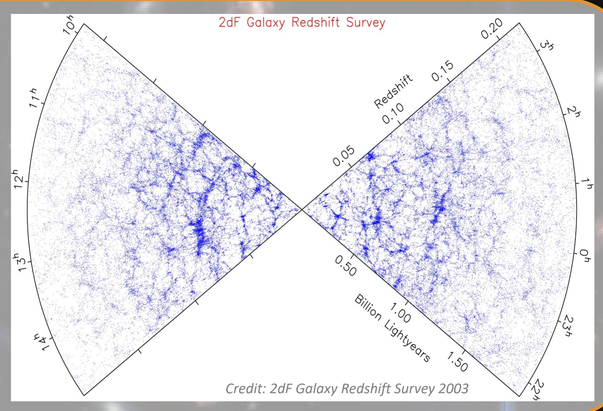
Monte Carlo Models of Galaxy Clustering

Pawel Janas, Prof. Peter Coles
Department of Theoretical Physics, Maynooth University
Pawel.Janas.2020@mumail.ie, Peter.Coles@mu.ie

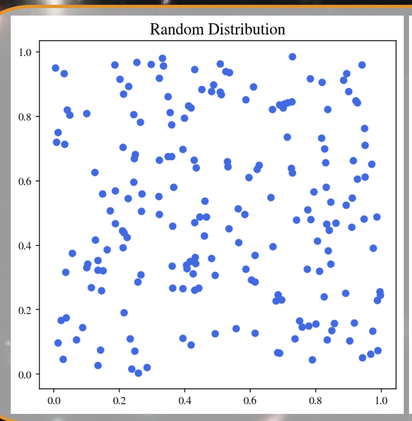
INTRODUCTION AND AIMS

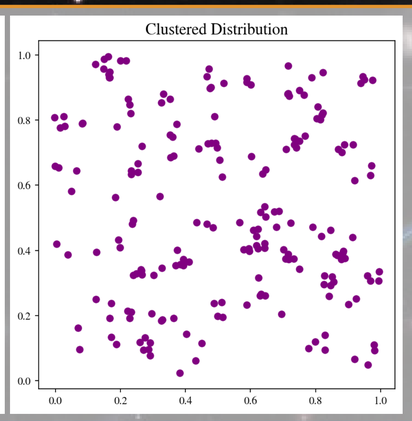
The universe has its own large scale structure. In order to investigate this, we need to look at it on the scale of galaxies. On this image on the right, there is a clear structure present. The aim of this project was to come up with tools to investigate this structure at a much smaller scale (<1000 galaxies), compared to the billions of galaxies surveyed like in the right image. To try and achieve good results while maintaining a high degree of accuracy, Monte Carlo models were built to simulate surveying different parts of the sky numerous times and performing analyses on the structure of galaxies observed.



Credit: 2dF Galaxy Redshift Survey 2003

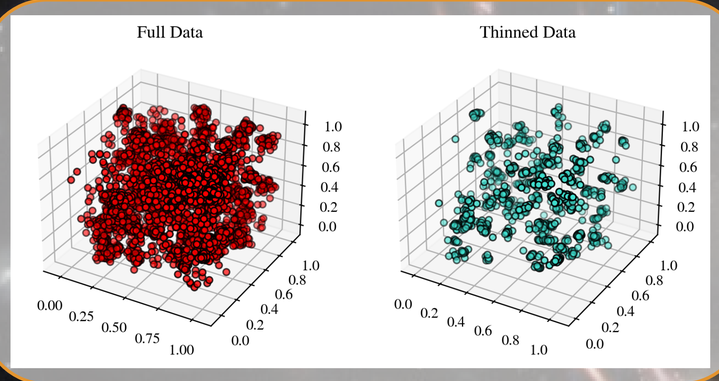
WHY NOT JUST LOOK?

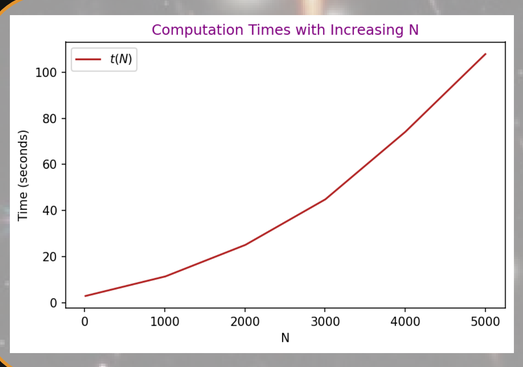




The human eye is biased when it comes to 'random stuff'. Take for instance, the left image which is said to have a random distribution, is there structure there? Many people (including myself) would say "YES!, of course there is!", but is this really the case? In order to justify this, algorithms had to be created in order to distinguish whether a distribution of galaxies (points on image) was random or did it have structure (clustered).

