Euclid Theoretical Predictions Specification Document (TPSD)

Document Num.: EC-???-???

Issue: 0

Date: April 26, 2016

Authors: This document is under the custodianship of the Euclid Inter-Science Working Group Taskforce for Forecasting.

Authorised by: TWG

Abstract: *This document contains the specification of the remit of the inter-science working group taskforce for forecasting and probe combination. Details of how this group relates to the SWGs is defined in the PPSAID.*

Document Change History

Title: *Euclid Theoretical Predictions Specification Document (TPSD)*			
Issue	**Rev.**	**Date**	**Reason for Change**
Draft 0	0	2015-09	Creation

Approved by:

NAME	**Organization**	**Date**	**Signature**

Contents

1 Aim

This document aims at responding to the general need of a common forecasting pipeline, developed in agreement with all SWGs. It outlines the approaches that Euclid cosmological parameter forecasting will take during the implementation phase (2015–) and up until launch. This will cover standard 'Fisher' forecasting methodology, as well as more complex predictions. Specifications for the primary probe forecasts are defined, as well as how these should be combined together and with external cosmological parameter constraints, in particular CMB data. This document will also define the general code infrastructure that the forecasting pipeline will use, in preparation for the theoretical calculations that will enter into the Euclid likelihood pipeline (that will take output from the SGS LE3, and theoretical computations outlined here).

Therefore this document covers:

1. Definition of the primary probe theoretical computation data flow
2. The specification of code comparisons at each stage of the data flow – in building a single common pipeline
3. Specifications for code comparison for existing non-trunk or external forecast codes
4. How to compare Fisher matrices for a minimal Fisher matrix code
5. Definition of coding practices – including language, licensing and publication policy when using IST products etc.
6. How the code interacts with the SGS
7. How to deal with new input provided by the SWGs (for example if there is a new parameter or a new forecasting method developed by the SWG, how does the IST react).

Applicable Documents

- PPSAID (Post Processing Science Analysis Implementation Document)
- Theory Parameter Defintion Document
- SGS Data Flow Down Description Data
- GDPRD (Ground Data Processing Requirements Document)
- OU RSDs (Requirement Specification Documents)
- MPD (Mission Performance Document)

2 Relation to the Euclid Likelihood Pipeline

This document describes how the *theoretical* calculations for the Euclid primary probes are made. These calculations are initially, before launch, to be used to create Euclid 'forecasts'. A forecast is a performance prediction for Euclid that makes a statement about the accuracy and precision that Euclid will be able to achieve the Level 0 objectives outlines in the Mission Performance Document (MPD). This document also describes how the primary probes will be combined.

The construction of the Euclid likelihood pipeline is described conceptually in the PPSAID. This describes how there are three elements that need to come together in order for Euclid to achieve its science objectives. These are 1) The output from SGS LE3 - i.e. statistics computed on the data, 2) Output from the cosmological simulations Science Working Group - who will create realisations of Universe that can be used to create cosmology-dependent

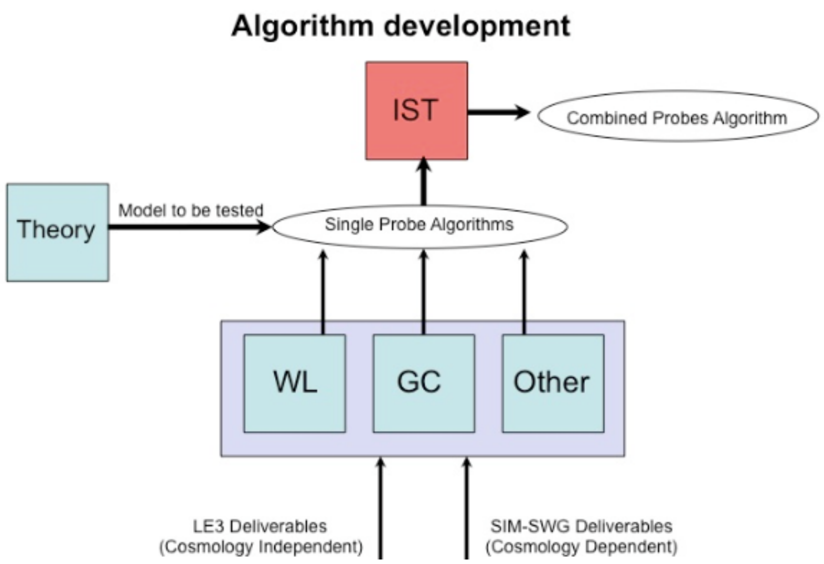

Figure 1: Figure from PPSAID. A schematic of the flow of information to create the algorithms for the individual probes and the combined probes. Note that the development of the algorithms is a collaborative effort between the cosmology SWGs and SGS.

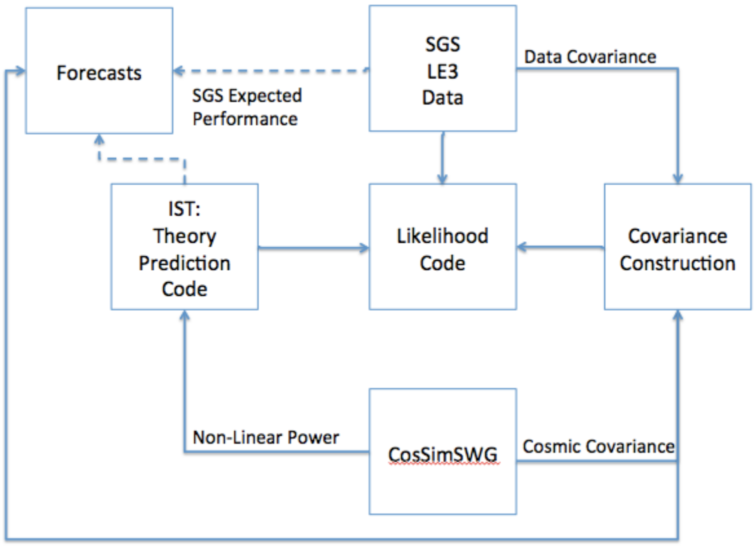

Figure 2: How the theory calculations relate to the SGS, CosmoSIMSWG and Euclid likelihood codes. The IST makes theoretical predictions that are incorporated into both forecasts and ultimately into likelihoods.

covariance estimates, and predictions for matter clustering, 3) Predictions of the measured statistics from theories against which the data can be tested (i.e. “forecasts” before launch).

The theory contruction formally does not involve the SGS algorithm development, but will be constructed such that the outputs from the SGS LE3 will be inputs into a likelihood pipeline that will also take inputs from the cosmological simulations and input from the theoretical algorithms. This relationship is show in Figure 2.

3 Definition of Cosmological Statistic Theory-Flows

The first step in creating the Euclid forecasting pipeline is to define the data-flow from cosmological parameters, and input survey characteristics, to theoretical predictions of relevant statistics. This is a similar procedure to that used in the Euclid ground segment to define the data-flow from pixels and spectro-photometry to Level-3 products. We adopt a very similar procedure here except that we refer to this as a theory-flow to demark it from SGS activities. The statistics, i.e. ‘observables’, that we will use as the primary probe are

- The weak lensing tomographic power spectrum
- Galaxy clustering: BAO, RSD, P(k)
- Secondary cosmological probes: Supernovae, clusters, ...

- **discussion point: do we need more e.g. 2nd-priority tiers for LE3, i.e. spherical-Bessel stats from the beginning**
- **discussion point: observables may change with the model: ex. isotropy tests may want different ℓ requirements.**

In the following we will define a theory-flow down for each of these primary probe statistics, and also for the combination of the primary probes, and the combination with CMB data. In doing this each element in the theory-flow will be referred to as a "**Processing Function (PF)**" using exactly the same naming convention as in the SGS for algorithm entities.

The broad grouping of the processing functions are into

- Inputs: Model and parameter definitions, including cosmological mode and parameters, and nuisance parameters.
- Inputs: Survey characteristics, including (expected) distributions of observed quantities, and survey geometry.
- Internal: Functions that act internally to the flow, for example matter power spectrum estimatation.
- Output: These are the expected theoretical statistics, and manipulations of those statistics e.g. derivatives, Fisher matrices etc.

Each processing function is given a unique identifier that refers to the pipeline(s) in which it is embedded, the nature of the processing function, and a short descriptor of its function.

Finally each theory-flow is required to have a matching data-flow in the SGS documentation i.e. that it is expected that the statistics that the theory-flow will compute from a theoretical side will also be computed by the SGS from the data for eventual inclusion in a Euclid likelihood pipeline. In the event that this document identifies missing data-flows then this should be discussed with the SWG leads, and SGS PO.

3.1 Weak Lensing: Tomographic Weak Lensing Theory-Flow

Here we specify the theory-flow for the tomographic weak lensing. In Figure 3 we show the flowdown from cosmological parameters, nuisance and survey parameters to output statistics and Fisher matrices.

3.2 Galaxy Clustering:

TBW

3.3 Weak Lensing+Galaxy Clustering:

TBW

4 Coding, Requirements & Verification

Here we discuss some aspects of the coding, requirements and verification of the PFs.

