TheHaloMod: An online calculator for the halo model

Steven G. Murray[a], Benedikt Diemer[b], Zhaoting Chen[c], Anton Glenn Neuhold Jr.[d], Alexander Schnapp[d], Tia Peruzzi[d], Daniel Blevins[d], Trent Engelman[d]

[a] *School of Earth and Space Exploration, Arizona State University, Tempe, AZ, USA*
[b] *Department of Astronomy, University of Maryland, College Park, MD 20742, USA*
[c] *Jodrell Bank Centre for Astrophysics, School of Physics and Astronomy, The University of Manchester, Manchester M13 9PL, UK*
[d] *ASU School of Computing, Informatics, and Decision Systems Engineering, Arizona State University, Tempe, AZ, USA*

Abstract

The halo model is a successful framework for describing the distribution of matter in the Universe – from weak lensing observables to galaxy ~~2-point~~ n-point correlation functions. We review the basic formulation of the halo model and several of its components in the context of galaxy ~~two-point~~ 2-point statistics, developing a coherent framework for its application.

We use this framework to motivate the presentation of a new ~~Python~~ PYTHON tool for simple and efficient calculation of halo model quantities, and their extension to galaxy statistics via a *halo occupation distribution*, called HALOMOD. This tool is efficient, simple to use, comprehensive and importantly provides a great deal of flexibility in terms of custom extensions. This ~~Python~~ PYTHON tool is complemented by a new web-application at `https://thehalomod.app` that supports the generation of many halo model quantities directly from the browser – useful for educators, students, theorists and observers.

Keywords: large-scale structure of universe – dark matter – galaxies: halos – methods: analytical – methods: numerical

1. Introduction

The halo model ~~(Neyman et al., 1953; Peacock and Smith, 2000; Seljak, 2000; Ma and Fry, 2000; Cooray and Sheth, 2002)~~ an enormously successful analytical description of the large-scale distribution of matter in our Universe. It describes the statistics of the dark matter density field well into the nonlinear regime, beyond the reach of perturbation theory. It does so by combining linear theory predictions with empirical properties of dark matter halos, via the assumption that the sum total of dark matter resides in these clumps, and that a handful of simple functions based on the mass of these halos – such as their radial density profile and clustering bias – can universally describe them.

In combination with a *halo occupation distribution* (HOD) model (Kauffmann et al., 1997; Scoccimarro et al., 2001; Berlind et al., 2003; Zheng et al., 2005), the predictions of the halo model can be extended to galaxy populations, and therefore used to model clustering in large galaxy surveys. One of the key advantages of halo model and HOD formalism is that they can predict any clustering statistic on any scale (Zehavi et al., 2011), from real-space or projected ~~two-point~~ 2-point correlation functions (2PCFs), to galaxy-galaxy lensing, to higher-order correlations.

In practice, the HOD formalism has been widely used in the interpretation of galaxy populations in the past decade. Most of these studies have focused on determining the parameters of the HOD (i.e. the galaxy-halo connection) from the 2PCF of galaxies ~~(eg. Moustakas and Somerville, 2002; Bullock et al., 2002; Zheng, 2004; Zehavi et al., 2005; Blake et al., 200...~~ the analytical framework that we present in this paper.

However, other observables such as galaxy-galaxy lensing ~~(Mandelbaum et al., 2006)~~ (Mandelbaum et al., 2006; Cacciato et a... 2000; Seljak, 2000; Ma and Fry, 2000; Cooray and Sheth, 2002) (M the 2PCF of radio galaxies ~~(Wake et al., 2008; Kim et al., 2011)~~ (Wa galaxy-quasar cross-correlations (Shen et al., 2013), near-UV cross-correlations (Krause et al., 2012), Fast Radio Bursts (FRBs; Rafiei-Ravandi et al., 2020) and H I intensity map cross-correlations ~~(Padmanabhan et al., 2016; Wolz et al.,~~ auto-correlations (Schneider et al., 2020; Umeh et al., 2021) have also received application through the same framework. Furthermore, several studies have successfully combined observables (either in cross-correlation or via joint likelihoods) to break degeneracies ~~(Leauthaud et al., 2011, 2012; More, 2013), and~~ (eg. Leauthaud et al., 2011, 2012; More, 2013; Schaan and Wh Finally, the combination of the galaxy 2PCF with other statistics, eg. weak-lensing ~~(More et al., 2015),~~ (More et al., 2015; ... cluster counts and auto-correlations (To et al., 2021), or both (To et al., 2020), stellar mass functions (Coupon et al., 2015) and group mass-to-number ratios (Reddick et al., 2014) has resulted in the ability to simultaneously constrain cosmological parameters along with those of the HOD.

Clearly, the halo model (hereafter HM) framework, complemented by an HOD or comparable mechanism, can be of wide utility in the interpretation of large surveys. Increasingly, observational studies are employing a non-analytic incarnation of the HM, in which simulated halos are "painted" with a particular galaxy sample, given detailed semi-analytic models of the galaxy-halo connection

(eg. Carretero et al., 2015). In particular, the HALOTOOLS[1] library has become a popular implementation of this approach. Nevertheless, a purely analytic construction of the HM is still of great importance and utility: it is our best fundamental model of the non-linear scales of the matter distribution.

The HM is a complex framework as it synthesises many related sub-components (eg. halo profiles, mass functions, bias models, spatial filters, halo exclusion models, concentration–mass relations) to produce spatial statistics. These sub-components can often be modelled independently via simulations, and new more accurate models are being produced regularly by the community. This highlights the need for an implementation of the HM framework that has the flexibility and modularity to enable easy switching between models for the various sub-components, and rapid development of new models to incorporate into the framework.

There are a few publicly available implementations of the analytic HM, many of which were used in the testing of our code although the development of most of them has been discontinued (e.g., CHOMP[2], HMCODE[3], CCL[4] and AUM[5]). In addition, methods of computing HM quantities based on emulation from simulations are becoming available (eg. DARKQUEST[6] and COSMICEMU[7] – though the latter only supports matter correlations). However, it is our experience from that a remarkable number of practitioners develop their own tools — whether based heavily on existing (public or private) code or from the ground up. Additionally, existing codes – where still in active development – do not tend to offer a great deal of flexibility in the composition and extension of the various components and models involved in the halo model, rather focusing on a limited set of well-honed models and tuned for maximum performance.

This paper presents a robust new implementation of the analytic HM, called HALOMOD[8], that aims to fill this important gap, and be as generally useful as possible by adhering to the following principles:

- **Intuitive.** The API is well-specified and intuitive for the user, and exhaustively documented. We illustrate the simplicity of the usage of HALOMOD in §4.1.1 and note that full online documentation is available at https://halomod.readthedocs.io, including API specifications and examples/tutorials. In addition, installing HALOMOD is as simple as running `pip install halomod`.

- **Simple.** Though many aspects of the calculations are unavoidably non-trivial, a simple layout of the code within a highly structured framework is important. We lay out HALOMOD's simple code framework in §4.1.2. This promotes future development, and usage by a broad cross-section of researchers.

- **Efficient.** Though not as immediately important as flexibility, it is important that the code be efficient. This includes both algorithmic and numerical efficiency, but also efficiency of the writing of user-side code. We outline our strategies for efficiency in §4.1.3.

- **Flexible/Extendible.** The HM is a rapidly evolving framework, with individual components constantly improving, and the framework itself being extended. Building a static implementation is therefore non-conducive to the development of the field. Components need to be as plug-and-play as possible, with new models easily created and inserted on the fly. Our implementation of such a plug-and-play system is outlined in §4.1.4.

- **Comprehensive.** HALOMOD acts as an archive for all the modelling that has been done by the community. It collates and compiles the various models and extensions in a cohesive way so that new models can be quickly compared, and insights gained. Our efforts towards this in HALOMOD are evidenced by the numerous tables of models throughout §4.3.

- **Open.** HALOMOD is open-source, not simply in the sense that it is publicly available. It is developed with many open-source best-practices, such as continuous integration, high test coverage, automated code linting/formatting, formal software versioning, modern version control practices and online documentation.

Both philosophically and technically, HALOMOD inherits from the HMF halo mass function package[9] which was first presented in Murray et al. (2013b). Many developments have occurred in HMF since its first publication, and the technical framework of HALOMOD presented in this paper is essentially inherited from the updates in HMF. Thus, this paper can also secondarily be considered an update of HMF.

Our vision is for HALOMOD to be useful as (i) a baseline standard for user-specific private codes, (ii) a simple interface for those not actively researching in the field, but who may wish to calculate clustering statistics for their data, (iii) a tool for fast exploratory analysis, and comparison between models and (iv) a stable framework for more rapid development of theoretical extensions to the HM, and modelling of its various components.

In addition to HALOMOD and HMF package, we also present a new web-application, THEHALOMOD[10], which is

---

[1] https://halotools.readthedocs.io

[2] https://github.com/morriscb/chomp – discontinued.

[3] https://github.com/alexander-mead/HMcode/

[4] https://github.com/LSSTDESC/CCL

[5] https://github.com/surhudm/aum discontinued.

[6] https://github.com/DarkQuestCosmology/dark_emulator_public

[7] https://github.com/lanl/CosmicEmu

[8] https://github.com/halomod/halomod

[9] https://github.com/halomod/hmf

[10] https://thehalomod.app

able to generate full halo model quantities (eg. ~~two-point~~ 2-point correlation functions and galaxy power spectra) without ever having to install the ~~Python~~ PYTHON package. It is a successor to the popular HMFCALC (Murray et al., 2013b) web-application, and includes the full range of functionality of HMFCALC. The presentation of THE-HALOMOD completes our vision for HALOMOD with (v) a tool for educators to easily and interactively present cosmologically relevant quantities graphically.

This paper is structured as follows: §2 details the theory of the HM particularly in the context of dark matter ~~two-point~~ 2-point statistics, collating the various components involved in a manner consistent with our implementation. Following this, we describe how to extend the halo model to tracer populations in §3. Then, we describe our code and its usage in §4, and in §6 we present an illustrative example. In §7 we define a prospectus for the future, before summarising and concluding in §8.

Note that the code to produce all figures of different component models in this paper, as well as the example application, are available publicly as examples in HALO-MOD's documentation[11]. This paper refers to HMF version ~~3.2~~3.4.1 and HALOMOD version ~~2.0~~2.1.0.

2. The Dark Matter Halo Model

The broad assumption underpinning the HM is that in hierarchical structure formation scenarios, all mass is expected to be bound to halos at some scale[12]. If this is the case, then the entire nonlinear density field may be reconstructed by summing contributions from the individual halos. If, in addition, we may describe the average radial density profile of halos as spherically-symmetric, with a shape that depends solely on the mass of the halo itself, we can write:

$$\rho(\mathbf{x}) = \sum \rho_{h,i}(|\mathbf{x} - \mathbf{x_i}|, m_i) \qquad (1)$$

where $\rho_h(r|m)$ is the density of a halo with mass m at radius r, and x_i are the coordinates of the halo centres.

The application of the HM rests in converting the premise of ~~Eq. 1~~Eq. 1 into a semi-analytic integration. This requires knowledge of four key components:

1. The ~~average radial density profile~~ moments of the density profiles of halos, ~~$\rho(r|m)$~~ $\langle \rho(r|m) \rangle$, $\langle \rho_{h,i}^2(r|m) \rangle$, etc.
2. The abundance of halos of a given mass (termed the halo mass function, or HMF), $n(m)$

3. The expected overdensity of halos of mass m, ~~$\delta_h(\mathbf{x}, m)$~~ $\Delta_h(\mathbf{x}, m)$ in a region with a given overdensity of matter $\delta(\mathbf{x})$, called the *halo bias*.
4. A model for the ~~linear~~ spatial distribution of matter, $\delta(\mathbf{x})$.

Furthermore, detailed calculations require the following extra components:

5. Additional parameters that describe the halo profile, most commonly parameterized as a concentration-mass relation, $c(m)$, linking the shape of a halo profile to its mass.
6. A model for "halo exclusion", which accounts for double-counting intra-halo correlations.
7. To extend the calculations to galaxies, a distribution function for the occupation of halos by galaxies, as a function of halo mass, $N(m)$ (the HOD; cf. §3).

Although the halo model can be used to describe the density field at any level of the n-point hierarchy, its most common application to date has been to compute ~~two-point~~ 2-point statistics. Thus, in this outline, we shall focus on the framework as it pertains to the 2PCF.

In particular, the 2PCF only requires the first two moments of the density profile, and we will furthermore assume that the second moment is just the square of the average density profile (i.e. we only require the first moment, cf. Giocoli et al. 2010). Our aim in this section is to introduce the theory in a manner conducive to our implementation, so we shall cover each of the six components in turn following the presentation of the core framework.

Throughout the section, we discuss isotropic ~~two-point~~ 2-point statistics, which measure the over-density of pairs of points at a certain scalar separation. This can be formulated either in Euclidean space, as the correlation function $\xi(r)$, or in Fourier-space, as the power spectrum $P(k)$. The correlation function is defined as the excess probability of locating a particle at separation $\mathbf{r}$ from a particle at position $\mathbf{x}$ in a given spatial distribution, and is written (for a homogeneous universe)

$$\xi(\mathbf{r}) = \frac{1}{V} \int_V \mathrm{d}^3\mathbf{x}\, \delta(\mathbf{x})\delta(\mathbf{x} - \mathbf{r}), \qquad (2)$$

where $\delta(\mathbf{x})$ is the overdensity

$$\delta(\mathbf{x}) = \frac{\rho(\mathbf{x}) - \bar{\rho}}{\bar{\rho}}. \qquad (3)$$

The power spectrum is merely the Fourier transform of the correlation function, and the standard Fourier convention in cosmology renders it

$$\xi(\mathbf{r}) = \int \frac{d^3\mathbf{k}}{(2\pi)^3} \frac{d^3\mathbf{k}}{(2\pi)^3} P(\mathbf{k}) e^{i\mathbf{k}\cdot\mathbf{r}}. \qquad (4)$$

Under the assumption of cosmic isotropy, the correlation function and power spectrum depend only on the amplitude of $\mathbf{r}$ and $\mathbf{k}$, respectively. Transformation from an

---

[11]At https://halomod.readthedocs.io/en/latest/examples/component-showcase.html and https://halomod.readthedocs.io/en/latest/examples/fitting.html respectively.

[12]This assumption is clearly an approximation, since even cold dark matter (CDM) has a free-streaming scale in the early Universe below which we expect non-virialized mass ~~(Frenk and White, 2012; ?)~~(Frenk and White, 2012; Schneider, 2015)

Figure 1: THeHaloMod in action. The main page (right panel) shows the currently computed models, with a selection of quantities to plot, and a number of actions available for each model (edit/clone/report bug/delete). The ~~plot itself is interactive. The~~ input form (~~two insets on the~~ left) provides a wide range of options, including all of the components presented throughout this paper. Each component ~~is defined in~~ has its own ~~tab~~section. ~~Those shown here are cosmology~~ The interface features responsive design and ~~the HOD model~~a dark mode.

(isotropic) power spectrum $P(k)$ to an (isotropic) correlation function $\xi(r)$ can thus be performed using an order-1/2 *Hankel* transform, which is the 3D Fourier transform of a spherically symmetric distribution:

$$\xi(r) = \frac{1}{2\pi^2} \int_0^\infty P(k) k^2 j_0(kr) \, dk, \quad (5)$$

where j_0 is the zeroth-order *spherical* Bessel function,

$$j_0(x) = \frac{\sin x}{x}. \quad (6)$$

We note that the halo model is *not limited* to such ~~two-point~~ 2-point statistics, but our implementation currently is.

2.1. Clustering Framework

It is convenient to formulate the framework of clustering in two regimes (Seljak, 2000), intra- and inter-halo pairs, called the 1-halo and 2-halo terms respectively. These approximately correspond to small- and large-scale structure (i.e. the contribution of each in the opposite regime is negligible), where 'small' is sub-megaparsec, and 'large' is ~~$\gtrsim\sim 5h^{-1}$~~ $\gtrsim 5h^{-1}$Mpc.

Throughout the following, the superscripts $1h$ and $2h$ will be used to denote this segregation. Furthermore, a subscript of DM will denote a dark matter statistic, while T will denote statistics of observable tracers (such as optical or HI-selected galaxies).

The total power spectrum and 2-point correlation function can be written simply as

$$P(k) = P_{DM}^{1h}(k) + P_{DM}^{2h}(k), \quad (7)$$

and

$$\xi(r) = [1 + \xi^{1h}(r)] + \xi^{2h}(r). \quad (8)$$

We will first describe the calculation of the 1-halo and 2-halo terms, each of which combines multiple physical components, which we will proceed to describe in §2.2-2.9.

2.1.1. 1-halo term

The pair-counts, proportional to $1 + \xi$, within a halo of mass m, are given by the self-convolution of the halo profile. Thus the total pair counts, normalised by a random field (specified by the constant $\bar{\rho}^2$) is given by

$$1 + \xi_{DM}^{1h}(r) = \frac{1}{\bar{\rho}^2} \int n(m) \frac{m}{\bar{\rho}}^2 \lambda(r|m) \, dm, \quad (9)$$

where λ is the ~~mass-normalised~~ self-convolution of the halo profile (i.e. the 3D integral of the profile multiplied by itself after a shift of length r, which has units of M^2/V, cf. §2.7), and $n(m)$ is the halo mass function (HMF).

It is common to express this in Fourier-space, since the self-convolution becomes a simple multiplication,

$$P_{DM}^{1h}(k) = \int n(m) \left(\frac{m}{\bar{\rho}}\right)^2 |u(k|m)|^2 \, dm, \quad (10)$$

where $u(k|m)$ is the ~~normalised~~ mass-normalised Fourier transform of the halo profile (which is unitless and approaches unity as $k \to 0$). This form has the distinct advantage that higher-order correlations merely involve an increase in the exponent of u, rather than an intractable multi-dimensional convolution. Conversely, for numerical purposes, if an analytic formulation of the self-convolution exists and the

4

intended quantity is the correlation function, then it is more efficient to use ~~Eq. 9~~ Eq. 9 directly.

Note that the behaviour as $k \to 0$ is to tend to a constant:

$$P_{\mathrm{DM}}^{\mathrm{1h}}(k \to 0) = \frac{1}{\bar{\rho}^2} \int n(m) m^2 \, \mathrm{d}m. \tag{11}$$

This "shot-noise-like" term (Cooray and Sheth, 2002; Ginzburg et al., 2017) arises due to exclusion features of the halo model, and is spurious. At the largest scales, it comes to dominate the 2-halo term. While true shot-noise is inversely proportional to the *number density* of the field (cf. §3.2), in this case we are considering the dark matter field to be essentially continuous (i.e. the mass of each DM 'particle' approaches zero, and its number density approaches infinity), and it therefore should have no shot-noise. We thus offer an option `force_1halo_turnover` in HALOMOD that forcibly removes this large-scale structure from the 1-halo term.

2.1.2. 2-halo term

The two-halo term is most elegantly expressed in Fourier-space,

$$P_{\mathrm{DM}}^{\mathrm{2h}}(k,r) = \int\int \mathrm{d}m_1 \mathrm{d}m_2 \; I_{\mathrm{DM}}(k,m_1) I_{\mathrm{DM}}(k,m_2)$$
$$\times P_{\mathrm{hh}}(k,r|m_1,m_2), \tag{12}$$

where

$$I_{\mathrm{DM}}(k,m) = n(m) u(k|m) \frac{m}{\bar{\rho}}. \tag{13}$$

~~Here, $P_{hh}(k,r|m_1,m_2)$~~ This equation can be confusing at first glance because it seems to be a function of both k and r – which are themselves not independent variables. We shall resolve this confusion presently. $P_{\mathrm{hh}}(k,r|m_1,m_2)$ is the power spectrum of halo centres for halos of mass m_1 and m_2. This is in general a complicated function of mass and scale, but the most common approach is to approximate it by a first-order linear bias of the matter power,

$$P_{\mathrm{hh}}(k,r|m_1,m_2) \approx b(m_1,r) b(m_2,r) P_{\mathrm{m}}(k), \tag{14}$$

where $b(m,r)$ is the first-order bias at m (cf. §2.6), possibly with a scale-dependent correction, and ~~$P_m(k)$~~ $P_{\mathrm{m}}(k)$ is the matter power spectrum (cf. §2.2)[13]. ~~Typically,~~ Depending on the application, it is not uncommon (eg. Cooray and Sheth, 2002) to adopt for P_{m} empirical nonlinear corrections on the linear matter power spectrum from HALOFIT (Smith et al., 2003)~~are used for this quantity.~~[14]. The motivation for using the nonlinear matter power here is to obtain better

agreement with simulations at quasi-linear scales, where the power is suppressed by a number of effects, including mutual exclusion of individual halos and the breakdown of linear theory (understandably) at the precise scales at which halos are forming (Mead and Verde, 2021). Unfortunately, these somewhat circular corrections are not self-consistent – adding contributions from the 1-halo term into the power for the 2-halo term. Thus corrections made in the scale-dependent bias (which fundamentally should account for the halo exclusion, cf. 2.9) are then forced to do two jobs: to model the true effect of halo exclusion, and to undo the fudged nonlinear corrections. In principle, it is better to keep these effects separate, and deal self-consistently with the latter when modelling P_{m}. Thus, studies of pure matter correlations most often adopt either the linear matter power spectrum, or corrections from perturbation theory or similar (eg. Seljak and Vlah, 2015; Mead and Verde, 2021). These extensions partially correct the quasi-linear regime in a self-consistent way. Still other studies use emulated halo-halo power spectra directly, and thus remove the need to model the one-to-two-halo transition entirely (eg. Nishimichi et al., 2021). We leave its definition free in our implementation [15] (with a default of 'linear'), and caution the user to pay close attention to its value for any particular application.

The general form for the 2-halo term then becomes

$$P_{\mathrm{DM}}^{\mathrm{2h}}(k,r) = \int\int I_{\mathrm{DM}}(k,m_1) I_{\mathrm{DM}}(k,m_2)$$
$$\times b(m_1,r) b(m_2,r) P_{\mathrm{m}}(k) \mathrm{d}m_2 \mathrm{d}m_1 \tag{15}$$

This is separable, so long as the integration limits are not inter-dependent (this is not the case in general, cf. §2.9), and in this simplest case (where we do not attempt to exclude correlations within halos) we have

$$P_{\mathrm{DM}}^{\mathrm{2h}}(k,r) = P_{\mathrm{m}}(k) \left[\int I_{\mathrm{DM}}(k,m) b(m,r) \mathrm{d}m \right]^2. \tag{16}$$

The right-most factor is often called the "effective bias" of matter, and under the halo model ansatz must by definition approach unity as $k \to 0$ (since matter is not biased against itself at large scales). In practice, several effects can artificially inhibit this behaviour, and we provide mechanisms for dealing with these in HALOMOD (cf. §4.2.1).

Note that all of these expressions denote the 2-halo term as a function of both k and r (as opposed to the 1-halo term which is just a function of k). This arises either due to the appropriate limits of the integration being dependent on the real-space size of halos, or because the halo bias may be scale-dependent. In cases when either of these is true, to recover the power spectrum as a function of k only, we Hankel-transform to $\xi(r)$ and then back to $P(k)$.

---

[13]Note that ~~thoughout~~ throughout, we use a subscript m to denote either the linear matter power or the nonlinear matter power derived using analytic corrections, as opposed to the full halo model.

[14]This is slightly circular, as the HALOFIT model itself is inspired by the halo model (and is tuned to emulate the clustering of dark matter within and between halos).

[15]~~This is slightly circular, as the HALOFIT model itself uses the halo model. However, this is a common practice in halo model calculations, eg. (Cooray and Sheth, 2002; Smith et al., 2011)~~

2.2. Matter Power Spectrum

The matter power spectrum $\cancel{P_m(k)}$ $P_m(k)$ characterises the distribution of matter density perturbations as a function of wavenumber k. When density fluctuations are small, so that $\delta \ll 1$, the statistical properties of the perturbations can be derived analytically via perturbation theory. Generally, it is assumed that the fluctuation spectrum was initially scale-free (i.e. a power-law), and the present day (linear) power spectrum is modified by a transfer function which captures the effects of the transition from a radiation-dominated to matter-dominated Universe (Bond and Efstathiou, 1984):

$$P_{\rm m,lin}(k) = A k^n T^2(k), \qquad (17)$$

where A is the normalisation constant and n is the spectral index. We follow convention and use the parameter σ_8, which measures the mass variance on a scale of $8 h^{-1}$Mpc at $z = 0$, to calculate A. The transfer function is particularly sensitive to the nature of the dark matter and the baryon density parameter $\Omega_{\rm b}$.

A commonly used dimensionless equivalent can be defined as

$$\Delta^2(k) = \frac{k^3}{2\pi^2} P(k), \qquad (18)$$

which is the contribution to the variance in logarithmic wavenumber bins.

The CDM transfer function has been modelled extensively in the literature. A number of functional forms have been proposed (Bond and Efstathiou, 1984; Bardeen et al., 1986; Eisenstein and Hu, 1998, hereafter EH), along with a series of *Boltzmann* codes, which calculate the transfer function from first principles with perturbation theory techniques (Zaldarriaga and Seljak, 2000; Lewis et al., 2000; Blas et al., 2011).

~~Figure 2~~ Fig. 2 shows the typical shape of the CDM linear power spectrum, while highlighting differences between forms for $T(k)$ found in the literature. Both small- and large-scales have small fluctuations, with a characteristic turnover at scales of $\sim 0.02 h/$Mpc. The low-k form asymptotes to $P(k \ll 0.02) \propto k^{n_s}$, while the high-$k$ behaviour is more complicated, not obeying a power-law relation. Only the Boltzmann code is able to fully capture the wiggles introduced by the baryon acoustic oscillations (BAOs), though the EH form including baryons closely follows up to $k \sim 2 h^{-1}$Mpc. In fact, all curves (except that of ~~Bond and Efstathiou (1984)~~ Bond and Efstathiou 1984) are within $\sim 10\%$ over the plotted range.

The calculation of the halo-centre power spectrum is commonly approximated by ~~Eq. 14~~ Eq. 14, i.e.

$$P_{\cancel{hh}\,{\rm hh}}(k, r | m_1, m_2) \approx b(m_1, r) b(m_2, r) P_{\cancel{m}\,{\rm m}}(k),$$

in which an estimate of the matter power spectrum is linearly biased. ~~Typically, a *nonlinear* estimate of the matter power is used, for which the corrections of Smith et al. (2003, hereafter HALOFIT) are employed~~ As previously discussed, it is not uncommon to
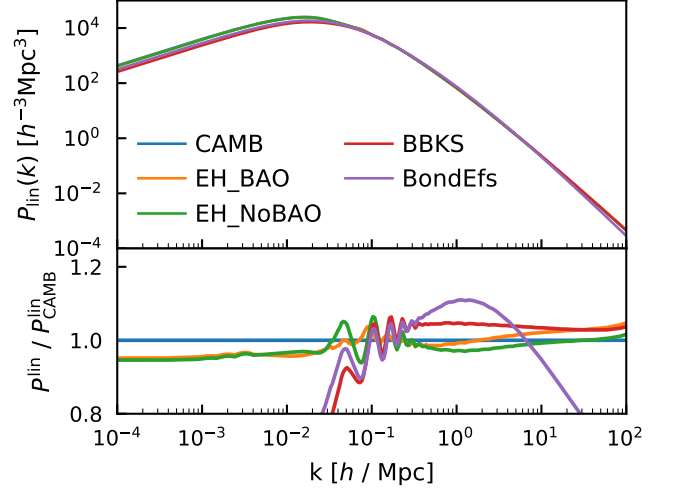

Figure 2: A selection of transfer function models, shown as linear power spectra, including analytic approximations (Bond and Efstathiou, 1984; Bardeen et al., 1986; Eisenstein and Hu, 1998) and Boltzmann solutions. Details for each model (referenced to the alias given in the legend) can be found in Table 3. The general shape is reproduced in each case, with details of baryon acoustic oscillations differing. In the bottom panel, we show the relative fraction compared to the CAMB transfer function, which renders all models without baryons as having oscillations on the acoustic scales. Note that the models are normalized to the same value of σ_8 (though some models, like BBKS, appear to be low across most of the range, their brief excursion above CAMB in the centre is highly weighted).

use the non-linear power spectrum from HALOFIT to model P_m. We do not present the details here, except to mention that the modifications are applied on large scales as ratios of $P_{\rm m,lin}(k)$, and on smaller scales via empirical fitting functions, which are obtained via arguments motivated by the halo model. These modifications are expressed as a series of equations with coefficients fitted to numerical simulations. Updated coefficients were introduced in Takahashi et al. (2012). Note also that the normalisation of $P_{\rm m,nl}(k)$ is identical to $P_{\rm m,lin}(k)$, and is calculated using the linear form, since σ_8 refers to the linear power spectrum. The effect of introducing non-linearities is to increase small-scale power, with the recent coefficient updates slightly enhancing this effect.

2.3. Mass Definition

The halo model centres on the 'halo' as its basic unit. One of the most fundamental properties of the halo is its mass, which is required to define the density profile, concentration-mass relation and its bias with respect to background matter, but there is no unique way in which to define this property.

The two dominant approaches to defining the halo mass are the *Friends-of-Friends* (FOF) and *Spherical Overdensity* (SO) definitions. The FOF definition is useful in the context of simulations, in which we have known particle locations. These particles are linked together into a halo if their separation is less than some chosen linking length,

and the fully linked network provides a halo of arbitrary shape.

The SO algorithm finds peaks in the density field and grows spheres around them until the contained density drops below a threshold density. All particles within the final sphere are considered part of the halo. This definition is more useful in the context of observations (of clusters, for example) as it can be applied to the measured (projected) density fields[15].

