Combining simulations and theory for LSS

Martin White

(UC Berkeley, LBNL)

Theory lunch, 22 Jun 2022

Combining PT and N-body simulations

- We've been wondering/thinking about how to combine the two most common approaches to LSS modeling
  - Numerical simulations.
  - Perturbation theory.
- They have a lot in common!
  - We normally start simulations using PT, and we can 'realize' PT on a grid.
- So what are the best ways they can help each other?
  - Accurate predictions from 'small' boxes.
  - Bias modeling and non-perturbative dynamics.

Accurate predictions from 'small' boxes

Martin White

(UC Berkeley, LBNL)

Theory lunch, 22 Jun 2022
Based on Kokron+22

Background: Control Variates

(First introduced into LSS by Chartier & Wandelt as "CARPool")

- Imagine I want $\langle \mathbf{x} \rangle$ but realizations of $\mathbf{x}$ are expensive to produce.
  - Example, the matter power spectrum or halo power spectrum.
  - Especially true for simulations of high resolution, or including hydrodynamics, or RT, where boxes tend to be 'small'.
- Further assume I can cheaply produce $\mathbf{c}$, where $\mathbf{c}$ is correlated with $\mathbf{x}$ and $\mu_c = \langle \mathbf{c} \rangle$ is known.
  - $\mathbf{c}$ is known as the *control variate*.
  - Example: $\mathbf{c}$ is the density power spectrum in the Zeldovich approximation (lowest order LPT).
- If we form

$$\mathbf{y} \equiv \mathbf{x} - \beta \left(\mathbf{c} - \mu_c \right)$$

then $\langle \mathbf{y} \rangle$ is an unbiased estimator of $\langle \mathbf{x} \rangle$ for any β.

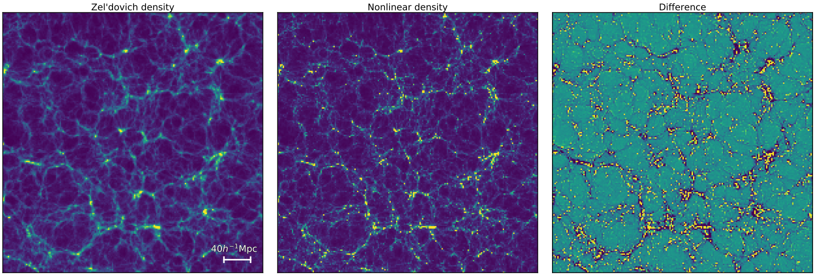
Background: Control Variates

- Now choose
$$\beta^\star = \frac{\mathrm{Cov}[\mathbf{x}, \mathbf{c}]}{\sigma_c^2}$$
(really a matrix expression but frequently just approximate as diagonal).
- Then
$$\mathrm{Var}[\mathbf{y}] = \mathrm{Var}[\mathbf{x}]\left(1 - \rho_{xc}^2\right) \quad , \quad \rho_{xc} \equiv \frac{\mathrm{Cov}[\mathbf{x}, \mathbf{c}]}{\mathrm{Std}[\mathbf{x}]\mathrm{Std}[\mathbf{c}]}$$
- If $\rho_{xc} \approx 1$ then $\mathbf{y}$ is a very low noise/scatter quantity that well-estimates $\langle \mathbf{x} \rangle$.
- Heuristically if $\mathbf{c}$ fluctuates above μ_c then $\mathbf{x}$ probably also fluctuated "up" so you should correct it down.

Zeldovich Control Variates

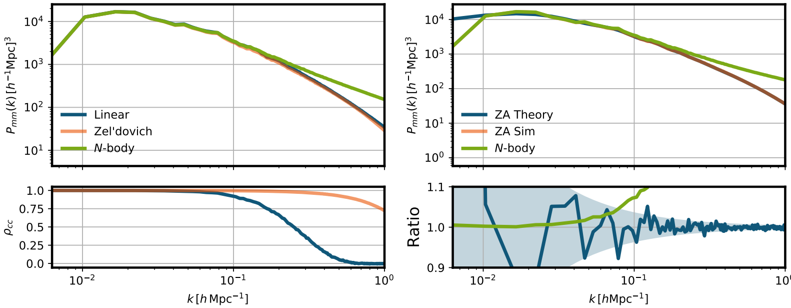
- Simulations always have limited dynamic range; in particular large scales are often "noisy" due to sample variance.
- PT works very well on large scales!
- If you start your simulation using Lagrangian PT (e.g. the Zeldovich approximation, or higher order) then you already have a surrogate field that is well correlated with your final density field (of matter, galaxies, gas, electrons, ...).
- No need to generate special ICs, rerun your simulation or do anything 'fancy'!!
- This technique is mathematically rigorous, and works for all spectra (including higher-order spectra, etc.)