4.1 Requirements

Each processing function must have an associated requirement. A requirement must be necessary, unambiguous and tracable. These requirements will be of severable different types. For example

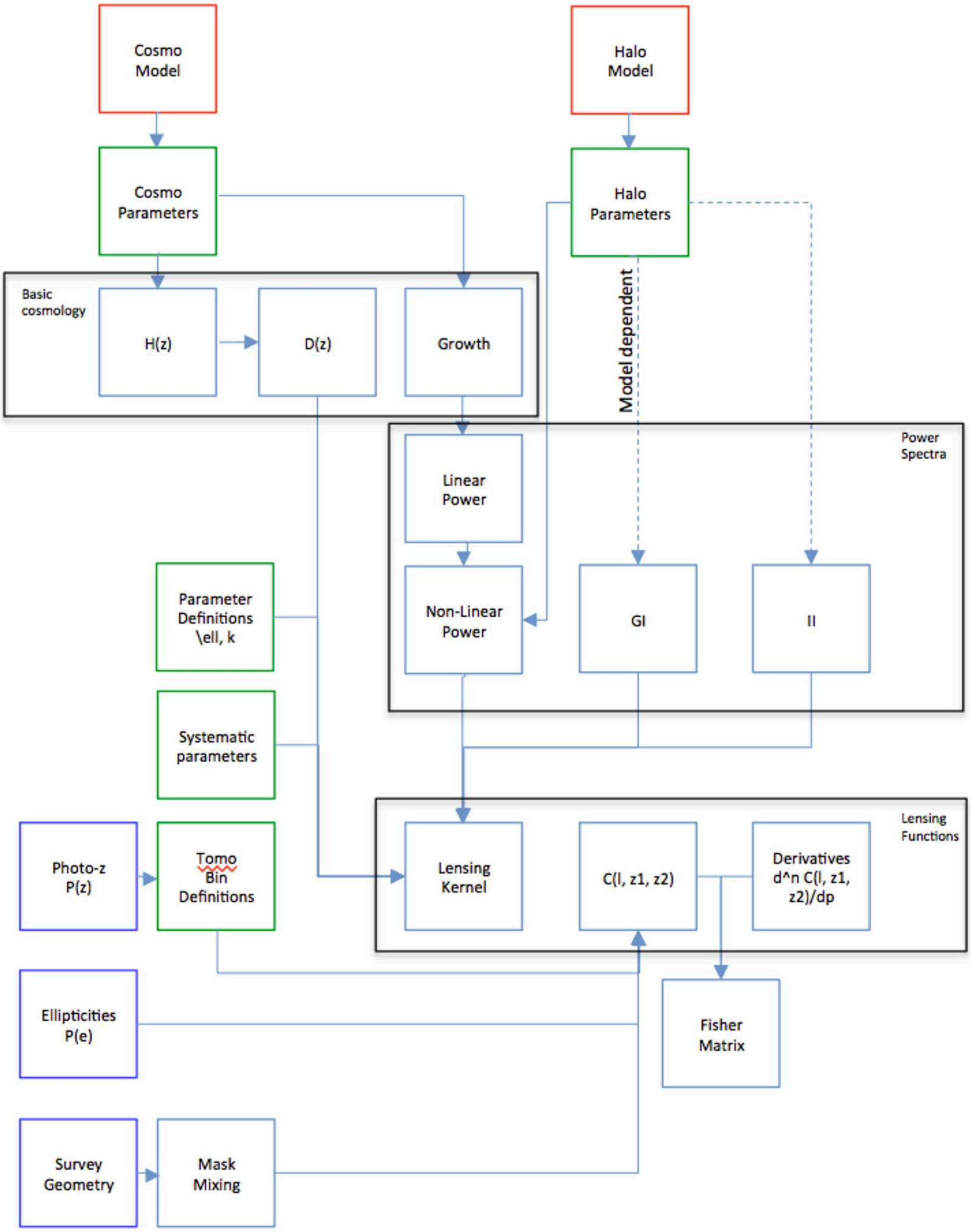

Figure 3: Weak lensing tomographic theory-flow. **discussion point: how detailed we make each box will be deligated to individuals to take ownership over each PF. For example in the growth PF, this could/should rather be for the**

- Some processing functions will have trivial 'existence' requirements flow that are simply to ensure that parameters exist in the specified header file and that they are read in correctly. These parameters will be expected to be traceable either to the Parameter definition document, GDPRD, OU RSD, or SciRD requirement in which the parameter is defined in exact detail; either a parameter of interest (i.e. cosmological) or a nuisance parameter (i.e. a systematic).
- Some processing functions will have numerical 'budget allocations' that have flow-down from higher level breakdowns with regard to the top-level science objectives. For example an error on the Figure of Merit caused by binning in some quantity may have flow-down from a requirement that has shared a larger error-on-FoM budget with other PFs in the SGS or cosmological simulations PFs.

4.2 Verification

As well as requirements each PF will be expected to have a verification test associated with it. This means a specified set of tests that with known input produce known outputs such that the code can be verified to be computing what it is intended. In the case of theory-flows these verfication test should ideally be of a theoretical (analytic) nature in particular regimes of applicability, and numerical in others. Cross-validation verification tests are also encouraged for each PF, where several (> 2) independently written codes can be compared and the output compared for each on.

In the SGS as well as verification tests there are also 'validation' tests that compare the PF outputs to calibration data. There is no such data in the theory-flow case, however there are retrodictive tests on current data that can be performed. These tests will require, where possible, the output of the functions and statistics to be tested against current data, which should be consistent with the outputs (albiet at much large tolerance than will be required for Euclid). Some PF may be more amenable for this, for example certain parts of PFs that involve CMB cross-correlations can be compared against Planck data.

4.3 Coding Standards

We will adopt the same general coding practices as the SGS, in terms of language, library use and tracability. These can be found in the SGS SIRD document. However, as it is expected to be scientists and not software engineers that create these codes, we do not adopt the next level of engineering i.e. the use of LODEEN and CODEEN environments.

In order to facilitate code production we will use the following tools: **to be dicussed, TBW**

There is also the possibility of using external codes, outside Euclid for forecasting purposes. If this is the case a special committee will be formed and asked to assess the pros and cons of the use of such code, and the amount of time estimated to code a similar Euclid-specific version estimated. In the event that it is voted by the IST to be the best course to adopt such a code then the IST coordinators will form an MOU between the code authors and the Euclid IST should be sought, that should be done independently, and in addition to, the license agreements of the code. The MOU will cover in particular the ownership and license of any Euclid-specific modules that may contain prioritary information, authorship rights incurred, and so forth.

5 Forecasting Procedures

In this section we outline the two mechanisms through which robust forecasting for Euclid will be achieved.

5.1 Holistic Approach – Comparing Currently Existing Pipelines

The first approach will be to use *existing* forecasting codes that are already at a mature stage. In this case the full end-to-end theory-flow is being tested in a single step by comparing, for a given set of well defined inputs, the final Fisher matrix output or similar. In this it is required that

1. The inputs are well defined and common to all forecasting pipelines (e.g. parameter definitions)
2. That the outputs are well defined (e.g two-derivative Fisher matrix elements)
3. The metrics that will be used for the comparison (e.g. the DETF Figure of Merit), and a measure of agreement in these metrics that must be achieved in order to confirm that the codes are consistent.

In this case it is not possible to objective verify that any given code is returning the true performance, but the inter-subjectivity of more than one code can be verified by having many code do the comparison. In the event that there is inconsistency in the code then in the first instance simple debugging is encouraged and in the second instance an atomistic approach should be adopted.

The holistic approach will result in inter-subjectively verified, consistent, forecasts for Euclid, but will not constitute a production of a verified Euclid theory code. For this we will pursue the atomistic approach below.

The IST will provide for each theory-flow a set of templates that can be used to test forecasting pipeline in this holistic manner. These are provide in Appendices **to be defined**.

5.2 Atomistic Approach – Building a Euclid Cosmology Pipeline

Although the holistic approach to forecasting can result in forecasts that are external self-consistent, we also wish to build a theory code that verified to be both internal and externally self consistent. To this end we will develop the atomistic approach to forecast building where each PF in the theory-flow is compared on a case-by-case basis, such that the result is a likelihood code that – by construction – has some objectively verified forecasting capability built into it. This approach is significantly slower than a holistic approach, which we will pursue for more short term forecasting objectives.

For each PF the IST will create a series of PF requirements that must be met in order for the PF to be agreed to be usable. Each PF can be developed in isolation, although this may cause mis-matches between lower-level PF and upper-level ones when internal PF are written. Therefore we will pursue a ‘top-down’ construction i.e. from the inputs subsequently down through the theory-flow to the output level. For each PF the requirement and verification will define

1. Analytic examples where the output can be verified against a known truth i.e. systems tests of the PF. That must be reproduced at machine-precision.
2. A series of numerical convergence tests that the PF must be shown to reach stability in. In particular for derivative and integral approximations.
3. A series of metrics against which more than one independently code can be compared. These metrics should cover the whole range of functional output expected to be applicable for Euclid e.g. comoving distance at all redshifts, not only comoving distance ‘figure of merit’, and such metrics should agree at machine-precision.

We will pursue the construction of several theory pipelines written in several different languages such that language-specific libraries and assumptions are more ready identified as development moves down through the pipeline. At each stage a record must be kept of the requirements and evidence of the PF being passed.

5.3 Definition of Cosmological & Nuisance Parameters

Here we define the common set of cosmological and nuisance parameters that will be used as a starting point for both approaches to forecasting.

need to refer to the theory parameter definition document here.

References for GC: [**?**]. References for WL: [**?**] General references on Fisher and Euclid: [**?**, **?**]

In the event that a SWG generates a new set of parameters, e.g. in the case that a new theory or new observations lead to the requirements that additional parameters are included in a forecast, this will be proposed by the SWG in question in a technical note written to the IST. The IST managers will then enter discussion with the SWG in question on the implementation. Any additional parameter will be recorded in the mission data base, and parameter definition documents, and this document will be updated accordingly in the even that a parameter is decided to be included.

6 Management of the IST

The IST will be subdivided into several workpackages to reflect the tasks at hand. These workpackages will be decided between the IST managers and the IST team, and recorded in the SWG Work Package Description document. Workpackages will be reviewed in an organic process, i.e. created and reviewed on a case by case basis as the need arises, on approximately a 6-month to 1-year rotating timetable.

7 End Comments

This document does the following:

1. Defines the scope of the IST with regard to the SGS, and SWGs.
2. Contains and defines the theory-flow for the primary cosmological probes.
3. Defines the procedure through which the primary cosmological probes forecasts will be constructed both through a holistic (end-to-end) approach and an atomistic (pipeline-building approach).
4. In the following appendices describes the defined holistic procedure of the GC and WL primary probes.

The following will be captured in specific separate documents that will refer to this one.

1. Requirements and verification specifications for each of the primary cosmological probe theory-flows (one for each flow).
2. Specifications of the atomistic specifications for each theory flow.
3. Results of the holistic and atomistic forecasts, these will appear documented as a Euclid technical notes, and may also lead to publications.

The above documents will be recorded on the Euclid redmine and uploaded to the Euclid ESA RSSD under the SWG section when ratified.

8 Galaxy Clustering Holistic Forecast Definitions

[1] We recall the main equations to be used. Observed linear spectrum at all redshifts:

$$P_{r,obs}(k_{ref}, \mu_{ref}; z) = P_s(z) + \frac{D_A(z)^2_{ref} H(z)}{D_A(z)^2 H_{ref}(z)} \frac{P(k,z)}{P(k,z=0)} b^2(z)(1+\beta(z)\mu^2)^2 P(k,z=0)\,, \tag{1}$$

We use:

- For the linear matter power spectrum $P(k,z) \to$ input files provided by Enea Di Dio and Bin Hu.
- For the growth index $f(z,k)$ (needed for $\beta_d(z) = f(z)/b(z)$), $\to$ input files provided by Enea Di Dio and Bin Hu (k dependence not necessary. k fixed to k = 0.1 h/Mpc). Input files need to be done differently for future tests).
- For the $H(z)$ and $D_A(z)$ we use analytical expressions (see below, sec. 8). SC checked that the analytical derivatives wrt ω_b and ω_c (calculated considering h as a parameter) now match also the numerical ones.

