MNRAS 000, 1–5 (2018) Preprint 31 May 2018 Compiled using MNRAS LATEX style file v3.0

Notes On Abacus BAO Analysis

Yutong Duan [1]★

Accepted XXX. Received YYY; in original form ZZZ

ABSTRACT

This serves as detailed notes on the procedures of BAO analysis with AbacusCosmos to accompany the code. Part 1 is on the calculation of correlation functions and covariance matrices. Part 2 is on the fitting methods of the BAO fitter.

Key words: keyword1 – keyword2 – keyword3

1 INTRODUCTION

Precision measurement of the BAO signal from galaxy correlation functions requires understanding the systematics. The standard approach is producing many mock catalogues of galaxies/quasars, test the analysis pipelines, and evaluate the sensitivity to systematics. The pipelines largely consist of two parts, statistics and fitting. The statistics of galaxy catalogues are correlation functions and covariance matrices. Then the statistics are fed to the fitting procedures, yielding best-fit α, the BAO scale parameter with respect to the fiducial model.

2 CORRELATION FUNCTIONS AND COVARIANCE MATRIX

2.1 Reading Halo Catalogue

Given a cosmology and a redshift, there are 16 boxes with varied phases. For each phase, load the halo catalogue produced by the Rockstar halo-finder. The virial mass and radius fields have inconsistent naming so add the following columns if they are not already present

$$\texttt{halo\_mvir} = \texttt{halo\_m} \tag{1}$$

$$\texttt{halo\_rvir} = \texttt{halo\_r} \tag{2}$$

to keep the Abacus catalogues compatible with halotools, where these fields are assumed to be present. The halo mass field `halo_mgrav` is preferred over `halo_mvir` in calculations because `halo_mvir` only works well for large halos and misbehaves for small halos as well as subhalos. For our analysis the distinction is not important though.

The NFW profile $\rho(r)$ is a mass distribution model for dark matter halos as a function of radius. The profile is completely characterised by `halo_mvir` and the the concentration $c_{\rm NFW} \equiv R_{\rm vir}/R_s$, where R_s is the scale radius of the halo (see halotools implementation for formulae). The Klypin definition of the halo scale radius $R_{\rm s}$ is considered more stable than the usual $R_{\rm s}$ for small halos. The NFW concentration field is added to halo table as well.

$$\texttt{halo\_nfw\_conc} = \frac{\texttt{halo\_rvir}}{\texttt{halo\_klypin\_rs}}. \tag{3}$$

2.2 Halotools Prebuilt HOD Models

Halotools treats all halos as spherical and uses NFW profiles to paint galaxies. Several mainstream HOD models are built in to halotools: `['zheng07', 'leauthaud11', 'tinker13', 'hearin15', 'zu_mandelbaum15', 'zu_mandelbaum16', 'cacciato09']`. We tune the model parameters to generate LRG-like samples. For a $(1100\,{\rm Mpc/h})^3$ simulation box, typical numbers are

$$N_{\rm halos} = 8 \times 10^6 \tag{4}$$

$$N_{\rm galaxies} = 5 \times 10^5. \tag{5}$$

The Zheng et al. (2007) model used in SDSS is an essential model with 5 parameters, $(M_{\rm cut}, \sigma, M_0, M_1, \alpha)$. The mean halo occupation numbers per halo for central and satellite galaxies are given by

$$\begin{aligned}\left\langle N_{\rm cen}(M)\right\rangle &= \frac{1}{2}\left[1 + {\rm erf}\left(\frac{\log M - \log M_{\rm cut}}{\sigma}\right)\right] \\ &= \frac{1}{2}{\rm erfc}\left(\frac{\log(M_{\rm cut}/M)}{\sigma}\right)\end{aligned} \tag{6}$$

$$\left\langle N_{\rm sat}(M)\right\rangle = \left\langle N_{\rm cen}(M)\right\rangle \left(\frac{M - M_0}{M_1}\right)^{\alpha}. \tag{7}$$

2.3 Generalised HOD Model

2.3.1 *Baseline Model (Zheng07 or White11)*

Since Abacus enables direct access to DM particles beyond standard halo metadata, a DM particle-based approach can be used to populate halos, without invoking NFW concentrations. Satellite galaxies are assigned to each DM particle within the halo with a certain probability, such that the resulting mean occupation is as specified by the HOD model. This avoids sphericalising halos and respects the morphology of halos when assigning synthetic galaxies. For the