Zeldovich and the cosmic web!



Kokron+22

Very correlated!



Kokron+22

Zeldovich approximation: algebra

In the Lagrangian picture of structure formation, $\delta(\mathbf{x})$ is calculated from the movement of Lagrangian fluid elements across time. Particles located at a position $\mathbf{q}$ at initial conditions are advected to

$$\mathbf{x}(a) = \mathbf{q} + \boldsymbol{\Psi}(\mathbf{q}, a).$$

where $a = (1+z)^{-1}$ is the scale factor.
If the initial distribution of densities is approximately uniform, $\rho(\mathbf{q}) \approx \bar{\rho}$, then at late times

$$1 + \delta_t(\mathbf{x}, a) = \int d^3q \, F[\delta(\mathbf{q})] \delta^{(D)} \left(\mathbf{x} - \mathbf{q} - \boldsymbol{\Psi}(\mathbf{q}, a)\right)$$

We expand the functional $F[\delta(\mathbf{q})]$

$$F[\delta(\mathbf{q})] \approx 1 + b_1 \delta(\mathbf{q}) + b_2(\delta^2(\mathbf{q}) - \langle \delta^2 \rangle) + \cdots$$

Zeldovich approximation: algebra

In the first order solution to Lagrangian Perturbation Theory, the Zeldovich approximation, fluid elements in the Universe propagate in straight lines, with a direction set by the potential sourced by the initial matter distribution. These displacements are obtained by solving the linearized continuity equation, and read

$$\Psi^{\mathrm{ZA}}(\boldsymbol{q}, a) = \int \frac{d^3k}{(2\pi)^3} e^{i\boldsymbol{k}\cdot\boldsymbol{q}} \frac{i\boldsymbol{k}}{k^2} \delta_{\mathrm{lin}}(\boldsymbol{k}).$$

Note that $\boldsymbol{\Psi}$ is Gaussian!

Zeldovich approximation: algebra

We evaluate $P(k)$ using a generating function(al)

$$\mathcal{Z}(\mathbf{q}, \lambda_n, a_n) = \left\langle e^{\mathcal{M}(\mathbf{q}_1, \mathbf{q}_2, \lambda_n, a_n)} \right\rangle$$

where the exponent is defined as

$$\mathcal{M}(\mathbf{q}_1, \mathbf{q}_2) = i\mathbf{k} \cdot \Delta + \sum_{n=1,2} \lambda_n \delta(\mathbf{q}_n) + a_{n,ij} s_{ij}(\mathbf{q}_n)$$

and $\Delta = \Psi(\mathbf{q}_1) - \Psi(\mathbf{q}_2)$.
Since $\mathcal{M}$ is Gaussian we have that $\mathcal{Z}$ is simply given by the exponentiated second cumulant (free field theory).

Zeldovich approximation: algebra

In order to obtain the biased-tracer power spectrum we use the substitutions

$$b_1\delta_n \to b_1\frac{d}{d\lambda_n} \quad , \quad b_2\delta_n^2 \to b_2\frac{d^2}{d\lambda_n^2} \quad , \quad b_s s_n^2 \to b_s\delta_{ia}\delta_{jb}\,\frac{d}{da_{n,ij}}\frac{d}{da_{n,ab}}.$$

in the bias functional $F(\mathbf{q}_n)$, which becomes an operator $\hat{F}_n$. The power spectrum is then given by

$$P(k) = \int d^3\mathbf{q}\ e^{i\mathbf{k}\cdot\mathbf{q}}\ \ \hat{F}_1\hat{F}_2\mathcal{Z}(\mathbf{q},\lambda_n,a_n)\Big|_{a_n,\lambda_n=0}$$

Zeldovich approximation: algebra

Tracer power spectra will receive contributions from correlations between the advected operators that compose the functional F. This decomposition will have the form

$$P^{tt}(k) = \sum_{i,j \in \{1,\delta,\delta^2,\cdots\}} b_i b_j P_{ij}(k),$$

and

$$P^{tm}(k) = \sum_{j \in \{1,\delta,\delta^2,\cdots\}} b_j P_{1j}(k),$$

where P_{1j} are cross-correlations between the matter density field and the bias operators.

