Discussion on Bispectrum measurements & "precise" modelling

Héctor Gil-Marín

Laboratoire de Physique Nucleaire et de Hautes Energies
Institut Lagrange de Paris

CosKASI-ICG-NAOC-YITP Workshop
8th September 2016

Outline

1. Bispectrum estimator
2. Modelling the dark matter bispectrum in real space
3. Modelling the dark matter bispectrum in redshift space
4. Other issues on modelling

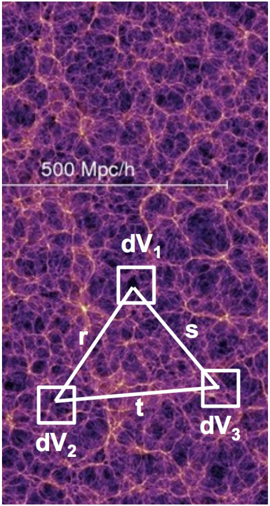
Estimation of the Bispectrum

Efficient algorithm for computing the Bispectrum

- Sefusatti E., 2005, PhD thesis, New York University, New York, USA (no public access)
- Baldauf et al. 2015 (arXiv: 1406.4135)

Based on Fourier Transforms and not in previous brute force approaches.

Definition of the Bispectrum



Probability of finding 3 galaxies separated by r, s and t: $P_3(r,s,t) =$ $[1+\xi_2(r)+\xi_2(s)+\xi_2(t)+\zeta(r,s,t)]dV_1\, dV_2\, dV_3$
The bispectrum is defined as the FT of ζ,

$$B(\mathbf{k}_1,\mathbf{k}_2) \equiv \int d\mathbf{r}\, d\mathbf{s}\, \zeta(\mathbf{r},\mathbf{s}) e^{-i\mathbf{r}\cdot\mathbf{k}_1} e^{-i\mathbf{s}\cdot\mathbf{k}_2}$$

Since, $\zeta(r,s,t) \equiv \langle \delta(\mathbf{x}+\mathbf{r})\delta(\mathbf{x}+\mathbf{t})\delta(\mathbf{x})\rangle_{\mathbf{x}}$

$$B(k_1,k_2,k_3) = \langle \delta(\mathbf{k}_1)\delta(\mathbf{k}_2)\delta(\mathbf{k}_3)\rangle \delta^D(\mathbf{k}_1+\mathbf{k}_2+\mathbf{k}_3)$$

Algorithm

```
printf("Computing the Bispectrum.
/////------------------------------
//available
//deltak_re
//deltak_im

static fftw_complex *in_bis1;    /
static fftw_complex *in_bis2;    /
static fftw_complex *in_bis3;    /

static fftw_complex *in_bis1_NT;
static fftw_complex *in_bis2_NT;
static fftw_complex *in_bis3_NT;


double K_eff,K_eff1,K_eff2,K_eff3
int N_eff,N_eff1,N_eff2,N_eff3;
long int *L = (long int*)malloc((r
long int *IJK = (long int*)malloc(
long int line_test;
double B,P,Number_triangles;
double *Power = (double*)malloc((r
double *Number= (double*)malloc((r

int new_ngrid=ngrid*ngrid*ngrid;//
double ijk;
double k1_min,k2_min,k3_min;
//sprintf(name,"/mnt/lustre/hecto
d);
sprintf(name,"/mnt/lustre/hectorg
```

Proposed estimator,

$$\hat{B}_{123} = \frac{k_f^3}{V_{123}} \int_{\mathcal{S}_1} d\mathbf{q}_1\, \delta(\mathbf{q}_1) \int_{\mathcal{S}_2} d\mathbf{q}_2\, \delta(\mathbf{q}_2)$$

$$\times \quad \int_{\mathcal{S}_3} d\mathbf{q}_3\, \delta(\mathbf{q}_3) \delta^D(\mathbf{q}_1 + \mathbf{q}_2 + \mathbf{q}_3),$$

$\mathcal{S}_i$ stands for $\mathcal{S}(k_i|\Delta k)$, which is the k-region contained by $k_i - \Delta k/2 \leq k \leq k_i + \Delta k/2$, given a k-bin, Δk

Algorithm

```
printf("Computing the Bispectrum.
/////------------------------------
//available
//deltak_re
//deltak_im

static fftw_complex *in_bis1;    /
static fftw_complex *in_bis2;    /
static fftw_complex *in_bis3;    /

static fftw_complex *in_bis1_NT;
static fftw_complex *in_bis2_NT;
static fftw_complex *in_bis3_NT;


double K_eff,K_eff1,K_eff2,K_eff3
int N_eff,N_eff1,N_eff2,N_eff3;
long int *L = (long int*)malloc((r
long int *IJK = (long int*)malloc(
long int line_test;
double B,P,Number_triangles;
double *Power = (double*)malloc((r
double *Number= (double*)malloc((r

int new_ngrid=ngrid*ngrid*ngrid;//
double ijk;
double k1_min,k2_min,k3_min;
//sprintf(name,"/mnt/lustre/hector
d);
sprintf(name,"/mnt/lustre/hectorg
```