Here $b(z)$ is the bias (provided by column 5 of Tab.(1)), $\beta(z)$ is the redshift-distortion factor, $P(k)$ is the undistorted linear matter spectrum (provided in the input with the correct normalization), μ is the direction cosine, D_A is the angular diameter distance (provided in the input) and H is the Hubble rate at the shell redshift z (provided in the input). $P_s(z)$ is the z- dependent shot-noise correction due to discreteness in the survey. The subscript ref (for 'reference') indicates quantities calculated in the fiducial model. In linear theory we have $\beta_d(z) = f(z)/b(z)$ where $f(z) \equiv d\log G/d\log a$ is the growth rate (f is provided directly in the input for every cosmology, G is the growth factor but you should not need it, the input provides directly the growth rate f).

The Fisher matrix for every redshift bin shell is integrated in k (Right now doing the μ integral from 0 to 1 and multiplying the integrand by 2)):

$$F_{ij} = \frac{1}{8\pi^2} \int_{-1}^{1} \mathrm{d}\mu \int_{k_{\min}}^{k_{\max}} k^2 \cdot \mathrm{d}k \frac{\partial \ln P_{obs}(z;k,\mu)}{\partial\theta_i} \frac{\partial \ln P_{obs}(z;k,\mu)}{\partial\theta_j} \cdot \left[\frac{n\,P_{obs}(z;k,\mu)}{n\,P_{obs}(z;k,\mu)+1}\right]^2 V_{\mathrm{s}} \tag{2}$$

where the θ's are the parameters: $\theta = \{\omega_c \equiv \Omega_{c_0} h^2, \omega_b \equiv \Omega_{b_0} h^2, h, n_s, \ln(10^{10} A_s), b_i, P_s\}$ (τ has been fixed to fiducial because it is degenerate when using only WL and GC).

The input provides the spectra $P(z,k)$ (already corresponding to the A_s normalization) and all other functions $f(z)$, $D_A(z)$, $H(z)$ for each model.

$n(z)$ is the galaxy number density at redshift z and V_s is the survey volume of the entire (so all 4π) redshift shell dz. The power spectra are all obtained numerically by solving the background and perturbation equations of the system for each value of the parameters and they are provided as input.

The derivative of the galaxy power spectrum, at each redshift (so z is fixed in the middle of the bin), with respect to the parameters (in this case ω_c) is:

$$\begin{aligned} \left.\frac{\mathrm{d}\ln P_{\mathrm{obs}}(\bar{z},k,\mu;\theta_i)}{\mathrm{d}\theta_i}\right|_{fid} &= \frac{\partial \ln P_{obs}(\bar{z},k,\mu;\theta_i)}{\partial \ln P_s(\bar{z})} \frac{\partial \ln P_s(\bar{z})}{\partial\theta_i} + \frac{\partial \ln P_{obs}(\bar{z},k,\mu;\theta_i)}{\partial \ln f(\bar{z})} \frac{\partial \ln f(\bar{z})}{\partial\theta_i} \\ &+ \frac{\partial \ln P_{obs}(\bar{z},k,\mu;\theta_i)}{\partial \ln H(\bar{z})} \frac{\partial \ln H(\bar{z})}{\partial\theta_i} + \frac{\partial \ln P_{obs}(\bar{z},k,\mu;\theta_i)}{\partial \ln D_A(\bar{z})} \frac{\partial \ln D_A(\bar{z})}{\partial\theta_i} \\ &+ \frac{\partial \ln P_{obs}(\bar{z},k,\mu;\theta_i)}{\partial \ln P(k,\bar{z})} \frac{\partial \ln P(k,\bar{z})}{\partial\theta_i} + \frac{\partial \ln P_{obs}(\bar{z},k,\mu;\theta_i)}{\partial \ln b(\bar{z})} \frac{\partial \ln b(\bar{z})}{\partial\theta_i} \\ &+ \frac{\partial \ln P_{obs}(\bar{z},k,\mu;\theta_i)}{\partial k} \frac{\partial k}{\partial\theta_i} + \frac{\partial \ln P_{obs}(\bar{z},k,\mu;\theta_i)}{\partial\mu} \frac{\partial\mu}{\partial\theta_i} \end{aligned} \tag{3}$$

[1] For the moment this part has been extracted from the TWG documentation of the code comparison, whose contributors were: L.Amendola, S. Camera, Santiago Casas, Enea Di Dio, Bin Hu, Elisabetta Majerotto, Valeria Pettorino, Domenico Sapone.

In general, one needs to include a redshift error σ_z and further multiply: $P_{obs} = P_{obs} e^{-k^2\mu^2\sigma_z^2/H(z)^2}$. The last two terms are non-vanishing if one takes the Alcock-Paczynski effect into account for k and μ, since they are also affected by geometrical terms.

The derivatives are:

$$\frac{\partial \ln P_{obs}(\bar{z}, k, \mu; \theta_i)}{\partial \ln P_s(\bar{z})} = \frac{1}{P_{obs}(\bar{z}, k, \mu; \theta_i)} \tag{4a}$$

$$\frac{\partial \ln P_{obs}(\bar{z}, k, \mu; \theta_i)}{\partial \ln f(\bar{z})} = \frac{2\beta(\bar{z})\mu^2}{1+\beta(\bar{z})\mu^2} \tag{4b}$$

$$\frac{\partial \ln P_{obs}(\bar{z}, k, \mu; \theta_i)}{\partial \ln b(\bar{z})} = \frac{2}{1+\beta(\bar{z})\mu^2} \tag{4c}$$

$$\frac{\partial \ln P_{obs}(\bar{z}, k, \mu; \theta_i)}{\partial \ln P(k, \bar{z})} = 1 \tag{4d}$$

$$\frac{\partial \ln P_{obs}(\bar{z}, k, \mu; \theta_i)}{\partial \ln H(\bar{z})} = 1 + \frac{4\beta(z)\mu}{1+\beta(z)\mu^2}(-\mu(\mu^2-1)) + \frac{\partial \ln P(\bar{z}, k; \theta_i)}{\partial k} k\mu^2 \tag{4e}$$

$$\frac{\partial \ln P_{obs}(\bar{z}, k, \mu; \theta_i)}{\partial \ln D_A(\bar{z})} = -2 + \frac{4\beta(z)\mu}{1+\beta(z)\mu^2}(-\mu(\mu^2-1)) + \frac{\partial \ln P(\bar{z}, k; \theta_i)}{\partial k} k(\mu^2-1) \tag{4f}$$

All these terms are set as follows:

- $\frac{\mathrm{d}\ln P(\bar{z},k)}{\mathrm{d}\theta_i}$ are numerical derivatives, coming from the input files (of course $P(\bar{z}, k)$ does not depend on bias or P_s). The input files are however provided at different k's; therefore, when calculating the derivative you need to interpolate in k first and then calculate the difference at fiducial $(1 \pm \epsilon)$.
- $f(\bar{z}, k)$ is given by the input file: while in the input f is a function of k, for the comparison, we fix k = 0.1 h/Mpc.
- $\frac{\mathrm{d}\ln f(\bar{z})}{\mathrm{d}\theta_i}$ are numerical derivatives, coming from the input files (k fixed as above).
- $\frac{\mathrm{d}\ln H(\bar{z})}{\mathrm{d}\theta_i}$ and $\frac{\mathrm{d}\ln D_A(\bar{z})}{\mathrm{d}\theta_i}$ are given by the analytical expressions described below (sec. 8)

We tested three different methods:

- Direct derivative: you integrate directly all terms of equation 3 into eq.2;
- Numerical derivative: evaluate the numerical derivative of eq.2 just calculating $P_{obs}(\bar{z}, k)$ at $\theta_{i,fid}(1 \pm \epsilon)$ and then divide by $2\epsilon\theta_{i,fid}$;
- Jacobian method (or extended BAO Seo-Eisenstein method, see [**?**])

All three methods match when there is no AP effect. With AP effect, the Jacobian method differs from the other two. For the following comparison, fisher matrices are provided using the direct derivative method.

The Hubble parameter and the angular diameter distance

We here assume that we consider h as a parameter, rather than Ω_Λ. The Hubble Parameter, for the cosmology we are using, is

$$H(z) = 100h\sqrt{\frac{\Omega_c h^2 + \Omega_b h^2 + \Omega_\nu h^2}{h^2}(1+z)^3 + 1 - \frac{\Omega_c h^2 + \Omega_b h^2 + \Omega_\nu h^2}{h^2}} \equiv 100E(z) \;\; [\text{h km/s/Mpc}] \tag{5}$$

We then take into account the variation of $\omega_c = \Omega_{c,0}h^2$ and $\omega_b = \Omega_{b,0}h^2$. The derivatives of the logarithm of Hubble parameter with respect to ω_c, ω_b and h are:

$$\frac{\partial \ln H}{\partial \omega_c}|_{\omega_b,h} = \frac{(1+z)^3 - 1}{2h^2E^2(z)} \quad (6)$$

$$\frac{\partial \ln H}{\partial \omega_b}|_{\omega_c,h} = \frac{(1+z)^3 - 1}{2h^2E^2(z)} \quad (7)$$

$$\frac{\partial \ln H}{\partial h}|_{\omega_b,\omega_c} = \frac{1}{hE^2(z)} - \frac{1}{h} \quad (8)$$

and the other derivatives of the Hubble parameters with respect to n_s, A_s and τ are clearly zero. Note that had we chosen Ω_i as variables instead of ω_i, the derivative of the Hubble function with respect to h in units of [h km/s/Mpc] would have been zero. The angular diameter distance is

$$D_A(z) = \frac{1}{H_0}\frac{c}{1+z}\int_0^z \frac{dx}{E(x)} \quad (9)$$

where c is the speed of light and it is important to have distances in Mpc/h and

$$H_0 = \frac{100\,h\,\mathrm{Km/s/Mpc}}{c} = \frac{1}{2997.92}\frac{h}{\mathrm{Mpc}}\,. \quad (10)$$

The derivatives of the angular diameter distance are:

$$\frac{\partial \ln D_A}{\partial \omega_c} = \frac{1}{D_A(z)}\frac{c}{1+z}\int_0^z -\frac{\partial \ln H(x)}{\partial \omega_c}\frac{dx}{H(x)} \quad (11)$$

$$\frac{\partial \ln D_A}{\partial \omega_b} = \frac{1}{D_A(z)}\frac{c}{1+z}\int_0^z -\frac{\partial \ln H(x)}{\partial \omega_b}\frac{dx}{H(x)} \quad (12)$$

$$\frac{\partial \ln D_A}{\partial h} = \frac{1}{D_A(z)}\frac{c}{1+z}\int_0^z -\frac{\partial \ln H(x)}{\partial h}\frac{dx}{H(x)} \quad (13)$$

All these derivatives will be used in the Fisher matrix. Galaxy density: take column 4 of Table (1), (dn3) as dn/dz/$d\Omega$ (with Ω in square degrees). There is no need to add in an extra redshift success rate (e.g. epsilon=0.35) as before - this is already included in the numbers, which now account for redshift- and flux-dependent success rate, in addition to using an improved mock catalogues and redshift-measurement technique. There is also no need to include any scaling. These numbers are for the standard integration time (540 sec per dither, standard dither pattern). As an example, to get the total number of galaxies in a bin of width $0.95 < z < 1.05$, one would calculate:

(Number of galaxies $0.95 < z < 1.05$) = $4825.80 * 0.1 * 15000 = 7,239,000$

So in a Euclid survey over $15,000deg^2$ we would get good redshifts for $7,239,000$ galaxies in the redshift range $0.95 < z < 1.05$. Note the 0.1 factor required to turn dn/dz from a derivative to a number in a bin of width $Dz = 0.1$.