★ E-mail: dyt@physics.bu.edu

first BOSS analyses White et al. (2011) used Zheng's formalism but with slightly different parametrisations,

$$\langle N_{\rm cen}(M)\rangle = \frac{1}{2}{\rm erfc}\left(\frac{\ln(M_{\rm cut}/M)}{\sqrt{2}\sigma'}\right) \tag{8}$$

$$\langle N_{\rm sat}(M)\rangle = \langle N_{\rm cen}(M)\rangle \left(\frac{M-\kappa M_{\rm cut}}{M_1}\right)^{\alpha}. \tag{9}$$

where the argument inside erf has an extra factor of $\ln 10/\sqrt{2}$, and M_0 is replaced with $\kappa M_{\rm min}$. The reason for the trivial change of parametrisation was that "our definition of σ can be interpreted as a fractional 'scatter' in mass at threshold". Conversion of SDSS/BOSS empirical values to the new parameters $(M_{\rm cut}, \sigma', \kappa, M_1, \alpha)$ can be done simply by

$$\sigma' = \frac{\ln 10}{\sqrt{2}}\sigma \tag{10}$$

$$\kappa = \frac{M_0}{M_{\rm min}} \tag{11}$$

with $M_{\rm cut}$, M_1 and α remaining the same.

The halo table and particle subsample table are loaded with appropriate mass cut applied and subhalos dropped. This selection cut help to remove small halos whose profiles are strongly affected by the force softening length, resulting in nonphysical halo profiles.When populating halo catalogues with galaxies, several observational effects are added to the truth so that the data appear realistic. The bias implementation in the generalised HOD model is similar to Yuan et al. (2018).

With 16 phase boxes, to further increase the signal-to-noise in the clustering statistics, 10-16 realisations of a given HOD is generated with different initial seeds for each box. Given a phase p and realisation r, the random number generator seed is chosen as

$$s = 100p + r \tag{12}$$

which guarantees that the random numbers only depend on the phase and realisation since we never go beyond 100 realisations. All different HOD models result in the same random numbers assigned to halo centres and subsample DM particles in a fixed order regardless of any HOD parameters.

2.3.2 RSD for Centrals and Satellites

Redshift-space distortion seen by an observer is just modifying the z-coordinate of the halo/galaxy, analytically

$$x'_z = x_z + \frac{v_z}{aH(a)} = x_z + \frac{v_z}{\frac{H(z)}{1+z}} = x_z + \frac{1+z}{H(z)}v_z \tag{13}$$

as derived in halotools documentation. This formula a very good approximation. However, halotools assumes $h = 1$ and $H_0 = 100h = 100\,{\rm km\,s^{-1}\,Mpc^{-1}}$, whereas our simulations are more realistic and were run with $h \approx 0.67$, so we are avoiding the Halotools implementation. There are two ways to manually add RSD, bypassing relevant halotools functionalities. One is to use Astropy's cosmology class, which comes as `halocat.cosmology` and includes the E function $E(z) = \int_0^z \frac{H_0}{H(z')}{\rm d}z'$ for the particular cosmology. With $H(z) \equiv H_0E(z)$, RSD can be implemented as

$$x'_z = x_z + \frac{1+z}{H_0E(z)}v_z \tag{14}$$

The other way is to use the conversion factor in halo catalogue header `halocat.header['VelZSpace_to_kms']`, defined as

$$\beta \equiv aH(z)L_{\rm box} = \frac{H(z)}{1+z}L_{\rm box} \tag{15}$$

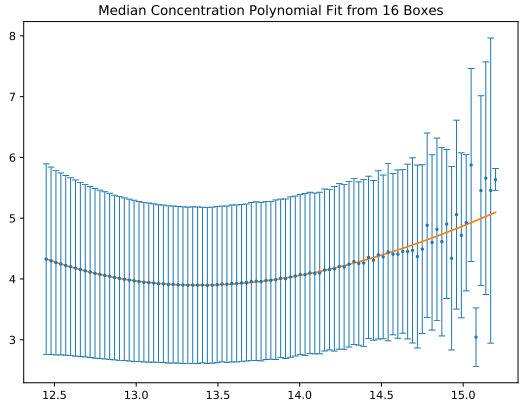

Figure 1. Median concentration as a function of log virial mass.

where β is in ${\rm km\,s^{-1}}$ and $L_{\rm box}$ in Mpc. RSD is then applied as

$$x'_z = x_z + \frac{L_{\rm box}}{\beta}v_z\,. \tag{16}$$

where x_z and v_z are in default catalogue units of Mpc and ${\rm km\,s^{-1}}$. These two methods are numerically equivalent up to 1 in 10^{15}.