For example

$$\begin{aligned}
P_{gg}(k) &= P_{11}(k) + 2b_1 P_{1\delta}(k) + b_1^2 P_{\delta\delta}(k) + \cdots \\
P_{gm}(k) &= P_{11}(k) + b_1 P_{1\delta}(k) + \cdots
\end{aligned}$$

Zeldovich approximation: algebra

If we write

$$A_{ij}(\mathbf{q}) = \langle \Delta_i \Delta_j \rangle = X(q)\delta_{ij} + Y(q)\hat{q}_i \hat{q}_j$$

then each of the component spectra can be written as

$$P_{ij}(k) = \int d^3q \ e^{i\mathbf{k}\cdot\mathbf{q} - \frac{1}{2}k_i k_j A_{ij}} \ F_{ij}(\mathbf{k}, \mathbf{q})$$

where

$$\begin{aligned}
(1,1) &: 1 \\
(1, b_1) &: ik_i \langle \delta_1 \Delta_i \rangle \\
(b_1, b_1) &: \xi_L - k_i k_j \langle \delta_1 \Delta_i \rangle \langle \delta_1 \Delta_j \rangle \\
\mathrm{etc}
\end{aligned}$$

Zeldovich approximation: algebra

All of the correlators, $\langle \delta_1 \Delta_i \rangle$, etc. can be done as Hankel transforms.
Each of the terms is scalar, vector, tensor, ... under rotations so the resulting expressions for the P_{ij} can also be written in terms of Hankel transforms:

$$P_{ij}(k) = 4\pi \sum_{\ell=0}^{\infty} \int q^2 dq\ e^{-k^2(X+Y)/2} \left(\frac{kY}{q}\right)^{\ell} f_{ij}^{(\ell)}(q,k) j_\ell(kq),$$

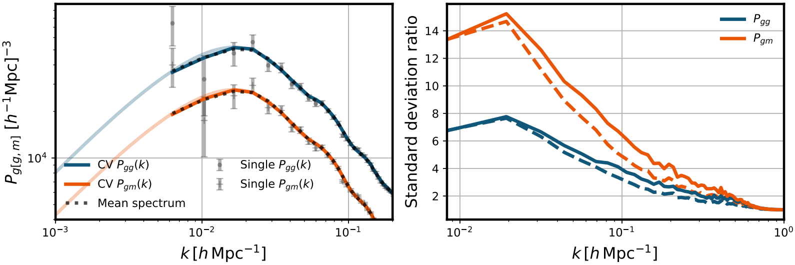
with the $f_{ij}^{(\ell)}$ are simply related to the F_{ij} kernels above.
Each of the Hankel transforms can be done numerically using FFTs (very fast!).

https://github.com/sfschen/ZeNBu

Does it work?

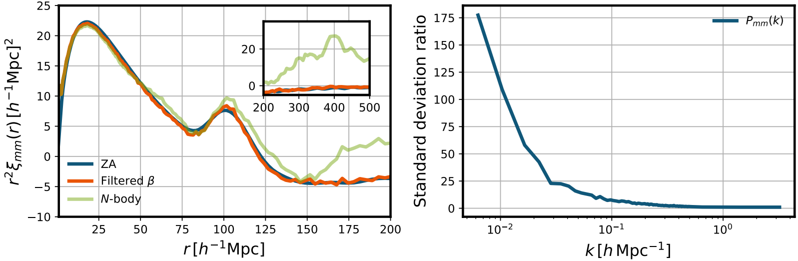
Accurate predictions from small boxes!

An example of measuring P_{gg} and P_{gm} from a single $1\,h^{-1}$Gpc box, after applying CV and compared to the average of 100 such boxes (light lines extend to low k using PT).
[Even better performance for P_{mm}; not shown.]