This gives you the total number of galaxies. Then the number density that you need in the fisher for GC, n in eq.2 in the formula for the fisher matrix, is the density per volume in each bin, therefore equal to:

$$n = dn3/dV_{dz} * dz \quad (14)$$

where the dz is the width of the redshift shell (i.e 0.1 but see the file) and

$$dV_{dz} = \frac{\frac{4\pi}{3}\left[(1+z_2)^3 D_A[z_2;\theta]^3 - (1+z_1)^3 D_A[z_1;\theta]^3\right]}{4\pi\left(\frac{180}{\pi}\right)^2} \quad (15)$$

is the volume in the redshift shell per degree squared. The survey Volume is

$$V_s = \text{Area(i.e. } 15,000 \text{ sq deg)} * dV_{dz} \quad (16)$$

The third column (dVol) should not be used anywhere.

Table 1: Specifications of the redbook, (derived from Orsi et al. 2009).

zmin	zmax	dVol	dn3	b3
0.35	0.45	3.60	0.000	0.992
0.45	0.55	5.10	0.000	1.024
0.55	0.65	6.40	0.000	1.053
0.65	0.75	7.90	2434.280	1.083
0.75	0.85	9.20	4364.812	1.125
0.85	0.95	10.30	4728.559	1.104
0.95	1.05	11.70	4825.798	1.126
1.05	1.15	12.30	4728.797	1.208
1.15	1.25	13.30	4507.625	1.243
1.25	1.35	14.00	4269.851	1.282
1.35	1.45	14.60	3720.657	1.292
1.45	1.55	15.10	3104.309	1.363
1.55	1.65	15.60	2308.975	1.497
1.65	1.75	16.00	1541.831	1.486
1.75	1.85	16.20	1474.707	1.491
1.85	1.95	16.50	893.716	1.573
1.95	2.05	16.70	497.613	1.568

GC comparison: where to start

In Case 1,2,3 the following common options are fixed:

1. k_{min} : 0.001 h/Mpc
2. $k_{max} = 0.2$ h/Mpc
3. k-binning: use directly the one of the spectra provided in the input
4. No relativistic corrections.
5. Area : 15,000 sq deg
6. Redshift range : $0.7 < z < 2.0$
7. Binning: use all bins as in the table, starting from 0.65 (the first non-zero in dn3).

GC case 1: minimal setting.

In addition to the common settings above, use these settings for your first comparison.

1. Bias: fixed to the fiducial value (column 5, b3 in the table 1). All derivatives wrt the bias are zero.
2. The AP effect is zero: the four derivatives of k and μ with respect to H and D_A are set to zero.
3. $P_{shot} = 0$ (fixed)
4. $\sigma_z = 0$

The Fisher matrix to compare with for this case, is case1.dat. If it matches, proceed to case2.

GC case 2: including the bias

In addition to the common settings above, case2 fixes:

1. Bias: marginalization over bias, derivatives as in 4c.
2. The AP effect is zero: the four derivatives of k and μ with respect to H and D_A are set to zero.
3. $P_{shot} = 0$ (fixed)
4. $\sigma_z = 0$

The Fisher matrix to compare with for this case, is case2.dat.
If it matches, proceed to case3.

GC case 3: including the AP effect

In addition to the common settings above, case3 fixes:

1. Bias: marginalization over bias, derivatives as in 4c.
2. The AP effect is included.
3. $P_{shot} = 0$ (fixed)
4. $\sigma_z = 0$

The Fisher matrix to compare with for this case, is case3.dat.
If it matches, proceed to case4.

GC case 4: including the shot noise

In addition to the common settings above, case4 fixes:

1. Bias: marginalization over bias, derivatives as in 4c.
2. The AP effect is included.
3. P_{shot}: marginalisation over P_{shot}; the fiducial is still zero; derivatives are calculated according to 4a
4. $\sigma_z = 0$

GC case 5: Constraining $\ln D_A(z)$, $\ln H(z)$, $f\sigma_8(z)$, $b\sigma_8(z)$, $P_s(z)$ at each redshift bin

In this case, we want to constrain some cosmological functions that depend on redshift, especially $f\sigma_8(z)$, $b\sigma_8(z)$. For doing this, we rewrite eqn. 1 as:

$$P_s(z) + \frac{D_A(z)^2_{\text{ref}} H(z)}{D_A(z)^2 H_{\text{ref}}(z)} \left(b\sigma_8(z) + f\sigma_8(z)\mu^2\right)^2 \frac{P(k,z)}{\sigma_8^2(z)} \tag{17}$$

The full list of parameters is the following:

- n cosmological parameters, in our case $n = 5$: $\theta_i = \{\Omega_c h^2,\, \Omega_b h^2,\, h,\, n_s,\, \ln 10^{10} A_s\}$
- $5 \times n_{bin}$ redshift-dependent parameters: $\{\ln D_A(z),\, \ln H(z),\, f\sigma_8(z),\, b\sigma_8(z),\, P_s(z)\}$

Now we will have small modifications on the derivatives:

$$\frac{\partial \ln P_{obs}\left(\bar{z},k,\mu;\theta_i\right)}{\partial \ln P_s(\bar{z})} = \frac{1}{P_{obs}\left(\bar{z},k,\mu;\theta_i\right)} \tag{18a}$$

$$\frac{\partial \ln P_{obs}\left(\bar{z},k,\mu;\theta_i\right)}{\partial f\sigma_8(\bar{z},k)} = \frac{2\mu^2}{b\sigma_8(\bar{z})+\mu^2 f\sigma_8(\bar{z},k)} \tag{18b}$$

$$\frac{\partial \ln P_{obs}\left(\bar{z},k,\mu;\theta_i\right)}{\partial b\sigma_8(\bar{z})} = \frac{2}{b\sigma_8(\bar{z})+\mu^2 f\sigma_8(\bar{z},k)} \tag{18c}$$

$$\frac{\partial \ln P_{obs}\left(\bar{z},k,\mu;\theta_i\right)}{\partial \ln H(\bar{z})} = 1+\mu^2\left(1-\mu^2\right)\frac{f\sigma_8(\bar{z},k)}{b\sigma_8(\bar{z})+\mu^2 f\sigma_8(\bar{z},k)} + k\mu^2\frac{\partial \ln P(k,z)}{\partial k} \tag{18d}$$

$$\frac{\partial \ln P_{obs}\left(\bar{z},k,\mu;\theta_i\right)}{\partial \ln D_A(\bar{z})} = -2+4\mu^2\left(1-\mu^2\right)\frac{f\sigma_8(\bar{z},k)}{b\sigma_8(\bar{z})+\mu^2 f\sigma_8(\bar{z},k)} - k(1-\mu^2)\frac{\partial \ln P(k,z)}{\partial k} \tag{18e}$$

$$\frac{\partial \ln P_{obs}\left(\bar{z},k,\mu;\theta_i\right)}{\partial \theta_i} = \frac{\partial \ln P(k,z)}{\partial \theta_i} - \frac{2}{\sigma_8(\bar{z})}\frac{\partial \sigma_8(\bar{z})}{\partial \theta_i}. \tag{18f}$$

The right hand side of eqn.18f is zero, when $\theta_i = \ln 10^{10} A_s$, so we should not add this parameter when constraining at the same time $\sigma_8(z)$. A Fisher matrix for comparison, will be added once this detail is solved.

9 Splinter Group 1: numerical codes

Coordinators

- Valeria Pettorino, Matteo Martinelli

Participants

- Matteo Costanzi, Yabebal Fantaye, Ken Ganga, Martin Kunz, Massimiliano Lattanzi, Julien Lesgourgues, Baojiu Li, Matteo Martinelli, Federico Marulli, Valeria Pettorino, Anna M. Porredon, Marco Raveri, Ziad Sakr, Valentina Salvatelli, Barbara Sartoris, Björn Malte Schaefer, Luca Amendola, Thejs Brinckmann, Mariana Penna-Lima.

Aim

Make a recommendation on existing codes for Atomistic Forecasting approach.
Expected output:

- A technical note explaining pros and cons of the various potential public codes for this purpose
- How do they meet our requirements:
  - on the division/requirements setting at fine-grained level
  - on individual probe work and on combinations
- Make a consensus of codes, and make a short presentation showing initial pros and cons.

9.1 Boltzmann codes

This section includes an overview of publicly available Boltzmann codes. For each code we present a schema in the same format, including pros and cons.