Proposed estimator,

$$\hat{B}_{123} = \frac{k_f^3}{V_{123}} \int_{\mathcal{S}_1} d\mathbf{q}_1\, \delta(\mathbf{q}_1) \int_{\mathcal{S}_2} d\mathbf{q}_2\, \delta(\mathbf{q}_2)$$
$$\times \quad \int_{\mathcal{S}_3} d\mathbf{q}_3\, \delta(\mathbf{q}_3) \delta^D(\mathbf{q}_1 + \mathbf{q}_2 + \mathbf{q}_3),$$

V_{123} is the associated volume ("total number of triangles"),

$$V_{123} = \frac{k_f^3}{V_{123}} \int_{\mathcal{S}_1} d\mathbf{q}_1 \int_{\mathcal{S}_2} d\mathbf{q}_2$$
$$\times \quad \int_{\mathcal{S}_3} d\mathbf{q}_3\, \delta^D(\mathbf{q}_1 + \mathbf{q}_2 + \mathbf{q}_3),$$

```
printf("Computing the Bispectrum.
/////------------------------------
//available
//deltak_re
//deltak_im

static fftw_complex *in_bis1;    /
static fftw_complex *in_bis2;    /
static fftw_complex *in_bis3;    /

static fftw_complex *in_bis1_NT;
static fftw_complex *in_bis2_NT;
static fftw_complex *in_bis3_NT;


double K_eff,K_eff1,K_eff2,K_eff3
int N_eff,N_eff1,N_eff2,N_eff3;
long int *L = (long int*)malloc((n
long int *IJK = (long int*)malloc(
long int line_test;
double B,P,Number_triangles;
double *Power = (double*)malloc((n
double *Number= (double*)malloc((n

int new_ngrid=ngrid*ngrid*ngrid;//
double ijk;
double k1_min,k2_min,k3_min;
//sprintf(name,"/mnt/lustre/hector
d);
sprintf(name,"/mnt/lustre/hectorgi
```

We can write the Dirac delta in its exponential form,
$\delta^D(\mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3) \to \int d\mathbf{r}\, e^{i\mathbf{r}\cdot\mathbf{q}_1} e^{i\mathbf{r}\cdot\mathbf{q}_2} e^{i\mathbf{r}\cdot\mathbf{q}_3}$
and re-write the estimator as,

$$\hat{B}_{123} \propto \int_{\mathcal{S}_1} d\mathbf{q}_1\, \delta(\mathbf{q}_1) \int_{\mathcal{S}_2} d\mathbf{q}_2\, \delta(\mathbf{q}_2)$$
$$\times \quad \int_{\mathcal{S}_3} d\mathbf{q}_3\, \delta(\mathbf{q}_3) \underbrace{\int d\mathbf{r}\, e^{i\mathbf{r}\cdot\mathbf{q}_1} e^{i\mathbf{r}\cdot\mathbf{q}_2} e^{i\mathbf{r}\cdot\mathbf{q}_3}}_{\delta^D(\mathbf{q}_1+\mathbf{q}_2+\mathbf{q}_3)}$$

Then, the estimator is partly separable,

$$\hat{B}_{123} \propto \int d\mathbf{r} \underbrace{\int_{\mathcal{S}_1} d\mathbf{q}_1 \delta(\mathbf{q}_1) e^{i\mathbf{r}\cdot\mathbf{q}_1}}_{\text{FT-like of } \delta(\mathbf{q}_1)} \ldots \{q_2\} \ldots \{q_3\}$$

Algorithm

```
printf("Computing the Bispectrum.
/////------------------------------
//available
//deltak_re
//deltak_im

static fftw_complex *in_bis1;     /
static fftw_complex *in_bis2;     /
static fftw_complex *in_bis3;     /

static fftw_complex *in_bis1_NT;
static fftw_complex *in_bis2_NT;
static fftw_complex *in_bis3_NT;


double K_eff,K_eff1,K_eff2,K_eff3
int N_eff,N_eff1,N_eff2,N_eff3;
long int *L = (long int*)malloc((r
long int *IJK = (long int*)malloc(
long int line_test;
double B,P,Number_triangles;
double *Power = (double*)malloc((r
double *Number= (double*)malloc((r

int new_ngrid=ngrid*ngrid*ngrid;//
double ijk;
double k1_min,k2_min,k3_min;
//sprintf(name,"/mnt/lustre/hecto
d);
sprintf(name,"/mnt/lustre/hectorg
```

In the end the algorithm looks like,

$$\hat{B}_{123} \propto \int d\mathbf{r}\, \mathcal{D}_{\mathcal{S}_1}(\mathbf{r})\mathcal{D}_{\mathcal{S}_2}(\mathbf{r})\mathcal{D}_{\mathcal{S}_3}(\mathbf{r})$$

where,

$$\mathcal{D}_{\mathcal{S}_j}(\mathbf{r}) \equiv \int_{\mathcal{S}_j} d\mathbf{q}_j\, \delta(\mathbf{q}_j) e^{i\mathbf{q}_j\cdot\mathbf{r}}.$$

$\mathcal{D}_{\mathcal{S}_j}$ can be though as the Fourier Transform of $\mathcal{F}(\mathbf{q})$ where

$$\mathcal{F}(\mathbf{q}) = \begin{cases} \delta(\mathbf{q}) & \text{if } \mathbf{q} \in \mathcal{S}_j \\ 0 & \text{otherwise} \end{cases}$$

The main complication when coding is to find an efficient way to select the elements of $\delta(\mathbf{q}_i) \in \mathcal{S}_i$.

