LSST DESC Notes

Core Cosmology Library: Precision Cosmological Predictions for LSST

Husni Almoubayyed,[1] David Alonso,[2] Jonathan Blazek,[3,4] Philip Bull,[5,6] Jean-Éric Campagne,[7] N. Elisa Chisari,[2] Alex Drlica-Wagner,[8] Zilong Du,[9] Tim Eifler,[10,11] John Ellison,[9] Renée Hlozek,[12] Mustapha Ishak,[13] Matthew Kirby,[14] David Kirkby,[15] Elisabeth Krause,[16] C. Danielle Leonard,[1] Christiane S. Lorenz,[2] Phil Marshall,[17] Thomas McClintock,[14] Sean McLaughlin,[18] Jérémy Neveu,[7] Stéphane Plaszczynski,[7] Javier Sanchez,[15] Sukhdeep Singh,[1,19] Anže Slosar,[20] Antonio Villarreal,[21] Michal Vrastil,[22] and Joe Zuntz[23]

(LSST Dark Energy Science Collaboration)

[1] *McWilliams Center for Cosmology, Department of Physics, Carnegie Mellon University, Pittsburgh, PA 15213, USA*
[2] *Department of Physics, University of Oxford, Denys Wilkinson building, Keble Road, Oxford OX1 3RH, United Kingdom*
[3] *Center for Cosmology and Astroparticle Physics, Ohio State, Columbus, OH 43210, USA*
[4] *Laboratory of Astrophysics, École Polytechnique Fédérale de Lausanne (EPFL), Observatoire de Sauverny, 1290 Versoix, Switzerland*
[5] *Department of Astronomy, University of California Berkeley, Berkeley, CA 94720, USA*
[6] *Radio Astronomy Laboratory, University of California Berkeley, Berkeley, CA 94720, USA*
[7] *Laboratoire de l'Accélérateur Linéaire, Université Paris-Sud, CNRS/IN2P3, Université Paris-Saclay, Orsay, France*
[8] *Fermi National Accelerator Laboratory, P. O. Box 500, Batavia, IL 60510, USA*
[9] *Department of Physics and Astronomy, University of California, Riverside, CA 92521, USA*
[10] *Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA 91109, USA*
[11] *Department of Physics, California Institute of Technology, Pasadena, CA 91125, USA*
[12] *Dunlap Institute for Astronomy and Astrophysics & Department for Astronomy and Astrophysics, University of Toronto, ON M5S 3H4*
[13] *Department of Physics, The University of Texas at Dallas, Richardson, TX 75083, USA*
[14] *Univeristy of Arizona, Tucson, AZ 85721, USA*
[15] *Department of Physics and Astronomy, University of California, Irvine, CA 92697, USA*
[16] *Kavli Institute for Particle Astrophysics and Cosmology, Stanford, CA 94305-4085, USA*
[17] *SLAC National Accelerator Laboratory, Menlo Park, CA 94025, USA*
[18] *Stanford University, Stanford, CA, 94305, USA*
[19] *Berkeley Center for Cosmological Physics and Department of Physics, University of California, Berkeley, California*
[20] *Brookhaven National Laboratory, Physics Department, Upton, NY 11973, USA*
[21] *Department of Physics and Astronomy, University of Pittsburgh, Pittsburgh PA 15260*
[22] *Institute of Physics CAS, Prague, 182 21, CZ*
[23] *Institute for Astronomy, Royal Observatory Edinburgh, Edinburgh EH9 3HJ, UK*

The Core Cosmology Library (CCL) provides routines to compute basic cosmological observables with validated numerical accuracy. These routines have been validated to an accuracy level, documented here, against the results of the Code Comparison Project. In the current version, predictions are provided for distances and background quantities, angular auto- and cross-spectra of cosmic shear and clustering, and the halo mass function. Fiducial specifications for the expected LSST galaxy distributions and clustering bias are also included, together with the capability of computing redshift distributions for a user-defined photometric redshift model. CCL is written in C, with a Python interface.

Contents

1. Introduction

Volunteer(s) in charge: Mustapha Ishak

Tentative: An era of precision cosmology is at the door with the advent of a number of high precision surveys. One of them is the Large Synoptic Survey Telescope (LSST) that is designed to address the question of cosmic acceleration and the Dark Energy associated with it.

In preparation for constraining cosmology with LSST, it is necessary to be able to produce rigorous (and accurate?) theoretical predictions for the cosmological quantities that will be measured. The Core Cosmology Library[1] (CCL) aims to provide, in one library, a way of making predictions that are validated to a well-documented numerical accuracy, for the purpose of constraining cosmology with LSST. By constructing a cosmology library specifically with LSST and its Dark Energy Science Collaboration (DESC) in mind, it is possible to ensure that it is flexible, adaptable, and validated for all cases of interest, as well as user-friendly and appropriate for the needs of all DESC working groups.

CCL computes basic cosmological functions including Hubble parameter, distances, density parameters, and linear growth functions. It calculates the matter power spectrum using various method including common approximations, calls to external software such as CLASS (Blas et al. 2011), or emulators such as "Cosmic Emulator" of Lawrence et al. (2017). It computes 2-point angular power spectra and correlation functions from various probes, going beyond the Limber approximation. CCL's overall structure is illustrated in figure **??**.

CCL's implementation includes spatially flat and curved Λ-Cold Dark Matter (ΛCDM) cosmologies, wCDM cosmologies with the option of using a time-dependent equation of state. It also allows the use of massive neutrinos. (is Table 1 up to date?).

Finally, it is worth noting that while CCL is built in light of LSST, its flexible and modular structure is designed to work with any other pipelines and surveys.

To do: Summarize the probes we cover. Explain future link to likelihood pipeline. Highlight CCL can be used by other surveys.

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

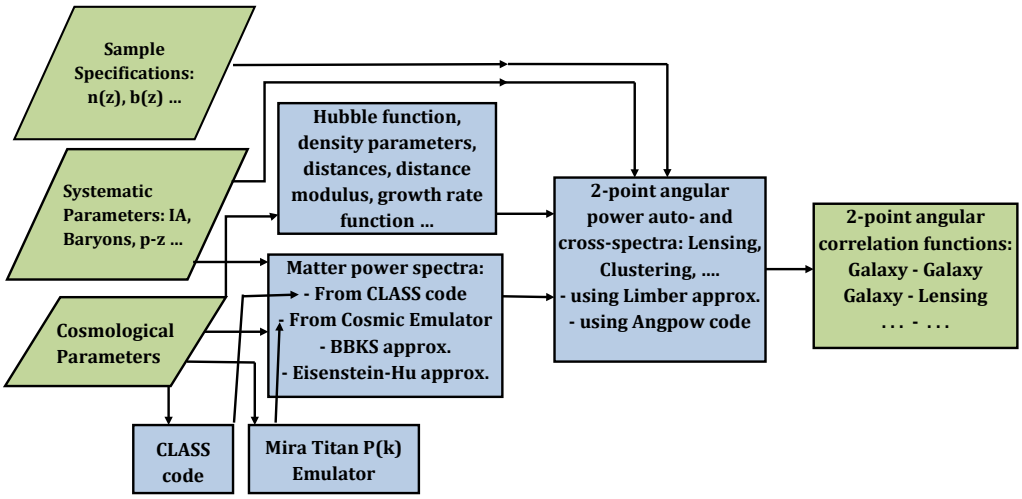

Figure 1. CCL structure flowchart. CCL is written in C with a Python interface. CCL routines calculate basic cosmological functions such as Hubble function, density parameters, distances and growth function. It uses various methods to compute the matter-power spectrum including CLASS, "Cosmic Emulator" developed by Lawrence et al. (2017), or common approximations. It computes 2-point angular power spectra and correlation functions from various probes. CCL is designed to accommodate multiple methods to calculate cosmological observables.

2. Cosmological models and observables

Volunteer(s) in charge: Renee

The overarching goal of CCL is to allow seamless integration of different cosmological models of interest to LSST, given some priors on the range of validity of the specific models.

The cosmological components include the matter density parameter Ω_m, the dark energy density Ω_Λ, the radiation density Ω_r, the energy in curvature (the curvature density parameter (MI)) Ω_K, the neutrino density of both massless and massive neutrinos, given by $\Omega_{\nu,\mathrm{rel}}$ and $\Omega_{\nu,\mathrm{m}}$ respectively, the optical depth τ, and parameters for the equation of state of dark energy.

Currently, the following families of models are supported:

- Flat ΛCDM cosmology with parameters $\Omega_b, n_s, A_s, \Omega_m, \Omega_\Lambda, \tau$ and a cosmological constant dark energy model with equation-of-state $w = -1$.

- wCDM and the Chevallier-Polarski-Linder (CPL) model for dark energy and it's redshift dependence ($w(a) = w_0 + w_a(1-a)$, Chevallier & Polarski 2001 and Linder 2003)

Table 1. Cosmologies implemented in CCL.

Observable/Model	flat ΛCDM	ΛCDM+K	ΛCDM + m_ν	wCDM	$w_0 + w_a$	MG
Distances	✓	✓	✓	✓	✓	X
Growth	✓	✓	X	✓	✓	✓
$P_m(k,z)$	✓	✓	✓	✓	✓	X
Halo Mass Function	✓	✓	X	✓	✓	X
C_l, number counts	✓	✓	X	✓	✓	X
C_l, weak lensing only	✓	✓	✓	✓	✓	X
Correlation function	✓	✓	✓	✓	✓	X

- A universe with non-zero curvature (K) so that the curvature energy density (density parameter (MI)) is the difference of sum of the energy densities of the other components with respect to unity (flatness), i.e. $\Omega_k = 1 - \sum_i \Omega_i$
- All of the above, plus an arbitrary, user-defined modified growth function (see description in Section 2.2)
- A single massive neutrino species or multiple equal-mass massive neutrinos, in combination with any of the above (except the user-defined modified growth function), specified by the neutrino mass Σm_ν (which maps on to the density $\Omega_{\nu,\mathrm{m}}$ above).

Not all features of `CCL` are available for all models. For a guide to which predictions are available for each model, see Table 1. Note that if users install their own version of `CLASS`, `CCL` can then make predictions for a more extended set of cosmologies. Users should take care to understand the validity of the `CCL` assumptions for their own models.

The power spectrum computed with `CLASS` is consistent with the power spectrum in Astropy to a precision of 10^{-4}, while the luminosity distance splines are accurate to 10^{-4} out to a redshift of $z = 2$. Similarly the co-moving distance comparisons yield similar results.