9.1.1 CAMB

(`http://camb.info/`)

Language: fortran90

Includes:

- standard neutrinos, sterile neutrinos
- CDM
- DE: w_0 w_a

Pros:

- updated very often
- tested by a large community
- there are several implementations for MG models (though not as well tested)
- python wrapper for CAMB to join to any MCMC
- documentation available and often updated: quick feedback by Anony Lewis and large community, via cosmocoffee

Cons:

- structure of the code requires to modify different parts of the code for different models
- no general distribution function for neutrinos
- only synchronous gauge
- maintainer (Antony Lewis) not currently in Euclid

9.1.2 EFTCAMB

(http://wwwhome.lorentz.leidenuniv.nl/~hu/codes/
https://github.com/EFTCAMB/EFTCAMB)

Language: fortran90

Includes:

- parametrized pure EFT approach to Horndeski (including the α parameterization), GLPV (?), Horava-Lifshitz
- exact (mapping) implementation of: designer f(R), Hu-Sawicki f(R), low-energy Horava
- several backgrounds and massive neutrinos

Pros:

- based on CAMB
- very general parametrization of the effective field theory (EFT) parametrisation of MG and DE models
- tested against HiCLASS/MGCAMB/private codes for some models
- several parameterisations of the background expansion and also of the perturbations (including Horndeski and beyond (e.g. GLPV))
- perturbation evolution computed self-consistently for EFT class of models
- Very clearly written, structured, documented.
- It has a MCMC wrapper.
- maintainers are active Euclid members

Cons:

- need further testing
- needs further development and understanding of the stability conditions
- MG starts from z = 100 (not tested earlier)

9.1.3 MGCAMB

(http://aliojjati.github.io/MGCAMB/)

Language: fortran90

Includes:

- Parametrization of linear scalar perturbations in DE/MG based on two functions of time and scale (μ, γ (or Q,R).
- LCDM and general w(z) (To be checked)
- massive neutrino
- Linder gamma parametrisation of the growth rate
- can treat the following specific models in the quasi-static approximation: BZ parametrization of f(R), (beta, m) parametrisation of Generalized Brans-Dicke, Hu-Sawicki f(R), Generalized Brans-Dicke (including symmetron and dilaton models)

Pros:

- it works with some well tested parametrization; easy to compare with previous results
- can test very general deviations from LCDM visible in the scalar metric perturbations
- It has MCMC wrapper

Cons:

- When used as a EFT replacement:
  - does not evolve all perturbation equations
  - assumes quasi static limit
- General deviations in scalar metric perturbations require careful parameterisation, some further developments (eg. Gaussian process) may be required
- Does not include similar deviations in vector and tensor perturbations
- Works with ΛCDM background

Warning: the code was recently updated (late february). These comments do not refer to the latest version.

9.1.4 CLASS

(http://class-code.net/)

Language: C
Includes:

- various forms of neutrinos and dark matter
- DE with a given equation of state, quintessence with scalar potentials.
- DE extension is HiClass but not public (Bellini, Miguel Zuma)
- newtonian and synchronous gauge

Pros:

- structure of code
- python wrapper
- two gauges (newtonian and synchronous: except the initial conditions which are only written in the synchronous gauge and gauge-transformed).
- maintainer (Julien Lesgourgues) is active Euclid member

Cons:

- community that uses/tested CLASS is smaller than CAMB

9.2 Parameter estimation codes

This subsection includes an overview of publicly available codes to calculate the likelihood.

9.2.1 Wish list definition

The basic requirement for codes which obtain cosmological predictions from Boltzmann codes and perform parameter estimation is "modularity".
A modular code is easy to modify in case some changes and updates need to be included, e.g. additional likelihoods, extra parameters, correction to observables (systematics).
Furthermore, separate modules for different physical quantities are needed for the atomistic approach as they allow to easily retrieve and compute the intermediate quantities of the theory flows (see e.g. Fig. 3) and to compare them between different codes.
However too much modularity may affect code performances.

The wish list of modules is:

1. module calling the Boltzmann/transfer function codes for primary science
2. module calling the sampler
3. module for each likelihood for each probe
4. module for background quantities
5. module for perturbation quantities
6. module for non-linear corrections (ex. halofit)
7. module that gets the standard output from Boltzmann codes and produces list of suitable output
8. postprocessing of the samples
9. plotting modules
10. to be discussed: module for cross correlation of probes

In the following, the publicly available codes are listed, together with the requirements which are satisfied by each of them. The missed requirements are also listed and the complexity of implementation is indicated as easy (E), medium (M) and difficult (D).

9.2.2 COSMOMC

(http://cosmologist.info/cosmomc/)

Language: fortran90

Includes:

- SN: JLA, Union
- CMB+CMB Lensing: Planck, WMAP, BICEP2+Keck
- LSS: WiggleZ, SDSS LRG, CFHTLenS
- BAO+RSD: several (noticeably BOSS)
- HST H0 measurements
- SZ CLUSTER: Planck

Pros:

- updated constantly
- very well tested, very efficient for standard cosmology
- used by a large community
- easy to add parameters
- easy to add an additional likelihood by adding a separate module
- nice python graphical interactive interface for first tests
- well documented
- there are many options and more control to analyse the samples.
- Updated to all currently available data (WL, GC, clusters, CMB).
- Likelihood Sampler works really well

Cons:

- if you increase the number of parameters, the sampler gets slower. Other samplers are publicly available and already interfaced with COSMOMC. However they are not included in the standard version (interface needs to be modified when COSMOMC is updated)
- maintainer (Antony Lewis) not currently in Euclid

Wish-list requirements:

- included: 1, 2 (add samplers), 3, 6, 8, 9
- missing: 4 (E CAMB has background functions collected in module ModelParams in modules.f90 which can be called by COSMOMC), 5 (D), 6 (alternatives to halofit), 7 (E/M)

9.2.3 MONTEPYTHON

(https://github.com/baudren/montepython_public or http://baudren.github.io/montepython.html)

Language: python

Includes:

- Euclid mock likelihood (weak lensing + galaxies) is publicly available
- CMB, SNIa, BAO, WiggleZ (CHFTLens and SDSS to be released);

Pros:

- modular environment
- can switch among different samplers
- it has a graphical interface
- can easily be interfaced with different Boltzmann codes
- can use MPI
- extra parameters need to be included only in the input file
- easy to add a new likelihood.

Cons:

- community that uses/tested MP is smaller than COSMOMC
- less frequently updated

Wish-list requirements:

- included: 1 (for generic Boltzmann code; currently points at Class wrapper; Camb wrapper in preparation), 2 (MH, Multinest, CosmoHammer, importance sampling, reprocessing chains, fisher matrices), 3, 4, 5, 6 (Halofit), 7, 8, 9 (can choose between MontePython or Getdist to produce plots or reprocess: chains are in the same format as cosmomc and can use 8,9 from COSMOMC)
- missing: 6 (alternatives to Halofit E/M)

9.2.4 COSMOSIS

(https://bitbucket.org/joezuntz/cosmosis/wiki/Home)

Language: Written in C++ and python. Modules can be written in C, C++, Fortran, or Python.

Includes:

- the models discussed here: https://bitbucket.org/joezuntz/cosmosis/wiki/default_modules
- Different samplers: including emcee (see schema discussed in this document), fisher matrices sampler and Multinest (that computes the Bayesian evidence and the posterior pdf's.) http://arxiv.org/pdf/1409.3409v1.pdf
- Likelihoods are listed here: https://bitbucket.org/joezuntz/cosmosis/wiki/default_modules in the likelihood section.
- CAMB, CLASS

Pros:

- can be wrapped to CAMB and CLASS
- it is modular (can introduce other samplers or Boltzmann codes or likelihoods)
- can add modules in different languages
- new parameters are defined only in the input parameter file
- They will include SDSS. Easy to implement a new module for a new likelihood.
- Used by DES.
- Issues are usually rapidly answered in the issues tab of the bitbucket repository.
- Postprocess tool that automatically makes python plots of the sampler outputs (some demos: https://bitbucket.org/joezuntz/cosmosis/wiki/Home)

Cons:

- not very well tested
- small community uses it
- Not clear how often it is updated (Responsible is: Joe Zuntz et al) Is he in Euclid?
- Quality of demos visualisation seems lower than available for getdist python plotting.

Wish-list requirements:

- included: 1 (generic Boltzmann code: CAMB, CLASS), 2 (MH, MultiNest, EMCEE), 3 (...add), 4 (?), 6 (Halofit, Halofit + T correction), 7
- Datablock the object passed down the pipeline. For a given set of parameters all module inputs are read from the datablock and all module outputs are written to it
- missing: 4 (E but not implemented?), 5 (?), 6 (alternatives to Halofit; E/M), 8 (format of chains? can one use getdist? how advanced as compared to getdist?), 9 (how advanced with respect to COSMOMC ones?)

9.2.5 NumCosmo

(http://www.nongnu.org/numcosmo/)
(https://github.com/NumCosmo/NumCosmo)

Language: C (main), Python, Perl, Java, Java Script, Vala, Lua, Ruby (for a complete list see https://wiki.gnome.org/Projects/GObjectIntrospection/Users)

Includes:

- some BAO, some H(z) , SNe Ia JLA, Planck, some cluster data
- resample codes (from a probability density function and bootstrap)
- sampler codes

Pros:

- modular environment
- can switch among different samplers
- MCMC object is modular and one can implement just a new transition kernel
- can easily be interfaced with different Boltzmann codes
- the code is parallelized through multithread and easily extended to MPI
- all parameter information is defined in the NcmModel. It is inherited by NcmMSet, NcmFit, NcmFitMCMC etc, avoiding repetition
- easy to add a new likelihood
- it call be called from any bind language or from a general purpose executable (in C) called darkenergy
- frequently updated
- portable to Linux, Mac OS and Window
- it uses autotools as its building system allowing NumCosmo to be easily compiled in several architectures
- a list of pre-compiled packages can be found at https://build.opensuse.org/package/show/home:vitenti/numcosmo
- the QA is assured by a set of tests using ”Unit testing”

Cons:

- community that uses/tested NumCosmo is much smaller than COSMOMC
- external dependencies (NLopt, Sundials, cfitsio, etc)
- users can call NumCosmo from any bind language, but the developers have to learn GObject and GObject Introspection. We plan to provide a cookbook to make it easier.
- no graphical interface

Wish-list requirements:

- included: 1 (CLASS wrapper, generic Boltzmann code in progress); 2 (statistical tools available: best-fit finders, Fisher Matrix, Monte Carlo (resample from a probability density function and bootstrap), Profile Likelihood, Markov Chain Monte Carlo (Metropolis-Hastings, Ensemble sampler MCMC), Approximate Bayesian Computation); 3; 4; 5; 6 (halofit, halo model); 7; 8; 9.

9.2.6 emcee

(`http://dan.iel.fm/emcee/current/`)

Language: python

Includes:

- 3 different samplers (unified calling interface) and plotting tools

Pros:

- affine invariant sampling for fast sampling of degenerate likelihoods
- parallel tempered sampler for multimodal likelihoods
- standard Metropolis-Hastings sampler
- many plotting tools included
- can use MPI
- interfaces reasonbly easy to write
- interface to lensing code exists (in C)
- interface to supernova code exists (in C or python)

Cons:

- not specifically interfaced to a Boltzmann code

Wish-list requirements:

- included: 2,8,9
- provided: 1,7 (wrapper for CLASS (working on it now)), 2,3 (lensing, supernovae and CMB spectra), 4, 5, 6 (work in progress)
- missing: 3 (iSW-effect, BAO, galaxy density)

9.3 Recommendation on codes and coding languages to use

9.3.1 Coding Language

The SGS use python and C++ only, and it would be good for us to follow them to have some coherence. Although there are pros and cons.
It would be ideal to have a python infrastructure: wrappers in python to other codes written in different open sources free languages (instead of IDL, Mathematica)
FR software to convert languages from one to another: help to write a python wrapper, for example SDC is using swig (http://www.swig.org/). Good experience with SWIG, it serves as a way of passing data between python and other languages (Bjoern: tried out python→C with openMP-parallelisation)
Make sure all libraries used/needed are available. SDC already has a list of libraries that are allowed to be used, at `http://euclid.roe.ac.uk/projects/codeen-users/wiki/EDEN`. Python slow for high level performances.

