tag{2.11}$$

where H_0 is the Hubble constant, $\Omega_{m,0}$ is the energy density today and $\Omega_{\Lambda,0}$ is the energy density in the lambda component (ie. the dark energy component) both in units of the critical density: $\rho_{c,0} = 9.47 \times 10^{-27} kg/m^3$. In the ΛCDM model $\Omega_{\Lambda,0} = 1 - \Omega_{m,0}$. The spatial distance separation between two objects (A1, A2) with given redshifts (z_{A1}, z_{A2}) respectively can be expressed by [11]

$$\Delta d = \sqrt{(\chi(z_{A1})\chi(z_{A2}))^2 - 2\chi(z_{A1})\chi(z_{A2})\cos(\Delta\sigma)} \tag{2.12}$$

2.5 Parallel Computing

Parallel computing is described as a form of computation where by many calculations are carried out simultaneously [87], using the principle that large problems can be broken down into smaller ones. There exists various forms of parallel computing often categorized as; bit-level, instruction level, data and task level parallelism. Parallelism has been employed for several years, particularly in the field high-performance computing (HPC) and it is a rapidly evolving field within computer science.

This section covers a brief overview of parallel computing and the development of HPC

systems, followed by a overview of the hardware classification of parallel computing systems and some main approaches taken to create parallel programs in these systems.

2.5.1 Parallel Computing and The History of HPC

Computer systems these days are highly complex, comprised of multiple component functional units, which enables them to operate simultaneously. Therefore, a computer is able to fetch datum from memory, multiply two floating point numbers and determine a branch condition all at the same time. This form of parallelism is a very low level of parallel processing commonly known as "instruction-level parallelism"[10]. Processors capable of supporting this are classified as having a super-scalar architecture, a common feature in general-purpose microprocessors today, including those used in laptops and desktop PCs. The careful ordering of these operations are critical for optimal parallelism. Most of the work required to order operations so that instruction level parallelism is utilised is performed by the compiler. However, studies [10] show that typical applications are not likely to contain more than three or four different instructions that can be fed to the computer simultaneously. Thus, there is a limit in the pay-off for hardware support for this instruction level parallelism

In the 1980's, computer vendors exploited an alternative form of architectural parallelism and built machines consisting of multiple processors that share memory, configured on a chip. This led to the development of HPC systems primarily designed based on powerful monolithic architecture systems. However, it was soon discovered that this approach had inherent physical limitations that restricted the development rate of these machines. Multiprocessor machines were then developed in order to distribute the workload over multiple execution threads, thus permitting a single processor to execute multiple instructions in an interleaved way. This approach is known as multicore, which together with simultaneous multi-threading and shared-memory parallel computers provide system support for execution of independent multiple instruction streams. These machines are classified as Symmetric Multiprocessing (SMP) systems and they managed to solve the operating frequency performance limitation. However, there were limitations to the scalability of such design, due to the supporting memory subsystems that operate on slower speeds. An illustration of a shared-memory SMP computer system is shown in *Figure*2.11 with it's inherent memory bottleneck.

Larger computational problems, led to the development of distributed memory-shared parallel systems. These distributed computing systems are constructed consisting of groups SMP computers often referred to as hosts or nodes which communicate over a

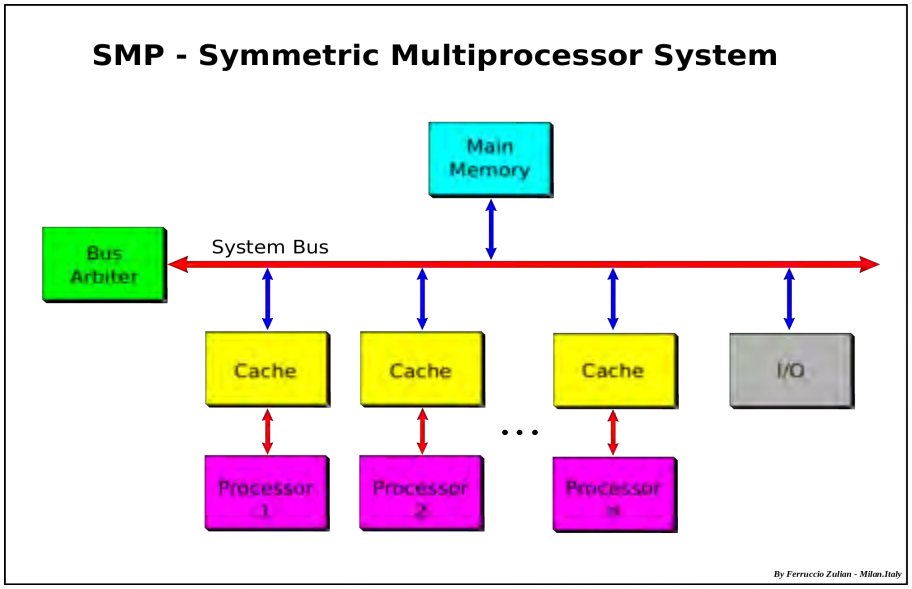


Figure 2.11: Symmetric multiprocessor system architecture diagram, where each processing unit (CPU) has access to a private local cache as well a the main shared memory. The available bandwidth of the shared memory imposes a bottleneck on this architecture [10].

network in order to distribute the workload for larger computational problems. This technology approach is commonly known as cluster computing. It has matured of recent years and since the inception of the Beowulf cluster in 1994, computer clustering has become somewhat a standard approach in the field of high performance computing. This is primarily due to the cost effectiveness, scalability, flexibility and expandability inherent in cluster's standardised design[78]. Since November 2012, clusters represent 82% of the TOP500 most powerful computer systems in the world[79].

Although SMP's are the most widespread kind of parallel systems in use today, there exist many other kinds of hybrid computing approaches employed particularly for high-end applications. These hybrid computing systems can include non-x86 task specific accelerators such as General Purpose Graphical Processing Units (GPGPUs), Intel Xeon Phi Coprocessors and Field Programmable Gate Arrays (FPGAs). An example of such a hybrid application platform is presented in [83].

2.5.2 Classification of Parallel Computers/Hardware

The major constraint in designing HPC parallel computing systems is in memory and the communication overhead associated with accessing it. Computer architectures are either classified as Uniform Memory Access (UMA) systems or Non-Uniform Memory Access (NUMA). The UMA is typically achieved by shared memory systems in which the memory

is not physically distributed, whereas the NUMA is usually implemented in distributed memory systems. Memory is also structured hierarchically with small, fast memories located close to the processor called memory caches. Caches store temporary copies of memory and for parallel computer systems there is a need to maintain cache coherency for correct program execution. In HPC, designing large cache coherence systems is very difficult in computer architecture, thus shared memory systems do not scale as well as distributed memory systems. Figure 2.12 shows some of the different memory architectures for parallel computer systems.

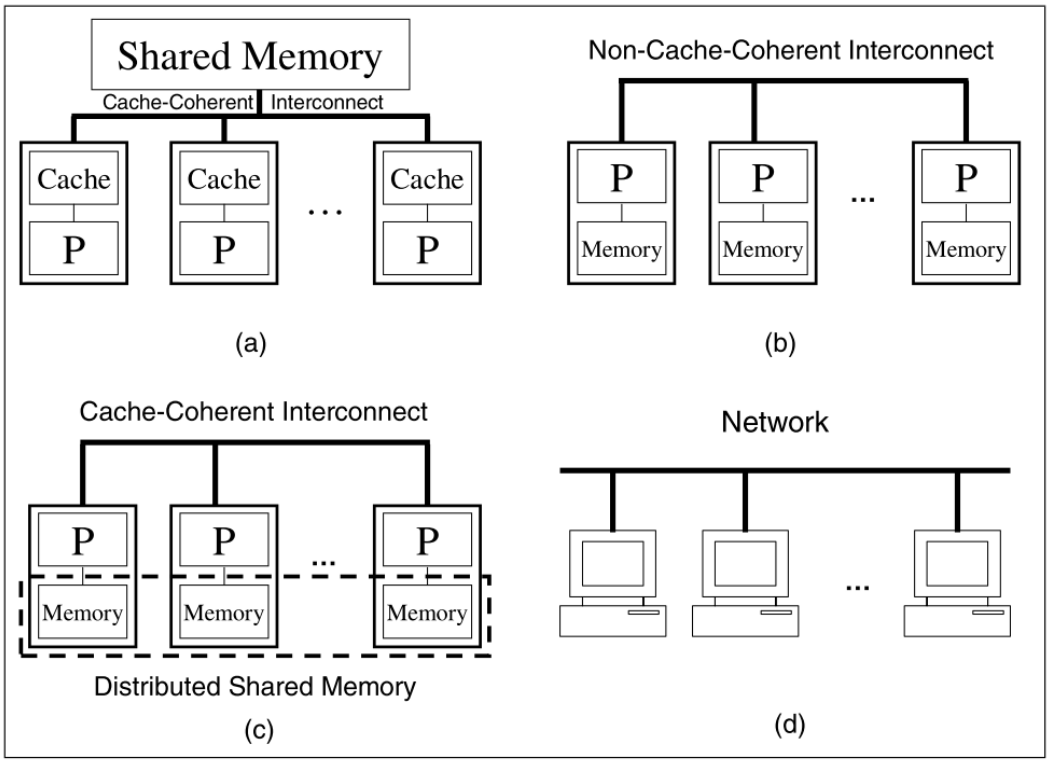


Figure 2.12: Distributed and shared-memory computers; (a) has physically shared memory, whereas the others have distributed memory. However, the memory in (c) is accessible to all processors. [10]

There are several ways to implement processor-processor and processor-memory communication in hardware such as; crossbar switch, shared bus or interconnect networks. Also, parallel computers can be roughly classified according to the level at which their hardware supports parallelism. Their classification is broadly related to the distance between computing nodes.

The following classifications presented below are not necessary mutually exclusive:

- **Multi-core computing:** A multicore processor is a single computing components which includes multiple execution units ("cores") that read and execute computer

instructions. These processors are different to super-scalar processes. Super-scalar processors can issue multiple instructions per cycle from one thread, whereas multicore processors issue multiple instructions per cycle from multiple threads. It possible to have each core in a multi-core as a super-scalar as well.

- **Symmetric multiprocessing:** A symmetric multiprocessor (SMP) is shared memory parallel computer system with identical processors, with each processor executing different programs and working on different data connected via a bus. Most multiprocessor systems today use SMP architecture. Also, with regards to the multi-core processors, the SMP architecture applies to the cores and treats them as separate processors. SMP are extremely cost-effective, provided that a sufficient amount of memory bandwidth exists.

- **Distributed computing:** Distributed computing is a distributed memory computer system in which the processing elements are connected by a network. In this system, each processor has it own private memory and information is exchanged by passing messages between the processors [23].

- **Cluster computing:** Cluster computing systems are distributed computers constructed by using standard workstations ("nodes") connected by a network. Load balancing (effective distribution of work load across available computing resources) is easier if machines in a cluster are symmetric. Modern clusters require a high bandwidth and low-latency interconnection network. This is can be achieved with standard off-the-shelf network hardware such as Gigabit Ethernet or InfiniBand.

- **Massive parallel processing:** Massive parallel processors (MPPs) are another form of distributed computing with similar characteristics as clusters. However, MPPS have specialised interconnect networks unlike standard off-the-shelf hardware used in cluster systems. MPPs also tend to be bigger than clusters and more expensive to build.

- **General-purpose computing on graphics processing units (GPGPU):** General-purpose computing on graphics processing units (GPGPU) makes use of specialized co-processors known as a graphical processing unit (GPU). These device are highly optimized for computer graphics processing, which is typically dominated by data parallel operations. A single GPU-CPU framework can provide advantages surpassing that of multiple CPUs on their own. Thus a GPGPU pipeline can be described as a kind of parallel processing systems between one or more GPUs and CPUs to analyses data as if it were in image or graphic form.

2.5.3 Parallel Programming Languages/Software

Given the various paradigms of constructing parallel computing systems described above, various programming methods and APIs (Application Programming Interfaces) have been developed for programming these machines. The classification of these methods can be divided generally according to the assumptions they make about the underlying memory architecture. Thus, in a shared memory system the programming languages communicate by manipulating shared memory variables, whereas distributed memory system uses massage passing. OpenMP (more info [85]) and POSIX Threads are most commonly used APIs in shared memory systems and for distributed systems Message Passing Interface (MPI) is the most common.

CAPS entreprise and Pathscale are also coordinating their effort to make HMPP (Hybrid Multicore Parallel Programming) directives an Open Standard called OpenHMPP. The OpenHMPP directive-based programming model offers a syntax to efficiently offload computations on hardware accelerators and to optimize data movement to/from the hardware memory. OpenHMPP directives describe remote procedure call (RPC) on an accelerator device (e.g. GPU) or more generally a set of cores. The directives annotate C or Fortran codes to describe two sets of functionalities: the offloading of procedures (denoted codelets) onto a remote device and the optimization of data transfers between the CPU main memory and the accelerator memory.

2.5.4 The Centre for High Performance Computing

The Centre for High Performance Computing (CHPC) is the largest HPC facility in Africa, located in Rosebank, Cape Town South Africa. It is an initiative funded by the South African Department of Science and Technology, managed by the Meraka Institute of the Council for Scientific and Industrial Research (CSIR). The Advanced Computer Engineering (ACE) Lab within the CHPC is primarily focused on research avenues within HPC, which include use of novel architectures (multi-core solutions, GPU accelerators, etc). The research conducted in this dissertation was conducted within this lab, utilising the computing infrastructure provided therein.

2.6 TPCF: Existing Research

As described in the previous section there are various ways to develop parallel computing, hence different means for parallelism of the TPCF computation. This section describes a selection of relevant codes/methodologies for computing the TPCF in parallel and the analysis of their results are presented.

2.6.1 TPCF: CUTE

The paper "CUTE solutions for two-point correlation functions from large cosmological datasets"[11], *by David Alonso* provides parallel solutions for computing cosmological TPCF estimations. The paper is accompanied by an open source code CUTE, short for "Correlation Utilities and Two-Point Estimation". The code is available in two versions; one developed for shared-memory systems using OpenMP and the other developed for GPU accelerators using NVidia's CUDA architecture.

There are 4 different types of correlation functions implemented in CUTE using various binning schemes and speed up methods. In this section, an outline of the correlation functions discussed in CUTE is presented, together with the methodology followed for parallelism and the performance evaluation of the code.

1. **Types of correlation functions**

   The distances used in the definition of the different types of correlation functions are shown in Figure 2.13 below.

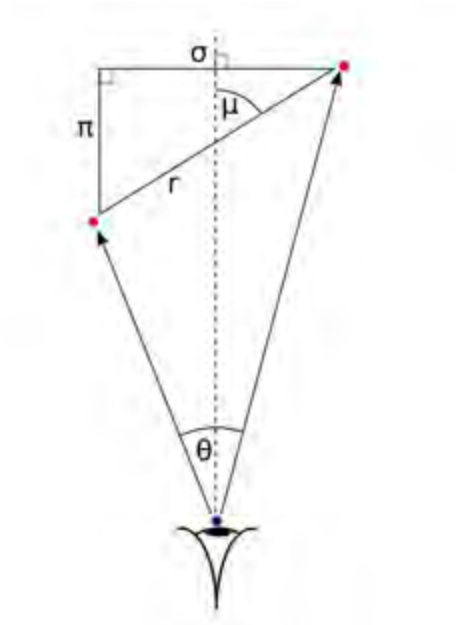


Figure 2.13: Definition of the different coordinate conventions used in CUTE. [11]

The four types of correlation functions implemented in CUTE are defined as follows:

- **3-D correlation function**: this function is described here in terms of a two coordinate system, either as $\xi(r, \mu)$ or $\xi(\sigma, \pi)$. The relation between these schemes for representing the 3-D functions can be expressed in the equations below

$$\pi = r\mu, \quad \sigma = \sqrt{r^2 - \pi^2} \tag{2.13}$$

  The $r - \mu$ scheme is usually preferred because it can be written as a multi-pole expansion according to these variables:

$$\xi(r, \mu) = \sum_l \xi_l(r) P_l(\mu) \tag{2.14}$$

  wherein P_l is the Legendre polynomials.

- **Angular correlation function**: this function is described as the TPCF of the density contrast field projected onto a sphere. It is expressed in terms of the $\xi(r, \mu)$ as follows

$$\omega(\theta) = \int dz_1 \phi(z_1) \int dz_2 \phi(z_2) \xi(r(z_1, z_2, \theta), \mu(r(z_1, z_2, \theta)) \tag{2.15}$$

  where $\phi(z)$ is the redshift selection function, $r(z_1, z_2, \theta)$ and $\mu(z_1, z_2, \theta)$ are expressed as

$$r(z_1, z_2, \theta) = \sqrt{\chi^2(z_1) + \chi^2(z_2) - 2\chi(z_1)\chi(z_2)\cos\theta} \tag{2.16}$$

$$\mu(z_1, z_2, \theta) = \frac{|\chi^2(z_1) - \chi^2(z_2)|}{\sqrt{(\chi^2(z_1) + \chi^2(z_2))^2 - 4\chi^2(z_1)\chi^2(z_2)\cos^2\theta}} \tag{2.17}$$

and $\chi(z)$ represents the comoving distance to the redshift z.

- **The monopole correlation function:** describes the angle-averaged function, derived from the first expansion of equation 2.14 expressed as

$$\xi_0(r) = \frac{1}{2}\int \xi(r, \mu)d\mu \tag{2.18}$$

- **The radial correlation function:** is used to correlate pairs of object (galaxies) aligned along the line of sight. This function can be calculated using only the redshift difference Δz between the pair.

2. **OpenMP and CUDA parallelism methods in CUTE**

   One of the methods used for parallelism employed in CUTE is implemented using the OpenMP API for shared-memory platforms. This was achieved by adding a few simple lines of the codes in the initial code shown in figure 2.10. The method declares one private histogram per execution thread to store the pair counts. Then the first loop is divided between all the available threads and finally all the partial histograms from the threads are added into the shared histogram, while avoiding read/write collisions.

   The other method implemented in CUTE was using GPU acceleration on the Nvidia CUDA architecture. This method is a bit more complicated compared to the simple changes achieved through OpenMP. It requires careful consideration of memory allocation. This was achieved by dividing the second loop among all the execution threads and declaring only one partial histogram per block instead of per thread. In order to handle the race condition of having all threads in a block attempting to add their values onto the same histogram, the CUDA atomicAdd () function was used. This introduced a bottleneck for the algorithm, since most threads would remain idle while waiting on other threads to update their histograms. Thus, to alleviate this problem the limit is that the maximum number of threads per block should be equal to the number of histogram bins defined. Finally all the partial histograms per block are added together onto the global shared histogram, also using the atomicAdd () function.

3. **Algorithmic tweaks for CUTE**

   Apart from the OpenMP and CUDA acceleration methods, there are other algorithmic techniques implemented in CUTE to speed up the computation. This techniques take advantage of the fact that often the scale of interest for computing the TPCF is much smaller than the size of the data catalog. This means that it is not necessary to

compute the distance between all pairs. Therefore, the data needs to be structured such that it allows ways to ignore calculating distances between useless pairs (pairs beyond the distance scale of interest). The catalog data structuring techniques used in CUTE are different depending on the type of correlation function being computed.

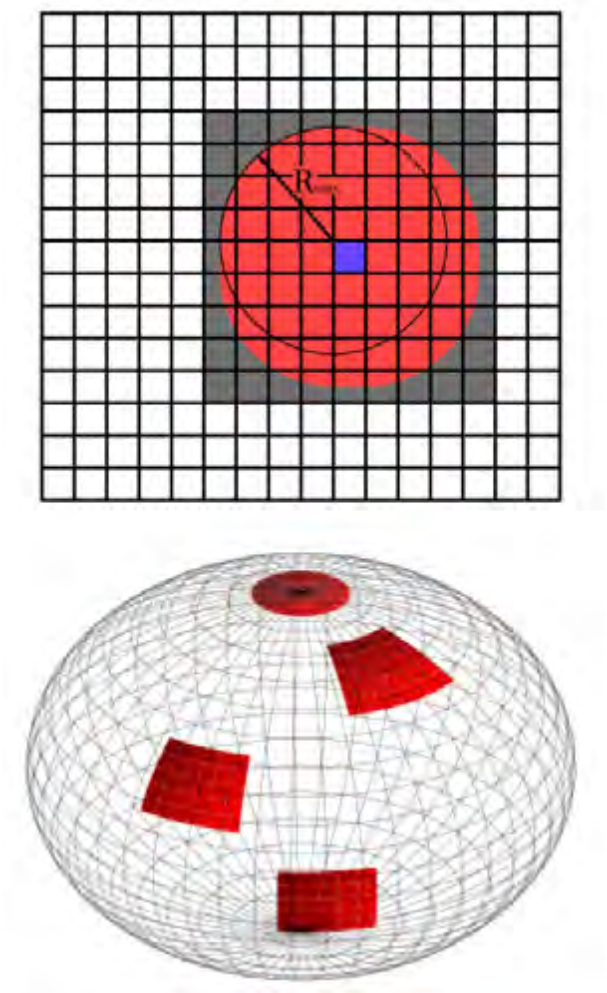


Figure 2.14: Illustration of the main neighbour-searching technique used by CUTE. In the three-dimensional case (top panel), the catalog is covered by cubical cells. Around each cell C_i (blue), a larger cube is drawn (gray), that safely contains all spheres of radius R_max centered within C_i (red). Neighbours of the objects within C_i are only searched for in the grey region. The bottom panel shows the similar neighbour-searching regions used on the sphere for the calculation of the angular 2PCF. In this case the shape of the region is different depending on the position of the central pixel. [11]

In the case of computing the 3-D correlation function, a the catalog firstly divided into cubical cells, associating each cell with the positions of the objects contained within it. Then, assuming the maximum distance of interest for the correlation function is limited to R_{max}, a larger cube in grey is constructed around the central cell C_i (blue cell of interest). Thus, enabling the construction of spheres with a radius R_{max}. This allows for an a neighbour-searching technique that ignores all

objects outside that form useless pairs and only computes distance for objects with the desired correlation distance limit. An illustration of how the catalog is broken up is shown in figure 2.14 (a)

For calculating the angular correlation function, a similar neighbour-search regions but express as spherical cube. The spherical cube is defined with constant limits of a region $\phi_0 < \phi_f$ and $\cos\theta_f < \cos\theta < \cos\theta_0$ in spherical coordinates. Using this convention, a spherical cube constructed with a maximum radius θ_{max} and a centre point at (θ, ϕ) as the following limitation

$$\Delta(\cos\theta) = \cos(\theta - \theta_{max}) - \cos(\theta + \theta_{max}), \quad \Delta(\phi) = \frac{\sqrt{\cos^2\theta_{max} - \cos^2\theta}}{\sin\theta} \tag{2.19}$$

for describing the length of the sides. The approach allows the use of pixels to be defined a small spherical cubes rather than cubical cells implemented in the 3-D case. Therefore enabling correlation of pairs within a desired angular distance θ_{max} only.

An alternative method for avoiding unnecessary correlation pairs can be achieved using the kd-Tree search method. This method is although more popular in literature, it is much sophisticated than those currently implemented in CUTE. The "Fast Algorithms and Efficient Statistics: $N - point$ Correlation Functions" by *Andrew Moore et al 2001* gives thorough description of the method and can be found in [81].

4. **Performance Evaluation for CUTE**

The results from a variety of tests/experiments conducted to evaluate the performance of CUTE were published in [11]. One of the tests was to the evaluate the performance of CUTE across 5 different computational platforms. The OpenMP version was run on a dual core $Inteli7 - 2620M$ Laptop-pc and on a large shared-memory $Nehalem - EX$ server with 80 cores, while the CUDA version was tested on a normal Laptop-PC gaming GPU and a high-end server GPU. The results for these implementations are summarised on the table 2.1 and 2.2 below.

	Name	Description	Num cores
CPUs	Sequential	Intel Core i7-2620M	1 cores (= 1 thread)
	Laptop-MP	Intel Core i7-2620M	2 cores (= 4 thread)
	Server-MP	Intel MP Nehalem-EX (x8)	80 core (= 160 thread)
CPUs	Laptop-GPU	NVIDIA NVS 4200M	48 CUDA cores
	Server-GPU	NVIDIA TELSA C2070 FERMI	448 CUDA cores

Table 2.1: Different devices in which CUTE has been tested: a single CPU core, a dual core, a multi-core shared-memory machine (160 threads), an ordinary graphics card and a high-end GPU.

Platform	$T(\xi(r))$	$T(\xi_{log}(r))$	$T(\omega(\theta))$	$T(\omega_{pix}(\theta))$	$\xi(\sigma,\pi)$
Sequential	877	5230	1374	21	2238
Laptop-MP	389	2676	628	5.3	1064
Laptop-GPU	113	185	283	6.2	297
Server-MP	25	52	32	0.51	50
Server-GPU	13	20	22	0.46	27

Table 2.2: Computational times ellapsed, for each of the 5 platforms listed in table 2.1, during the calculation of 5 different 2PCFs: monopole ($\xi(r)$), monopole with logarithmic binning ($\xi_{log}(r)$), angular ($\omega(\theta)$), angular with pixels of resolution $\Delta\Omega \equiv 5 \times 10^3$ sq-deg ($\omega_{pix}(\theta)$) and 3-D ($\xi(\sigma,\pi)$) Times are in seconds and correspond to the calculation of the DR histogram (the full calculation of the 2PCF is estimated to be 2-3 times longer)

The computational times in these results were of implementations of CUTE correlations, used without the neighbour-search feature techniques for ignoring useless distance pairs. This was done with the intent of showing fair comparisons between the platforms being tested. Also, these computation were using data from a mock catalog from the MICE project [69] constrained to: $0^\circ < DEC < 18^\circ$, $0^\circ < RA < 18^\circ$ *and* $0.5 < z < 06$, containing roughly 3×10^5 objects.

Another evaluation conducted, was testing the scaling of the code's performance with an increasing dataset. This is shown in the results in Figure 2.15, an implementation of the monopole on datasets with different sizes ranging from $10^3 - 10^7$ objects. It's important to note that the results shown here only account for the computation of the cross correlation DR and the full computation of the TPCF is expected to actually take roughly $2 - 3$ times longer than the times stated in these results.