2.1. **Background cosmology**

Volunteer(s) in charge: Renee

Suggested content: Background cosmology parameter definition. Expansion and distance functions. Growth function and growth rate. The models that are specified

above map directly onto cosmological observables like the expansion rate of the universe, which is paramterised through the Hubble parameter (and assuming the CPL parameterization described above) as:

$$\frac{H(a)}{H_0} = a^{-3/2}\Big(\Omega_{M,0} + \Omega_{\Lambda,0}a^{-3(w_0+w_a)}\exp[3w_a(a-1)] + \Omega_{K,0}a + (\Omega_{g,0} + \Omega_{\nu,\mathrm{rel}})a^{-1} + \Omega_{\nu,\mathrm{m}}(a)a^3\Big)^{\frac{1}{2}}. \tag{1}$$

In order to compute distances, we use the radial comoving distance, which is calculated via a numerical integral as

$$\chi(a) = c\int_a^1 \frac{da'}{a'^2 H(a')}. \tag{2}$$

The transverse comoving distance is then computed in terms of the radial comoving distance as:

$$r(\chi) = \begin{cases} k^{-1/2}\sin(k^{1/2}\chi) & k > 0 \\ \chi & k = 0 \\ |k|^{-1/2}\sinh(|k|^{1/2}\chi) & k < 0 \end{cases} \tag{3}$$

This function can be written directly in terms of the energy density in cosmic curvature, Ω_k, as:

$$r(\chi) = \frac{c}{H_0}\sqrt{\Omega_k}\sinh(\sqrt{\Omega_k}\chi), \tag{4}$$

which is valid for all forms of the curvature energy density.

The usual angular diameter distance is $d_A = a\,r(a)$, and the luminosity distance is $d_L = r(a)/a$, leading to the familiar relation $d_A = a^2 d_L$ which is valid in general for all metric theories of gravity.

The `CCL` suite also have the functionality to compute the distance modulus, defined as,

$$\mu = 5\log_{10}(d_L/\mathrm{pc}) - 5 \tag{5}$$

and $a(\chi)$, the inverse of $\chi(a)$.

2.2. **Growth of perturbations**

Volunteer(s) in charge: Mustapha Ishak

To compute the linear growth factor of matter perturbations, $D(a)$, CCL solves the following usual differential equation:

$$\frac{d}{da}\left(a^3 H(a)\frac{dD}{da}\right) = \frac{3}{2}\Omega_M(a) a H(a) D, \tag{6}$$

using a Runge-Kutta Cash-Karp algorithm.

In doing this, CCL simultaneously computes the logarithmic growth rate $f(a)$, defined as:

$$f(a) = \frac{d\ln D}{d\ln a}. \tag{7}$$

CCL provides different functions that return the growth normalized to $D(a=1)=1$ and to $D(a \ll 1) \to a$. It employs an accelerated spline that is linearly spaced in the scale factor to interpolate the growth functions.

The growth calculations cover flat and curved ΛCDM, wCDM, and $(w0 + w_a)$CDM cosmologies. It should be noted though that the above is strictly valid for a Universe containing only dust-like matter components. A scale-independent growth rate is, for example, ill-defined in the presence of massive neutrinos which are not included in CCL's growth calculation.

Finally, CCL allows for an alternative 'modified gravity' cosmological model defined by a regular background $(w_0 + w_a)$CDM (with arbitrary K???) as well as a user-defined $\Delta f(a)$, such that the true growth rate in this model is given by $f(a) = f_0(a) + \Delta f(a)$, where $f_0(a)$ is the growth rate in the background model. Note that this model is only consistently implemented with regards to the computation of the linear growth factor and growth rates (which will also scale the linear power spectrum). All other CCL functions (including the non-linear power spectrum) will ignore these modifications. This model, and the interpretation of the predictions given by CCL, should therefore be used with care.

2.3. **Matter power spectrum**

Volunteer(s) in charge: Elisa Chisari

Suggested content: Matter power spectrum and transfer function definition. Approximations. Cosmic emulator. Baryonic physics model.

Theoretical predictions for cosmological observables such as galaxy clustering, gravitational lensing and cluster mass functions rely on knowledge of the distribution of matter from small to large scales in the Universe. The quantity most frequently used to describe the distribution of matter at a given wavenumber and redshift is the matter power spectrum, $P(k,z)$, defined as

$$\langle\tilde{\delta}(\mathbf{k},z)\tilde{\delta}(\mathbf{k},z)\rangle = (2\pi)^3 P(k,z)\ \delta_D^3(\mathbf{k}-\mathbf{k}') \tag{8}$$

where $\tilde{\delta}(\mathbf{k})$ is the Fourier component of the overdensity field at a given wavenumber and δ_D^3 is the Dirac delta function. $P(k,z)$ has units of volume and a dimensionless analogue is often defined as

$$\Delta^2(k,z) \equiv \frac{k^3}{2\pi^2}P(k,z). \tag{9}$$

`CCL` implements several different methods for making predictions for the matter power spectrum. Two of those methods, the BBKS (Bardeen et al. 1986) and Eisenstein & Hu (1998) approximations are only accurate to within a few percent and are implemented for validation purposes only. These approximations provide analytical expressions for the trasnfer function, $T(k)$, which is related to the matter power spectrum by $\Delta(k) \propto T^2(k)k^{3+n}$ and $\Delta^2(k) \propto k^3P(k)$. The normalization of the power spectrum is defined at $z=0$ by setting σ_8 to its value today.

The default `CCL` implementation uses the `CLASS` algorithm Blas et al. (2011) to obtain predictions for $P(k,z)$. In addition, `CCL` can also generate $P(k,z)$ predictions by emulation of cosmological numerical simuations, in particular, using the "Cosmic Emulator" developed by Lawrence et al. (2017). **Note this branch hasn't been merged yet and it doesn't have validated implementation for cosmologies with neutrinos**.

None of the previous methods account for the impact of baryonic physics on the distribution of matter, which is known to exceed percent level at scales $k \gtrsim 1/\mathrm{Mpc}$ (van Daalen et al. 2011; Vogelsberger et al. 2014; Hellwing et al. 2016; Springel et al. 2017) and can affect the extraction of cosmological parameters (Semboloni et al. 2011, 2013; Mohammed & Seljak 2014; Eifler et al. 2015; Mohammed & Gnedin 2017). To account for this effect, we incorporate in `CCL` an effective parametrisation (Schneider & Teyssier 2015) of the redistribution of matter as a consequence of feedback from Active Galactic Nuclei and adiabatic cooling. This parametrisation reproduces results from the OWLS simulations at $z=0$. We give an overview of each method in what follows.

BBKS approximation. CCL implements the analytical BBKS approximation to the transfer function (Bardeen et al. 1986), given by

$$T\left(q \equiv k/\Gamma h\mathrm{Mpc}^{-1}\right) = \frac{\ln[1+2.34q]}{2.34q} \times \\ [1+3.89q+(16.2q)^2+(5.47q)^3+(6.71q)^4]^{-0.25} \tag{10}$$

where $\Gamma = \Omega_m h$. The BBKS power spectrum option is primarily used as a precisely-defined input for testing the numerical accuracy of CCL routines (as described in Sec. **??**), and it is not recommended for other uses.

Eisenstein & Hu approximation. CCL also provides an approximation to the matter power spectrum as implemented by Eisenstein & Hu (1998) (E&H; we refer the reader to this paper for a detailed discussion of the fitting formulae).[2]

CLASS. The default configuration of CCL adopts predictions for the nonlinear matter power spectrum from this publicly available software (Blas et al. 2011). CLASS currently computes the non-linear power spectrum using the HaloFit prescription of Takahashi et al. (2012).

Cosmic emulator. The emulator (Lawrence et al. 2017) provides accurate predictions for the nonlinear matter power spectrum, at the 1% level for $z \leq 2$ and in the wavenumber range $k = [10^{-3}, 5]$ Mpc^{-1}. The allowed range of cosmological parameters that can be passed to the emulator is as follows:

$$\begin{aligned} 0.12 \leq \Omega_m h^2 &\leq 0.155, \\ 0.0215 \leq \Omega_b h^2 &\leq 0.0235, \\ 0.7 \leq \sigma_8 &\leq 0.9, \\ 0.55 \leq h &\leq 0.85, \\ 0.85 \leq n_s &\leq 1.05, \\ -1.3 \leq w_0 &\leq -0.7, \\ -1.73 \leq w_a &\leq -0.7,^{3} \\ 0.0 \leq \Omega_\nu h^2 &\leq 0.01.^{4} \end{aligned} \tag{11}$$

Baryonic correction model (BCM). CCL incorporates the impact of baryons on the total matter power spectrum via the BCM of Schneider & Teyssier (2015). The main consequences of baryonic processes are: to suppress the power spectrum at intermediate scales ($k \sim$ a few h/Mpc) due to the ejection of gas by Active

[2] Note that the implementation in CCL modifies Eq. 5 of Eisenstein & Hu (1998) using $a^{-1} = 1+z$ instead of the approximation $a^{-1} \sim z$. The difference in the resulting power spectra is negligible, but larger than 1 part in 10^4 for $k < 10\,h\,\mathrm{Mpc}^{-1}$.

Galactic Nuclei feedaback, and to enhance it at smaller scales due to adiabatic cooling. To account for these effects, BCM uses an effective decomposition for the impact of gas ejection (G) and the enhancement of the small scale profile due to star formation (S) to estimate the fractional effect of baryonic processes on the dark matter-only power spectrum ($P_{\rm DMO}$):

$$P_{\rm BCM}(k,z) = P_{\rm DMO}(k,z)G(k|M_c,\eta_b,z)S(k|k_s) \tag{12}$$

Three effective parameters govern the contribution of baryonic processes to modifying the total matter power spectrum:

- $\log_{10}[M_c/({\rm M}_\odot/h)]$: the mass of the clusters responsible for feedback, which regulates the amount of suppression of the matter power spectrum at intermediate scales;
- η_b: a dimensionless parameter which determines the scale at which suppression peaks;
- and k_s [h/Mpc]: the wavenumber that determines the scale of the stellar profile.

If these parameters are not specified by the user, `CCL` assumes the default parameters of Schneider & Teyssier (2015), which are in agreement with results from OWLS (van Daalen et al. 2011).

2.4. 2-point correlators

Volunteer(s) in charge: David Alonso, Tom McClintock

Suggested content: Kernels. Correlation functions. Nuisance parameters and functions.

In this section we will distinguish between *tracers* (quantities observed on the sky, such as number counts in a redshift bin, shear or CMB temperature fluctuations) and *contributions* to the total observed fluctuations of these tracers (such as the biased matter density term in number counts, redshift-space distortions, magnification, etc.).