2.3.3 Assembly Bias for Centrals (Ranking)

First all catalogues from 16 phases are loaded with mass cut $4\times10^{12}M_\odot$ and subhalos dropped, and for 100 log halo mass bins between the minimum and maximum log masses, the median NFW concentration is extracted. With 100 bin centre mvir values in unit of $h^{-1}M_\odot$ and 100 $c_{\rm median}$ values, a 3rd order polynomial fit is performed and looks like Fig. 1.

Instead of the usual weight $w = 1/\sigma(c)$ which yields very large weight for bins with very small number of halos and very small standard deviation in c, a correction is applied to explicitly weigh the number of halos in the bin

$$w = \frac{1}{\sigma(c)}\sqrt{N_{\rm halos}-1}. \tag{17}$$

Data are also masked for extremal cases where $N_{\rm halos}$ in a bin is only 0 or 1.

Now, with this polynomial fit, we may compare the $c_{\rm NFW}$ of any given halo with the $c_{\rm median}$ at that halo mass scale of the halo. Define pseudomass of centrals as

$$\log M_{\rm pseudo} = logM + A_{\rm cen}\left[2\Theta(c-c_{\rm med})-1\right] \tag{18}$$

so that when $c > c_{\rm med}$ there is a $+A_{\rm cen}$ correction, and when $c < c_{\rm med}$, $-A_{\rm cen}$.

Next the masses and pseudomass are sorted and re-assigned to halos: the halo with highest pseudomass (rank 0 halo) gets the highest actual mass, and so forth. We have not modified any halo mass values, just re-assigned them by pseudomasses. The new halo masses are used as input for HOD models and calculating theoretical mean occupation $\langle N_{\rm cen}\rangle$, and never written to the original halo table.

2.3.4 Velocity Bias for Centrals

The velocity bias for central galaxies is added by randomly drawing the peculiar velocity from a normal distribution corresponding to

the RMS dispersion within the halo,

$$v_{\rm pec} \sim N(0, \frac{v_{\rm rms}}{\sqrt{3}}\alpha_c) \tag{19}$$

$$v'_{\rm los} = v_{\rm los} + v_{\rm pec} \tag{20}$$

assuming $v_{\rm rms} = \sqrt{v_x^2 + v_y^2 + v_z^2}$ and $v_{\rm los} = v_z$.

2.3.5 *Assembly Bias for Satellites (Ranking)*

The assembly bias for satellites is completely independent from that for centrals, and either can be turned on or off. The original halo masses are read from halo table again, and re-assigned using pseudomass

$$\log M_{\rm pseudo} = logM + A_{\rm sat}\left[2\Theta(c - c_{\rm med}) - 1\right] \tag{21}$$

and the HOD models take new masses as input and produce $\langle N_{\rm sat}\rangle(M)$. No particle property is involved here and $\overline{p} = \langle N_{\rm sat}\rangle / N_{\rm part}$ is the same for all particles in the same halo.

2.3.6 *Host Centric Distance Ranking for Satellites*

The probability of a particle hosting a satellite straight from an HOD model $\overline{p} = \langle N_{\rm sat}\rangle / N_{\rm part}$ is modified by several ranking procedures. The first is ranking by host centric distance, i.e. the distance between the particle and halo centre. The furthest particle gets rank $r_i = 0$, and so forth. With a constant modulation parameter s specified, the probability is modified as

$$p_i = \overline{p}\left[1 + s(1 - \frac{2r_i}{N_{\rm part} - 1})\right]. \tag{22}$$

2.3.7 *Velocity Bias for Satellites (Ranking)*

The second particle ranking procedure adds velocity bias for satellites. The peculiar speed of each halo particle is calculated, and within each halo, the particles are ranked by their peculiar speed: particle with highest peculiar speed has rank $r_i = 0$. The probability then becomes

$$p'_i = p_i\left[1 + s_v(1 - \frac{2r_i}{N_{\rm part} - 1})\right]. \tag{23}$$

2.3.8 *Perihelion Distance Ranking for Satellites*

The last particle ranking procedure ranks particles by their distance of closest approach to halo centre. The perihelion distance $r_{\rm min}$ is numerically approximated with an iterative approach (Yuan et al. (2018)), which takes 10 iterations for average fractional error to reach 10% and 30 iterations for the maximum fractional error to drop below 10%. Particle with the largest $r_{\rm min}$ gets rank $r_i = 0$. Then again

$$p'_i = p_i\left[1 + s_P(1 - \frac{2r_i}{N_{\rm part} - 1})\right]. \tag{24}$$

Finally, each particle gets a random number between 0 and 1, and those below the theoretical p_i get satellites. Even when the HOD models give negative probabilities, this will not misbehave. For each realisation of each HOD model in each phase box, the galaxy table is saved as in astropy ASCII csv format.