Algorithm

```
printf("Computing the Bispectrum.
/////-----------------------------
//available
//deltak_re
//deltak_im

static fftw_complex *in_bis1;
static fftw_complex *in_bis2;
static fftw_complex *in_bis3;

static fftw_complex *in_bis1_NT;
static fftw_complex *in_bis2_NT;
static fftw_complex *in_bis3_NT;


double K_eff,K_eff1,K_eff2,K_eff3
int N_eff,N_eff1,N_eff2,N_eff3;
long int *L = (long int*)malloc((
long int *IJK = (long int*)malloc(
long int line_test;
double B,P,Number_triangles;
double *Power = (double*)malloc((
double *Number= (double*)malloc((

int new_ngrid=ngrid*ngrid*ngrid;/
double ijk;
double k1_min,k2_min,k3_min;
//sprintf(name,"/mnt/lustre/hecto
d);
sprintf(name,"/mnt/lustre/hectorg
```

Features,

- **Fast**. $\sim$ 850 triangles in $\sim$ 2 hours using 20 processors. Speed can be tuned by increasing the size of k-bins (i.e. reducing the number of total triangles).
- **Easy-to-code**. It relies on FT libraries such as FFTW, which are optimised and widely used.
- **General**. No limitation on scales or survey geometry other than the power spectrum estimator

Algorithm

```
printf("Computing the Bispectrum.
/////-----------------------------
//available
//deltak_re
//deltak_im

static fftw_complex *in_bis1;    /
static fftw_complex *in_bis2;    /
static fftw_complex *in_bis3;    /

static fftw_complex *in_bis1_NT;
static fftw_complex *in_bis2_NT;
static fftw_complex *in_bis3_NT;


double K_eff,K_eff1,K_eff2,K_eff3
int N_eff,N_eff1,N_eff2,N_eff3;
long int *L = (long int*)malloc((
long int *IJK = (long int*)malloc(
long int line_test;
double B,P,Number_triangles;
double *Power = (double*)malloc((
double *Number= (double*)malloc((

int new_ngrid=ngrid*ngrid*ngrid;/
double ijk;
double k1_min,k2_min,k3_min;
//sprintf(name,"/mnt/lustre/hecto
d);
sprintf(name,"/mnt/lustre/hectorg
```

- **Recyclable**. No need to recompute the 3 $\mathcal{D}$ new terms for each new triangle, as many triangles share one or two sides.
- **Higher orders**. The algorithm can be trivially generalised to higher order functions, such as the Trispectrum.
- **Publicity**. Might be a release of a public code in the near(?) future.

Modelling the Bispectrum in real space

Different models (full shape),

- Perturbation Theory: SPT, RPT, EFT, . . .
- Phenomenological: Halo model PT, Benchmark model, 2-halo boost, Fitting formulae

Having a working model for the bispectrum is much more complicated than the power spectrum..., but for BAO-feature only may not be that difficult.

Modelling the Bispectrum in real space

Tree level approach based on Perturbation Theory,

$$B(\mathbf{k}_1, \mathbf{k}_2) = 2P_{\rm lin}(k_1)P_{\rm lin}(k_2)F_2(\mathbf{k}_1, \mathbf{k}_2) + \mathrm{cyc.},$$

where F_2 is the two-point kernel of the density of the form,

$$F_2(\mathbf{k}_i, \mathbf{k}_j) = \frac{5}{7} + \frac{1}{2}\cos(\alpha_{ij})\left(\frac{k_i}{k_j} + \frac{k_j}{k_i}\right) + \frac{2}{7}\cos^2(\alpha_{ij})$$

This kernel has a weak dependence with the cosmological parameters but may be sensible to modifications of GR.
Here we follow the empirical approach of taken the tree level formula and modify the kernels (Scoccimarro & Couchman 2001, HG-M et al. 2012):

$$F_2 \to F_2^{\rm eff}$$

$$P_{\rm lin} \to P_{\rm nl}$$

Modelling the Bispectrum in real space

$$
\begin{aligned}
F_2^{\mathrm{eff}}(\mathbf{k}_i, \mathbf{k}_j) &= \frac{5}{7} a(n_i, k_i, \mathbf{a}^F) a(n_j, k_j, \mathbf{a}^F) \\
&+ \frac{1}{2} \cos(\alpha_{ij}) \left(\frac{k_i}{k_j} + \frac{k_j}{k_i} \right) b(n_i, k_i, \mathbf{a}^F) b(n_j, k_j, \mathbf{a}^F) \\
&+ \frac{2}{7} \cos^2(\alpha_{ij}) c(n_i, k_i, \mathbf{a}^F) c(n_j, k_j, \mathbf{a}^F)
\end{aligned}
$$

where a, b, c are empirical functions that depend on k_i, on the local slope of the power spectrum, n, and on a set of 9 free parameters, $\mathbf{a}^F \equiv \{a_1^F, \ldots, a_9^F\}$ to be fitted from N-body simulations.

Modelling the Bispectrum in real space

Comparing different models with Nbody...