9.3.2 Boltzmann code

No specific choice, but recommendation is that we need at least 2 codes that match for each model (match in the sense explained in the methodology below); in addition, our recommendation is to use MGCAMB only for tests (and clearly specify limitations of the code and assumptions in it).

9.3.3 Parameter estimation and likelihood code map

A summary of how the likelihood codes compare to the wish list is visualised in Fig.4. MontePython is at present the code that includes the highest number of requirements in the wish list. JL & team within Euclid guarantee fast update for wrappers to Boltzmann code version updates and to new data which become available.
To be discussed: how the different codes perform on cross correlation of probes.

9.3.4 Analysis and visualization

Getdist and plotting tools provided by cosmomc.

9.4 Methodology to test a code

9.4.1 How to trust a Boltzmann code in Modified Gravity

For every MG model, there should be at least 2 Boltzmann codes, agreeing. Agreement is defined as follows:

- check them on LCDM, need to agree at less than 0.1%
- check them in a MG scenario in different ranges of parameters: they should match at the same level at which they agree in LCDM on all background quantities, all perturbation quantities, all spectra

9.4.2 How to trust a likelihood code

The code needs to pass a self-consistency check: use a simulation to produce a set of data and check if you can recover the initial cosmological model.
Available simulations:
galaxy mock catalogue

1. from MICE simulations (HOD) in LCDM cosmology `http://internal.euclid-ec.org/?page_id=1551`
2. from Durham simulations (SAM) in LCDM cosmology

Needed:

- simulations from non-standard cosmological models: e.g. f(R), DE
- simulations with neutrinos
- mock catalogue for WL (to be asked to the WL splinter-2)

9.5 Required output from other groups

- SDC: list of python libraries (`http://euclid.roe.ac.uk/projects/codeen-users/wiki/EDEN`) Does not seem to contain a list of python libraries
- From all other splinters: check that the wish-list satisfies their list of requirements. List specific output required by the codes that satisfies theory/observations?
- TWG:
  - parameter definition document
  - list of Boltzmann codes available for MG
- Simulations: list of simulations available

Figure 4: Likelihood Code Map. Fields correspond to the modules in the wish list. Rows correspond to the numerical codes available and discussed in the text. Easy/Medium?Difficult indicate the estimated level of difficulty to implement the missing modules.

10 Splinter Group 2: Weak Lensing Holistic Forecast Definitions

10.1 Lensing projections

Starting from the Born approximation of the convergence, the exact equation for the convergence power spectrum with (correlated) sources in redshift bins i and j for a flat universe is

$$P_{ij}(\ell) = \frac{9}{4}\Omega_{\rm m}^2 \left(\frac{H_0}{c}\right)^4 \int_0^{\chi_{\rm lim}} \mathrm{d}\chi W_i(\chi) \int_0^{\chi'_{\rm lim}} \mathrm{d}\chi' W_i(\chi') \int_0^{\infty} \mathrm{d}k\, k^2\, P_{\rm m}(k,\chi,\chi')\, \mathrm{j}_\ell(k\chi)\, \mathrm{j}_\ell(k\chi'), \tag{19}$$

where j_ℓ is the spherical Bessel function of order ℓ, $W_i(\chi)$ is the window function or *lensing efficiency*. and χ the comoving distance. In the Limber approximation, this equation simplifies to

$$P_{ij}(\ell) = \frac{9}{4}\Omega_{\rm m}^2 \left(\frac{H_0}{c}\right)^4 \int_0^{\chi_{\rm lim}} \frac{\mathrm{d}\chi}{\chi^2} W_i(\chi) W_j(\chi) P_{\rm m}\left(k = \frac{\ell + 1/2}{f_K(\chi)}, \chi\right), \tag{20}$$

This latter equation is the case of general curvature K, with f_K the comoving angular diameter distance. Note that often the relation between 3D and 2D modes $k = \ell / f_K(\chi)$ is used, which leads however to less accurate results. See Sect. 10.7 for a discussion on the various assumptions and approximations under which this equation is derived. The lensing efficiency is given as

$$W_i(\chi) = \int_\chi^{\chi_{\rm lim}} \mathrm{d}\chi'\, n_i(\chi') \frac{f_K(\chi' - \chi)}{f_K(\chi')}. \tag{21}$$

The redshift distribution for bin i is denoted with n_i, and it is normalized to the number of galaxies belonging to that bin,

$$\int_0^{\chi_{\rm lim}} \mathrm{d}\chi\, n_i(\chi) = \int_0^{z_{\rm lim}} \mathrm{d}z\, n_i(z) = 1. \tag{22}$$

10.2 Redshift distribution

The window function depends on $n_i(z)$, the galaxy distribution in the i-th redshift bin: this is convolved with a Gaussian to account for photometric redshift errors σ_z (value specified below), i.e.

$$n_i(z) = A \int_{i\text{-th bin}} n(z') \exp\left(\frac{-(z\prime - z)^2}{2\sigma_z^2}\right) \mathrm{d}z' \tag{23}$$

where the integral is done over z' for the single i-th bin. $A = \frac{1}{\sqrt{2\pi}\sigma_z}$ a normalization factor.

10.2.1 n(z)

In eq. (23), we use the number density

$$n(z) = a_1 \exp\left[-\frac{(z - 0.7)^2}{b_1^2}\right] + c_1 \exp\left[-\frac{(z - 1.2)^2}{d_1^2}\right] \tag{24}$$

where $(a_1, b_1, c_1, d_1) = (1.50, 0.32, 0.20, 0.46)$. This distribution is taken from van Waerbeke et al. (2013) and is the number density as a function of redshift for the CFHTLS survey. This is the most applicable $n(z)$ for Euclid as CFHTLS covers each of the ground based broad-band filters to the depth required for Euclid ground-based photometry.

[Tom K: Need to specify the exact binning in redshift for this functional form.]

10.2.2 p(z)

The photometric redshift behaviour needs to be able to model the three top-level requirements for this posterior redshift estimates. These are: the scatter in the photometric redshifts (variance of the $p(z)$), the fraction of outliers, and any residual bias in the phototmetric redshifts. These three features can be captured in a double-Gaussian distribution where the functional form is a sum of the main sample and the outying sample

$$p(z|z_p) = \frac{1-f_{\rm out}}{\sqrt{2\pi}\sigma_z(z_p)} \exp\left[-(z - c_{\rm cal}z_p + z_{\rm bias})^2/2\sigma_z^2(z_p)\right] + \frac{f_{\rm out}}{\sqrt{2\pi}\sigma_z^O(z_p)} \exp\left[-(z - c^O_{\rm cal}z_p + z^O_{\rm bias})^2/2[\sigma_z^O(z_p)]^2\right], \tag{25}$$

Here we have assumed that the photo-z distribution is calibrated, but imperfectly, so that the median spectroscopic redshift distribution is biased and inclined so that it lies along a line $z_s = c_{\rm cal}z_p + z_{\rm bias}$, where $z_{\rm bias}$ is some bias and $c_{\rm cal}$ is a calibration. A value $c_{\rm cal} = 1$ would mean that the photometric redshift estimation is perfectly calibrated to a spectroscopic sample. The redshift error $\sigma_z^O(z)$ is assumed to be unknown and the distribution is also assumed to lie between some photometric redshift range $z_{\rm range}$. We also assume a fraction $f_{\rm out}$ of outlying galaxies in the sample, inclined on a slope described by $z^O_{\rm bias}$ and $c^O_{\rm cal}$. The outlying sample's redshift error is range also unknown $z^O(z)$, which is the range in photometric redshift over which the outliers subtend. Note that we do not include outliers that have low spectroscopic redshifts but a broad range in estimated photometric redshift. Our analysis may be pessimistic in this case since by including such a sample some redshift biasing effects may cancel-out in this analytic approximation.

[Tom K: Contacting OUPHZ for current best estimates.]

10.3 Covariance matrix

To write down the covariance matrix, we first define the observed power spectrum [**?**] as

$$P^{\rm obs}_{ij}(\ell) = P_{ij}(\ell) + N_{ij}(\ell), \tag{26}$$

where the tomographic power spectrum P_{ij} is given in eq. (20). The shot noise spectrum $N_{ij}(\ell)$ contaminating $P_{ij}(\ell)$ is given by the diagonal matrix

$$N_{ij}(\ell) = \delta_{ij} \left\langle \gamma^2_{\rm int} \right\rangle {q_i}^{-1} \tag{27}$$

where $\left\langle \gamma^2_{\rm int} \right\rangle$ is the intrinsic galaxy shear dispersion per component and

$$q_j = n_\theta = \frac{n_\theta}{\rm arcmin^2} 3600 \left(\frac{180}{\pi}\right)^2. \tag{28}$$

Here n_θ is the galaxy density and $n_j(z)$ is the galaxy density for the j-th redshift bin, as defined in eq. 10.2). Note that n_j is normalized for every j (eq. 22).

The Gaussian component of the covariance of the observed power spectrum for a shell width width $\Delta\ell$ is then according to [**?**, **?**]

$$C[P_{ij}(\ell), P_{mn}(\ell')] = \frac{2\delta_{\ell\ell'}}{(2\ell+1)\Delta\ell f_{\rm sky}} \left[P^{\rm obs}_{im}(\ell)P^{\rm obs}_{jn}(\ell) + P^{\rm obs}_{in}(\ell)P^{\rm obs}_{jm}(\ell)\right], \tag{29}$$

where $f_{\rm sky} = A/(4\pi)$ is the fraction of sky observed by the survey with area A. [Martin K: Prefactors seem to be different in Joachimi and Takada].

The non-Gaussian contribution consists of integrals over the convergence trispectrum. A corresponding expression can be found in [**?**] and [**?**]. The trispectrum has been approximated by various authors using the halo model. [**?**] only take into consideration the one-halo term and the perturbation-theory term. The former is the dominant contribution in the highly non-linear regime on sub-halo scales. The latter appxorimates the contribution from different halos (2-, 3-, 4-halo terms) on intermediate scales. On large scales the non-Gaussian errors are subdominant compared to the Gaussian errors. All four halo terms were computed in [**?**], and compared to N-body simulations.