2.4.1. Angular power spectra

The angular power spectrum between two tracers a and b can be written as:

$$C_\ell^{ab} = 4\pi \int_0^\infty \frac{dk}{k} \mathcal{P}_\Phi(k) \Delta_\ell^a(k) \Delta_\ell^b(k), \tag{13}$$

where $\mathcal{P}_\Phi(k)$ is the dimensionless power spectrum of the primordial curvature perturbations, and Δ^a and Δ^b are the transfer functions corresponding to these tracers. Each transfer function will receive contributions from different terms. Currently `CCL` supports three types of tracers: number counts, galaxy shape distortions and lensing convergence, with the following contributions:

Number counts. —The transfer function for number counts can be decomposed into three contributions: $\Delta^{\rm NC} = \Delta^{\rm D} + \Delta^{\rm RSD} + \Delta^{\rm M}$, where

- $\Delta^{\rm D}$ is the standard density term proportional to the matter density:

$$\Delta_\ell^{\rm D}(k) = \int dz\, p_z(z)\, b(z)\, T_\delta(k,z)\, j_\ell(k\chi(z)), \tag{14}$$

  where $j_\ell(x)$ is ℓ-th order spherical Bessel function, T_δ is the matter transfer function, $b(z)$ is the linear clustering bias for this tracer and $p_z(z)$ is the normalized distribution of sources in redshift (i.e. the selection function). The fluctuations in the number density of sources in different redshift bins are therefore treated by `CCL` as different tracers. Note that `CCL` currently does not support non-linear or scale-dependent bias, but future releases of it will do so under a number of schemes, including perturbative approaches as implemented in e.g. McEwen et al. (2016).

- $\Delta^{\rm RSD}$ is the linear contribution from redshift-space distortions:

$$\Delta_\ell^{\rm RSD}(k) = \int dz\, \frac{(1+z)p_z(z)}{H(z)} T_\theta(k,z)\, j_\ell''(k\chi(z)), \tag{15}$$

  where $T_\theta(k,z)$ is the transfer function of θ, the divergence of the comoving velocity field. Note that the RSD contribution to number counts is currently computed by `CCL` assuming a linear-theory relation between the matter overdensity and peculiar velocity fields, mediated by the scale-independent growth rate f. While this should not be problematic for wide photometric redshift bins and standard cosmological models, users should exercise care when interpreting results for narrow window functions or exotic cosmologies.

- $\Delta^{\rm M}$ is the contribution from lensing magnification:

$$\Delta_\ell^{\rm M}(k) = -\ell(\ell+1) \int \frac{dz}{H(z)} W^{\rm M}(z) \\ T_{\phi+\psi}(k,z)\, j_\ell(k\chi(z)), \tag{16}$$

where $T_{\phi+\psi}$ is the transfer function for the Newtonian-gauge scalar metric perturbations, and $W^{\rm M}$ is the magnification window function:

$$W^{\rm M}(z) \equiv \int_z^\infty dz'\, p_z(z') \frac{2-5s(z')}{2} \frac{r(\chi'-\chi)}{r(\chi')r(\chi)}. \tag{17}$$

Suggested content: Is it not $r(chi') - r(\chi)$ normally at numerator ? same for other integrals below. (from Jeremy, comparing with Bonvin et al. and other references) Here $s(z)$ is the magnification bias, given as the logarithmic derivative of the number of sources with magnitude limit, and $r(\chi)$ is the angular comoving distance (see Eq. 3).

Note that CCL currently does not compute relativistic corrections to number counts Challinor & Lewis (2011); Bonvin & Durrer (2011). Although these will be included in the future, their contribution to the total fluctuation is largely subdominant (see Alonso et al. (2015) and the two references above), and therefore it is safe to work without them in most cases.

Correlated galaxy shapes.—The transfer function for correlated galaxy shapes (intrinsic and lensed) is currently decomposed into two terms: $\Delta^{\rm SH} = \Delta^{\rm WL} + \Delta^{\rm IA}$, where

- $\Delta^{\rm L}$ is the standard lensing ("cosmic shear") contribution:

$$\Delta^{\rm L}_\ell(k) = -\frac{1}{2}\sqrt{\frac{(\ell+2)!}{(\ell-2)!}} \int \frac{dz}{H(z)} W^{\rm L}(z) T_{\phi+\psi}(k,z) \\ j_\ell(k\chi(z)), \tag{18}$$

  where $W^{\rm L}$ is the lensing kernel, given by

$$W^{L}(z) \equiv \int_z^\infty dz' p_z(z') \frac{r(\chi'-\chi)}{r(\chi')r(\chi)}. \tag{19}$$

- $\Delta^{\rm IA}$ is the transfer function for intrinsic galaxy alignments. CCL currently supports the so-called "non-linear alignment model", in which the intrinsic galaxy inertia tensor is proportional the local tidal tensor Hirata & Seljak (2004); Hirata et al. (2007):

$$\Delta^{\rm IA}_\ell(k) = \sqrt{\frac{(\ell+2)!}{(\ell-2)!}} \int dz\, p_z(z)\, b_{\rm IA}(z)\, f_{\rm red}(z) \\ T_\delta(k,z)\, \frac{j_\ell(k\chi(z))}{(k\chi(z))^2}. \tag{20}$$

Here $b_{\rm IA}$ is the so-called alignment bias, and $f_{\rm red}$ is the fraction of red galaxies in the sample. It is understood that only red galaxies are subject to this type of alignment mechanism. Suggested content: Elisa, check if phrasing is correct?

Lensing convergence.—The transfer function for the lensing convergence of a given source plane at redshift z_* receives only one contribution, given by

$$\Delta^{\kappa}_{\ell}(k) = -\frac{\ell(\ell+1)}{2}\int_0^{\chi_*}\frac{dz}{H(z)}\frac{r(\chi_*-\chi)}{r(\chi)r(\chi_*)}T_{\phi+\psi}(k,z), \tag{21}$$

where $\chi_* \equiv \chi(z_*)$.

It is worth noting that the equations above should be modified for non-flat cosmologies by replacing the spherical Bessel functions j_ℓ with their hyperspherical counterparts Kamionkowski & Spergel (1994). These are currently not supported by `CCL`, and their impact is mostly relevant on low multipoles. This will be revisited in future versions of the library.

2.4.2. **Correlation functions**

Volunteer(s) in charge: Elisa Chisari, Sukhdeep Singh? SS:Is this section necessary? The following expressions relating the angular power spectra and correlation functions are valid in the flat-sky approximation[5]. In all cases, $f_K(\chi)$ is comoving angular diameter distance, which differs from the radial comoving distance χ only in the case of cosmologies with non-zero curvature.

Galaxy-galaxy. The angular correlation function between two number-count tracers (labeled a and b here) is given by

$$\xi^{ab}(\theta) = \int d\ell \frac{\ell}{2\pi} C_\ell^{ab} J_0(\ell\theta), \tag{22}$$

where C_{ab} is the angular power spectrum between both tracers.

Lensing-lensing. The lensing correlation functions are [6]

$$\xi^{ab}_{+}(\theta) = \int_0^\infty d\ell \frac{\ell}{2\pi} J_0(\ell\theta) C_\ell^{ab}, \tag{23}$$

$$\xi^{ab}_{-}(\theta) = \int_0^\infty d\ell \frac{\ell}{2\pi} J_4(\ell\theta) C_\ell^{ab}, \tag{24}$$

[5] See the weak lensing review by Bartelmann & Schneider (2001), page 44 and Joachimi & Bridle (2010).

[6] from Schneider 2002 and Bartelmann & Schneider section 6.4.1

where the angular lensing convergence power spectrum C_ℓ^{ab} is given above (see Equations 18 and 44).

Galaxy-lensing. The correlation between a number count tracer a and a shear tracer b is given by

$$\xi^{ab}(\theta) = \int d\ell \frac{\ell}{2\pi} C_\ell^{ab} J_2(\ell\theta), \tag{25}$$

3-dimensional spatial correlation function.

In addition to the angular correlation functions, the 3-dimensional spatial matter correlation function $\xi(r)$ is calculated from the matter power spectrum $P(k)$ using

$$\xi(r) = \frac{1}{2\pi^2} \int dk\, k^2 P(k) \frac{\sin(kr)}{kr}\,. \tag{26}$$

2.5. Halo mass function

Volunteer(s) in charge: Antonio Villarreal, Tom M

Being able to calculate the halo abundance as a function of mass is a necessary step to being able to constrain cosmology with probes such as galaxy clustering, cluster abundance, and cluster clustering. While analytic functions are the traditional means of predicting evolution, calibration is frequently required against cosmological simulations. In order to reach the high precision for cosmological constraints in a self-consistent fashion, it is ultimately necessary to use cosmological simulations; we implement this with halo mass functions with parameters fit to these simulations. The halo mass functions implemented here are based on the spherical overdensity method of halo finding, in which the size of a halo can be defined with:

$$\bar{\rho}(r_\Delta) = \Delta \times \bar{\rho}_{\rm m}\,, \tag{27}$$

where a spherical halo with radius r_Δ has an average density $\bar{\rho}$ equal to the overdensity parameter Δ times the mean background density of the universe, $\rho_{\rm m}$. Within the literature, the choice of Δ can vary considerably, as observations focusing on the compact cores of haloes often will often take much larger values of Δ, while the fiducial definition in most halo clustering studies is a definition of $\Delta = 200$. We note that another common definition exists which utilizes the critical density of the universe, $\rho_{\rm crit}$; this introduces a conversion factor between the two definitions that must be accounted for. Currently, `CCL` only accepts overdensity parameters with respect to the mean matter density, but we plan to allow for self-consistent handling of critical density based definitions in the future.

The halo mass function is defined as

$$\frac{dn}{dM} = f(\sigma)\frac{\bar{\rho}_{\rm m}}{M}\frac{d\ln\sigma^{-1}}{dM}, \tag{28}$$

where n is the number density of halos of a given mass M associated with the RMS variance of the matter density field σ^2. In CCL calling this function returns the mass function in logarithmic mass bins, $dn/d\log_{10}M$, where the input is the halo mass M and scale factor a.

The halo mass M is related to σ by first computing the radius R that would enclose a mass M in a homogeneous Universe at $z = 0$:

$$M = \frac{H_0^2}{2G}R^3 \rightarrow \frac{M}{M_\odot} = 1.162\times10^{12}\Omega_M h^2\left(\frac{R}{1\,{\rm Mpc}}\right)^3. \tag{29}$$

The RMS density contrast in spheres of radius R can then be computed as

$$\sigma_R^2 = \frac{1}{2\pi^2}\int dk\,k^2\,P(k)\,\tilde{W}_R^2(k) \tag{30}$$

where $P(k)$ is the linear matter power spectrum and $\tilde{W}(kR)$ is the Fourier transform of a spherical top hat window function,

$$\tilde{W}_R(k) = \frac{3}{(kR)^3}[\sin(kR) - kR\cos(kR)]. \tag{31}$$

One commonly used halo mass function definition within the literature is the Tinker et al. (2010) fitting function. This fitting function has been developed using collisionless N-body simulation data, using haloes determined by spherical overdensities. This is an extension of the Tinker et al. (2008) halo mass function, which is also included within `CCL` as a comparative option. This fitting function assumes little change with respect to cosmological parameters. Further, it includes a redshift scaling with is assumed to sharply end at a redshift of $z = 3$. This halo mass function is calibrated within the range of $10^{10.5}h{\rm M}_\odot \leq M \leq 10^{15.5}h{\rm M}_\odot$ at a redshift of $z = 0$.