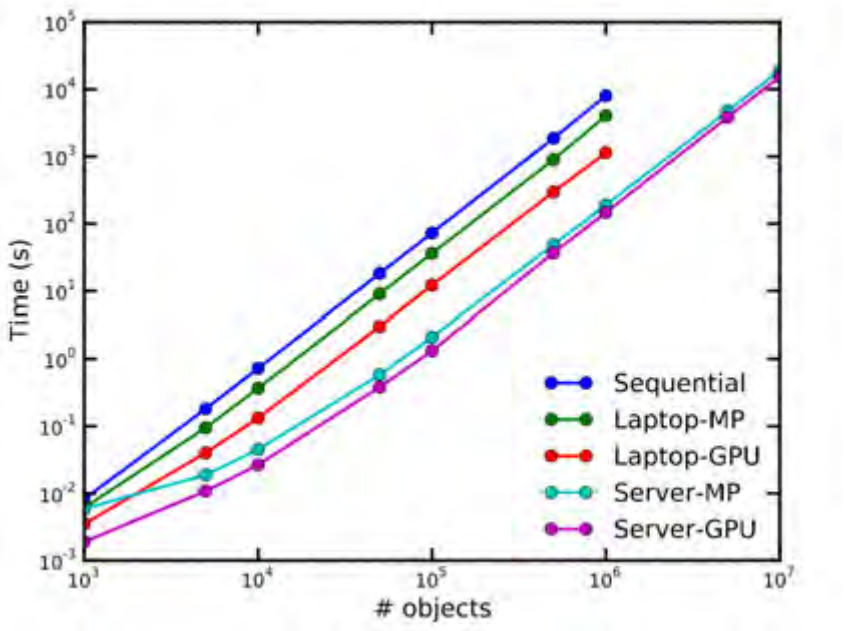


Figure 2.15: Computational times employed by different devices to compute the monopole 2PCF of catalogs of different sizes. A speed-up factor of O (100) can be gained by using a high- end GPU with respect to a sequential approach on a high-end CPU. Even with a regular gaming GPU the increase in speed is substantial (O (10)). The different devices are described in table I. [11]

2.6.2 TPCF: GP2PCF

Another related study is from the paper "Application of GPUs for the calculation of two point correlation functions in cosmology" *by Rafael Ponce et al 2012.* They investigated the advantages and disadvantages of GPU accelerators instead of CPU solutions for correlation analysis. In the study, the authors implemented an angular correlation function algorithm using Nvidia CUDA on different GPUs and compared it against standard C implementation on a CPUs in a parallel implementations using MPI on distributed nodes. The key features in their solution were:

- Used shared memory rather than global memory for the dot product and arc-cosine operations. This was done because when using the *cudaMalloc* fucntion, visibility is limited to threads within the same block on shared memory and only exist until a block of a kernel finishes.
- Used the atomic add operations in shared memory to efficiently allow multi-threading for the histogram updates. Then partial histograms were generated in parallel in shared memory and finally combined into a single histogram in global memory

- One of the architectures, they implemented a multi-GPU solution using 3 GPUs to each for compute DD,RR and DR respectively.

A summary of the devices used in this study is shown in figure/tables 2.16 below. A comparison of execution times for the standard CPU implementation against that on the different GPUs, is also shown in 2.17

CPU	GPU	MPI
CPU with two Intel Xeon E5520 processors at 2.27 GHz	GTX295 C1060 (Tesla) C2050 (Tesla)	1920 cores (two Intel Xeon E5570 at 2.93 GHz, per node)

Figure 2.16: Hardware specifications of the devices used in the GP2PCF study[12]

Eusebio Sánchez and Ignacio Sevilla

Input file lines	CPU (s)	GTX295 (s)	C1060 (s)	C2050 (s)
$0.43 \cdot 10^6$	$3.60 \cdot 10^4$	$3.01 \cdot 10^2$	$2.91 \cdot 10^2$	$2.19 \cdot 10^2$
$0.86 \cdot 10^6$	$1.44 \cdot 10^5$	$1.20 \cdot 10^3$	$1.16 \cdot 10^3$	$8.76 \cdot 10^2$
$1.00 \cdot 10^6$	$1.98 \cdot 10^5$	$1.61 \cdot 10^3$	$1.56 \cdot 10^3$	$1.17 \cdot 10^3$
$1.29 \cdot 10^6$	$3.24 \cdot 10^5$	$2.68 \cdot 10^3$	$2.59 \cdot 10^3$	$1.97 \cdot 10^3$
$1.72 \cdot 10^6$	$5.76 \cdot 10^5$	———	$4.64 \cdot 10^3$	$3.51 \cdot 10^3$
$3.45 \cdot 10^6$	$2.32 \cdot 10^6$	———	$1.88 \cdot 10^4$	$1.41 \cdot 10^4$
$6.89 \cdot 10^6$	$9.22 \cdot 10^6$	———	$7.45 \cdot 10^4$	$5.61 \cdot 10^4$

Figure 2.17: Comparison between CPU execution time and diverse GPUs execution time.[12]

The results indicate that the GPU provides a significant speed up of about a 100-fold to the initial standard CPU execution. The study also compared GPUs against several MPI configurations (64, 128, 256 and 512 nodes), with the MPI surpassing the performance of the GPU configurations consisting of more than 64 nodes. This is illustrated in 2.18, with the MPI and GPU time in a box-plot graphic.

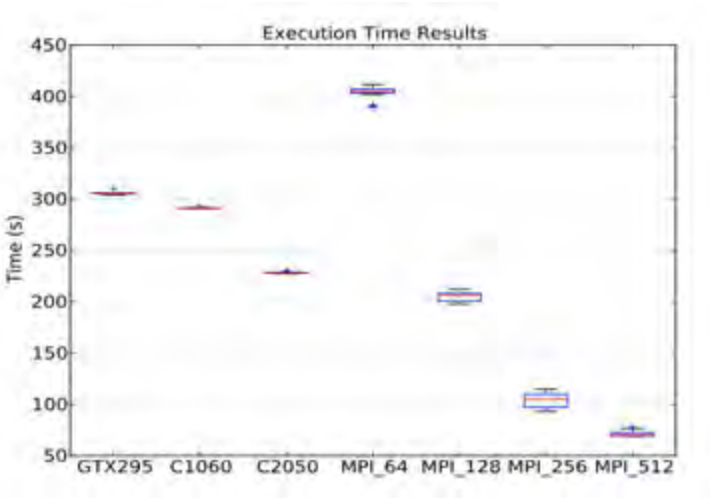


Figure 2.18: Comparison between GPUs against MPI execution times.[12]

2.6.3 Application TPCF in cosmology

In the paper "New constraints on σ_8 from a joint analysis of stacked gravitational lensing and clustering of galaxy clusters"[13] by *by Mauro Sereno et al 2015*, the clustering of clusters detected in SDSS was combined with observations of gravitational lensing to constrain the parameter σ_8. They computed the TPCF for four subsamples of $\sim$ 7000 clusters which they divided by richness class. (Richness describes the number of galaxies in a cluster and is used as a proxy for the mass). Figure 2.19 shows their results, illustrating that more massive clusters are more clustered.

Galaxy clusters are biased tracers of the underlying dark matter with bias, b defined as

$$b = (\xi_{gal}(r)/\xi_{DM}(r))^{1/2} \tag{2.20}$$

where ξ_{gal} represents the clustering of the clusters and ξ_{DM} the clustering of dark matter.

In simulations it is possible to measure the bias associated with different mass clusters. Seljak and Warren (2004)[15] provided a calibration of this relation (see equation 7.2). It is thus possible to measure the clustering of a sample, determine bias from equation 2.20 and then use the bias-mass relation from simulations such as [15] to infer the mass of objects in the sample.

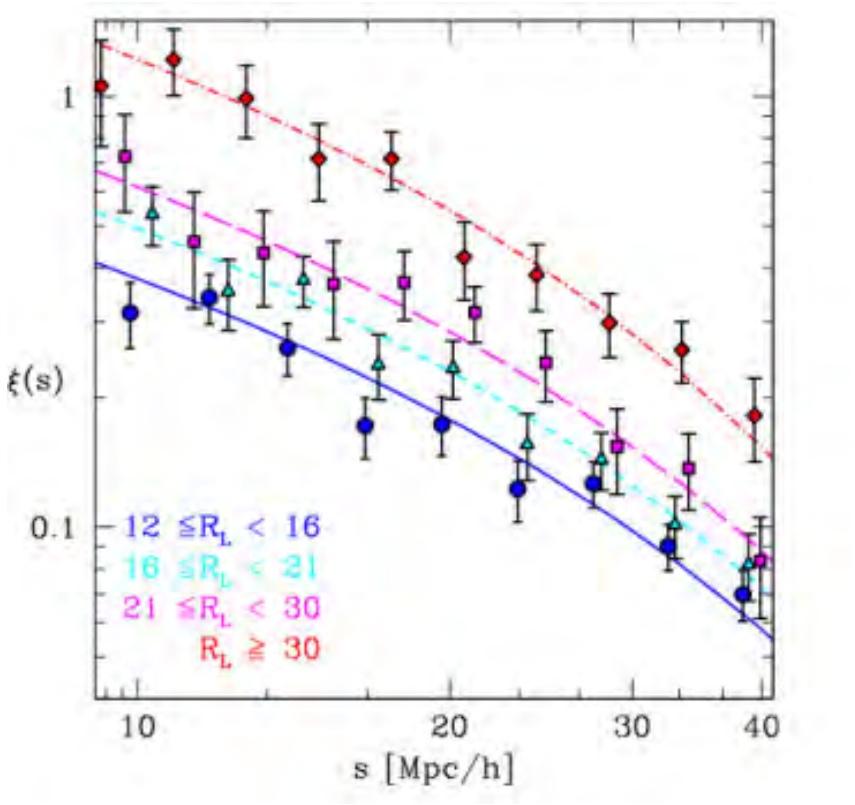


Figure 2.19: The redshift-space two-point correlation function of the four richness-selected cluster samples (dots), compared to the best-fit model. The blue, magenta, purple, and red colour codes refer to the $12 < R_L < 16$, $16 < R_{L_*} < 21$, $21 <= R_{L_*} < 30$ and $R_{L_*} > 30$, respectively. The error bars show the square roots of the diagonal elements of the covariance matrix. [13]

2.7 Application of the Review of Literature in the Remainder of the Dissertation

This chapter presented the background information on different topics related to the theme of this dissertation.

A scientific context was provided in section 2.1, containing an overview of cosmology and the standard theoretical model. This was followed by analysis of cosmology from the Planck satellite, highlighting the conflict of their recently published results and some possible ways to address the issue. In this dissertation we use the TPCF presented in section 2.3.1 as a tool for measuring clustering. In chapter 5 we explore a prototype solution for measuring the spatial 3D TPCF and draw comparison with precious work. Then in chapter 7 we use the clustering signature as a probe for the cluster mass.

This chapter also provided the engineering context for this dissertation, covering a broad overview of HPC presented in section 2.5 and section 2.6 with a focussed analysis of

existing parallel codes used for measuring the TCPF. In chapter 6, we explore the computational advantages of using shared-memory systems and GPU accelerators respectively using the CUTE code to measure the 3D spatial correlation function. We further explore the efficiency of the boxing scheme employed in the code and test it's scaling with an expected increasing dataset. The best solution is then subsequently used in the study presented in chapter 7.

Chapter 3

Research Methodology

This chapter provides an overview of the research methodology undertaken to investigate computational techniques necessary to compute correlation functions efficiently on large datasets, as well as applying these techniques to clustering studies in cosmology to answer the question about the mass of galaxy clusters. Firstly, a plan of development is described in section 3.1. Then a discussion on the research environment set-up is provided. This includes the configuration of the hardware, software and libraries, as well as the datasets (catalogs and masks) needed in this research. The chapter ends with a clear identification of the experiments to be performed and the data collecting methods involved.

3.1 Plan of Development

The process followed during this investigation can be summarised as follows:

1. Develop scripts to be used for data preprocessing. This involves extracting data from catalogs, generating appropriate masks and random catalogs required to compute the TPCF.
2. Define a conceptual prototype algorithm and develop python code/scripts to compute angular and 3D spatial TPCF. This provides the grounding necessary for understating the accelerated codes studied later.
3. Test the prototype and validate code's output with previous work in the field.
4. Review existing available acceleration techniques such as CUTE & GP2PCF from literature

5. Compare the results of the prototype python scripts against the CUTE outputs.

6. Test the CUTE against larger datasets and evaluate the performance of the code. Make adaptation to the code and optimise performance if necessary.

7. Use the code to tackle a science question, by using the clustering signature of galaxy clusters to probe the mass of the clusters

8. Finally, analyse and interpret the results and provide recommendations for future work.

Figure 3.1 shows an illustration of these steps based on the research question raised in section 1.2.1.

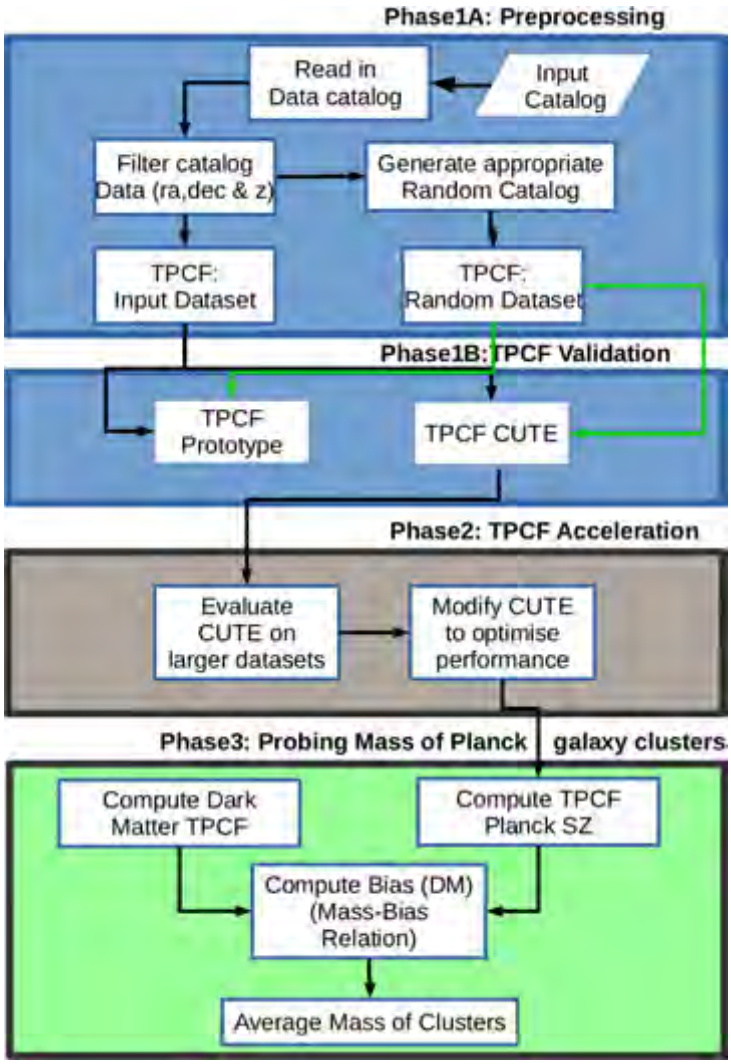


Figure 3.1: Diagram to illustrate the different phases for the dissertation life cycle. The phases are grouped based on research questions this dissertation is set out to address.

3.2 Research Environment Set-up

The section details the project environment set-up prior to the development and implementation of the TPCF. This involves hardware specification and configuration of the software libraries, dependencies and compilers needed to implement the TPCF efficiently.

3.2.1 Computational Hardware available

The main computational methods to be investigated in this project are based on solutions implemented on CPUs configured as shared-memory systems and GPUs accelerators using the Nvidia architecture. A summary of the different computational devices that were available during this investigation is listed in table 3.1. The CPU shared memory systems consists of: a quad core Lenovo laptop and two different server nodes, one with 20 Intel Xeon E5-2690 cores, the other with 24 Intel Xeon E5-2670. The GPU platforms consisted of the high-end Nvidia Tesla K40 and K80.

	Name	Description	Num cores	Memory
CPUs	Lenovo-Z580	Intel Core i7-3612QM	4 cores	6 GB
	Server-Research-Node	Intel Xeon E5-2690	20 cores	128 GB
	Server-C4130-Node	Intel Xeon E5-2670	24 cores	132 GB
GPUs	Server-k40-GPU	NVIDIA TELSA K40	1248 CUDA cores	12 GB
	Server-C4130-GPU	NVIDIA TELSA K80	4992 CUDA cores	24 GB

Table 3.1: List of different devices available to be used in this study.

3.2.2 Ace lab Cluster configuration

Figure 3.2 shows the layout of the network topology for the ACE Lab's HPC cluster. The storage server (node) is use to host a shared file system, while the head node is used for management, compilation and as an interface for the user to gain access to the cluster. The cluster consists of six compute nodes, each with two Intel Xeon E5-2690 v2 IvyBridge CPUs (20 physical cores in total) and 128 GB of DDR3 RAM, as well as four GPU nodes, each containing two Intel Xeon E5-2695 v3 IvyBridge CPUs (28 physical cores in total), 132 GB of DDR3 RAM and two Nvidia Tesla K40 cards. The Dell c4130 node, is s more compact node similar to the gpu nodes, although it consists of two Intel Xeon E5-2670 v3 IvyBridge CPUs (28 physical cores in total), 132 GB of DDR3 RAM and two Nvidia Tesla K80 cards.

The cluster runs on the CentOS 6.5 (Community Enterprise Operating System), an Linux distribution based on the Red Hat Enterprise Linux (RHEL). The operating system is actively developed for enterprise and HPC environments, widely used within the industry. A combination of TORQUE (Terascale Open-source Resource) and the Maui Cluster Scheduler is used for the scheduling and administration on the cluster.

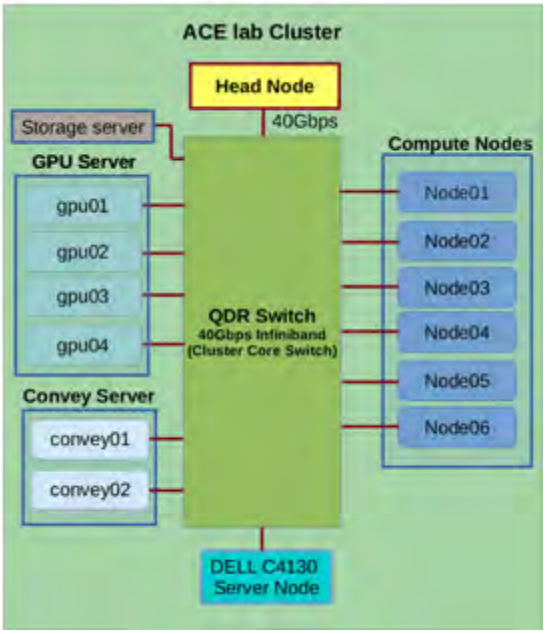


Figure 3.2: A network topology diagram representing the configuration set-up for ACE Lab's HPC cluster.

3.2.3 Software compilers and Libraries dependencies

Python is a popular programming language, that supports multiple programming paradigms, with features of a dynamic type system, automatic memory management and a comprehensive standard library[28]. It is also widely used in the field of astronomy, with a strong community of developers and variety of useful packages in astronomy research. The prototyping scripts/code are written in Python and they depend on some of these useful library packages.

These are the core packages required for the pythons scripts:

- **Python:** version 2.7 was the last major release for the 2.x, with an expected long term support that leads on to 2020. Python 3.x was not considered, as some astronomy packages such astroML are not supported yet.[74]
- **Numpy:** is fundamental package for scientific computing with Python. Here, it

was used for various reasons including it's powerful multi-dimensional array objects, broadcasting functions and vectorization among other useful functions.

- **Scipy:** required for the efficient user friendly numerical routines such as numerical integration, optimization, etc.
- **Matplotlib:** is a useful python 2D plotting library. Here it's required to produce the figures and plots of the results throughout this investigation.
- **Astropy[73]:** is required to read Flexible Image Transport System (FITS) files. Most catalogs and datasets used in astronomy are stored this FITS format
- **Cosmolopy[62]:** is a required to convert the angular diameter distance from the given redshift z of the galaxy clusters.
- **Healpy[60]:** provide an interface to the HEALPix pixelation scheme, and is required to generate the mask map for the random catalog

The other codes that were explored and tested in this investigation are the CUTE (OpenMP & CUDA) and the GP2PCF. The codes where written in C and required some libraries for compiling and running the code. These are the key library packages necessary:

- **gcc C compiler:** version 4.8.4 for the Ubuntu distribution
- **GNU GSL library:** (version 1.16) numerical library for C and C++ programs, providing a wide range of mathematical routines such random number generators, special functions, etc.
- **CUDA:** Toolkit 7.5 is the latest version of cuda available and offers a comprehensive development environment for GPU accelerated applications.

The libraries selected above were the latest available packages during the course of the investigation. The first two package are relatively easy to install relative to the CUDA environment. It is very important to ensure that the environment variables are sourced to the correct path and directories containing these libraries. Details on how to install and set-up the CUDA environment can be found [30]

3.2.4 Data catalogs

HIPASS (the HI All Sky Survey)[61] is a wide are survey of galaxies detected via their neutral hydrogen (HI) emission by the Parkes radio telescope. These are the kind of sources that will eventually be detected by the SKA and clustering analysis with HIPASS was carried out locally, thus making this a good initial test catalog for this study.

As already mentioned, the main interest in this study is to investigate efficient computational techniques for measuring the 3D spatial correlation functions and using these functions to probe the masses of galaxy clusters. Therefore the main selection criteria for the catalogs used in this study was based on their size to test the computational aspects of the project, scientific significance in terms of the cosmological studies (mass/richness of the clusters) and redshift information for computing the 3D distance. The Sloan Digital Sky Survey (SDSS) is one of the largest ongoing surveys, with best maps of large scale structure containing spectroscopic redshifts of thousands of galaxy clusters. The catalogs considered in this study are the GMBCG cluster catalog from SDSS-DR7 [14] and the SDSS-III[22] catalog, respectively.

Although the Planck SZ catalogs only contains a few thousand clusters of galaxies, they pose a very interesting question in terms of the results from the cluster counts and mass, as mentioned in section 2.2. A list of the catalogs used in this investigation is given below. The Planck catalogs are accompanied by their selection function masks, that are useful in generating the random catalogs used to cross correlate with the data catalog. These catalogs and selection masks are publicly available on [20]. Table 3.3 list the the selection function/mask files that were used with the Planck's SZ catalogs.

	Name	Num of Objects	Objects with Redshift	Year Released
HIPASS	Hi_cat.txt	4315	-	2009
SDSS Clusters	GMBCG_full Cluster_DR9SZ	55 424 132 684	20 119 52 683	2010 2012
PlanckSZ Clusters	PSZ_union_validation_v1.fits HFI_PCCS_SZ-union_R2.08.fits	1227 1653	813 1094	2013 2015

Table 3.2: List of galaxy and cluster catalogs used during the clustering investigation.

Catalog	Mask
PSZ_union_validation_v1.fits	COM_PCCS_SZ-unionMask_2048_R1.11.fits
HFI_PCCS_SZ-union_R2.08.fits	HFI_PCCS_SZ-selfunc-union-survey_R2.08.fits HFI_PCCS_SZ-selfunc-union-cosmolog_R2.08.fits

Table 3.3: A list of selection function masks used for generating the random fields for the PLANCK catalog datasets..

3.3 Research Experimentation

This section presents the different investigations that are going to be performed in order to answer the research questions of the project presented in section 1.2.1.

3.3.1 Experiment 1: Spatial 3D TPCF vs previous work

This experiment is designed to address the question regarding the algorithm used for computing the spatial (3D) TPCF function in astronomy. It involves exploring a prototype solution based on literature and testing that solution against previous results. In addition, prior data preparation procedures are necessary, since the data from the catalogs is required as input for the correlation function.

Therefore, the following procedure is follwed to perform this experimentation:

1. Develop python scripts for dataset pre-processing:
   - Read the data catalogs and extract (RA, DEC, z, Mass), information from the datasets
   - Filter the data information with respect to desired (RA, DEC, z) limits
   - Group data according to the desired mass bins, using richness for the SDSS
   - Generate the random catalog to cross correlate with the data catalogs. (*NB: the random catalogs need to have same distribution (redshift and boundaries) as the input catalogs*
   - Store the input and random catalog in a format (RA,DEC, z) to be used for the next steps (Computing TPCF)
2. Develop python scripts to prototype the 3D TPCF algorithm.
3. Compute the the correlation function using prototype scripts and the datasets produced in step 1

4. Compare the output of the prototype scripts and compare with previous results

3.3.2 Experiment 2: Code Review and Performance Test

The goal of this experiment is to investigate accelerated solutions for computing the correlation function and evaluating their performance against different catalog datasets with various lengths and density. It also involved looking at methods to optimize these solutions. CUTE was identified as an appropriate code to conduct this experimentation.

The steps to achieve this goal are as follows:

1. Generate the random datasets of various lengths based on the distribution of the SDSS and Planck cluster catalogs
2. Compute the spatial 3D correlation function on the different available platforms using CUTE (both OpenMP and CUDA versions) and increasing datasets produced in step 1
3. Time the computational time for correlating the different datasets and evaluate the performance of the code
4. Modify the code in CUTE to investigate methods to optimize the computation of the 3D spatial correlation function.
5. Evaluate the performance of the modified CUTE code.

3.3.3 Experiment 3: Code application in cosmology

This experiment deals with the practical application of the TPCF in the context of cosmology, by using the function to probe the mass of the Planck galaxy clusters. This required some additional astronomy concepts, involving the computation of the bias and models based on standard structure formation theory. We also measure the clustering in different richness classes of SDSS to compare with the Planck clusters

This process followed for this experiment is described below:

1. For SDSS, generate datasets grouped according to the different mass bins from the input catalog. (Similar to step 1, in experiment 1)

2. Compute the TPCF of these SDSS datasets and the Planck clusters sample. Then plot them on the same axis for comparisons
3. Use the LAMBDA-CAMB Web interface to generate the power spectrum of Dark Matter, from the cosmology parameters used in the construction of the survey/catalogs.
4. Compute the clustering of Dark Matter from the power spectrum generated in step 2
5. Measured the bias(offset), using the clustering of the galaxy clusters and that of Dark matter measured in step 2, 4 respectively.
6. Identify a calibration of the bias-halo mass relation based on standard structure formation theory
7. Use the model from step 6 to infer the mass of the clusters using the bias computed in step 5.

3.4 Data Collection and Analysis Methods

This section of the methodology explains how data is obtained and recorded from the experiments described in the previous section. Each experiment has a different goal to address, hence different analysis methods are employed dependent on the experiment.

3.4.1 Experiment 1: Spatial 3D TPCF vs previous work

The data to be collected and analysed for this experiment can be summarized as follows:

- **catalog dataset plots:** plots of the data extracted from the input catalogs. This includes; 2D distribution of sources and the (RA, DEC, Z) distribution for the clusters.
- **random dataset plots:** plots of the random catalog datasets produced from information of the input catalogs. This can be compared with the data in the previous point above
- **histogram outputs** tabulates the values of auto-correlation and cross-correlation histograms (DD, RR, DR). This is used to estimate the TPCF.
- **correlation plots:** plots of the output of TPCF measured.

3.4.2 Experiment 2: Code Review and Performance Test

This experiment deals mostly with the measurement and the performance analysis of the CUTE code. Thus, the primary information to be analysed during this experiment is the computational time required to perform the function and the given resource.

This data is presented in the form of:

- **time list:** tabulates the execution time for the auto-correlation(DD,RR) and cross-correlation (DR) for different sized datasets.
- **execution time:** tabulates the execution times for the different solutions and platforms when measuring the correlation functions.
- **time figure:** this plots the variables from the time list onto the same axis for easy comparisons of the data.
- **ganglia charts:** shows the CPU utilisation of the code during runtime on the HPC node cluster.