[?] compared the Gaussian contribution to N-body simulation results. Fitting formulae for the non-Gaussian cosmic variance term have been provided by [?].

The above mentioned expressions for the non-Gaussian covariance as integrals over the trispectrum do not include couplings of small-scale modes with long wavelength modes that are larger than the observed survey volume. These super-survey modes were first introduced as beat coupling in [?], and also discussed in [?]. They were later modeled in the halo model framework as halo sample variance (HSV; [?], [?] Contrary to the other terms of the covariance that scale inversely with the survey area fsky, this super-survey covariance (SSC) decreases faster. Therefore it is important for small survey areas [?]. A rigorous treatment of the non-Gaussian covariance including the SSC is presented in [?]. SSC in numerical simulations were studied in detail in [?].

[?] only account for the 1-halo term and HSV for the covariance. [?] compute the full covariance in the halo model accounting for all four halo terms and HSV.

[Martin K: Todo: look for analytical approaches (renormalization approach)]

Since we consider multipoles up to $\ell_{\max} = 5000$, we need to apply non-linear corrections to the matter power spectrum, for which we use the halofit (provided in the input spectra).

Then the Fisher matrix is summed over all multiples:

$$F_{\alpha\beta} = f_{\rm sky} \sum_{\ell,i,j,k,m} \frac{(2\ell+1)\Delta\ell}{2} \frac{\partial P_{ij}(\ell)}{\partial\theta_\alpha} C^{-1}_{jk} \frac{\partial P_{km}(\ell)}{\partial\theta_\beta} C^{-1}_{mi}. \tag{30}$$

[Martin K: This term of the Fisher matrix is the one used in Hu 1999, using 20 diretcly as covariance. Using 29 the other Fisher-matrix term is used in Takada&Jain. Check whether both are consistent.]

10.4 Likelihood function

For an ideal full-sky experiment, the likelihood of the observed spectra given the theoretical spectra reads

$$\mathcal{L} = \mathcal{N}\Pi_{\ell,m} \left\{ \frac{1}{(d^{\rm th}_\ell)^{1/2}} \exp\left[-\frac{1}{2} a^{\rm obs\dagger}_{\ell m} (\tilde{C}^{\rm th}_l)^{-1} a^{\rm obs}_{\ell m} \right] \right\} , \tag{31}$$

where $\mathcal{N}$ is a normalization factor. The likelihood simplifies to

$$\mathcal{L} = \mathcal{N}\Pi_{\ell,m} \left\{ \frac{1}{(d^{\rm th}_\ell)^{1/2}} \exp\left[-\frac{1}{2} (2\ell+1) \frac{d^{\rm mix}_\ell}{d^{\rm th}_\ell} \right] \right\} , \tag{32}$$

where

$$d^{\rm th}_\ell = \det\left[C^{\rm th}_{ij}(\ell) \right] \tag{33}$$

$$d^{\rm obs}_\ell = \det\left[C^{\rm obs}_{ij}(\ell) \right] \tag{34}$$

and $d^{\rm mix}_\ell$ is a quantity obtained by replacing one after each other the theoretical spectra $C^{\rm th}_{ij}(\ell)$ by the observed ones $C^{\rm th}_{ij}(\ell)$ (see [?] for details and an exemple).

By using a first-order approximation to take into account the limited sky coverage, the χ^2 relative to the best-fit model is given by [?]

$$\Delta\chi^2_{\rm eff} \equiv -2\ln\frac{\mathcal{L}}{\mathcal{L}_{\max}} \equiv \sum_\ell (2\ell+1) f_{\rm sky} \left(\frac{d^{\rm mix}_\ell}{d^{\rm th}_\ell} + \ln\frac{d^{\rm th}_\ell}{d^{\rm obs}_\ell} - \mathcal{N} \right) \tag{35}$$

which should be used in MCMC forecast codes.

10.5 Weak lensing code comparison

WL comparison: where to start

In all cases the following specifications and options are fixed:

1. Area: 15,000 sq deg
2. $f_{sky} = Area/(4\pi(180/\pi)^2)$
3. Median Redshift: $z_{mean} = 0.9$
4. $\gamma_{\rm int} = 0.22$
5. $n_\theta = 30$ galaxies per $\rm arcmin^2$
6. Error on photometric redshift: $\sigma_z = 0.05(1+z)$
7. Method to correct for non-linearities: Halo model
8. Binning: 10 bins from z = 0 - 2.5. We assume equipopulated bins: the redshift bins chosen such that each contain the same amount of galaxies. need to add formula
9. k-range: $k_{min} = 0.001$ h/Mpc, $k_{max} = 0.5$ h/Mpc
10. k-binning: use directly the one provided in the input
11. ℓ-range: $\ell_{min} = 5$, ℓ_{max} fixed as a function of z, using the table provided by Enea (uploaded in the wiki). No cut in z.
12. ℓ-binning: 100 bins (with log spaced binning)

WL Case 1: minimal setting

1. Linear power spectrum
2. $\sigma_z = 0.05(1+z)$
3. $\gamma_{\rm int} = 0.22$

WL Case 2: ground based $n(z)$

1. Linear power spectrum
2. Ground based $n(z)$ prescription
3. $\sigma_z = 0.05(1+z)$
4. $\gamma_{\rm int} = 0.22$

WL Case 3: simple non-linear corrections

1. Non-linear power spectrum from HaloFit, with massive neutrinos corrections if needed. Parameters: Eqs. A06-A14 of http://arxiv.org/abs/1208.2701
2. $\sigma_z = 0.05(1+z)$
3. $\gamma_{\rm int} = 0.22$
4. $n(z)$ either standard prescription or ground based (we should decide)

WL Case 4: global uncorrelated theoretical error

Same as Case 4, but add a nuisance parameter in the covariance matrix accounting for a global uncorrelated theoretical error, following the procedure of Ref. [**?**] (Appendix B4). Marginalize over this parameter.

In the case of massive neutrinos, add an extra neutrino-related error, as in Appendix B5 Ref. [**?**]

WL Case 5 and beyond: extended cases (to be considered in a second phase?)

1. Other analytical formulae for non-linear corrections, possibly including baryonic feedback
2. Method for intrinsic alignment
3. others

(to be further discussed)

10.6 Real-space forecasts

The numerical calculation of real-space quantities is straight-forward, accurate, and fast. All real-space functions can be represented as 1D integrals over the filtered convergence-power spectrum $P_\kappa = P_\kappa^{\mathrm{E}} + P_\kappa^{\mathrm{B}}$. For the 2PCF ξ_+ and ξ_-, these are efficiently carried out as Hankel transformation,

$$\begin{aligned} \xi_+(\theta) &= \frac{1}{2\pi} \int \mathrm{d}\ell \, \ell \mathrm{J}_0(\ell\theta)[P_\kappa^{\mathrm{E}}(\ell) + P_\kappa^{\mathrm{B}}(\ell)]; \\ \xi_-(\theta) &= \frac{1}{2\pi} \int \mathrm{d}\ell \, \ell \mathrm{J}_4(\ell\theta)[P_\kappa^{\mathrm{E}}(\ell) - P_\kappa^{\mathrm{B}}(\ell)]. \end{aligned} \tag{36}$$

For COSEBIs, the aperture-mass dispersion, and other real-space functions, the filter functions are smooth and have none to relatively few oscillations. The general expression for an E-/B-mode separating function $X_{\mathrm{E,B}}$ is

$$X_{\mathrm{E,B}} = \frac{1}{2\pi} \int_0^\infty \mathrm{d}\ell \, \ell \, P_\kappa^{\mathrm{E,B}}(\ell) \tilde{U}^2(\ell). \tag{37}$$

Fast and acurate codes exist to carry out the integrals in (36) and (37) (`FFTLog` [**?**]; `nicaea`, part of `CosmoPMC` [**?**]; non-public codes from Patrick Simon, Marika Asgari).

Some of these transformations are valid for a flat sky, and we need to check their accuracy for the curved sky case, and potentially modify those tranformations if necessary.

The covariance matrix for real-space functions often shows stronger off-diagonal parts than the power spectrum. The latter, in the ideal case of a Gaussian shear field on the full sky, has zero-off diagonal elements. However, the non-Gaussian LSS on small scales and mask effects introduce off-diagonal correlations. In fact, the relation between the observable (band-power) spectrum and the underlying true convergence power spectrum P_κ has the same form (37) as the relation between a real-space function and P_κ. The amplitude of off-diagonal elements depends mainly on the width of the filter function.

ξ_+ and ξ_- have a covariance with very large off-diagonal elements. The calculation of the cosmic variance term even in the Gaussian case is time-consuming due to the product of two Bessel function with different arguments $\ell\theta_i$, $\ell\theta_j$ for the off-diagonal element C_{ij} with $i \neq j$. A code exists (Benjamin Joachimi) that has been used within Euclid. Even with parallelization on $\mathcal{O}(50)$ CPUs, for $N_\theta = 50$ and $N_z = 12$ it takes a up to two weeks to compute the $N_\theta^2[N_z(N_z+1)]/2 \times 4$ covariance elements for ξ_+ and ξ_- including the mixed terms $(+-, -+)$. The corresponding expressions for this covariance can be found in [**?**].

10.7 Systematics: overview

Often for Fisher matrix forecast, we implicitly assume that the observed power spectra is an unbiased realization of the underlying theoretical one so that we can directly compare them to infer constraints on the cosmological parameters. Actually, systematics make this assumption less than obvious. Roughly speaking, we can divide systematics sources in three classes.

- *Astrophysical/theoretical systematics.* This includes all those lensing-related effects which are usually taken for granted and neglected in the standard analysis. These are listed in Sect. 10.8, with the exception of intrinsic aligment (IA). As one of the dominant contaminations of the lensing signal, IA is discussed in the separated Section 10.9.
- *Instrumental systematics.* Textbook example is CTI correction which introduces an artificial distortion pattern which impacts the ellipticity of each galaxy and hence the shear determination. As a result, a fake correlation can be induced leading to a biased observed power spectrum.
- *Observational systematics.* These are related to errors in the measurement process. The most well known case concerns shape measurement codes. In a first good approximation, one can assume that the measured ellipticity ϵ_{obs} is related to the true one ϵ as $\epsilon_{obs} = (1+m)\epsilon + c$, with (m, c) the multiplicative and additive bias respectively. Depending on the code, (m, c) can depend on the galaxy properties; e.g. [**?**] have used mock CFHTLenS data analysed with the *lensfit* code to show that, while c is negligible, m is a function of the galaxy size and S/N ratio. As a second example, one can consider colour gradient bias which can still be parameterized with the same formalism with m dependent on the source redshift and color as preliminarily found by [**?**].