Matter bispectrum of large-scale structure: Three-dimensional comparison between theoretical models and numerical simulations

Andrei Lazanu,[1,*] Tommaso Giannantonio,[2,1,†] Marcel Schmittfull,[3,‡] and E.P.S. Shellard[1,§]

[1]*Centre for Theoretical Cosmology, DAMTP, University of Cambridge, CB3 0WA, United Kingdom*
[2]*Kavli Institute for Cosmology Cambridge, Institute of Astronomy, University of Cambridge, Madingley Road, Cambridge CB3 0HA, United Kingdom*
[3]*Berkeley Center for Cosmological Physics, Department of Physics and Lawrence Berkeley National Laboratory, University of California, Berkeley, California 94720, USA*

(Dated: April 26, 2016)

We study the matter bispectrum of the large-scale structure by comparing different perturbative and phenomenological models with measurements from N-body simulations obtained with a modal bispectrum estimator. Using shape and amplitude correlators, we directly compare simulated data with theoretical models over the full three-dimensional domain of the bispectrum, for different redshifts and scales. We review and investigate the main perturbative methods in the literature that predict the one-loop bispectrum: standard perturbation theory, effective field theory, resummed Lagrangian and renormalised perturbation theory, calculating the latter also at two loops for some

Lanzau et al. 2016; arXiv:1510.04075

Modelling the Bispectrum in real space: PT

Tree$\rightarrow B^{\rm SPT}_{tree} = 2P_{\rm lin}P_{lin}F_2 + cyc.$

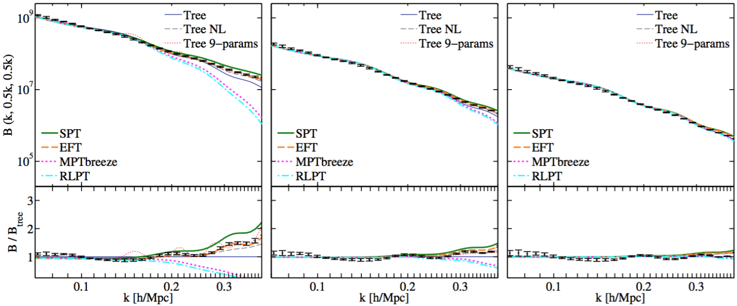
SPT$\rightarrow B^{\rm SPT}_{\rm 1-loop} = B^{\rm SPT}_{tree} + B_{222} + B^{(I)}_{321} + B^{(II)}_{321} + B_{411}$

EFT$\rightarrow B^{\rm EFT} = B^{\rm SPT} + B_{c_S}$ [no extra free parameters wrt $P^{\rm EFT}$]

MPTBreeze$\rightarrow B = [B^{\rm SPT}_{tree} + B_{222} + B^{(I)}_{321}]\exp[f(k_1)+f(k_2)+f(k_3)]$

RLPT$\rightarrow B =$

$$\exp\left[\frac{k_1^2+k_2^2+k_3^2}{12\pi^2}\int dp\, P_{\rm lin}(p)\right]\left[B^{\rm SPT\,1loop}_{\rm tree} + \frac{k_1^2+k_2^2+k_3^3}{12\pi^2}B^{\rm SPT}_{\rm tree}\int dp\, P_{\rm lin}(p)\right]$$



Modelling the Bispectrum in real space: PT

Tree$\rightarrow B^{\rm SPT}_{tree} = 2P_{\rm lin}P_{lin}F_2 + cyc.$

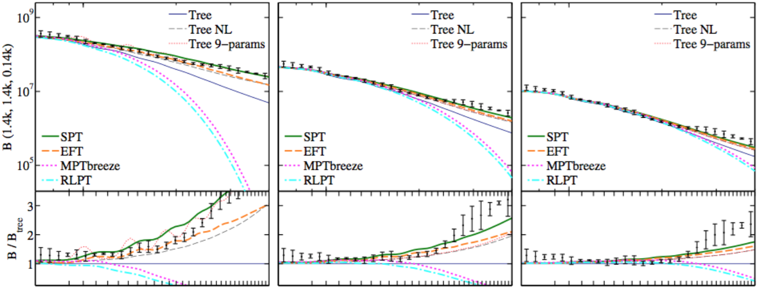
SPT$\rightarrow B^{\rm SPT}_{\rm 1-loop} = B^{\rm SPT}_{tree} + B_{222} + B^{(I)}_{321} + B^{(II)}_{321} + B_{411}$

EFT$\rightarrow B^{\rm EFT} = B^{\rm SPT} + B_{c_S}$ [no extra free parameters wrt $P^{\rm EFT}$]

MPTBreeze$\rightarrow B = [B^{\rm SPT}_{tree} + B_{222} + B^{(I)}_{321}]\exp[f(k_1)+f(k_2)+f(k_3)]$

RLPT$\rightarrow B =$

$$\exp\left[\frac{k_1^2+k_2^2+k_3^2}{12\pi^2}\int dp\, P_{\rm lin}(p)\right]\left[B_{\rm tree}^{\rm SPT_{1loop}} + \frac{k_1^2+k_2^2+k_3^3}{12\pi^2}B^{\rm SPT}_{\rm tree}\int dp\, P_{\rm lin}(p)\right]$$