Why do we need Monte Carlo and how is it useful for this project?





Take the left image above, there's a lot of data and with more galaxies there's bound to be more noise in the results. Also, it's not even very realistic to consider so many galaxies within a small volume (here a unit of 1 would be approximately 10 mega parsecs i.e. 300 billion trillion metres or 3.08×10^{22} m). Instead, galaxies get sampled randomly and without replacement from the total catalogue and analysed. This is where Monte Carlo comes in. In short, the Monte Carlo (MC) method involves carrying out the same task numerous times and observing the overall trend that arises from doing so. At each iteration of the 'MC loop', a different set of galaxies gets sampled like the right image above, and analysed. Not only is this more powerful, it was computationally faster to analyse a set of N=200 galaxies compared to N=1000 galaxies.

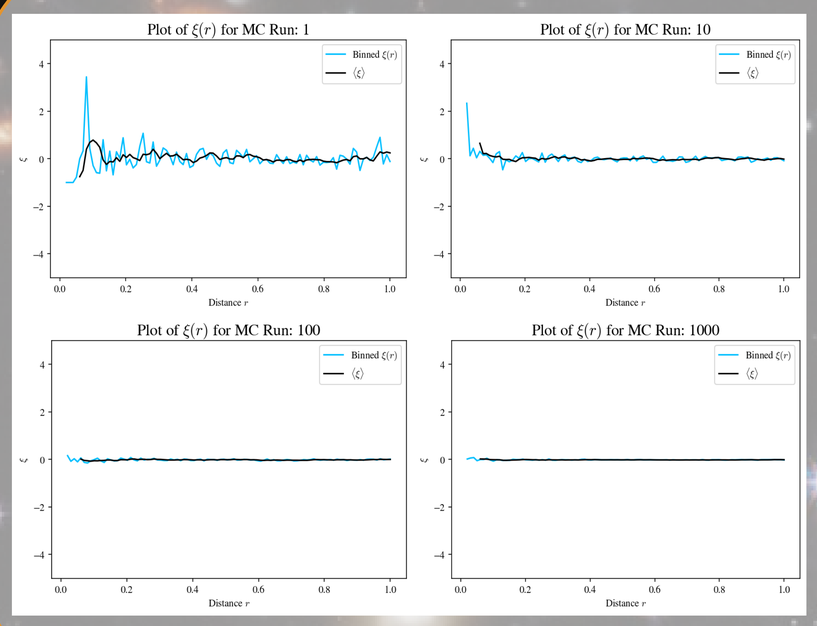
(Left) Computation times with increasing galaxy count N. Computation time is proportional to N^2.

SIMULATING CLUSTERED PATTERNS

- A number of clustered patterns were generated.
- The most challenging (but the most realistic) one to simulate was the power law model which involved generating a probability density function (pdf) (top right) to sample distance (r) values.
- The density profile $\rho(r)$ of each power-law cluster was fit to be proportional to $r^{-1.8}$ as intended (bottom right).
- This was done to simulate current redshift survey like data which presents a correlation function equal to $\xi(r) \sim r^{-1.8}$, just like the density profile.
- Before a full data catalogue was to be simulated, a singular test cluster would be generated to help debug any issues that may have arisen (bottom left) with the generation process.
- A total of 3 different cluster model algorithms were implemented.

```
#-------------------------MAIN PROGRAM START-------------------------#
L = 1
N = 100 # number of galaxies to place within each sphere
N_pois = 1000 # number of random points
RMAX = 0.07 # max radius of sphere 0.07

points_to_generate = N_pois//10 # number of points to place spheres on (N_pois//10

data_random = np.transpose(get_coords(L, N_pois))
# data_random = remove_low_pairs(data_random, RMAX) # (seems to have no effect)

N_pois = len(data_random) # update max count if remove_low_pairs() called
x = np.zeros(N_pois + points_to_generate * N)
y = np.zeros(N_pois + points_to_generate * N)
z = np.zeros(N_pois + points_to_generate * N)
x[:N_pois] = data_random[:,0]
y[:N_pois] = data_random[:,1]
z[:N_pois] = data_random[:,2]

number_of_rows = data_random.shape[0]

data = np.transpose([x, y, z])

sp_data = [] # split up data to run loop only on segments where generating new data
sp_data.append(data[:N_pois])
sp_data.append(np.array_split(data[N_pois:], points_to_generate))

# generate clusters around a random point
for i in range(points_to_generate):
    rand_indice = np.random.choice(number_of_rows)
    centre = data_random[rand_indice,:]
    for j in range(N):
        r = np.random.choice(rad) # set random radius from radius distribution
        sp_data[1][i][j,0], sp_data[1][i][j,1], \
            sp_data[1][i][j,2] = sample_spherical(1, r, centre)

# recombine the data into single array
data = sp_data[0]
for k in range(points_to_generate):
    ar_to_add = sp_data[1][k]
    data = np.concatenate((data, ar_to_add))

rand_data = np.transpose(get_coords(L, len(data)))

cluster_dist3D(L, data[:,0], data[:,1], data[:,2])
```