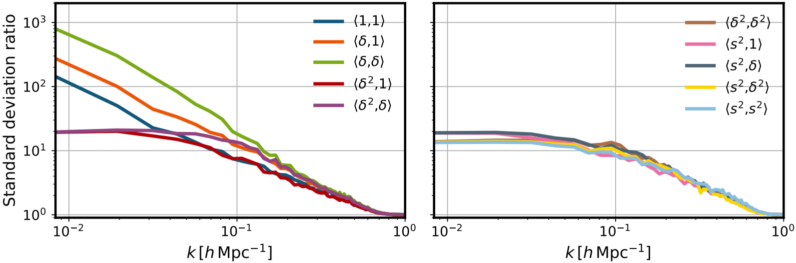
Kokron+22

Also in configuration space



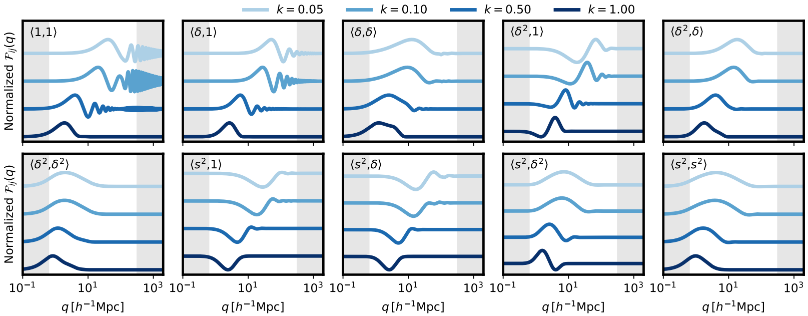
Kokron+22

Large reduction in errors!



Kokron+22

Resolution requirements



Kokron+22

Simulations and symmetries

Martin White

(UC Berkeley, LBNL)

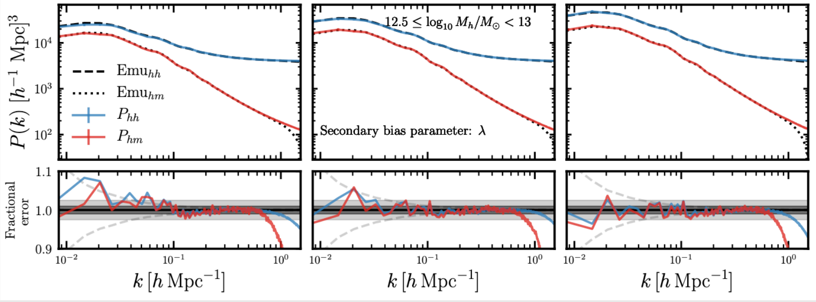
Theory lunch, 22 Jun 2022
Based on Modi+20, Kokron+21

Simulations and Symmetries

- We can simulate structure formation in a DM-only Universe pretty well.
- It's the baryonic component that is "hard"!
  - Don't understand cooling, star-formation, feedback, ...
  - Resort to parameterized models (when to stop adding parameters, how to test for numerical convergence?)
- Symmetries-based thinking is ubiquitous in PT studies and very powerful.
- PT folks and simulators are trying to solve the same problems ...
- Can we have the best of both worlds?
  - Use dynamics from N-body simulations, but the "galaxies" (symmetries-based bias technique) from perturbation theory [Modi+20, Kokron+21, Hadzhiyska+21, Zennaro+21,...].

Can push into the non-linear regime

Can fit mock catalog data for "3 × 2pt analyses" to 1-2% even for samples with assembly bias and other complex selections and even including hydrodynamics.

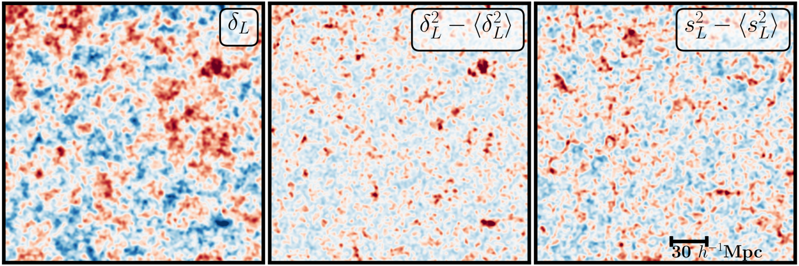


Now we can simply "emulate" the basis spectra using standard techniques (no need to emulate the bias parameters – analytic)!

Kokron+21

The hybrid EFT procedure in pictures

Generate initial conditions as per usual ... from δ_L you can also compute δ_L^2 and the shear field, s_{ij}:

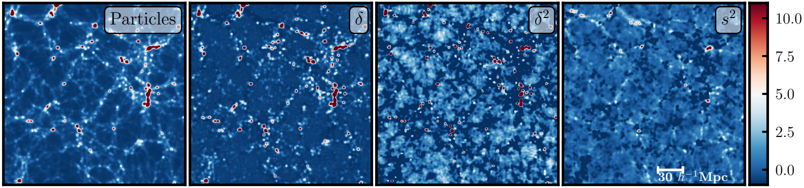


Each particle is assigned the δ_L, ... at its initial position.