Modelling the Bispectrum in real space: PT

Tree$\rightarrow B^{\rm SPT}_{tree} = 2P_{\rm lin}P_{lin}F_2 + cyc.$

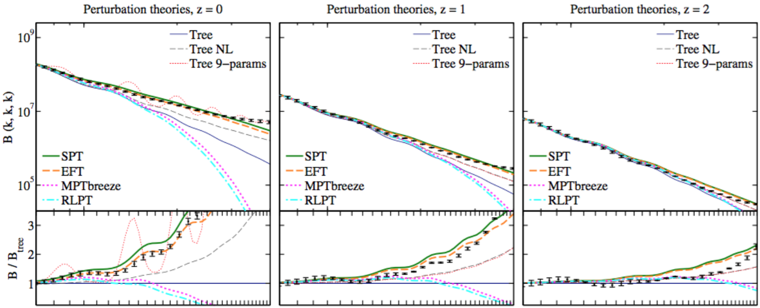
SPT$\rightarrow B^{\rm SPT}_{\rm 1-loop} = B^{\rm SPT}_{tree} + B_{222} + B^{(I)}_{321} + B^{(II)}_{321} + B_{411}$

EFT$\rightarrow B^{\rm EFT} = B^{\rm SPT} + B_{c_S}$ [no extra free parameters wrt $P^{\rm EFT}$]

MPTBreeze$\rightarrow B = [B^{\rm SPT}_{tree} + B_{222} + B^{(I)}_{321}]\exp[f(k_1)+f(k_2)+f(k_3)]$

RLPT$\rightarrow B =$

$$\exp\left[\frac{k_1^2+k_2^2+k_3^2}{12\pi^2}\int dp\, P_{\rm lin}(p)\right]\left[B^{\rm SPT_{1loop}}_{\rm tree} + \frac{k_1^2+k_2^2+k_3^3}{12\pi^2}B^{\rm SPT}_{\rm tree}\int dp\, P_{\rm lin}(p)\right]$$

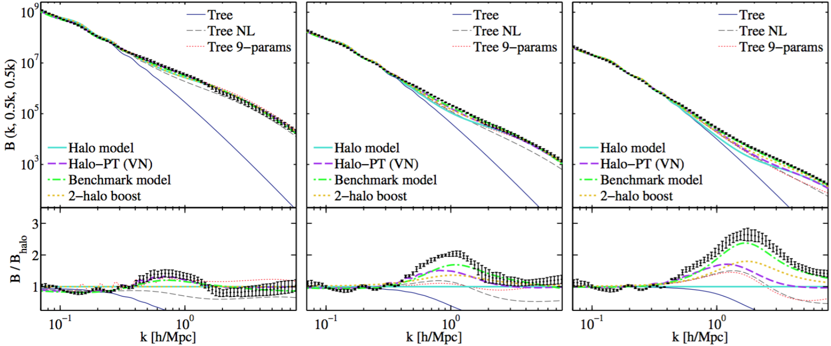


Halo Model$\rightarrow B^{\mathrm{HM}} = B_{1h} + B_{2h} + B_{3h}$

Halo Model PT$\rightarrow$ Valageas and Nishimichi 2011a, 2011b

Benchmark model$\rightarrow B = f(K) S^{\mathrm{SPT}}_{tree}$; S is shape function

2-halo boost.

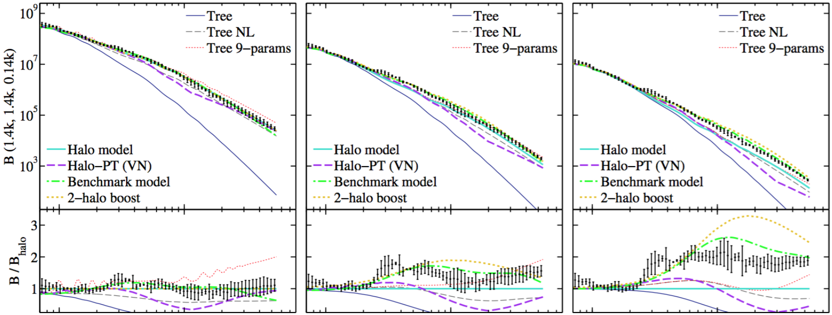


Halo Model$\rightarrow B^{\mathrm{HM}} = B_{1h} + B_{2h} + B_{3h}$

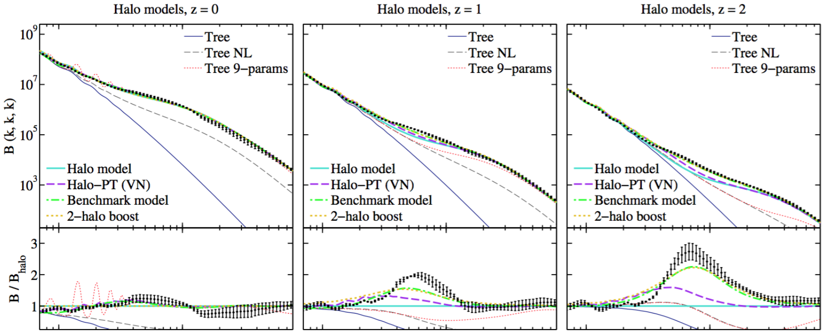
Halo Model PT$\rightarrow$ Valageas and Nishimichi 2011a, 2011b

Benchmark model$\rightarrow B = f(K) S^{\mathrm{SPT}}_{tree}$; S is shape function

2-halo boost.



Halo Model$\rightarrow B^{\mathrm{HM}} = B_{1h} + B_{2h} + B_{3h}$
Halo Model PT$\rightarrow$ Valageas and Nishimichi 2011a, 2011b
Benchmark model$\rightarrow B = f(K) S^{\mathrm{SPT}}_{tree}$; S is shape function
2-halo boost.