CORRELATING DATA SETS



The main aim of doing Monte Carlo models was to observe an emerging behaviour of statistics after some amount of iterations of the program. Here on the left we can see clearly just that. Two random catalogues of galaxies were correlated against each-other using a two point correlation function. Wherever you see a spike in these graphs, the two data sets are correlated. To put it another way, if ξ is positive at some distance r, there are more galaxies located at that distance from one another in the primary data set compared to the other and vice versa if it's negative.

It can be seen from the images on the left that for two random catalogues, they do not correlate as expected, yet it took approximately 100 runs of the program to see that clearly. We looked further at what the correlation function looks like for clustered distributions of galaxies.

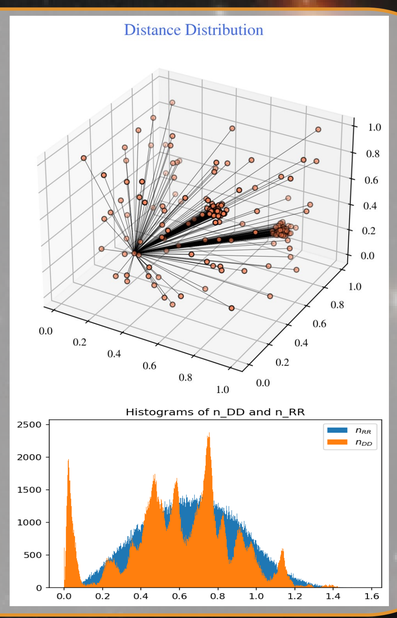
THE TWO-POINT CORRELATION FUNCTION

The two-point correlation function is one of the main statistical tools used for determining galaxy distributions in the universe. It describes the excess probability of finding two galaxies separated by some distance r. In this project, it was calculated as follows:

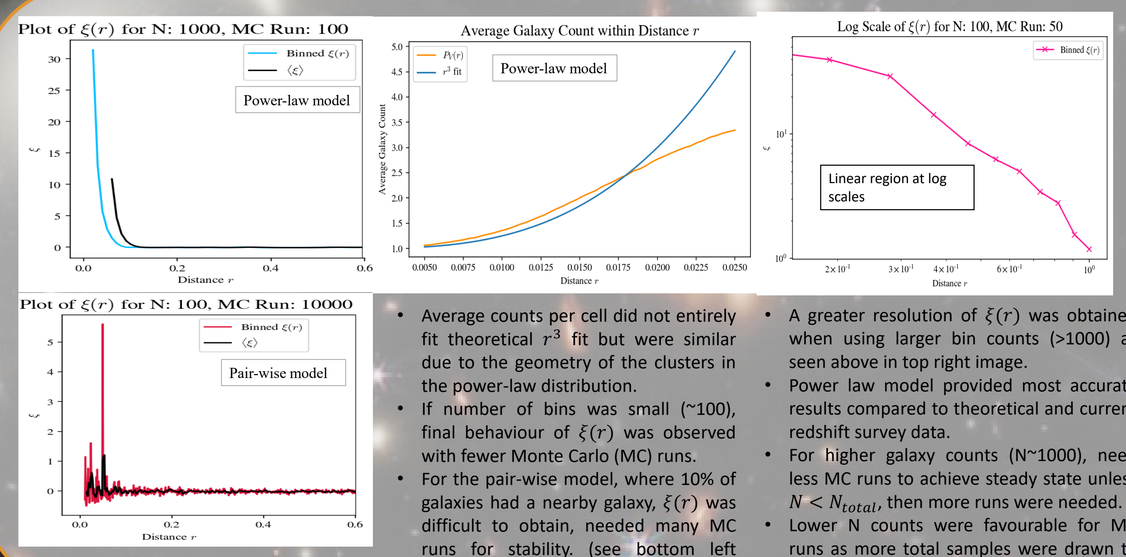
- The pairwise distance between each galaxy was calculated for the entire catalogue (image on right depicts one of these iterations).
- There is a total of N^2 of these values where N is the total number of galaxies in the catalogue.
- Three of these distance calculations were carried out; for the data catalogue (n_{DD}), cross correlation between data and random data catalogues (n_{DR}) and for the random catalogue itself (n_{RR}).
- These distance values were stored in two-dimensional arrays.
- Arrays were flattened and the values were 'binned' in a histogram in small intervals of distance r (100 bins to start).
- A ratio of these frequencies were taken to calculate the correlation function as follows;