For purposes of comparison, we also have included the results of Angulo et al. (2012), which uses the Millenium XXL simulation in order to study galaxy cluster scaling relations. As part of this study, they have calculated their own fit to the Tinker et al. (2010) fitting function. While this additional halo mass function is available, it has not been extended to a broad range of overdensity parameter Δ, nor has it been extended beyond a redshift of $z = 0$.

The Tinker et al. (2008) fitting function uses the following parameterisation for the multiplicity function:

$$f(\sigma) = A\left[\left(\frac{\sigma}{b}\right)^{-a} + 1\right]e^{-c/\sigma^2}, \tag{32}$$

where A, a, b, and c are fitting parameters that have additional redshift scaling and σ is the RMS variance of the density field smoothed on some scale M at some scale factor a. This basic form is modified for the Angulo et al. (2012) formulation. The resulting form is

$$f(\sigma) = A\left[\left(\frac{b}{\sigma} + 1\right)^{-a}\right]e^{-c/\sigma^2}, \tag{33}$$

where the only change is in the formulation of the second term. Note that the fitting parameters in the Angulo et al. (2012) formulation do not contain any redshift dependence and the use of it is primarily for testing and benchmark purposes.

The Tinker et al. (2010) model parameterizes the halo mass function in terms of th peak height, $\nu = \delta_c/\sigma(M)$, where $\delta_c = 1.686$ is the critical density for collapse. The multiplicity function is then written as

$$f(\nu) = \alpha[1 + (\beta\nu)^{-2\phi}]\nu^{2\eta}e(-\gamma\nu^2/2). \tag{34}$$

We note that these halo mass functions, while implemented to high *numerical* accuracy, carry their own uncertainties. It has not been significantly studied whether the halo mass function (or halo bias function) is universal with respect to changes in dark energy parameterisation. Tinker et al. (2008, 2010) quote 5% accuracy of their mass funcitons. This result is consistent with the work of Watson et al. (2013), which also finds a 5% level difference in comparison to the Tinker fitting function. Further study will be required in the future in order to gain percent level accuracy in determining the halo mass function.

2.6. **Halo bias**

Volunteer(s) in charge: Antonio Villarreal, Tom M

An important step in many interpretations of the halo model is to have a measure of the bias of dark matter halos, defined as the ratio of the halo-matter power spectrum to the linear dark matter power spectrum,

$$b^2(k) = \frac{P_{hm}(k)}{P_{\rm lin}(k)}. \tag{35}$$

As with measures of the halo mass function, high accuracy cosmological constraints requires the use of numerical simulations to develop fitting functions and

emulators. We note that we will define haloes as in the above subsection focusing on the halo mass function. CCL currently implements the halo bias fitting function results in Tinker et al. (2010), though future improvements will likely require the use of emulator methods.

The Tinker et al. (2010) model parameterizes the halo bias in terms of the peak height, $\nu = \delta_c/\sigma(M)$, where δ_c is the critical density for collapse and is chosen to be 1.686 for this particular parameterization. We can then parameterize the halo function and halo bias as

$$b(\nu) = 1 - A\frac{\nu^a}{\nu^a + {\delta_c}^a} + B\nu^b + C\nu^c, \tag{36}$$

$$f(\nu) = \alpha[1 + (\beta\nu)^{-2\phi}]\nu^{2\eta}e(-\gamma\nu^2/2). \tag{37}$$

Again, while high *numerical* accuracy has been verified, there is a remaining uncertainty. Tinker et al. (2010) found a $\sim 6\%$ scatter when determining the halo bias due to differences in simulations alone. In addition, this parameterization does not include a careful exploration of any impact due to changes in the dark energy equation of state. As with the halo mass function, studies will be required to reach accuracy at the percent level for any cosmological predictions.

2.7. **Massive neutrinos**

Volunteer(s) in charge: Danielle Leonard

Suggested content: Neutrino mass/density definition. Transfer function. Definitions of hierarchies etc.

To include massive neutrinos, the suffix '_nu' may be appended to the last four ccl_parameters_create functions above. Using these functions without the _nu suffix will set the number of massive neutrinos to 0 and the effective number of massless neutrinos to 3.046. In the case of non-zero massive neutrinos, it may be desirable to set N_nu_rel such that $N_{\rm eff} = 3.046$ in the early universe. In agreement with CLASS, when using the default value of $T_{\rm NCDM}$ as described in section 3.1 below, N_nu_rel can be set to 2.0328, 1.0196, and 0.00641 for 1, 2 and 3 massive neutrinos respectively to achieve this.

For massive neutrinos, $\Omega_{\nu,\rm m}(a)$ is calculated by calling a set of functions contained in ccl_neutrinos.c. First, we assume that the mass of one neutrino is equal to $m_\nu^{\rm tot}/N_{\rm m}^\nu$ (recalling that we assume a single massive neutrino or equal-mass

neutrinos). We then compute a quantity called the effective temperature:

$$T_\nu^{\rm eff} = T_{\rm CMB} T_{\rm NCDM}. \tag{38}$$

Here, $T_{\rm NCDM}$ is used to explicitly set the value of m_ν/Ω_ν^0. Note that despite its name, $T_{\rm NCDM}$ is a dimensionless temperature rescaling rather than an actual temperature, following the nomenclature used by CLASS. We choose a default value of $T_{\rm NCDM} = 0.71611$, which corresponds to $m_\nu/\Omega_\nu^0 = 93.14$ eV, to agree with the default value set by CLASS. We define

$$\mu = \frac{m_\nu^{\rm tot} a}{N_{\rm m}^\nu T_\nu^{\rm eff}} \tag{39}$$

in units such that μ is dimensionless. We then conduct the phase-space integral required to get the neutrino density, and multiply by appropriate factors to obtain $\Omega_{\nu,{\rm m}}(a)$:

$$\Omega_{\nu,{\rm m}}(a) = N_{\rm m}^\nu \frac{8\pi^2 (\pi k_b)^3 k_b}{15 (c h_{\rm P})^3} \frac{8\pi G}{3h^2c^2} \left(\frac{T_\nu^{\rm eff}}{a}\right)^4 \times \left(\frac{7}{8} \int_0^{x_{\rm max}} dx\, x^2 \frac{\sqrt{x^2+\mu^2}}{\exp(x)+1}\right) \tag{40}$$

where $h_{\rm P}$ is Planck's constant and h is $H_0/100$ with H_0 in units of km / s / Mpc. $x_{\rm max}$ is set to 1000. The final bracketed term which includes the phase-space integral can be simplified in the limit where μ is very large or very small: for small μ, it is set to $\frac{7}{8}$, and for large μ, it becomes $\frac{5\zeta(3)}{18\pi^4}\mu \sim 0.2776\mu$.

3. Implementation of high-accuracy cosmological functions

3.1. Background functions & growth of perturbations

Volunteer(s) in charge: Mustapha Ishak Suggested content: Initial conditions. Normalization. Units and sensitivity to physical constants (G, c etc).

Including/excluding radiation in the computation of the comoving distances and the growth function can easily make a difference of 10^{-4} at the redshifts required in this comparison.

We have performed a comparison of the physical constants used in CCL to those used in GSL and CLASS as well as published sources such as NIST and Particle

Physics Handbook. Where possible, we have set constants to the values that are used internally in CLASS. This includes the value of the gravitational constant, the Boltzmann constant, the Planck constant, the speed of light, and the electron charge. CLASS does not include a definition of the solar mass or the Stefan-Boltzmann constant so we use the values used by GSL. After comparison between the physical constants used in CCL and those of the sources mentioned above, we have found better than 10^{-4} agreement with everything except the gravitational constant.

The value of the gravitational constant, G, enters into the critical density. We found that failure to define G with sufficient precision would result in lack of convergence at the 10^{-4} level between the different submissions. Importantly, note that CAMB barely has 10^{-4} precision in G (and similarly, there might be other constants within CAMB/CLASS for which one should check the precision level). For CCL, we are using the value from CLASS.

3.2. **Matter power spectrum**

Volunteer(s) in charge: Elisa Chisari, who else?

Suggested content: Interpolation strategy. Sensitivity to spline parameters and sampling in k. Settings that control speed vs. accuracy.

For the matter power spectrum, the spline is performed in two variables: the logarithmically-spaced wavenumber and the linearly-spaced scale factor. Splining the CLASS output leads to some precision loss (compared to direct outputs from CLASS). We quantify this, along with the impact of extrapolation, in the following subsection.

The computation of the power spectrum from CLASS can be significantly sped up by extrapolating in the range $k >$ K_MAX_SPLINE and $k <$ K_MIN. In this section, we describe the implementation of the extrapolation and the accuracy attained. These tests are performed in a flat ΛCDM cosmology with $\Omega_c = 0.25$, $\Omega_b = 0.05$, $A_s = 2.1 \times 10^{-9}$, $h = 0.7$ and $n_s = 0.96$.

We first describe the extrapolation at high wavenumbers. The introduction of the parameter K_MAX_SPLINE allows us to spline the matter power spectrum within the cosmo structure up to that value of k (in units of $1/$Mpc). A separate K_MAX parameter sets the limit for evaluation of the matter power spectrum. The range between

K_MAX_SPLINE$< k <$K_MAX is evaluated by performing a second order Taylor expansion in $\ln k$ within the static routine ccl_power_extrapol_highk.

First, we compute the first and second derivative of the $\ln P(k,z)$ at $k_0 =$ K_MAX $- 2\Delta \ln k$ by computing the numerical derivatives by finite differences using GSL. The fiducial choice for $\Delta \ln k$ is 10^{-2}. We then apply a second order Taylor expansion to extrapolate the matter power spectrum to $k >$K_MAX_SPLINE. The Taylor expansion gives

$$\ln P(k,z) \simeq \ln P(k_0,z) + \frac{d\ln P}{d\ln k}(\ln k_0, z)(\ln k - \ln k_0) + \frac{1}{2}\frac{d^2 \ln P}{d\ln k^2}(\ln k_0, z)(\ln k - \ln k_0)^2. \quad (41)$$

The accuracy of this approximation is shown in Figure 2 for redshifts $z = 0$ and $z = 3$. We compare the nonlinear matter power spectrum at $z = 0$ computed with the previously described approximation, to the matter power spectrum obtained by directly evaluating CLASS at the desired k value. In figure 2, we vary $\Delta \ln k$ and K_MAX_SPLINE at a time to determine the impact of the choice of parameters. We find that for typical values of $\Delta \ln k = 10^{-2}$ and K_MAX_SPLINE$= 50$/Mpc, $\ln P$ has converged to an accuracy that surpasses the expected impact of baryonic effects on the matter power spectrum at $k > 10$/Mpc. (For an estimate of the impact of baryons on the total matter power spectrum, see Schneider & Teyssier 2015.) The lower K_MAX_SPLINE is, the faster CCL will run. The optimum choice of K_MAX_SPLINE is left to the user for their particular application.