Modelling the Bispectrum in real space: Summary

TABLE II. Wavenumber $k^*_{\rm max}$ where the total correlator $\mathcal{T}$ (Eq. 7) between the perturbative theory and the benchmark model deviates by more than 10% (5%) from unity. In the case of $z = 0$, we only report the 10% results, as the accuracy of the benchmark model is lower.

Perturbation theories			
Threshold 10% (5%)	$k^*_{\rm max}$ $[h/{\rm Mpc}]$		
Theory	$z = 0$	$z = 1$	$z = 2$
Tree-level	0.13	0.22 (0.17)	0.27 (0.20)
NL tree-level	0.17	0.30 (0.22)	0.42 (0.31)
SPT	0.11	0.37 (0.14)	0.66 (0.49)
EFT	0.29	0.45 (0.36)	0.60 (0.50)
MPTbreeze	0.16	0.24 (0.21)	0.32 (0.28)
RLPT	0.15	0.22 (0.19)	0.30 (0.26)

TABLE III. Wavenumber $k^*_{\rm max}$ where the amplitude deviation for phenomenological halo models is greater than 20% when compared to the three-shape benchmark model matched to simulations. (The small k excess problem of the standard halo model is ignored.) At $z = 0$ all models agree within 20% over the entire range of scales.

Phenomenological halo models			
Threshold 20%	$k^*_{\rm max}$ $[h/{\rm Mpc}]$		
Theory	$z = 0$	$z = 1$	$z = 2$
Standard halo model	> 8	0.47	0.51
Combined halo-PT model	> 8	0.48	0.68
9-parameter fit	> 8	0.82	0.90

Modelling the Bispectrum in redshift space

The tree level prediction for the redshift-space bispectrum,

$$B^s(\mathbf{k}_1, \mathbf{k}_2) = D^B_{\rm fog}\left[2P_{\rm lin}(k_1)\, Z_1(\mathbf{k}_1)\, P_{\rm lin}(k_2)\, Z_1(\mathbf{k}_2)\, Z_2(\mathbf{k}_1, \mathbf{k}_2) + \text{cyc.}\right]$$

$$\begin{aligned} Z_1(\mathbf{k}_i) &\equiv (1 + f\mu_i^2) \\ Z_2(\mathbf{k}_1, \mathbf{k}_2) &\equiv \left[F_2(\mathbf{k}_1, \mathbf{k}_2) + \frac{f\mu k}{2}\left(\frac{\mu_1}{k_1} + \frac{\mu_2}{k_2}\right)\right] + f\mu^2 G_2(\mathbf{k}_1, \mathbf{k}_2) + \\ &+ \frac{f^3\mu k}{2}\mu_1\mu_2\left(\frac{\mu_2}{k_1} + \frac{\mu_1}{k_2}\right) \end{aligned}$$

$f \equiv d\ln\delta/d\ln a$, $\mu \equiv (\mu_1 k_1 + \mu_2 k_2)/k$, $k^2 = (\mathbf{k}_1 + \mathbf{k}_2)^2$.
G_2 is the second-order real-space kernels of the velocities.
Non-linear damping due to intra-halo velocity dispersion (FoG)

$$D^B_{\rm fog} = D^B_{\rm fog}(k_1, k_2, k_3, \sigma^B_{\rm fog}[z])$$

Modelling the Bispectrum in redshift space

The tree level prediction for the redshift-space bispectrum,

$$B^s(\mathbf{k}_1, \mathbf{k}_2) = D^B_{\rm fog}\left[2P_{\rm lin}(k_1)\, Z_1(\mathbf{k}_1)\, P_{\rm lin}(k_2)\, Z_1(\mathbf{k}_2)\, Z_2(\mathbf{k}_1, \mathbf{k}_2) + \text{cyc.}\right]$$

$$\begin{aligned} Z_1(\mathbf{k}_i) &\equiv (1 + f\mu_i^2) \\ Z_2(\mathbf{k}_1, \mathbf{k}_2) &\equiv \left[F_2(\mathbf{k}_1, \mathbf{k}_2) + \frac{f\mu k}{2}\left(\frac{\mu_1}{k_1} + \frac{\mu_2}{k_2}\right)\right] + f\mu^2 G_2(\mathbf{k}_1, \mathbf{k}_2) + \\ &+ \frac{f^3\mu k}{2}\mu_1\mu_2\left(\frac{\mu_2}{k_1} + \frac{\mu_1}{k_2}\right) \end{aligned}$$

$f \equiv d\ln\delta/d\ln a$, $\mu \equiv (\mu_1 k_1 + \mu_2 k_2)/k$, $k^2 = (\mathbf{k}_1 + \mathbf{k}_2)^2$.
G_2 is the second-order real-space kernels of the velocities.
Non-linear damping due to intra-halo velocity dispersion (FoG)

$$D^B_{\rm fog} = D^B_{\rm fog}(k_1, k_2, k_3, \sigma^B_{\rm fog}[z])$$

Modelling the Bispectrum in redshift space

Inspired by the real space modification, we perform a similar transformation to G_2 (HG-M et al. 2014),

$$
\begin{aligned}
G_2^{\rm eff}(\mathbf{k}_i,\mathbf{k}_j) &= \frac{3}{7} a(n_i,k_i,\mathbf{a}^G)a(n_j,k_j,\mathbf{a}^G) \\
&+ \frac{1}{2}\cos(\alpha_{ij})\left(\frac{k_i}{k_j}+\frac{k_j}{k_i}\right) b(n_i,k_i,\mathbf{a}^G)b(n_j,k_j,\mathbf{a}^G) \\
&+ \frac{4}{7}\cos^2(\alpha_{ij}) c(n_i,k_i,\mathbf{a}^G)c(n_j,k_j,\mathbf{a}^G)
\end{aligned}
$$

where $\mathbf{a}^G = \{a_1^G, \ldots, a_9^G\}$ to be fitted from N-body simulations.