$$\xi(r) = \frac{n_{DD} n_{RR}}{n_{DR}^2} - 1.$$

Top image on the right shows just one iteration of the pairwise distance calculation for the data catalogue n_{DD}, for which values would be stored in one column of the 2D array.
Bottom image shows what histograms of n_{DD} and n_{RR} look like with many bins (high resolution).



FINDINGS



- Average counts per cell did not entirely fit theoretical r^3 fit but were similar due to the geometry of the clusters in the power-law distribution.
- If number of bins was small (~100), final behaviour of $\xi(r)$ was observed with fewer Monte Carlo (MC) runs.
- For the pair-wise model, where 10% of galaxies had a nearby galaxy, $\xi(r)$ was difficult to obtain, needed many MC runs for stability. (see bottom left image)
- A greater resolution of $\xi(r)$ was obtained when using larger bin counts (>1000) as seen above in top right image.
- Power law model provided most accurate results compared to theoretical and current redshift survey data.
- For higher galaxy counts (N~1000), need less MC runs to achieve steady state unless $N < N_{total}$, then more runs were needed.
- Lower N counts were favourable for MC runs as more total samples were drawn to simulate sampling different parts of sky.

SCOPE FOR FUTURE RESEARCH

- Introduce boundary conditions for future defined geometries instead of just the base "cube" geometry as used in this project.
- Run program for larger boundaries of >1 (width = 10 or more) with increased total galaxy count.
- Try running with different geometries such as spherical or cone shaped that are in line with current surveys of this type.
- Introduce various more realistic clustering algorithms for data generation into one 'master' data set, perhaps with a higher width value L of the cube. Run analysis tools on this and see if results are closer to the expected values as obtained by current surveys.
- Using K-D trees for pairwise distance calculation for faster computation times instead of current blunt-force approach.
- Introduction of real, non-simulated data into the programs and run analysis tools on it. Specific data reduction techniques would be required.

CONCLUSIONS

- Catalogues of random, three dimensional data were generated.
- Successful models to simulate Poisson and Pairwise galaxy cluster models along with a model to fit a $r^{-1.8}$ density profile were created.
- Analysis methods such as the two-point correlation $\xi(r)$ function were implemented on the data sets and it was found that for two random catalogues, $\xi(r)$ tended to zero. This was clearer to see with more Monte Carlo loops of the program and with even greater accuracy if the bin number for histogram generation was higher.
- The correlation function and density profiles did not fit the $r^{-1.8}$ polynomial perfectly as seen on previous images but can be attributed to insufficient data sets, lack of variety in clustering regime and perhaps inadequate Monte Carlo runs of the program to achieve desired results.
- Galaxy counts for increasing distance r fit an r^3 polynomial for smaller separations as expected due to the geometry of the clusters themselves being three-dimensional sphere-like. Discrepancies at larger separations (>0.02) may originate from the fact that the clusters themselves were small in size with most galaxies clustered centrally near the core (see r distribution above) while the galaxy count algorithm averaged counts over numerous sites over the data set, some of which may not have clustering present.
- Ways of improving the experiment and scope for future development was also mentioned.

Background Image Credit: NASA/ESA/CSA/STScI

Maynooth University
National University of Ireland Maynooth

MU SUMMER PROGRAMME FOR UNDERGRADUATE RESEARCH SPUR

References:

1. Peebles, P.J.E., 2020. *The large-scale structure of the universe* (Vol. 98). Princeton university press.
2. Kerscher, M., Szapudi, I. and Szalay, A.S., 2000. A comparison of estimators for the two-point correlation function. *The astrophysical journal*, *535*(1), p.L13.
3. Basilakos, S. and Plionis, M., 2004. Modelling the two-point correlation function of galaxy clusters in the Sloan Digital Sky Survey. *Monthly Notices of the Royal Astronomical Society*, *349*(3), pp.882-888.

Acknowledgements:

I would like to thank Prof. Peter Coles for the opportunity to work in an area of physics I haven't had the pleasure of exploring yet. His insight and experience in the subject was priceless. I would also like to thank the Department of Theoretical Physics and the SPUR team for giving me this opportunity to work in research during my undergraduate degree.

Scan here to check out the source code on GitHub!

Github.com/pjjanas/janas-gcluster