Within each definition approach exists the potential for an infinite number of explicit definitions. The FOF approach has the linking length parameter (typically denoted b), whilst the SO approach has the overdensity threshold, Δ_b. Moreover, the conversion between FOF and SO masses is difficult because a linking length does not uniquely correspond to an overdensity (More et al., 2011). Indeed, even conversion of relevant quantities (eg. mass functions, concentration-mass relations and bias functions) between different SO definitions is only ever approximate, due to practical difficulties in identifying halos in simulations. Accuracy of such conversions is discussed for concentration-mass relations in Diemer and Kravtsov (2015), who give an uncertainty of 15-20% at high mass, and for mass functions in Bocquet et al. (2016) whose fits imply inaccuracies of 20-40% below halo masses of $10^{15} M_\odot h^{-1}$.

While it is thus difficult to compare, even in principle, between FOF and SO definitions, comparison of halo model quantities between explicit SO definitions is approximately possible as long as one has a model for the density distribution within a halo (cf. §4.3.7). Armed with an average spherical halo profile, one can compute the average mass of a halo under a different density threshold (clearly higher density thresholds resulting in lower masses) (cf. §4.3.3). We will use the standard symbols m and R for the halo mass and radius throughout this paper, typically without specifying which definition is being applied (most equations are agnostic of this definition).

2.4. Variance and Spatial Filter

The halo model depends on knowing the overdensity (and variance) associated with a position $\vec{x}$. The mass variance is defined over some smoothing scale, R_L:

$$\sigma^2(R_L(m)) = \frac{1}{2\pi^2} \int_0^\infty k^2 P_{\mathrm{m,lin}}(k) W^2(kR) dk. \quad (19)$$

$W(x)$ here denotes a spatial filter. In particular, halos arise from patches in the primordial density field with some shape, described by a function $w(\mathbf{x})$. While the shape of the filter may be arbitrary (and should be matched to the typical shapes of the primordial patches that give rise to the halos (Dalal et al., 2010; Chan et al., 2017; Diemer and Joyce, 2019)) the most common choices are a spherical top-hat (either

in real or Fourier space) or a Gaussian. Under the assumption that the chosen filter is spherically symmetric (not the case in general, but assumed in HALOMOD), the Fourier-transform of the filter is given by the spherical-hankel transform:

$$W(k) = 4\pi \int_0^\infty w(r) r^2 j_0(kr) dr. \quad (20)$$

Having defined a spatial filter on the primordial density field, one can proceed to define the Lagrangian radius, $R(m)$, which is the radius of a spherical region of homogeneous space containing the mass m (or more generally, the scale of a particular filter function required to contain the mass, cf. §4.3.4). For the standard top-hat, this is

$$R_L(m) = \left(\frac{3m}{4\pi \bar\rho_0} \right)^{1/3}, \quad (21)$$

with $\bar\rho_0$ the mean density of the Universe at the present epoch.

The "mass variance", $\sigma^2(R_L)$ is thus the variance on a scale from which the matter in halos is thought to have originated. Notably, the mass variance in spheres of Lagrangian radius 8 Mpc/h yields the power spectrum normalisation, σ_8.

2.5. Mass Function

The mass function quantifies the number of dark matter halos per unit mass per unit comoving volume of the Universe. The most commonly adopted theoretical formulation of the mass function is via the extended Press-Schechter (EPS) formalism (Press and Schechter, 1974; Bond et al., 1991), in which density peaks form through gravitational instability and evolve into halos when they cross a density threshold, δ_c. In this formalism, the mass function is a universal function of the peak-height, $\nu = \delta_c/\sigma$[16], which captures the variation from cosmological parameters and the redshift (Jenkins et al., 2001, but note that several recent studies find significant non-universality with redshift, and have provided fits that explicitly account for this departure).

The mass function can be defined as

$$n(m) \equiv \frac{dn}{dm} = \frac{\rho_0}{m^2} f(\sigma) \left| \frac{d\ln\sigma}{d\ln m} \right|, \quad (22)$$

or using the peak-height, as

$$\frac{dn}{dm} = \frac{\rho_0}{m} f(\nu) \frac{d\nu}{dm}. \quad (23)$$

For an Einstein-de Sitter universe (in which the matter density equals the critical density and there is no dark

---

[15]Note that the 3D SO definition is not *immediately* applicable to observations without forward-modelling and scaling relations, but they are far more applicable than FoF masses.

[16]Note that some authors use $\nu = \delta_c^2/\sigma^2(m)$, i.e. the square of our definition.

energy), $\delta_c \approx 1.686$, and its value is very weakly dependent on cosmology[17] The function $f(\sigma) \equiv \nu f(\nu)$, denoting the fraction of matter collapsed into halos of peak-height in a logarithmic bin around ν, has been modelled extensively in the literature, and is generally fit to numerical simulations[18].

It can be advantageous (but not strictly necessary) to choose a fitting function which is normalisable, ie.

$$\int f(\nu)\mathrm{d}\nu = 1. \tag{24}$$

Several fits in the literature obey this relation (eg. Sheth et al., 2001a; Peacock, 2007; Tinker et al., 2010), which ensures that all mass is bound in halos. This is particularly attractive for HM calculations, in which it is consistent with the basic assumptions of the framework.

Note also that from ~~Eq. 23~~ Eq. 23 we can determine $d\nu$ as a function of dm, and integrals of the form $\int n(m)g(m)dm$ may be re-written

$$\int g(\nu(m))\frac{\rho_0}{m}f(\nu)\mathrm{d}\nu. \tag{25}$$

In particular, the sum of all mass in the universe requires setting $g(m) = m$, and the requirement that the mass density in halos equal ρ_0 is easily seen to be equivalent to the condition ~~Eq. 24. Integrating Eq. 25~~ Eq. 24. Integrating Eq. 25 can often be numerically advantageous, since relevant bounds on $f(\nu)$ can often be found independently of cosmology and redshift.

~~Figure 3~~ Fig. 3 shows the HMF for ~~four~~ several of the fits available in HALOMOD (via HMF). According to these fitting functions, the HMF behaves as a power-law at low masses, with an exponential cut-off above a typical mass $m_* \approx 5 \times 10^{14}$ (this mass changes with redshift). There is clearly significant variations between the fits, especially at high masses, for a range of reasons, including halo finder differences, alternate mass definitions, and simulation resolution (cf. Murray et al., 2013b; Knebe et al., 2011).

2.6. Bias

It is well-known that dark matter halos tend to trace the underlying density field in a biased manner (Cole and Kaiser, 1989, eg.). In principal, the form of this bias can be of arbitrary order (as a function of the matter overdensity), non-local (i.e. dependent on the surrounding environment) and stochastic. However, it is most common within studies involving the halo model to consider only the first-order, local, deterministic bias as a function of the halo mass (Mo and White, 1996):

$$\delta_h \Delta_\mathrm{h}(\bar{x}, m) = b(m)\delta(\bar{x}). \tag{26}$$

Figure 3: ~~Seven~~ Several HMFs from the literature, including the 'original' form of Press and Schechter (1974, 'PS'), motivated by spherical collapse, and SMT motivated by ellipsoidal collapse from Sheth et al. (2001a). Note that the mass definitions and cosmologies assumed by the fits are rather disparate – this is not meant to be an apples-to-apples comparison. While $\nu f(\nu)$ is almost independent of cosmology (Jenkins et al., 2001), and in principle independent of mass definition (peak-height is based on the initial Lagrangian field), in practice $f(\nu)$ is measured from simulations which must employ some halo-finding algorithm and mass definition.

Peaks theory is able to make predictions of the form of $b(m)$, by considering the probability of collapse of a region of given smoothing scale m in the context of an altered density threshold $\delta_c - \delta(\vec{x})$, often called the peak-background split method (for more details, see eg. Bond et al., 1991; Zentner, 2007; Tinker et al., 2010; Manera et al., 2010). These predictions are specified in terms of the peak-height parameter ν, and are therefore expected to exhibit universality with respect to cosmology and redshift.

In the spherical collapse paradigm, the bias is intimately related to the mass function (Cole and Kaiser, 1989; Cooray and Sheth, 2002),

$$b(m) = 1 - \frac{\partial}{\partial\delta_c}\ln n(m), \tag{27}$$

which when combined with ~~Eq. 22~~ Eq. 22, gives

$$b(m) = 1 - \frac{1}{\delta_c} - \frac{\partial}{\partial\delta_c}\ln f(\nu). \tag{28}$$

In this formalism, if all mass is contained within halos, according to ~~Eq. 24~~ Eq. 24, then by construction, the following consistency relation holds,

$$\int b(\nu)f(\nu)\mathrm{d}\nu = 1, \tag{29}$$

which merely says that the average bias is unity, so that matter is not biased against itself[19]. Note that if Eq. 29

---

[17]Note that even small changes in δ_c can affect the high-mass end of the HMF significantly. However, many authors assume $\delta_c = 1.686$ regardless of cosmology and thus many fits of the HMF are defined under this assumption.

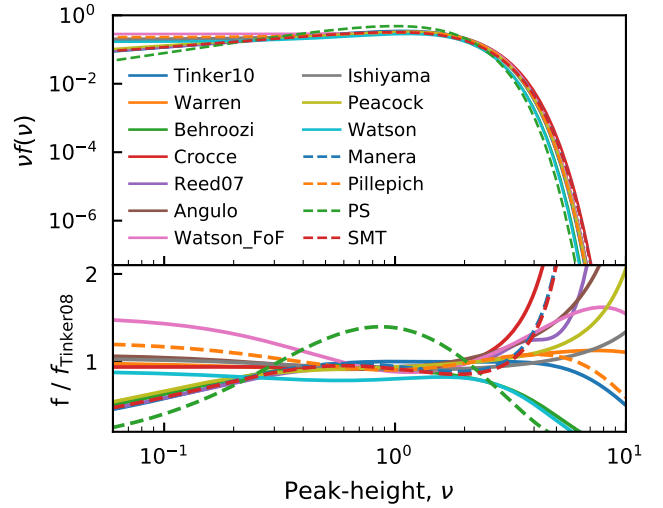[18]Note that some authors use $f(\nu)$ to describe what we have defined as $\nu f(\nu)$.

[19]Non-unity average bias in the context of the halo model would be an inconsistency, as all halos contain all the mass, and the mass cannot be biased, on average, against itself.

holds, the large-scale ($k \to 0$) matter power is gauranteed to be equivalent to P_m (which just says that the effective bias at large scales is unity).

In the approximation of spherical collapse, under the original model of (Press and Schechter, 1974), we have $f(\nu) \propto \exp(-\nu^2/2)$, which yields a bias of (Cole and Kaiser, 1989; Mo and White, 1996)

$$b(\nu) = 1 + \frac{\nu - 1}{\delta_c}. \quad (30)$$

The general qualities of the spherical collapse form are (recall that $\nu = 1$ at $M_\star \sim 5 \times 10^{12}$ at $z = 0$, and corresponds the mass of halos that are, on average, collapsing at a given redshift):

- a transition from low-mass anti-bias to high-mass bias at $\nu = 1$

- a small-scale asymptotic limit of $b = 1 - 1/\delta_c \approx 0.4$

- a large scale asymptotic behaviour of $b \propto \nu$.

Given the simplistic nature of the spherical collapse model, it is remarkable that this bias function roughly reproduces the results of N-body simulations. However, in detail, this form over-predicts the bias at high masses ($\nu >\sim 1.5$), and consequently slightly under-predicts the bias for low-mass halos.

The approximations induced by the spherical collapse assumption have been relaxed in several studies, most notably by Sheth et al. (2001a), who obtain a form motivated by ellipsoidal collapse, but calibrated with simulations. Forms motivated by stochastic barriers (Corasaniti and Achitouv, 2011), non-Markovian barriers (Ma et al., 2011) and local density-maxima (Paranjape et al., 2013b) are more recent developments.

Unfortunately, the spherical collapse paradigm leads to an artificial enhancement of the bias for halos with $\nu <\sim 1$ $\nu \lesssim 1$ (Manera et al., 2010; Tinker et al., 2010), due to the fact that low-mass halos are increasingly elliptical. This has resulted in a proliferation of direct empirical fits to high-resolution N-body simulations (Jing, 1998, 1999; Seljak and Warren, 2004; Tinker and Weinberg, 2005; Mandelbaum et al., 2005; Pillepich et al., 2010). Motivated by the broad success of spherical collapse, these fits are usually (but see Jing, 1998; Seljak and Warren, 2004) still formulated in terms of the peak-height (indeed, often they utilise the forms yielded by the peak-background split, with updated parameters based on direct fits to simulations).

In figure 4 Fig. 4 we show a sample of linear scale-independent bias relations found in the literature and included in HALOMOD. We note the similarity of high-mass behaviour, though the high-mass amplitude is substantially modified between several of the forms.

2.6.1. Scale-dependence

In reality, the assumption that the bias is a function only of halo mass, to the exclusion of environment, is
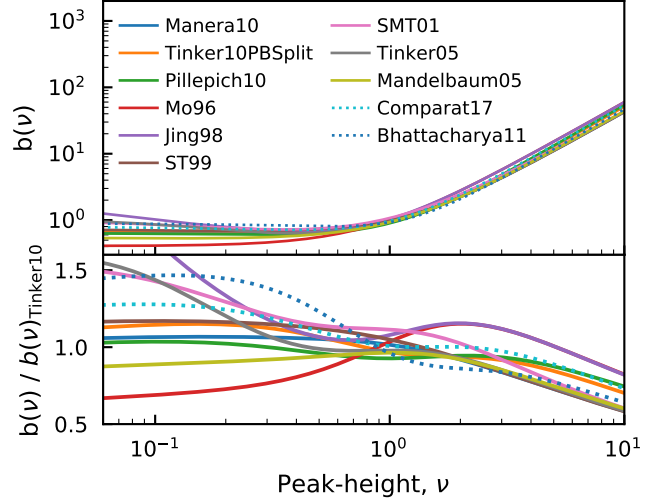

Figure 4: Several linear bias functions from the literature. Most functions, including Mo and White (1996), Sheth and Tormen (1999), Sheth et al. (2001a) and those with the same form (Mandelbaum et al., 2005; Tinker and Weinberg, 2005) have high-mass asymptotic behaviour of $b \propto \nu$. The (default) form of Tinker et al. (2010) is an empirical fit with an arbitrarily chosen form, and has a high-mass asymptotic limit of $b \propto \nu^{1.2}$. The models plotted in dotted lines are from COLOSSUS, called by HALOMOD using its explicit interface

overly simplistic (Sunayama et al., 2015). The bias depends not only on the masses of the paired halos, but also the scale over which we consider the correlations. The scale-dependence of halo bias is a complex phenomenon, which has received a great deal of attention (eg. Paranjape et al., 2013a; Lapi and Danese, 2014; Poole et al., 2015). Foregoing these complexities, simple empirical models specify the scale-dependence as a function $S(\xi_m(r)), S(\xi_m(r)),$

$$b^2(m, r) = b^2(m) S(\xi_{m\,\mathrm{m,nlin}}(r)). \quad (31)$$

An example form for S is given by Tinker and Weinberg (2005) as

$$S(r) = \frac{[1 + 1.17\xi_m(r)]^{1.49}}{[1 + 0.69\xi_m(r)]^{2.09}} \frac{[1 + 1.17\xi_{m,\mathrm{nlin}}(r)]^{1.49}}{[1 + 0.69\xi_{m,\mathrm{nlin}}(r)]^{2.09}}. \quad (32)$$

At large scales, where $\xi \ll 1$, the scale-dependence is negligible, but smaller scales can receive significant ($\sim 50\%$) suppression. The correlation function used here is the non-linear matter correlation function from HALOFIT.

2.7. Halo Density Profiles

It is widely accepted that the density profiles of DM halos are more or less self-similar if they are rescaled by a suitable central density and radial scale. These two parameters can then be expressed more intuitively as a mass and a concentration. In this section, we define our nomenclature for density profiles. Our general, systematic

formulation enables the user to elegantly insert specific functional forms. In particular, the average density profile follows the form

$$\rho(r|m,z) = \rho_s(c,\bar\rho) \times f(r/r_s), \qquad (33)$$

where $\rho(r|m,z)$ has units of density, and $c(m,z) = r_{\Delta_h}/r_s$ is the *concentration parameter*, which we shall treat as an explicit function of the halo mass, redshift, and cosmology (although it exhibits appreciable scatter around any such relationship). Here, $\bar\rho$ is the mean matter density of the universe at redshift z. The *scale radius* r_s is often defined as the radius where the logarithmic slope of the profile is -2, but other definitions are possible in principle. The factor ρ_s controls the amplitude of the profile and is set by the known mass given the halo's radius (here we use halo radius generically as r_{Δ_h}, where Δ_h is the halo overdensity with respect to the background). In particular, it is always true that the halo radius and mass are related by

$$m = \frac{4\pi r_{\Delta_h}^3}{3}\bar\rho\Delta_h. \qquad (34)$$

We may consider halos to have either a 'hard' or 'soft' edge; in the former case, the extent of the halo is precisely at r_{Δ_h}, and the density abruptly falls to zero beyond this. In the latter case, the halo extends to infinity. The latter case is only well-defined in the case that the total integrated mass converges. The HM generally considers halos with a hard boundary, but HALOMOD includes models for both, and we will present the equations for 'non-truncated' halos as well.

While ρ_s does not directly depend on mass in our formulation, the dependence on the mean density and the secondary dependence of c on mass translate into a relation where more massive and earlier-forming halos have higher central densities. The $f(r/r_s)$ term controls the shape of the profile and is a function of the scaled radius $x = r/r_s$. Throughout the following discussion we shall often refer to a 'reduced' form of the profile (or related quantities) in which simple normalisation factors are removed. An example of such a reduced form is f in Eq. 33.

2.7.1. Truncated Halos

For halos truncated at the halo radius, the amplitude of the profile is set by the total mass inside the halo radius, which we recover by integrating Eq. 33:

$$m = 4\pi\rho_s(c)r_s^3 \int_0^c x^2 f(x)\,dx, \qquad (35)$$

where we have made the substitution $r = xr_s$ in which $x(r_{\Delta_h}) = c$. If we set

$$h(c) = \int_0^c x^2 f(x)\,dx \qquad (36)$$

then we obtain

$$\rho_s(c) = \frac{m}{4\pi r_s^3 h(c)} \equiv \frac{c^3\Delta_h}{3h(c)}\bar\rho \qquad (37)$$

Thus for all profiles, $h(c)$ uniquely defines the amplitude.

The calculation of 2-point statistics involves two more functions derived from the halo density profile: the normalised Fourier transform $u(k|m,z)$ and the self-convolution $\lambda(r|m,z)$. The normalised Fourier transform of the halo profile is defined as (Cooray and Sheth, 2002)

$$u(\mathbf{k}|m,z) = \frac{1}{m}\int \rho(\mathbf{r}|m,z)e^{-i\mathbf{r}\cdot\mathbf{k}}\,d^3\mathbf{r} \qquad (38)$$

in which $\mathbf{k},\mathbf{r}$ are the Cartesian wave-vector and position vector respectively. In the case of a spherically symmetric distribution truncated at the halo radius (as we assume) this may be treated by a Hankel transform, in which we use $\kappa = kr_s$, and x as previously defined:

$$u(\kappa|m,z) = \frac{4\pi r_s^3}{m}\rho_s(c)\int_0^c x^2 \frac{\sin(\kappa x)}{\kappa x} f(x)\,dx. \qquad (39)$$

Using Eq. 37 we have

$$u(\kappa|m,z) = \frac{p(\kappa,c)}{h(c)} \qquad (40)$$

where

$$p(\kappa,c) = \int_0^c x^2 \frac{\sin(\kappa x)}{\kappa x} f(x)\,dx \qquad (41)$$

is the reduced form.

The self-convolution of the density profile is

$$\lambda(x|m,z) = \rho_s^2(c)r_s^3 l(x,c) \qquad (42)$$

and has units of M^2/V. The reduced form is given by

$$l(x,c) = \int f(\mathbf{y},c)f(|\mathbf{y}+\mathbf{x}|,c)\,d^3y, \qquad (43)$$

in which x and y are in units of r_s as usual. It is generally difficult, if not impossible, to solve this integral analytically. However, there are several profiles for which do afford a solution, including the popular profile of Navarro et al. (1997, hereafter NFW).

Thus, any profile is fully defined by its reduced profile shape, $f(x)$, which can be used to produce the dimensionless quantities $h(c)$, $p(\kappa,c)$, and $l(x,c)$. We utilise this set of variables in the `halomod` implementation.

2.7.2. Non-truncated Profiles

In the case that the profile is not truncated at r_{Δ_h}, we still define $c \equiv r_{\Delta_h}/r_s$, and assume Eq. 34 relates halo mass to r_{Δ_h}. The integrated mass of the halo is defined by Eq. 35, but the upper

integration limit is set to $+\infty$. This integral must converge for the given profile in order for it to be well-defined as a non-truncated profile.

For non-truncated profiles, neither h, p or l are dependent on the concentration parameter, and for each, the integration is performed to $+\infty$. Nevertheless, the overall profile is still scaled by the concentration and central density.

2.8. Concentration-Mass Relation

Given the logic of self-similar fitting functions for the density profile laid out in the previous section, it is clear that the c–m relation plays a critical role: if we can find a universally valid description of $c(m)$, the halo density profile can be determined solely based on the mass. However, concentration has long been known to depend on mass, redshift, and cosmology in a complex fashion, leading to a proliferation of models that fall into roughly three categories.

First, NFW suggested that concentration is intimately related to the assembly history of halos, which can be summarized as their age or formation time. Physically, this dependence can be understood by the transition from the fast to the slow accretion regime, after which the halo boundary (and its concentration) grow mostly via pseudo-evolution while the scale radius remains more or less constant (Navarro et al., 1997; Bullock et al., 2001; Bosch, 2002; Wechsler et al., 2002; Zhao et al., 2003; Dalal et al., 2010; Diemer et al., 2013; Ludlow et al., 2013; Wang et al., 2020). This logic has motivated numerous models based on some quantification of the age of halos (e.g., Eke et al., 2001; Zhao et al., 2009; Giocoli et al., 2012; Ludlow et al., 2014, 2016; Correa et al., 2015).

While age-based models follow physical insight, apply to individual halos, and are often universal (meaning they apply to all masses, redshifts, and cosmologies), they can be computationally complicated. Thus, a second popular way of modeling the c–m relation is as a series of power laws

$$c(m_{\Delta_h}, a) = A \left(\frac{m}{M'} \right)^B a^C, \quad (44)$$

which are typically valid only over the particular mass and redshift range where they were calibrated (e.g., Dolag et al., 2004; Duffy et al., 2008; Macciò et al., 2008; Bhattacharya et al., 2013; Dutton and Macciò, 2014; Klypin et al., 2016; Child et al., 2018). Moreover, each calibration is valid only for a particular cosmology, which poses a significant barrier for purposes such as halo modeling.

Recently, a third avenue for modeling concentration has emerged. Prada et al. (2012) noticed that the redshift evolution of the c–ν relation is much smaller than that of the c–m relation; they parameterized the differences with a fitting function. Subsequent work has shown that the residual evolution can be understood as additional, weaker dependencies on parameters that break the universality of structure formation in a ΛCDM universe, such as the power

spectrum slope, n_{eff}, and the evolution of the linear growth factor, α_{eff} (Diemer and Kravtsov, 2015; Diemer and Joyce, 2019; Ishiyama et al., 2020). While similar parameters have been considered previously (e.g., Bullock et al., 2001; Eke et al., 2001; Zhao et al., 2009), the new models explicitly avoid reliance on any other variables. For example, when expressing concentration as $c(\nu, n_{\mathrm{eff}}, \alpha_{\mathrm{eff}})$, any dependence in cosmology is implicitly taken into account.

2.9. Halo exclusion

On small scales, the 2-halo term as outlined in §2.1.2 and integrated over all halo masses naturally counts pairs between theoretically overlapping haloes. In reality, correlations at these scales are much more likely to arise from within the one halo (or, alternatively put, particles within a halo's radius in a simulation are typically assumed to be a part of that halo). Modelling halo exclusion in the 2-halo term attempts to account for this fact, so that these pairs aren't double-counted in both the 1-halo and 2-halo term.

In practice, this affects the correlation function at the crossover between the 1-halo and 2-halo terms at the 30% level (Schneider et al., 2012), by suppressing the 2-halo term.

2.9.1. Empirical approaches

There have been a number of proposed models to account for this effect. At a purely empirical level, one may consider modifying $P_{hh}(k)$. While it is customary to use a (biased) nonlinear estimate of the matter power, Cooray and Sheth (2002) suggest that using the linear matter power will decrease the power at small scales, therefore increasing the fidelity of the model. In a similar way, Schneider et al. (2013) use a smoothing of the power on roughly transition scales to eliminate the excess correlations:

$$P_{hh}(k, m_1, m_2) \approx b(m_1)b(m_2)W(kR_T)P_{m,nl}(k), \quad (45)$$

where $P_{m,nl}(k)$ is the matter power from HALOFIT, W is the Fourier transform of a top-hat in real space,

$$W(x) = 3 \frac{\sin x - x \cos x}{x^3}, \quad (46)$$

and $R_T \approx 2 \mathrm{Mpc} h^{-1}$. This latter argument can be used to reduce errors to below 10%.

2.9.2. Physical approaches

By far the more common approach is to set upper limits (say m_1' and m_2') on the 2-halo integral, Eq. 15, using physical arguments. This approach has been described in some detail in Tinker and Weinberg (2005) (hereafter T05), and we briefly reproduce that discussion here. While we follow T05 and discuss the galaxy quantities, precisely the same arguments hold for DM quantities with the replacement $I_g \rightarrow I_{\mathrm{DM}}$ (and $\bar{n}_g \rightarrow \bar{\rho}$).

The underlying idea is to set m_1' and m_2' such that the scale in question is not smaller than the sum of the halo radii, i.e. $r \geq R_{\mathrm{vir}}(m_1') + R_{\mathrm{vir}}(m_2')$.

Note that in this approach, the density is also modified, so that

$$\bar{n}_g'^2 = \int_0^{m_1'} \int_0^{m_2'} \prod_{i=1}^{2} n(m_i) N_{tt}(m_i)\,dm\,dm_i. \quad (47)$$

In this case, the final 2-halo correlation function needs to be re-scaled by

$$\xi_{g\ gg}^{2h\,2h}(r) = \left(\frac{\bar{n}_g'}{\bar{n}_g}\right)^2 \left[1 + \xi_{g\ gg}'^{2h\,'2h}(r)\right] - 1. \quad (48)$$

The simplest model is to assume spherical halos, and that the two halos in question are the same size, so that (Zheng et al., 2005):

$$m_1' = m_2' = \frac{4}{3}\pi \left(\frac{r}{2}\right)^3 \Delta_{hh}\bar{\rho}. \quad (49)$$

This is efficient since the limits are independent, and we can use the separated form of Eq. 16. However, it severely under-counts pairs due to edge effects.

We can do better by relaxing the assumption that the two halos are the same size. In this case, m_1' and m_2' are related by

$$R_\Delta(m_2') \quad = \quad r - R_\Delta(m_1). \quad (50)$$

In this case, we cannot separate the integrals, and so double-integration must be performed, which is relatively inefficient.

It is well-known that halos are typically triaxial (Bullock, 2001; Taylor, 2011; Zemp et al., 2011), and therefore one can improve the results by considering the probability that each pair is in different halos, given an empirical distribution of axis ratios. In this case, Eq. 15 Eq. 15 is augmented by the distribution:

$$\frac{P_g^{2h}(k,r)}{P_m(k,r)} \frac{P_g^{2h}(k,r)}{P_m(k,r)} = \int \int \prod_{i=1}^{2} I_{gg}(k,m_i) b(m_i,r)\mathcal{P}(x)\,dm\,dm_i. \quad (51)$$

Here, $\mathcal{P}(x)$ is the probability that pairs at a given separation are in separate halos, where $x = r/(R_\Delta(m_1) + R_\Delta(m_2))$. T05 suggest using

$$\mathcal{P}(y) \quad = \quad 3y^2 - 2y^3 \quad (52)$$
$$y \quad = \quad \frac{x - 0.8}{0.29} \quad (53)$$

where $\mathcal{P}(y < 0) = 0$ and $\mathcal{P}(y > 1) = 1$. This is a significant improvement over the spherical case, however it still requires double-integration (at every value of k!) due to the appearance of m_1 and m_2 in $\mathcal{P}(x)$.

T05 propose a solution to the efficiency problem by performing one double-integral to calculate $\bar{n}_g'$ for the full ellipsoidal case (where $\mathcal{P}(x)$ also augments Eq.47). From here, one can "match" the number density numerically with the simpler single-integral:

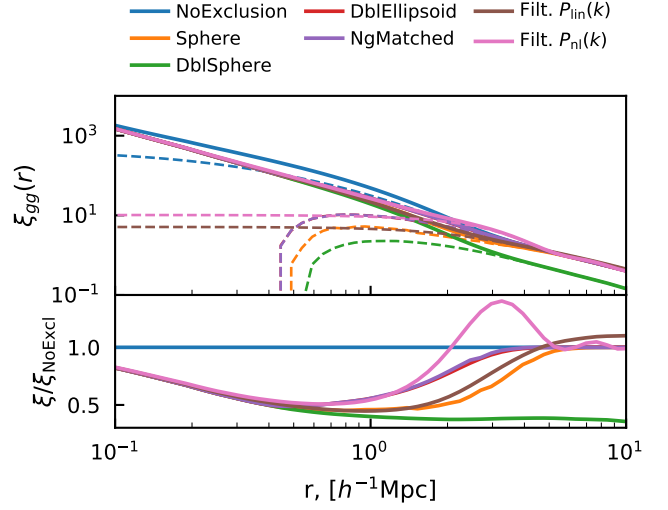$$\bar{n}_g' = \int_0^{m'} n(m) N_{tt}(m)\,dm\,dm, \quad (54)$$

Figure 5: Effects of various halo exclusion models around the transition region. Solid lines are the full $\xi_{gg}(r)$ $\xi_{gg}(r)$, while the dashed lines show the 2-halo term. In both cases where the halo-centre power spectrum is modified (brown and pink curves), the halo exclusion is not performed.

cumulatively integrating until the integral matches $\bar{n}_g'$, thus acquiring m'. Setting $m_1' = m_2' = m'$, one can use the separated form, Eq. 16 Eq. 16, to generate the final galaxy power spectrum.