We also extrapolate the power spectrum at small wavenumbers within the static routine ccl_power_extrapol_lowk. In this case, the power spectrum below K_MIN is obtained by a power-law extrapolation with index n_s:

$$\log P(k < \text{K\_MIN}, z) = \log P(\text{K\_MIN}, z) + n_s(\log k - \log \text{K\_MIN}) \quad (42)$$

The value adopted for K_MIN depends on the choice of power spectrum method. For CLASS and the nonlinear power spectrum, we adopt K_MIN that coincides with the smallest wavenumber output by CLASS, K_MIN$= 7 \times 10^{-6}$/Mpc.[7] Note that this parameter is different from K_MIN_DEFAULT, which sets the minimum k for integrations and which is set by default to K_MIN_DEFAULT$= 5 \times 10^{-5}$/Mpc. Hence, in practice, no extrapolation is occurring in this case, unless the user specifically asks for an output power spectra below K_MIN_DEFAULT for their own purposes.

[7] For BBKS, the power spectrum is computed analytically at all k, there is no extrapolation. For the Eisenstein & Hu implementation, the splines of the power spectrum span K_MIN_DEFAULT$<$ $k <$K_MAX_SPLINE, so there is only extrapolation at high k.

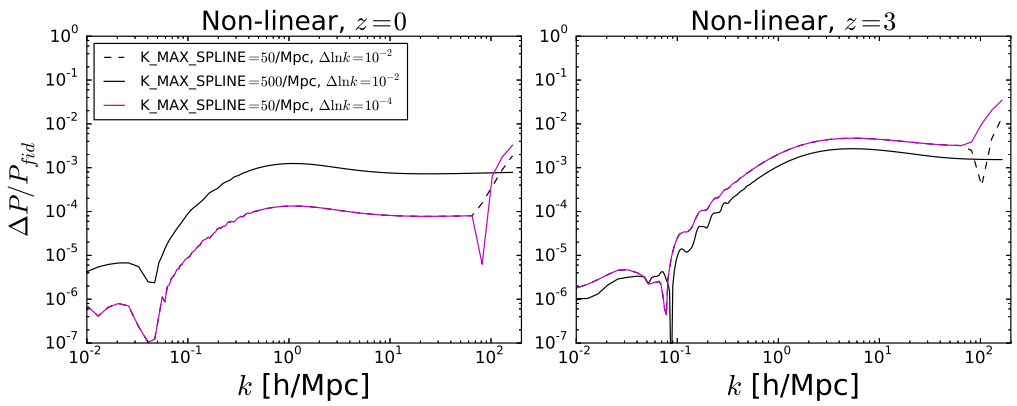

Figure 2. The relative error compared to direct CLASS outputs, P_{fid}, produced by splining the nonlinear matter power spectrum up to `K_MAX_SPLINE` and extrapolating beyond this value with a second order Taylor expansion the natural logarithm of the matter power spectrum. The left panel shows the results at $z = 0$. The right panel shows the results at $z = 3$. The standard `CCL` parameters adopted are those corresponding to the black dashed curve. The solid curve shows the accuracy when using `K_MAX_SPLINE`$= 500$/Mpc. The magenta curve shows the impact of using the higher resolution $\Delta \ln k = 10^{-4}$ in Eq. 41. For comparison, the impact of baryonic physics on the matter power spectrum is $\sim 10\%$ at $k = 1$/Mpc (Schneider & Teyssier 2015).

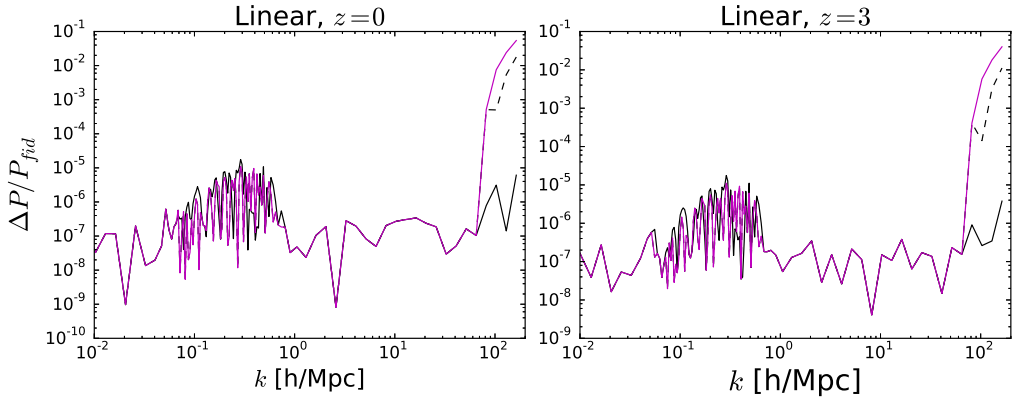

Figure 3. Same as Fig. 2 but for the linear matter power spectrum at $z = 0$ (left) and $z = 3$ (right).

With the implementation described above, the power spectrum splines are initialized up to `K_MAX_SPLINE`. This is also true for the linear matter power spectrum, which is used within `CCL` in particular to obtain σ_8 (see Eq. 30). We have tested here how the procedure described in the previous section affects the convergence of the linear matter power spectrum. We compare the fiducial `CCL` output to the case where we set `K_MAX_SPLINE`$= 5 \times 10^3$/Mpc. The result is shown in Figure 3. Although there is a significant difference ($\gtrsim 10\%$) between the linear power spectra at large k, we have confirmed that the difference in σ_8 is negligible. Nevertheless, for other applications that use the linear power spectrum, the user might need to increase the value of `K_MAX_SPLINE`.

As in the previous section, the power spectrum at small wavenumber is extrapolated using a power-law. This extrapolation is performed below a fiducial value of `K_MIN_DEFAULT`$= 5 \times 10^{-5}$.

We have found that changing `A_SPLINE_NA_PK` to 200, or changing `N_K` to 5000, does not change the results presented in Figures 2 and 3 in this section.

3.3. **Angular power spectra**

Volunteer(s) in charge: David Alonso

Different numerical approaches have been implemented in the library in order to expedite the computation of angular power spectra. We describe these here.

3.3.1. **Limber approximation**

Volunteer(s) in charge: David Alonso, Jérémy Neveu

As shown in Section 2.4.1, computing each transfer function contributing to a given power spectrum involves a radial projection (i.e. an integral over redshift or z or χ), and thus computing full power spectra consists of a triple integral for each ℓ. This can be computationally intensive, but can be significantly simplified in certain regimes by using the Limber approximation, given by:

$$j_\ell(x) \simeq \sqrt{\frac{\pi}{2\ell+1}}\,\delta\left(\ell+\frac{1}{2}-x\right). \tag{43}$$

Thus for each k and ℓ we can define a radial distance $\chi_\ell \equiv (\ell+1/2)/k$, with corresponding redshift z_ℓ. This approximation works best for wide radial kernels and high multipoles.

Substituting this in the expressions above, the simplified versions become:

$$C_\ell^{ab} = \frac{2}{2\ell+1}\int_0^\infty dk\, P_\delta\left(k, z_\ell\right)\tilde{\Delta}_\ell^a(k)\tilde{\Delta}_\ell^b(k). \tag{44}$$

where

$$\tilde{\Delta}^{\rm D}_\ell(k) = p_z(z_\ell)\, b(z_\ell)\, H(z_\ell) \tag{45}$$

$$\tilde{\Delta}^{\rm RSD}_\ell(k) = \frac{1+8\ell}{(2\ell+1)^2}\, p_z(z_\ell)\, f(z_\ell)\, H(z_\ell) - \tag{46}$$

$$\frac{4}{2\ell+3}\sqrt{\frac{2\ell+1}{2\ell+3}} p_z(z_{\ell+1})\, f(z_{\ell+1})\, H(z_{\ell+1}) \tag{47}$$

$$\tilde{\Delta}^{\rm M}_\ell(k) = 3\Omega_{M,0} H_0^2 \frac{\ell(\ell+1)}{k^2}\, \frac{(1+z_\ell)}{\chi_\ell} W^{\rm M}(z_\ell) \tag{48}$$

$$\tilde{\Delta}^{\rm L}_\ell(k) = \frac{3}{2}\Omega_{M,0} H_0^2 \sqrt{\frac{(\ell+2)!}{(\ell-2)}} \frac{1}{k^2}\, \frac{1+z_\ell}{\chi_\ell} W^{\rm L}(z_\ell) \tag{49}$$

$$\tilde{\Delta}^{\rm IA}_\ell(k) = \sqrt{\frac{(\ell+2)!}{(\ell-2)!}} \frac{p_z(z_\ell)\, b_{\rm IA}(z_\ell) f_{\rm red}(z_\ell) H(z_\ell)}{(\ell+1/2)^2} \tag{50}$$

3.3.2. **Beyond Limber:** `Angpow`

Volunteer(s) in charge: Jérémy Neveu

`CCL` incorporates natively routines to compute the C_ℓ^{ab} angular power spectra as described above without the Limber approximation. The algorithm performs first the integrals over z for both tracers, and ends with the k integral. This computation is much slower than using the Limber approximation, but it ends up with precise angular power spectra at low ℓ, and correct cross-correlations between tracers (the Limber approximation fails at reproducing the beat phenomenon $j_\ell(x) \times j_\ell(x')$). The integration of these routines has been tested against the `CLASS` code and recover the same angular power spectra when precision parameters are set to high values.

However, the computation of the C_ℓ^{ab} without the Limber approximation is too costly in terms of computing time using this method, if one wants to explore extensively a full cosmological parameter space. The aim of the `Angpow` software (Campagne et al. 2017) is to compute the angular power spectra C_ℓ^{ab} without any Limber numerical approximation in a faster way but still accurately. `CCL` has been linked to the `Angpow` code, which is briefly described here.

The angular power spectrum for two tracers C_ℓ^{ab} is computed in `Angpow` according to the following expression

$$C_\ell^{ab} = \iint_0^\infty {\rm d}z{\rm d}z' p_{z_1}(z_1) p_{z_2}(z') \times \int_0^\infty {\rm d}k\, f_\ell(z,k) f_\ell(z',k). \tag{51}$$

The auxiliary function $f_\ell(z,k)$ can be defined without loss of generality as

$$f_\ell(z,k) \equiv \sqrt{\frac{2}{\pi}}\, k\sqrt{P_\delta(k,z)}\, \widetilde{\Delta}_\ell(z,k) \tag{52}$$

with $\widetilde{\Delta}_\ell(z,k)$ the function describing the physical processes such as matter density fluctuations, redshift-space distortions as described for instance in references Durrer (2008); Yoo et al. (2009); Yoo (2010); Challinor & Lewis (2011); Bonvin & Durrer (2011). Currently, the `Angpow` version delivered with `CCL` only can deal with galaxy clustering tracers (no lensing) and this without the magnification lensing term (equation **??**). The incorporation of those transfer functions is left for a future work, but in principle `Angpow` has already the capability to treat them. For now, for galaxy clustering tracers we define $\widetilde{\Delta}_\ell(z,k)$ as

$$\widetilde{\Delta}_\ell(z,k) \approx b(z)\, j_\ell(k\chi(z)) - f(z)\, j''_\ell(k\chi(z)) \tag{53}$$

with $j_\ell(x)$ and $j''_\ell(x)$ the spherical Bessel function of order ℓ and its second derivative, and $f(z)$ the growth rate of structures.