3.4.3 Experiment 2: Code application in cosmology

The goal of this experiment is to test the TPCF developed on real data, specifically probing the masses of Planck clusters.

The information collected in this experiment consists of:

- **clustering signature:** plots of the output of the correlation functions measured of the different datasets of the catalog, separated according to their mass/richness bins from the SDSS. It also includes clustering of Dark Matter obtained from DM power spectrum.
- **DM power Spectrum:** power spectrum of Dark Matter obtained from LAMBDA-CAMB web interface, using cosmological parameters from the catalog.
- **bias offset:** describes the relationship between the clustering signature of the Planck galaxy clusters and DM.
- **mass estimates** estimate mass of the cluster halos using a bias-mass relation obtained from simulations.

Chapter 4

Dataset Preprocessing & TPCF Solution Design

This chapter details the description of the data preprocessing method implemented for extracting the required information to compute the TPCF from the data. The chapter also includes a description for the conceptual prototype solution developed, as well as the details for the the 3D boxing technique used in CUTE, for optimizing neighbour searching of objects in the datasets.

4.1 Data Preprocessing

Astronomical data catalogs contain arrays of information on the different properties of objects observed during a survey. These catalogs can be made available in different formats depending on the source of the catalog and not all the information contained in them is needed for computing the correlation function. In this chapter we look at the python script ("*loadCat_genRdat.py*") developed for the dataset preparation step, which includes; reading and filtering(redshift, RA, DEC & Richness/Mass) data, generating random catalog/s and then formatting data into the required structure.

4.1.1 Reading the Data Catalogs

The main information required for computing the spatial correlation function is the RA,DEC and redshift. The RA and Dec are also sufficient for computing the angular

correlation alone. However, our interest also involves understanding how clustering is related to richness/Mass of the galaxy clusters, so we extract information on these quantities too. A list of the catalogs that were used during this study is given in Table 3.2.

Here we provide a brief description of how the required information was extracted from different catalogs:

- **HIPASS:** this catalog was used as a test case of the initial prototype script testing the angular clustering against previous work. The catalog is available in text file (*'hi_cat.txt'*) format. The RA, DEC are recorded in columns 3, 4 respectively and stored using the *HH:MM:SS* notation. The catalog does not contain the redshift information, however column 18 contains a velocity mask which can be used to estimate the redshift. In order to read this data from the HIPASS catalog the script needs to be run as follows:

  – *"python loadCat_genRdat.py path_to_catalog -f 03"*

  where path_to_catalog is the path to the hi_cat.txt file and "-f" flag executes the functions for handling this type of catalog. This will load the data into a Numpy array and converts the RA, DEC from *HH:MM:SS* notation into degrees (°) and the velocity masks into their equivalent redshift values. Then, the data structure from this catalog is a 3-D NumPy array consisting of information about [RA, DEC, Z], with RA and DEC in degrees.

- **SDSS:** we use catalogs of clusters generated from the original galaxy catalogs. These catalogs are available as ASCII tables stored in a text file. The information about the [RA, DEC, Z_{ph}, Z_{sp},Richness] is contained in columns [1, 2, 3, 5, 21] for the GMBCG_full.txt catalog and columns [0, 1, 2, 3, 4, 7] respectively for the Cluster_DR9sz.dat catalog. The catalogs contain information for both the photometric(Z_{ph}) and spectroscopic(Z_{sp}) redshifts of the clusters. Thus, when reading the catalog one needs to choose which redshift to use between. The default selection of redshift in the script is set to use the spectroscopic redshift, however this can be changed by setting the '-z' to 0 when running the script.

  For reading the GMBCG catalog, with spectroscopic redshift. The script is executed as follows:

  – *"python loadCat_genRdat.py GMBCG_catalog "*

  and for reading the photometric redshift

  – *"python loadCat_genRdat.py GMBCG_catalog -z 0"*

The Cluster_DR9sz catalog is read in a similar way, with the exception that the '-f' must be set to 1. Then, the data structure from this catalog is a 4-D NumPy array consisting of information about [RA, DEC, z, Richness], with RA and DEC in degrees. The Richness is used as a proxy for the mass of the clusters.

- **Planck SZ:** catalogs are available in "FITS" format, with the data stored within ASCII tables. The script uses the python pyfits module to load data table from the catalog and then extract the data fields corresponding to ("RA", "DEC", "Redshift", "MSZ/M_YZ" and "COSMO"). The "MSZ/M_YZ" represent the mass of the clusters inferred using the SZ effect and the "COSMOS" is the flag for the clusters used for cosmological samples in Planck. These catalogs are provided together with the selection/mask function files that are useful when generating the random catalog. Therefore, reading these catalogs requires that we load the appropriate masks for the catalogs. The script is executed as follows when reading these catalogs:

  – *"python loadCat_genRdat.py "Planck_catalog" "Planck_mask" -f 4"*

  The data structure from reading this catalog is a 5-D NumPy array consisting of information about [RA, DEC, z, Mass, Cosmo], with RA and DEC in degrees.

Once we've loaded the appropriate data from the catalog we are interested in, we filtered out clusters that do not contain valid redshift information (i.e z=[0, -1, nan]). Also, due to the irregular distribution of clusters noticed in the Sloan catalogs and the lack of a mask for the catalog, we used the data within the continuous region ($140 < RA < 220$ and $20 < DEC < 60$).

4.1.2 Generating Random Catalog

In order to accurately estimate the TPCF, we needed to produce random catalogs with the same coverage as the input data read from the catalog. We used the coverage of the input data array to define the boundary conditions for the random catalog and used re-sampling of redshifts in order to archive the same redshift distribution as the input catalog. Due to the spherical coordinate system used to specify positions, the distribution towards the poles thins out in the rectangular projection

The HIPASS and SDSS cluster catalogs had enough data samples contained within a continuous region. Thus, we only used samples of the catalogs that were contained within those continuous region from these catalogs. Now, assuming we need to generate a random

catalog with N objects, we generate $10 \times N$ random positions within the defined (RA, DEC) boundaries from the input catalog, and then randomly select N unique positions. We then generate N samples with same z distribution as the input data catalog.

For the Planck catalogs we used the masks provided to filter regions not included in the catalogs. The mask is read in as a HEALPIX(Hierarchical Equal Area isoLatitude Pixelation) map using the healpy module.T his produces subdivisions of a spherical surface, wherein each pixel on it covers equal surface area. The "$NSIDE$" value is very important for determining the number of pixels in the Healpix map and corresponding the pixel value with the position on the map. Therefore, once we've read the mask we extract the $NSIDE$ value using the healpy function *"get_nside"*. Then the following steps are implemented in generating the random catalog:

1. We then generate $10 \times N$ random pairs of (RA, DEC) with similar boundaries as the input catalog data
2. Convert the (RA, DEC) pairs into radians (ϕ, θ), with $\phi = RA$ and $\theta = 90 - DEC$ in radians
3. We then used the healpy function *"ang2pix(NSIDE,ϕ, θ)"* to convert (ϕ, θ) into pixels, using the NSIDE value read from the mask
4. Then apply the read mask map on the pixels generated on the previous step, discarding pixels that do not belong in the mask map. (Note: We used NumPy broadcasting in this step. Thus instead of converting the pixels back into their (RA, DEC) we could index them using the mask array.
5. Then randomly select N (RA, DEC) unique pairs to form part of the random catalog

Finally, the output when generating the random data catalog is a 3D NumPy array containing data with the desired N pairs of (RA, DEC) with the same boundaries and redshift distribution as the input data catalog.

4.1.3 Formatting and Storing

After successfully reading the input catalog, filtering the data and generating the appropriate random catalogs for them, we had to store the data in a format easily usable for computing the TPCF in the next phase. For the python prototyping scripts the (RA, DEC) are

converted into their radian equivalent (ϕ, θ) and the redshift (z) converted into the comoving-distance (χ). Then the data is stored as a 3-D NumPy array(ϕ, θ, χ), as they are faster to load in python relative to a text file. The data for the RA, DEC and z is also stored into text files to be used as inputs to the CUTE code.

Therefore, the output produced from the Dataset preprocessing step can summarised and categorised as follows:

- **Data Verification**
  - Input & Random catalogs data 2D and 3D plots
  - Input & Random catalogs data RA, DEC and z distribution plots

  These plots help to verify that the generated random catalogs are within the requested constrains of the input catalog *(see figures 5.2)*.

- **TPCF Prototype Scripts input catalogs**
  - 3D NumPy array contain data of the (ϕ, θ, χ) for the input catalog
  - 3D NumPy array contain data of the (ϕ, θ, χ) for the random catalog

  Used as input and random catalogs for the TPCF pythons scripts

- **CUTE input catalogs**
  - InpuDat text file contain data of the (RA, DEC, z) from the input catalog
  - randomDat text file contain data of the (ϕ, θ, χ) for the random catalog
  - mask_file text file with defined boundaries of the InputDat
  - z_dist text file with the redshift(z) distribution of the InputDat

  These are used as input parameters for the CUTE code.

4.2 Conceptual Prototype Solution

This section presents the development of the initial conceptual prototype solution for computing the TPCF algorithm.

Figure 4.1 shows a flow diagram for a general overview of how the python prototype solution is implemented to compute the TPCF. Firstly, the input and random catalog produced from the data preprocessing phase are read. Then the separation bins to be

used are either read from a file*(bins from CUTE)* or generated by the code. Thereafter, the type of correlation function to be computed is determined *(see section 2.4)*, followed by the auto-correlation and cross-correlation of DD, RR and DR respectively*(see section 2.3)*. Once the DD, RR and DR have been computed, the TPCF is estimated using either the Landy-Szalay or Peebles & Hauser estimators. Finally, the output of the TPCF is written to a file and a plot of the function is generated.

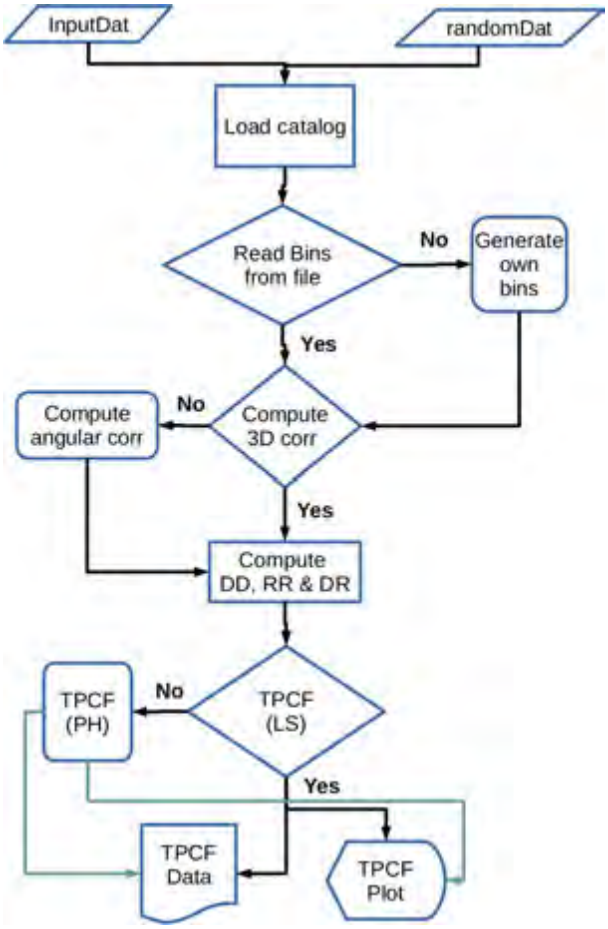


Figure 4.1: Flow diagram for the Prototype solution

The code defining the functions for the auto-correlation, cross-correlation and the estimation of the TPCF is attached in *appendix: B.1, B.2 & B.3* respectively.

4.3 CUTE 3D Boxing Solution

Often when computing the TPCF, the maximum distance scale of interest is substantially smaller relative to the size of dataset in the catalogs. Therefore, it is important to investigate methods that reduce the unnecessary calculation of pairs of objects outside the region of interest. A brief discussion on the neighbour searching methods implemented

in CUTE was presented in section 2.6.1, together with an illustration in figure 2.14. This section explores the details of CUTE's 3D boxing scheme, used when computing the spatial 3D correlation function.

Figure 4.2 shows a flows diagram of how the 3D boxing scheme is implemented in CUTE. This was derived from analysing the CUTE source code file (*"boxes3D.c"*) and some snippets of the code are provided in appendix B.4. The methods for this boxing scheme can be broken up into three main categories, which involves; first determining a box to encompass the full catalog, followed by dividing the the box into small boxes and finally associating objects with a cell box.

This method was designed such that it mostly depends on the number density of the catalog, the maximum distance the scale of interest(Rmax) and the number of boxes used[11]. This is evaluated by the *"optimal_nside"* function shown in appendix B.4 (code lines $1 - 11$). The function was defined to generate the minimal number of boxes from the whole catalog.

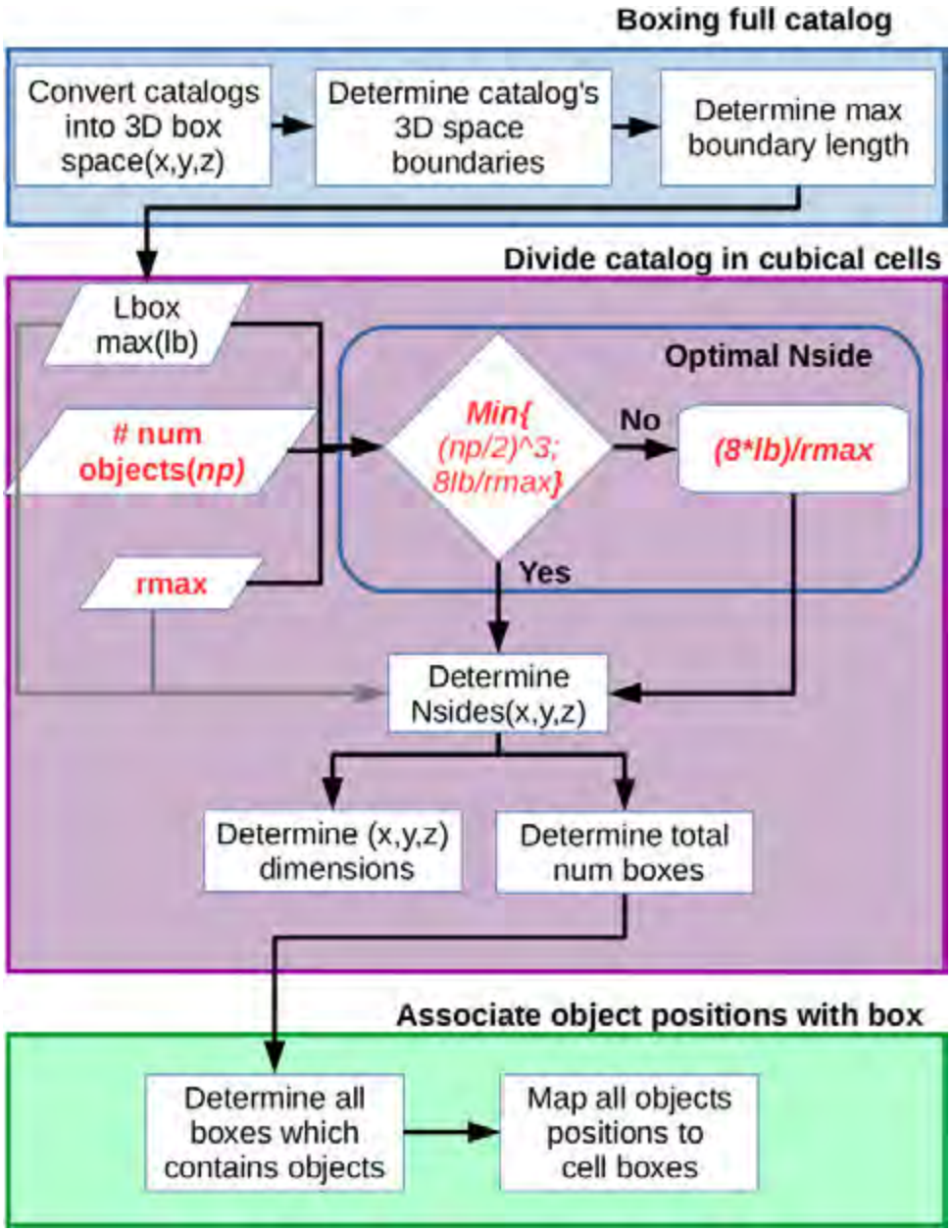


Figure 4.2: An illustration for the 3D boxing method used in CUTE to optimise neighbour searching

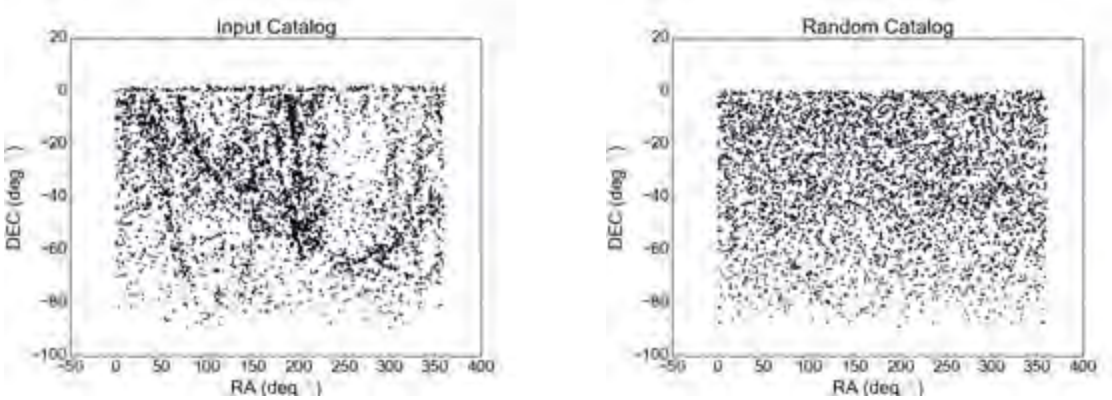
Chapter 5

Experiment 1: Spatial 3D TPCF vs previous work

This chapter details the implementation and results found during the first phase of experimentation. In this experiment, we test various aspects of our investigation such as correctly reading the data catalogs, generating the appropriate random catalogs and testing the proposed python prototype solution against known past results. Firstly, we test the python script by measuring the clustering of HI galaxies detected in HIPASS and compare with past results. Then, we further test the prototype scripts with a subset of SDSS cluster catalog.

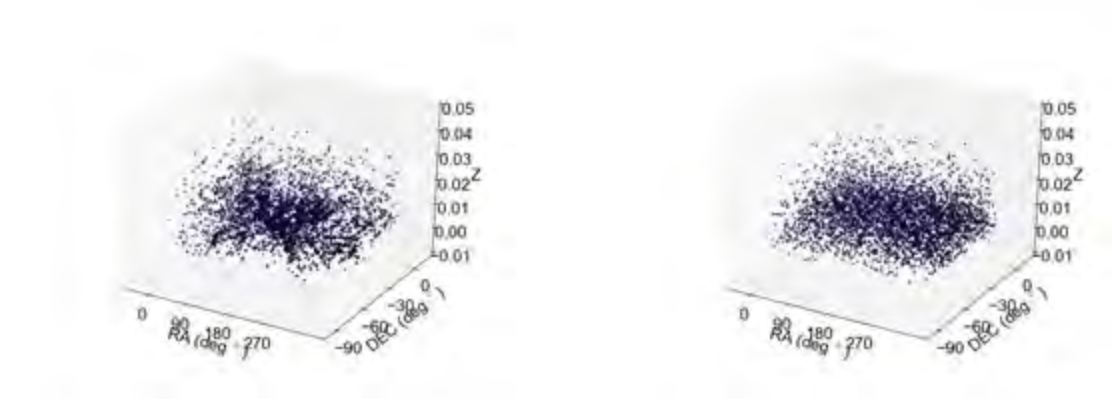
5.1 Clustering of HI galaxies in HIPASS

The HIPASS catalog has been used previously by Passmoor at el (2011)[17] to study the clustering of HI galaxies. In this section we present an initial test of the proposed prototype by measuring clustering of these galaxies to verify our solution against the results found in their paper. In section 5.1.1 we present the data extracted from the catalog and the appropriate random catalogs generated during the preprocessing step. In section 5.1.2, we compare our result with those reported in the previous study.



(a) Input (b) Random

Figure 5.1: 5.1a presentes the 2D Plot of the HIPASS input data and 5.1b the random catalog generated.



(a) Input (b) Random

Figure 5.2: 5.2a presented the 3D plot of the HIPASS input data and 5.2b the random catalog generated

5.1.1 Data

The HIPASS survey covered the whole southern sky and the catalog contained 4315 galaxies detected in HI. Using the python script of the data preprocessing step we managed to read the catalog and generate the appropriate random catalog. Figure 5.1a shows the 2D plot of the galaxies represented using their (RA, DEC) and Figure 5.2a presents 3D plot of the same galaxies using the redshift information. The plots of the random catalogs that were generated using this dataset are shown in Figures 5.1b & 5.2b.

Observing the plots of data from the input catalog, we can see a clear indication of clustering in some regions of the sky. We also notice a decrease in the density of sources towards the pole at dec= $-90°$(a result of the projection of the sphere onto a plane). The generated random catalog consist of non-clustered sources, accurately reproduced within

the same constrains of the input catalog. From the 3D plots we notice that the sources are observed at very low redshift, in the ranges $0.01 < z < 0.05$.

5.1.2 Results of HIPASS clustering

Figure 5.3a shows results from the previous study by Passmoor et .al (2011), with the blue plots measuring the angular correlation function for HIPASS data. The dashed line represents the corresponding power law fit for the data, defined by the equation 2.4 and the parameters given in table 5.1. Figure 5.3b shows our result from measuring the angular correlation using the prototype solution. We also included the fit line defined by parameters in table 5.1 and the plot shows that our results are consistent with those found in previous study to within a measurement of uncertainty.

	HIPASS
A_ω	$0.603 \pm 0.04°$
δ	0.56 ± 0.02

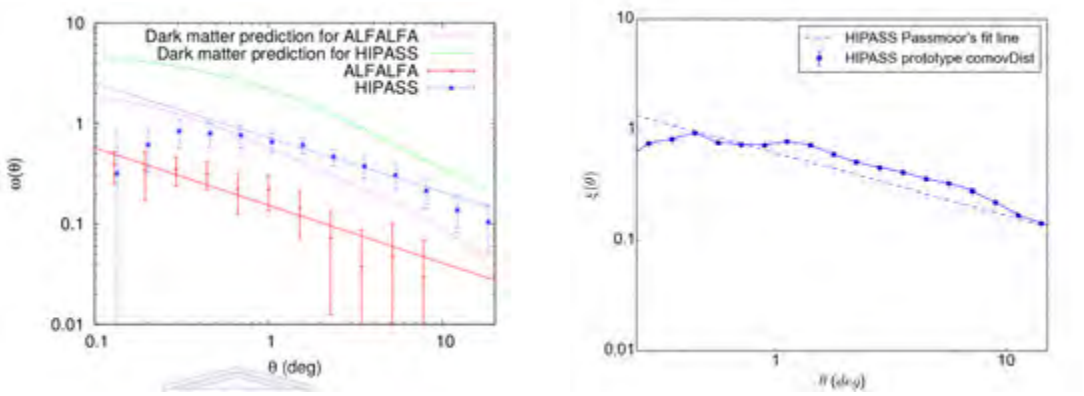
Table 5.1: The angular fitted parameters, A_ω and δ [17]

In the previous study, the authors did not compute the spatial 3D correlation function directly from the data. However, they provided a projected correlation with a fit line described by equation 2.5 and the following parameters in table 5.2

	HIPASS
r_0	$2.89 \pm 0.08 h^{-1}$Mpc
γ	1.56 ± 0.02

Table 5.2: The projected 3D clustering fitted parameters, $1/r_0$ and γ [17]

We used the prototype scripts to compute the spatial 3D correlation function directly from the data and our result is shown figure 5.4. The dashed line shows the projected correlation function from the previous study. The plot demonstrates our result from directly computing the 3D spatial correlation function are in agreement with those expected from the previous study. However, it is worth noting that the HIPASS catalog consisted of galaxies with very low redshifts($0.01 < z < 0.05$). Thus, there was still a need for a sample with a larger redshift distribution to test the prototype scripts.



(a) Previous Study[17] (b) Current Study

Figure 5.3: An angular correlation function for HIPASS, comparing the current solution with work previously in literature. The plot on the left 5.3a is from *Passmoor et al.(2011)* and the one on the right 5.3b is from our current implementation

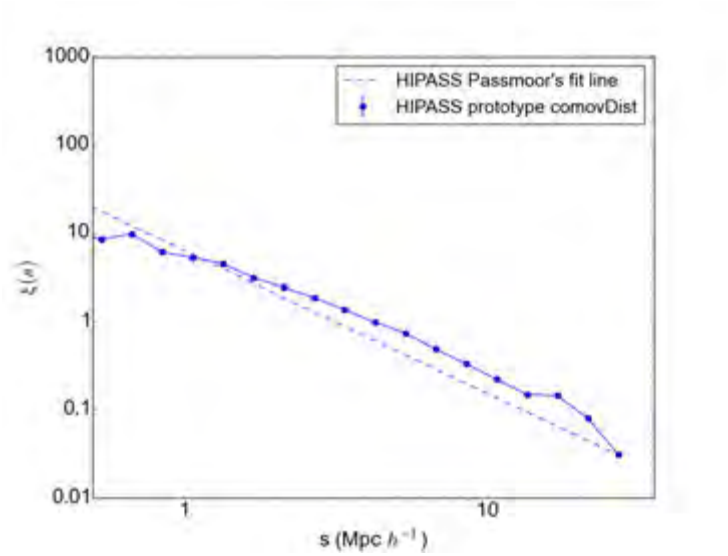


Figure 5.4: The spatial 3D correlation function for HIPASS, computed using the conceptual prototype script. The dashed blue line shows the projected power law fit using parameters in table 5.2.

5.2 Clustering in SDSS clusters

In this section we continued to test the prototype script using a samples of the GMBCG SDSS cluster catalog. This catalog consists of considerably more sources than HIPASS, across a wide redshift range of $0.1 < z < 0.6$. This offers a better sample for testing the spatial correlation function of the prototype solution and it is useful in our study of the clustering of Planck clusters. In section 5.2.1, we show the preprocessing of the data read from the catalog. Then in section 5.2.2 we present the results from computing the angular and the spatial 3D correlation functions.