We show the effect of several of these models in figure 5 Fig. 5. The spherical exclusion is the most effective at reducing pairs, with the n_g-matched method exhibiting a more gentle suppression. Interestingly, merely using the linear power without any exclusion is the closest to the n_g-matched case at the transition scales.

Note that the recent model of Garcia et al. (2020), in which halo exclusion is included explicitly via a soft halo radius, is not yet included in HALOMOD.

3. Extending the Halo Model to Galaxies

3.1. HOD models

Before we can extend the previous theory to a treatment of tracers (eg. optical galaxies), we must have a model for the expected occupation of the tracer within a given halo, $P(T|m)$, termed the halo occupation distribution (for discrete tracers such as galaxy counts, this is typically written $P(N|m)$ and must be a discrete distribution). Note that this model is assumed to depend solely on the halo mass, which perhaps surprisingly captures the majority of the behaviour of interest. There is a known second-order dependence on environment, the so-called 'assembly bias' (Sunayama et al., 2015), but we do not consider it in this work.

Any HOD model is principally composed of two parts – a parametrisation of the mean occupation of halos of a

given mass, $\langle N_{\rm t}\rangle_m$[20], and a distribution about this mean ~~$P(N_t|\langle N_t\rangle_m)$~~ $P(N_{\rm t}|\langle N_{\rm t}\rangle_m)$ (note that throughout this work, we imply that any values $\langle N\rangle$ are functions of mass, and drop the notation from here on). The mean occupation plays the major role in providing a form for the mass dependence of typical galaxy abundances, while the distribution is crucial in calculating pair (and triplet etc.) count statistics.

Historically, the parametrisation of the mean occupation began as a single function, but early work (Kauffmann et al., 2004; Zheng et al., 2005; Zehavi et al., 2005) showed an advantage in treating 'central' and 'satellite' galaxies independently. In hydrodynamical simulations, these are empirically identifiable classes of galaxies, and semi-analytic methods also treat them separately. Central galaxies (in optical) tend to be large and bright, and occupy the region near the halo's central potential well. Satellite galaxies are less bright, and tend to follow the halo's density profile[21]. Note that these qualitative descriptions are not necessarily true for all samples of tracers, but it typically remains useful to separate occupation of the tracer into such central and satellite classes.

Finally then, we have a mean occupation function for each class, and the total occupation is their sum:

$$\langle N_{\rm t}\rangle = \langle N_{\rm c}\rangle + \langle N_{\rm s}\rangle. \tag{55}$$

Identifying these classes as independent is helpful in determining the distributions about the means. Halos may only contain a single central galaxy (or none at all if the sample selection does not allow it). Thus their distribution is simply Bernoulli (or Binomial with a single trial):

$$N_{\rm c} \sim {\rm Bern}(\langle N_{\rm c}\rangle). \tag{56}$$

Satellite galaxies have been found to approximately follow Poisson distributions (Kauffmann et al., 2004; Zheng et al., 2005):

$$N_{\rm s} \sim {\rm Pois}(\langle N_{\rm s}\rangle). \tag{57}$$

Most studies also impose what we will term the 'central condition'. Explicitly, the condition states that ~~$N_c = 0 \Rightarrow N_s = 0$~~ $N_{\rm c}=0 \Rightarrow N_{\rm s}=0$ (i.e. no central galaxies in a halo implies no satellites). It arises from the fact that central galaxies are more luminous than their satellite counterparts, and therefore for any sample selection based on luminosity (or similar) thresholds, the central galaxy will be the first to be included. Our implementation does not enforce this condition (as there are conceivably sample selections which circumvent it), but does enable it, which simplifies some calculations.

---

[20]Note that we use notation consistent with discrete tracers throughout, but these functions are generalizable to non-discrete distributions.

[21]This is not strictly true – satellites may follow an alternate profile to the underlying dark matter (eg. flattening towards the center). The point is that they follow a profile. HALOMOD allows both the halo and tracer profiles to be specified independently.

For ~~two-point statistics, an important quantity~~ 2-point statistics, important quantities arising from the HOD ~~is~~ are the mean pair counts ~~, $\langle N_t(N_t-1)\rangle$~~ within a halo, which we address separately for centrals and satellites. We wish to derive ~~this quantity in a form~~ these quantities in forms explicitly dependent only on the mean occupation functions $\langle N_{\rm c}\rangle$ and $\langle N_{\rm s}\rangle$, for which we have specific parametrisations. ~~We begin by noting, after Zheng et al. (2005), that~~

$$\langle N_t(N_t-1)\rangle = \langle N_c(N_c-1)\rangle + \langle N_s(N_s-1)\rangle + \langle N_c N_s\rangle,$$

Furthermore, we treat self-pairs separately from cross-pairs, as self-pairs have different correlation properties. Noting that ultimately we will be comparing these pair counts to the mean number density squared (which counts each pair twice), we have:

$$N_{\rm cross-pair}^{\rm cen-cen} = 0 \tag{58}$$

$$N_{\rm cross-pair}^{\rm cen-sat} = 2\langle N_c N_s\rangle \tag{59}$$

$$N_{\rm cross-pair}^{\rm sat-sat} = \langle N_s(N_s-1)\rangle \langle N_s\rangle^2 \tag{60}$$

~~in which the first term on the right vanishes since N_c is either 0 or 1, and~~ where the last equality is made under the assumption ~~that N_s has a Poisson distribution, we can simplify to~~

$$\langle N_t(N_t-1)\rangle = \langle N_s\rangle^2 + \langle N_c N_s\rangle.$$

~~The last~~ of Poisson-distributed satellite counts. The cen-sat term is problematic (as it is not a function purely of mean counts of either central or satellite), however if we impose the central condition, then ~~$\langle N_c N_s\rangle = \langle N_s\rangle$~~ $\langle N_c N_s\rangle = \langle N_s\rangle$, since we need only count halos in which ~~$N_c = 1$. Thus~~ $N_c = 1$. Conversely, if the ~~central condition holds, we achieve the final form~~

$$\langle N_t(N_t-1)\rangle = \langle N_s\rangle(1 + \langle N_s\rangle).$$

~~If the~~ parameterisations for $\langle N_{\rm c}\rangle$ and $\langle N_{\rm s}\rangle$ do not impose the central condition then we can approximate, following Zehavi et al. (2005), with ~~$\langle N_c N_s\rangle = \langle N_c\rangle\langle N_s\rangle$~~ $\langle N_c N_s\rangle = \langle N_c\rangle\langle N_s\rangle$ given that $\langle N_{\rm s}\rangle \ll 1$ when $\langle N_{\rm c}\rangle < 1$. For $\langle N_{\rm c}\rangle = 0$ or 1, it is no longer an approximation, and thus for step-function parameterisations of $\langle N_{\rm c}\rangle$, it is completely accurate (for arbitrary parameterisations of $\langle N_{\rm s}\rangle$).

Finally, if the central condition is required, but the parameterisations of choice do not impose it (i.e. $\langle N_{\rm s}\rangle > 0$ when $\langle N_{\rm c}\rangle < 1$, or "satellites may exist in a particular halo where centrals do not"), one can manipulate the equations to enforce it using a common technique (eg. Beutler et al., 2013), in which we set $\langle N_{\rm s}\rangle = \langle N_{\rm c}\rangle\langle N_{\rm s}\rangle'$, where $\langle N_{\rm s}\rangle'$ is the original parameterisation. In this case, the truncation of $\langle N_{\rm c}\rangle$ forces the appropriate truncation of $\langle N_{\rm s}\rangle$. One must be careful in this method, however, to ensure that the HOD
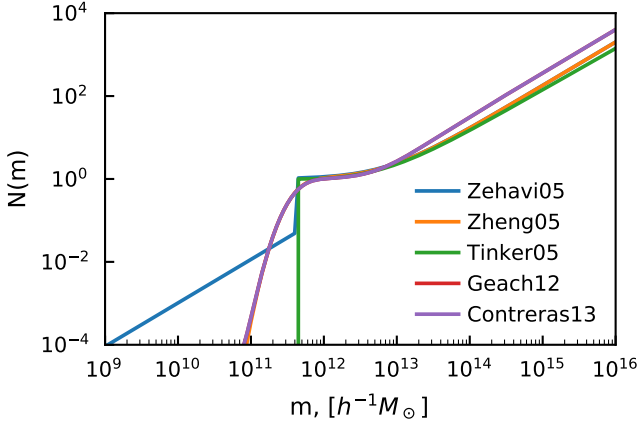
Figure 6: Total halo occupation for several parameterisations, highlighting the core similarities and differences between the models. Note that the central condition is false for the models shown.

parameters retain the same meaning as the original. For a step-function $\langle N_c \rangle$ this is not a problem, but there may be subtle differences induced for smoothed truncations (these are likely to be negligible for reasonable models).

Clearly, the number of self-pairs is simply $\langle N_t \rangle$.

There are several parameterisations for the HOD in the literature. In this work, and correspondingly in the `halomod` framework, we focus on those that utilise the separation of central and satellite galaxies. We present a summary of those included in HALOMOD in Table 8.

Most formulations contain the same essential features, so the most important features of the HOD are present in the simplest case. For the central term: a step function characterised by a step at the parameter M_{min}, which is heavily dependent on the sample selection. The deeper the survey, the lower this parameter will be. The satellite term is primarily characterised by a power law which quantifies the growth of galaxies with halo mass by the index α, and a scaling mass M_1 which is the mass at which an average halo in the sample contains a single satellite galaxy.

The various refinements to this general structure include smoothing the truncation of central galaxies assuming a log-normal distribution of halo masses at this threshold, and truncating the satellite power law at an arbitrary mass scale M_0. Further refinements can be specific to a certain kind of sampling, for example samples gained at different wavelengths. An illustrative plot of the core similarities and detailed differences can be found in ~~figure 6~~ Fig. 6.

3.2. Extension of framework to galaxies

The extension of equations 9, 10 and 16 to include galaxies is relatively simple under the assumption that all galaxies reside in halos, and their intra-halo abundance follows that of the density profile for satellites.

~~The primary modification of~~ For the 2-halo term, we make a simple modification to the DM equations ~~is to replace~~, replacing factors of $m/\bar{\rho}$ with ~~$N_t(m)/\bar{n}_g$, where $N_t(m)$~~

$N_t(m)/\bar{n}_g$, where $N_t(m)$ is the mean occupation function from the HOD (where for simplicity we omit angle brackets for the rest of this work), and $\bar{n}_g$ is the mean number density of galaxies,

$$\bar{n}_g = \int n(m) N_t(m)\, dm.\tag{61}$$

With this modification the 2-halo term is simply described by

$$I_g(k,m) = n(m) u(k|m) N_t(m)/\bar{n}_g,\tag{62}$$

which serves as a direct replacement for I_{DM} in equations 15 and 16. Note that this is not *quite* accurate given the separation of galaxies into centrals and satellites, because in this case the central galaxy is constrained to be at the center instead of following the profile (whereas Eq. 62 assumes all of the galaxies—including the central—are located following the density profile). Nevertheless, the error induced by this approximation is extremely small. Note further that no "self-pairs" are included in the 2-halo term, since it always considers two separate halos (thus it contains no "shot-noise").

The 1-halo term needs a little more care. ~~The primary modification gives~~

$$1 + \xi_g^{1h}(r) = \frac{1}{\bar{n}_g^2} \int n(m) \frac{\langle N_t(N_t - 1)\rangle}{2} \lambda(r|m)\, dm,$$

~~however using~~ Using the formalism of §3.1, we ~~can~~ separate the pair counts into contributions from central-satellite $(c-s)$ ~~pairs,~~ and satellite-satellite $(s-s)$ ~~pairs (cf. Eq. 3.1~~ cross-pairs (we shall deal with self-pairs shortly). The first of these is a one-point quantity dependent ~~only on the profile, rather than the~~ on the density profile, while the latter is given by the self-convolution of the profile (similar to DM). Thus we have ~~, for the correlations~~

$$1 + \xi_g^{1h}(r) = \frac{2}{\bar{n}_g^2}\frac{1}{\bar{n}_g^2} \int n(m) \left[2\langle N_c N_s \rangle \rho(r|m)\frac{\rho(r|m)}{m} + N_s^2 \lambda(r| \right.$$
$$\tag{63}$$

where the quantity ~~$\langle N_c N_s \rangle$~~ $\langle N_c N_s \rangle$ is treated differently depending on whether the central condition is imposed (cf. §3.1). ~~The Fourier-space counterpart is defined likewise.~~

The behaviour of this expression as the separation approaches zero is of some interest. We firstly note that as r falls below some (very small) threshold, we should no longer trust the HOD model – some level of "galaxy exclusion" must apply (Ginzburg et al., 2017), rendering the the distribution of N_s non-Poisson and non-independent of the surrounding locations. Thus Eq. 63 will tend to over-estimate the correlation function below this threshold. This is most evident at $r = 0$, where Eq. 63 predicts a non-zero correlation, even though we have explicitly avoided self-pairs.

On the other hand, self-pairs are really there. They form a delta-function contribution at $r = 0$, with a magnitude

14

given by the mean galaxy density[22]. This translates into a constant with magnitude $1/\bar{n}_g$ in the power spectrum, and this is properly the 'shot-noise'. In HALOMOD, the power spectra are defined without shot-noise (i.e. purely as the FT of Eq. 63), which can be added in by the user if desired simply as a constant. The power spectrum is thus

$$P_{\text{gg}}^{\text{1h}}(k) = \int \mathrm{d}m\, n(m)u(k|m)\left[N_s^2 u(k|m) + 2\langle N_s N_c\rangle\right].$$
(64)

Note again that the `force_1halo_turnover` option forces the 1-halo term to turnover on scales larger than the size of halos (cf. §4.2.1), to reduce the spurious 'shot-noise', and the true shot-noise can be manually added back in to the power spectrum.

3.3. Derived Quantities

There are several quantities of interest which can be derived from the HM/HOD framework.

Firstly, on large scales, where $u \to 1$, and the 1-halo term is negligible, the 2-halo term is approximated by

$$P_{\text{gg}}(k) \approx b_{\text{eff}}^2 P_{\text{mm}}(k),$$
(65)

where

$$b_{\text{eff}} = \frac{1}{\bar{n}_g}\int n(m)b(m)N_t(m)\mathrm{d}m$$
(66)

is called the "effective bias".

We may likewise calculate an "effective concentration" which is the mean halo concentration within the sample:

$$c_{\text{eff}} = \frac{1}{\bar{n}_g}\int n(m)c(m)N_t(m)\mathrm{d}m,$$
(67)

and an "effective mass":

$$M_{\text{eff}} = \frac{1}{\bar{n}_g}\int n(m)m N_t(m)\mathrm{d}m.$$
(68)

An important quantity in studies of galaxy formation and evolution is the fraction of galaxies that are satellites. The evolution of this quantity through time can trace the effects of various physical processes. It is simply defined as

$$f_{\text{sat}} = \frac{\int n(m)N_s(m)dm}{\int n(m)N_t(m)dm}.$$
(69)

[22]Note that for non-discrete tracers, which HALOMOD supports, the 'shot-noise' contribution is not simply given by the tracer number density, but requires knowledge of the scatter in the tracer HOD population (which is no longer Poisson, cf. Chen et al. 2021).

4. The halomod library

In this section we present our new implementation of the HM, HALOMOD. Our intention is first to give an overview of the philosophy behind, and general characteristics and features of the code. From there, we describe in detail various parts of the code, beginning in §4.2 with the "frameworks" which tie together the various components, and then the individual components themselves in §4.3. These sections have a pragmatic focus, introducing any pertinent numerical techniques, and summarising the various models included. Finally, we comment on some of the extra functionality present in the package in §4.4.

4.1. Philosophy and Architecture

HALOMOD is a ~~Python~~ PYTHON package, built on the HMF package[23] (Murray et al., 2013b), which contains the necessary components and algorithms to calculate HM quantities, and also tracer statistics via a HOD[24]. HMF provides modules for cosmology, transfer functions, spatial filters, mass definitions and mass functions, along with defining a coherent and flexible software framework. HALOMOD inherits that framework and provides halo profiles, concentration relations, bias functions, halo exclusion models and HOD models and combines them into the full halo model.

The principles outlined in the introduction, i.e. "Intuitive", "Simple", "Efficient", "Extendible/Flexible", "Comprehensive" and "Open", have been the guiding principles in the development of HALOMOD (and also HMF). Our code provides a well-defined architecture for defining components and frameworks for halo modelling, including many basic and derived quantities of interest, and a large fraction of the available models in the literature. Each component is completely flexible, either by supplying custom parameters or entirely new parameterisations *without touching the source code.* We now expand upon some of these aspects.

Note that many of the features that we discuss in this section reflect updates to the HMF package (compared to its presentation in Murray et al. (2013a)~~)~~, which are inherited in HALOMOD. Furthermore, clearly this paper represents a snapshot of the package in time, with updates likely to occur.

4.1.1. Intuitive Usage

To demonstrate the intuitive usage of HALOMOD, it is perhaps most useful to give a quick example of how `halomod` can be invoked. Necessarily, the following example is both simple and rather arbitrary, but it should give a flavour of the usage of HALOMOD.

To create a new halo model class (this one will support galaxy power spectra and correlation functions) does not require passing any parameters, as all parameters have reasonable defaults:

[23]https://github.com/halomod/hmf
[24]For now, it only implements the ~~two-point~~ 2-point statistics, but this will be expanded in future versions.
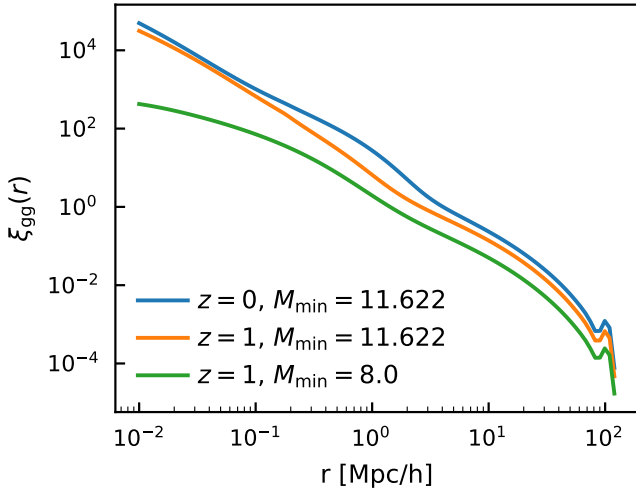
Figure 7: Simple example of galaxy auto-correlation functions produced with three inputs (two different redshifts and a different HOD model). Notice the BAO peak at $\sim 100h^{-1}$Mpc. Code to produce the figure is presented in §4.1.1.

```
import halomod as hm
tr = hm.TracerHaloModel()
```

After creating the model, all of the relevant quantities may be accessed as attributes (in fact, they are not simple attributes that have been pre-computed, they are calculated lazily on first access and then cached). Thus, we can plot the auto-correlation function of galaxies, for example;

```
plt.loglog(tr.r, tr.corr_auto_tracer, label=
    'z=0')
```

To update a parameter, one can simply set it on the instance, so for example to change the redshift and plot the new correlation function:

```
tr.z = 1
plt.loglog(tr.r, tr.corr_auto_tracer,
  label='z=1')
```

Updating any parameter immediately invalidates the cache of quantities that depend on it, and thus that quantity is recalculated (but not quantities on which it depends that don't themselves depend on the changed parameter – like the cosmological transfer function in this example).

Every sub-component of the halo model (see the following sections) allows its parameters to be changed directly from this top-level interface. So, if we wanted to update the $M_{\min}$ parameter of the halo model, we could simply do

```
tr.hod_params = {"M_min": 8.0}
plt.loglog(tr.r, tr.corr_auto_tracer, label=
    'Mmin=8')
```

After applying relevant axis labels, we then produce Fig. 7.

4.1.2. Simple Conceptual Framework

The backbone of HALOMOD, directly inherited from HMF, is split into two categories: 'frameworks' and 'components', which are well-defined by a base class, `Framework` and `Component` respectively. Individual elements of the calculation, such as a halo profile, HMF fitting function, or any other unit which may be calculated in more than one way (e.g., growth factors) are represented by a `Component`. `Framework`s are the structures which tie each `Component` together to calculate a set of quantities (e.g., correlation functions, power spectra or mass functions). Frameworks may also contain other frameworks (for example the `Cross-Correlation` framework contains two `TracerHaloModel` frameworks – one for each tracer to be cross-correlated).

This structure makes it simple to identify a component of interest for extended modelling, and provides a common interface and set of methods to each type. For example, if one is developing a new improved model of halo bias, it is clearly identified as a `Component`, with a particular structure. Its place in the overall framework is well-specified, it by default receives any defining parameters it requires, and specific examples of implementations of other models of the component are easily found in the `bias` module.

It is expected that the most common usage will be to create a `Framework` instance, and access available quantities through it. Our general approach has been to treat frameworks as a self-contained unit. This means that all parameters are passed to the initial constructor, after which every relevant quantity is ready for immediate access, without method-specific parameters. Indeed, to the user, once a `Framework` instance has been created (with all its options), any derived quantity (such as mass functions, correlation functions, mass variance etc.) is accessed just like a stored variable.

This approach is not typical of much software – usually there is benefit in segregating object-wide variables from method-specific ones. However, we find in our case that most variables are used across many methods, and are more naturally interpreted as being a part of the framework itself.

In fact, our adherence to this approach is bolstered by our generic distinction (within `Framework`s) between a `parameter` and a `cached_quantity`. Every input variable is treated as a class-wide `parameter`, which receives its own characteristics (this is very useful for our caching mechanism, see §4.1.3). Every evaluated quantity is treated as a `cached_quantity`, which likewise has its own properties. The set of all `parameter`s of a given `Framework` is then a unique description of the entire object (and can be used to test equality between framework instances).

Though the state of a `Framework` is defined by its parameters, we provide a simple mechanism of modifying the parameters for dynamic calculations, through an `update` method, which can receive any `parameter` (or, as noted in the previous subsection, simply by setting the attribute)[25].

---

[25]This method is defined in the base `Framework` class, and is therefore simply inherited by any specific framework.

Our approach to the passing of parameters has been to provide defaults for *every* variable in a `Framework`[26]. These defaults are chosen to be broadly in line with the most common usage, which greatly simplifies common tasks. However, it must be noted that this has the drawback that if user code relies on defaults, and code updates modify them, then results can vary without being noticed.

In summary, the architecture of HMF and HALOMOD consists of

- A flexible `Component` for each independent element of the HM.

- A range of specific *models* for each `Component`.

- Powerful `Framework` classes that synthesise multiple `Components` to provide plug-and-play functionality.

- `Frameworks` consist of `parameters` which can be input and modified by the user, and `cached_quantity` objects, which are calculated from the `parameters` and are read-only and always consistent with the current settings.

4.1.3. Efficiency

While HALOMOD generally prioritizes ease-of-use over efficiency, it is optimized in a number of ways:

- Heavy use of NUMPY[27] to enable vectorized calculations, close to compiled code performance in most cases.

- Calculation of all quantities at a fixed set of user-defined vectors (as opposed to providing methods that ingest evaluated variables such as mass or wavenumber)[28].

- Use of "just-in-time" (JIT) compiled functions (via NUMBA[29]) where vectorization is impossible or disadvantageous.

- A novel and robust caching system with extremely accurate and efficient cache invalidation (i.e. cached quantities always stay up-to-date with parameter changes, but nothing updates that doesn't have to). See Appendix A for details on this system, which utilises the core architecture of the `Framework` to achieve its goals.

4.1.4. Flexibility

We have referred to our imposed dichotomy between `Framework` and `Component` objects, and in the previous subsection we outlined a performance benefit derived from the internal structure of the `Framework`. Likewise, we have implemented a uniform convention for `Component` specification which enhances flexibility.

Any component that is implemented as a `Component` is a variable within a `Framework`. As such, it is determined by passing a `parameter`, in the form of either a string representing the name of the model, or the class itself.

Every `Component` has a top-level dictionary specifying the model-specific parameters and their default values (these are not fundamental quantities used in the calculation, eg. ν in the $f(\nu)$ for the HMF, but rather free parameters of the model, if any). These are processed by the base class so that any newly defined model automatically has them available. In the `Framework`, the model parameters are passed, by convention, as `<cmpnt>_params`. Furthermore, the `update` method in each `Framework` intelligently updates parameter dictionaries so that previous updates are persistent unless explicitly overwritten.

This general system offers the highest level of flexibility and extendibility. In a single session, it is extremely simple to switch between several models for a given component, thereby making useful comparisons (and this without needing to recalculate basic quantities that do not depend on the variable component). On the other hand, using entirely new custom models can be achieved without modifying the source code, by inheriting from the base `Component` and passing the new model directly to a `Framework`. We reiterate this point, as we expect this to be particularly advantageous in the current climate of rapid development: `Components` are *pluggable*. A user can write their own new HMF fit, or bias function, or HOD model, in their own notebook (or wherever they write code), and use it right away in the context of the full halo model.

Furthermore, to a greater or lesser degree, each `Component` has built-in methods which either remove the burden on the user, or supply extra functionality for free when the user develops a new model. This is perhaps most clear in the `Profiles` component (cf. §4.3.7), though some components have very little to add to the basic model (eg. the `Bias` component, cf. §4.3.6).

`Components` are primarily designed to integrate with the `Frameworks`, however, they are not limited to this usage. They are also intended to be useful in their own right, for more specific applications.

4.1.5. Interoperability

HALOMOD is intended to be a package that is adopted and maintained by the community. To this end, it uses well-known and well-maintained libraries to enable much of its functionality. For example, cosmologies are defined via ASTROPY[30] and we use the standard open-source NUMPY

---

[26] These defaults are currently majoritarily static, but we plan on dynamifying them in future versions.

[27] http://www.numpy.org/

[28] An exception to this rule applies in the case of the 2-halo term, which can require two sequential Hankel transforms. For this, we evaluate $\xi_{2h}(r)$ on a background grid of r which adequately captures the behaviour required for the inverse transform, and we interpolate this grid for the user's input vector of r.

[29] http://numba.pydata.org/

[30] https://www.astropy.org/

and SCIPY packages for numeric and scientific routines. We also adopt the open-source HANKEL[31] (Murray and Poulin, 2019) package for computing some of the Hankel transforms.

We also provide extensive interoperability with the ~~COLOSSUS~~ COLOSSUS code[32] (Diemer, 2018), by providing component constructors for models contained in ~~COLOSSUS~~ COLOSSUS. In practice, this means that the user can invoke something like the following

```
bias = make_colossus_bias("comparat17")
model = TracerHaloModel(bias_model=bias)
```

and get the full range of functionality from the ~~COLOSSUS~~ COLOSSUS-defined bias model within the HALOMOD framework (even setting of model-dependent parameters). At the time of publishing, these constructors are defined for bias and halo concentration models, as these models are part of the core functionality of ~~COLOSSUS~~ COLOSSUS and thus there is a real chance that the models may exist in ~~COLOSSUS~~ COLOSSUS and not HALOMOD.

Furthermore, versions of COLOSSUS from v1.2.16 onwards have implemented a simple pair of functions enabling conversion of ASTROPY cosmologies to and from the native COLOSSUS cosmologies. Since HALOMOD cosmologies are essentially derived from ASTROPY, this enables a greater degree of interoperability between COLOSSUS and HALOMOD.

4.2. Frameworks

In this section we present ~~two~~ several of the frameworks that HALOMOD provides. We note that several other frameworks are found in the HMF package, from which those presented here inherit. Furthermore, HALOMOD also provides a `DMHaloModel` framework for calculating non-linear matter power spectra and correlation functions ~~, and a Cross-Correlations framework for cross-correlating two tracers. We do not present these here explicitly, but note that the `DMHaloModel` is a strict subset of the `TracerHaloModel` which we do present.~~ (the same are available within the `TracerHaloModel`, but the `DMHaloModel` requires fewer inputs). `DMHaloModel` is a strict subset of the `TracerHaloModel` which we do present.

4.2.1. Base Halo Model

The basic framework in HALOMOD is `TracerHaloModel`, which provides general access to most derived quantities relating to ~~two-point~~ 2-point statistics. Table 1 contains a useful summary of parameters for `TracerHaloModel`, giving an indication of its flexibility. Likewise, table 2 lists the available properties. We note that these are expected to be updated and expanded in future versions, and thus are not intended to provide a strict API, but rather an illustrative summary.

Some of the features of the `HaloModel` framework, ignoring contributions of individual components, are as follows.

Mean density matching. Most HOD parameterisations have a parameter $M_{\min}$, which either explicitly or loosely defines the minimum halo mass expected to host galaxies in the sample. This parameter is tightly correlated with the mean galaxy density, as it determines the lower limit of the integral in ~~Eq. 61.~~ Eq. 61. It is common (though not ubiquitous) in survey analyses to let the known value of $\bar{n}_g$ determine $M_{\min}$ (given other parameters of the HOD) (eg Beutler et al., 2013). `TracerHaloModel` supports this technique for all HOD models, either by direct cumulative integration, or by numerical minimization. This is caught at every update of parameters.