In `Angpow`, the inner integral in k is computed first. To conduct the computation of such integral of highly oscillating functions, the 3C-algorithm described in details in reference (Campagne et al. 2017) is used. In brief, it lies on the projection of the oscillating $f_\ell(z,k)$ onto Chebyshev series of order 2^N, the product of the two Chebyshev series is performed with a 2^{2N} Chebyshev serie; then, the integral is computed thanks to a Clenshaw-Curtis quadrature. At the end the integrals over z are performed via again an optimised Clenshaw-Curtis quadrature. All the Chebyshev expansions and the Clenshaw-Curtis quadrature are performed via the *Discrete Cosine Transform* of type I from the DCT-I fast transform of FFTW library.

The `Angpow` software was tested against `CLASS` and the native `CCL` computation and can perform the same computations approximately an order of magnitude faster ($\mathcal{O}(1s)$). Its precision and speed parameters are optimised so that the relative numerical error compared with an high precision computation is two orders of magnitude below the relative cosmic variance $\sqrt{2/(2\ell+1)}$, from $\ell = 2$ to $\ell = 1000$.

As in the general case the Limber approximation is valid at high ℓ values, the `CCL` user can define an ℓ threshold to switch from the non-Limber slow computation to the faster Limber approximation.

3.4. **Correlation functions**

Volunteer(s) in charge: Elisa Chisari, Sukhdeep Singh?

Suggested content: Which basis to work in. Approach to performing oscillatory integrals.

In this section we describe the computation of correlation functions given the angular power spectrum.

We begin by briefly deriving the relation between correlation functions and power spectra. A (scalar) angular space observable, X, with spin s can be decompsed into spin spherical harmonics

$$X(\vec{\Omega}) = \sum_{\ell m} \tilde{X}_{\ell m} \ {}_sY_{\ell m}(\vec{\Omega}) \tag{54}$$

where $\vec{\Omega}$ refers to the angular coordinates on the sky. The angular cross correlation function of two tracers, X and Z of the large scale structure can be written in terms of their harmonic components, $\tilde{X}_{\ell m}$ and $\tilde{Z}_{\ell' m'}$ as

$$\langle XZ\rangle(\theta) = \left\langle \sum_{\ell,m} \sum_{\ell',m'} \tilde{X}_{\ell m} \tilde{Z}_{\ell' m'} \ {}_{s_X}Y_{\ell m}(\vec{\Omega}) \ {}_{s_Z}Y_{\ell' m'}(\vec{\Omega}+\theta) \right\rangle \tag{55}$$

$$= \sum_{\ell,m} C_\ell \ {}_{s_X}Y_{\ell m}(\vec{\Omega}) \ {}_{s_Z}Y_{\ell m}(\vec{\Omega}+\theta) \tag{56}$$

where s_X and s_Z are the spins of tracers, ${}_sY_{\ell m}$ are spin spherical harmonics and we also used the identity

$$\langle \tilde{X}_{\ell m} \tilde{Z}_{\ell' m'} \rangle = C_\ell \delta_D(m,m') \delta_D(\ell,\ell'), \tag{57}$$

Using the addition theorem for sping spherical harmonics and the relation between spherical harmonics and wigner D-matrices[8] (Ng & Liu 1999), we can write

$$\langle XZ\rangle(\theta) = \frac{1}{4\pi} \sum_\ell (2\ell+1) C_\ell D^\ell_{s_X,-s_Z}(0,\theta,0) \tag{58}$$

[8] https://en.wikipedia.org/wiki/Wigner_D-matrix

The galaxy number count is spin zero and shear is a spin-2 object. Thus the expressions for various auto and cross correlations are

$$\xi_{gg}(\theta) = \frac{1}{4\pi}\sum_\ell (2\ell+1) C_\ell^{gg} D^\ell_{0,0}(0,\theta,0) = \frac{1}{4\pi}\sum_\ell (2\ell+1) C_\ell^{gg} P_\ell(\cos\theta) \tag{59}$$

$$\langle g\gamma_T\rangle(\theta) = \frac{1}{4\pi}\sum_\ell (2\ell+1) C_\ell^{g\gamma} D^\ell_{2,0}(0,\theta,0) = \frac{1}{4\pi}\sum_\ell (2\ell+1)\sqrt{\frac{(\ell-2)!}{(\ell+2)!}} C_\ell^{g\gamma} P^2_\ell(\cos\theta) \tag{60}$$

$$\langle g\kappa\rangle(\theta) = \frac{1}{4\pi}\sum_\ell (2\ell+1) C_\ell^{g\kappa} D^\ell_{0,0}(0,\theta,0) = \frac{1}{4\pi}\sum_\ell (2\ell+1) C_\ell^{g\kappa} P_\ell(\cos\theta) \tag{61}$$

$$\xi_+(\theta) = \frac{1}{4\pi}\sum_\ell (2\ell+1) C_\ell^{\gamma\gamma *} D^\ell_{2,2}(0,\theta,0) \tag{62}$$

$$\xi_-(\theta) = \frac{1}{4\pi}\sum_\ell (2\ell+1) C_\ell^{\gamma\gamma *} D^\ell_{2,-2}(0,\theta,0) \tag{63}$$

Using relations from Kilbinger et al. (2017), we can write

$$C_\ell^{g\kappa} = \frac{1}{2}\frac{(\ell+1)!}{(\ell-1)!} C_\ell^{g\psi} \tag{64}$$

$$C_\ell^{g\gamma} = \frac{1}{2}\sqrt{\frac{(\ell+2)!}{(\ell-2)!}} C_\ell^{g\psi} \tag{65}$$

$$\langle g\gamma_T\rangle(\theta) = \frac{1}{4\pi}\sum_\ell \frac{(2\ell+1)}{\ell(\ell+1)} C_\ell^{g\kappa} P^2_\ell(\cos\theta)\,. \tag{66}$$

The `ccl_tracer_corr_legendre` routine computes these transform to convert angular power spectra, C_ℓ, into correlation functions. We use the associated Legendre function implementation from the `GSL` library. `ccl_tracer_corr_legendre` routine evaluations can be very slow, especially for polynomials P^m_ℓ with $m > 0$. Note that P^m_ℓ evaluations need to be done only once and can then be saved as long as ℓ and θ values do not change. However, `CCL` has not yet implemented this feature.

The `ccl_tracer_corr_legendre` methods currently does not support the computation of the shear correlation functions $\xi_\pm$, and the approximate `ccl_tracer_corr_fftlog` (described below) should be used instead. As demonstrated in Kilbinger et al. (2017) and Kitching et al. (2017), this approximation should be sufficient even for futuristic surveys.

Hankel Transform—Notice that in the flat-sky limit, the expressions in Eqs. 59–63 can be written as Hankel transforms using the relation between $D^\ell_{m,m'}$ and bessel

functions J (for $\ell \gg m, m'$)

$$D^{l}_{m,m'}(0,\theta,0) \approx J_{m-m'}(\ell\theta) \tag{67}$$

Same expressions can also be derived from Eqs. **??–??** using relation between P^m_ℓ and bessel functions J_m (de Putter & Takada 2010)

$$P^m_\ell(\cos\theta) = (-1)^m \frac{(\ell+m)!}{(\ell-m)!} \ell^{-m} J_m(\ell\theta) \tag{68}$$

which yields final expressions that coincide with Eq. 24.

For numerical integration to get (projected) correlation functions using hankel transform, we make use of the public code `FFTlog`[9] (Hamilton 2000; Talman 2009). In brief, `FFTlog` works on functions periodic in log space, by writing the Hankel Transform as a convolution between Bessel functions and the function of interest (in this case C_ℓ). A version of this code is included in `CCL` with minor modifications.

3.5. **Halo mass function**

Volunteer(s) in charge: Antonio Villarreal, Tom M

For achieving 10^{-4} precision in $\sigma(M)$ and the normalisation of the power spectrum, we checked that the integral of σ_8 and $\sigma(M)$ converged for the chosen values of $\{k_{\rm min}, k_{\rm max}\}$. The desired precision is set taking into account the fact that the numerical integral calculated $\sigma^2(M)$ while we require $\sigma(M)$.

Derivatives are calculated utilizing a spline that is built off of the previously described $\sigma(M)$ spline. As such, these splines cover the range from 10^6 to $10^{17} M_\odot$. For each value of $\log(M)$ in our spline evaluation, we calculate the value of $\sigma(M)$ half a step in either direction. We use the difference compared to the mass spacing to calculate an approximate derivative, which is then used in the spline interpolation. This has been tested to meet our necessary precision for the halo mass function within the mass range explored by Tinker et al. (2010). We note that the accuracy is reduced at the edges of these splines. Exploring extreme mass ranges require manually changing the initialization of these splines.

In order to accomodate a wide range of values of the overdensity parameter Δ, we have generated a spline interpolation between best fit values as defined by Tinker et al. (2008) and Tinker et al. (2010). This covers a dynamic range from $\Delta = 200$

[9] http://casa.colorado.edu/~ajsh/FFTLog/

to 3200. Within this range, we interpolate in the space of the fit parameter and $\log \Delta$ using the Akima interpolation built from piecewise third order polynomials. We have chosen this rather than the fitting formulas utilized in Tinker et al. (2010) in order to assure high precision match to the Tinker halo mass function when choosing a value of $Delta$ directly from the paper.

3.6. **Halo Bias**

Volunteer(s) in charge: Tom M

Suggested content: Need someone that has looked at the code to check the statement about how the fit function parameters change.

Details of the halo bias implementation are similar to those of subsection 3.5. The same spline for $\sigma(M)$ is used to calculate the peak height ν used in the fitting functions available in CCL.

Similar to the treatment of the halo mass function, the fitting function parameters in the mass functions depend on halo definition. We consider only changes in overdensity Δ, as was done in Tinker et al. (2010). We implement the explicit dependence of the fit parameters on Δ, following Table 2 in Tinker et al. (2010). These do not require the use of interpolators.

3.7. **Massive neutrinos**

Volunteer(s) in charge: Danielle Leonard

Suggested content: Approximations. Approach to performing integrals.

3.8. **Implementation of photometric redshifts**

Volunteer(s) in charge: Danielle Leonard

Suggested content: Explain approach to photometric redshifts and flexibility to pass user-defined function.

Redshifts of LSST galaxies will be obtained via photometry. Performing any cosmological analysis which incorporates redshift information therefore requires a

model for the probability of measuring a photometric redshift $z_{\rm ph}$ for an object with hypothetical spectroscopic redshift $z_{\rm s}$. In order to maintain agnosticism towards the optimal model, and hence to allow for the future inclusion of advancements from ongoing research, CCL allows the user to flexibly input a photometric redshift model. In addition, for ease of use, CCL provides the option of using a built-in function for a simple Gaussian photometric redshift probability distribution.