2.4 Galaxy Pair Counting

All counting is done in fine (s, μ) bins: s bin edges are from 0 to 150 h^{-1}Mpc at 1 h^{-1}Mpc steps, and $\mu \equiv \cos\theta$ bin edges are from 0 to 1 at 0.01 steps, meaning a total of (150, 100) bins.

We need the auto-counts for each box to calculate the auto-correlation for the whole box, and the cross-counts between the box and a subvolume of itself to calculate cross-correlation and estimate the covariance matrix. All these counts are saved to disk. Raw counts are saved as `paircount-DD.npy` in the original Currfunc count format, i.e. a structured array. Note that `c_api_timer` needs to be turned off in Corrfunc for this file to be saved and recovered properly and not as "object" type.

The optimal s bin size and range for fitting were studied in BOSS DR12 analysis Ross et al. (2017). Based on these results, we choose the bin size 5 h^{-1}Mpc and the range 50 h^{-1}Mpc $< s <$ 150 h^{-1}Mpc. Also we use a μ bin size of 0.05 to be consistent with BOSS DR12. The raw counts were re-binned into (30, 20) coarse bins and total pair counts verified.

2.5 Auto-correlation Functions

The PH natural estimator of the auto-correlation requires DD and RR counts for each phase box. DD is calculated using the re-binned DD counts for (30, 20) bins. RR pair counts are calculated for the same bins analytically as given in Appendix A.

The auto-correlation function ξ is decomposed into the monopole component ξ_0 for $\ell = 0$ and the quadrupole component ξ_2 for $\ell = 2$ in the Legendre spherical harmonics basis using

$$\xi(s, \mu) = \sum_\ell \xi_\ell(s) P_\ell(\mu) \tag{25}$$

$$\xi_\ell(s) = \frac{2\ell + 1}{2}\int_{-1}^{1} \xi(s, \mu) P_\ell(\mu)\, {\rm d}\mu \tag{26}$$

as shown in Fig. 2. 16 ξ samples from 16 realisations are averaged and become a single auto-correlation result for the box. This averaging step can be done before or after multipole decomposition which is a linear operation.

Then we jackknife co-add the 16 boxes, dropping one box at a time and taking the mean of the other 15. This yields 16 jackknifed correlation function samples, and each one is passed on to the fitter for BAO fitting.

2.6 Covariance Matrix

The BAO fitter takes correlation functions and covariance matrix as input. The covariance is empirically estimated. Each simulation box is divided into $N_{\rm sub}^3 = 3^3$ subvolumes. The cross-correlation function between one subvolume and the entire box is calculated for all 16 phases, resulting in $27 \times 16 = 432$ correlation function samples if there is only 1 realisation. For 10 realisations there would be 4320 samples. With (30, 20) total (s, μ) bins for $\xi_0(s, \mu)$ and $\xi_2(s, \mu)$, there are $30{\times}20{\times}2 = 1200$ bins, monopole and quadrupole included. The covariance between every two (s, μ) bins is calculated from 432 samples, including the covariance between monopole and quadrupole. Thus resulting covariance matrix has shape 1200×1200.

Finally this covariance matrix needs to be rescaled to account for our volume and averaging tricks. The covariance scales inversely with volume, so the covariance derived from the subvolume (1/27 of the box's volume) is 27 times the actual. In addition, if we run 10 realisations and average over all of them, there is another factor