Mass range setting. The mass range in `TracerHaloModel` is user-definable, but sets intelligent limits. An upper limit of $M_{\max} = 10^{18} h^{-1} M_\odot$ is set to ensure convergence of mass integral, and the lower limit is set to $M_{\min} = 10^0 h^{-1} M_\odot$ for matter calculations, and is based on the value of $M_{\min}$ in the HOD for tracer calculations. For step-function models, the lower limit is exactly $M_{\min}$, which increases accuracy in an important regime. For smooth-cutoff models, the lower limit is determined by the `mmin` property of the HOD (see §4.3.9).

Halo-centre power spectrum. A key quantity in the large-scale clustering is the power spectrum of halo centres, ~~$P_{hh}^c(k)$~~ $P_{\rm hh}^c(k)$ (cf. §2.1.2 and §2.9). This can be modelled via the linear matter power, nonlinear matter power, or even some smoothed version of either. We leave this choice free for the user, through the parameter `hc_spectrum`.

Convolutions. The satellite-satellite term can be calculated directly in real-space if and only if there is an analytic solution for the self-convolution of the halo profile. `TracerHaloModel` uses this form directly if possible, saving integration time, but resorts to a calculation in Fourier-space (and subsequent Hankel transform) if necessary.

Hankel transforms. The Hankel transform is a delicate operation, involving the integration of a highly oscillatory function. We use a novel method for performing this transform, which we detail in Appendix B.

Tabulation. The Hankel transform of a power spectrum to a correlation function is typically relatively simple and accurate (using the framework for Hankel transforms defined in Appendix B). This is especially so since the power spectrum is naturally defined over a large range and at quite high resolution (as is required for the calculation of the mass variance, for example). However, the converse is not true – transforming a correlation function to a power spectrum is typically quite limited in its range of accuracy.

It is thus tempting to calculate power spectra at the k-vector at which the linear power spectrum is defined, and consider only ever transforming into real space. If this is the case, the vector of r at which the correlation functions are defined can be low-resolution and limited in range, since

---

[31] https://github.com/steven-murray/hankel
[32] http://www.benediktdiemer.com/code/colossus/

Parameter	Default[†]	Description
Components/Parameters*		
`cosmo_<>`	Planck13	Underlying cosmology. Must be an `FLRW` object from astropy. This model differs from other `Components`, being passed as an *instance*, not a *class*.
`transfer_<>`	CAMB	A model to use for the transfer function.
`growth_<>`	GrowthFactor	A model for the growth function. The default is to perform the explicit integration.
`mdef_<>`	None	A mass definition (default is the definition under which the chosen `hmf` was measured).
`hmf_<>`	Tinker08	A model specifying the HMF fitting function, $\nu f(\nu)$.
`filter_<>`	TopHat	A filter function model.
`hod_<>`	Zehavi05	A HOD model.
`profile_<>`	NFW	A halo profile.
`concentration_<>`	Duffy08	A concentration-mass-redshift relation
`bias_<>`	Tinker10	A scale-independent linear bias model.
`sd_bias_<>`	Tinker_SD05	A model for the scale-dependence of the linear bias.
`exclusion_<>`	NgMatched	A halo exclusion model.
Resolution/Location		
`lnk_min`, `lnk_max`, `dlnk`	-8,8, 0.05	Min/Max/Δ logarithmic wavenumber for linear power spectra and tabulated halo model spectra
`hm_logk_min`, `hm_logk_max`, `hm_dlogk`	-2.5,1.5, 0.05	Min/Max/Δ logarithmic wavenumber for output halo model spectra
`dlog10m`	0.01	Logarithmic mass interval
`rmin`, `rmax`, `rnum`, `rlog`	0.1, 50.0, 20, True	Min/Max/# for pair-separation vector, and whether its vector should be equi-log-spaced.
`r_table_num`	100	Length of background table of r
Physical		
`sigma_8`	0.8344	Normalisation of power spectrum, σ_8
`n`	0.9624	Spectral index, n_s
`z`	0.0	Redshift
`delta_c`	1.686	Critical overdensity for collapse, δ_c.
`ng`	None	Optional specification of mean galaxy density. If present, an HOD parameter (generally $M_{\min}$) is fixed.
Options		
`takahashi`	True	Apply updated halofit coefficients from Takahashi et al. (2012), otherwise use original coefficiencts from Smith et al. (2003).
`hc_spectrum`	~~Nonlinear~~ Linear	Model for $P_{hh}^c(k)$ $P_{hh}^c(k)$. Either "linear", "nonlinear" (halofit), or "filtered-nl" (smoothed on a scale of $2h^{-1}$Mpc).
`force_1halo_turnover`	True	Whether to cut off the 1-halo power on extremely large scales.
`force_unity_dm_bias`	True	Whether to renormalize the 2-halo term such that the DM bias is unity for $k \to 0$.

Table 1: All parameters of `TracerHaloModel`. Parameters in the "Components/Parameters" section are each specified by two parameters, ending in `model` and `params` (i.e. `hmf_<>` represents two parameters: `hmf_model` and `hmf_params`). Using the HMF as an example, `hmf_model` is the actual model specification, and this is listed in the Default column. Conversely, `hmf_params` is a dictionary of parameters for the model, which is not listed in this table, but can be found in the online API documentation. The parameters available for each *model* within a particular component may be different. Notice also that the physical parameters σ_8 and n_s are directly set within the framework, rather than being a part of the cosmology parameters. The reason for this is outlined in §4.3.1.

Type	Quantities*
Scalar	$b_{\rm eff}$, $f_{\rm cen}$, $f_{\rm sat}$, $D^+(z)$, $M_{\rm eff}$, $M_\star$, $\bar\rho(z)$, $\bar n_g$, $n_{\rm eff}$
Length-m	R, b, $\bar c$, $\nu f(\nu)$, $\frac{dn}{d\ln m}$, $\frac{dn}{d\log_{10} m}$, $\frac{dn}{dm}$, $n(>m)$, $\langle N_{\rm c}\rangle$, $\langle N_{\rm s}\rangle$, $\langle N_{\rm t}\rangle$, $L(n=1)$, $\ln\sigma^{-1}$, ν, $\rho(>m)$, $\rho(<m)$
Length-k	T, P_m, $P_{m,\rm halofit}$ P_m, $P_{m,\rm halofit}$, Δ^2, $\Delta^2_{\rm halofit}$, P^{ss}_{gg}, P^{2h}_{gg} P^{ss}_{gg}, P^{2h}_{gg}, P_{mm}, P^{1h}_{mm}, P^{2h}_{mm}, P^{1h}_{mg}, P^{2h}_{mg} P^{1h}_{mm}, P^{2h}_{mm}, P^{1h}_{mg}, P^{2h}_{mg}, P_{mg}
Length-r	ξ^{1h}_{gg}, ξ^{cs}_{gg}, ξ^{ss}_{gg}, ξ^{2h}_{gg}, ξ_{gg}, ξ^{1h}_{DM}, ξ^{2h}_{DM}, ξ_{DM}, ξ_m ξ^{1h}_{gg}, ξ^{cs}_{gg}, ξ^{ss}_{gg}, ξ^{2h}_{gg}, ξ_{gg}, ξ^{1h}_{DM}, ξ^{2h}_{DM}, ξ_{DM}, ξ_m, $\xi_{m,\rm halofit}$, ξ^{1h}_{mg}, ξ^{2h}_{mg} ξ^{1h}_{mg}, ξ^{2h}_{mg}, ξ_{mg}, $S(\xi_m(r))$ $S(\xi_m(r))$
Function of r,m	ρ, λ
Function of k,m	u

Table 2: All properties of `TracerHaloModel`. Listed are those quantities that are directly accessible as explicit properties. Other quantities, such as the window function, are accessible indirectly through a model instance variable.

it would be used only as an output, not as an input to some other function that requires it to be high-resolution.

However, this turns out not to be realistic in practice. In particular, some models of halo exclusion mix a scale r – related to the halo radius – with the 2-halo power spectrum. Scale-dependent bias does likewise. To obtain a standard $P(k)$ thus requires performing a Hankel transform of the power *at each r*, then an inverse Hankel transform to get back to $P(k)$.

This requires using a higher-resolution, and importantly, large-range vector of r for the intermediate correlation function. Furthermore, the calculation must be confident that the vector of r is suitably well-chosen. We thus use a set tabulated background r vector, and interpolate the results of $\xi(r)$ onto the user-provided r-vector.

We also provide an option for the user to define a more limited range of k at which to output the halo model power spectra, and these are also interpolated from their base table, which is defined on the k-vector used for the linear spectra.

Problematically, the Hankel transform algorithm we employ often requires *extrapolated* values of the integrand in order for the integral to converge for some scales[33], particularly when transforming from $\xi(r)$ to $P(k)$. Typical cubic spline interpolation performs very poorly on extrapolation in general, often blowing up so much that the transform can be completely ruined. We implement a robust `ExtendedSpline` object to circumvent these issues. The `ExtendedSpline` interpolates (typically with cubic interpolation) interior to the given co-ordinates, but is supplemented by settable conditions at both ends. The conditions at either end can be to return zeros, to continue to extrapolate the spline, to return some user-defined asymptotic function (which is normalised to match at the

boundary point) or to extrapolate linearly in log-space (i.e. assuming the asymptotic behaviour on either end of the function is roughly power-law-like). We typically set correlation functions to return zero above the point at which they cross zero (typically about 125 Mpc/h) and exhibit power-law behaviour on small scales (as long as the slope of the function is shallower than an index of -2 the smallest scales do not contribute significantly in any case). Power spectra are considered to be power-laws at high k and power laws explicitly with an index of n_s at low k (except when they are cut-off for some reason, like the 1-halo term, in which case they return zero below the cutoff).

We provide callable functions for each of the power spectra and correlation functions of the halo model, based on these extended splines evaluated on the high-resolution background tabulated data. Nevertheless, we still provide the default evaluation of these splines on the user-defined r and k vectors.

1-halo turnover. At large scales, the 1-halo power spectrum converges to a non-zero constant (this is clear from ~~Eq. 10~~ Eq. 10, since u asymptotes to unity as $k \to 0$). This is generally considered to be the "shot-noise" term, and is clearly unphysical when considering a smooth distribution. `TracerHaloModel` offers to force the 1-halo term to turnover on scales larger than about 10 halo radii.

Explicitly, it changes the lower mass limit of the integration in ~~Eq. 10 to be~~ Eq. 10 to be

$$m_{\rm lim}(k) = \frac{4\pi}{3}\left(\frac{\pi}{10k}\right)^3 \bar\rho_0 \Delta_{\underline{h}\rm h}. \tag{70}$$

Large-scale matter-matter bias. The large-scale matter power under the halo model ansatz (i.e. "all mass is inside halos") is by definition equivalent to the linear power. Another way of saying this is that the large-scale effective bias, Eq. 66, for matter is unity. This, in principle, is true if the consistency relation, Eq. 29, holds. However, there are several ways in which this condition will fail – either in principle or in practice.

---

[33]This is partly because the default k range is quite small, as seen in Table 1. However, for numerical transfer functions such as those from CAMB, the intrinsically calculated range of k is rather small and requires extrapolation even onto the set of k input by the user.

The first is that in some halo models, the ansatz itself is specifically broken. For example, halo models of warm dark matter may invoke a "smooth" dark matter component that is not collapsed into halos (eg. Smith and Marković, 2011). In this case, the HMF model is specifically tuned to not integrate to the mean density.

The second is when a bias or HMF relation is chosen such that Eq. 29 is not satisfied. Sometimes these relations may provide better accuracy at large masses, at the expense of being incompatible with the halo model ansatz. Typically, in these cases, one still *desires* the behaviour that the matter power converge to the linear power at large scales. The best way to avoid this situation is to avoid choosing incompatible relations when computing matter statistics (such relations are usually fine for tracer statistics, as these generally trace only halos above some minimum mass).

The third is simply due to the inherent limitation of the numerical integration. Even with compatible HMF and bias relations, the numerical integration to compute $b(k,r)$ is over a finite range of mass, and thus will not necessarily converge to unity at large scales. This is always undesirable.

To deal with these issues, HALOMOD includes an option `force_unity_dm_bias`, which if set, manually sets the effective bias of matter to unity, and then renormalizes the 2-halo term such that it approaches these effective bias at large scales. It does this irrespective of the HMF and bias relation chosen, which is generally the desired behaviour.

4.2.2. Projected correlation function

In analyses of galaxy surveys, it is more common to measure *projected*, rather than real-space, correlation functions (another alternative is the *angular* CF, which we outline in the next section). We provide an extended framework, inheriting from `TracerHaloModel`, for this calculation. The primary reason it is separated from the basic class is that it requires extra parameters specific to its calculation.

The primary addition with respect to the base `Tracer-HaloModel` class is the conversion of the real-space 2PCF to projected space. The transformation from real-space to projected is defined as

$$w_p(r_p) = 2 \int_{r_p}^{\infty} \frac{r\xi(r)}{\sqrt{r^2 - r_p^2}} dr \frac{r\xi(r)}{\sqrt{r^2 - r_p^2}} dr. \quad (71)$$

The implementation of this integral is rather delicate due to the singularity at the lower bound. This renders both limits of the integration non-physical, and convergence requirements must be met by both. We discuss our prescription for these limits in appendix Appendix C.

4.2.3. Angular correlation functions

Angular correlation functions can be computed using the `AngularCF` class (which again inherits from `TracerHaloModel`).

The angular correlation function between two density fields (potentially the same field in an auto-correlation) is defined as (Simon, 2007)

$$w(\theta) \approx \int_0^{\infty} dr dr_1 \int_0^{\infty} dr dr_2 \, p_1(r_1) p_2(r_2) \xi(R, \bar{r}), \quad (72)$$

where p_1 and p_2 are probability distributions of finding tracers from population 1 and 2 respectively at line-of-sight comoving distance r (i.e. the redshift distribution of sources), and

$$R \equiv \sqrt{r_1^2 + r_2^2 - 2r_1 r_2 \cos\theta}, \quad (73)$$

is a projected separation, and

$$\bar{r} = (r_1 + r_2)/2. \quad (74)$$

In HALOMOD, we exclusively use the Limber approximation (Limber, 1953), which is a good approximation if $p(r)$ is a wide distribution, and θ is relatively small. In this approximation we have

$$w(\theta) = 2 \int_0^{\infty} d\bar{r} p_1(\bar{r}) p_2(\bar{r}) \int_0^{\infty} d\Delta r \xi(R, \bar{r}), \quad (75)$$

where

$$R \equiv \sqrt{\bar{r}^2 \theta^2 + \Delta r^2}. \quad (76)$$

4.2.4. Tracer-Tracer Cross-Correlations

While the `TracerHaloModel` class is able to compute cross-correlations between matter and a tracer field, it is often useful to be able to compute cross-correlations between two tracer fields (eg. optical galaxies and neutral hydrogen, or two classes of optical galaxies). This functionality is supported via the `CrossCorrelations` class, to which the user provides two `TracerHaloModel` instances. Note that currently, halo exclusion is not supported in cross-correlations.

In addition to defining the properties of the individual tracer populations, one must define the way the two populations correlate within halos (this is immediately clear when considering two overlapping populations of galaxies, for example). Within HALOMOD, we use the following relations:

$$\langle T_1^{c,s} T_2^{c,s} \rangle = \langle T_1^{c,s} \rangle \langle T_2^{c,s} \rangle + \sigma_1^{c,s} \sigma_1^{c,s} R_{12}^{c-s,s-c,s-s} - \delta_{ss} Q_{ss}, \quad (77)$$

where c and s stand for central and satellite, and may be the same or different for each of the populations (i.e. we have central-satellite, satellite-central and satellite-satellite terms, but not a central-central term since the central tracer, if it exists, must be identical between the populations). The final term, which is only non-zero in the satellite-satellite case, is in that case the average occupation of tracers shared between the population per halo of mass m. This relation is nothing more than the standard definition of the variance of a product of random variables, in which the σ is the standard deviation in the occupation for each

population, and R is a correlation coefficient between -1 and 1. For independent/uncorrelated populations, $R = 0$. For an autocorrelation, $R = 1$ and $Q = N_s$, and we obtain the same expressions presented in §3.1. More details and a worked example in the context of cross-correlations of optical galaxies and H I is provided in Wolz et al. (2019).

In HALOMOD, we allow the user to specify the variance of the occupation (for both centrals and satellites) within the `HOD` component (cf. §4.3.9). We additionally provide a `HODCross` component that allows combining two HOD models, and allows specifying R for the three available combinations of centrals and satellites.

4.2.5. Warm dark matter models

For each of the frameworks, we also implement a version suited for warm dark matter (WDM) models. The WDM frameworks are designed to set relevant default models and perform any framework-level WDM-specific modifications.

In particular, we implement the WDM transfer function from Viel et al. (2005), the WDM-compatible concentration relation of Ludlow et al. (2016), along with the Sharp-k filter proposed by Schneider et al. (2012) and the general large-scale structure framework for WDM outlined in Smith and Marković (2011).

4.3. Components

This section provides a brief summary of each of the components within HALOMOD. For each component, we provide a table or otherwise concise summary of the models already implemented within HALOMOD, and a comparison plot if relevant.

4.3.1. Cosmology

Instead of defining our own basic cosmology models, we use the mature ASTROPY package[34] (Robitaille et al., 2013). In brief, this package implements a connected series of cosmogaphic models based on the Friedmann-Lemaitre-Robertson-Walker (FLRW) metric, with several standard instances based on results from CMB missions.

We do not present the summary details here, but note that basic cosmographic quantities such as comoving distance, and density parameters such as the mean density as a function of redshift, are implemented in these models.

One commonly misunderstood aspect of the cosmology component is that it *does not* include the large-scale-structure (LSS) parameters σ_8 and the spectral index n_s. This is perhaps counter-intuitive, since CMB constraints such as those from WMAP and Planck typically provide constraints on these parameters as part of the fundamental cosmology, and furthermore these parameters are covariant with cosmographic parameters (especially σ_8 and Ω_m). However, ASTROPY cosmologies are purely cosmographic, and HMF inherits this behaviour. These LSS parameters are instead provided as inputs to the `Transfer` framework

for which they are necessary to determine the normalisation and slope of the power spectrum.

4.3.2. Transfer functions

We provide a common interface for calculating transfer functions using a variety of methods: from analytic fits, to Boltzmann code, to importing data from file. In each case, the required function returns the logarithmic transfer as a function of logarithmic wavenumber, as lower-order splines can be accurately fit in log space.

We use the PYCAMB[35] wrapper to provide on-the-fly access to CAMB routines. We also note that we include a standalone version of HALOFIT that may be used to apply nonlinear corrections to any of the included power spectra.

The various transfer models included in HMF are summarized in Table 3. A comparison of the models with default parameters is shown in ~~figure 2~~Fig. 2.

4.3.3. Mass Definitions

HALOMOD contains a flexible system for defining halo masses. Both FOF and SO halos are supported, and three general SO subclasses in which the overdensity is with respect to the mean background or the critical density, or is defined as the virial overdensity as a function of redshift (Bryan and Norman, 1998).

All mass definitions have a free parameter – either the linking length or the overdensity criterion. All of the definitions are able to calculate the halo density (or overdensity with respect to either mean or critical). For the FOF definition we use the approximate formula from White et al. (2001);

$$\rho_h^{\text{FOF}} \approx \bar{\rho}\frac{9}{2\pi b^3}, \tag{78}$$

with b the linking length. Note that this formula is very approximate (More et al., 2011), as FOF halos can't readily be identified with SO halos.

To change mass definition requires solving for a new halo concentration, according to

$$\frac{c_{\text{old}}^3}{c_{\text{new}}^3}\frac{h(c_{\text{new}})}{h(c_{\text{old}})} = \frac{\Delta_{\text{new}}}{\Delta_{\text{old}}}, \tag{79}$$

which is clearly profile-dependent. Then the new mass is just

$$m_{\text{new}} = m_{\text{old}}\frac{c_{\text{new}}^3}{c_{\text{old}}^3}\frac{\Delta_{\text{new}}}{\Delta_{\text{old}}}. \tag{80}$$

HALOMOD supports changing masses and concentrations (partially) between mass definitions in this way. See Fig. 8 for a comparison of mass and concentration for different halo mass definitions.

---

[34] http://www.astropy.org

[35] https://github.com/cmbant/CAMB

REF	NAME	FORMULA	PARAMS
Bond and Efstathiou (1984)	`BondEfs`	$\left[1+\left(\sum_{i=2}^{4}(ck)^i\right)^\nu\right]^{-1/\nu}$	$\dfrac{\Omega_m h^2}{0.3\times0.7^2}c=(37.1,21.1,10.8),$ $\nu=1.12$
Bardeen et al. (1986)	`BBKS`	$\dfrac{\ln(1+aq)}{aq}\left[\sum_{i=0}^{4}(c_i q)^i\right]^{-1/4}$	$q=\dfrac{k\exp(\Omega_b(1+\sqrt{2h}/\Omega_m))}{\Gamma}$ $q=\dfrac{k\exp[\Omega_b(1+\sqrt{}}{\Gamma h\mathrm{Mpc}}$ $\Gamma=\Omega_m h,\ a=2.34,\ c=(1,\,3.89,\,16.1,$ ~~5.47~~ 5.46, 6.71)
Eisenstein and Hu (1998) (No BAO)	`EH_NoBAO`	Original paper, Eq. [26] and [28]-[31]	
Eisenstein and Hu (1998) (w/ BAO)	`EH_BAO`	Original paper, §2 and Eq. [16]-[24]	
Lewis et al. (2000)	`CAMB`	Boltzmann Code	Many options
	`FromFile`	Read from file	Filename
	`FromArray`	Use pre-computed array	Array

Table 3: Summary of included `Transfer` models. Note that the Ω_b-dependent exponential factor in the BBKS model of q is drawn from Sugiyama (1995) and Liddle and Lyth (2000), and is optional within HALOMOD. Note also that these equations assume k to be in units of h/Mpc, which explains why q is divided by h instead of h^2.

4.3.4. Window functions/Filters

The window/filter function contributes to the basic unit of the spherical collapse formalism, the peak-height ν. It is also responsible for the conversion of distance scales to mass scales (cf. §2.4).

Our general approach in this component is to specify all the quantities in terms of the Fourier co-ordinate, k, and the smoothing scale R. The class itself defines the relationship between mass m and the smoothing scale. The primary quantity of interest is the window function itself, specified in Fourier space, $W(x)$ (one can optionally specify the real-space version as well, though it is not involved in typical HM calculations). With this function, the mass variance can be calculated using ~~Eq. 19.~~ Eq. 19.

To calculate dn/dm requires also the logarithmic derivative of the variance with mass. To remain as general as possible, we use the following identity

$$\frac{d\ln\sigma}{d\ln m}\equiv\frac{1}{2}\frac{d\ln\sigma^2}{d\ln R}\frac{d\ln R}{d\ln m}, \tag{81}$$

where

$$\frac{d\ln\sigma^2}{d\ln R}=\frac{1}{\pi^2\sigma^2}\int W(kR)\frac{dW(kR)}{d\ln(kR)}P(k)k^2\,dk. \tag{82}$$

The factor $d\ln R/d\ln m$ is typically $1/3$, though this is not necessarily the case for window functions of arbitrary shape[36] (Schneider et al., 2013). Thus, the defining quan-

---
[36] The mass referenced here is the mass within a shaped volume defined by the window function.

Figure 8: Conversion of mass and concentration between different halo definitions. Top-panel shows ratio of converted mass to input mass, as a function of input mass. Different colors represent different halo definitions. Bottom panel shows ratio of converted concentration to input concentration, as a function of input mass. Input mass definition is 200m. Note that the trend with increasing overdensity is to decrease both the mass and concentration. The halo profile used in the conversion is the NFW profile, and the concentration-mass relation is the simple power law of Duffy et al. (2008).
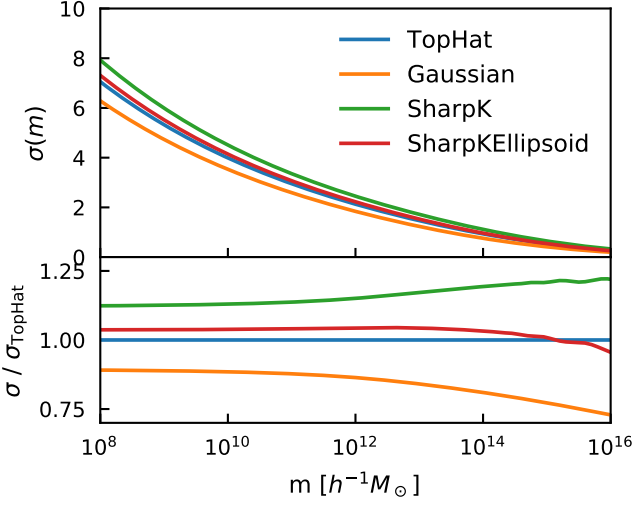
Figure 9: The mass variance determined by the three most common window functions. The normalisation of the power spectrum in each case uses the `TopHat` filter, for which σ_8 is defined.

tity is the derivative of the window function, $dW(x)/dx$.

We also note that we generalise the calculation of the mass variance to all moments of the smoothed density field:

$$\sigma_n^2(r) = \frac{1}{2\pi^2} \int k^{2n} P(k) W^2(kR)\, \underline{dk}\underset{\sim}{\mathrm{d}k},\qquad (83)$$

for which σ_0^2 is the usual mass variance.

Table 4 summarises the window functions included in HALOMOD, while ~~figure 9~~ Fig. 9 displays the value of $\sigma(m)$ for each. Note that there is a complex interplay between the role of $W(kR)$ and the mass assignment which drives changes between the models. Note also that details for `SharpKEllipsoid` can be found in Appendix A of Schneider et al. (2013).

4.3.5. HMF

The diversity among fitting functions for the HMF, not only in functional forms, but in general approach and explicit parameter dependence, requires that quite a number of variables be available in the general case. However, the typical fit uses just ν (or sometimes σ) and its own model parameters, reflecting the expected universality of the fit.

Each HMF model has a class attribute `normalized`, which specifies whether the particular HMF integrates to the mean density (i.e. all mass is in halos). Note that this is an *in principle* attribute – when actually integrating the HMF, a limited mass range is used, and there is no guarantee that any particular model (even if normalized in principle) will integrate to the mean density.

The mass function is dependent on the halo mass definition. For a given halo mass, a higher overdensity criterion means that fewer peaks will exist. This effectively changes $f(\nu)$, since all other factors in Eq. 23 remain constant with mass definition (as they depend on the Lagrangian mass). The precise form of this dependence is extremely complicated;

NAME	FORMULA
`TopHat`	$W(r,R) = H(R-r)$ $W(x=kR) = \dfrac{3}{x^3}\left(\sin x - x\cos x\right)$ $m(R) = \dfrac{4\pi}{3}R^3\bar{\rho}$ $\dfrac{dW}{d\ln x} = \dfrac{9x\cos x + 3(x^2-3)\sin x}{x^3}$ $\underset{\sim}{\dfrac{\mathrm{d}W}{\mathrm{d}\ln x}} = \dfrac{9x\cos x + 3(x^2-3)\sin x}{x^3}$
`SharpK`	$W(r,R) =$ $\dfrac{\sin(r/R) - (r/R)\cos(r/R)}{2\pi^2 r^3}$ $W(x=kR) = H(kR-1)$ $m(R) = \dfrac{4\pi}{3}\left[cR\right]^3\bar{\rho}$ $\dfrac{dW}{d\ln x} = \delta_D(x-1)$ $\underset{\sim}{\dfrac{\mathrm{d}W}{\mathrm{d}\ln x}} = \delta_D(x-1)$ $\dfrac{d\ln\sigma^2(r)}{d\ln r} = -\dfrac{P(1/r)}{2\pi^2\sigma^2(r)r^3}$ $\underset{\sim}{\dfrac{\mathrm{d}\ln\sigma^2(r)}{\mathrm{d}\ln r}} = -\dfrac{P(1/r)}{2\pi^2\sigma^2(r)r^3}$
`SharpKEllipsoid`	Same as `SharpK` except $m(R)$.
`Gaussian`	$W(r,R) = \dfrac{\exp(-r^2/2R^2)}{(2\pi)^{3/2}R^3}$ $W(x=kR) = \exp(-x^2/2)$ $m(R) = (2\pi)^{3/2}R^3\bar{\rho}$ $\dfrac{dW}{d\ln x} = -xW(x)\quad \underset{\sim}{\dfrac{\mathrm{d}W}{\mathrm{d}\ln x}} = -xW(x)$

Table 4: Summary of included `Filter` models. Note: H is the Heaviside step-function, δ_D is the Dirac-delta function and P is the power spectrum. Also, in `SharpK`, the mass-assignment is not well-defined, but we use the given formula with $c \approx 2.5$ a fit to simulations (Benson et al., 2012; Schneider et al., 2013).

24

for one, changing mass definition does not necessarily evolve the masses of individual halos smoothly – there is considerable scatter in the distribution of profiles for halos of a given mass, and for high masses where the mass function is extremely sensitive to mass, this can result in Eddington bias. Furthermore, the under the halo model ansatz, all mass is contained in halos, so by contracting a halo to a new overdensity, one does not merely change the mass of the given halo, but also necessarily creates several new lower-mass halos that were previously subsumed in the larger halo. Thus, converting a mass function from one definition to another is fraught with peril.