$$Z_2 \to Z_2^{\rm eff}$$

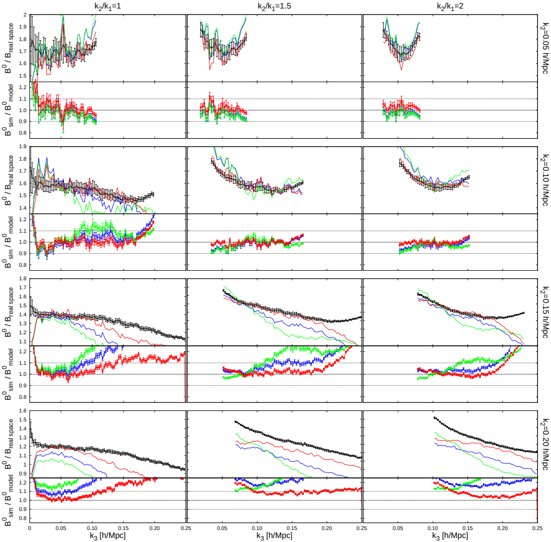
We will work with the bispectrum monopole (angle averaged),

$$B^{(0)}(\mathbf{k}_1,\mathbf{k}_2) = \int d\mu_1\, d\mu_2\, B^s(\mathbf{k}_1,\mathbf{k}_2)$$

Modelling the Bispectrum in redshift space

$z = 0$

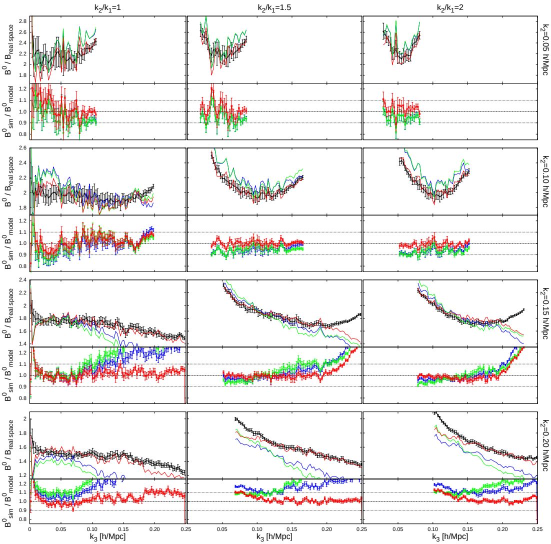
- We fit $\mathbf{a}^G$ to $B^{(0)}$ for $k_1/k_2 = 1.0, 1.5, 2.0, 2.5$, for $z = 0, 0.5, 1.0, 1.5$.
- We allow the $\sigma_{\rm fog}^B[z]$ to be free as well.
- $F_2^{\rm eff}$ & $G_2^{\rm eff}$ in $Z_2^{\rm eff}$
- $F_2^{\rm eff}$ & G_2 in $Z_2^{\rm eff}$
- F_2 & G_2 in Z_2



Modelling the Bispectrum in redshift space

$z = 0.5$

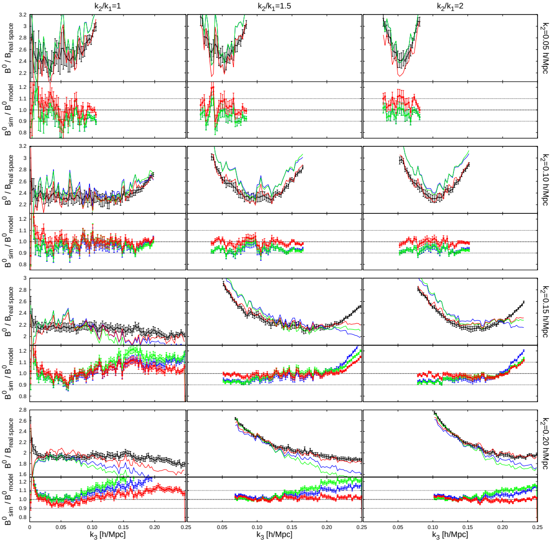
- We fit $\mathbf{a}^G$ to $B^{(0)}$ for $k_1/k_2 = 1.0, 1.5, 2.0, 2.5$, for $z = 0, 0.5, 1.0, 1.5$.
- We allow the $\sigma^B_{\rm fog}[z]$ to be free as well.
- $F_2^{\rm eff}$ & $G_2^{\rm eff}$ in $Z_2^{\rm eff}$
- $F_2^{\rm eff}$ & G_2 in $Z_2^{\rm eff}$
- F_2 & G_2 in Z_2