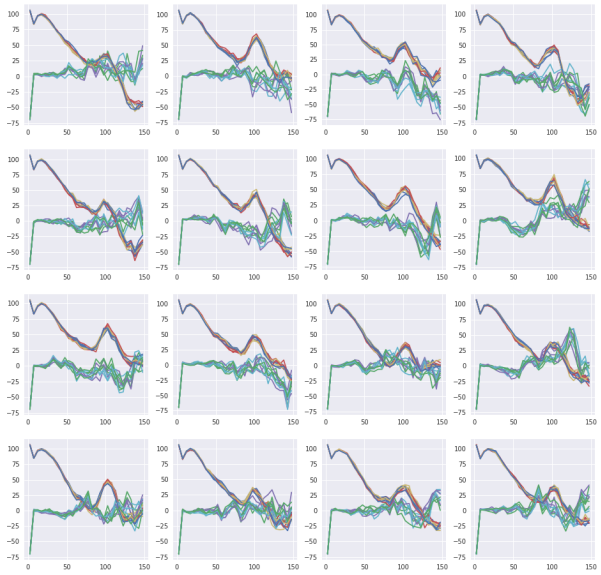

Figure 2. Monopole and quadrupole of auto-correlation functions for 16 phase boxes. Each box has 16 realisations overplotted, which are averaged to give one result.

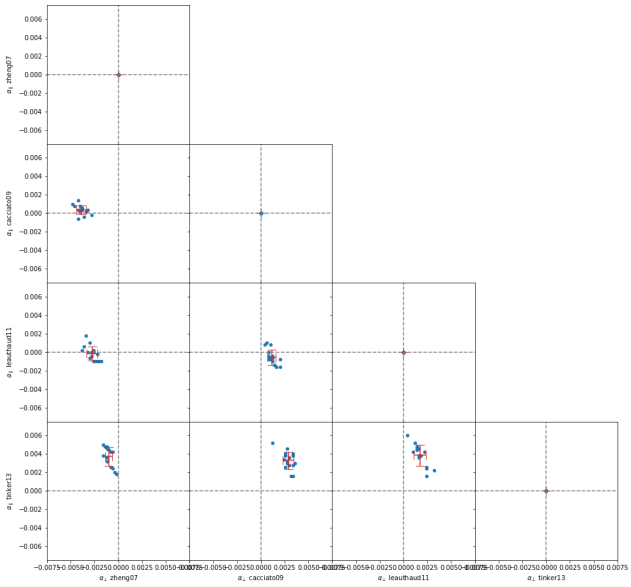

Figure 3. Comparison of $(\alpha_{/\!/}, \alpha_\perp)$ for four HOD models done in halotools. vertical axis is $\alpha_{/\!/}$, horizontal $\alpha_\perp$.

of 10. Also due to jackknife, each sample is an average of 15 boxes, thus a factor of 15. Therefore the true covariance matrix is the one obtained from subvolume divided by $(27 \times 15 \times 10)$.

3 BAO FITTER

A recent, functional fitter is one by Ross https://github.com/ashleyjross/LSSanalysis used for BOSS DR12 analysis. The fitter takes correlation function ξ_0, ξ_2 in all (s, μ) bins and the covariance between all bins as input, and evaluates χ^2 on the $(\alpha_{/\!/}, \alpha_\perp)$ parameter grid.

The log likelihood ratio follows a χ^2 distribution. In the $(\alpha_{/\!/}, \alpha_\perp)$ two-parameter plane, we may find the confidence region for any confidence level σ by calculating the constant $\Delta\chi^2$ contour. The probability corresponding to confidence level σ is $P = \mathrm{Erf}(\sigma/\sqrt{(2)})$, and the contour can be found by

$$\Delta\chi^2 = Q(P) \tag{27}$$

where Q is the quantile function (percent-point function) for a χ^2 distribution with dof = 2.

With 16 jackknife ξ samples and a covariance matrix for each HOD model, we obtain 16 $(\alpha_{/\!/}, \alpha_\perp)$ values by finding the global χ^2 minimum. Then for every pair of HOD models M_i and M_j, we calculate 16 phase-matched differences $\alpha_{ij} = \alpha_i - \alpha_j$, plotted in Fig. 3.

We will rewrite the fitter in a modern style in Python conforming to PEP standards and optimising for performance.

Derivation of relevant equations. In fiducial configuration space,

$$r^2 = r_{/\!/}^2 + r_\perp^2 \tag{28}$$

$$\mu^2 = \cos^2\theta = \frac{r_{/\!/}^2}{r_\perp^2}. \tag{29}$$

Define spherically averaged distance $D_V(z)$ and an anisotropy parameter $F_{\rm AP}(z)$,