Nevertheless, there is a reasonably simple approximate conversion that can be made. For example, Bocquet et al. (2016) uses the chain rule to give the relation

$$\frac{dn}{dm'} = \frac{dn}{dm}\frac{dm}{dm'} = f(\nu)\frac{\rho_0}{m'}\frac{d\nu}{dm'} \times \frac{m'}{m}, \qquad (84)$$

where the modification to the mass function is contained in the final m'/m, which can be computed (approximately) using Eqs. 79 and 80. However, it is important to notice the assumptions in this conversion: here we are preserving the total number density of halos (over all mass), but not the total mass density. That is, for each existing halo at some mass definition, we are reducing/extending that halo to encompass less/more mass, but not creating/destroying more halos from the remains. Importantly, this means that normalized HMFs will no longer obey this important halo model consistency relation for other mass definitions. Furthermore, there are significant errors introduced by such 'translations' in comparison to properly re-fitting the mass function measured at the alternative definition (eg. fits from Bocquet et al. (2016) suggest a 20% error at low mass which diverges exponentially above the nonlinear mass). HMF allows automatically converting mass functions from their native mass definition to other definitions using this formalism, but this feature must be switched on manually by turning off the setting `disable_mass_conversion` (otherwise an inconsistent halo definition will cause an error to be raised). Note that most fits that use spherical overdensity halos are directly calibrated to varying overdensity criteria (eg. Tinker et al., 2008; Watson et al., 2013; Bocquet et al., 2016) and HALOMOD does not use the above conversion in these cases.

We present a ~~small selection of 7 out of~~ selection of the 21 available fitting functions available in HMF in ~~Figure 3~~ Fig. 3. We refer the reader to Murray et al. (2013b) for a table of included fitting functions, noting that five more fits have been added to the collection, specifically those of Tinker et al. (2010); Behroozi et al. (2013); Manera et al. (2010); Pillepich et al. (2010) and Ishiyama et al. (2015).

4.3.6. Bias

Though the majority of bias functions are specified in 'universal' form with respect to the peak-height, there are some older functions that explicitly require cosmology and are specified in terms of a scaled mass, $m/M_\star$. We thus provide access to these quantities, which may also serve to enable more fine-tuned models than possible with pure peak-background split arguments in the future.

Table 5 compiles the range of included models, and ~~Figure 4~~ Fig. 4 displays a selection of them. Note that HALOMOD defines an explicit interface with COLOSSUS, enabling all of its bias models to be used in HALOMOD, including those of Bhattacharya et al. (2011) and Comparat et al. (2017), which are not included directly in HALOMOD.

Importantly, Table 5 lists the paired HMF for each bias model, for which the consistency relation, Eq. 29, holds. By default, HALOMOD will choose the paired HMF once a bias function is chosen. The paired HMF of any bias model can be determined by accessing the `pair_hmf` attribute of the model.

Note also that in the case that the HMF and bias function chosen support the consistency relation, by *default* the large-scale 2-halo term of matter will be normalized to ensure it converges to $P_m(k)$ (i.e. a unity effective bias of matter). This needs to be manually enforced, because the numerical integration of the biased HMF does not necessarily converge to the mean matter density over the restricted range of mass included in the calculation. This normalization is implemented as a scalar multiplication of the entire 2-halo term, so as to avoid sharp features in the spectrum. This behaviour can be turned *off* by setting `force_unity_dm_bias` to false.

4.3.7. Halo Profiles

We developed a systematic representation of two-parameter universal halo profiles in §2.7. Our implementation follows this development closely. There is a richness of predictions available given just the basic profile shape, which makes the class structure of the `Profile` component extremely valuable. We have not yet implemented several generic predictions that may be added in future versions, such as the gravitational potential profile.

The `Profile` component provides a wide range of derived quantities, accessible with a simple definition of the profile shape. To make this more explicit, suppose that one wished to implement a "cored" NFW, such that the inner density was a constant ρ_s. The minimum code that the user must write (remember, this is user-side code, not touching the HALOMOD source) would be:

```
from halomod.profiles import Profile
class CoredNFW(Profile):
  def f(self,x):
    return 1./(1 + x*(1+x)**2)
```

The class itself can then be used directly or passed to any relevant `Framework`, and all of the necessary quantities will be available. For increased efficiency, it may also be beneficial to analytically define $p(\kappa, c)$, $h(c)$ and even $l(x, c)$ if available. If not specified, these quantities are evaluated via numerical integration (except for $l(x, c)$ – it is usually

REF	NAME	FORMULA	PARAMS	HMF
Mo and White (1996)	`Mo96`	$1 + \dfrac{\nu - 1}{\delta_c}$		PS
Jing (1998)	`Jing98`	$(a/x^2 + 1)^{b - cn_s}\left(1 + \dfrac{x - 1}{\delta_c}\right)$	$x = (m/M_\star)^2$, $a = 0.5$, $b = 0.06$, $c = 0.02$	
Sheth and Tormen (1999)	`ST99`	$1 + \dfrac{q\nu - 1}{\delta_c} + \dfrac{2p/\delta_c}{1 + (q\nu)^p}$	$q = 0.707$, $p = 0.3$	SMT
Mandelbaum et al. (2005)	`Mandelbaum05`		$q = 0.73$, $p = 0.15$	
Manera et al. (2010)	`Manera10`		$q = 0.709$, $p = 0.248$	
Sheth et al. (2001a) Tinker and Weinberg (2005)	`SMT01` `Tinker05`	$1 + \dfrac{1}{\sqrt{a}\delta_c}\left[s\sqrt{a}(a\nu) + \sqrt{a}b(a\nu)^{1-c} - \dfrac{(a\nu)^c}{(a\nu)^c + b(1-c)(1-c/2)}\right]$	$a = 0.707$, $b = 0.5$, $c = 0.6$ $a = 0.707$, $b = 0.35$, $c = 0.8$	SMT
Seljak and Warren (2004)	`Seljak04`	$a + bx^c + \dfrac{d}{ex + 1} + fx^g$	$x = m/M_\star$, $a = 0.53$, $b = 0.39$, $c = 0.45$, $d = 0.13$, $e = 40$, $f = 5 \times 10^{-4}$, $g = 1.5$	
Seljak and Warren (2004)	`Seljak04Cosmo`	$b_{\mathrm{Sel}} +$ $\log_{10} x\left[a_1(\Omega'_m + n'_s) + a_2(\sigma'_8 + h')\right]$	$a_1 = 0.4$, $a_2 = 0.3$ $\Omega'_m = \Omega_m - 0.3$, $n'_s = n_s - 1$, $\sigma'_8 = \sigma_8 - 0.9$, $h' = h - 0.7$	
Pillepich et al. (2010)	`Pillepich10`	$B_0 + B_1\sqrt{\nu} + B_2\nu$	$B_0 = 0.647$, $B_1 = -0.320$, $B_2 = 0.568$	
Tinker et al. (2010)	`Tinker10`	$1 - A\dfrac{\nu^{a/2}}{\nu^{a/2} + \delta_c^a} + B\nu^{b/2} + C\nu^{c/2}$	$A = 1 + 0.24y\exp[-(4/y)^4]$, $a = 0.44y - 0.88$, $B = 0.183$, $b = 1.5$, $C = 0.019 + 0.107y + 0.19\exp[-(4/y)^4]$, $c = 2.4$, $y = \log_{10}\Delta_h$.	
Tinker et al. (2010)	`Tinker10PBsplit`	$1 + \dfrac{\gamma\nu - (1 + 2\eta)}{\delta_c} + \dfrac{2\phi/\delta_c}{1 + (\beta^2\nu)^\phi}$	Same as f_{Tinker10}	`Tinker10`
	`UnityBias`	$b(\nu) = 1$		

Table 5: Summary of included `Bias` models. Note that the fit of `Manera10` is dependent on Friends-of-Friends linking length and redshift, and we give the result for $l = 0.2$ at $z = 0$.

more efficient to use an analytic Fourier-transform $u^2(k|m)$ and then use a Hankel transform on the result).

We define our implemented profiles in Table 6, and show them in and we note that for each, analytic forms have been given wherever possible to improve efficiency. In general it is non-trivial to produce analytic forms for the self-convolution. Fortunately, this can be achieved with the popular NFW profile (Sheth et al., 2001b), as we show in Table 6. The associated forms for the T_i are

$$T_1 = \frac{-4(1+a) + 2ax(1+2a) + a^2 x^2}{2x^2(1+a)^2(2+x)}, \tag{85a}$$

$$T_2 = \frac{1}{x^2} \ln\left[\frac{(1+a-ax)(1+x)}{1+a}\right], \tag{85b}$$

$$T_3 = \frac{\ln(1+x)}{x(2+x)^2}, \tag{85c}$$

$$T_4 = \frac{\ln[(1+a)/(ax+a-1)]}{x(2+x)^2}, \tag{85d}$$

$$T_5 = \frac{a^2 x - 2a}{2x(1+a)^2(2+x)}. \tag{85e}$$

For the profile of Moore et al. (1998) and the generalised NFW and Moore profiles, the non-truncated self-convolution can be reduced to a single integral (Ma and Fry, 2000),

$$l(x) = \frac{2\pi}{x} \int_0^\infty y f(y) F_\alpha^X(x,y) \, dy, \tag{86}$$

where F describes the angular part of the 3D integral. In the case of the generalised NFW, this is fully analytic:

$$F_\alpha^N(x,y) = \frac{1}{2-\alpha} \frac{(x+y)^{2-\alpha}}{(1+x+y)^{2-\alpha}} - \frac{|x-y|^{2-\alpha}}{(1+|x-y|)^{2-\alpha}},$$

while in the generalised Moore profile, we have

$$F_\alpha^M(x,y) = \int_{|x-y|}^{x+y} \frac{z\,dz}{z^p(1+z^{3-\alpha})}, \tag{88}$$

which for the Moore profile with $\alpha = 3/2$ yields

$$F_{3/2}^M = \frac{1}{3} 2\sqrt{3} \tan^{-1}\frac{-1+2\sqrt{z}}{\sqrt{3}} + \ln\frac{1+2\sqrt{z}+z}{1-\sqrt{z}+z}\Big|_{|x-y|}^{x+y}.$$

4.3.8. Concentration-mass relations

In section 2.8, we described different classes of models of the c–m relation. We have implemented a number of models in HALOMOD; we summarize them in table 7 but refer the reader to the respective papers for details.

In particular, we have implemented two age-based models, namely those of Bullock et al. (2001) and Ludlow et al. (2016). We use interpolating splines to efficiently perform the numerical inversions involved. The results of both models can be approximated with simpler fitting functions, which are also included. Moreover, we add the power-law fits of Duffy et al. (2008) and Zehavi et al. (2011). Note that the power-law fits provided are based on a particular cosmology, and will be inaccurate if used for a model with a different cosmology (the sensitivity to the cosmology is not high, and HALOMOD does not warn the user if the cosmology does not match). Thus, for general purpose applications we recommend an analytic model.

Concentration-mass relations, as we have seen in §4.3.3, are sensitive to the halo definition. In principle, the concentrations can be 'translated' from one definition to another (Diemer and Kravtsov, 2015) given a halo profile. However, there is a significant scatter in this relationship, and it is better to work within the halo definition native to the measured concentration-mass relation. For included concentration models in HALOMOD, if the halo definition is not compatible with the measurement, a warning is issued – but there is no attempt to translate the relation. This behaviour may be updated in the future to automatically translate the relation between halo definitions.

Fig. 11 illustrates the variations between the models at $z = 0$ for the default cosmology of Planck Collaboration (2015).

Note that HALOMOD includes an explicit interface for using the excellent suite of concentration-mass relations from COLOSSUS, which provides access to extra models such as the recent analytical models of Prada et al. (2012) and Diemer and Joyce (2019). Importantly, models from COLOSSUS are able to translate between halo mass definitions. These models are plotted for comparison in Fig. 11, but we do not include them in Table 7 and refer the reader to their respective papers.

4.3.9. HOD

We keep the implementation of the HOD simple, requiring only the two mean occupation functions for centrals and satellites. Furthermore, there is an optional definition of $m_{\min}$, which can be used to define a precise lower bound on halo mass. Generally this will be predominantly determined by the parameter $M_{\min}$ and it defaults to this value if not specified. In general $m_{\min}$ may be some function of the parameters.

As alluded to in §3.1, there is a subtlety in the HOD definition – the so-called 'central condition' – which is the statement that if no central galaxy exists in a halo in a given sample, then no satellites will be present. Note that the central condition *implies* the statement $\langle N_c \rangle = 0 \Rightarrow \langle N_s \rangle = 0$, but the reverse is not true in general (one could then still have $\langle N_c \rangle = 0.5$ and $\langle N_s \rangle > 0$, and then a specific halo could still have a satellite but not central). However,

NAME & REFS.	FORMULAE
NFW Navarro et al. (1997)	$f(x) = x^{-1}(1+x)^{-2}$ $h(c) = \dfrac{c}{1+c} + \ln(1+c)$
Sheth et al. (2001b)	~~$p(\kappa,c) = \cos(\kappa)\,(\mathrm{Ci}(c\kappa+\kappa) - \mathrm{Ci}(\kappa)) + \sin(\kappa)\,(\mathrm{Si}(c\kappa+\kappa) - \mathrm{Si}(\kappa)) - \dfrac{\sin(c\kappa)}{c\kappa+\kappa}$~~ $\underline{p(\kappa,c) = \cos(\kappa)\,[\mathrm{Ci}(c\kappa+\kappa) - \mathrm{Ci}(\kappa)] + \sin(\kappa)\,[\mathrm{Si}(c\kappa+\kappa) - \mathrm{Si}(\kappa)] - \dfrac{\sin(c\kappa)}{c\kappa+\kappa}}$
Ma and Fry (2000)	$p(\kappa) = \dfrac{1}{2}\left[(\pi - 2\mathrm{Si}|\kappa|)\sin|\kappa| - 2\cos(\kappa)\mathrm{Ci}|\kappa|\right]$ $l(x,c) = 4\pi \begin{cases}(T_1 + T_2 + T_3) & 0 \le x \le c \\ (T_4 + T_5) & c \le x \le 2c\end{cases}$ $l(x) = \dfrac{8\pi}{x^2(x+2)}\left[\dfrac{(x^2+2x+2)\ln(1+x)}{x(x+2)} - 1\right]$
Moore Moore et al. (1998) Ma and Fry (2000)	$f(x) = \dfrac{1}{x^{3/2}(1+x^{3/2})}$ $h(c) = \dfrac{2}{3}\ln\left(c^{3/2}+1\right)$ $p(\kappa) = \dfrac{G_{3,9}^{7,3}\left(\frac{\kappa^6}{46656}\,\Big|\,\frac{1}{12}\,\begin{pmatrix}2,5,11\\2,2,5,6,8,10,11,0,4\end{pmatrix}\right)}{4\sqrt{3}\pi^{5/2}|\kappa|}$ ~~$l(x) = \dfrac{2\pi}{x}\int_0^\infty y f(y) F_{3/2}^M(x,y)dy$~~ $\underline{l(x) = \dfrac{2\pi}{x}\int_0^\infty y f(y) F_{3/2}^M(x,y)\mathrm{d}y}$
Hernquist Hernquist (1990) Sheth et al. (2001b)	$f(x) = \dfrac{1}{x(1+x)^3}$ $h(c) = \dfrac{c^2}{2(1+c)^2}$ $p(\kappa) = \dfrac{1}{4}\left[-|\kappa|(2\mathrm{Ci}|\kappa|\sin|\kappa| + \pi\cos(\kappa)) + 2\kappa\mathrm{Si}(\kappa)\cos(\kappa) + 2\right]$ $p(\kappa,c) = 1/2 - \dfrac{\sin(c\kappa) + (c+1)\kappa\cos(c\kappa)}{2\kappa(c+1)^2}$ $\quad + \dfrac{k}{2}\left[\sin(\kappa)(\mathrm{Ci}(c\kappa+\kappa) - \mathrm{Ci}(\kappa)) + \cos(\kappa)(\mathrm{Si}(\kappa) - \mathrm{Si}(c\kappa+\kappa))\right]$ $l(x) = \dfrac{16\pi(h_1 - h_2)}{x^4 * (2+x)^4}$ $h_1(x) = \dfrac{24 + 60x + 56x^2 + 24x^3 + 6x^4 + x^5}{1+x}$ $h_2(x) = 12(1+x)(2+2x+x^2)\ln(1+x)/x$
GeneralizedNFW Ma and Fry (2000)	$f(x) = \dfrac{1}{x^\alpha(1+x)^{3-\alpha}}, \quad h(c) = -(-c)^\alpha c^{-\alpha} B_{-c}(3-\alpha, \alpha-2)$ $p(\kappa) = \dfrac{2^{-\alpha}}{\sqrt{\pi}\Gamma(3-\alpha)}G_{2,4}^{3,2}\left(\dfrac{\kappa^2}{4}\,\Big|\,\begin{matrix}(\alpha-2)/2,(\alpha-1)/2\\0,0,1/2,-1/2\end{matrix}\right)$ ~~$l(x) = \dfrac{2\pi}{(2-\alpha)x}\int_0^\infty y f(y) F_\alpha^N(x,y)dy$~~ $\underline{l(x) = \dfrac{2\pi}{(2-\alpha)x}\int_0^\infty y f(y) F_\alpha^N(x,y)\mathrm{d}y}$
GeneralizedMoore Ma and Fry (2000)	$f(x) = \dfrac{1}{x^\alpha(1+x^{3-\alpha})}$ ~~$l(x) = \dfrac{2\pi}{x}\int_0^\infty y f(y) F_\alpha^M(x,y)dy$~~ $\underline{l(x) = \dfrac{2\pi}{x}\int_0^\infty y f(y) F_\alpha^M(x,y)\mathrm{d}y}$
Uniform	$f(x) = 1, \quad h(c) = c^3/3, \quad p(\kappa,c) = \dfrac{\sin(c\kappa) - c\kappa\cos(c\kappa)}{\kappa^3}$
Einasto Einasto (1965)	$f(x|\alpha) = \exp\left(-\dfrac{2(x^\alpha - 1)}{\alpha}\right)$ $h(c|\alpha) = \dfrac{e^{2/\alpha}}{\alpha}\left(\dfrac{2}{\alpha}\right)^{-3/\alpha}\Gamma\left(\dfrac{3}{\alpha}, \dfrac{2c^\alpha}{\alpha}\right)$

Table 6: Summary of included `Profile` models. Note that for p and l, functions explicitly including c are the usual profiles truncated at r_Δ, whereas those that do not explicitly include c are not truncated (these are rather unhelpful in terms of the HM). G is the Meijer-G function. The functions T_i in the NFW definition are found in the text, ~~Eq.28.~~ Eq.85.
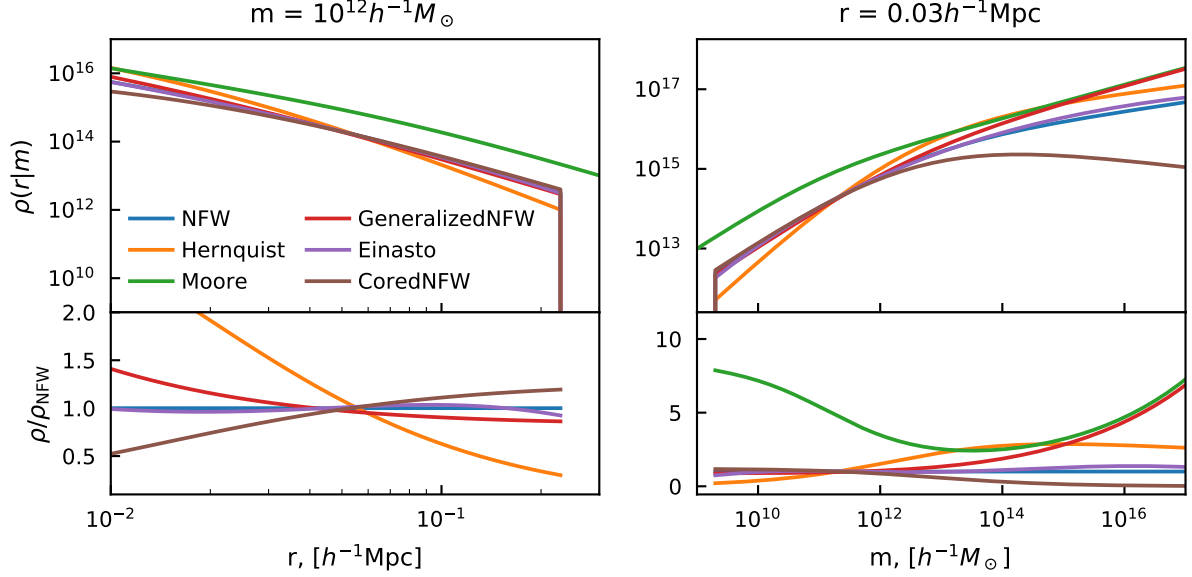
Figure 10: Illustration of the halo profile models implemented in `halomod`. Left panel shows the profile as a function of radius at $m = 10^{12} h^{-1} M_\odot$, and the right panel shows the profile as a function of mass at $r = 0.03 h^{-1}\text{Mpc}$. The general trends are similar, though the inner and outer slopes vary between models. The generalized NFW shown here has $\alpha = 1.5$.

Ref.	Name	Formula	Params.
Bullock et al. (2001)	`Bullock01`	$K\dfrac{1+z_c}{1+z}$	$F = 0.001$, $K = 3.4$, z_c given in Ref.
Ludlow et al. (2016)	`Ludlow16`	See ref.	
Ludlow et al. (2016)	`Ludlow16Empirical`	$\dfrac{c_0(\nu/\nu_0)^{-\gamma_1}}{\left(1 + \sqrt[\beta]{\nu/\nu_0}\right)^{\beta(\gamma_2-\gamma_1)}}$	$c_0 = 3.395(1+z)^{0.215}$, $\beta = 0.307(1+z)^{0.54}$, $\gamma_1 = 0.628(1+z)^{-0.047}$, $\gamma_2 = 0.317(1+z)^{-0.893}$
Bullock et al. (2001)	`Bullock01_Power`	$\dfrac{a}{(1+z)^c}\left(\dfrac{m}{M_\star}\right)^b$	$a = 9$, $b = -0.13$, $c = 1$
Duffy et al. (2008)	`Duffy08`		$a = 6.71$, $b = -0.091$, $c = 0.44$, $M_\star = 2 \times 10^{12}$
Zehavi et al. (2011)	`Zehavi11`		$a = 11$, $b = -0.13$, $c = 1$, $M_\star = 2.26 \times 10^{12}$

Table 7: Summary of concentration-mass-redshift relations implemented in HALOMOD. Additional models can be imported from the ~~COLOSSUS~~ COLOSSUS code. Note that the relation from Duffy et al. (2008) has multiple sets of parameters for different mass definitions and different halo samples. HALOMOD includes all of these, and they can be manually specified and/or intelligently chosen.
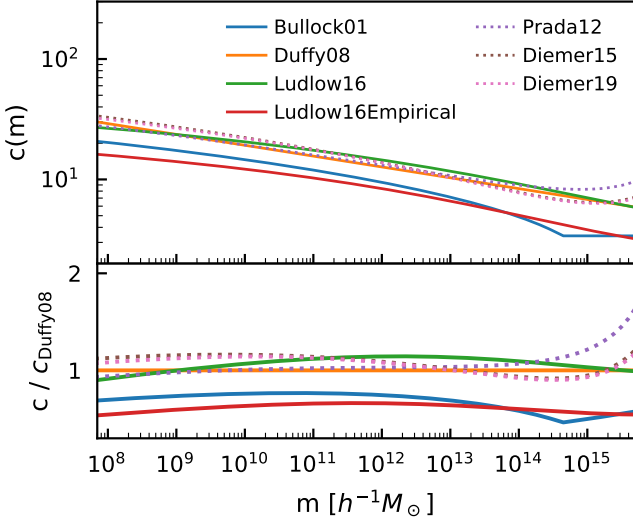
Figure 11: Illustration of the concentration-mass relations implemented in HALOMOD, focusing on analytic models (with the exception of the class power-law fit of ~~Duffy et al. (2008)~~ Duffy et al. 2008). Models plotted with dotted curves are obtained via the explicit interface with COLOSSUS, and are able to be used within the entire halo model framework.

if $\langle N_c \rangle$ is a step function from zero to one at some mass (and $\langle N_s \rangle = 0$ when $\langle N_c \rangle = 0$) then the reverse is true.

The following points outline our approach to this condition:

1. The average satellite occupancy is taken to be the average over *all* haloes, with and without centrals. This has subtle implications for how to mock up the galaxy population, because if one requires a central before placing a satellite, then the average number of satellites placed into *available* haloes is increased if the central occupation is less than 1.

2. If the central condition is enforced, then for all HOD classes (except see point 4), the mean satellite occupancy is modified. If the defined occupancy is $\langle N_s \rangle'$, then the returned occupancy is $\langle N_s \rangle = \langle N_c \rangle \langle N_s \rangle'$. This merely ensures that $\langle N_s \rangle = 0$ when $\langle N_c \rangle = 0$. Note that this will change the interpretation of parameters in the $\langle N_s \rangle$ model, unless $\langle N_c \rangle$ is a simple step function.

3. The pair-wise counts involve a term ~~$\langle N_c N_s \rangle$~~ $\langle N_c N_s \rangle$. When the central condition is enforced, this reduces trivially to $\langle N_s \rangle$. However, if the central condition is not enforced we *assume* that the variates ~~N_c and N_s~~ N_c and N_s are uncorrelated, and use ~~$\langle N_c N_s \rangle = \langle N_c \rangle \langle N_s \rangle$~~ $\langle N_c N_s \rangle = \langle N_c \rangle \langle N_s \rangle$.

4. For A HOD class that is defined with the central condition intrinsically *satisfiable* (i.e. $\langle N_s \rangle = 0 \Rightarrow \langle N_c \rangle = 0$), a variable can be set in the class definition, which will avoid the extra modification of point 2. Note that due to our above observation that being *satisfiable* does not in general imply that it has been *satisfied*, the pairwise counts still depend on whether

the user asserts that the central condition is enforced or not (note that in the one case in which *satisfiable* implies *satisfied*, namely a step function $\langle N_c \rangle$, the pairwise counts are the same with or without the central condition).

We present the implemented models in Table 8. An illustration of the various effects introduced by each HOD is presented in ~~figure~~ Fig. 6. Note that many of the HOD models are extensions of the more simple models, and are equivalent when setting the extra parameters to zero or one. Typically, HALOMOD will by default render these models the same as the more simple models, with the option of changing the extra parameters.

In addition to the galaxy-focused HOD models presented in Table 8, HALOMOD provides base classes for diffuse tracers of the underlying DM distribution, such as HI gas. In these distributions, it is assumed that there is no distinction between central and satellite, and the only relevant quantity is the mean value of the tracer squared (as a function of mass), which enters the power spectrum integral. Furthermore, non-unitless HODs are available in which the occupation can be defined in terms of a *quantity* (eg. HI brightness temperature), but in which that quantity is assumed to be confined to discrete locations within the halos (like galaxies). In these distributions, the central-satellite distinction is maintained.

4.3.10. Halo Exclusion

The primary task of the `Exclusion` component is to evaluate ~~Eq. 15 (or Eq. 16~~ Eq. 15 (or Eq. 16 if appropriate) and the associated (potentially modified) mean density, ~~Eq. 54.~~ Eq. 54. Each model, as discussed in §2.9, is quite unique, and therefore there is not a great deal of redundancy that can be mitigate by class inheritance. Nevertheless, for the sake of consistency, we implement exclusion using the standard `Component` architecture.

Performance becomes a definite issue with these calculations, especially for those requiring many double-integrations. Thus we introduce alternative methods, where necessary, accelerated by just-in-time compilation using NUMBA.

We show the effects of the halo exclusion for seven of the implemented models (two of them being modifications to ~~$P_{hh}^c(k)$~~ $P_{hh}^c(k)$ rather than physical halo exclusion models) in ~~figure~~ Fig. 5. Of these models, it is expected that `DblEllipsoid` and `NgMatched` are the most accurate, and the latter is far more performant.

4.4. Extra functionality

~~In addition to~~ the core `Framework` and `Component` elements, we have implemented several pieces of functionality aimed at performing the most commonly required tasks.

4.4.1. Command-line interface

Though we expect that the vast majority of usage will be interactive or via custom scripts, for additional flexibility

Ref.	$\langle N_{\rm c} \rangle_m$	$\langle N_{\rm s} \rangle_m$
Zehavi et al. (2005)	$H(m - M_{\min})$	$\left(\frac{m}{M_1}\right)^\alpha$
Zheng et al. (2005)	$\frac{1}{2}\left[1 + \mathrm{erf}\left(\frac{\log m - \log M_{min}}{\sigma_{\log M}}\right)\right]$	$\left(\frac{m - M_0}{M_1'}\right)^\alpha$
Tinker and Weinberg (2005)	$H(m - M_{\min})$	$\exp\left(-\frac{M_{\rm cut}}{m - M_{\min}}\right)\left(\frac{m}{M_1'}\right)H(m - M_{\min})$
Geach et al. (2012)	$F_c^B(1 - F_c^A)\exp\left[\frac{\log_{10}(m/M_c)^2}{2\sigma_{\log M}^2}\right]$ $\times F_c^A\left[1 + \mathrm{erf}\left(\frac{\log_{10}(m/M_c)}{\sigma_{\log M}}\right)\right]$	$F_s\left[1 + \mathrm{erf}\left(\frac{\log_{10}(m/M_{min})}{\delta_{\log M}}\right)\right]\left(\frac{m}{M_{min}}\right)^\alpha$
Contreras et al. (2013)	$\mathcal{N}_c^{\rm Geach}$, $\sigma_{\log M} \to x\sigma_{\log M}$	$\mathcal{N}_s^{\rm Geach}$

Table 8: Summary of included HOD parameterisations. Here H is the Heaviside step-function. Note that this table does not include HOD models for diffuse tracers.

we also provide a command-line interface (CLI), which provides a convenient way to compute specified quantities for a range of parameters. This may be useful for batch scripts which only require computation of basic quantities over a large parameter space, or for creating simple interfaces with other programming languages.

The underlying machinery for the CLI is contained in the `functional` module, which provides a top-level interface for calculating given quantities within a given `Framework`, for all combinations of a set of given parameters. Due to the homogeneity of the `Framework` definitions, this interface needs only be defined once for all frameworks[37].