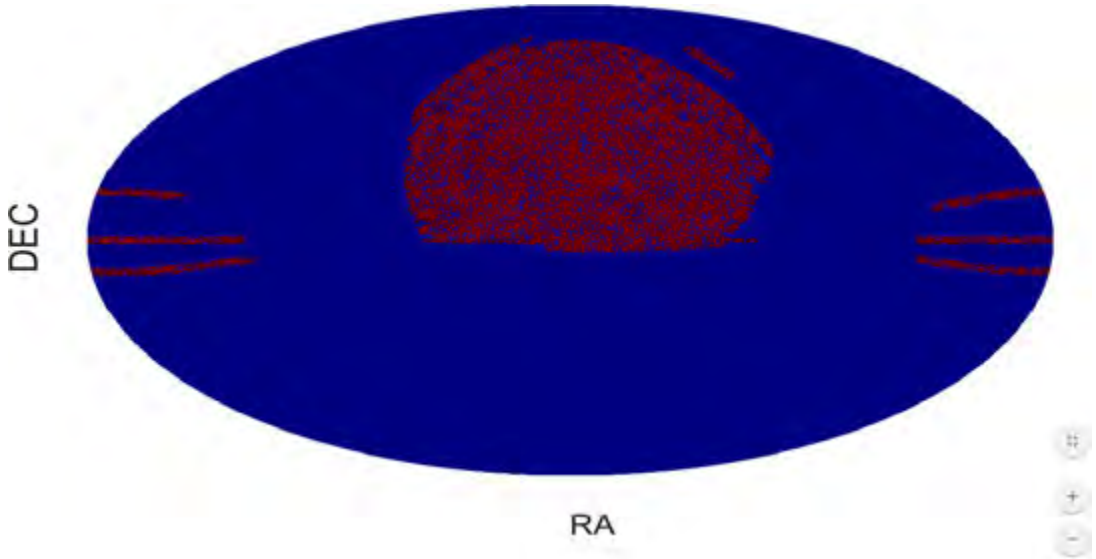
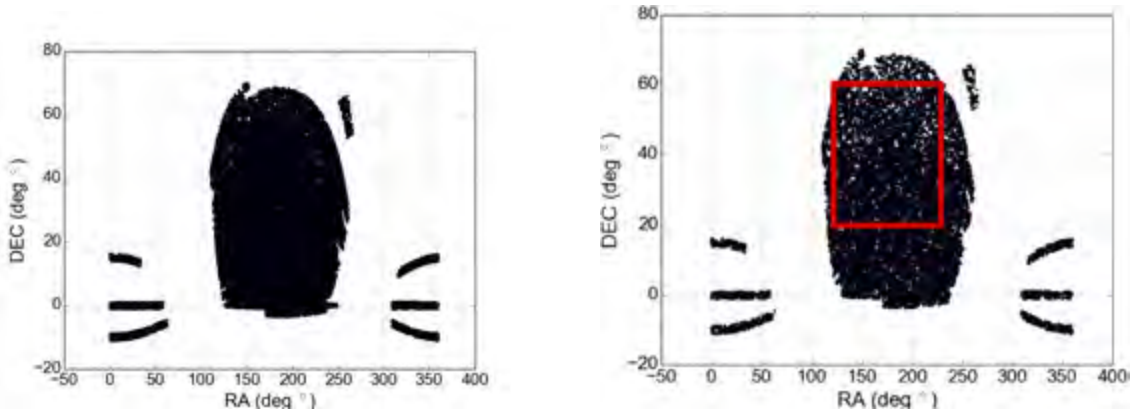


Figure 5.5: Sky coverage in the GMBCG public catalog based on SDSS DR7. Each point shows the position of one cluster on the sky. [14]

5.2.1 Extracting The Data

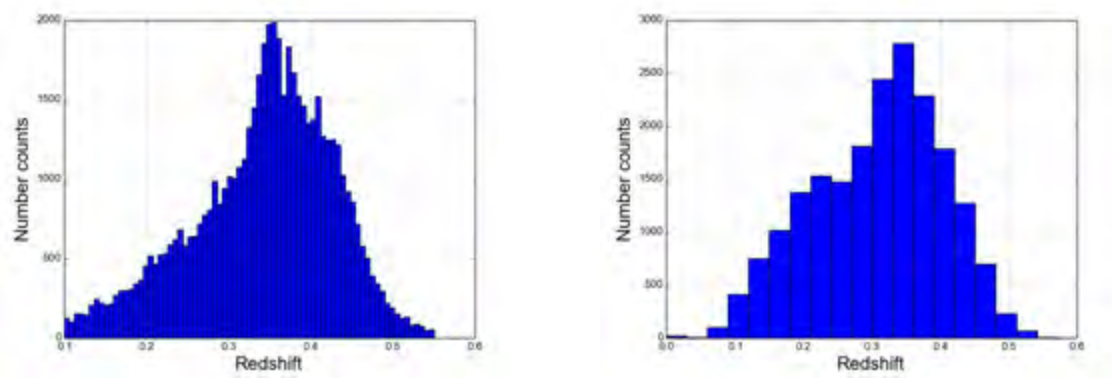
Figure 5.5 shows the sky coverage of the GMBCG public catalog that was published together with the catalog. In figure 5.6a we show the output of the data read during the preprocessing step from the same catalog and note that sky coverage is similar to that seen in the previous figure. Then, we filtered out clusters without a valid redshift data shown in figure 5.6b. Some of the clusters contained both spectroscopic and photometric redshifts, while some only have photometric data. Photometric redshifts are estimated using the magnitudes in a few photometric bands and have large uncertainties. Figure 5.7 shows the difference between the distribution of the two types redshifts presented in the catalog.

Due to the lack of a mask/selection function file for the SDSS catalogs, we only used



(a) SDSS Input Data (b) SDSS Filtered Z

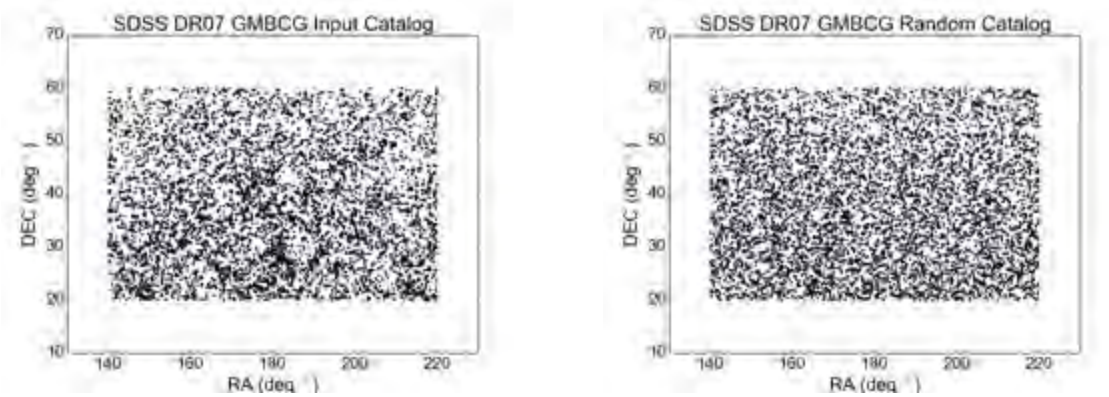
Figure 5.6: Input data read from GMBCG cluster catalog. Fig:5.6a shows all the clusters from the catalog and fig:5.6b shows only those with valid redshift information. The red rectangle shows the continious region used in our analysis.



(a) Photometric Redshift (b) Spectroscopic Redshift

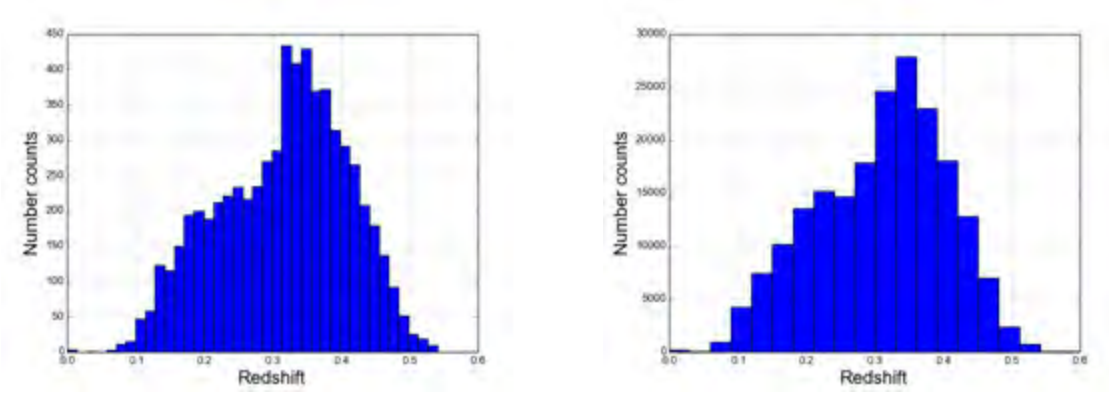
Figure 5.7: The photometric and spectroscopic redshift distribution of the SDSS clusters

a sample of clusters found within a continuous region as shown in figure 5.8a. Figure 5.8b shows the random catalog generated, consisting of sources reproduced within the same region of the selected data. The random catalog also needs to have a smoothed redshift distribution similar to the input catalog as shown figure 5.9. In order to minimize the error in estimating the the TPCF function, we generated the random particles with more sources than the input data. It is important to note that although the random catalog generated had more sources than the input catalog source, they still have the same redshift distribution.



(a) Input Filtered XY (b) Random Filtered XY

Figure 5.8: Selected continuous(red box fig5.6b) region of the SDSS clusters, to be used as the sample for computing the correlation function.



(a) Redshift Input XY (b) Redshift Random XY

Figure 5.9: Redshift distribution for the input(left) filtered data and the random(right) catalog. The random catalog contains 10 times the number of sources of the input data, but still same redshift distribution.

.

5.2.2 Results of SDSS clustering

Figure 5.10 shows the spatial 3D correlation functions of the SDSS clusters computed using the prototype script. The blue plot shows the 3D spatial correlation function computed using spectroscopic redshift and the cyan plot corresponds to the computation using the photometric redshifts of the clusters. A summary of the total number of objects from each sample is given in the table 5.3.

redshift	Num Clusters
Spectroscopic	6388
Photometric	17588

Table 5.3: Number of clusters from GMBCG catalog, used for computing the correlation function

From the plots we can observe a difference between the clustering signal measured using the spectroscopic redshift compared to the photometric sample. This is due the fact that photometric redshifts are estimates using 5 measurements across the spectrum. Particularly on small scales, distances are not captured accurately and clustering measurements need to be corrected for the lack of information, thus there is an expectation of error associated with photometric redshift. This is explained in detail by *Sereno .et al (2015)* in their paper [13]. We note that our results are similar to theirs.

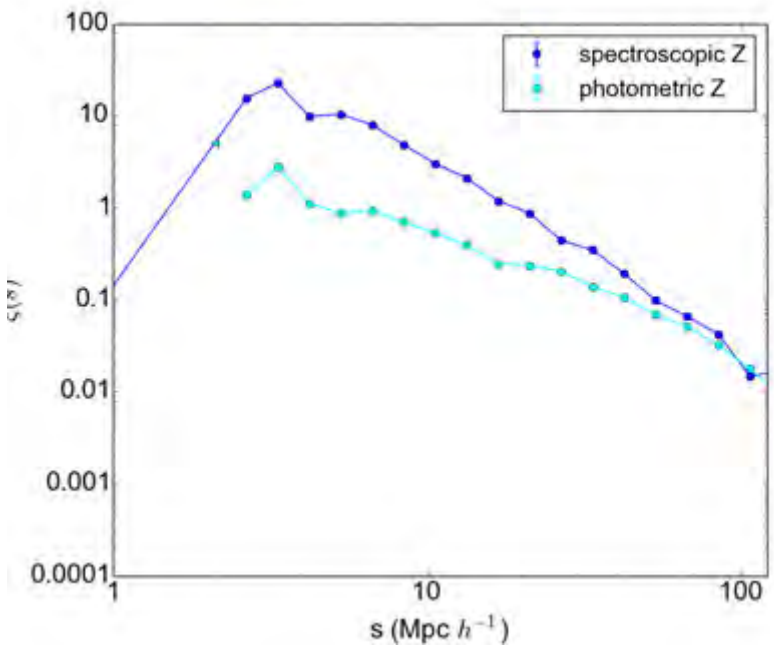


Figure 5.10: The 3D correlation function for SDSS GMBCG clusters. Comparing the clustering of samples of clusters based on their spectroscopic and photometric redshifts.

5.2.3 Estimating errors in TPCF

We investigated the inclusion of error bars on each point in the TPCF. We initially included Poissonian uncertainties and also explored the bootstrapping technique used in Passmoor et al. In Chapter 6 and 7 we move from the prototype script to the CUTE code which includes routines dealing with uncertainty estimates. We thus did not explore this further in the context of the prototype script.

Chapter 6

Experiment 2: Code Review and Performance Test

This chapter presents the investigation into the performance value offered by computing the TPCF on shared memory systems and GPU accelerators respectively. The main goal is to test how well these solutions scale with an increasing dataset. We used the CUTE code during this investigation phase, since it was publicly available in both OpemMP and CUDA. Thus, enabling us to test both platforms effectively. Firstly, section 6.1 provides some comparison between the results from the prototype scripts and a single thread OpenMP solution of CUTE code. We motivate for the use of CUTE.

In section 6.2 evaluate the performance of the original CUTE code against an increasing dataset and on different platforms. We test the code on two different datasets in terms of distribution density and sky coverage, while keeping the number of objects within each dataset the same. The results show a poor load balancing and scaling for one of the datasets, thus indicating a problem within the code. In section 6.3 we present the results from a modified version of CUTE, wherein we made slight changes to the CUTE boxing scheme. This offered a significant performance improvement relative to the original performance of the code.

6.1 Prototype Scripts vs CUTE

In the previous chapter we presented the results of measuring the 3D spatial correlation using the proposed prototype scripts. We compared results from our prototype scripts to those obtained using the CUTE code and obtained similar results, with some minor

differences. CUTE measures the 3-D correlation function as a function of r and μ described in Figure 2.16 and we needed to integrate over μ to compare directly with our code. Approximations in this integral lead to the minor differences but given the differences in execution time discussed later, we decided to concentrate on CUTE and not investigate the integration further.

Table 6.1 contains a list of execution times it takes to compute the correlation function for both the CUTE and prototype code on the datasets presented in section 5.2. The prototype solution was based on a simple serial approach, using python's NUMPY array broadcasting to compute the distance only once between every point in the dataset. This means given N_{in} & N_{rnd} as number of objects for the input and random catalog respectively, we compute the distance for only $N_{in}(N_{rnd}-1)/2$ objects for the autocorrelations (DD, RR) and then the cross-correlation (DR) between them is $N_{in} \times N_{rnd}$ objects. The CUTE results were taken from a serial implementation using a single thread execution. The boxing technique was used to avoid computing distances between objects beyond the region of interest.

As expected, in the case where the input and random catalogs have the same number objects, the computational time for autocorrelating (DD, RR) takes roughly the same time, while the cross-correlation time approximately takes twice as long. This similar for both the prototype solution and the CUTE code. However, in the case where the random catalog consists of ten times the input dataset, the prototype solution takes ~ 100 times longer to compute the autocorrelation of (RR) relative to that of (DD). This illustrate an N^2 computational complexity. Whereas, the CUTE code takes $\sim (60, 86)$ times longer for the $(63880, 175880)$ datasets respectively. This shows the improvement offered by the boxing technique in CUTE, although it is not constant on both datasets

Code	Dataset # objects	Correlating DD	RR	DR	$T(\xi(s))$ (seconds)
PyScripts	$N_{rnd} = N_{in} = 6388$	9	9	15	34
	$N_{rnd} = N_{in} \times 10 = 63880$	9	829	130	969
	$N_{rnd} = N_{in} = 17588$	56	59	91	206
	$N_{rnd} = N_{in} \times 10 = 175880$	64	6568	1292	7895
CUTE (1 thread)	$N_{rnd} = N_{in} = 6388$	0.039	0.038	0.076	0.156
	$N_{rnd} = N_{in} \times 10 = 63880$	0.039	2.356	0,333	2.731
	$N_{rnd} = N_{in} = 17588$	0.139	0.138	0.277	0.556
	$N_{rnd} = N_{in} \times 10 = 175880$	0.139	12.079	2.014	14.304

Table 6.1: Execution times for correlation computed using the python prototyping scripts compared to that using CUTE OpenMP code using 1 thread.

The overall execution time for the codes is shown in the last column of the table. The CUTE code on a single thread is much faster than the python prototype script, offering a speed up factor of $\sim$ 200 and $\sim$ 500 for the catalog datasets of 6388 and 17588 objects respectively. This is because the prototype scripts had only been developed as a conceptual tool for computing the 3D spatial correlation function and not fully optimised for performance. However, noticing the great speed up factor offered by a single thread OpenMP run of CUTE, we used this code to investigate further the research questions of this thesis for the remainder of our study.

6.2 CUTE Performance Evaluation Results

In this test the spatial 3D_rm (uses the r, μ convention) correlation function was calculated for catalogs of different sizes within the range $10^4 - 10^7$ number of objects. These catalogs were generated based on SDSS catalog with the same sky coverage and redshift distribution presented in figures 5.8 and 5.9 respectively. Figure 6.1 shows the computational times for a single auto-correlation(RR) of the random catalogs, taken from different platforms. The GPU implementation offers the best improvement, with a speed up factor of $\sim$ 40 relative to the single core implementation. The OpenMP version on a laptop with 4 cores achieved a speed up of 4, whereas speed up of $\sim$ 14 can be achieved on a single node server with 20 cores and a Dell c4130 node with 24 cores.

In order to test the code further, we generated a different dataset of catalogs based on the PlanckSZ catalog, while keeping the same number of objects. The sky coverage and redshift distribution for these datasets are shown in Figures 7.1 and (7.2a) respectively. Figure 6.2 presents a comparison of the computational times required to calculate the auto-correlation(RR) on the different dataset, for both OpenMP and GPU implementations. Although both datasets contain the same number of objects, there is a significant difference in the computational times required in both techniques.

Figure 6.3 shows the CPU utilisation graphs extracted using the Ganglia[55] monitoring tool from the compute nodes *(CUTE_OMP Research 20 cores)* when computing correlation functions for the results presented in Figure 6.2a. The gaps between the graphs represent the transition between datasets, starting from the the lowest to the highest number of objects. The SDSS(fig 6.3a) datasets indicate a well balanced CPU work load, with a relatively quick roll off time towards the end of the dataset. Therefore, very little time was spent with idle CPUs. However, for the Planck(6.3b) datasets the graph shows a poorly balanced load on the node. The code start off with work distributed across all the

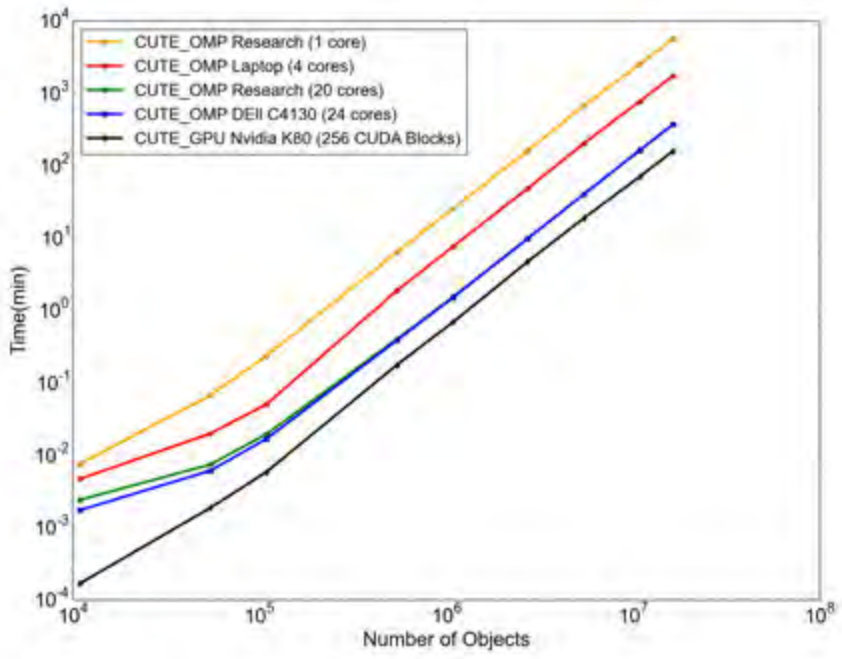
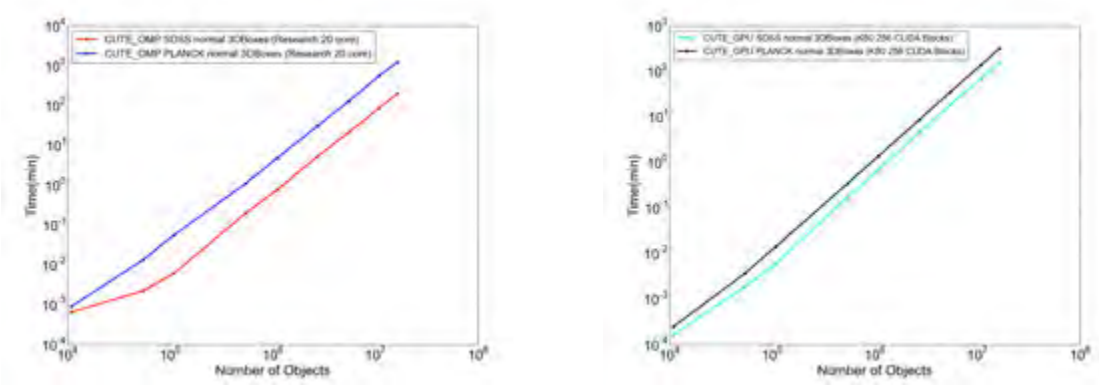


Figure 6.1: Computational times from different devices, calculating the spatial (3D_rm) auto-correlation (RR) of different sized catalogs. The GPU implementation show a substantial speed up factor of $\sim$ 40 relative to the sequential single core approach. While parallel implementations of the OpenMP on a Laptop and the servers(20_Cores, 24_Cores) show a speed up factor of $\sim (3.5 - 14)$ respectively.

CPUs, then drops of with gradually leaving most of time computational time spent with idle CPUs. This difference in notable in both the OpenMP and CUDA implementations of the code.

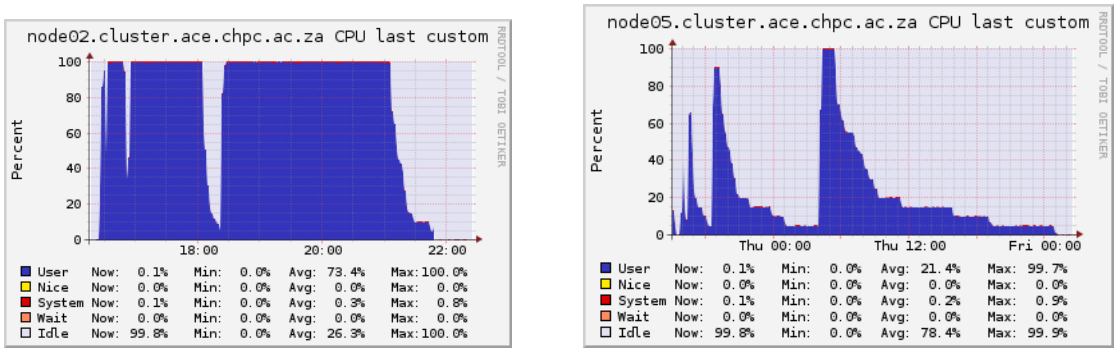
The Planck data is spread out over a much larger sky region than the SDSS data, so the results indicate that the computation in CUTE is largely dependent on the number density of the objects in the catalog. Therefore, this problem is likely due to the boxing technique employed by CUTE, resulting in some boxes with more densely packed objects than others. The parallelism scheme for CUTE involves sending boxes of data to multiple processors and as soon as a box is complete, a new box of data is sent to the processor. Thus if the boxes are big and inhomogeneously populated, this can result in a serious load balancing issue as seen with Planck dataset. In the next section, we present results of a modified version of the boxing scheme employed in CUTE.



(a) Scaling OMP SDSS vs Planck

(b) Scaling GPU SDSS vs Planck

Figure 6.2: Computational times of different sized catalogs, comparing datasets generated based on the SDSS and Planck catalogs respectively. Figure 6.2a presents the OpenMP version on a server node with 20 cores and figure 6.2b shows that GPU implementation.



(a) Node perfomance SDSS datasets

(b) Node perfomance Planck datasets

Figure 6.3: Ganglia report on the CPU utilisation extracted from the compute nodes when running spatial 3D_rm (expressed in terms of (r, μ)) correlation with CUTE, for the results shown in figure 6.2a. The SDSS datasets(fig 6.3a) show a well balanced load scaled across the CPUs, whereas the Planck(fig 6.3b) dataset indicate very poor load balancing on the node.

6.3 CUTE Modified 3D Boxing Results

Instead of generating the boxes based on the number density of the source, we modified the code to generate the boxing scheme based on the ratio between the maximum distance(R_max) of interest and the physical boundaries of the catalog. This effectively reduced the box sizes generated, allowing for optimal load balancing across the available CPUs. A summary of the actually number of objects, the computational times for each datasets and code implementation is given in table 6.2.

These results are also shown in figure 6.4, comparing the modified code against the original implementation. For relatively small numbers of objects ($\#obj < 5 \times 10^5$) the computational times are similar. However, as the datasets increase ($\#obj > 5 \times 10^5$), the plots indicate that the modified boxing scheme provides a significant performance improvement, achieving a speed up factor between $(2-4)$ and $(12-20)$ on SDSS and Planck datasets respectively.

Number of Objects	SDSS Dataset $T(\xi_{RR}(r,\mu))$		Planck Dataset $T(\xi_{RR}(r,\mu))$	
	normal 3DBoxes	modified 3DBoxes	normal 3DBoxes	modified 3DBoxes
10940	3.99E+01	3.61E+01	5.50E+01	8.31E+01
54700	1.35E+02	1.69E+02	8.15E+02	1.33E+02
109400	3.73E+02	4.19E+02	3.42E+03	3.51E+02
547000	1.22E+04	5.61E+03	6.76E+04	5.61E+03
1094000	4.59E+04	2.26E+04	2.88E+05	2.03E+04
2735000	3.22E+05	1.22E+05	1.87E+06	1.13E+05
5470000	1.28E+06	4.52E+05	7.75E+06	4.50E+05
10940000	5.32E+06	1.85E+06	3.42E+07	1.76E+06
16410000	1.22E+07	4.18E+06	7.49E+07	3.98E+06

Table 6.2: Execution times on different size catalogs, for the spatial 3D(r,μ), comparing performance of the normal and modified CUTE 3D boxes technique. The elapsed times were measured using the OpenMP timing functions. The times are in milliseconds (ms) and correspond to the auto-correlation of RR (for a full calculation of the TPCF it would take $(2-3)$ times longer.)

The yellow and green plots in figure 6.4, also shows that the computational time for the modified code is similar for both the different datasets. Figure 6.5 shows the ganglia report on the CPU utilisation when running the modified CUTE code for these datasets respectively. The graphs shows that the modified code offers much improved performance, with a better workload balancing across all CPUs. Unlike the previous report shown in Figure 6.3, there are no idle CPUs during the execution of the code. This results in the optimal utilisation of available resources, therefore improving the overall computing time for the CUTE code.