In order to use CCL with a custom-input photometric redshift model, the user writes a function which accepts as input a photometric redshift, a spectroscopic redshift, and a pointer to a structure containing any further parameters of the model. This function will return the probability of measuring the input photometric redshift given the input spectroscopic redshift. The photometric redshift model can then be used, for example, when computing $\frac{dN}{dz}^i$ in photometric redshift bin i, as given by:

$$\frac{dN}{dz}^i = \frac{\frac{dN}{dz}\int_{z_i}^{z_{i+1}} dz' p(z,z')}{\int_{z_{\rm min}}^{z_{\rm max}} dz \frac{dN}{dz}\int_{z_i}^{z_{i+1}} dz' p(z,z')} \tag{69}$$

where $p(z,z')$ is the photometric redshift probability distribution, and z_i and z_{i+1} are the photo-z edges of the bin in question. In the case of the simple Gaussian photometric redshift model for which support is included in CCL out of the box, $p(z,z')$ is given as

$$p(z,z') = \frac{1}{\sqrt{2\pi}\sigma_z}\exp\left(-\frac{(z-z')^2}{2\sigma_z^2}\right). \tag{70}$$

4. Validation over the ΛCDM parameter space

4.1. Accuracy criteria

Volunteer(s) in charge: ?

Suggested content: Why we chose the 10^{-4} level of accuracy. Accuracy summary statistic used in VARRIC.

4.2. Validation tests on fiducial parameter sets

Volunteer(s) in charge: Elisa Chisari, Antonio Villarreal, Phil Bull, Elisabeth Krause

Suggested content: Overview of the code comparison. How the codes used for comparison were selected. Validation plots for several observables.

Cosmological models for code comparison project								
Model	Ω_m	Ω_b	Ω_Λ	h_0	σ_8	n_s	w_0	w_a
flat LCDM	0.3	0.05	0.7	0.7	0.8	0.96	-1	0
w_0 LCDM	0.3	0.05	0.7	0.7	0.8	0.96	-0.9	0
w_a LCDM	0.3	0.05	0.7	0.7	0.8	0.96	-0.9	0.1
open w_a LCDM	0.3	0.05	0.65	0.7	0.8	0.96	-0.9	0.1
closed w_a LCDM	0.3	0.05	0.75	0.7	0.8	0.96	-0.9	0.1

Table 2. Cosmological models for code comparison project.

Our goal is for outputs of `CCL` to be validated against the results of a code-comparison project run within LSST-DESC down to a 10^{-4} or pre-established accuracy level if possible. In some cases, this level of accuracy is not necessary, as other systematics which have not been considered in this version of `CCL` yet are expected to have a larger fractional impact. In the cases where this applies, we make it clear below.

A code comparison project was carried out among members of TJP where the following outputs of cosmological forecast codes were compared and validated:

1. growth factor at $z = 0, 1, 2, 3, 4, 5$,
2. comoving radial distance $[\mathrm{Mpc}/h]$ at the same redshifts, as well as the corresponding distance moduli,
3. linear matter power spectrum, $P(k)$, from BBKS (Bardeen et al. 1986) in units of $(\mathrm{Mpc}/h)^3$ at $z = 0, 2$ in the range $10^{-3} \leq k \leq 10h/\mathrm{Mpc}$ with 10 bins per decade,
4. Eisenstein & Hu matter power spectrum in units of $(\mathrm{Mpc}/h)^3$ at $z = 0$ in the range $10^{-3} \leq k \leq 10h/\mathrm{Mpc}$ with 10 bins per decade, and
5. the mass variance at $z = 0$, $\sigma(M, z = 0)$ for $M = \{10^6, 10^8, 10^{10}, 10^{12}, 10^{14}, 10^{16}\}\mathrm{M}_\odot/h$.

These predictions were produced and compared for different cosmologies, which are listed in the table below. The results agree to better than 0.1% relative accuracy for comoving distance and growth factor among all submissions, and for $P(k)$ and $\sigma(M)$ among codes which use the same BBKS conventions.

We noticed that there are 2 typos for the BBKS transfer function in "Modern Cosmology" (Dodelson 2004) compared to the original BBKS paper. The quadratic

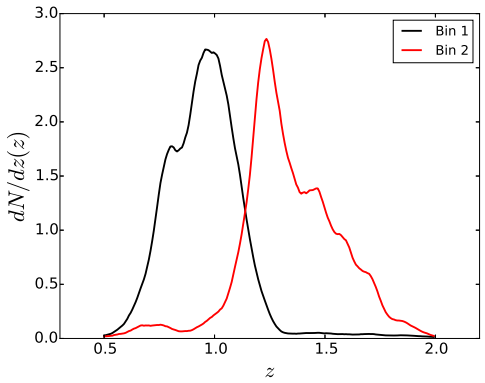

Figure 4. Binned redshift distributions used for code comparison project.

term should be $(16.1q)^2$ and the cubic term should be $(5.46q)^3$. The BBKS equation is correct in Peacock (1999). Using the wrong equation can give differences in the results above the 10^{-4} level.

In a second stage, we used the BBKS linear matter power spectrum from the previous step to compare two-point statistics for two redshift bins, resulting in three tomography combinations, $(1-1)$,$(1-2)$,$(2-2)$. We computed the following quantities:

- projected galaxy clustering tomography power spectra: density term only (no magnification, RSD, etc.) with non-evolving linear bias $b(z) = 1$, in the range $10 < \ell < 10000$, using 5 bins per decade,
- angular convergence tomography power spectra: leading order convergence term only (no magnification), in the same range and with the same resolution as the case above,
- angular galaxy clustering tomography correlation function, in the range $0.01\,\mathrm{deg} < \theta < 5\,\mathrm{deg}$, using 5 bins per decade, and
- angular shear tomography correlation functions (ξ_+,ξ_-), similarly to above.

We adopted the following analytic redshift distributions: a Gaussian with $\sigma = 0.15$, centered at $z_1 = 1$; and another Gaussian with the same dispersion but centered at $z_2 = 1.5$. We repeated the exercise for two redshift distribution histograms shown in Figure 4.

In this second step, only 2 codes have been compared so far. More outputs are needed to guarantee convergence. Preliminarily, from these outputs, we have concluded that:

- The cross-correlation between bins is particularly sensitive to the number of points where the kernels have been sampled.
- The accuracy of the correlation function is sensitive to $\ell_{\rm max}$. We had to use up to $\ell_{\rm max} = 3 \times 10^4$ for convergence (and we could not achieve 0.01% convergence).
- The large scales of the correlation function are sensitive to $\ell_{\rm min}$. The use of the flat-sky approximation is also relevant on these scales.
- For sufficiently high precision, the correlation functions are sensitive to how the power spectrum is sampled and interpolated.

For C_ℓ computations, we required the relative difference between CCL and the benchmarks to be $< 10^{-3}$. We performed the test both for analytic redshift distributions and histograms.

To obtain realistic targets for the convergence of correlation function computations for LSST analyses, we calculate the expected statistical uncertainty of the clustering and lensing correlation functions of the LSST gold sample (c.f. Sect. **??**), assuming an effective source galaxy density of $n_{\rm eff} = 26 {\rm gal/sq\,arcmin}$ for galaxy shape distortions, and galaxy density of $n_{\rm gold} = 45 {\rm gal/sq\,arcmin}$ for number counts. Specifically, we calculate the Gaussian covariance of angular correlation functions following the formalism of Joachimi et al. (2008), and note that leaving out the non-Gaussian covariance terms makes this convergence criterion more conservative. We split the galaxy samples into 10 tomography bins, defined to contain equal numbers of galaxies. The accuracy test then proceeds as follows. We compared the difference between CCL calculated lensing and clustering correlations and the benchmarks for the analytic redshift distributions and for auto-correlations of redshift bins only. To pass the benchmark test, we required that this difference be smaller than half of the value of the errorbar derived from the covariance for each correlation function computed. Specifically, we take the value of the covariance in the bins centered at $z = 1$ and $z = 1.5$ to compare to the benchmarks.

The 3-dimensional correlation function $\xi(r)$ was validated by comparing with an independent transform code based on the method of Ogata (2005) implemented in the hankel.py python package. We calculate $\xi(r)$ by transforming the CCL BBKS linear power spectrum using this independent code and using a fine sampling of $P(k)$. This is done for the five cosmologies listed in Table 2 and redshifts $z = 0, 1, 2, 3, 4, 5$. We then compare with the $\xi(r)$ from CCL with a sampling of $P(k)$ equal to N_K_3DCOR bins per decade. The default value of N_K_3DCOR = 1000 results in fractional agreement at the level of $< 4 \times 10^{-3}$ for $0.1 < r < 50$ Mpc increasing to

< 0.01 at $r = 100$ Mpc. Above this scale $\xi(r)$ approaches zero. If higher accuracy is desired, the user can increase the value of `N_K_3DCOR`.

Additionally, independent codes were utilized to test the accuracy of halo mass function predictions. For the halo mass function, we compare the value of σ, $\log(\sigma^{-1})$, and the value of the halo mass function in the form used in Tinker et al. (2008),

$$\log[(M^2/\bar{\rho}_m)dn/dM]. \tag{71}$$

We note that while we maintain the 10^{-4} for our evaluations of σ, the accuracy degrades to a value of 5×10^{-3} for the halo mass function evaluation, primarily at the high halo mass and high redshift domains. We find that this increased error is acceptable, as the level of precision is significantly better than the accuracy of current halo mass function models.

4.3. **Validation of the power spectrum over parameter space**

Volunteer(s) in charge: Phil Bull

Suggested content: 'Fair' sampling of parameter space with Latin Hypercubes. Summary statistics used, and binning of power spectrum in k and z. Precision settings used for CLASS. Estimates of run-time. Results. Range of validity.

While concentrating on individual points in cosmological parameter space allows us to perform detailed validation tests, as above, it is important for `CCL` to also be validated across a wide range of cosmological parameter values, e.g. to ensure validity for MCMC analyses. In this section, we present a set of validation tests for the `CCL` linear and non-linear matter power spectrum functions that spans a broad rnage of ΛCDM parameters.

Covering a full range of all 5 ΛCDM parameters on a regular grid would be prohibitively expensive, so an alternative method for fairly (but more sparsely) sampling the parameter space is needed. We use Latin Hypercube Sampling to determine a tractably-sized set of sample points. This splits the parameter space into a regular grid with a given number of cells per dimension. The sample points are then chosen by going through each dimension in turn and choosing a cell at random *without replacement*, so that a given cell of a given dimension is only ever chosen once. This is repeated until all cells in all dimensions contain a single sample. This has the effect of covering the space uniformly but sparsely. The exact location of the sample within each cell can be chosen from a uniform distribution

Parameter	Range
h	$[0.5, 0.9]$
Ω_c	$[0.1, 0.4]$
Ω_b	$[0.018, 0.052]$
A_s	$[1.5 \times 10^{-9}, 2.5 \times 10^{-9}]$
n_s	$[0.93, 0.99]$

Table 3. Ranges of ΛCDM parameters.

within that cell, but for simplicity we put each sample at the cell centre. The ranges used for each parameter are shown in Table 3.