The most challenging aspect of this function is the ordering of the implicit loops. As an example, consider a case where the user wishes to range over both redshift and Ω_m to calculate $n(m)$. It is much more efficient to use redshift as the inner loop, since fewer quantities depend on it, and therefore fewer re-calculations will be performed. This implicit ordering of parameters is accounted for in the routine (if necessary) by performing a very fast low-resolution calculation and determining the number of child quantities for each parameter. This is made possible through the caching system which we have already described (cf. §4.1.3.

Along with the calculated quantities, the routine optionally returns unique labels for each of the iterations, since the return order is not pre-specified.

The CLI wraps this routine, allowing any parameter to be specified by name, either as a scalar or list of values to be iterated over. The CLI ingests a configuration file in TOML[38] format, but may be over-ridden via command-line arguments. The CLI outputs a complete configuration file (with all default parameters set explicitly) which can

be used to re-compute the quantities later. It also outputs each desired quantity in its own file.

4.4.2. HOD population

Quite apart from the analytic formalism outlined in this paper, the HOD allows for creating galaxy catalogues by directly populating halos from N-body simulations. Though less efficient than its analytic counterpart, this method is more robust, since it does not depend on approximations for the several halo-based components. Indeed, several authors have used this method, either alone or in conjunction with the analytic calculation (a. Skibba et al., 2015; Zheng and Guo, 2015). It is also useful as a sanity check.

We provide a very basic set of tools for populating halo catalogues with galaxies using the HOD models included in HALOMOD, synthesised into a CLI. Our implementation assumes a Bernoulli (Poisson) distribution for centrals (satellites), and a spherically symmetric profile for each halo, corresponding precisely to the analytic calculations. It is possible that this functionality will be extended in future versions to account for triaxial profiles (Jing and Suto, 2002) and a range of galaxy classes (such as colour, cf. a. Skibba and Sheth (2009)).

5. TheHaloMod web-application

Murray et al. (2013b) presented HMFCALC, an online HMF calculator that has since been widely used in the community. Web-applications can be useful, as they circumvent the overhead of installing custom software and using a command-line or interpreter interface. As such, they are more readily employed by researchers seeking to obtain a single model to corroborate the output of a simulation or observation. Perhaps more interestingly, the intuitive and graphical nature of web-apps make them extremely useful for educational purposes – whether in a classroom setting, or for self-learning. It is with these motivations

---

[37]Indeed, due to HALOMOD being inherited from HMF, HALOMOD requires only a thin wrapper around the function in HMF to update some defaults.

[38]https://toml.io/en/

that we present a successor to HMFcalc[39] that computes the full range of halo model quantities we have outlined in this paper – called THEHALOMOD[40].

5.1. Interface

The interface of THEHALOMOD is expansive but simple. ~~The user is presented immediately with a default model, which occupies a central plot. A~~ It is a single-page application (SPA) that uses the VUE.JS[41] framework to render input forms dynamically. The primary view presented to the user consists of a left-hand column containing an overview of the models the user has created and a drop-down menu ~~lets the user select which quantity to view. Using a TracerHaloModel on the backend, these quantities are computed on-the-fly and then cached so they can be view again quickly.~~ for downloading various data products, and a right-hand panel for viewing plots of arbitrary halo model quantities.

~~Below the plot itself is a table of computed models (consisting of just the default plot at first), each of which has a unique user-provided label. The user can remove or edit a model, or create a new model based on any existing model. Editing or model creation bring the user to~~ The Models panel is initially empty until the user creates a model. Creating a model opens the input form view (technically this is on the same page, and overlays the primary view). We will discuss this view momentarily, but once a model has been defined, and a unique label is designated for it, the model appears in its own row in the Models panel on the main view. From here, the model may be renamed, edited (taking the user back to the input form view), cloned or deleted. Cloning the model makes a copy of model, but also takes the user to the input form view, to edit its parameters. In addition, there is an option for the user to report a bug specific to a particular model, which—along with a useful message from the user—collects and sends detailed information about the model to the developers.

The Download panel contains a single drop-down menu, offering choices of file downloads. Presently, the available options are (i) an image of ~~the same interface containing a rather large input form. The form itself is separated into tabs – one for each,~~ plot (in SVG format), (ii) configuration files containing all of the parameters specified for each model (these come as ZIP archives, with one file per user-specified model, and each file can be in TOML or JSON format, where the TOML files are precisely in the format required to be input to the CLI, cf. §4.4.1), and (iii) a ZIP archive containing ASCII data files with all possible halo model quantities stored in them (i.e. all quantities defined within

the TracerHaloModel). The latter makes it simple for the user to produce their own plots with their preferred plotting tools.

The Plot panel contains a number of tools to customize what is being plotted. Most importantly, there are two drop-downs that specify the quantities that comprise the x- and ~~one for each. Each, in particular, has a~~ y-axes. While typically the x-axis will either be halo mass, comoving scale or comoving wavenumber, more general quantities can be chosen (for example the peak-height ν instead of mass). The options available for the y-axis are dynamically updated based on the value of the x-axis, and include every possible quantity in the TracerHaloModel. Beyond this, there are options for setting the scale on either axis to be logarithmic (the default is intelligently chosen for each quantity). Finally, the plot itself contains the chosen quantity for all models defined by the user. A legend to the right of the plot indicates corresponding models.

The interface is further available in both light and dark modes. It is also fluid – adjusting to fit on phone, tablet and monitor screens.

The input form view (cf. left-hand panel of Fig. 1) presents all possible input parameters to a model. This is quite a few possible parameters – including the choice of each kind of component model as well as all possible parameters to each of those models (eg. HOD models, bias models and HMF models). To make the interface concise and simple, the view consists of a left-hand panel that contains a dynamic index of parameter groups (essentially each of the components) so that particular parameters are easier to locate. Each component has a similar layout for its inputs: a drop-down menu ~~for choosing the particular~~ to choose the model, and a ~~unique set of input parameters for each model~~ dynamically-rendered set of numerical inputs to change the unique parameters of the model (i.e. if Tinker08 is chosen as the HMF model, only its parameters are shown to the user, and not those of any other mass function). The user is able to create and edit as many models as they wish, and these models persist between sessions. They are able to refresh and remove all their models via the Model panel on the main page. ~~Once parameters have been chosen (including the unique label),~~

5.2. Architecture

The overall architecture of THEHALOMOD is broken up into two parts: server and client. From a high-level perspective, the server handles the API portion of ~~the user is taken back to the main plot interface, and is free to continue adding and editing models~~ website, and the client holds the front-end. This way, the front-end can benefit from modern JavaScript UI libraries and the backend interfaces with the Python libraries HMF and HALOMOD.

~~The user may also download various pieces of data: the plot itself, a readable list of parameters for each of the models, or data files containing *all* the quantities of the halo model for each model.~~ Automatic testing on

---

[39]HMFcalc was located at https://hmf.icrar.org. This address now automatically forwards to the address of THEHALOMOD, at https://thehalomod.app. All the functionality of HMFcalc is available in THEHALOMOD.

[40]The *code* for THEHALOMOD is open-source and available at https://github.com/halomod/TheHaloMod-SPA.

[41]https://www.vuejs.org/

both server and client are performed via Github Actions, which also handle automatic continuous deployment on every commit to the main branch.

5.3. ~~Architecture~~

~~is powered by DJANGO[42] and uses the BOOTSTRAP[42] CSS framework and JQUERY[42] Javascript library to achieve a reasonable UI with minimal fuss~~The server is built with the FLASK[42] Python web framework. This framework handles the (very simple) API for the single-page app. Since it is a Python framework, it is able to interface directly with HALOMOD. Models defined by the user (on the input form view) are serialized as a set of parameters, and used to construct `TracerHaloModel` objects on the server. These objects are serializable by construction, and are stored on the server (uniquely for each user) using REDIS[43]. Each time a quantity is requested by the user (to create the plots on the main view), each model object is deserialized, and the relevant quantity is computed (or obtained from cache if it has already been computed, cf. §4.1.3). Cloning a model (via the Model panel on the main view) literally copies the instance of the model on the server, and uses the inherent caching and updating ability of HMF `Framework` objects to improve the speed of calculation of the new model. In this way, the web-app essentially provides a thin, visual, wrapper around the full capability of a `TracerHaloModel` object.

The frontend client is built with the VUE.JS web framework, and uses the MATERIAL[44] CSS framework. This provides a simple yet modern and flexible framework for both visual and interface design. The plots are created using `d3.js`[45].

~~Architecturally, makes great use of the object-oriented forms in DJANGO, in concert with the consistency of all the~~ `Components`. ~~This consistency means that very few lines of code are required to create separate form tabs for each .~~ To enhance performance, an initial high-resolution object is pre-computed on the server, which is cloned and updated upon the first model definition by the user. In particular, this is very effective at producing high-resolution transfer functions from CAMB, which otherwise are computationally intensive.

~~Javascript functionality is kept to a minimum, but provides extremely useful UI improvements in some key cases. In particular, each model has potentially unique parameters. Javascript is used to make only relevant parameters visible to the user, dependent on their current choices. It is also used to provide a smooth experience when changing the quantity to be viewed in the plot (i. e. this can occur without reloading the page).~~

The THEHALOMOD server is hosted by the Low-Frequency Cosmology (LOCO) Lab at Arizona State University. The client is separately hosted by NETLIFY[46], which aids load speeds for users worldwide. Deployment is aided by containerizing the entire application using Docker[47].

6. Example Application

To illustrate the utility of HALOMOD in semi-realistic applications, we here present a worked example of fitting parameters to mock galaxy data[48]. We do not use real data, but instead create some simplistic mock data using HALOMOD itself.

We note that we do *not* include native fitting capabilities in HALOMOD. The reason for this is that applications in constraining parameters, and their relevant likelihoods, are so diverse. It is impossible to capture the range of realistic likelihoods, and it is more useful to let the user adopt one of the many excellent Bayesian (or more general fitting) frameworks in the ~~Python~~ PYTHON ecosystem[49]. However, HALOMOD was partially motivated by the desire to perform HOD fits on galaxy data, and it is thus written in a manner conducive to this end. For instance, each `Framework` is pickle-able by the `dill.pickle` library. Furthermore, due to the fact that all quantities of interest are simply obtained by accessing attributes, quite general likelihoods can be written that are able to dynamically switch between different quantities. We will utilise this feature in our example here.

In our example, we will simply create mock data to which we fit a matching HOD model. The mock data is generated as $\xi_{gg}(r)\xi_{\rm gg}(r)$, with diagonal covariance matrix and Gaussian noise at an amplitude of 10% of the amplitude of the correlation function itself. In addition, we use the constraint of galaxy number density (with a Gaussian uncertainty of $10^{-4}h^3{\rm Mpc}^{-3}$) to break a degeneracy in the HOD parameters. The appropriate likelihood is thus a simple χ^2 between the mock and model $\xi_{gg}(r)\xi_{\rm gg}(r)$, along with an extra Gaussian term for the mean galaxy number density.

We use EMCEE to perform the actual Bayesian inference in this example, though it should be clear how to adapt the likelihood to other frameworks. We can write a fairly generic likelihood function:

```
def log_prob(
  param_values: np.ndarray,
  param_names: List[str],
  data: np.ndarray,
  model: TracerHaloModel,
  derived: Optional[Tuple[str]]=()
):
```

[42]
[42]
[42]
[42] https://flask.palletsprojects.com/en/2.0.x/
[43] https://redis.io/
[44] https://vuematerial.io/
[45] https://d3js.org/

[46] https://d3js.org/
[47] https://www.docker.com/
[48] The full example notebook can be found at https://halomod.readthedocs.io/en/latest/examples/fitting.html
[49] Examples include EMCEE, PYMC3, PYSTAN and COBAYA.

```python
# Pack parameters into a dict
params = dict(zip(param_names,
  param_values))

# Allow for simple bounded flat priors.
bounds = bounds or {}
for key, val in params.items():
  bound = bounds.get(key, (-np.inf, np.inf
  ))
  if not bound[0] < val < bound[1]:
    return -np.inf, []

# Update model with all fitting parameters
params = flat_to_nested_dict(params)
model.update(**params)

# Get arbitrary derived data
derived = [getattr(model, d) for d in
  derived]

# Calculate chi^2 likelihood
logl = chi_square(
  model=model.corr_auto_tracer,
  data=data,
  sigma=0.2 * np.abs(model.
  corr_auto_tracer)
)

# Return derived to make emcee "blobs"
return logl, derived
```

This function contains a reference to another custom function, `flat_to_nested_dict`, whose purpose is to take a dictionary with dot-path keys and convert it to a nested dictionary, so that eg. {'nested.key': 3} is converted to {'nested': {'key': 3}}. This makes it possible set arbitrary parameters of components within the Tracer-HaloModel, as we shall see.

Furthermore, notice that arbitrary derived data is able to be extracted due to the consistent API of the `Framework`, in which every quantity is accessed as an attribute. Due to the efficient caching system (cf. §4.1.3 and Appendix A), the order in which quantities are accessed is irrelevant, and only the quantities that are required are actually calculated. We return these arbitrary derived data as a list of `emcee` "blobs", so that we can plot posteriors of the derived parameters.

This likelihood function is more than generic enough for this example, and many others, and is a good starting place for most likelihoods that the user might wish to write. One clear deficiency in the log probability function is the absence of sophisticated priors on the parameters. Most sophisticated Bayesian frameworks have abstract mechanisms for defining priors, but `emcee` assumes that the priors will be added to the probability inside the `log_prob` function. Due to the non-general nature of such a code, we omit it for these examples, implicitly assuming flat priors on all parameters. We do however include a potential bound on each parameter, which should serve as a simple starting point for extensions to more complex priors.

We start the MCMC chains in a small Gaussian ball around the true parameters (or close to the true parameters in the case of the mismatched model) and let them expand to fill the posterior[50]. The chains have 100 walkers, and comprise 10000 iterations, of which the first 1000 are removed as burnin. As this is a toy examples, explicit convergence tests are not performed as they should be in real applications.

Our mock data is created with a default `TracerHaloModel` set at a redshift of $z = 0.2$, and using the transfer model of Eisenstein and Hu (1998). The HOD model is `Zehavi05` (cf. Table 8), with $M_{min} = 10^{12}$, $M_1 = 10^{12.8}$ and $\alpha = 1.05$.

Constraining the parameters is simple. We construct the sampler like so:

```python
sampler = emcee.EnsembleSampler(
  nwalkers = 100,
  ndim = 3,
  log_prob_fn = log_prob,
  kwargs = {
    'param_names': [
      'hod_params.M_min',
      'hod_params.M_1',
      'hod_params.alpha'
    ],
    'data': (mock_data, mock_ngal),
    'model': model,
    'derived': [
      'satellite_fraction',
      'mean_tracer_den',
      'bias_effective_tracer',
      'corr_auto_tracer'
    ],
  },
  blobs_dtype=[
    ("sat_frac", float),
    ("tracer_den", float),
    ("bias_effective_tracer", float),
    ("corr_auto_tracer", (float, len(
    mock_data)))
  ]
)
```

and then create the initial positions and sample for 10000 iterations:

```python
initialpos = np.array([
  fiducial_model.hod.params['M_min'],
  fiducial_model.hod.params['M_1'],
  fiducial_model.hod.params['alpha']
]) + 1e-4 * np.random.normal(
  size=(sampler.nwalkers, sampler.ndim)
)

sampler.run_mcmc(initialpos, nsteps=10000)
```

Note that in the setup of the `EnsembleSampler` we have specified the parameters we wish to constrain simply by

---

[50]This is recommended in the EMCEE documentation.

passing a list of dot-pathed names. The first part of the name refers to the `parameter` name in the `Framework` that we would like to update. In this case, `hod_params` is itself a dictionary of parameters for the HOD, and so the second part of the dot-path is the name of the parameter within that dictionary. This makes it possible to constrain any continuous parameter defined at any level of the framework.

Note also that the `model` variable is itself a `TracerHaloModel` instance, cloned from the original fiducial model from which we created the `mock_data` using the simple command `model = fid_model.clone()`. This `model` is mutable and is internally updated in the `log_prob` function, so it is useful to keep the original fiducial model untouched.

After removing burnin and thinning by every 5th sample to reduce correlations, we can make Fig. 12, which shows that that parameters were recovered with moderate accuracy. Note that in this "corner plot", we have also shown the joint posteriors of the three derived scalars – the satellite fraction, the mean galaxy density and the linear galaxy bias.

Furthermore, Fig 13 shows the posterior 68% percentile range of the correlation function, along with the ratio of the mock data to the median. The posterior of the model $\xi(r)$ is here easily obtained from the output sampler: `sampler.get_blobs(flat=True)['corr_auto_tracer']`. Notice that the residuals are noise-like, as is expected from this toy model.

It is easy to imagine extensions to this example – using projected or angular correlation functions, more sophisticated HOD models, allowing cosmology to be free, or using multiple datasets simultaneously. In most cases, the necessary modifications to the example code presented here will be small, illustrating the simplicity and flexibility of HALOMOD.

7. Future Development

The current implementation of HALOMOD (likewise HMF) forms a solid framework for evaluating halo model quantities. Its flexible and extendable architecture enables user-side development with updated components and paradigms (eg. warm dark matter, or alternate dark energy scenarios). However, several features considered for future versions are worth mentioning:

Additional and higher-order statistics. Though the most popular usage of the halo model has been to analyse the ~~two-point~~ 2-point clustering of galaxies, there has also been a range of studies with other quantities. In particular, galaxy-galaxy weak lensing, as an independent observable, has been shown to be able to break HOD degeneracies (Leauthaud et al., 2012). Future versions of HALOMOD will include these quantities in addition to higher-order statistics such as the bispectrum and 3-point correlation function.

Alternative Cosmologies. As already mentioned, HMF already has the infrastructure to support alternative cosmological scenarios. However, only one scenario is actually fleshed out – WDM. It will be interesting to implement more exotic models such as Early Dark Energy, or $f(R)$ gravity. Ideally, such implementations will arise from the community.

Support for CLASS. CLASS[51] (Lesgourgues, 2011) is a popular Boltzmann-solver for the transfer function. As such, it is a prime candidate as an extra model for the `Transfer Component`.

Performance and Accuracy Improvements. There are known bottlenecks in the code – especially as related to Hankel transforms and halo exclusion – both in memory and CPU. These do not affect the user experience greatly over a single model, but make a definite impact when running many models. Further improvements in the interpolation and integration required for Hankel transforms will be highly useful. Reducing the number of nodes required for these integrals will also lower the burden of halo exclusion.

Improved Web Interface. Though THEHALOMOD has a clean and usable interface, it was designed and implemented by a definite non-expert in the field of web development. Many advanced features that would enhance the user experience are possible: from fitting more useful information into a one-page app, to making the figure(s) interactive, to having the ability to store "sessions" for later use.

Most importantly, HMF, HALOMOD and THEHALOMOD are all open-source and intended to be projects nurtured and maintained by the community. This will make them of more impact and usefulness for the community. Up-to-date planned improvements for all of these codes can be found by browsing their issues on GitHub[52]. We highly encourage the reader to suggest new features via this mechanism!

8. Summary

We have presented a new code, HALOMOD, accompanied by a web-application, THEHALOMOD, for the calculation of quantities within the halo model framework. This code aims to satisfy six principles: simple, intuitive, flexible/extendible, comprehensive, efficient and open; we have shown how these criteria have shaped the architecture of the library.

HALOMOD is designed to be a valuable resource for the community. To this end, we have presented an illustrative application which helps to get a flavour of how HALOMOD may be used 'in the wild' for constraining physical parameters using galaxy surveys.

We expect THEHALOMOD to be useful for both theorists and observers, but are especially excited about the

---

[51] https://lesgourg.github.io/class_public/class.html
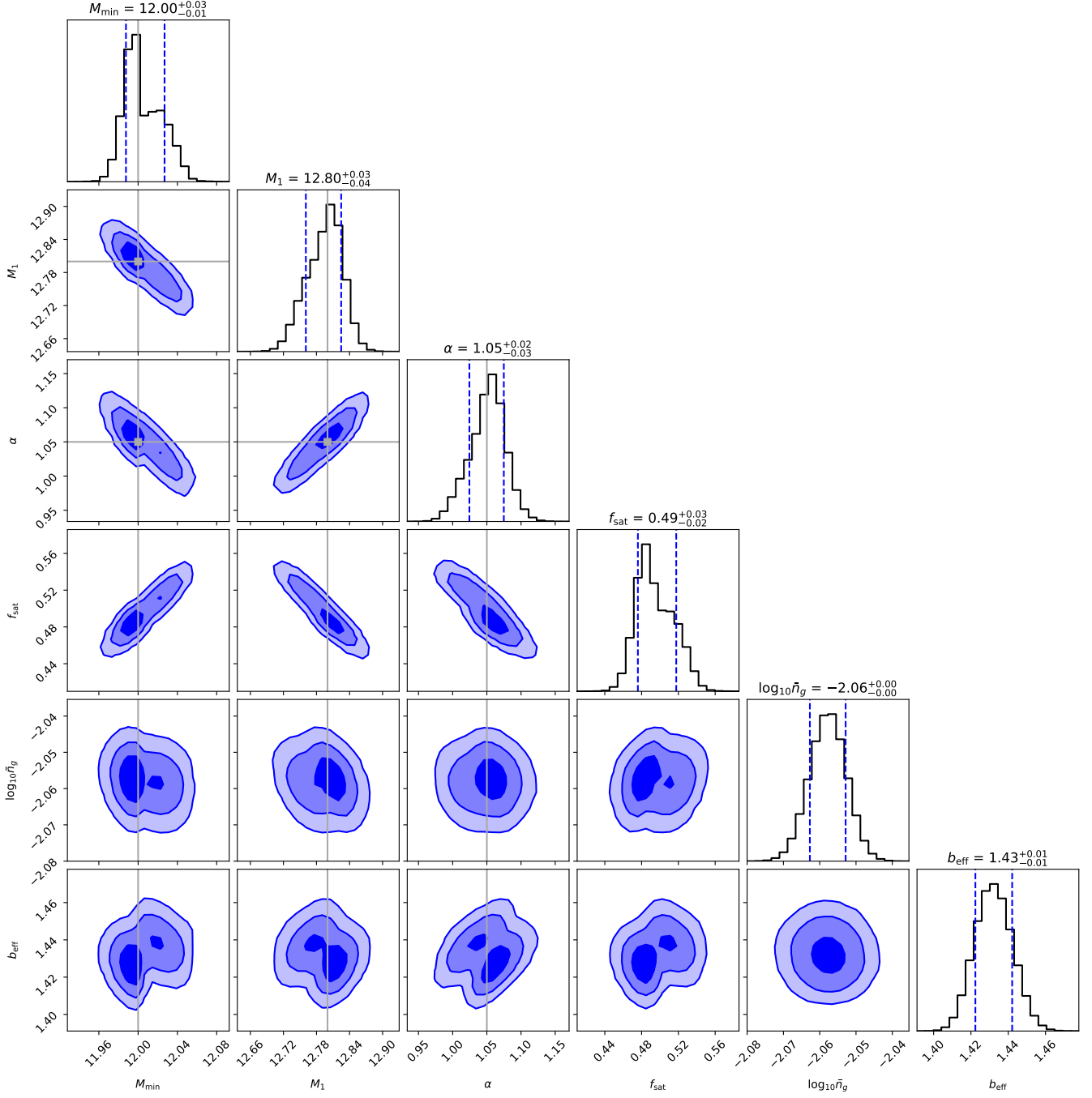[52] Eg. https://github.com/halomod/halomod/issues

Figure 12: Corner-plot showing joint posteriors for each of the three HOD parameters, as well as the three derived parameters (satellite fraction, mean galaxy number density and linear galaxy bias). Example code to reproduce this plot is found throughout §6.
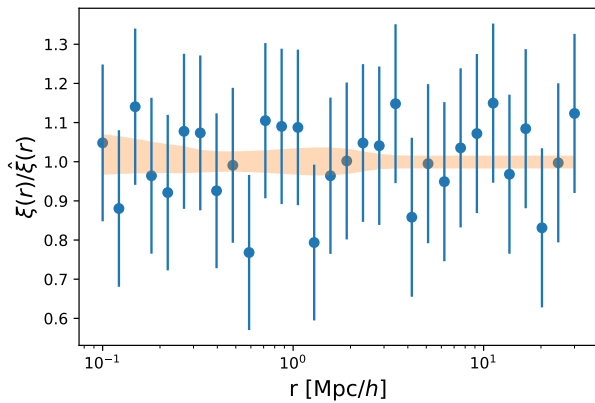
Figure 13: Ratio of mock data to the median posterior of $\xi(r)$. Here, the median posterior is calculated as the median (per r-bin) of the posterior *samples* (not the model at the median of the posterior of the parameters). Error-bars shown are 10% uncertainties as input to the mock data. The orange region is the 68% central quantile of the posterior. The residuals are noise-like.

prospects for using it educationally. THEHALOMOD also serves as an example and motivation for other broadly-applicable codes to make use of scientific web-applications in order to reach a wider user-base.

Acknowledgements

References

Astropy Collaboration, Price-Whelan, A.M., Sipőcz, B.M., Günther, H.M., Lim, P.L., Crawford, S.M., Conseil, S., Shupe, D.L., Craig, M.W., Dencheva, N., Ginsburg, A., VanderPlas, J.T., Bradley, L.D., Pérez-Suárez, D., de Val-Borro, M., Aldcroft, T.L., Cruz, K.L., Robitaille, T.P., Tollerud, E.J., Ardelean, C., Babej, T., Bach, Y.P., Bachetti, M., Bakanov, A.V., Bamford, S.P., Barentsen, G., Barmby, P., Baumbach, A., Berry, K.L., Biscani, F., Boquien, M., Bostroem, K.A., Bouma, L.G., Brammer, G.B., Bray, E.M., Breytenbach, H., Buddelmeijer, H., Burke, D.J., Calderone, G., Cano Rodríguez, J.L., Cara, M., Cardoso, J.V.M., Cheedella, S., Copin, Y., Corrales, L., Crichton, D., D'Avella, D., Deil, C., Depagne, É., Dietrich, J.P., Donath, A., Droettboom, M., Earl, N., Erben, T., Fabbro, S., Ferreira, L.A., Finethy, T., Fox, R.T.,

Garrison, L.H., Gibbons, S.L.J., Goldstein, D.A., Gommers, R., Greco, J.P., Greenfield, P., Groener, A.M., Grollier, F., Hagen, A., Hirst, P., Homeier, D., Horton, A.J., Hosseinzadeh, G., Hu, L., Hunkeler, J.S., Ivezić, Ž., Jain, A., Jenness, T., Kanarek, G., Kendrew, S., Kern, N.S., Kerzendorf, W.E., Khvalko, A., King, J., Kirkby, D., Kulkarni, A.M., Kumar, A., Lee, A., Lenz, D., Littlefair, S.P., Ma, Z., Macleod, D.M., Mastropietro, M., McCully, C., Montagnac, S., Morris, B.M., Mueller, M., Mumford, S.J., Muna, D., Murphy, N.A., Nelson, S., Nguyen, G.H., Ninan, J.P., Nöthe, M., Ogaz, S., Oh, S., Parejko, J.K., Parley, N., Pascual, S., Patil, R., Patil, A.A., Plunkett, A.L., Prochaska, J.X., Rastogi, T., Reddy Janga, V., Sabater, J., Sakurikar, P., Seifert, M., Sherbert, L.E., Sherwood-Taylor, H., Shih, A.Y., Sick, J., Silbiger, M.T., Singanamalla, S., Singer, L.P., Sladen, P.H., Sooley, K.A., Sornarajah, S., Streicher, O., Teuben, P., Thomas, S.W., Tremblay, G.R., Turner, J.E.H., Terrón, V., van Kerkwijk, M.H., de la Vega, A., Watkins, L.L., Weaver, B.A., Whitmore, J.B., Woillez, J., Zabalza, V., Astropy Contributors, 2018. The Astropy Project: Building an Open-science Project and Status of the v2.0 Core Package. The Astronomical Journal 156, 123. doi:10.3847/1538-3881/aabc4f.

Bardeen, J.M., Bond, J.R., Kaiser, N., Szalay, A.S., 1986. The statistics of peaks of Gaussian random fields. The Astrophysical Journal 304, 15–15. doi:10.1086/164143.

Behroozi, P.S., Wechsler, R.H., Conroy, C., 2013. THE AVERAGE STAR FORMATION HISTORIES OF GALAXIES IN DARK MATTER HALOS FROM z = 0-8. The Astrophysical Journal 770, 57–57. doi:10.1088/0004-637X/770/1/57.

Benson, A.J., Farahi, a., Cole, S., a. Moustakas, L., Jenkins, a., Lovell, M., Kennedy, R., Helly, J., Frenk, C., 2012. Dark matter halo merger histories beyond cold dark matter - I. Methods and application to warm dark matter. Monthly Notices of the Royal Astronomical Society 428, 1774–1789. doi:10.1093/mnras/sts159.

Berlind, A.A., Weinberg, D.H., Benson, A.J., Baugh, C.M., Cole, S., Frenk, C.S., Jenkins, A., Katz, N., Lacey, C.G., 2003. The Halo Occupation Distribution And The Physics of Galaxy Formation. The Astrophysical Journal 593, 1–25.

Beutler, F., Blake, C., Colless, M., Jones, D.H., Staveley-Smith, L., Campbell, L.A., Parker, Q., Saunders, W., Watson, F., 2013. The 6dF Galaxy Survey: Dependence of halo occupation on stellar mass. Monthly Notices of the Royal Astronomical Society 429, 3604–3618. doi:10.1093/mnras/sts637.

Bhattacharya, S., Habib, S., Heitmann, K., Vikhlinin, A., 2013. Dark Matter Halo Profiles of Massive Clusters: Theory versus Observations. The Astrophysical Journal 766, 32. doi:10.1088/0004-637X/766/1/32.