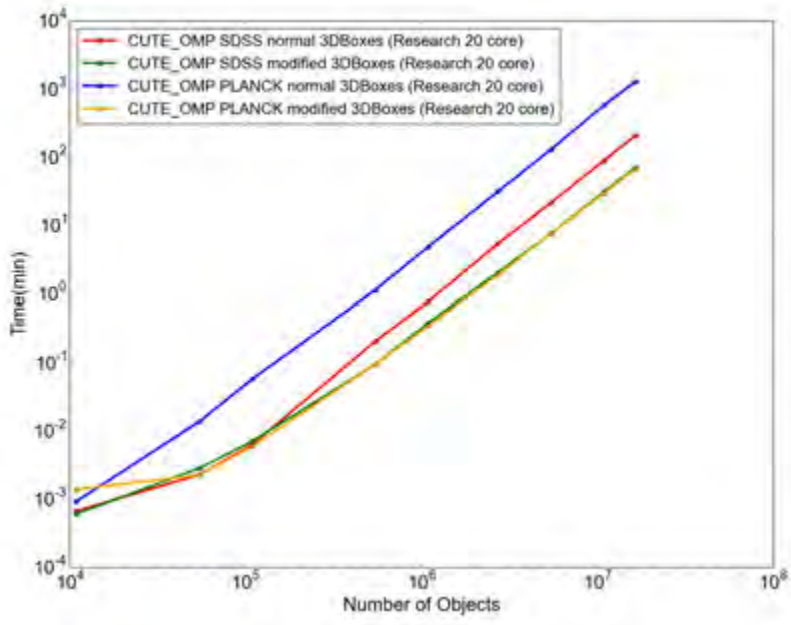
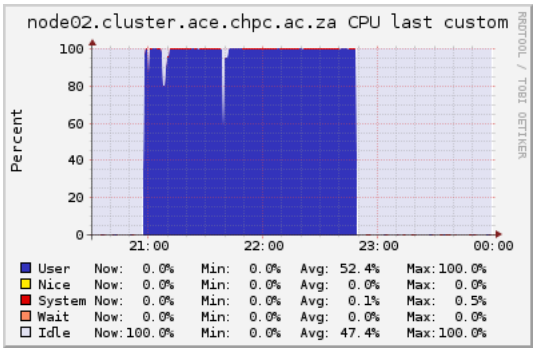
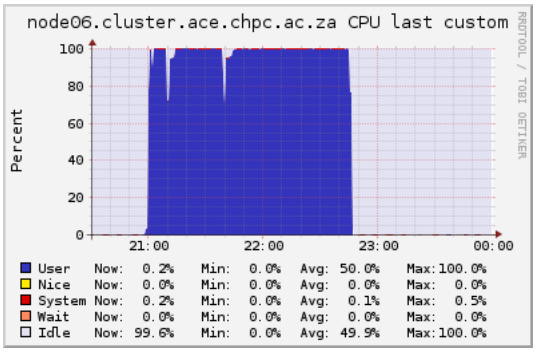


Figure 6.4: Computational times of different sized catalogs, comparing results from the modified boxing technique(modified 3DBoxes) of CUTE (OpenMP) against CUTE's original(normal 3DBoxes) version (similar to the ones presented in fig 6.2a).



(a) Node perfomance SDSS dataset



(b) Node perfomance Planck dataset

Figure 6.5: Ganglia report on the CPU utilisation from the compute nodes when running the correlation function with the modified CUTE boxing technique. The graphs shows a well balanced work load for both datasets, with no idle CPUs.

Optimally, the expected speed-up factor that can be achieved by increasing the number of cores would be linear. Thus, doubling the number of cores should halve the required runtime. Figure 6.6 shows a plot testing scalability of the code when increasing the number of cores for different sized catalogs. The plot also indicates the the speed up profile is consistent across the different catalogs, with an improved performance achieved by increasing the number of cores. The average speed up factor observed by doubling the number of processing elements between the different configurations is shown listed in table 6.3. The lowest average speed up achieved is 1.848, with an efficiency of 92.4% relative to the expected speed-up of 2.

Doubling # Cores	Average Speed up
$2-4$	1.990
$4-8$	1.828
$8-16$	1.880
$10-20$	1.936
$12-24$	1.996
$14-28$	1.848

Table 6.3: Average speed up factor achieved by doubling the number of cores.

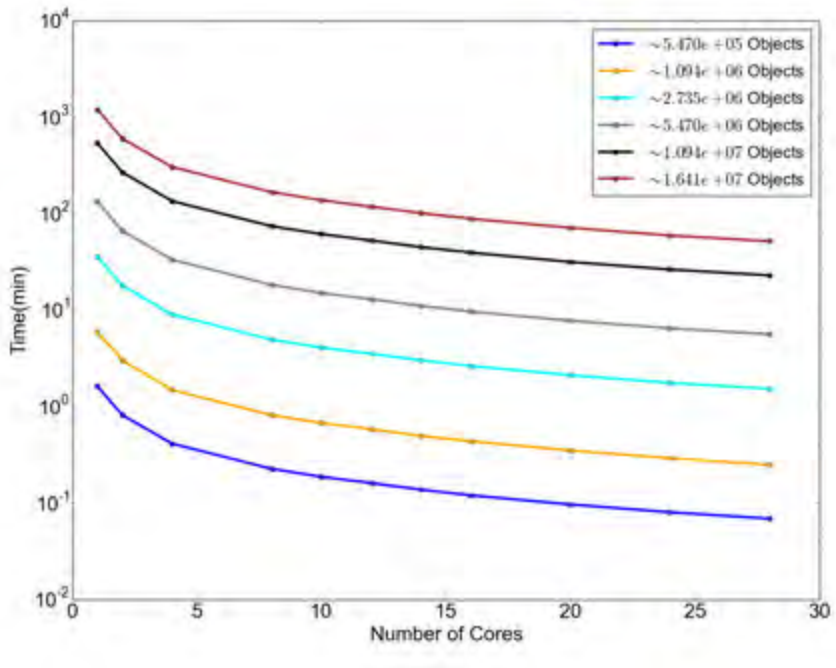


Figure 6.6: Computational times of different sized catalogs, comparing the scaling of the modified code with an increasing number of cores for the CUTE OpenMP (modified 3DBoxes) version.

Figure 6.7 shows the performance results of the modified boxing scheme implemented on the GPU platforms. The plot shows that for relatively small datasets which contain objects less than $\sim 1 \times 10^5$, the original version of the CUTE CODE is more efficient. However, this difference is insignificant considering that all the runtimes are still within 30 seconds. As sizes of datasets increases ($number\ of\ objects > 5 \times 10^5$), we can see a performance improvement with the modified boxing scheme, achieving a speed up factor between $(1.12 - 1.78)$ and $(2.13 - 10.37)$ for the SDSS and Planck datasets respectively.

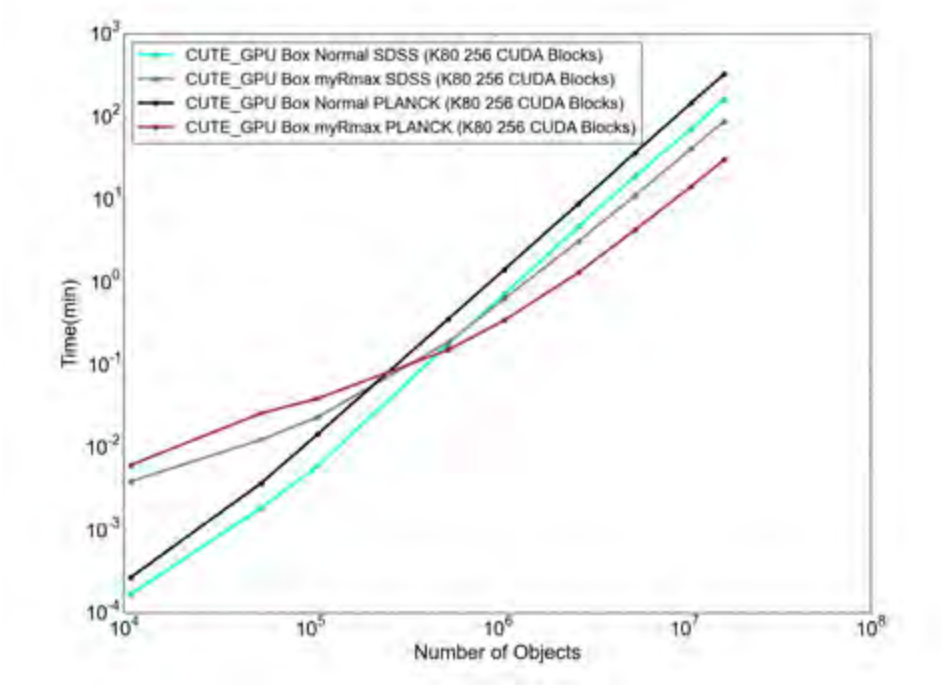


Figure 6.7: Computational times of different sized catalogs, comparing results from a modified boxing technique(myRmax) of CUTE (CUDA) against CUTE's original(Normal) version (similar to the ones presented in fig 6.2a).

6.4 Projections for larger catalogs

As mentioned in Chapter 1, the SKA could generate catalogs containing billions of objects. Figure 6.8 shows that more than 20 cores, or the current high end GPU accelerator would be required if clustering analysis is to be done in a reasonable amount of time for these large surveys. We note that estimating the uncertainties on the points in the TPCF is even more computationally intensive than computing the function itself, often requiring a factor of 10 more runtime. Also, higher order statistics such as three-point functions[86] are useful in cosmology and are even more computationally intensive. This reinforces the idea that clustering studies will need significant HPC resources in the future.

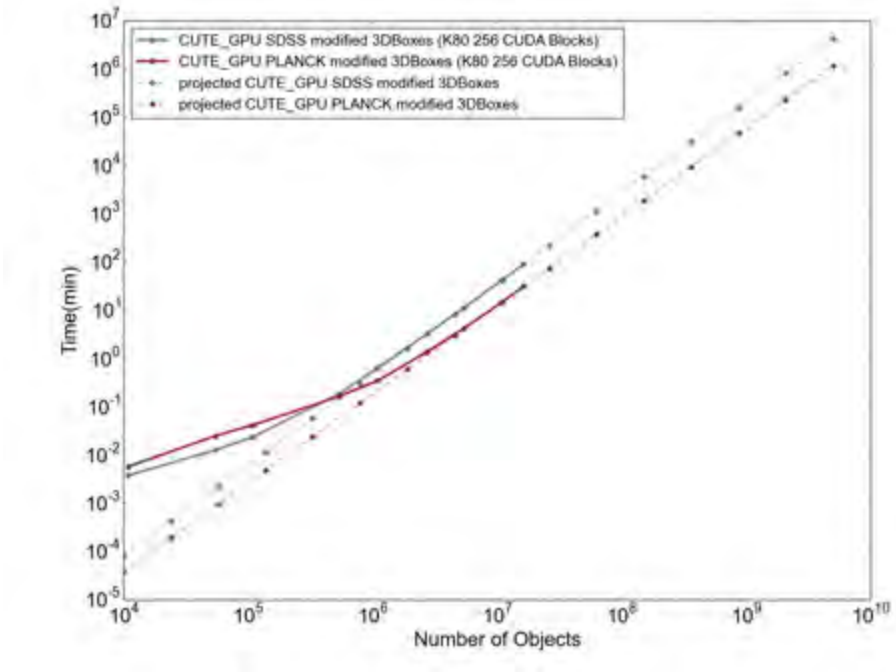


Figure 6.8: Projected times for larger datasets expected from future surveys such as the SKA

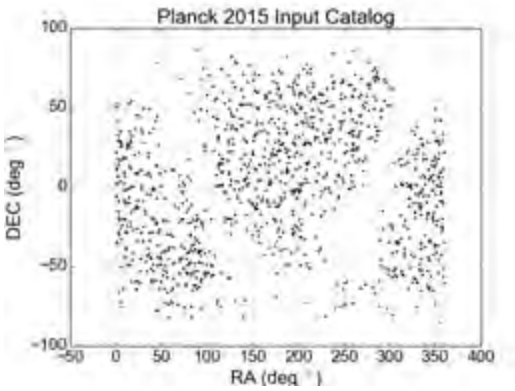
Chapter 7

Experiment 3: Code application in cosmology

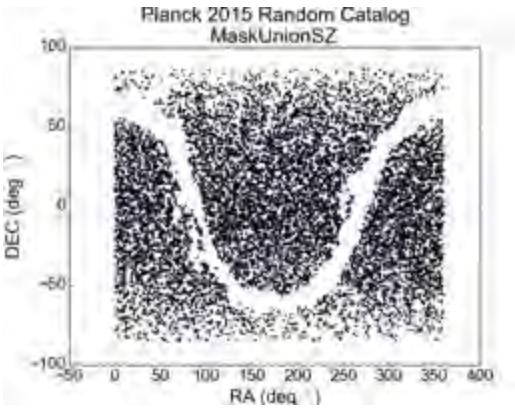
This chapter presents the practical use of code in a clustering study to address the significant cosmological question described in section 2.2. The study deals with using the clustering signature measured from the correlation functions as a probe of the mass of clusters, particularly those detected in Planck. In section 7.1 we present the data that we used in this study. This includes a selection of SDSS clusters from the cluster_-dr9sz catalog(see section 3.2.4) based on their richness, using the same range selection as that presented in [13]. Section 7.2 presents the results of the clustering as a function a mass/richness. Then in section 7.3 we present the results of probing the mass of the Planck clusters detected through their SZ signature.

7.1 Data

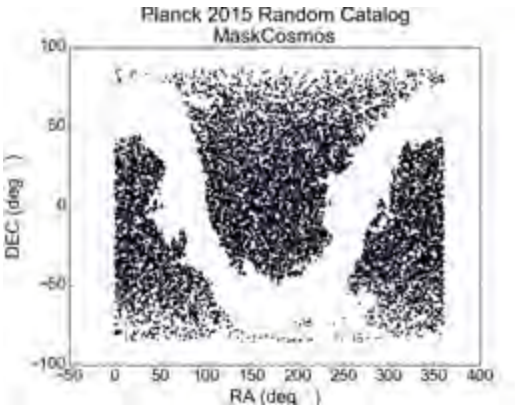
Figure 7.1 shows a plot of the data from the Planck's SZ 2015 union catalog of clusters, together with the random catalog data produced using the two masks provided. The catalog contained 1093 clusters with confirmed redshift information and the random catalogs were generated with 10 times the numbers of clusters in the input catalog. As seen the from the plots the Planck survey covers most of the sky. The masked out region represents the section of the sky in which the galactic plane of our galaxy is located, making it difficult to observe sources within this region. Also, the random catalogs generated using the mask files satisfy the constraints and boundaries observed in the input catalog.



(a) Input Data



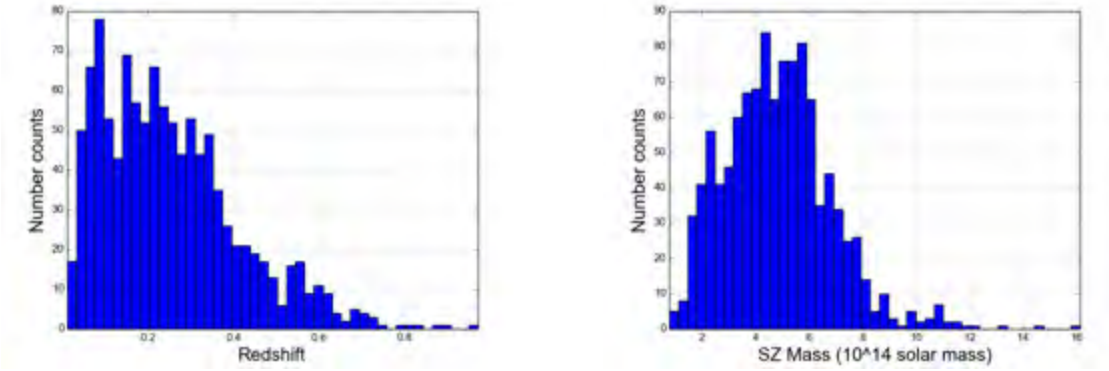
(b) Random Data Union Mask



(c) Random Data Cosmology Mask

Figure 7.1: Input data of 1093 clusters with redshift (7.1a) from Planck's 2015 SZ catalog, together with the two masked random data catalogs produced using the Union mask (7.1b) and the cosmology mask (7.1c containing 10 times the number of input sources)

The redshift distribution of these clusters is shown in figure 7.2a with an average redshift z of ~ 0.25. In figure 7.2b we plot the distribution of the measured SZ mass from the catalog. The average mass of these clusters based on their SZ signature is $\sim 4.8 \times 10^{14}$ solar masses. Figure 7.3 shows the richness(ngals) distribution of the SDSS clusters. The richness can be used as a proxy for the mass in the clusters. In our study we used it to select sub-samples of these clusters based on different richness bins to measure the clustering at these scales.



(a) Redshift Distribution (b) SZ Mass Distribution

Figure 7.2: The redshift and SZ Mass distribution of the clusters from Planck's 2015 SZ catalog

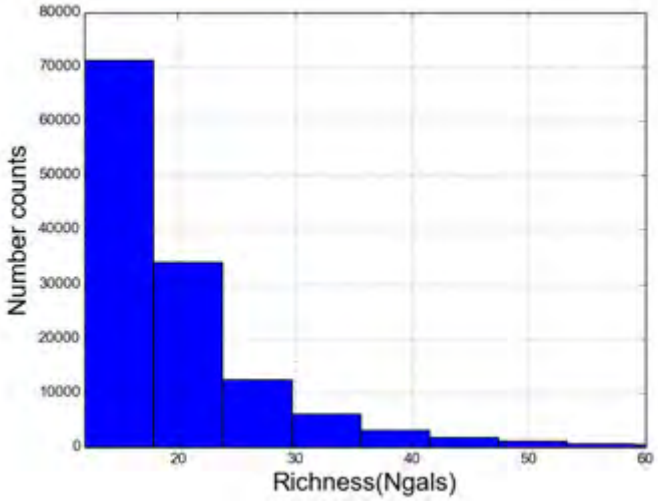


Figure 7.3: Richness Distribution in SDSS DR09 of clusters

7.2 Result: Clustering of with Richness

Figure 7.4 shows the results of clustering in four richness selected cluster samples from the SDSS survey, compared with those of the Planck SZ clusters(using the mask in Figure 7.2a). The blue, red, green and cyan plots refer to the increasing richness selection $12 < Ngals < 16$, $16 < Ngals < 21$, $21 < Ngals < 30$ and $Ngals > 30$, respectively. As it can be seen from the plot, the amplitude of the clustering increases with richness selection. This indicates that sources with a higher mass/richness have stronger clustering.

The magenta plot represents the clustering of the Planck SZ clusters. We can also observe that the Planck SZ clusters have a much higher clustering signature compared to the SDSS samples. This suggests that the mass of Planck clusters is more higher than those

detected in the $Ngals > 30$ bin. We compared the clustering of the Planck clusters from the cosmology sample with the full sample and the results where consistent with each other.

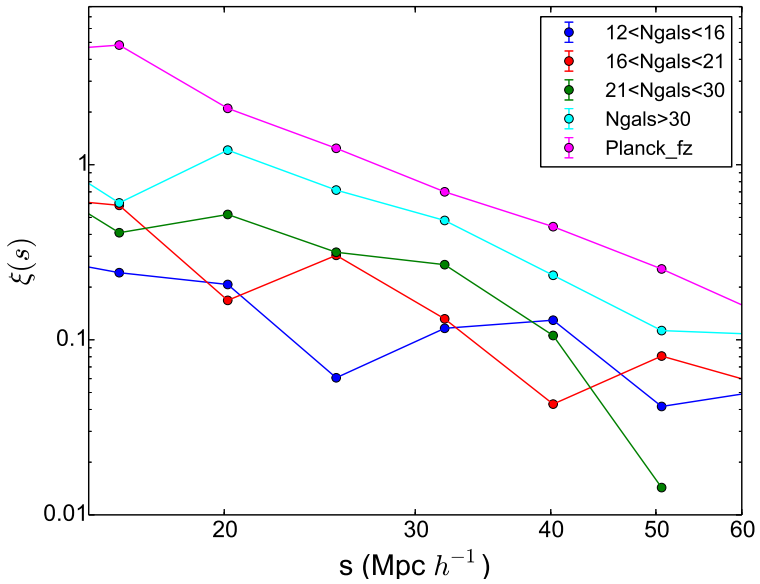


Figure 7.4: The 3D correlation function of 4 richness selected SDSS DR09 cluster samples, compared to the Planck SZ selected clusters.

7.3 Probing The Mass of SZ Clusters

In this section use the clustering signature of the Planck clusters to estimate average mass of the clusters. Firstly we need to calculate the clustering of Dark matter. Then we use that together with SZ clustering to get the bias, which is useful for estimating the mass of the clusters.

7.3.1 Clustering of Dark Matter

The power spectrum of dark matter is well understood within the ΛCDM model and can be easily generated using the correct parameter configurations. We used the LAMBDA-CAMB web interface tool and the Planck cosmological parameters provided in the paper [18] to generate the dark matter power spectrum. We then used the dark matter power spectrum data from LAMBDA-CAMB to calculate the spatial correlation function of Dark matter using equation 2.2 as described in section 2.3. A summary of the relevant cosmological parameters used is given in table 7.1.

Parameters		$TT, TE, EE + lowP$
Cosmological	$\Omega_b h^2$	0.02226 ± 0.00016
	$\Omega_c h^2$	0.1193 ± 0.0015
	$\Omega_v h^2$	0.00064 ± 0.0015
	Helium Fraction	0.24
	Redshift	0.25
Power Spectrum	Number	$1, 2.46e-9$
	Scalar Amplitude	0.96
Transfer Function	kmax	20

Table 7.1: LAMBDA-CAMB parameters used for generating the power spectrum of Dark matter. The cosmological parameters are based on Planck 2015 results XXIV [18].

Figure 7.5 shows the results of the spatial correlation function of the Dark matter compared to that of Planck's SZ clusters. The error bars for the Planck clustering were estimated by calculating the standard deviation of the TPCF applied on 10 different random datasets. The clustering signature of the Planck SZ clusters has a much higher amplitude compared to the clustering expected in dark matter. This offset in the clustering relative to that of dark matter can be described as a bias(b), which can then be used to estimate the mass of the clusters. This is discussed in the following section.

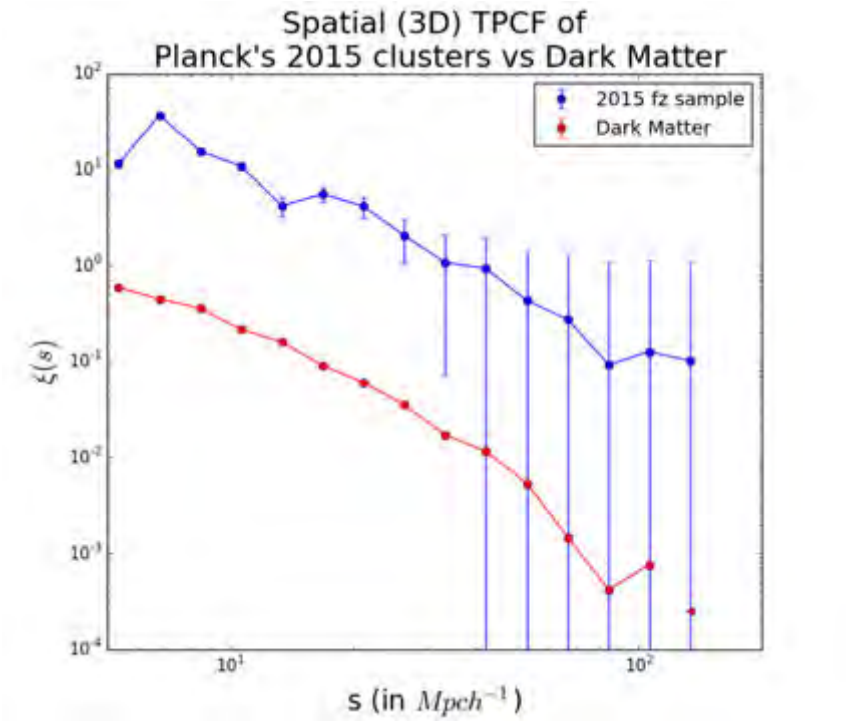


Figure 7.5: The correlation function of Planck's SZ clusters compared to the Dark matter function derive from the power spectrum. This is useful in estimating the bias(offset in clustering of luminous matter versus Dark Matter)

7.3.2 Bias Estimation and Mass Function

The bias (b_{DM}) describes the offset in the clustering observed in luminous matter(in this study, clusters of galaxies) compared to that expected in cold dark matter. The estimation for this bias within the ΛCDM model is defined by the following equation:

$$b_{DM} = (\xi_{lum}/\xi_{DM})^{1/2} \tag{7.1}$$

where ξ_{lum} and ξ_{DM} are 3D spatial correlation functions for the luminous matter and dark matter respectively. Here we assume scale independent bias.

Figure 7.6 shows the relation between the bias and the halo mass from the paper "Large-scale bias and stochasticity of haloes and dark matter" by *Seljak & Warren (2004)*[15]. This mass-bias relation was derived from several N-body simulations with $384^3 - 1024^3$ particles and box sizes of $96 - 1152h^1Mpc$. A fitting function that describes this halo bias−mass relationship is expressed as

$$b_0(x = M/M_{nl}) = 0.53 + 0.39x^{0.45} + \frac{0.13}{40x + 1} + 5 \times 10^{-4}x^{1.5} \tag{7.2}$$

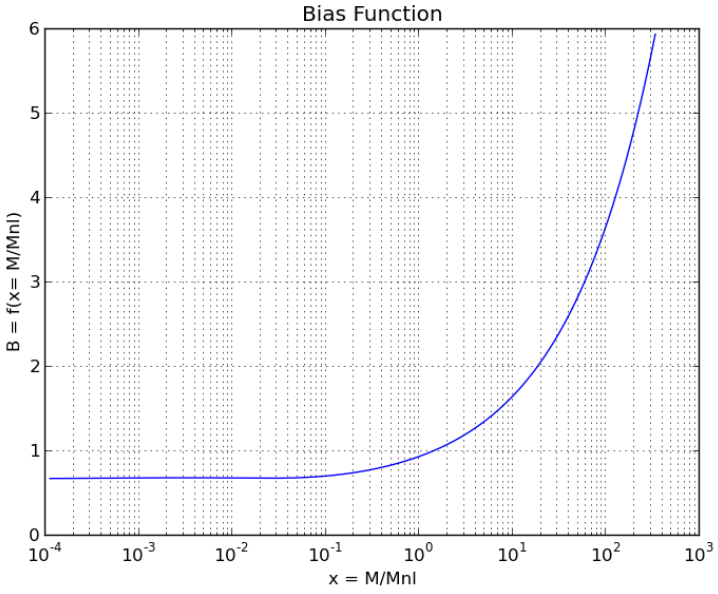


Figure 7.6: Fitting function from simulations by *Seljak & Warren (2004)*, relating the bias of the halos to the mass of the halos*(in units of the non-linear mass* $M_{nl} = 8.73 \times 10^{12}h^1M_\odot$*)* [15]

Applying equation 7.1 using the correlation functions presented in the previous section, we found that the average bias is $b_{DM} \sim 8$. From the mass-bias relation presented above, we find that a bias of 8 correspond to a mass of 2.5×10^{15} solar mass. Even when bias

evolution is considered (the clusters have a mean redshift of ~ 0.25 when bias would have been slightly higher than today), the mass determined from equation 7.2, indicates that masses are greater than 10^{15} solar masses. This is an unreasonably large mass, because other studies such as weak lensing in [42], [70][71] and X-ray observation in [42] do not give any indication that the mass should be this high.