$$D_V(z) = \left(D_A^2(z)\frac{cz}{H(z)}\right)^{1/3} \tag{30}$$

$$F_{\rm AP}(z) = D_A(z)\frac{H(z)}{c}. \tag{31}$$

One parametrisation is

$$\alpha_\perp = \frac{D_A(z) r_s^{\rm fid}}{D_Z^{\rm fid} r_s} \tag{32}$$

$$\alpha_{/\!/} = \frac{H^{\rm fid}(z) r_s^{\rm fid}}{H(z) r_s} \tag{33}$$

and an alternative parametrisation is

$$\alpha = \frac{D_V(z) r_s^{\rm fid}}{D_V^{\rm fid}(z) r_s} = \left[\frac{D_A^2(z) H^{\rm fid}(z)}{D_A^{\rm fid\,2}(z) H(z)}\right]^{1/3} \frac{r_s^{\rm fid}}{r_s} \tag{34}$$

$$1 + \epsilon = \left[\frac{H^{\rm fid}(z) D_A^{\rm fid}(z)}{H(z) D_A(z)}\right]^{1/3}. \tag{35}$$

The conversion between $(\alpha_{/\!/}, \alpha_\perp)$ and (α, ϵ) is

$$\alpha_{/\!/} = \alpha(1+\epsilon)^2 \tag{36}$$

$$\alpha_\perp = \alpha(1+\epsilon)^{-1} \tag{37}$$

$$\alpha = \alpha_{/\!/}^{1/3}\alpha_\perp^{2/3} \tag{38}$$

$$1 + \epsilon = \left(\frac{\alpha_{/\!/}}{\alpha_\perp}\right)^{1/3} \tag{39}$$

REFERENCES

Ross A. J., et al., 2017, MNRAS, 464, 1168

Wang Y., Yang X., Mo H. J., van den Bosch F. C., Weinmann S. M., Chu Y., 2008, ApJ, 687, 919
White M., et al., 2011, ApJ, 728, 126
Yuan S., Eisenstein D. J., Garrison L. H., 2018, MNRAS, p. 1043
Zheng Z., Coil A. L., Zehavi I., 2007, ApJ, 667, 760

APPENDIX A: AUTO-CORRELATION FUNCTION ESTIMATOR

In this implementation, raw pair counts are saved, and $N_{\rm D}, N_{\rm R}$ normalisation is done only at the estimator step.

For auto-correlation of a sample in periodic box, the Peebles & Hauser (1974) estimator (natural estimator) is

$$\xi = \frac{N_{\rm R}(N_{\rm R}-1)}{N_{\rm D}(N_{\rm D}-1)}\frac{DD}{RR} - 1 \tag{A1}$$

where DD is the auto-correlation data-data pair count and RR is the random-random pair count.

One way to obtain RR is to generate a random sample of particle number $N_{\rm R}$ in the same volume and calculate its auto-correlation pair counts. Alternatively, RR can be calculated analytically as follows. Let $\mathrm{d}V$ be the volume of the (s,μ) bin and $V = L_{\rm box}^3$ be the volume the data sample occupies. The random sample must have the same number density as the data sample, $n_{\rm R} = n_{\rm D}$. For simple survey geometry, such as a cube, we may well let the random sample have the same number count, volume, and number density as the data sample.

$$RR(s,\mu) = \frac{N_{\rm R}(N_{\rm R}-1)}{V}\,\mathrm{d}V(s,\mu) \tag{A2}$$

where $\mathrm{d}V(s,\mu)$ is the volume of the (s,μ) bin.

APPENDIX B: CROSS-CORRELATION FUNCTION ESTIMATOR

For cross-correlation between a sample D_1 in the periodic simulation box and a subsample D_2 in its subvolume, the Davis & Peebles (1983) estimator is

$$\xi = \frac{\bar{n}_{\rm R1}}{\bar{n}_{\rm D1}}\frac{D_1D_2}{R_1D_2} - 1 \tag{B1}$$

where $\bar{n}_{\rm D1}$ is the mean number density of data sample D_1, $\bar{n}_{\rm R1}$ is the mean number density of R_1, the random sample corresponding to D_1, D_1D_2 is the cross-correlation pair count between two data samples, and R_1D_2 is the cross-correlation pair count between a random and a data sample. Usually a random sample is about 7 times the size of the corresponding data sample Wang et al. (2008).

Again one may generate a random sample R_1 in the same volume and do the cross counting, or alternatively calculate it analytically,

$$R_1D_2 = \frac{N_{\rm R1}N_{\rm D2}}{V_1}\,\mathrm{d}V(s,\mu) = \bar{n}_{\rm R1}N_{\rm D2}\,\mathrm{d}V(s,\mu). \tag{B2}$$

This paper has been typeset from a TEX/LATEX file prepared by the author.