Bhattacharya, S., Heitmann, K., White, M., Lukić, Z., Wagner, C., Habib, S., 2011. MASS FUNCTION PREDICTIONS BEYOND ΛCDM. The Astrophysical Journal 732, 122–122. doi:10.1088/0004-637X/732/2/122.

Blake, C., Collister, A., Lahav, O., 2008. Halo-model signatures from 380 000 Sloan Digital Sky Survey luminous red galaxies with photometric redshifts. Monthly Notices of the Royal Astronomical Society 385, 1257–1269. doi:10.1111/j.1365-2966.2007.11925.x.

Blas, D., Lesgourgues, J., Tram, T., 2011. The Cosmic Linear Anisotropy Solving System (CLASS). Part II: Approximation schemes. Journal of Cosmology and Astroparticle Physics 2011, 034–034. doi:10.1088/1475-7516/2011/07/034.

Bocquet, S., Saro, A., Dolag, K., Mohr, J.J., 2016. Halo mass function: Baryon impact, fitting formulae, and implications for cluster cosmology. Monthly Notices of the Royal Astronomical Society 456, 2361–2373. doi:10.1093/mnras/stv2657.

Bond, J.R., Cole, S., Efstathiou, G., Kaiser, N., 1991. Excursion set mass functions for hierarchical Gaussian fluctuations. The Astrophysical Journal 379, 440–460. doi:10.1086/170520.

Bond, J.R., Efstathiou, G., 1984. Cosmic background radiation anisotropies in universes dominated by nonbaryonic dark matter. The Astrophysical Journal 285, L45–L45. doi:10.1086/184362.

Bosch, F.C.V.D., 2002. The universal mass accretion history of cold dark matter haloes. Monthly Notices of the Royal Astronomical Society 331, 98–110. doi:10.1046/j.1365-8711.2002.05171.x.

Bryan, G.L., Norman, M.L., 1998. Statistical Properties of X-Ray

Clusters: Analytic and Numerical Comparisons. The Astrophysical Journal 495, 80–99. doi:10.1086/305262.

Bullock, J.S., 2001. Shapes of dark matter halos. Arxiv e-prints .

Bullock, J.S., Kolatt, T.S., Sigad, Y., Somerville, R.S., Kravtsov, a.V., a. Klypin, a., Primack, J.R., Dekel, A., 2001. Profiles of dark haloes: Evolution, scatter and environment. Monthly Notices of the Royal Astronomical Society 321, 559–575. doi:10.1046/j.1365-8711.2001.04068.x.

Bullock, J.S., Wechsler, R.H., Somerville, R.S., 2002. Galaxy halo occupation at high redshift. Monthly Notices of the Royal Astronomical Society 329, 246–256.

Cacciato, M., Lahav, O., van den Bosch, F.C., Hoekstra, H., Dekel, A., 2012. On combining galaxy clustering and weak lensing to unveil galaxy biasing via the halo model. Monthly Notices of the Royal Astronomical Society 426, 566–587. doi:10.1111/j.1365-2966.2012.21762.x.

Carretero, J., Castander, F.J., Gaztañaga, E., Crocce, M., Fosalba, P., 2015. An algorithm to build mock galaxy catalogues using MICE simulations. Monthly Notices of the Royal Astronomical Society 447, 646–670. doi:10.1093/mnras/stu2402.

Chan, K.C., Sheth, R.K., Scoccimarro, R., 2017. Effective window function for Lagrangian halos. Physical Review D 96, 103543. doi:10.1103/PhysRevD.96.103543.

Chen, Z., Wolz, L., Spinelli, M., Murray, S.G., 2021. Extracting HI Astrophysics from Interferometric Intensity Mapping. MNRAS 502, 5259–5276. doi:10.1093/mnras/stab386.

Child, H.L., Habib, S., Heitmann, K., Frontiere, N., Finkel, H., Pope, A., Morozov, V., 2018. Halo Profiles and the Concentration-Mass Relation for a ΛCDM Universe. The Astrophysical Journal 859, 55. doi:10.3847/1538-4357/aabf95.

Clampitt, J., Sánchez, C., Kwan, J., Krause, E., MacCrann, N., Park, Y., Troxel, M.A., Jain, B., Rozo, E., Rykoff, E.S., Wechsler, R.H., Blazek, J., Bonnett, C., Crocce, M., Fang, Y., Gaztanaga, E., Gruen, D., Jarvis, M., Miquel, R., Prat, J., Ross, A.J., Sheldon, E., Zuntz, J., Abbott, T.M.C., Abdalla, F.B., Armstrong, R., Becker, M.R., Benoit-Lévy, A., Bernstein, G.M., Bertin, E., Brooks, D., Burke, D.L., Carnero Rosell, A., Carrasco Kind, M., Cunha, C.E., D'Andrea, C.B., da Costa, L.N., Desai, S., Diehl, H.T., Dietrich, J.P., Doel, P., Estrada, J., Evrard, A.E., Fausti Neto, A., Flaugher, B., Fosalba, P., Frieman, J., Gruendl, R.A., Honscheid, K., James, D.J., Kuehn, K., Kuropatkin, N., Lahav, O., Lima, M., March, M., Marshall, J.L., Martini, P., Melchior, P., Mohr, J.J., Nichol, R.C., Nord, B., Plazas, A.A., Romer, A.K., Sanchez, E., Scarpine, V., Schubnell, M., Sevilla-Noarbe, I., Smith, R.C., Soares-Santos, M., Sobreira, F., Suchyta, E., Swanson, M.E.C., Tarle, G., Thomas, D., Vikram, V., Walker, A.R., 2017. Galaxy–galaxy lensing in the Dark Energy Survey Science Verification data. Monthly Notices of the Royal Astronomical Society 465, 4204–4218. doi:10.1093/mnras/stw2988.

Cole, S., Kaiser, N., 1989. Biased clustering in the cold dark matter cosmogony. Monthly Notices of the Royal Astronomical Society 237, 1127–1127.

Comparat, J., Prada, F., Yepes, G., Klypin, A., 2017. Accurate mass and velocity functions of dark matter haloes. Monthly Notices of the Royal Astronomical Society 469, 4157–4174. doi:10.1093/mnras/stx1183.

Contreras, S., Baugh, C.M., Norberg, P., Padilla, N., 2013. How robust are predictions of galaxy clustering? Monthly Notices of the Royal Astronomical Society 432, 2717–2730. doi:10.1093/mnras/stt629.

Cooray, A., Sheth, R., 2002. Halo models of large scale structure. Physics Reports 327, 1–129.

Corasaniti, P.S., Achitouv, I., 2011. Excursion set halo mass function and bias in a stochastic barrier model of ellipsoidal collapse. Physical Review D 84, 023009–023009. doi:10.1103/PhysRevD.84.023009.

Correa, C.A., Wyithe, J.S.B., Schaye, J., Duffy, A.R., 2015. The accretion history of dark matter haloes – III. A physical model for the concentration–mass relation. Monthly Notices of the Royal Astronomical Society 452, 1217–1232. doi:10.1093/mnras/stv1363.

Coupon, J., Arnouts, S., van Waerbeke, L., Moutard, T., Ilbert, O., van Uitert, E., Erben, T., Garilli, B., Guzzo, L., Heymans, C., Hildebrandt, H., Hoekstra, H., Kilbinger, M., Kitching, T., Mellier, Y., Miller, L., Scodeggio, M., Bonnett, C., Branchini, E., Davidzon, I., De Lucia, G., Fritz, a., Fu, L., Hudelot, P., Hudson, M.J., Kuijken, K., Leauthaud, a., Fèvre, O.L., McCracken, H.J., Moscardini, L., Rowe, B.T.P., Schrabback, T., Semboloni, E., Velander, M., Le Fevre, O., McCracken, H.J., Moscardini, L., Rowe, B.T.P., Schrabback, T., Semboloni, E., Velander, M., 2015. The galaxy-halo connection from a joint lensing, clustering and abundance analysis in the CFHTLenS/VIPERS field. Monthly Notices of the Royal Astronomical Society 449, 1352–1379. doi:10.1093/mnras/stv276.

Dalal, N., Lithwick, Y., Kuhlen, M., 2010. The Origin of Dark Matter Halo Profiles. arXiv e-prints 1010, arXiv:1010.2539.

Diemer, B., 2018. COLOSSUS: A Python Toolkit for Cosmology, Large-scale Structure, and Dark Matter Halos. The Astrophysical Journal Supplement Series 239, 35. doi:10.3847/1538-4365/aaee8c.

Diemer, B., Joyce, M., 2019. An Accurate Physical Model for Halo Concentrations. The Astrophysical Journal 871, 168. doi:10.3847/1538-4357/aafad6.

Diemer, B., Kravtsov, A.V., 2015. A Universal Model for Halo Concentrations. The Astrophysical Journal 799, 108. doi:10.1088/0004-637X/799/1/108.

Diemer, B., More, S., Kravtsov, A.V., 2013. THE PSEUDO-EVOLUTION OF HALO MASS. The Astrophysical Journal 766, 25–25. doi:10.1088/0004-637X/766/1/25.

Dolag, K., Bartelmann, M., Perrotta, F., Baccigalupi, C., Moscardini, L., Meneghetti, M., Tormen, G., 2004. Numerical study of halo concentrations in dark-energy cosmologies. Astronomy & Astrophysics 416, 853–864. doi:10.1051/0004-6361:20031757.

Duffy, A.R., Schaye, J., Kay, S.T., Dalla Vecchia, C., 2008. Dark matter halo concentrations in the Wilkinson Microwave Anisotropy Probe year 5 cosmology. Monthly Notices of the Royal Astronomical Society: Letters 390, L64–L68. doi:10.1111/j.1745-3933.2008.00537.x.

Dutton, A.A., Macciò, A.V., 2014. Cold dark matter haloes in the Planck era: Evolution of structural parameters for Einasto and NFW profiles. Monthly Notices of the Royal Astronomical Society 441, 3359–3374. doi:10.1093/mnras/stu742.

Dvornik, A., Hoekstra, H., Kuijken, K., Schneider, P., Amon, A., Nakajima, R., Viola, M., Choi, A., Erben, T., Farrow, D.J., Heymans, C., Hildebrandt, H., Sifón, C., Wang, L., 2018. Unveiling galaxy bias via the halo model, KiDS, and GAMA. Monthly Notices of the Royal Astronomical Society 479, 1240–1259. doi:10.1093/mnras/sty1502.

Einasto, J., 1965. On the Construction of a Composite Model for the Galaxy and on the Determination of the System of Galactic Parameters. Trudy Astrofizicheskogo Instituta Alma-Ata 5, 87–100.

Eisenstein, D.J., Hu, W., 1998. Baryonic Features in the Matter Transfer Function. The Astrophysical Journal 496, 605–614. doi:10.1086/305424.

Eke, V.R., Navarro, J.F., Steinmetz, M., 2001. The Power Spectrum Dependence of Dark Matter Halo Concentrations. The Astrophysical Journal 554, 114–125. doi:10.1086/321345.

Frenk, C., White, S., 2012. Dark matter and cosmic structure. Annalen der Physik 524, 507–534. doi:10.1002/andp.201200212.

Garcia, R., Rozo, E., Becker, M.R., More, S., 2020. A Redefinition of the Halo Boundary Leads to a Simple yet Accurate Halo Model of Large Scale Structure. arXiv e-prints 2006, arXiv:2006.12751.

Geach, J.E., Sobral, D., Hickox, R.C., a. Wake, D., Smail, I., Best, P.N., Baugh, C.M., Stott, J.P., 2012. The clustering of Hα emitters at z =2.23 from HiZELS. Monthly Notices of the Royal Astronomical Society 426, 679–689. doi:10.1111/j.1365-2966.2012.21725.x.

Ginzburg, D., Desjacques, V., Chan, K.C., 2017. Shot noise and biased tracers: A new look at the halo model. Physical Review D 96, 083528. doi:10.1103/PhysRevD.96.083528.

Giocoli, C., Bartelmann, M., Sheth, R.K., Cacciato, M., 2010. Halo model description of the non-linear dark matter power spectrum at k$\gg$1 Mpc-1. Monthly Notices of the Royal Astronomical Society

408, 300–313. doi:10.1111/j.1365-2966.2010.17108.x.

Giocoli, C., Tormen, G., Sheth, R.K., 2012. Formation times, mass growth histories and concentrations of dark matter haloes. Monthly Notices of the Royal Astronomical Society 422, 185–198. doi:10.1111/j.1365-2966.2012.20594.x.

Hernquist, L., 1990. An analytical model for spherical galaxies and bulges. The Astrophysical Journal 356, 359–359. doi:10.1086/168845.

Hunter, J.D., 2007. Matplotlib: A 2D graphics environment. Computing in Science & Engineering 9, 90–95.

Ishikawa, S., Okumura, T., Oguri, M., Lin, S.C., 2021. Halo-model analysis of the clustering of photometric luminous red galaxies at $0.10 \leq z \leq 1.05$ from the Subaru Hyper Suprime-Cam Survey. arXiv e-prints 2103, arXiv:2103.08628.

Ishiyama, T., Enoki, M., Kobayashi, M.A.R., Makiya, R., Nagashima, M., Oogi, T., 2015. The ν 2 GC Simulations : Quantifying the Dark Side of the Universe in the Planck Cosmology. Publications of the Astronomical Society of Japan 67, 61–61.

Ishiyama, T., Prada, F., Klypin, A.A., Sinha, M., Metcalf, R.B., Jullo, E., Altieri, B., Cora, S.A., Croton, D., de la Torre, S., Millán-Calero, D.E., Oogi, T., Ruedas, J., Vega-Martínez, C.A., 2020. The Uchuu Simulations: Data Release 1 and Dark Matter Halo Concentrations. arXiv e-prints 2007, arXiv:2007.14720.

Jenkins, A., Frenk, C.S., White, S.D.M., Colberg, J.M., Cole, S., Evrard, A.E., Couchman, H.M.P., Yoshida, N., 2001. The mass function of dark matter haloes. Monthly Notices of the Royal Astronomical Society 321, 372–384. doi:10.1046/j.1365-8711.2001.04029.x.

Jing, Y.P., 1998. Accurate Fitting Formula for the Two-Point Correlation Function of Dark Matter Halos. The Astrophysical Journal 503, L9–L13. doi:10.1086/311530.

Jing, Y.P., 1999. Accurate determination of the Lagrangian bias for the dark matter halos. The Astrophysical Journal 515, L45–L48. doi:10.1086/311978.

Jing, Y.P., Suto, Y., 2002. Triaxial Modeling of Halo Density Profiles with High-Resolution N -Body Simulations. The Astrophysical Journal 574, 538–553. doi:10.1086/341065.

Kauffmann, G., Nusser, A., Steinmetz, M., 1997. Galaxy formation and large-scale bias. Monthly Notices of the Royal Astronomical Society 286, 795–811.

Kauffmann, G., White, S.D.M., Heckman, T.M., Ménard, B., Brinchmann, J., Charlot, S., Tremonti, C., Brinkmann, J., 2004. The environmental dependence of the relations between stellar mass, structure, star formation and nuclear activity in galaxies. Monthly Notices of the Royal Astronomical Society 353, 713–731. doi:10.1111/j.1365-2966.2004.08117.x.

Kim, H.S., Baugh, C.M., Benson, A.J., Cole, S., Frenk, C.S., Lacey, C.G., Power, C., Schneider, M., 2011. The spatial distribution of cold gas in hierarchical galaxy formation models. Monthly Notices of the Royal Astronomical Society 414, 2367–2385. doi:10.1111/j.1365-2966.2011.18556.x.

Klypin, A., Yepes, G., Gottlöber, S., Prada, F., Heß, S., 2016. Multi-Dark simulations: The story of dark matter halo concentrations and density profiles. Monthly Notices of the Royal Astronomical Society 457, 4340–4359. doi:10.1093/mnras/stw248.

Knebe, A., Knollmann, S.R., Muldrew, S.I., Pearce, F.R., Aragon-Calvo, M.A., Ascasibar, Y., Behroozi, P.S., Ceverino, D., Colombi, S., Diemand, J., Dolag, K., Falck, B.L., Fasel, P., Gardner, J., Gottlöber, S., Hsu, C.H., Iannuzzi, F., Klypin, A., Lukić, Z., Maciejewski, M., Mcbride, C., Neyrinck, M.C., Planelles, S., Potter, D., Quilis, V., Rasera, Y., Read, J.I., Ricker, P.M., Roy, F., Springel, V., Stadel, J., Stinson, G., Sutter, P.M., Turchaninov, V., Tweed, D., Yepes, G., Zemp, M., 2011. Haloes gone MAD: The Halo-Finder Comparison Project. Monthly Notices of the Royal Astronomical Society 415, 2293–2318. doi:10.1111/j.1365-2966.2011.18858.x.

Krause, E., Hirata, C.M., Martin, C., Neill, J.D., Wyder, T.K., 2012. Halo occupation distribution modelling of green valley galaxies. Monthly Notices of the Royal Astronomical Society 428, 2548–2564. doi:10.1093/mnras/sts221.

Lapi, A., Danese, L., 2014. Statistics of Dark Matter Halos in the Excursion Set Peak Framework. Journal of Cosmology and Astroparticle Physics 7, 44–44. doi:10.1088/1475-7516/2014/07/044.

Leauthaud, A., Tinker, J., Behroozi, P.S., Busha, M.T., Wechsler, R.H., 2011. A THEORETICAL FRAMEWORK FOR COMBINING TECHNIQUES THAT PROBE THE LINK BETWEEN GALAXIES AND DARK MATTER. The Astrophysical Journal 738, 45–45. doi:10.1088/0004-637X/738/1/45.

Leauthaud, A., Tinker, J., Bundy, K., Behroozi, P.S., Massey, R., Rhodes, J., George, M.R., Kneib, J.P., Benson, A.J., Wechsler, R.H., Busha, M.T., Capak, P., Cortês, M., Ilbert, O., Koekemoer, A.M., Le Fèvre, O., Lilly, S., McCracken, H.J., Salvato, M., Schrabback, T., Scoville, N., Smith, T., Taylor, J.E., 2012. NEW CONSTRAINTS ON THE EVOLUTION OF THE STELLAR-TO-DARK MATTER CONNECTION: A COMBINED ANALYSIS OF GALAXY-GALAXY LENSING, CLUSTERING, AND STELLAR MASS FUNCTIONS FROM z = 0.2 to z = 1. The Astrophysical Journal 744, 159–159. doi:10.1088/0004-637X/744/2/159.

Lesgourgues, J., 2011. The Cosmic Linear Anisotropy Solving System (CLASS) I: Overview. arXiv e-prints 1104, arXiv:1104.2932.

Lewis, A., Challinor, A., Lasenby, A., 2000. Efficient Computation of Cosmic Microwave Background Anisotropies in Closed Friedmann-Robertson-Walker Models. The Astrophysical Journal 538, 473–476. doi:10.1086/309179.

Liddle, A.R., Lyth, D.H., 2000. Cosmological Inflation and Large-Scale Structure. Cambridge University Press, Cambridge.

Limber, D.N., 1953. The Analysis of Counts of the Extragalactic Nebulae in Terms of a Fluctuating Density Field. The Astrophysical Journal 117, 134. doi:10.1086/145672.

Ludlow, A.D., Bose, S., Angulo, R.E., Wang, L., Hellwing, W.A., Navarro, J.F., Cole, S., Frenk, C.S., 2016. The Mass-Concentration-Redshift Relation of Cold and Warm Dark Matter Halos. Monthly Notices of the Royal Astronomical Society 000, 0–0. doi:10.1093/mnras/stw1046.

Ludlow, A.D., Navarro, J.F., Angulo, R.E., Boylan-kolchin, M., Springel, V., Frenk, C., White, S.D.M., 2014. The mass-concentration-redshift relation of cold dark matter haloes. Monthly Notices of the Royal Astronomical Society 441, 378–388. doi:10.1093/mnras/stu483.

Ludlow, A.D., Navarro, J.F., Boylan-Kolchin, M., Bett, P.E., Angulo, R.E., Li, M., White, S.D.M., Frenk, C., Springel, V., 2013. The Mass Profile and Accretion History of Cold Dark Matter Halos. Monthly Notices of the Royal Astronomical Society 432, 1103–1113. doi:10.1093/mnras/stt526.

Ma, C., Fry, J.N., 2000. Deriving the nonlinear cosmological power spectrum and bispectrum from analytic dark matter halo profiles and mass functions. The Astrophysical Journal 10, 503–513.

Ma, C., Maggiore, M., Riotto, A., Zhang, J., 2011. The bias and mass function of dark matter haloes in non-Markovian extension of the excursion set theory. Monthly Notices of the Royal Astronomical Society 411, 2644–2652. doi:10.1111/j.1365-2966.2010.17871.x.

Macciò, A.V., Dutton, A.A., van den Bosch, F.C., 2008. Concentration, spin and shape of dark matter haloes as a function of the cosmological model: WMAP 1, WMAP 3 and WMAP 5 results. Monthly Notices of the Royal Astronomical Society 391, 1940–1954. doi:10.1111/j.1365-2966.2008.14029.x.

Mandelbaum, R., Seljak, U., Kauffmann, G., Hirata, C.M., Brinkmann, J., 2006. Galaxy halo masses and satellite fractions from galaxy-galaxy lensing in the Sloan Digital Sky Survey: Stellar mass, luminosity, morphology and environment dependencies. Monthly Notices of the Royal Astronomical Society 368, 715–731. doi:10.1111/j.1365-2966.2006.10156.x.

Mandelbaum, R., Tasitsiomi, A., Seljak, U., Kravtsov, A.V., Wechsler, R.H., 2005. Galaxy-galaxy lensing: Dissipationless simulations versus the halo model. Monthly Notices of the Royal Astronomical Society 362, 1451–1462. doi:10.1111/j.1365-2966.2005.09417.x.

Manera, M., Sheth, R.K., Scoccimarro, R., 2010. Large-scale bias and the inaccuracy of the peak-background split. Monthly Notices of the Royal Astronomical Society 402, 589–602. doi:10.1111/j.1365-2966.2009.15921.x.

Mead, A.J., Verde, L., 2021. Including beyond-linear halo bias in

halo models. Monthly Notices of the Royal Astronomical Society 503, 3095–3111. doi:10.1093/mnras/stab748.

Miyatake, H., Kobayashi, Y., Takada, M., Nishimichi, T., Shirasaki, M., Sugiyama, S., Takahashi, R., Osato, K., More, S., Park, Y., 2020. Cosmological inference from emulator based halo model I: Validation tests with HSC and SDSS mock catalogs. arXiv e-prints 2101, arXiv:2101.00113.

Mo, H.J., White, S.D.M., 1996. An analytic model for the spatial clustering of dark matter haloes. Monthly Notices of the Royal Astronomical Society 282, 347–361. doi:10.1093/mnras/282.2.347.

Moore, B., Governato, F., Quinn, T., Stadel, J., Lake, G., 1998. Resolving the Structure of Cold Dark Matter Halos. The Astrophysical Journal 499, L5–L8. doi:10.1086/311333.

More, S., 2013. Cosmological Dependence of the Measurements of Luminosity Function, Projected Clustering and Galaxy-Galaxy Lensing Signal. The Astrophysical Journal 777, L26–L26. doi:10.1088/2041-8205/777/2/L26.

More, S., Kravtsov, A.V., Dalal, N., Gottlöber, S., 2011. The Overdensity and Masses of the Friends-of-friends Halos and Universality of Halo Mass Function. The Astrophysical Journal Supplement Series 195, 4. doi:10.1088/0067-0049/195/1/4.

More, S., Miyatake, H., Mandelbaum, R., Takada, M., Spergel, D.N., Brownstein, J.R., Schneider, D.P., 2015. The Weak Lensing Signal and the Clustering of Boss Galaxies. Ii. Astrophysical and Cosmological Constraints. The Astrophysical Journal 806, 2–2. doi:10.1088/0004-637X/806/1/2.

Moustakas, L.A., Somerville, R.S., 2002. The Masses, Ancestors and Descendents of Extremely Red Objects: Constraints from Spatial Clustering. The Astrophysical Journal 577, 1–10. doi:10.1086/342133.

Murray, S., Poulin, F., 2019. Hankel: A Python library for performing simple and accurate Hankel transformations. The Journal of Open Source Software 4, 1397. doi:10.21105/joss.01397.

Murray, S., Power C., Robotham, A., 2013a. How well do we know the halo mass function? Monthly Notices of the Royal Astronomical Society: Letters 434, L61–L65. doi:10.1093/mnrasl/slt079.

Murray, S.G., Power, C., Robotham, A.S.G., 2013b. HMFcalc: An online tool for calculating dark matter halo mass functions. Astronomy and Computing 3, 23–34. doi:10.1016/j.ascom.2013.11.001.

Navarro, J.F., Frenk, C.S., White, S.D.M., 1997. A Universal Density Profile from Hierarchical Clustering. The Astrophysical Journal 490, 493–508. doi:10.1086/304888.

Neyman, J., Scott, E.L., Shane, C.D., 1953. On the Spatial Distribution of Galaxies: A Specific Model. The Astrophysical Journal 117, 92–92. doi:10.1086/145671.

Nicola, A., Alonso, D., Sánchez, J., Slosar, A., Awan, H., Broussard, A., Dunkley, J., Gawiser, E., Gomes, Z., Mandelbaum, R., Miyatake, H., Newman, J.A., Sevilla-Noarbe, I., Skinner, S., Wagoner, E.L., 2020. Tomographic galaxy clustering with the Subaru Hyper Suprime-Cam first year public data release. Journal of Cosmology and Astroparticle Physics 03, 044. doi:10.1088/1475-7516/2020/03/044.

Nishimichi, T., Takada, M., Takahashi, R., Osato, K., Shirasaki, M., Oogi, T., Miyatake, H., Oguri, M., Murata, R., Kobayashi, Y., Yoshida, N., 2021. DarkEmulator: Cosmological emulation code for halo clustering statistics. Astrophysics Source Code Library, record ascl:2103.009 , ascl:2103.009.

Nusser, A., Tiwari, P., 2015. THE CLUSTERING OF RADIO GALAXIES: BIASING AND EVOLUTION VERSUS STELLAR MASS. The Astrophysical Journal 812, 85–85. doi:10.1088/0004-637X/812/1/85.

Ogata, H., 2005. A Numerical Integration Formula Based on the Bessel Functions. Publ. RIMS, Kyoto Univ. 41, 949–970.

Padmanabhan, H., Refregier, A., Amara, A., 2016. A halo model for cosmological neutral hydrogen : Abundances and clustering. eprint arXiv:1611.06235 arXiv:1611.06235.

Paranjape, A., Sefusatti, E., Chan, K.C., Desjacques, V., Monaco, P., Sheth, R.K., 2013a. Bias deconstructed: Unravelling the scale dependence of halo bias using real-space measurements. Monthly Notices of the Royal Astronomical Society 436, 449–459. doi:10.1093/mnras/stt1578.

Paranjape, A., Sheth, R.K., Desjacques, V., 2013b. Excursion set peaks: A self-consistent model of dark halo abundances and clustering. Monthly Notices of the Royal Astronomical Society 431, 1503–1512. doi:10.1093/mnras/stt267.

Peacock, J.A., 2007. Testing anthropic predictions for   and the cosmic microwave background temperature. Monthly Notices of the Royal Astronomical Society 379, 1067–1074. doi:10.1111/j.1365-2966.2007.11978.x.

Peacock, J.A., Smith, R.E., 2000. Halo occupation numbers and galaxy bias. Monthly Notices of the Royal Astronomical Society 318, 1144–1156. doi:10.1046/j.1365-8711.2000.03779.x.

Pillepich, A., Porciani, C., Hahn, O., 2010. Halo mass function and scale-dependent bias from N-body simulations with non-Gaussian initial conditions. Monthly Notices of the Royal Astronomical Society 402, 191–206. doi:10.1111/j.1365-2966.2009.15914.x.

Planck Collaboration, 2015. Planck 2015 reslts. XIII. Cosmological parameters. Astronomy & Astrophysics , 20–20.

Poole, G.B., Blake, C., Marin, F.A., Power, C., Mutch, S.J., Croton, D.J., Colless, M., Couch, W., Drinkwater, M.J., Glazebrook, K., 2015. The Gigaparsec WiggleZ simulations: Characterizing scale-dependant bias and associated systematics in growth of structure measurements. Monthly Notices of the Royal Astronomical Society 449, 1454–1469. doi:10.1093/mnras/stv314.

Prada, F., Klypin, A.A., Cuesta, A.J., Betancort-Rijo, J.E., Primack, J., 2012. Halo concentrations in the standard Λ cold dark matter cosmology. Monthly Notices of the Royal Astronomical Society 423, 3018–3030. doi:10.1111/j.1365-2966.2012.21007.x.

Press, W.H., Schechter, P., 1974. Formation of galaxies and clusters of galaxies by self-similar gravitational condensation. The Astrophysical Journal 187, 425–438.

Rafiei-Ravandi, M., Smith, K.M., Masui, K.W., 2020. Characterizing fast radio bursts through statistical cross-correlations. Physical Review D 102, 023528. doi:10.1103/PhysRevD.102.023528.

Reddick, R.M., Tinker, J.L., Wechsler, R.H., Lu, Y., 2014. COSMOLOGICAL CONSTRAINTS FROM GALAXY CLUSTERING AND THE MASS-TO-NUMBER RATIO OF GALAXY CLUSTERS: MARGINALIZING OVER THE PHYSICS OF GALAXY FORMATION. The Astrophysical Journal 783, 118–118. doi:10.1088/0004-637X/783/2/118.