This points to problems with the results from the simulations of [15] which were carried out over 10 years ago. Larger simulations are required to sample cluster halos effectively and this is now feasible with increased computing power available. In another project we are using 4Gpc boxes from the Magneticum simulations *(Dolag et al.)* to calibrate the bias-mass relation. Initial results from this work indicate that our measured bias corresponds to a mass of $\sim 6 \times 10^{14}$ solar masses. somewhat higher than the $\sim 4.8 \times 10^{14}$ solar masses obtained from the SZ signature. However, we still require further work to determine the significance of this measurement. Also, thorough testing of sample selection effects in the Planck cluster catalogs needs to be considered.

Chapter 8

Conclusions and Recommendations

8.1 Response to Research Questions

This section presents the conclusions that can be drawn from the results and discussions provided in chapters $5-7$. These chapters were structured and designed specifically to address the research questions posed in section 1.2.1. Therefore, the conclusions to be drawn will presented in a similar manner.

8.1.1 Question 1

Question: "What is the two-point correlation function (TPCF) and what are the current computational techniques?"

In chapter 5 we showed that the prototype script produced estimates of the TPCF which were consistent with previous studies: both for HIPASS and for the GMBCG cluster catalog from SDSS. This confirmed that the basic computational techniques for the TPCF had been understood. It also provided an opportunity to understand the implications of photometric versus spectroscopic redshift estimation in cluster catalogs.

8.1.2 Question 2

Question: "What methods/techniques are available to accelerate the computation of the TPFC and how well to these techniques scale with different sized datasets/catalogs?"

After testing both the OpenMP and CUDA versions of CUTE on different datasets and the available device platforms *(see section 6.2)*, the CUDA version was found to provide the best performance across the different datasets generated. However, when testing CUTE's 3D boxing scheme the code was found to be sufficiently lacking in terms of scalability of the workload balancing across the available resources.

After modifying the source code in CUTE for the 3D boxing scheme, a significant performance improvement was observed. The modified code provides a better distribution of the work load across the available CPU, offering substantial speed up factor relative to the original code. Although the CUDA version for the modified codes is slower for smaller datasets ($\#objects < 50^5$) relative to original code, it provides a sufficiently better performance for the datasets with more objects ($\#objects > 50^5$) Therefore, the CUDA version is the optimal method for computing on larger datasets. This implementation is still not sufficient for the number of objects expected from large surveys to be conducted on projects such a the SKA. However, these datasets will only be available after 2020 and presumably by then better GPU technology will be available then.

8.1.3 Question 3

Question: "How can we use the TPCF tools to answer some of the questions in cosmology? (Probing the mass of PLANCK galaxy clusters)"

We demonstrated that the TPCF implementation identified in chapter 6 could be used to study an interesting question in cosmology. Figure 7.4 shows galaxy clusters with higher richness or mass produce a stronger clustering signal, also revealing that PLANCK datasets consist of clusters with a higher mass relative to the SDSS datasets. There is some indication that the Planck clusters have higher masses than those inferred from the SZ-signature but the result is sensitive to the bias-mass relation used. We await a new calibration of this relation from a large enough simulation before making a quantitative statement about this.

8.2 Future Work

Based on the conclusions presented in section 8.1 the following areas have been identified for future work are made.

8.2.1 Further Development of TPCF solutions

As mentioned in section 1.1 the CHPC will host the MeerKAT data, providing tools to access and analyse the data. It would be useful to develop a user friendly interface that would allow the users easily perform clustering analysis of the sources in the various survey catalogs. The could be done by developing a wrapper package for the catalog preprocessing and GUI interface to use CUTE as the back end processing code to compute the TPCF.

The CHPC will soon make available a new cluster with over 24000 cores it could be interesting to explore scaling of the CUTE code on a much larger machine. This can be achieved by porting the code to support MPI parallelism. Considering the load balancing issues observed during the study, better algorithms for neighbour searching of objects in the catalog can be explored. The kd-tree[43] is one of the methods that offer some improvements for this code.

8.2.2 Probing the Mass of SZ Clusters

In this study, various issues have been identified and need to considered carefully going forward

The issues include:

- A better bias-halo mass relation needs to be determined using large enough simulations (at the mean redshift of the cluster sample so that bias evolution is less of an issue). This is still work progress, with preliminary findings showing some reasonable results. More thorough testing of sample selection effects in the Planck cluster catalogs should be carried out.
- More thorough testing of sample selection effects in the Planck cluster catalogs should be carried out.
- We should consider scale dependent bias and assembly bias
- Obviously, more quantitative estimation of the uncertainties on the bias and mass estimate will be required.

Bibliography

[1] http://map.gsfc.nasa.gov/media/060915/index.html.

[2] G. Hinshaw, M. R. Nolta, C. L. Bennett, R. Bean, O. Dore, M. R. Greason, M. Halpern, R. S. Hill, N. Jarosik, A. Kogut, E. Komatsu, M. Limon, N. Odegard, S. S. Meyer, L. Page, H. V. Peiris, D. N. Spergel, G. S. Tucker, L. Verde, J. L. Weiland, E. Wollack, and E. L. Wright. ThreeYear Wilkinson Microwave Anisotropy Probe (WMAP) Observations: Temperature Analysis. *Astrophys. J. Suppl. Ser.*, 170(2):288–334, jun 2007.

[3] P. A. R. Ade, N. Aghanim, M. I. R. Alves, C. Armitage-Caplan, M. Arnaud, M. Ashdown, F. Atrio-Barandela, J. Aumont, H. Aussel, C. Baccigalupi, A. J. Banday, R. B. Barreiro, R. Barrena, M. Bartelmann, J. G. Bartlett, N. Bartolo, S. Basak, E. Battaner, R. Battye, K. Benabed, A. Benoît, A. Benoit-Lévy, J.-P. Bernard, M. Bersanelli, B. Bertincourt, M. Bethermin, P. Bielewicz, I. Bikmaev, A. Blanchard, J. Bobin, J. J. Bock, H. Böhringer, A. Bonaldi, L. Bonavera, J. R. Bond, J. Borrill, F. R. Bouchet, F. Boulanger, H. Bourdin, J. W. Bowyer, M. Bridges, M. L. Brown, M. Bucher, R. Burenin, C. Burigana, R. C. Butler, E. Calabrese, B. Cappellini, J.-F. Cardoso, R. Carr, P. Carvalho, M. Casale, G. Castex, A. Catalano, A. Challinor, A. Chamballu, R.-R. Chary, X. Chen, H. C. Chiang, L.-Y Chiang, G. Chon, P. R. Christensen, E. Churazov, S. Church, M. Clemens, D. L. Clements, S. Colombi, L. P. L. Colombo, C. Combet, B. Comis, F. Couchot, A. Coulais, B. P. Crill, M. Cruz, A. Curto, F. Cuttaia, A. Da Silva, H. Dahle, L. Danese, R. D. Davies, R. J. Davis, P. de Bernardis, A. de Rosa, G. de Zotti, T. Déchelette, J. Delabrouille, J.-M. Delouis, J. Démoclès, F.-X. Désert, J. Dick, C. Dickinson, J. M. Diego, K. Dolag, H. Dole, S. Donzelli, O. Doré, M. Douspis, A. Ducout, J. Dunkley, X. Dupac, G. Efstathiou, F. Elsner, T. A. Enßlin, H. K. Eriksen, O. Fabre, E. Falgarone, M. C. Falvella, Y. Fantaye, J. Fergusson, C. Filliard, F. Finelli, I. Flores-Cacho, S. Foley, O. Forni, P. Fosalba, M. Frailis, A. A. Fraisse, E. Franceschi, M. Freschi, S. Fromenteau, M. Frommert, T. C. Gaier, S. Galeotta, J. Gallegos, S. Galli, B. Gandolfo, K. Ganga, C. Gauthier,

R. T. Génova-Santos, T. Ghosh, M. Giard, G. Giardino, M. Gilfanov, D. Girard, Y. Giraud-Héraud, E. Gjerløw, J. González-Nuevo, K. M. Górski, S. Gratton, A. Gregorio, A. Gruppuso, J. E. Gudmundsson, J. Haissinski, J. Hamann, F. K. Hansen, M. Hansen, D. Hanson, D. L. Harrison, A. Heavens, G. Helou, A. Hempel, S. Henrot-Versillé, C. Hernández-Monteagudo, D. Herranz, S. R. Hildebrandt, E. Hivon, S. Ho, M. Hobson, W. A. Holmes, A. Hornstrup, Z. Hou, W. Hovest, G. Huey, K. M. Huffenberger, G. Hurier, S. Ilić, A. H. Jaffe, T. R. Jaffe, J. Jasche, J. Jewell, W. C. Jones, M. Juvela, P. Kalberla, P. Kangaslahti, E. Keihänen, J. Kerp, R. Keskitalo, I. Khamitov, K. Kiiveri, J. Kim, T. S. Kisner, R. Kneissl, J. Knoche, L. Knox, M. Kunz, H. Kurki-Suonio, F. Lacasa, G. Lagache, A. Lähteenmäki, J.-M. Lamarre, M. Langer, A. Lasenby, M. Lattanzi, R. J. Laureijs, A. Lavabre, C. R. Lawrence, M. Le Jeune, S. Leach, J. P. Leahy, R. Leonardi, J. León-Tavares, C. Leroy, J. Lesgourgues, A. Lewis, C. Li, A. Liddle, M. Liguori, P. B. Lilje, M. Linden-Vørnle, V. Lindholm, M. López-Caniego, S. Lowe, P. M. Lubin, J. F. Macías-Pérez, C. J. MacTavish, B. Maffei, G. Maggio, D. Maino, N. Mandolesi, A. Mangilli, A. Marcos-Caballero, D. Marinucci, M. Maris, F. Marleau, D. J. Marshall, P. G. Martin, E. Martínez-González, S. Masi, M. Massardi, S. Matarrese, T. Matsumura, F. Matthai, L. Maurin, P. Mazzotta, A. McDonald, J. D. McEwen, P. McGehee, S. Mei, P. R. Meinhold, A. Melchiorri, J.-B. Melin, L. Mendes, E. Menegoni, A. Mennella, M. Migliaccio, K. Mikkelsen, M. Millea, R. Miniscalco, S. Mitra, M.-A. Miville-Deschênes, D. Molinari, A. Moneti, L. Montier, G. Morgante, N. Morisset, D. Mortlock, A. Moss, D. Munshi, J. A. Murphy, P. Naselsky, F. Nati, P. Natoli, M. Negrello, N. P. H. Nesvadba, C. B. Netterfield, H. U. Nørgaard-Nielsen, C. North, F. Noviello, D. Novikov, I. Novikov, I. J. O'Dwyer, F. Orieux, S. Osborne, C. O'Sullivan, C. A. Oxborrow, F. Paci, L. Pagano, F. Pajot, R. Paladini, S. Pandolfi, D. Paoletti, B. Partridge, F. Pasian, G. Patanchon, P. Paykari, D. Pearson, T. J. Pearson, M. Peel, H. V. Peiris, O. Perdereau, L. Perotto, F. Perrotta, V. Pettorino, F. Piacentini, M. Piat, E. Pierpaoli, D. Pietrobon, S. Plaszczynski, P. Platania, D. Pogosyan, E. Pointecouteau, G. Polenta, N. Ponthieu, L. Popa, T. Poutanen, G. W. Pratt, G. Prézeau, S. Prunet, J.-L. Puget, A. R. Pullen, J. P. Rachen, B. Racine, A. Rahlin, C. Räth, W. T. Reach, R. Rebolo, M. Reinecke, M. Remazeilles, C. Renault, A. Renzi, A. Riazuelo, S. Ricciardi, T. Riller, C. Ringeval, I. Ristorcelli, G. Robbers, G. Rocha, M. Roman, C. Rosset, M. Rossetti, G. Roudier, M. Rowan-Robinson, J. A. Rubiño-Martín, B. Ruiz-Granados, B. Rusholme, E. Salerno, M. Sandri, L. Sanselme, D. Santos, M. Savelainen, G. Savini, B. M. Schaefer, F. Schiavon, D. Scott, M. D. Seiffert, P. Serra, E. P. S. Shellard, K. Smith, G. F. Smoot, T. Souradeep, L. D. Spencer, J.-L. Starck, V. Stolyarov, R. Stompor, R. Sudiwala, R. Sunyaev, F. Sureau, P. Sutter, D. Sutton, A.-S. Suur-Uski, J.-F. Sygnet, J. A. Tauber, D. Tavagnacco, D. Taylor,

L. Terenzi, D. Texier, L. Toffolatti, M. Tomasi, J.-P. Torre, M. Tristram, M. Tucci, J. Tuovinen, M. Türler, M. Tuttlebee, G. Umana, L. Valenziano, J. Valiviita, B. Van Tent, J. Varis, L. Vibert, M. Viel, P. Vielva, F. Villa, N. Vittorio, L. A. Wade, B. D. Wandelt, C. Watson, R. Watson, I. K. Wehus, N. Welikala, J. Weller, M. White, S. D. M. White, A. Wilkinson, B. Winkel, J.-Q. Xia, D. Yvon, A. Zacchei, J. P. Zibin, and A. Zonca. Planck 2013 results. I. Overview of products and scientific results. *Astron. Astrophys.*, 571:A1, oct 2014.

[4] M. Seldner, B. Siebers, E. J. Groth, and P. J. E. Peebles. New reduction of the Lick catalog of galaxies. *Astron. J.*, 82:249, apr 1977.

[5] Matthew Colless, Gavin Dalton, Steve Maddox, Will Sutherland, Peder Norberg, Shaun Cole, Joss Bland-Hawthorn, Terry Bridges, Russell Cannon, Chris Collins, Warrick Couch, Nicholas Cross, Kathryn Deeley, Roberto De Propris, Simon P. Driver, George Efstathiou, Richard S. Ellis, Carlos S. Frenk, Karl Glazebrook, Carole Jackson, Ofer Lahav, Ian Lewis, Stuart Lumsden, Darren Madgwick, John A. Peacock, Bruce A. Peterson, Ian Price, Mark Seaborne, and Keith Taylor. The 2dF Galaxy Redshift Survey: spectra and redshifts. *Mon. Not. R. Astron. Soc.*, 328(4):1039–1063, dec 2001.

[6] Planck Collaboration, P. A. R. Ade, N. Aghanim, M. Arnaud, M. Ashdown, J. Aumont, C. Baccigalupi, A. J. Banday, R. B. Barreiro, J. G. Bartlett, N. Bartolo, E. Battaner, R. Battye, K. Benabed, A. Benoît, A. Benoit-Lévy, J. P. Bernard, M. Bersanelli, P. Bielewicz, A. Bonaldi, L. Bonavera, J. R. Bond, J. Borrill, F. R. Bouchet, M. Bucher, C. Burigana, R. C. Butler, E. Calabrese, J. F. Cardoso, A. Catalano, A. Challinor, A. Chamballu, R. R. Chary, H. C. Chiang, P. R. Christensen, S. Church, D. L. Clements, S. Colombi, L. P. L. Colombo, C. Combet, B. Comis, F. Couchot, A. Coulais, B. P. Crill, A. Curto, F. Cuttaia, L. Danese, R. D. Davies, R. J. Davis, P. de Bernardis, A. de Rosa, G. de Zotti, J. Delabrouille, F. X. Désert, J. M. Diego, K. Dolag, H. Dole, S. Donzelli, O. Doré, M. Douspis, A. Ducout, X. Dupac, G. Efstathiou, F. Elsner, T. A. Enßlin, H. K. Eriksen, E. Falgarone, J. Fergusson, F. Finelli, O. Forni, M. Frailis, A. A. Fraisse, E. Franceschi, A. Frejsel, S. Galeotta, S. Galli, K. Ganga, M. Giard, Y. Giraud-Héraud, E. Gjerløw, J. González-Nuevo, K. M. Górski, S. Gratton, A. Gregorio, A. Gruppuso, J. E. Gudmundsson, F. K. Hansen, D. Hanson, D. L. Harrison, S. Henrot-Versillé, C. Hernández-Monteagudo, D. Herranz, S. R. Hildebrandt, E. Hivon, M. Hobson, W. A. Holmes, A. Hornstrup, W. Hovest, K. M. Huffenberger, G. Hurier, A. H. Jaffe, T. R. Jaffe, W. C. Jones, M. Juvela, E. Keihänen, R. Keskitalo, T. S. Kisner, R. Kneissl, J. Knoche, M. Kunz, H. Kurki-Suonio, G. Lagache, A. Lähteenmäki, J. M. Lamarre, A. Lasenby, M. Lattanzi, C. R. Lawrence, R. Leonardi, J. Lesgourgues,

F. Levrier, M. Liguori, P. B. Lilje, M. Linden-Vørnle, M. López-Caniego, P. M. Lubin, J. F. Macías-Pérez, G. Maggio, D. Maino, N. Mandolesi, A. Mangilli, P. G. Martin, E. Martínez-González, S. Masi, S. Matarrese, P. Mazzotta, P. McGehee, P. R. Meinhold, A. Melchiorri, J. B. Melin, L. Mendes, A. Mennella, M. Migliaccio, S. Mitra, M. A. Miville-Deschênes, A. Moneti, L. Montier, G. Morgante, D. Mortlock, A. Moss, D. Munshi, J. A. Murphy, P. Naselsky, F. Nati, P. Natoli, C. B. Netterfield, H. U. Nørgaard-Nielsen, F. Noviello, D. Novikov, I. Novikov, C. A. Oxborrow, F. Paci, L. Pagano, F. Pajot, D. Paoletti, B. Partridge, F. Pasian, G. Patanchon, T. J. Pearson, O. Perdereau, L. Perotto, F. Perrotta, V. Pettorino, F. Piacentini, M. Piat, E. Pierpaoli, D. Pietrobon, S. Plaszczynski, E. Pointecouteau, G. Polenta, L. Popa, G. W. Pratt, G. Prézeau, S. Prunet, J. L. Puget, J. P. Rachen, R. Rebolo, M. Reinecke, M. Remazeilles, C. Renault, A. Renzi, I. Ristorcelli, G. Rocha, M. Roman, C. Rosset, M. Rossetti, G. Roudier, J. A. Rubiño-Martín, B. Rusholme, M. Sandri, D. Santos, M. Savelainen, G. Savini, D. Scott, M. D. Seiffert, E. P. S. Shellard, L. D. Spencer, V. Stolyarov, R. Stompor, R. Sudiwala, R. Sunyaev, D. Sutton, A. S. Suur-Uski, J. F. Sygnet, J. A. Tauber, L. Terenzi, L. Toffolatti, M. Tomasi, M. Tristram, M. Tucci, J. Tuovinen, M. Türler, G. Umana, L. Valenziano, J. Valiviita, B. Van Tent, P. Vielva, F. Villa, L. A. Wade, B. D. Wandelt, I. K. Wehus, J. Weller, S. D. M. White, D. Yvon, A. Zacchei, and A. Zonca. Planck 2015 results. XXIV. Cosmology from Sunyaev-Zeldovich cluster counts. page 17, feb 2015.

[7] Anja von der Linden, Adam Mantz, Steven W. Allen, Douglas E. Applegate, Patrick L Kelly, R Glenn Morris, Adam Wright, Mark T Allen, Patricia R Burchat, David L Burke, David Donovan, and Harald Ebeling. Robust Weak-lensing Mass Calibration of Planck Galaxy Clusters. 5(February):5, 2014.

[8] C. M. Baugh, D. J. Croton, E. Gaztanaga, P. Norberg, M. Colless, I. K. Baldry, J. Bland-Hawthorn, T. Bridges, R. Cannon, S. Cole, C. Collins, W. Couch, G. Dalton, R. De Propris, S. P. Driver, G. Efstathiou, R. S. Ellis, C. S. Frenk, K. Glazebrook, C. Jackson, O. Lahav, I. Lewis, S. Lumsden, S. Maddox, D. Madgwick, J. A. Peacock, B. A. Peterson, W. Sutherland, and K. Taylor. The 2dF Galaxy Redshift Survey: hierarchical galaxy clustering. *Mon. Not. R. Astron. Soc.*, 351(2):L44–L48, jun 2004.

[9] Alison L. Coil. *Planets, Stars and Stellar Systems*. Springer Netherlands, Dordrecht, feb 2013.

[10] Barbara Chapman, Gabriele Jost, and Ruud van der Pas. *Using OpenMP: Portable Shared Memory Parallel Programming, Volume 10*. MIT Press, 2008.

[11] David Alonso. CUTE solutions for two-point correlation functions from large cosmological datasets. *arXiv Prepr. arXiv1210.1833*, pages 1–9, 2012.

[12] Rafael Ponce, Miguel Cardenas-Montes, Juan Jose Rodriguez-Vazquez, Eusebio Sanchez, and Ignacio Sevilla. Application of GPUs for the Calculation of Two Point Correlation Functions in Cosmology. apr 2012.

[13] M. Sereno, A. Veropalumbo, F. Marulli, G. Covone, L. Moscardini, and A. Cimatti. New constraints on 8 from a joint analysis of stacked gravitational lensing and clustering of galaxy clusters. *Mon. Not. R. Astron. Soc.*, 449(4):4147–4161, apr 2015.

[14] Jiangang Hao, Timothy A. McKay, Benjamin P. Koester, Eli S. Rykoff, Eduardo Rozo, James Annis, Risa H. Wechsler, August Evrard, Seth R. Siegel, Matthew Becker, Michael Busha, David Gerdes, David E. Johnston, and Erin Sheldon. A GMBCG GALAXY CLUSTER CATALOG OF 55,424 RICH CLUSTERS FROM SDSS DR7. *Astrophys. J. Suppl. Ser.*, 191(2):254–274, dec 2010.

[15] Uroš Seljak and Michael S. Warren. Large-scale bias and stochasticity of haloes and dark matter. *Mon. Not. R. Astron. Soc.*, 355(1):129–136, nov 2004.

[16] Michał J. Chodorowski. Is Space Really Expanding? A Counterexample. *Old New Concepts Phys.*, 4(1):15–33, mar 2007.

[17] S. S. Passmoor, C. M. Cress, and A. Faltenbacher. Clustering of HI galaxies in HIPASS and ALFALFA. page 5, jan 2011.

[18] Planck Collaboration, P. A. R. Ade, N. Aghanim, M. Arnaud, M. Ashdown, J. Aumont, C. Baccigalupi, A. J. Banday, R. B. Barreiro, J. G. Bartlett, N. Bartolo, E. Battaner, R. Battye, K. Benabed, A. Benoit, A. Benoit-Levy, J. P. Bernard, M. Bersanelli, P. Bielewicz, A. Bonaldi, L. Bonavera, J. R. Bond, J. Borrill, F. R. Bouchet, F. Boulanger, M. Bucher, C. Burigana, R. C. Butler, E. Calabrese, J. F. Cardoso, A. Catalano, A. Challinor, A. Chamballu, R. R. Chary, H. C. Chiang, J. Chluba, P. R. Christensen, S. Church, D. L. Clements, S. Colombi, L. P. L. Colombo, C. Combet, A. Coulais, B. P. Crill, A. Curto, F. Cuttaia, L. Danese, R. D. Davies, R. J. Davis, P. de Bernardis, A. de Rosa, G. de Zotti, J. Delabrouille, F. X. Desert, E. Di Valentino, C. Dickinson, J. M. Diego, K. Dolag, H. Dole, S. Donzelli, O. Dore, M. Douspis, A. Ducout, J. Dunkley, X. Dupac, G. Efstathiou, F. Elsner, T. A. Ensslin, H. K. Eriksen, M. Farhang, J. Fergusson, F. Finelli, O. Forni, M. Frailis, A. A. Fraisse, E. Franceschi, A. Frejsel, S. Galeotta, S. Galli, K. Ganga, C. Gauthier, M. Gerbino, T. Ghosh, M. Giard, Y. Giraud-Heraud, E. Giusarma, E. Gjerlow, J. Gonzalez-Nuevo, K. M. Gorski, S. Gratton, A. Gregorio, A. Gruppuso, J. E. Gudmundsson, J. Hamann, F. K. Hansen, D. Hanson, D. L. Harrison, G. Helou, S. Henrot-Versille, C. Hernandez-Monteagudo, D. Herranz, S. R. Hildebrandt, E. Hivon, M. Hobson, W. A. Holmes, A. Hornstrup, W. Hovest,

Z. Huang, K. M. Huffenberger, G. Hurier, A. H. Jaffe, T. R. Jaffe, W. C. Jones, M. Juvela, E. Keihanen, R. Keskitalo, T. S. Kisner, R. Kneissl, J. Knoche, L. Knox, M. Kunz, H. Kurki-Suonio, G. Lagache, A. Lahteenmaki, J. M. Lamarre, A. Lasenby, M. Lattanzi, C. R. Lawrence, J. P. Leahy, R. Leonardi, J. Lesgourgues, F. Levrier, A. Lewis, M. Liguori, P. B. Lilje, M. Linden-Vornle, M. Lopez-Caniego, P. M. Lubin, J. F. Macias-Perez, G. Maggio, D. Maino, N. Mandolesi, A. Mangilli, A. Marchini, P. G. Martin, M. Martinelli, E. Martinez-Gonzalez, S. Masi, S. Matarrese, P. Mazzotta, P. McGehee, P. R. Meinhold, A. Melchiorri, J. B. Melin, L. Mendes, A. Mennella, M. Migliaccio, M. Millea, S. Mitra, M. A. Miville-Deschenes, A. Moneti, L. Montier, G. Morgante, D. Mortlock, A. Moss, D. Munshi, J. A. Murphy, P. Naselsky, F. Nati, P. Natoli, C. B. Netterfield, H. U. Norgaard-Nielsen, F. Noviello, D. Novikov, I. Novikov, C. A. Oxborrow, F. Paci, L. Pagano, F. Pajot, R. Paladini, D. Paoletti, B. Partridge, F. Pasian, G. Patanchon, T. J. Pearson, O. Perdereau, L. Perotto, F. Perrotta, V. Pettorino, F. Piacentini, M. Piat, E. Pierpaoli, D. Pietrobon, S. Plaszczynski, E. Pointecouteau, G. Polenta, L. Popa, G. W. Pratt, G. Prezeau, S. Prunet, J. L. Puget, J. P. Rachen, W. T. Reach, R. Rebolo, M. Reinecke, M. Remazeilles, C. Renault, A. Renzi, I. Ristorcelli, G. Rocha, C. Rosset, M. Rossetti, G. Roudier, B. Rouille D'Orfeuil, M. Rowan-Robinson, J. A. Rubino-Martin, B. Rusholme, N. Said, V. Salvatelli, L. Salvati, M. Sandri, D. Santos, M. Savelainen, G. Savini, D. Scott, M. D. Seiffert, P. Serra, E. P. S. Shellard, L. D. Spencer, M. Spinelli, V. Stolyarov, R. Stompor, R. Sudiwala, R. Sunyaev, D. Sutton, A. S. Suur-Uski, J. F. Sygnet, J. A. Tauber, L. Terenzi, L. Toffolatti, M. Tomasi, M. Tristram, T. Trombetti, M. Tucci, J. Tuovinen, M. Turler, G. Umana, L. Valenziano, J. Valiviita, B. Van Tent, P. Vielva, F. Villa, L. A. Wade, B. D. Wandelt, I. K. Wehus, M. White, S. D. M. White, A. Wilkinson, D. Yvon, A. Zacchei, and A. Zonca. Planck 2015 results. XIII. Cosmological parameters. feb 2015.