For each set of parameters, we then calculate the linear and non-linear power spectra using `CCL` for a range of redshifts ($z \in \{0.0, 0.5, 1.0, 1.5, 2.0, 2.5\}$) and wavenumbers ($k \in [10^{-4}, 10^{0}]$ Mpc^{-1}). A corresponding set of reference power spectra is then produced using CLASS, which we run with settings chosen to produce high-precision results, taken from the `pk_ref.pre` precision file that is bundled with CLASS.

To quantify the level of agreement between the `CCL` and reference power spectra, we use the following summary statistic that can be summed over a chosen set of bins in redshift and wavenumber:

$$\Delta = \sum_{ij} \Theta \left(\log_{10} \left| \frac{P_{\rm CCL}(k_i, z_j) - P_{\rm ref}(k_i, z_j)}{P_{\rm ref}(k_i, z_j) \Delta_{\rm thres}} \right| \right) . \tag{72}$$

Here, $\Delta_{\rm thres}$ is a target threshold for the fractional deviation between the power spectra, and we have defined $\Theta(x) \equiv x \;\; (x \geq 0)$ and 0 otherwise. Bins where the `CCL` power spectrum deviates from the reference power spectrum by a fraction less than $\Delta_{\rm thres}$ do not contribute to the statistic, so the aim is to have $\Delta = 0$ (i.e. no deviation beyond the threshold in any bin). If deviations are found, however, they are weighted logarithmically – one large deviation of several orders of magnitude affects the statistic as much as a few smaller deviations of order $\sim \Delta_{\rm thres}$.

4.4. **Validation of the Cosmic Emulator**

Volunteer(s) in charge: Elisa Chisari Suggested content: Explain automated tests for cosmic emulator

This branch has not been merged yet. Lawrence et al. (2017) quantified the accuracy of the matter power spectrum from their emulation scheme by comparing their predictions to the resulting power spectra from the numerical simulations used

for the calibration of their scheme. We repeated this procedure making the emulator predictions via CCL for a subset of emulator cosmologies. For cosmologies without neutrinos, we required the matter power spectrum at $z = 0$ to be within 1% of the smoothed simulated power spectrum from Lawrence et al. (2017) (see their Figure 6). Similarly, we required 3% accuracy for cosmologies with neutrinos (their Figure 5).

5. Usage

Volunteer(s) in charge: Renee

The CCL is released as part of the code from the Dark Energy Science Collaboration of LSST, and as such can be downloaded from the collaboration's github repository[10].

The documentation of the code is provided through Doxygen[11]. CCL needs the GNU Scientific Library to install **reference needed**, more detailed instructions are given in the README file of the repository. Instructions on how to generate a Docker **reference needed** image are provided for portability to different architectures.

There are a suite of test files located in the `tests` directory within the main CCL code, these tests run, for example, a user-defined form for the photometric redshift of a source and how it integrates and tests with the various codes and libraries within CCL. **Connect to validation section above. Explain automated tests. Mention Jupyter notebooks.**

Something about our license.

6. Outlook

Volunteer(s) in charge: David Alonso, Elisabeth Krause

Suggested content: Extended parameter spaces. Example use case (integration into LSS pipeline).

[10] https://github.com/LSSTDESC/CCL
[11] https://github.com/LSSTDESC/CCL/blob/master/doc/html/index.html

We would like to thank the organisers of the the DESC collaboration meetings at: Oxford (July 2016), SLAC (March 2016), and ANL (2015), and the LSST-DESC Hack Week organisers (CMU, November 2016), where this work was partly developed. We would also like to acknowledge the contribution of the participants of the TJP Code Comparison Project, some of whom are among the CCL contributors, for providing the benchmarks for testing CCL. Finally, we are grateful for the feedback received from other working groups of DESC, including Strong Lensing, Supernovae and Photometric Redshifts.

Author contributions are listed below.
Husni Almoubayyed: Reviewed code/contributed to issues.
David Alonso: Co-led project; developed structure for angular power spectra; implemented autotools; integrated into LSS pipeline; contributed to: background, power spectrum, mass function, documentation and benchmarks; reviewed code
Jonathan Blazek: Planning capabilities and structure; documentation and testing.
Philip Bull: Implemented the Python wrapper and wrote documentation for it; general bug fixes, maintenance, and code review; enhanced the installer and error handling system.
Jean-Éric Campagne: Angpow builder and contributed to the interface with CCL.
N. Elisa Chisari: Co-led project, coordinated hack projects & communication, contributed to: correlation function & power spectrum implementation, documentation, and comparisons with benchmarks.
Alex Drlica-Wagner: Helped with document preparation.
Zilong Du: Implemented the 3d correlation function.
Tim Eifler: Reviewed/tested code.
John Ellison: Implemented the 3d correlation function; documentation of 3d correlation function.
Renée Hlozek: Contributed initial code for error handling structures, reviewed other code edits.
Mustapha Ishak: Contributed to planning of code capabilities and structure; reviewed code; identified and fixed bugs.
Matthew Kirby: Performed comparison of physical constants.
David Kirkby: Writing, testing and reviewing code. Asking questions.
Elisabeth Krause: Initiated and co-led project; developed CLASS interface and error handling; contributed to other code; reviewed pull requests.
C. Danielle Leonard: Wrote and tested code for LSST specifications, user-defined photo-z interface, and support of neutrinos; reviewed other code; wrote text for this note.
Christiane S. Lorenz: Contributed to accurate high-redshift cosmological background quantities and benchmarked background splines.
Phil Marshall: Helped with document preparation.

Thomas McClintock: Wrote Python and doxygen documentation.
Sean McLaughlin: Wrote doxygen documentation and fixed bugs/added functionality to distances.
Jérémy Neveu: Contributed to Angpow and built the interface with CCL.
Stéphane Plaszczynski: Contributed to Angpow and contributed to the interface with CCL.
Javier Sanchez: Modified setup.py to allow pip installation and uninstall.
Sukhdeep Singh: Contributed to the correlation functions code.
Anže Slosar: Wrote and reviewed code.
Antonio Villarreal: Contributed to initial benchmarking, halo mass function code, and general code and issues review.
Michal Vrastil: Wrote documentation and example code, reviewed code.
Joe Zuntz: Wrote initial infrastructure, C testing setup, and reviewed code.

References

Alonso, D., Bull, P., Ferreira, P. G., Maartens, R., & Santos, M. G. 2015, ApJ, 814, 145

Angulo, R. E., Springel, V., White, S. D. M., et al. 2012, MNRAS, 426, 2046

Bardeen, J. M., Bond, J. R., Kaiser, N., & Szalay, A. S. 1986, ApJ, 304, 15

Bartelmann, M., & Schneider, P. 2001, PhR, 340, 291

Blas, D., Lesgourgues, J., & Tram, T. 2011, CLASS: Cosmic Linear Anisotropy Solving System, Astrophysics Source Code Library, ascl:1106.020

Bonvin, C., & Durrer, R. 2011, PhRvD, 84, 063505

Campagne, J.-E., Neveu, J., & Plaszczynski, S. 2017, ArXiv e-prints, arXiv:1701.03592

Challinor, A., & Lewis, A. 2011, PhRvD, 84, 043516

Chevallier, M., & Polarski, D. 2001, International Journal of Modern Physics D, 10, 213

de Putter, R., & Takada, M. 2010, PhRvD, 82, 103522

Dodelson, S. 2004, Modern Cosmology, Vol. 57, 60–61, doi:10.1063/1.1784308

Durrer, R. 2008, The Cosmic Microwave Background (Cambridge University Press)

Eifler, T., Krause, E., Dodelson, S., et al. 2015, MNRAS, 454, 2451

Eisenstein, D. J., & Hu, W. 1998, ApJ, 496, 605

Hamilton, A. J. S. 2000, MNRAS, 312, 257

Hellwing, W. A., Schaller, M., Frenk, C. S., et al. 2016, MNRAS, 461, L11

Hirata, C. M., Mandelbaum, R., Ishak, M., et al. 2007, MNRAS, 381, 1197

Hirata, C. M., & Seljak, U. 2004, PhRvD, 70, 063526

Joachimi, B., & Bridle, S. L. 2010, A&A, 523, A1

Joachimi, B., Schneider, P., & Eifler, T. 2008, A&A, 477, 43

Kamionkowski, M., & Spergel, D. N. 1994, ApJ, 432, 7

Kilbinger, M., Heymans, C., Asgari, M., et al. 2017, ArXiv e-prints, arXiv:1702.05301

Kitching, T. D., Alsing, J., Heavens, A. F., et al. 2017, MNRAS, 469, 2737

Lawrence, E., Heitmann, K., Kwan, J., et al. 2017, ApJ, 847, 50

Linder, E. V. 2003, Physical Review Letters, 90, 091301

McEwen, J. E., Fang, X., Hirata, C. M., & Blazek, J. A. 2016, JCAP, 9, 015
Mohammed, I., & Gnedin, N. Y. 2017, ArXiv e-prints, arXiv:1707.02332
Mohammed, I., & Seljak, U. 2014, MNRAS, 445, 3382
Ng, K.-W., & Liu, G.-C. 1999, International Journal of Modern Physics D, 8, 61
Ogata, H. 2005, Publications of the Research Institute for Mathematical Sciences, 41, 949
Peacock, J. A. 1999, Cosmological Physics, 704
Schneider, A., & Teyssier, R. 2015, JCAP, 12, 049
Semboloni, E., Hoekstra, H., & Schaye, J. 2013, MNRAS, 434, 148
Semboloni, E., Hoekstra, H., Schaye, J., van Daalen, M. P., & McCarthy, I. G. 2011, MNRAS, 417, 2020
Springel, V., Pakmor, R., Pillepich, A., et al. 2017, ArXiv e-prints, arXiv:1707.03397
Takahashi, R., Sato, M., Nishimichi, T., Taruya, A., & Oguri, M. 2012, ApJ, 761, 152
Talman, J. 2009, Computer Physics Communications, 180, 332
Tinker, J., Kravtsov, A. V., Klypin, A., et al. 2008, ApJ, 688, 709
Tinker, J. L., Robertson, B. E., Kravtsov, A. V., et al. 2010, ApJ, 724, 878
van Daalen, M. P., Schaye, J., Booth, C. M., & Dalla Vecchia, C. 2011, MNRAS, 415, 3649
Vogelsberger, M., Genel, S., Springel, V., et al. 2014, Nature, 509, 177
Watson, W. A., Iliev, I. T., D'Aloisio, A., et al. 2013, MNRAS, 433, 1230
Yoo, J. 2010, PhRvD, 82, 083508
Yoo, J., Fitzpatrick, A. L., & Zaldarriaga, M. 2009, PhRvD, 80, 083514