Kokron+21

The hybrid EFT procedure in pictures

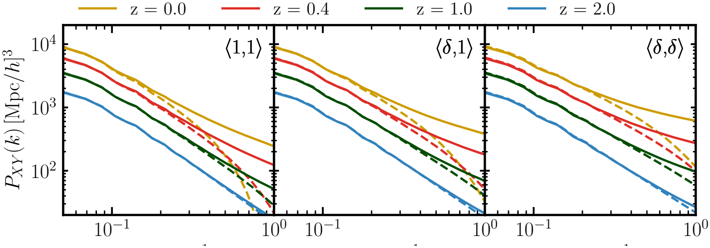
Advect the particles to their final positions using the full N-body dynamics (i.e. run the simulation), and bin using weights 1, δ_L, δ_L^2, etc.



(No need for halo or subhalo finding, merger trees, etc.)

The hybrid EFT procedure in pictures

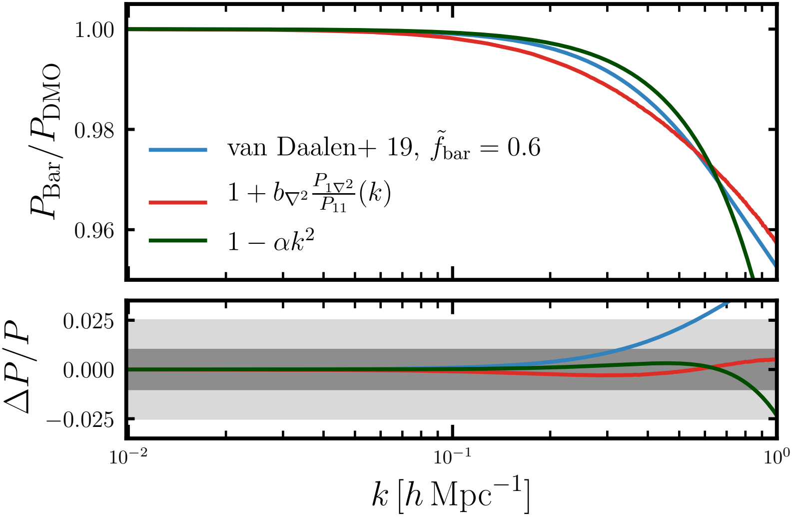
Take all of the cross-spectra, $P_{XY}(k)$ using standard FFT methods, e.g.



The power spectrum for any biased tracer, or the cross-spectrum between any two tracers, is a linear combination of these "basis spectra" (10 in all) with analytic "bias dependence": $\sum_{ij} b_i b_j P_{ij}$.

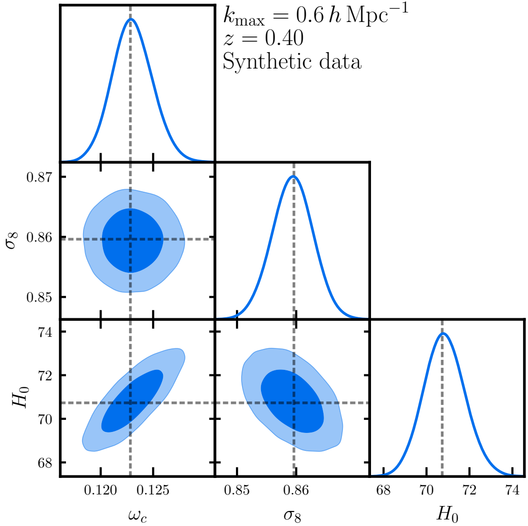
Also does 'baryons'

Classic Jeans argument would tell you that $\nabla p \sim \nabla c_s^2 \delta$ and $\nabla \Phi \sim \nabla \nabla^{-2} \delta$ should be related by k^2:



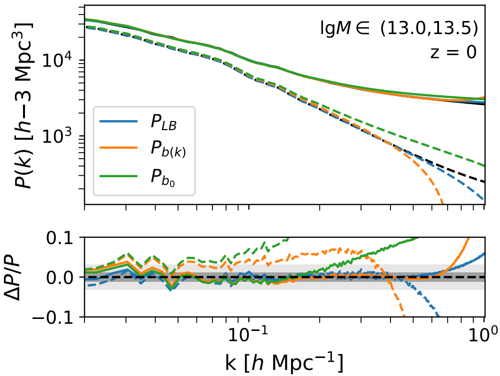
And leads to unbiased fits

Cosmological parameter inference using an emulator with noiseless data



Is this just really easy?

A comparison of this “symmetries based bias expansion” or “hybrid” approach with a constant or polynomial (multiplicative) bias



Modi+20

The End!