[19] George Francis Rayner Ellis. On the philosophy of cosmology. *Stud. Hist. Philos. Sci. Part B - Stud. Hist. Philos. Mod. Phys.*, 46(1):5–23, feb 2014.

[20] 2015 Catalogues - Planck PLA 2015 Wiki.

[21] Eric Gawiser and Joseph Silk. The Cosmic Microwave Background Radiation, 2000.

[22] Z. L. Wen, J. L. Han, and F. S. Liu. A CATALOG OF 132,684 CLUSTERS OF GALAXIES IDENTIFIED FROM SLOAN DIGITAL SKY SURVEY III. *Astrophys. J. Suppl. Ser.*, 199(2):34, apr 2012.

[23] Gregory R. Andrews. *Foundations of Multithreaded, Parallel, and Distributed Programming.* Addison-Wesley, 2000.

[24] Jatin Chhugani, Changkyu Kim, Hemant Shukla, Jongsoo Park, Pradeep Dubey, John Shalf, and Horst D. Simon. Billion-particle SIMD-friendly two-point correlation on large-scale HPC cluster systems. In *2012 Int. Conf. High Perform. Comput. Networking, Storage Anal.*, pages 1–11. IEEE, nov 2012.

[25] P. Kenneth Seidelmann, United States Naval Observatory. Nautical Almanac Office, and Great Britain. Nautical Almanac Office. *Explanatory Supplement to the Astronomical Almanac*. University Science Books, 2005.

[26] Stephen D. Landy and Alexander S. Szalay. Bias and variance of angular correlation functions. *Astrophys. J.*, 412:64, jul 1993.

[27] R. Laureijs, J. Amiaux, S. Arduini, J. L. Augueres, J. Brinchmann, R. Cole, M. Cropper, C. Dabin, L. Duvet, A. Ealet, B. Garilli, P. Gondoin, L. Guzzo, J. Hoar, H. Hoekstra, R. Holmes, T. Kitching, T. Maciaszek, Y. Mellier, F. Pasian, W. Percival, J. Rhodes, G. Saavedra Criado, M. Sauvage, R. Scaramella, L. Valenziano, S. Warren, R. Bender, F. Castander, A. Cimatti, O. Le Fèvre, H. Kurki-Suonio, M. Levi, P. Lilje, G. Meylan, R. Nichol, K. Pedersen, V. Popa, R. Rebolo Lopez, H. W. Rix, H. Rottgering, W. Zeilinger, F. Grupp, P. Hudelot, R. Massey, M. Meneghetti, L. Miller, S. Paltani, S. Paulin-Henriksson, S. Pires, C. Saxton, T. Schrabback, G. Seidel, J. Walsh, N. Aghanim, L. Amendola, J. Bartlett, C. Baccigalupi, J. P. Beaulieu, K. Benabed, J. G. Cuby, D. Elbaz, P. Fosalba, G. Gavazzi, A. Helmi, I. Hook, M. Irwin, J. P. Kneib, M. Kunz, F. Mannucci, L. Moscardini, C. Tao, R. Teyssier, J. Weller, G. Zamorani, M. R. Zapatero Osorio, O. Boulade, J. J. Foumond, A. Di Giorgio, P. Guttridge, A. James, M. Kemp, J. Martignac, A. Spencer, D. Walton, T. Blümchen, C. Bonoli, F. Bortoletto, C. Cerna, L. Corcione, C. Fabron, K. Jahnke, S. Ligori, F. Madrid, L. Martin, G. Morgante, T. Pamplona, E. Prieto, M. Riva, R. Toledo, M. Trifoglio, F. Zerbi, F. Abdalla, M. Douspis, C. Grenet, S. Borgani, R. Bouwens, F. Courbin, J. M. Delouis, P. Dubath, A. Fontana, M. Frailis, A. Grazian, J. Koppenhöfer, O. Mansutti, M. Melchior, M. Mignoli, J. Mohr, C. Neissner, K. Noddle, M. Poncet, M. Scodeggio, S. Serrano, N. Shane, J. L. Starck, C. Surace, A. Taylor, G. Verdoes-Kleijn, C. Vuerli, O. R. Williams, A. Zacchei, B. Altieri, I. Escudero Sanz, R. Kohley, T. Oosterbroek, P. Astier, D. Bacon, S. Bardelli, C. Baugh, F. Bellagamba, C. Benoist, D. Bianchi, A. Biviano, E. Branchini, C. Carbone, V. Cardone, D. Clements, S. Colombi, C. Conselice, G. Cresci, N. Deacon, J. Dunlop, C. Fedeli, F. Fontanot, P. Franzetti, C. Giocoli, J. Garcia-Bellido, J. Gow, A. Heavens, P. Hewett, C. Heymans, A. Holland, Z. Huang, O. Ilbert, B. Joachimi, E. Jennins, E. Kerins, A. Kiessling, D. Kirk, R. Kotak, O. Krause, O. Lahav, F. van Leeuwen, J. Lesgourgues, M. Lombardi, M. Magliocchetti, K. Maguire, E. Majerotto, R. Maoli,

F. Marulli, S. Maurogordato, H. McCracken, R. McLure, A. Melchiorri, A. Merson, M. Moresco, M. Nonino, P. Norberg, J. Peacock, R. Pello, M. Penny, V. Pettorino, C. Di Porto, L. Pozzetti, C. Quercellini, M. Radovich, A. Rassat, N. Roche, S. Ronayette, E. Rossetti, B. Sartoris, P. Schneider, E. Semboloni, S. Serjeant, F. Simpson, C. Skordis, G. Smadja, S. Smartt, P. Spano, S. Spiro, M. Sullivan, A. Tilquin, R. Trotta, L. Verde, Y. Wang, G. Williger, G. Zhao, J. Zoubian, and E. Zucca. Euclid Definition Study Report. page 116, oct 2011.

[28] Python Software Foundation. About Python — Python.org, 2015.

[29] P. J. E. Peebles. *Principles of Physical Cosmology*. Princeton University Press, 1993.

[30] NVIDIA. Nvidia Cuda Getting Started Guide For Linux, 2014.

[31] S. A. Gregory and L. A. Thompson. The Coma/A1367 supercluster and its environs. *Astrophys. J.*, 222:784, jun 1978.

[32] Tamara M. Davis and Charles H. Lineweaver. Expanding confusion: Common misconceptions of cosmological horizons and the superluminal expansion of the universe. *Publ. Astron. Soc. Aust.*, 21(1):97–109, mar 2004.

[33] Tamara M. Davis and Charles H. Lineweaver. Expanding confusion: Common misconceptions of cosmological horizons and the superluminal expansion of the universe. *Publ. Astron. Soc. Aust.*, 21(1):97–109, mar 2004.

[34] Sukumar Ghosh. *Distributed Systems: An Algorithmic Approach, Second Edition*. CRC Press, 2014.

[35] D. E. Applegate, A. von der Linden, P. L. Kelly, M. T. Allen, S. W. Allen, P. R. Burchat, D. L. Burke, H. Ebeling, A. Mantz, and R. G. Morris. Weighing the Giants - III. Methods and measurements of accurate galaxy cluster weak-lensing masses. *Mon. Not. R. Astron. Soc.*, 439(1):48–72, feb 2014.

[36] G. B. Dalton, R. A. C. Croft, G. Efstathiou, W. J. Sutherland, S. J. Maddox, and M. Davis. The Two-Point Correlation Function of Rich Clusters of Galaxies: Results from an Extended APM Cluster Redshift Survey. page 5, jul 1994.

[37] Nobelprize.org. The Nobel Prize in Physics 2011, 2014.

[38] D. Gruen, S. Seitz, F. Brimioulle, R. Kosyra, J. Koppenhoefer, C.-H. Lee, R. Bender, A. Riffeser, T. Eichner, T. Weidinger, and M. Bierschenk. Weak lensing analysis of SZ-selected clusters of galaxies from the SPT and Planck surveys. *Mon. Not. R. Astron. Soc.*, 442(2):1507–1544, jun 2014.

[39] CosmoloPy: a cosmology package for Python.

[40] David W. Hogg. Distance measures in cosmology. *arXiv*, 2000.

[41] The Dark Energy Survey Collaboration. The Dark Energy Survey. page 42, oct 2005.

[42] Megan Donahue, G. Mark Voit, Andisheh Mahdavi, Keiichi Umetsu, Stefano Ettori, Julian Merten, Marc Postman, Aaron Hoffer, Alessandro Baldi, Dan Coe, Nicole Czakon, Mattias Bartelmann, Narciso Benitez, Rychard Bouwens, Larry Bradley, Tom Broadhurst, Holland Ford, Fabio Gastaldello, Claudio Grillo, Leopoldo Infante, Stephanie Jouvel, Anton Koekemoer, Daniel Kelson, Ofer Lahav, Doron Lemze, Elinor Medezinski, Peter Melchior, Massimo Meneghetti, Alberto Molino, John Moustakas, Leonidas A. Moustakas, Mario Nonino, Piero Rosati, Jack Sayers, Stella Seitz, Arjen Van der Wel, Wei Zheng, and Adi Zitrin. CLASH-X: A COMPARISON OF LENSING AND X-RAY TECHNIQUES FOR MEASURING THE MASS PROFILES OF GALAXY CLUSTERS. *Astrophys. J.*, 794(2):136, oct 2014.

[43] Jon Louis Bentley. Multidimensional binary search trees used for associative searching. *Commun. ACM*, 18(9):509–517, 1975.

[44] J. Hunter, D. Dale, and M. Droettboom. matplotlib: python plotting Matplotlib v1.0.0 documentation, 2010.

[45] Eugenie Samuel Reich. Missing galaxy mass found. *Nature*, 506(7488):274–5, feb 2014.

[46] P. J. E. Peebles. Statistical analysis of catalogs of extragalactic objects. VI - The galaxy distribution in the Jagellonian field. *Astrophys. J.*, 196:647, mar 1975.

[47] Marios D. Dikaiakos and Joachim Stadel. A performance study of cosmological simulations on message-passing and shared-memory multiprocessors. In *Proc. 10th Int. Conf. Supercomput. - ICS '96*, pages 94–101, New York, New York, USA, jan 1996. ACM Press.

[48] N. Kaiser. Clustering in real space and in redshift space. *Mon. Not. R. Astron. Soc.*, 227(1):1–21, jul 1987.

[49] M. G. Hauser and P. J. E. Peebles. Statistical Analysis of Catalogs of Extragalactic Objects. 11. the Abell Catalog of Rich Clusters. *Astrophys. J.*, 185:757, nov 1973.

[50] Edwin Hubble. The Distribution of Extra-Galactic Nebulae. *Astrophys. J.*, 79:8, jan 1934.

[51] M. Davis, J. Huchra, D. W. Latham, and J. Tonry. A survey of galaxy redshifts. II - The large scale space distribution. *Astrophys. J.*, 253:423, feb 1982.

[52] R. P. Kirshner, Jr. Oemler, A., and P. L. Schechter. A study of field galaxies. I - Redshifts and photometry of a complete sample of galaxies. *Astron. J.*, 83:1549, dec 1978.

[53] Andrew Liddle. *An Introduction to Modern Cosmology.* John Wiley & Sons, 2015.

[54] M. Joeveer, J. Einasto, and E. Tago. Spatial distribution of galaxies and of clusters of galaxies in the southern galactic hemisphere. *Mon. Not. R. Astron. Soc.*, 185(2):357–370, nov 1978.

[55] Matt Massie and Contributors. Ganglia Monitoring System, 2014.

[56] Micol Bolzonella, Joan-Marc Miralles, and Roser Pello'. Photometric Redshifts based on standard SED fitting procedures. page 19, mar 2000.

[57] George R. Blumenthal, S. M. Faber, Joel R. Primack, and Martin J. Rees. Formation of galaxies and large-scale structure with cold dark matter. *Nature*, 311(5986):517–525, oct 1984.

[58] C. D. Shane and C. A. Wirtanen. The distribution of extragalactic nebulae. *Astron. J.*, 59:285, sep 1954.

[59] David W. Hogg. Distance measures in cosmology. may 1999.

[60] Welcome to healpy documentation! healpy 1.9.1 documentation.

[61] HIPASS Data Access.

[62] CosmoloPy: a cosmology package for Python.

[63] Donald G. York, J. Adelman, John E. Anderson, Jr., Scott F. Anderson, James Annis, Neta A. Bahcall, J. A. Bakken, Robert Barkhouser, Steven Bastian, Eileen Berman, William N. Boroski, Steve Bracker, Charlie Briegel, John W. Briggs, J. Brinkmann, Robert Brunner, Scott Burles, Larry Carey, Michael A. Carr, Francisco J. Castander, Bing Chen, Patrick L. Colestock, A. J. Connolly, J. H. Crocker, István Csabai, Paul C. Czarapata, John Eric Davis, Mamoru Doi, Tom Dombeck, Daniel Eisenstein, Nancy Ellman, Brian R. Elms, Michael L. Evans, Xiaohui Fan, Glenn R. Federwitz, Larry Fiscelli, Scott Friedman, Joshua A. Frieman, Masataka Fukugita, Bruce Gillespie, James E. Gunn, Vijay K. Gurbani, Ernst de Haas, Merle Haldeman, Frederick H. Harris, J. Hayes, Timothy M. Heckman, G. S. Hennessy, Robert B. Hindsley, Scott Holm, Donald J. Holmgren, Chi-hao

Huang, Charles Hull, Don Husby, Shin-Ichi Ichikawa, Takashi Ichikawa, Željko Ivezić, Stephen Kent, Rita S. J. Kim, E. Kinney, Mark Klaene, A. N. Kleinman, S. Kleinman, G. R. Knapp, John Korienek, Richard G. Kron, Peter Z. Kunszt, D. Q. Lamb, B. Lee, R. French Leger, Siriluk Limmongkol, Carl Lindenmeyer, Daniel C. Long, Craig Loomis, Jon Loveday, Rich Lucinio, Robert H. Lupton, Bryan MacKinnon, Edward J. Mannery, P. M. Mantsch, Bruce Margon, Peregrine McGehee, Timothy A. McKay, Avery Meiksin, Aronne Merelli, David G. Monet, Jeffrey A. Munn, Vijay K. Narayanan, Thomas Nash, Eric Neilsen, Rich Neswold, Heidi Jo Newberg, R. C. Nichol, Tom Nicinski, Mario Nonino, Norio Okada, Sadanori Okamura, Jeremiah P. Ostriker, Russell Owen, A. George Pauls, John Peoples, R. L. Peterson, Donald Petravick, Jeffrey R. Pier, Adrian Pope, Ruth Pordes, Angela Prosapio, Ron Rechenmacher, Thomas R. Quinn, Gordon T. Richards, Michael W. Richmond, Claudio H. Rivetta, Constance M. Rockosi, Kurt Ruthmansdorfer, Dale Sandford, David J. Schlegel, Donald P. Schneider, Maki Sekiguchi, Gary Sergey, Kazuhiro Shimasaku, Walter A. Siegmund, Stephen Smee, J. Allyn Smith, S. Snedden, R. Stone, Chris Stoughton, Michael A. Strauss, Christopher Stubbs, Mark SubbaRao, Alexander S. Szalay, Istvan Szapudi, Gyula P. Szokoly, Anirudda R. Thakar, Christy Tremonti, Douglas L. Tucker, Alan Uomoto, Dan Vanden Berk, Michael S. Vogeley, Patrick Waddell, Shu-i Wang, Masaru Watanabe, David H. Weinberg, Brian Yanny, and Naoki Yasuda. The Sloan Digital Sky Survey: Technical Summary. *Astron. J.*, 120(3):1579–1587, sep 2000.

[64] N. Kaiser. On the spatial correlations of Abell clusters. *Astrophys. J.*, 284:L9, sep 1984.

[65] E. P. Hubble. Extragalactic nebulae. *Astrophys. J.*, 64:321, dec 1926.

[66] D. Schlegel, F. Abdalla, T. Abraham, C. Ahn, C. Allende Prieto, J. Annis, E. Aubourg, M. Azzaro, S. Bailey. C. Baltay, C. Baugh, C. Bebek, S. Becerril, M. Blanton, A. Bolton, B. Bromley, R. Cahn, P. H. Carton, J. L. Cervantes-Cota, Y. Chu, M. Cortes, K. Dawson, A. Dey, M. Dickinson, H. T. Diehl, P. Doel, A. Ealet, J. Edelstein, D. Eppelle, S. Escoffier, A. Evrard, L. Faccioli, C. Frenk, M. Geha, D. Gerdes, P. Gondolo, A. Gonzalez-Arroyo, B. Grossan, T. Heckman, H. Heetderks, S. Ho, K. Honscheid, D. Huterer, O. Ilbert, I. Ivans, P. Jelinsky, Y. Jing, D. Joyce, R. Kennedy, S. Kent, D. Kieda, A. Kim, C. Kim, J. P. Kneib, X. Kong, A. Kosowsky, K. Krishnan, O. Lahav, M. Lampton, S. LeBohec, V. Le Brun, M. Levi, C. Li, M. Liang, H. Lim, W. Lin, E. Linder, W. Lorenzon, A. de la Macorra, Ch. Magneville, R. Malina, C. Marinoni, V. Martinez, S. Majewski, T. Matheson, R. McCloskey, P. McDonald, T. McKay, J. McMahon, B. Menard, J. Miralda-Escude, M. Modjaz, A. Montero-Dorta, I. Morales, N. Mostek, J. Newman,

R. Nichol, P. Nugent, K. Olsen, N. Padmanabhan, N. Palanque-Delabrouille, I. Park, J. Peacock, W. Percival, S. Perlmutter, C. Peroux, P. Petitjean, F. Prada, E. Prieto, J. Prochaska, K. Reil, C. Rockosi, N. Roe, E. Rollinde, A. Roodman, N. Ross, G. Rudnick, V. Ruhlmann-Kleider, J. Sanchez, D. Sawyer, C. Schimd, M. Schubnell, R. Scoccimaro, U. Seljak, H. Seo, E. Sheldon, M. Sholl, R. Shulte-Ladbeck, A. Slosar, D. S. Smith, G. Smoot, W. Springer, A. Stril, A. S. Szalay, C. Tao, G. Tarle, E. Taylor, A. Tilquin, J. Tinker, F. Valdes, J. Wang, T. Wang, B. A. Weaver, D. Weinberg, M. White, M. Wood-Vasey, J. Yang, X. Yang. Ch. Yeche, N. Zakamska, A. Zentner, C. Zhai, and P. Zhang. The BigBOSS Experiment. jun 2011.

[67] J. Beringer, J. F. Arguin, R. M. Barnett, K. Copic, O. Dahl, D. E. Groom, C. J. Lin, J. Lys, H. Murayama, C. G. Wohl, W. M. Yao, P. A. Zyla, C. Amsler, M. Antonelli, D. M. Asner, H. Baer, H. R. Band, T. Basaglia, C. W. Bauer, J. J. Beatty, V. I. Belousov, E. Bergren, G. Bernardi, W. Bertl, S. Bethke, H. Bichsel, O. Biebel, E. Blucher, S. Blusk, G. Brooijmans, O. Buchmueller, R. N. Cahn, M. Carena, A. Ceccucci, D. Chakraborty, M. C. Chen, R. S. Chivukula, G. Cowan, G. D'Ambrosio, T. Damour, D. de Florian, A. de Gouvêa, T. DeGrand, P. de Jong, G. Dissertori, B. Dobrescu, M. Doser, M. Drees, D. A. Edwards, S. Eidelman, J. Erler, V. V. Ezhela, W. Fetscher, B. D. Fields, B. Foster, T. K. Gaisser, L. Garren, H. J. Gerber, G. Gerbier, T. Gherghetta, S. Golwala, M. Goodman, C. Grab, A. V. Gritsan, J. F. Grivaz, M. Grünewald, A. Gurtu, T. Gutsche, H. E. Haber, K. Hagiwara, C. Hagmann, C. Hanhart, S. Hashimoto, K. G. Hayes, M. Heffner, B. Heltsley, J. J. Hernández-Rey, K. Hikasa, A. Höcker, J. Holder, A. Holtkamp, J. Huston, J. D. Jackson, K. F. Johnson, T. Junk, D. Karlen, D. Kirkby, S. R. Klein, E. Klempt, R. V. Kowalewski, F. Krauss, M. Kreps, B. Krusche, Yu. V. Kuyanov, Y. Kwon, O. Lahav, J. Laiho, P. Langacker, A. Liddle, Z. Ligeti, T. M. Liss, L. Littenberg, K. S. Lugovsky, S. B. Lugovsky, T. Mannel, A. V. Manohar, W. J. Marciano, A. D. Martin, A. Masoni, J. Matthews, D. Milstead, R. Miquel, K. Mönig, F. Moortgat, K. Nakamura, M. Narain, P. Nason, S. Navas, M. Neubert, P. Nevski, Y. Nir, K. A. Olive, L. Pape, J. Parsons, C. Patrignani, J. A. Peacock, S. T. Petcov, A. Piepke, A. Pomarol, G. Punzi, A. Quadt, S. Raby, G. Raffelt, B. N. Ratcliff, P. Richardson, S. Roesler, S. Rolli, A. Romaniouk, L. J. Rosenberg, J. L. Rosner, C. T. Sachrajda, Y. Sakai, G. P. Salam, S. Sarkar, F. Sauli, O. Schneider, K. Scholberg, D. Scott, W. G. Seligman, M. H. Shaevitz, S. R. Sharpe, M. Silari, T. Sjöstrand, P. Skands, J. G. Smith, G. F. Smoot, S. Spanier, H. Spieler, A. Stahl, T. Stanev, S. L. Stone, T. Sumiyoshi, M. J. Syphers, F. Takahashi, M. Tanabashi, J. Terning, M. Titov, N. P. Tkachenko, N. A. Törnqvist, D. Tovey, G. Valencia, K. van Bibber, G. Venanzoni, M. G. Vincter, P. Vogel, A. Vogt, W. Walkowiak, C. W. Walter, D. R. Ward, T. Watari, G. Weiglein, E. J. Weinberg, L. R. Wiencke,

L. Wolfenstein, J. Womersley, C. L. Woody, R. L. Workman, A. Yamamoto, G. P. Zeller, O. V. Zenin, J. Zhang, R. Y. Zhu, G. Harper, V. S. Lugovsky, and P. Schaffner. Review of Particle Physics. *Phys. Rev. D*, 86(1):010001, jul 2012.

[68] Willem B. Drees. *Beyond the Big Bang: Quantum Cosmologies and God*. Open Court Publishing, 1990.

[69] Pablo Fosalba, Enrique Gaztañaga, Francisco J. Castander, and Marc Manera. The onion universe: all sky lightcone simulations in spherical shells. *Mon. Not. R. Astron. Soc.*, 391(1):435–446, nov 2008.

[70] Henk Hoekstra, Andisheh Mahdavi, Arif Babul, and Chris Bildfell. The Canadian Cluster Comparison Project: weak lensing masses and SZ scaling relations. *Mon. Not. R. Astron. Soc.*, 427(2):1298–1311, dec 2012.

[71] N. Battaglia, A. Leauthaud, H. Miyatake, M. Hasselfield, M.B. Gralla, R. Allison, J.R. Bond, E. Calabrese, D. Crichton, M.J. Devlin, J. Dunkley, R. Dünner, T. Erben, S. Ferrara, M. Halpern, M. Hilton, J.C. Hill, A.D. Hincks, R. Hložek, K.M. Huffenberger, J.P. Hughes, J.P. Kneib, A. Kosowsky, M. Makler, T.A. Marriage, F. Menanteau, L. Miller, K. Moodley, B. Moraes, M.D. Niemack, L. Page, H. Shan, N. Sehgal, B.D. Sherwin, J.L. Sievers, C. Sifón, D.N. Spergel, S.T. Staggs, J. Taylor, R. Thornton, L. vanWaerbeke, and E.J. Wollack. Weak-Lensing Mass Calibration of the Atacama Cosmology Telescope Equatorial Sunyaev-Zeldovich Cluster Sample with the Canada-France-Hawaii Telescope Stripe 82 Survey. *eprint arXiv:1509.08930*, 2015.

[72] ACE Lab.

[73] Astropy.

[74] Jacob VanderPlas, Andrew J. Connolly, Zeljko Ivezic, and Alex Gray. Introduction to astroML: Machine learning for astrophysics. In *2012 Conf. Intell. Data Underst.*, pages 47–54. IEEE, oct 2012.

[75] Intel Xeon E5-2690 vs E5-2670.

[76] R. K. Sheth and G. Tormen. Large-scale bias and the peak background split. *Mon. Not. R. Astron. Soc.*, 308(1):119–126, sep 1999.

[77] A. A. Penzias and R. W. Wilson. A Measurement of Excess Antenna Temperature at 4080 Mc/s. *Astrophys. J.*, 142:419, jul 1965.

[78] Mark Baker and Rajkumar Buyya. Cluster computing: the commodity supercomputer. *Softw. Pract. Exp.*, 29(6):551–576, may 1999.

[79] E. Strohmaier, J. Dongarra, H. Simon, and M. Meuer. June 2015 — TOP500 Supercomputer Sites, 2015.

[80] M. Crocce, F. J. Castander, E. Gaztañaga, P. Fosalba, and J. Carretero. The MICE Grand Challenge lightcone simulation II. Halo and galaxy catalogues. *Mon. Not. R. Astron. Soc.*, 453(2):1513–1530, aug 2015.

[81] John S Lewis. *Mining the Sky*. ESO ASTROPHYSICS SYMPOSIA. Springer-Verlag, Berlin/Heidelberg, dec 1996.

[82] Martin Kerscher, István Szapudi, and Alexander S. Szalay. A Comparison of Estimators for the Two-Point Correlation Function. *Astrophys. J.*, 535(1):L13–L16, may 2000.