Modelling the Bispectrum in redshift space

$z = 1.0$

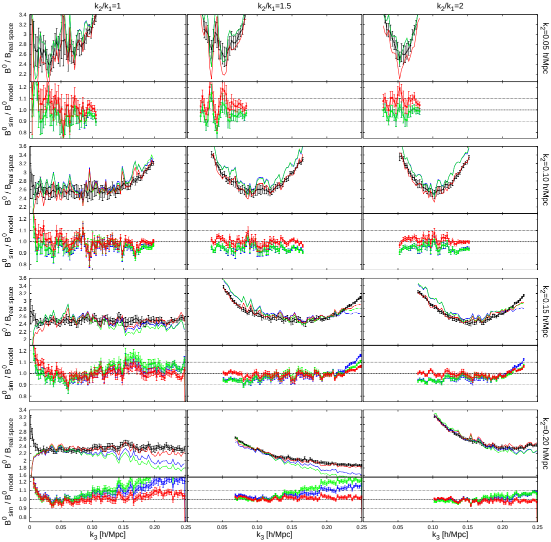
- We fit $\mathbf{a}^G$ to $B^{(0)}$ for $k_1/k_2 = 1.0, 1.5, 2.0, 2.5$, for $z = 0, 0.5, 1.0, 1.5$.
- We allow the $\sigma^B_{\rm fog}[z]$ to be free as well.
- $F_2^{\rm eff}$ & $G_2^{\rm eff}$ in $Z_2^{\rm eff}$
- $F_2^{\rm eff}$ & G_2 in $Z_2^{\rm eff}$
- F_2 & G_2 in Z_2



Modelling the Bispectrum in redshift space

$z = 1.5$

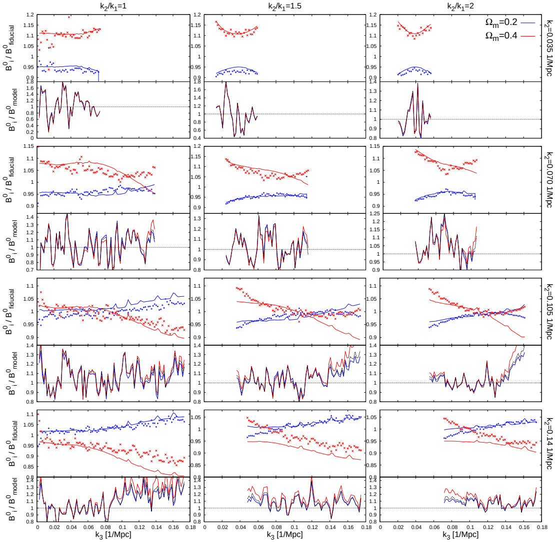
- We fit $\mathbf{a}^G$ to $B^{(0)}$ for $k_1/k_2 = 1.0, 1.5, 2.0, 2.5$, for $z = 0, 0.5, 1.0, 1.5$.
- We allow the $\sigma_{\rm fog}^B[z]$ to be free as well.
- $F_2^{\rm eff}$ & $G_2^{\rm eff}$ in $Z_2^{\rm eff}$
- $F_2^{\rm eff}$ & G_2 in $Z_2^{\rm eff}$
- F_2 & G_2 in Z_2



Dependence with Cosmology?

The best-fitting $\mathbf{a}^F$ & $\mathbf{a}^G$ parameters present a weak dependence with σ_8 and Ω_m.

- $\Omega_m = 0.2, 0.27(\mathit{fid}), 0.4$.
- $\Omega_m h^2 = \mathrm{constant}$ (SV cancellation).
- σ_8 adjusted to have same amplitude in PS



Modelling the Bispectrum in redshift space

Other phenomenological approaches for redshift space bispectrum based on halo model (Smith, Sheth & Scoccimarro 2008)

16

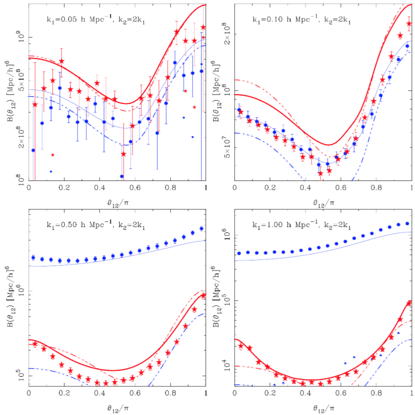


FIG. 4: Configuration dependence of the bispectrum in real and redshift space measured for the dark matter particles in the ensemble of LCDM simulations. The four panels show the configuration dependence for k-space triangles with $k_2/k_1 = 2$ and for scales $k_1 = \{0.05, 0.1, 0.5, 1.0\}\, h\mathrm{Mpc}^{-1}$. Red and blue colors distinguish between real and redshift space quantities. The solid points with error bars show measurements: large solid points are for the real (dots) and redshift space (stars) monopole bispectra; the corresponding smaller points are the imaginary bispectra (which should be zero). The solid lines show the predictions from HM (thin) and HMs (thick). The triple dot-dash lines show the predictions from PT (thin) and PTs (thick). In the top left panel, for clarity we have suppressed the errors on the imaginary bispectra.

Other issues

There are other issues concerning the bispectrum modelling,

- Galaxy Bias. Non-linear and non-local bias model seems to be needed (McDonald & Roy 09).
- How to model the survey selection function? Wilson et al. 2016 seems the best approach but needs to be extended to the bispectrum.
- Is Poisson shot noise prediction enough? How the bispectrum shot noise is modified by halo exclusion?
- Bispectrum BAO. How much information is it there? How much does the reconstruction solve this problem? After reconstruction is it worth to look and use bispectrum BAO?
- ...