Robitaille, T.P., Tollerud, E.J., Greenfield, P., Droettboom, M., Bray, E., Aldcroft, T., Davis, M., Ginsburg, A., Price-Whelan, A.M., Kerzendorf, W.E., Conley, A., Crighton, N., Barbary, K., Muna, D., Ferguson, H., Grollier, F., Parikh, M.M., Nair, P.H., Günther, H.M., Deil, C., Woillez, J., Conseil, S., Kramer, R., Turner, J.E.H., Singer, L., Fox, R., Weaver, B.A., Zabalza, V., Edwards, Z.I., Azalee Bostroem, K., Burke, D.J., Casey, A.R., Crawford, S.M., Dencheva, N., Ely, J., Jenness, T., Labrie, K., Lim, P.L., Pierfederici, F., Pontzen, A., Ptak, A., Refsdal, B., Servillat, M., Streicher, O., 2013. Astropy: A community Python package for astronomy. Astronomy & Astrophysics 558, A33–A33. doi:10.1051/0004-6361/201322068.

Rodríguez-Torres, S.A., Prada, F., Chuang, C.H., Guo, H., Klypin, A., Behroozi, P., Hahn, C.H., Comparat, J., Yepes, G., Montero-Dorta, A.D., Brownstein, J.R., Maraston, C., McBride, C.K., Tinker, J., Gottlöber, S., Favole, G., Shu, Y., Kitaura, F.S., Bolton, A., Scoccimarro, R., Samushia, L., Schlegel, D., Schneider, D.P., Thomas, D., 2015. The clustering of galaxies in the SDSS-III Baryon Oscillation Spectroscopic Survey: Modeling the clustering and halo occupation distribution of BOSS-CMASS galaxies in the Final Data Release. eprint arXiv:1509.06404 arXiv:1509.06404.

Schaan, E., White, M., 2021. Multi-tracer intensity mapping: Cross-correlations, Line noise & Decorrelation. arXiv e-prints 2103, arXiv:2103.01964.

Scherrer, R.J., Bertschinger, E., 1991. Statistics of primordial density perturbations from discrete seed masses. The Astrophysical Journal 381, 349–349. doi:10.1086/170658.

Schneider, A., 2015. Structure formation with suppressed small-scale perturbations. Monthly Notices of the Royal Astronomical Society 451, 3117–3130. doi:10.1093/mnras/stv1169.

Schneider, A., Giri, S.K., Mirocha, J., 2020. A halo model approach

40

for the 21-cm power spectrum at cosmic dawn. arXiv e-prints 2011, arXiv:2011.12308.

Schneider, A., Smith, R.E., Macciò, A.V., Moore, B., 2012. Non-linear evolution of cosmological structures in warm dark matter models. Monthly Notices of the Royal Astronomical Society 424, 684–698. doi:10.1111/j.1365-2966.2012.21252.x.

Schneider, A., Smith, R.E., Reed, D.S., 2013. Halo Mass Function and the Free Streaming Scale. Monthly Notices of the Royal Astronomical Society 433, 16–16. doi:10.1093/mnras/stt829.

Scoccimarro, R., Sheth, R.K., Hui, L., Jain, B., 2001. How many galaxies fit in a halo? Constraints on galaxy formation efficiency from spatial clustering. The Astrophysical Journal 546, 20–34.

Seljak, U., 2000. Analytic model for galaxy and dark matter clustering. Monthly Notices of the Royal Astronomical Society 213.

Seljak, U., Vlah, Z., 2015. Halo Zel'dovich model and perturbation theory: Dark matter power spectrum and correlation function. Physical Review D 91, 123516. doi:10.1103/PhysRevD.91.123516.

Seljak, U., Warren, M.S., 2004. Large-scale bias and stochasticity of haloes and dark matter. Monthly Notices of the Royal Astronomical Society 355, 129–136. doi:10.1111/j.1365-2966.2004.08297.x.

Shen, Y., McBride, C.K., White, M., Zheng, Z., Myers, A.D., Guo, H., Kirkpatrick, J.A., Padmanabhan, N., Parejko, J.K., Ross, N.P., Schlegel, D.J., Schneider, D.P., Streblyanska, A., Swanson, M.E.C., Zehavi, I., Pan, K., Bizyaev, D., Brewington, H., Ebelke, G., Malanushenko, V., Malanushenko, E., Oravetz, D., Simmons, A., Snedden, S., 2013. CROSS-CORRELATION OF SDSS DR7 QUASARS AND DR10 BOSS GALAXIES: THE WEAK LUMINOSITY DEPENDENCE OF QUASAR CLUSTERING AT z ∼ 0.5. The Astrophysical Journal 778, 98–98. doi:10.1088/0004-637X/778/2/98.

Sheth, R.K., Diaferio, A., Hui, L., Scoccimarro, R., 2001a. On the streaming motions of haloes and galaxies. Monthly Notices of the Royal Astronomical Society 326, 463–472. doi:10.1046/j.1365-8711.2001.04457.x.

Sheth, R.K., Hui, L., Diaferio, A., Scoccimarro, R., 2001b. Linear and non-linear contributions to pairwise peculiar velocities. Monthly Notices of the Royal Astronomical Society 325, 1288–1302. doi:10.1046/j.1365-8711.2001.04222.x.

Sheth, R.K., Tormen, G., 1999. Large scale bias and the peak background split. Monthly Notices of the Royal Astronomical Society 308, 119–126. doi:10.1046/j.1365-8711.1999.02692.x.

Simon, P., 2007. How accurate is Limber's equation? Astronomy & Astrophysics 473, 711–714. doi:10.1051/0004-6361:20066352.

a. Skibba, R., Coil, A.L., Mendez, A.J., Blanton, M.R., Bray, A.D., Cool, R.J., Eisenstein, D.J., Guo, H., Miyaji, T., Moustakas, J., Zhu, G., 2015. DARK MATTER HALO MODELS OF STELLAR MASS-DEPENDENT GALAXY CLUSTERING IN PRIMUS+DEEP2 AT 0.2 < z < 1.2. The Astrophysical Journal 807, 152–152. doi:10.1088/0004-637X/807/2/152.

a. Skibba, R., Sheth, R.K., 2009. A halo model of galaxy colours and clustering in the Sloan Digital Sky Survey. Monthly Notices of the Royal Astronomical Society 392, 1080–1091. doi:10.1111/j.1365-2966.2008.14007.x.

Smith, R.E., Desjacques, V., Marian, L., 2011. Nonlinear clustering in models with primordial non-Gaussianity: The halo model approach. Physical Review D 83, 043526–043526. doi:10.1103/PhysRevD.83.043526.

Smith, R.E., Markovič, K., 2011. Testing the warm dark matter paradigm with large-scale structures. Physical Review D 4, 1–18.

Smith, R.E., Peacock, J.A., Jenkins, A., White, S.D.M., Frenk, C.S., Pearce, F.R., Thomas, P.A., Efstathiou, G., Couchman, H.M.P., 2003. Stable clustering, the halo model and non-linear cosmological power spectra. Monthly Notices of the Royal Astronomical Society 341, 1311–1332. doi:10.1046/j.1365-8711.2003.06503.x.

Sugiyama, N., 1995. Cosmic Background Anisotropies in Cold Dark Matter Cosmology. The Astrophysical Journal Supplement Series 100, 281. doi:10.1086/192220.

Sunayama, T., Hearin, A.P., Padmanabhan, N., Leauthaud, A., 2015. The Scale-Dependence of Halo Assembly Bias. Arxiv e-prints .

Szapudi, I., Pan, J., Prunet, S., Budavári, T., 2005. Fast Edge Corrected Measurement of the Two-Point Correlation Function

and the Power Spectrum. Arxiv e-prints , 4–4doi:10.1086/496971.

Takahashi, R., Sato, M., Nishimichi, T., Taruya, A., Oguri, M., 2012. REVISING THE HALOFIT MODEL FOR THE NONLINEAR MATTER POWER SPECTRUM. The Astrophysical Journal 761, 152–152. doi:10.1088/0004-637X/761/2/152.

Taylor, J.E., 2011. Dark Matter Halos from the Inside Out. Advances in Astronomy 2011, 1–17. doi:10.1155/2011/604898.

Tinker, J.L., Kravtsov, A.V., Klypin, A., Abazajian, K.N., Warren, M.S., Yepes, G., Gottlöber, S., Holz, D.E., 2008. Toward a halo mass function for precision cosmology: The limits of universality. The Astrophysical Journal 688, 709–728.

Tinker, J.L., Robertson, B.E., Kravtsov, A.V., Klypin, A., Warren, M.S., Yepes, G., Gottlöber, S., 2010. The large-scale bias of dark matter halos: Numerical calibration and model tests. The Astrophysical Journal 724, 878–886. doi:10.1088/0004-637X/724/2/878.

Tinker, J.L., Weinberg, D., 2005. On the mass-to-light ratio of large-scale structure. The Astrophysical Journal 631, 41–58.

To, C., Krause, E., Rozo, E., Wu, H., Gruen, D., Wechsler, R.H., Eifler, T.F., Rykoff, E.S., Costanzi, M., Becker, M.R., Bernstein, G.M., Blazek, J., Bocquet, S., Bridle, S.L., Cawthon, R., Choi, A., Crocce, M., Davis, C., DeRose, J., Drlica-Wagner, A., Elvin-Poole, J., Fang, X., Farahi, A., Friedrich, O., Gatti, M., Gaztanaga, E., Giannantonio, T., Hartley, W.G., Hoyle, B., Jarvis, M., MacCrann, N., McClintock, T., Miranda, V., Pereira, M.E.S., Park, Y., Porredon, A., Prat, J., Rau, M.M., Ross, A.J., Samuroff, S., Sánchez, C., Sevilla-Noarbe, I., Sheldon, E., Troxel, M.A., Varga, T.N., Vielzeuf, P., Zhang, Y., Zuntz, J., Abbott, T.M.C., Aguena, M., Annis, J., Avila, S., Bertin, E., Bhargava, S., Brooks, D., Burke, D.L., Carnero Rosell, A., Carrasco Kind, M., Carretero, J., Chang, C., Conselice, C., da Costa, L.N., Davis, T.M., Desai, S., Diehl, H.T., Dietrich, J.P., Everett, S., Evrard, A.E., Ferrero, I., Flaugher, B., Fosalba, P., Frieman, J., García-Bellido, J., Gruendl, R.A., Gutierrez, G., Hinton, S.R., Hollowood, D.L., Huterer, D., James, D.J., Jeltema, T., Kron, R., Kuehn, K., Kuropatkin, N., Lima, M., Maia, M.A.G., Marshall, J.L., Menanteau, F., Miquel, R., Morgan, R., Muir, J., Myles, J., Palmese, A., Paz-Chinchón, F., Plazas, A.A., Romer, A.K., Roodman, A., Sanchez, E., Santiago, B., Scarpine, V., Serrano, S., Smith, M., Suchyta, E., Swanson, M.E.C., Tarle, G., Thomas, D., Tucker, D.L., Weller, J., Wester, W., 2020. Dark Energy Survey Year 1 Results: Cosmological Constraints from Cluster Abundances, Weak Lensing, and Galaxy Correlations. arXiv e-prints 2010, arXiv:2010.01138.

To, C.H., Krause, E., Rozo, E., Wu, H.Y., Gruen, D., DeRose, J., Rykoff, E., Wechsler, R.H., Becker, M., Costanzi, M., Eifler, T., da Silva Pereira, M.E., Kokron, N., DES Collaboration, 2021. Combination of cluster number counts and two-point correlations: Validation on mock Dark Energy Survey. Monthly Notices of the Royal Astronomical Society 502, 4093–4111. doi:10.1093/mnras/stab239.

Umeh, O., Maartens, R., Padmanabhan, H., Camera, S., 2021. The effect of finite halo size on the clustering of neutral hydrogen. arXiv e-prints 2102, arXiv:2102.06116.

van der Walt, S., Colbert, S.C., Varoquaux, G., 2011. The NumPy Array: A Structure for Efficient Numerical Computation. Computing in Science & Engineering 13, 22–30. doi:10.1109/MCSE.2011.37.

Viel, M., Lesgourgues, J., Haehnelt, M.G., Matarrese, S., Riotto, A., 2005. Constraining warm dark matter candidates including sterile neutrinos and light gravitinos with WMAP and the Lyman-α forest. Physical Review D 71, 1–10. doi:10.1103/PhysRevD.71.063534.

Virtanen, P., Gommers, R., Oliphant, T.E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S.J., Brett, M., Wilson, J., Millman, K.J., Mayorov, N., Nelson, A.R.J., Jones, E., Kern, R., Larson, E., Carey, C.J., Polat, İ., Feng, Y., Moore, E.W., VanderPlas, J., Laxalde, D., Perktold, J., Cimrman, R., Henriksen, I., Quintero, E.A., Harris, C.R., Archibald, A.M., Ribeiro, A.H., Pedregosa, F., van Mulbregt, P., 2020. SciPy 1.0: Fundamental algorithms for scientific computing in Python. Nature Methods 17, 261–272. doi:10.1038/s41592-019-0686-2.

Wake, D.A., Croom, S.M., Sadler, E.M., Johnston, H.M., 2008. The

clustering of radio galaxies at z = 0.55 from the 2SLAQ LRG survey. Monthly Notices of the Royal Astronomical Society 391, 1674–1684. doi:10.1111/j.1365-2966.2008.14039.x.

Wang, K., Mao, Y.Y., Zentner, A.R., Lange, J.U., van den Bosch, F.C., Wechsler, R.H., 2020. Concentrations of Dark Haloes Emerge from Their Merger Histories. arXiv e-prints 2004, arXiv:2004.13732.

Watson, W.A., Iliev, I.T., D'Aloisio, A., Knebe, A., Shapiro, P.R., Yepes, G., 2013. The halo mass function through the cosmic ages. Monthly Notices of the Royal Astronomical Society 433, 1230–1245. doi:10.1093/mnras/stt791.

Wechsler, R.H., Bullock, J.S., Primack, J.R., Kravtsov, A.V., Dekel, A., 2002. Concentrations of Dark Halos from Their Assembly Histories. The Astrophysical Journal 568, 52–70. doi:10.1086/338765.

White, M., Hernquist, L., Springel, V., 2001. The Halo Model and Numerical Simulations. The Astrophysical Journal 550, L129–L132. doi:10.1086/319644.

Wolz, L., Murray, S.G., Blake, C., Wyithe, J.S., 2019. Intensity mapping cross-correlations II: HI halo models including shot noise. Monthly Notices of the Royal Astronomical Society 484, 1007. doi:10.1093/mnras/sty3142.

Zaldarriaga, M., Seljak, U., 2000. CMBFAST for Spatially Closed Universes. The Astrophysical Journal 129, 431–434.

Zehavi, I., Zheng, Z., Weinberg, D.H., Blanton, M.R., a. Bahcall, N., a. Berlind, A., Brinkmann, J., a. Frieman, J., Gunn, J.E., Lupton, R.H., Nichol, R.C., Percival, W.J., Schneider, D.P., a. Skibba, R., a. Strauss, M., Tegmark, M., York, D.G., 2011. Galaxy Clustering in the Completed Sdss Redshift Survey: The Dependence on Color and Luminosity. The Astrophysical Journal 736, 59–59. doi:10.1088/0004-637X/736/1/59.

Zehavi, I., Zheng, Z., Weinberg, D.H., Frieman, J.A., Berlind, A.A., Blanton, M.R., Scoccimarro, R., Sheth, R.K., Strauss, M.A., Kayo, I., Suto, Y., Fukugita, M., Nakamura, O., Bahcall, N.A., Brinkmann, J., Gunn, J.E., Hennessy, G.S., Ivezić, Ž., Knapp, G.R., Loveday, J., Meiksin, A., Schlegel, D.J., Schneider, D.P., Szapudi, I., Tegmark, M., Vogeley, M.S., York, D.G., 2005. The Luminosity and Color Dependence of the Galaxy Correlation Function. The Astrophysical Journal 630, 1–27. doi:10.1086/431891.

Zemp, M., Gnedin, O.Y., Gnedin, N.Y., Kravtsov, A.V., 2011. On Determining The Shape Of Matter Distributions. The Astrophysical Journal Supplement Series 197, 30–30. doi:10.1088/0067-0049/197/2/30.

Zentner, A.R., 2007. The Excursion Set Theory of Halo Mass Functions, Halo Clustering, and Halo Growth. International Journal of Modern Physics D 16, 763–815. doi:10.1142/S0218271807010511.

Zhao, D.H., Jing, Y.P., Mo, H.J., Börner, G., 2009. ACCURATE UNIVERSAL MODELS FOR THE MASS ACCRETION HISTORIES AND CONCENTRATIONS OF DARK MATTER HALOS. The Astrophysical Journal 707, 354–369. doi:10.1088/0004-637X/707/1/354.

Zhao, D.H., Mo, H.J., Jing, Y.P., Borner, G., 2003. The growth and structure of dark matter haloes. Monthly Notices of the Royal Astronomical Society 339, 12–24. doi:10.1046/j.1365-8711.2003.06135.x.

Zheng, Z., 2004. Interpreting the Observed Clustering of Red Galaxies at z~3. The Astrophysical Journal 610, 61–68. doi:10.1086/421542.

Zheng, Z., Berlind, A.A., Weinberg, D.H., Benson, A.J., Baugh, C.M., Frenk, C.S., Katz, N., Lacey, C.G., Cole, S., Dave, R., 2005. Theoretical Models of the Halo Occupation Distribution: Separating Central and Satellite Galaxies.

Zheng, Z., Guo, H., 2015. Accurate and Efficient Halo-based Galaxy Clustering Modelling with Simulations. Arxiv e-prints .

Zhou, R., Newman, J.A., Mao, Y.Y., Meisner, A., Moustakas, J., Myers, A.D., Prakash, A., Zentner, A.R., Brooks, D., Duan, Y., Landriau, M., Levi, M.E., Prada, F., Tarle, G., 2021. The clustering of DESI-like luminous red galaxies using photometric redshifts. Monthly Notices of the Royal Astronomical Society 501, 3309–3331. doi:10.1093/mnras/staa3764.

Appendix A. halomod's Caching System

The HMF package has always included some form of caching, whereby derived quantities are stored for later retrieval, and only updated when absolutely necessary. At the time of its initial publication, HMF relied on each parameter being hard-coded into an `update` method in a specific way so as to trigger re-calculation of all child properties. This required great care when modifying the code, and especially when adding new parameters, as breaking the chain of dependencies could cause subtle errors.

We have since updated the caching system of HMF to be much more robust. We use ~~Python~~PYTHON's function decorators, a kind of functional programming technology, to abstract the caching logic from the main `Framework`. There is now a specific module containing two methods and a single base class which contain all the machinery for the cache system. This increases the readability of the main code, and maximises the usage of the DRY (Don't Repeat Yourself) principle.

The caching system makes heavy use of the general distinction between a `parameter` and `cached_quantity`. All parameters receive a `@parameter` decorator, which uses function closure to modify its behaviour. Likewise, all quantities receive a `@cached_quantity` decorator. Every time a `parameter` or `cached_quantity` is accessed, though it may seem that a simple attribute is being returned, an entire wrapping function is called (though it is built to add very little overhead).

The `@cached_property` decorator performs several operations each time its function is called. Firstly, it is useful to understand that four private tables are stored on the `Framework`. There are two dictionaries which map parameter names to quantities they affect and vice versa (call these `par_prop` and `prop_par` respectively), a dynamic list of quantities that are *currently being evaluated* called `activeq` (there can be more than one because quantities can be nested), and a simple dictionary mapping quantities to a boolean value which specifies whether it needs to be recalculated, called `recalc`.

The first thing required when accessing a quantity is to update any other active quantities' `prop_par` dictionaries with its own parameter dependencies. This will only do anything if this particular quantity has already been calculated and cached (and thus we know its parameter dependencies). Following this, if the quantity is in `recalc` and it doesn't need to be recalculated, just return the cached value. Otherwise, if it's in `recalc` and *does* need to be recalculated, call the the evaluation function, cache it, switch `recalc` to false for this parameter, and return. Finally, if it's not in `recalc` at all, this must be the first time we've ever tried to access this quantity. In this case, we first add it to the `activeq`, so that any parameters used while trying to evaluate this quantity will be added to its dependents (in `prop_par`). We then evaluate the function and cache it, then invert the (now full) `prop_par` dictionary to obtain `par_prop`. We then remove it from

the `activeq` and set its value in `recalc` to false so it isn't recalculated next time, and then return the value.

The missing ingredient to this is how accessing the `parameters` adds itself to the `prop_par` of all quantities in `activeq`. This is of course achieved by all `parameters` also being ~~Python~~ PYTHON descriptors (i.e. decorated), and it is simple for the `parameter` to iterate through the `activeq` list and add itself to each quantity's `prop_par` entry.

This method is as efficient as possible for such a generalised process. No re-calculations are performed unless necessary. However, higher efficiency could be achieved by simply using low-level functions chained together in an appropriate way, storing only the variables that needed storing for a particular application. This is due to the extra memory usage required to store all variables in the caching system, and the minimal overhead of re-checking the cache.

Importantly, this process is completely transparent to the user, who merely accesses quantities as if they were pre-stored variables. Furthermore, it is simple for the developer, who merely needs to attach a `@parameter` or `@cached_quantity` to any new parameters or quantities they define.

Appendix B. Hankel Transform

The transformation from 3D power spectrum to real-space correlation function is given by ~~Eq. 5~~ Eq. 5, and is a zeroth-order spherical Hankel transformation. It is necessary to perform this transformation to calculate the two-halo correlations, and also for the one-halo term if an analytic form for the self-convolution of the density profile does not exist.

The transformation poses a challenge (eg. Diemer, 2018), as it is highly oscillatory in log-space, in which it is necessary to perform the integration.

`halomod` largely avoids this issue by using a technique outlined in Szapudi et al. (2005), developed in Ogata (2005), and implemented in Murray and Poulin (2019). The technique uses a quadrature rule based on the roots of the Bessel function and a double-exponential transformation.

The general rule is defined as

$$\int_0^\infty f(x) J_\nu(x)\,dx \approx \pi \sum_{k=1}^\infty w_{\nu k} f(y_{\nu k}) J_\nu(y_{\nu k}) \psi'(h r_{\nu k}),$$
(B.1)

where h is an integration step-size (similar to the trapezoidal rule) and

$$y_{\nu k} = \pi \psi(h r_{\nu k})/h \quad \text{(B.2)}$$
$$\psi(t) = t \tanh(\pi \sinh(t)/2) \quad \text{(B.3)}$$
$$\psi'(t) = \frac{\pi t \cosh(t) + \sinh(\pi \sinh(t))}{1 + \cosh(\pi \sinh(t))} \quad \text{(B.4)}$$
$$w_{\nu k} = \frac{Y_\nu(\pi r_{\nu k})}{J_{\nu+1}(\pi r_{\nu k})}. \quad \text{(B.5)}$$

Here $J_\nu(x)$ represents a Bessel function of the first kind of order ν, and $Y_\nu(x)$ a Bessel function of the second kind.

Lastly, $r_{\nu k}$ are the roots of the Bessel function J_ν, divided by π (alternatively, the roots of $J_\nu(\pi x)$).

In the case of the zeroth-order spherical Hankel transform, we may make some simplifications. We firstly take ~~Eq. 5~~ Eq. 5 and make the substitution $z = kr$:

$$\xi(r) = \frac{1}{2\pi^2 r^3} \int_0^\infty P(z/r) z^2 j_0(z)\,dz,$$
(B.6)

where j_0 is the spherical Bessel function of order zero. We also note that the spherical Bessel functions are related to the standard functions by

$$j_\nu(x) = \sqrt{\frac{\pi}{2x}} J_{\nu+1/2}(x),$$
(B.7)

so we may update ~~Eq. B.6 to~~ Eq. B.6 to

$$\xi(r) = \frac{1}{2\pi^2 r^3} \int_0^\infty P(z/r) z^2 \sqrt{\frac{\pi}{2z}} J_{1/2}(z)\,dz.$$
(B.8)

~~Eq. B.8~~ Eq. B.8 is in the form required to implement the method of Ogata (2005), with $\nu = 1/2$. In this case, the zeros of the function, $r_{\nu k} = 1, 2, 3....$

We may make one further optimization by noticing that

$$w_{1/2k} = \frac{y_0(x_k)}{j_1(x_k)} = \frac{-\cos(x_k)/x_k}{\sin(x_k)/x_k^2 - \cos(x_k)/x_k} = 1, \quad \text{(B.9)}$$

where the last equality holds because $x_k = \pi r_{1/2,k}$ are defined as the roots of $\sin(x)$. Thus we finally arrive at the equation

$$\xi(r) \approx \pi \sum_{k=1}^\infty f(y_{1/2,k}) J_{1/2}(y_{1/2,k}) \psi'(h r_{1/2,k}),$$
(B.10)

with

$$f(x) = P(x/r) x^2 \sqrt{\frac{\pi}{2x}}.$$
(B.11)

In practice, the infinite sum is truncated at some finite value N. The benefit of this method, due to the double-exponential transformation, is that this N can be quite small for a very accurate result. Since the values, $y_{\nu k}$, at which the sum is evaluated approach the zeros of the bessel function as $k \to 0$, Murray and Poulin (2019) found that a maximum of $N = 3.2/h$ steps are required (for reasonably small h).

The required value of h is mostly dictated by the existence of information in the integrand at low x. As expected, a smaller value of h will in general give more accurate results (modulo accumulated numerical addition error). The requisite value of h will be different when transforming different functions.

To avoid potential memory issues, we sum ~~Eq. B.10~~ Eq. B.10 in chunks of 100 terms, checking for convergence after each chunk. We find in general that using $h = 0.005$ for the $P(k) \to \xi(r)$ transformation is sufficient.

A consistent way to test the accuracy of the transform is to perform a full round-trip transform from $P(k) \to \xi(r) \to P(k)$ and check whether the result is close to the input. Since the transform uses values of $f(x)$ for x as small as $\sim 10h$ and as large as $\sim 10/h$, corresponding to k values between $(10h/r_{\max}, 10/(hr_{\min})$ (and similarly for r), the vectorized $P(k_i)$ and $\xi(r_i)$ need to be extrapolated – a procedure that must be done carefully, but will still yield some error for points outside the original domain. Thus the accuracy of the round-trip transform fundamentally depends on the choice of r-vector: $r_{\min}$, $r_{\max}$ and $d \log r$, as well as h_k and h_r (the resolution of the integration in the forward and backwards transforms). We find that for the linear power spectrum, we achieve an error of less than 1% over scales $k \in (10^{-2}, 10^4)$ using the parameters $r_{\min} = 10^{-3}$, $r_{\max} = 10^{3.5}$, $d \log r = 0.01$, $h_k = 10^{-4}$ and $h_r = 10^{-3}$.

For scales $k < 10^{-2}$, the 2-halo term may be very well approximated by the linear power multiplied by the effective bias (~~Eq. 66~~ Eq. 66), and we switch to that approximation on these scales (otherwise $r_{\max}$ would need to grow significantly, reducing performance).

Appendix C. Projected Correlation Function Limits

The finite upper limit can be handled empirically, keeping in mind that the correlation function may become negative at large scales (to a small degree). To perform this convergence test, we cumulatively integrate the real-space correlation function at several scales r_p, determining where the contribution to the integral becomes less than a percent. While there is no continuous relation which describes this location as a function of r_p, we deduce the following stepwise prescription:

$$r_{\max} = \max(80.5, 5r_p). \tag{C.1}$$

This prescription captures the idea that for small r_p, we do not want to integrate out to negative correlation, while for larger r_p, we must integrate well past the point at which the correlation becomes negative. In practice, measurements of the projected correlation function usually set a finite upper limit of $\sim 60h^{-1}\mathrm{Mpc}$, and we allow the user to set this arbitrarily if desired.

The finite lower limit has more difficult convergence properties. We make the substitution $y = r - r_p$ in ~~Eq. 71~~ Eq. 71 and multiply by y to estimate the contribution of each (logarithmic) scale to the integral. We then proceed to perform an approximate analytic analysis of the convergence by assuming that as $y \to 0$, the correlation can be expressed as a power law $\xi(r) \approx r^{-a}$. This yields the following expression for the contribution of each scale r:

$$C(y, a, r_p) = \frac{y(y + r_p)^{(1-a)}}{\sqrt{y(y + 2r_p)}}. \tag{C.2}$$

This equation shows a characteristic shape, with a peak around the scale r_p. This indicates that the integral limit must be less than the scale of this peak. In fact, we deduce the scale at which the contribution is 1% of the value at the peak, to ensure sub-percent accuracy.

To do this, we solve ~~$dC(y, a, r_p)/dy = 0$~~ $dC(y, a, r_p)/dy = 0$ for $y = r_{\mathrm{peak}}$, and calculate $C(r_{\mathrm{peak}}, a, r_p) = C_{\max}$. The next step is to solve $C(y, a, r_p) = f_{\mathrm{peak}}C_{\max}$, where f_{peak} is the fraction of the peak value we wish to locate (we set this to 0.01 in our implementation). However this is not analytically solvable. We make the valid assumption that at the solution, $y \ll r_p$ (since the peak is around r_p). This enables us to solve the simple equation

$$r_p^{1-a}\sqrt{y_{\mathrm{cut}}/(2r_p)} = f_{\mathrm{peak}}C_{\max}, \tag{C.3}$$

which has the solution

$$y_{\mathrm{cut}} = \theta(a)r_p f_{\mathrm{peak}}^2, \tag{C.4}$$

where

$$\theta(a) = \frac{-2a^3 + (p+9)a^2 - (3p+13)a + 3p + 7}{2(a-1)^2[p + 22(a-1)]\left(\frac{p+1}{a-1}\right)^{2a}} \tag{C.5}$$

$$p = \sqrt{4a^2 - 8a + 5} \tag{C.6}$$

$\theta(a)$ is a weakly but monotonically decreasing function, of the order ≈ 0.1 in the range of a we are interested in. In `halomod`, y_{cut} is calculated for each r_p, ensuring accuracy at any scale.