[83] Tsuyoshi Hamada and Keigo Nitadori. 190 TFlops Astrophysical N-body Simulation on a Cluster of GPUs. In *2010 ACM/IEEE Int. Conf. High Perform. Comput. Networking, Storage Anal.*, pages 1–9. IEEE, nov 2010.

[84] Tony Greicius. Planck: Exploring the Birth of Our Universe, jun 2013.

[85] Crystian Sadiel Venegas-Barrera and Javier Manjarrez. Patrones espaciales de la riqueza espec??fica de las culebras Thamnophis en M??xico. *Rev. Mex. Biodivers.*, 82(1):179–191, nov 2011.

[86] M. Takada and B. Jain. The three-point correlation function in cosmology. *Mon. Not. R. Astron. Soc.*, 340(2):580–608, apr 2003.

[87] G. S. Almasi and A. Gottlieb. Highly parallel computing. jan 1989.

[88] Planck Collaboration, P. A. R. Ade, N. Aghanim, M. Arnaud, M. Ashdown, J. Aumont, C. Baccigalupi, A. J. Banday, R. B. Barreiro, R. Barrena, J. G. Bartlett, N. Bartolo, E. Battaner, R. Battye, K. Benabed, A. Benoît, A. Benoit-Lévy, J. P. Bernard, M. Bersanelli, P. Bielewicz, I. Bikmaev, H. Böhringer, A. Bonaldi, L. Bonavera, J. R. Bond, J. Borrill, F. R. Bouchet, M. Bucher, R. Burenin, C. Burigana, R. C. Butler, E. Calabrese, J. F. Cardoso, P. Carvalho, A. Catalano, A. Challinor, A. Chamballu, R. R. Chary, H. C. Chiang, G. Chon, P. R. Christensen, D. L. Clements, S. Colombi, L. P. L. Colombo, C. Combet, B. Comis, F. Couchot, A. Coulais, B. P. Crill, A. Curto, F. Cuttaia, H. Dahle, L. Danese, R. D. Davies, R. J. Davis, P. de Bernardis, A. de Rosa, G. de Zotti, J. Delabrouille, F. X. Désert, C. Dickinson, J. M. Diego, K. Dolag, H. Dole, S. Donzelli, O. Doré, M. Douspis, A. Ducout, X. Dupac, G. Efstathiou, P. R. M. Eisenhardt, F. Elsner, T. A. Enßlin, H. K. Eriksen, E. Falgarone, J. Fergusson, F. Feroz, A. Ferragamo, F. Finelli, O. Forni, M. Frailis, A. A. Fraisse, E. Franceschi, A. Frejsel, S. Galeotta,

S. Galli, K. Ganga, R. T. Génova-Santos, M. Giard, Y. Giraud-Héraud, E. Gjerløw, J. González-Nuevo, K. M. Górski, K. J. B. Grainge, S. Gratton, A. Gregorio, A. Gruppuso, J. E. Gudmundsson, F. K. Hansen, D. Hanson, D. L. Harrison, A. Hempel, S. Henrot-Versillé, C. Hernández-Monteagudo, D. Herranz, S. R. Hildebrandt, E. Hivon, M. Hobson, W. A. Holmes, A. Hornstrup, W. Hovest, K. M. Huffenberger, G. Hurier, A. H. Jaffe, T. R. Jaffe, T. Jin, W. C. Jones, M. Juvela, E. Keihänen, R. Keskitalo, I. Khamitov, T. S. Kisner, R. Kneissl, J. Knoche, M. Kunz, H. Kurki-Suonio, G. Lagache, J. M. Lamarre, A. Lasenby, M. Lattanzi, C. R. Lawrence, R. Leonardi, J. Lesgourgues, F. Levrier, M. Liguori, P. B. Lilje, M. Linden-Vørnle, M. López-Caniego, P. M. Lubin, J. F. Macías-Pérez, G. Maggio, D. Maino, D. S. Y. Mak, N. Mandolesi, A. Mangilli, P. G. Martin, E. Martínez-González, S. Masi, S. Matarrese, P. Mazzotta, P. McGehee, S. Mei, A. Melchiorri, J. B. Melin, L. Mendes, A. Mennella, M. Migliaccio, S. Mitra, M. A. Miville-Deschênes, A. Moneti, L. Montier, G. Morgante, D. Mortlock, A. Moss, D. Munshi, J. A. Murphy, P. Naselsky, A. Nastasi, F. Nati, P. Natoli, C. B. Netterfield, H. U. Nørgaard-Nielsen, F. Noviello, D. Novikov, I. Novikov, M. Olamaie, C. A. Oxborrow, F. Paci, L. Pagano, F. Pajot, D. Paoletti, F. Pasian, G. Patanchon, T. J. Pearson, O. Perdereau, L. Perotto, Y. C. Perrott, F. Perrotta, V. Pettorino, F. Piacentini, M. Piat, E. Pierpaoli, D. Pietrobon, S. Plaszczynski, E. Pointecouteau, G. Polenta, G. W. Pratt, G. Prézeau, S. Prunet, J. L. Puget, J. P. Rachen, W. T. Reach, R. Rebolo, M. Reinecke, M. Remazeilles, C. Renault, A. Renzi, I. Ristorcelli, G. Rocha, C. Rosset, M. Rossetti, G. Roudier, E. Rozo, J. A. Rubiño-Martín, C. Rumsey, B. Rusholme, E. S. Rykoff, M. Sandri, D. Santos, R. D. E. Saunders, M. Savelainen, G. Savini, M. P. Schammel, D. Scott, M. D. Seiffert, E. P. S. Shellard, T. W. Shimwell, L. D. Spencer, S. A. Stanford, D. Stern, V. Stolyarov, R. Stompor, A. Streblyanska, R. Sudiwala, R. Sunyaev, D. Sutton, A. S. Suur-Uski, J. F. Sygnet, J. A. Tauber, L. Terenzi, L. Toffolatti, M. Tomasi, D. Tramonte, M. Tristram, M. Tucci, J. Tuovinen, G. Umana, L. Valenziano, J. Valiviita, B. Van Tent, P. Vielva, F. Villa, L. A. Wade, B. D. Wandelt, I. K. Wehus, S. D. M. White, E. L. Wright, D. Yvon, A. Zacchei, and A. Zonca. Planck 2015 results. XXVII. The Second Planck Catalogue of Sunyaev-Zeldovich Sources. page 41, feb 2015.

[89] D. J. Fixsen. THE TEMPERATURE OF THE COSMIC MICROWAVE BACKGROUND. *Astrophys. J.*, 707(2):916–920, dec 2009.

[90] P.J.E. Peebles. The large-scale structure of the universe. *Res. Support. by Natl. Sci. Found. Princet.*, 1980.

[91] B. Kirk, M. Hilton, C. Cress, S. M. Crawford, J. P. Hughes, N. Battaglia, J. R. Bond, C. Burke, M. B. Gralla, A. Hajian, M. Hasselfield, A. D. Hincks, L. Infante,

A. Kosowsky, T. A. Marriage, F. Menanteau, K. Moodley, M. D. Niemack, J. L. Sievers, C. Sifon, S. Wilson, E. J. Wollack, and C. Zunckel. SALT spectroscopic observations of galaxy clusters detected by ACT and a type II quasar hosted by a brightest cluster galaxy. *Mon. Not. R. Astron. Soc.*, 449(4):4010–4026, apr 2015.

Appendix A

A

A.1 Celestial Coordinates

This section presents the celestial coordinates used in astronomy to measure the position of objects in the sky, which includes a brief discussion on equatorial coordinates, right ascension and declination. It also includes an overview look at redshift.

A.1.1 Equatorial coordinate system

The equatorial systems is the most widely used celestial coordinate system for specifying positions of objects and modern star maps almost exclusively use this system[25]. The coordinates are geocentric, thus the origin is at the center of the Earth. The fundamental plane is a projection of the Earth's equator onto the celestial sphere and the primary direction towards the vernal equinox. This means that while the coordinate system is aligned with the Earth's equator and pole, it doesn't rotate with the Earth, but is relatively fixed against the background stars. The popular choices of pole and equator are the B1950 and the modern J2000 systems, that allows positions established at various dates to be compared directly. The coordinates are often expressed as a pair of right ascension and declination. An illustration of the equatorial celestial coordinate system is shown in Figure A.1.

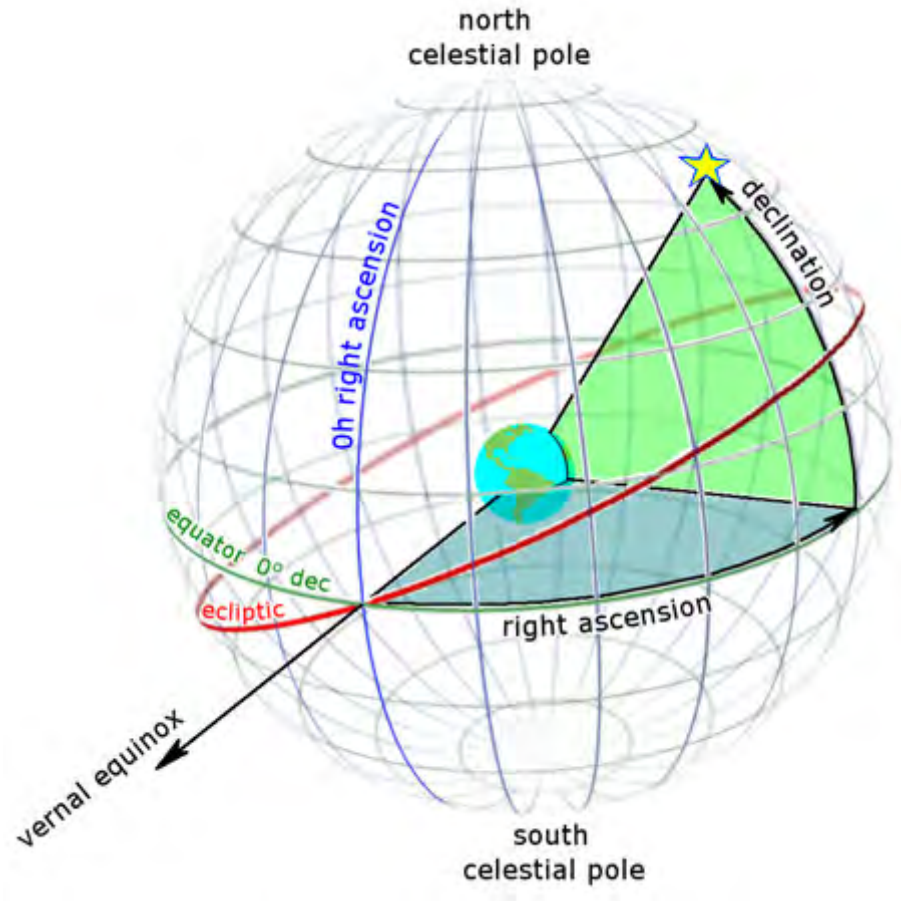


Figure A.1: The equatorial coordinate system in spherical coordinates. The fundamental plane is formed by projection of the Earth's equator onto the celestial sphere, forming the celestial equator (blue). The primary direction is established by projecting the Earth's orbit onto the celestial sphere, forming the ecliptic (red), and setting up the ascending node of the ecliptic on the celestial equator, the vernal equinox. Right ascensions are measured eastward along the celestial equator from the equinox, and declinations are measured positive northward from the celestial equator - two such coordinate pairs are shown here. Projections of the Earth's north and south geographic poles form the north and south celestial poles, respectively.

Declination(δ) is analogous to the geographical latitude. It is used to measure the angular distance of an object perpendicular to the celestial sphere. Objects located north of the celestial equator have positive declinations, whereas those south have negative declinations. The standard units of measurement used are degrees($^\circ$), minutes($'$), seconds($''$), with 90° equivalent to $\frac{1}{4}$ circle. Therefore, the south and north celestial poles are located at (-90°, 90°) respectively, while the the celestial equator is at (0°).

The right ascension (α) is equivalent to the geographical longitude. It is used to measure the angular distance of an object eastward along the equator relative to the vernal equinox. The standard units of measure is hours(^{h}), minutes($'$) and seconds ($''$) instead of degree. This is done because the objects location is measured by timing its passage

across the highest point in the sky (meridian) as the Earth rotates. There are $24h$ of right ascension across the full celestial equator, thus one hour of right ascension an equivalent of $15°$ ($360°$ / $24h$).

A.2 Cosmological Redshift

Considering the Universe is expanding, the light/electromagnetic radiation travelling through space time is effectively stretched. This effect appears as "reddening" of the original light source in the visible spectrum, hence it is known as the cosmological redshift. This phenomenon is similar to the Doppler Effect, wherein the is a change in frequency of a wave for an observer moving relative to the source of signal. However, cosmological redshift cannot be equivalent to the Doppler Effect. This is because two objects with zero velocity cannot experience a Doppler frequency shift in a signal sent between them, whereas on sufficiently large scales the same object can experience a frequency shift due to the cosmological expansion of the space between them [16]. The redshift of an object can be expressed by the following equation:

$$z = \frac{\lambda_0 - \lambda}{\lambda} \tag{A.1}$$

where z is the symbol for redshift, λ is the emission wavelength and λ_0 is the observed wavelength. An illustration of redshift is shown by arrows in Figure A.2, which shows the change in frequency of received signal relative to a non-redshifted source. The redshift effect is also used to describe the relationship between a objects apparent velocity and it's distance from the observer.

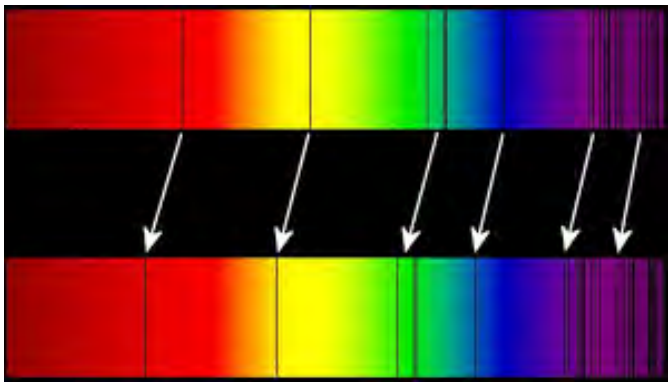


Figure A.2: Diagram illustrating the frequency shift (redshift) resulting from a signal's passage through space-time[16].This shift results in a emitted signal being detected at a lower frequency. If the original emission frequency is known, the distance to the source of the signal can be calculated using above equation A.1

A.3 comoving distance

The comoving distance describes the distance between two points along a path measured at the present cosmological time. It factors out the expansion of the universe, providing a constant distance that does not change with the expansion of space. Therefore, the proper distance is equivalent to the comoving distance at the present time, yielding a scale factor of 1. The expression for computing the comoving distance is given by [32]

$$\chi = \int_{t_e}^{t} c\frac{dt'}{a(t')} \tag{A.2}$$

where $a(t')$ is the scale factor, t_e is the time of emission of the protons observed, t is the present time and c is the speed of light in a vacuum.

Although the proper distance between objects changes due to the expansion of the universe, the comoving distance remains unchanged. This allows for the definition of the comoving spatial coordinates system, which offers a more natural and easier to work with coordinate system. The scale factor increases with the expansion of the Universe and that can be use to reconcile constant comoving distances with the actual proper distance.

Appendix B

CODE SNIPPETS

B.1 Auto-correlating data distance separation

```
def comp_auto_dists(in_data, binsx, corr_type, Z_type):
    auto_hist = np.zeros(len(binsx)-1)
    N_indata = len(in_data)

    for i in range(N_indata -1):
        # Selecting Redshift (either photometric or spectroscopic)
        if(Z_type == 0):
            z_ph0 = in_data[i,2]
            z_ph1 = in_data[i+1:,2]
        else:
            z_ph0 = in_data[i,3]
            z_ph1 = in_data[i+1:,3]

        #   Refference point to compute the distance FROM
        RA_0 = in_data[i,0]
        DEC_0 = in_data[i,1]

        #   Refference points to compute the distance TO(all other point
    from       input catalog)
        RA_0 = in_data[i+1:,0]
        DEC_0 = in_data[i+1:,1]

        # Compute the angular seperation distance(great circle) between the
         points
        coefA = np.sin(DEC_0) * np.sin(DEC_1)
        coefB = np.cos(DEC_0) * np.cos(DEC_1) *  np.cos(np.fabs(RA_0 - RA_1
    ))

```

```
        ang_sep = np.arccos(coefA + coefB)

        if(corr_type == "3D"):  #  Compute the 3D distance based on (r,mu)
            d1_d2 = ((z_ph0)**2 + (z_ph1)**2)
            d1d1cosQ = (2*(z_ph0)*(z_ph1))*(np.cos(ang_sep))

            #Apply cosine rule and compute r
            sq_dist = np.sqrt(d1_d2 - d1d1cosQ)

            ## Computing mu
            d1d2_num = np.abs((z_ph0)**2 - (z_ph1)**2)
            d1d2_den = np.sqrt( (d1_d2)**2 - (d1d1cosQ)**2 )
            mu = d1d2_num/d1d2_den

            # combine distances of r,mu for each
            corr_rm = np.column_stack((sq_dist , mu))

            # filter the array for (0<mu<1)
            corr_rm = corr_rm[corr_rm[:,1]>0]
            corr_rm = corr_rm[corr_rm[:,1]<1]

            # filter the array for (0<r<1)
            corr_rm2 = corr_rm[corr_rm[:,0]>=min_CR]
            corr_rm2 = corr_rm2[corr_rm2[:,0]<=max_CR]

            #compute the counts bins distance seperation
            tmp_hist, binsO = np.array(np.histogram(corr_rm2[:,0], bins=
    binsx))

            #update the histogram
            auto_hist = auto_hist + tmp_hist
        else:   # Compute the angular distance based on (r,mu)
            ang_deg = np.degrees(ang_sep)
            ang_deg = ang_deg[ang_deg[:,]<=max_CR]
            tmp_hist, binsO = np.array(np.histogram(ang_deg, bins=binsx))
            auto_hist = auto_hist + tmp_hist

    return auto_hist
```

B.2 Cross-correlating data distance separation

```
def comp_cross_dists(in_data, rn_data, binsx, corr_type, Z_type):
    cross_hist = np.zeros(len(binsx)-1)
    N_indata = len(in_data)

    for i in range(N_indata -1):
```

```
        # Selecting Redshift (either photometric or spectroscopic)
        if(Z_type == 0):
            z_ph0 = in_data[i,2]
            z_ph1 = rn_data[:,2]
        else:
            z_ph0 = in_data[i,3]
            z_ph1 = rn_data[:,3]

        #   Refference point to compute the distance FROM(input catalog)
        RA_0 = in_data[i,0]
        DEC_0 = in_data[i,1]

        #   Refference points to compute the distance TO(random catalog)
        RA_1 = rn_data[:,0]
        DEC_1 = rn_data[:,1]

        # Compute the angular seperation distance(great circle) between the
         points
        coefA = np.sin(DEC_0) * np.sin(DEC_1)
        coefB = np.cos(DEC_0) * np.cos(DEC_1) *  np.cos(np.fabs(RA_0 - RA_1
    ))

        ang_sep = np.arccos(coefA + coefB)

        if(corr_type == "3D"):  #  Compute the 3D distance based on (r,mu)
            d1_d2 = ((z_ph0)**2 + (z_ph1)**2)
            d1d1cosQ = (2*(z_ph0)*(z_ph1))*(np.cos(ang_sep))

            #Apply cosine rule and compute r
            sq_dist = np.sqrt(d1_d2 - d1d1cosQ)

            ## Computing mu
            d1d2_num = np.abs((z_ph0)**2 - (z_ph1)**2)
            d1d2_den = np.sqrt( (d1_d2)**2 - (d1d1cosQ)**2 )
            mu = d1d2_num/d1d2_den

            # combine distances of r,mu for each
            corr_rm = np.column_stack((sq_dist, mu))

            # filter the array for (0<mu<1)
            corr_rm = corr_rm[corr_rm[:,1]>0]
            corr_rm = corr_rm[corr_rm[:,1]<1]

            # filter the array for (0<r<1)
            corr_rm2 = corr_rm[corr_rm[:,0]>=min_CR]
            corr_rm2 = corr_rm2[corr_rm2[:,0]<=max_CR]

```

```
            #compute the counts bins distance seperation
            tmp_hist, binsO = np.array(np.histogram(corr_rm2[:,0], bins=
    binsx))

            #update the histogram
            cross_hist = cross_hist + tmp_hist
        else:   # Compute the angular distance based on (r,mu)
            ang_deg = np.degrees(ang_sep)
            ang_deg = ang_deg[ang_deg[:,]<=max_CR]
            tmp_hist, binsO = np.array(np.histogram(ang_deg, bins=binsx))
            cross_hist = cross_hist + tmp_hist

    return cross_hist
```

B.3 Estimating the correlation function

```
def compute_2PCF_func(histDat, corr_Est="LS"):
    bins = histDat[:,0]     # Distance: Bins separation

    dd = (histDat[:,1])/(Nd*(Nd-1)/2)
    rr = (histDat[:,2])/(Nr*(Nr-1)/2)
    dr = (histDat[:,3])/(Nr*Nd)

    if(corr_Est == "LS"): # Landy-Szalay Estimator
        corr_fac = ((dd) - 2*(dr) + rr)/(rr)
        err_corr = (corr_fac+1)/(np.sqrt(histDat[:,1]))
        corr_data = np.column_stack((bins, corr_fac, err_corr,
           histDat[:,1], histDat[:,2], histDat[:,3]))
    else: # Peebles & Hauser Estimator
        corr_fac = (dd/rr)-1
        err_corr = (corr_fac+1)/(np.sqrt(histDat[:,1]))
        corr_data = np.column_stack((bins, corr_fac, err_corr,
          histDat[:,1], histDat[:,2]))

    return corr_data
```

B.4 CUTE 3D boxing scheme code

[11]

```
static int optimal_nside(double lb,double rmax,int np)
{
  //////
  // Estimates a good candidate for the size
```

```
5   // of a set of neighbor boxes
6
7   int nside1=(int)(FRACTION_AR*lb/rmax);    //nside1 -> 8 boxes per rmax
8   int nside2=(int)(pow(0.5*np,0.3333333)); //nside1 -> nside2^3<np/2
9
10  print_info("\n (nside1,nside2) = (%d, %d)    \n ", nside1,nside2);
11  return MIN(nside1,nside1);
12
13
14  // Determine optimal number of sides, total number of boxes for catalog
15  // and the dimensions for the boxes
16
17  nside=optimal_nside(l_box_max,rmax,cat_dat.np);
18
19  n_side[0]=(int)(nside*l_box[0]/l_box_max)+1;
20  n_side[1]=(int)(nside*l_box[1]/l_box_max)+1;
21  n_side[2]=(int)(nside*l_box[2]/l_box_max)+1;
22  n_boxes3D=n_side[0]*n_side[1]*n_side[2];
23
24  double dx=l_box[0]/n_side[0];
25  double dy=l_box[1]/n_side[1];
26  double dz=l_box[2]/n_side[2];
27
28  // Counts boxes containing objects
29  nfull=0;
30  for(ii=0;ii<cat.np;ii++) {
31  double x=cat.red[ii];
32  double y=cat.cth[ii];
33  double z=cat.phi[ii];
34  int ibox=xyz2box(x,y,z);
35  int np0=boxes[ibox].np;
36  if(np0==0) nfull++;
37  boxes[ibox].np++;
38
39  nfull=0;
40  for(ii=0;ii<n_boxes3D;ii++) {
41    if(boxes[ii].np>0) {
42      boxes[ii].pos=(double *)my_malloc(N_POS*boxes[ii].np*sizeof(double));
43      boxes[ii].np=0;
44
45      //Get box index
46      (*box_indices)[nfull]=ii;
47      nfull++;
48    }
49  }
50
51  //Associate each box cell with all objects contained within then.
```

```
52   for(ii=0;ii<cat.np;ii++) {
53     double x=cat.red[ii];
54     double y=cat.cth[ii];
55     double z=cat.phi[ii];
56     int ibox=xyz2box(x,y,z);
57     int np0=boxes[ibox].np;
58     boxes[ibox].pos[N_POS*np0]=x;
59     boxes[ibox].pos[N_POS*np0+1]=y;
60     boxes[ibox].pos[N_POS*np0+2]=z;
61 #ifdef _WITH_WEIGHTS
62     boxes[ibox].pos[N_POS*np0+3]=cat.weight[ii];
63 #endif //_WITH_WEIGHTS
64     boxes[ibox].np++;
65   }
66 }
```

B.5 Ethics Forms

EBE Faculty: Assessment of Ethics in Research Projects

Any person planning to undertake research in the Faculty of Engineering and the Built Environment at the University of Cape Town is required to complete this form before collecting or analysing data. When completed it should be submitted to the supervisor (where applicable) and from there to the Head of Department. If any of the questions below have been answered YES, and the applicant is NOT a fourth year student, the Head should forward this form for approval by the Faculty EIR committee: submit to Ms Zakiya Chikte (Zakiya.chikte@uct.ac.za); New EBE Building, Ph 021 650 5739). Students must include a copy of the completed form with the dissertation/thesis when it is submitted for examination.

Name of Principal Researcher/Student: Israel Tshililo **Department:** Electrical Engineering

If a Student: Yes **Degree:** Msc Electrical Engineering **Supervisor:** Prof Catherine Cress
Dr Simon Winberg

If a Research Contract indicate source of funding/sponsorship: CHPC studentship programme

Research Project Title: Galaxy Evolution, Cosmology and HPC: Clustering Studies Applied to Astronomy

Overview of ethics issues in your research project:

Question 1: Is there a possibility that your research could cause harm to a third party (i.e. a person not involved in your project)?		NO
Question 2: Is your research making use of human subjects as sources of data? If your answer is YES, please complete Addendum 2.		NO
Question 3: Does your research involve the participation of or provision of services to communities? If your answer is YES, please complete Addendum 3.		NO
Question 4: If your research is sponsored, is there any potential for conflicts of interest? If your answer is YES, please complete Addendum 4.		NO

If you have answered YES to any of the above questions, please append a copy of your research proposal, as well as any interview schedules or questionnaires (Addendum 1) and please complete further addenda as appropriate.

I hereby undertake to carry out my research in such a way that

- there is no apparent legal objection to the nature or the method of research; and
- the research will not compromise staff or students or the other responsibilities of the University;
- the stated objective will be achieved, and the findings will have a high degree of validity;
- limitations and alternative interpretations will be considered;
- the findings could be subject to peer review and publicly available; and
- I will comply with the conventions of copyright and avoid any practice that would constitute plagiarism.

Signed by:

	Full name and signature	Date
Principal Researcher/Student:	Israel Tshililo	16 February 2016

This application is approved by:

Supervisor (if applicable):		18 Feb 2016
HOD (or delegated nominee): Final authority for all assessments with NO to all questions and for all undergraduate research.		16/2/16
Chair : Faculty EIR Committee For applicants other than undergraduate students who have answered YES to any of the above questions.		