The impact of systematics in the first class can be explicitely evaluated and their impact on the cosmic shear power spectrum quantified and included if necessary. On the contrary, second and third class systematics ask for a more empirical analysis. Under general assumptions, we can model their impact stating that the following relation hold

$$\mathcal{C}_{sys}(\ell, z_i, z_j) = (1 + \mathcal{M}_{ij})\mathcal{C}_{lens}(\ell, z_i, z_j) + \mathcal{A}(\ell, z_i, z_j)$$

where $\mathcal{C}_{sys}$ and $\mathcal{C}_{lens}$ are the cosmic shear power spectra with systematics included and the theoretical one, $\mathcal{M}_{ij}$ a redshift dependent multiplicative correction and $\mathcal{A}(\ell, z_i, z_j)$ is a scale and redshift dependent additive correction.

Forecasting codes should include the relevant astrophysical systematics as a first step. A parameterization for the multiplicative and additive bias terms should be worked out to be added as soon as second and third class systematics have been quantified in some way.

10.8 Systematics: astrophysical

Weak lensing probes the cosmic large-scale structure through its null-geodesics: While this property follows almost universally from all relativistic theories of gravity and is easy enough to understand, the technical computation of weak lensing observables, in particular on small scales, is challenging as it is influenced by a number of second-order effects. With Euclid, large multipoles are probed with high statistical precision such that the exact prediction of spectra on small scales is necessary in order to have unbiased measurements: Euclid's weak lensing data set will have a statistical significance of about $10^3\sigma$, such that for example an over- or underpredicition of the spectra by 10^{-3} will be a 1σ-bias.

Second order effects in weak lensing fall into a couple of categories, according to the physical mechanisms, but they have in common that the weak lensing signal is weighted by a second field, which, under some circumstances, can generate B-modes in the ellipticity field - with impact on the calibration of the ellipticity measurements. Typically, the effects are important on small scales around and above $\ell = 10^3$ and modify the spectra by a factor of $10^{-3\ldots-8}$ depending on the nature of the weighting field.

10.8.1 description of fluctuations

Several approximations are usually made, and should be examined, to describe density flucutations, their correlations, and their projections. The Limber approximation (eq. 20, see [?, ?]) assumes that the power spectrum is slowly varying compared to the spherical Bessel functions in (eq. 19). In this approximation, correlations of modes between different comoving distances are neglected. (This is also explicitely assumed when deriving the deflection angle by linearly summing up contributions from isolated thin lenses along the line of sight [?]. This is a very good approximation of the exact convergence power spectrum for small angles up to 10 degrees, or $\ell > 20$.

10.8.2 geodesic effects

The implict lens equation is solved by a perturbative expansion, where the weak lensing deflection is collected by integration along a straight line replacing the actual light path. At higher order, there are corrections due to better approximations of the actual light path (Born-corrections). Similarly, the change in shape of a light bundle is computed relative to a light bundle with circular cross section, and corrected at higher order by gravitational distortion of an already deformed bundle (lens-lens coupling).

10.8.3 clustering effects

The weak lensing effect assumes a uniform sampling of the tidal fields generated by the cosmic large-scale structure. The source galaxies, however, are clustered due to structure formation and introduce a weighting into the weak lensing signal which reflects their angular clustering (source-source clustering; [?]). In addition, there is a positive correlation between lensing structures and structures hosting lensed galaxies (source-lens clustering; [?]), again breaking the uniform sampling in the idealised picture.

10.8.4 relativistic effects

Weak lensing is computed in the limit of a weakly perturbed FLRW-metric with gradients in the potential being responsible for deflection and tidal shear fields for shape distortions of galaxies. There are, however, corrections to the metric of order Φ/c^2 as well as corrections due to the momentum density. In general, these effects are small but might be relevant looking for deviations from general relativity as the theory of gravity.

10.8.5 source motion and location

Lensing requires the conversion from the observed redshift of a source galaxy to comoving distance, and in this conversion one usually assumes that the redshift is purely cosmological. But there can be non-cosmological contributions to the redshift, for instance by peculiar motion, or by gravitational redshifting at the source itself (Sachs-Wolfe-type corrections) or by cumulative gravitational redshifting between the source and the observer (integrated Sachs-Wolfe-type corrections).

10.8.6 lensing-specific effects

The quantity relevant for mapping the ellipticity of a galaxy is the reduced shear $g = \gamma/(1-\kappa)$ instead of the lensing shear γ, implying corrections to the weak lensing shear spectra at second order. Furthermore, the survey depth and the number of observed galaxies is modulated by weak lensing magnification, both leading to a weighting of the shear signal with another lensing quantity (magnification bias; [?].

10.8.7 General remarks

As a general observation we emphasise that the corrections affect higher-order statistical measures, in particular the bispectrum, at a lower order in perturbation theory.

Useful references that compile, compare, and discuss various lensing-related systematics are [?] and [?].

10.9 Systematics: Intrinsic alignments

Weak lensing commonly operates under the assumption that ellipticities are intrinsically uncorrelated, and that weak lensing is the only effect that generates ellipticity correlations due to correlated tidal shear fields between light bundles. Galaxies can have intrinsically correlated shape due to their formation history or due to their interaction with the cosmic large-scale structure. While the exact mechanisms are not yet fully understood, there are two types of interaction based on tidal fields which are thought to be relevant for the alignment of spiral and elliptical galaxies.

10.9.1 linear alignment model for elliptical galaxies

The dynamical model of elliptical galaxies is that of a cloud of stars in virial equilibirum inside the hosting dark matter structure. The gravitational potential of this structure can be perturbed by tidal gravitational fields, which cause a change in shape of the stellar component.

10.9.2 quadratic alignment model for spiral galaxies

The shape of a spiral galaxy is determined by the inclination angle under which we see the stellar disc. One assumes that the symmetry axis of the stellar disc coincides with the angular momentum direction of the host halo, which in turn is determined in its formation process through tidal torquing.

10.10 Non-linear power spectrum and baryonic corrections

Weak lensing measures the projected 3D density distribution probing therefore the matter power spectrum $P(k)$ and the gravitational theory determining it. While the perturbation theory efficiently works at computing $P(k)$ in the linear regime, corrections are needed to take into account deviations on scales where the condition $\delta\rho/\rho << 1$ does not hold anymore. Moreover, on small scales, baryons collapse to form the visible structures we observe. Both nonlinearities and baryon collapse make the power spectrum $P(k)$ to differ from the linear one $P_{lin}(k)$. Different methods, tailored on N - body simulations, have been developed to phenomenologically model these corrections with the Halofit approach [**?**], in its updated version [**?**, **?**], being the most used one .

Recently, Mead et al. [**?**, **?**] developed a more accurate method based on a combination of the halo model approach [**?**, **?**, **?**] and tuning to power spectra from the COSMIC EMU simulations [**?**, **?**]. We summarize below the main formulae referring the reader to the original papers for more details. Rather than $P(k)$ itself, it is more convenient to estimate the equivalent quantity

$$\Delta^2(k) = 4\pi V \left(\frac{k}{2\pi}\right)^3 P(k)$$

representing the fractional contribution to the variance per logarithmic interval in k with V the periodic volume. Following the standard halo model approach, we write $\Delta^2(k)$ as the sum of the one halo and two halo terms. The first reads

$$\Delta^2_{1H}(k) = 4\pi \left(\frac{k}{2\pi}\right)^3 \frac{1}{\bar{\rho}^2} \int_0^\infty M^2 W^2(\nu^\eta k, M) n(M,z) dM \tag{38}$$

where $\bar{\rho}$ is the mean matter density. Let us detail the different quantities entering Eq.(38). First, we have the normalized Fourier transform of the halo density profile given by

$$W(k,M) = \frac{1}{M} \int_0^{R_v} \frac{\sin(kr)}{kr} 4\pi r^2 \rho_h(r,M) dr \tag{39}$$

where the mass M and the virial radius R_v are related as

$$R_v = \left[\frac{3M}{4\pi \Delta_v(z) \bar{\rho}} \right]^{1/3} \tag{40}$$

with $\Delta_v(z) = 418[\Omega_M(z)]^{-0.352}$. In order to compute $W(k, M)$, one needs to choose a model for the halo density profile. On the scales of interest, the NFW profile [?, ?] is the common choice with

$$\rho_h(r) = \frac{\rho_M}{(r/R_s)(1 + r/R_s)^2} \tag{41}$$

with ρ_M a normalization constant set so that the mass within R_v is the the virial mass M and the scale radius R_s given by $R_s = R_v/c_v$. The concentration c_v is related to the halo mass as

$$c_v = A_v \frac{1 + z_f(M)}{1 + z} \left[\frac{g(z \to \infty)}{g_\Lambda(z \to \infty)} \right]^{1.5} \tag{42}$$

with A_v a scaling constant and $g(z)$ the linear growth factor. Eq.(42) generalizes the Bullock et al. (2001) relation introducing a correction term to account for deviations from the ΛCDM model as proposed in [?]. The formation redshift z_f is a function of the virial mass M and is estimated solving

$$\frac{g[z_f(M)]}{g(z)} \sigma(f_v M, z) = \delta_c \tag{43}$$

where f_v is a constant and $\sigma(M, z)$ is the variance of the linear density field smoothed over a top hat filter with aperture $R = (3M/4\pi\bar{\rho})^{1/3}$, namely

$$\sigma^2(M, z) = \int_0^\infty \Delta^2_{lin}(k) \mathcal{T}^2(kR) d\ln k \tag{44}$$

with $\mathcal{T}(x) = 3(\sin x - x \cos x)/x^3$. Note that $\sigma(M, z) = g(z)\sigma(M, z = 0)$ because of the linear regime. In Eq.(43), δ_c is the critical overdensity for spherical collapse. This is usually fixed to the canonical value $\delta_c = 1.686$, but it has been found that a better agreement with simulations can be obtained taking [?]

$$\delta_c(z) = [1.59 + 0.0314 \ln \sigma_8(z)] \times [1 + 0.0123 \ln \Omega_M(z)] \tag{45}$$

with $\sigma_8(z) = \sigma_8 g(z)$ and σ_8 the present day variance on scale $R = 8h^{-1}$ Mpc.

The last ingredient needed to compute the one halo term in Eq.(38) is the halo mass function, i.e., the number of halo with mass in the range $(M, M + dM)$ in the unit voume. Following [?], it is

$$n(\nu) = A_{MF} \left[1 + \frac{1}{(a_{MF}\nu^2)^p} \right] \exp\left(-a_{MF}\nu^2/2\right) \tag{46}$$

with $A_{MF} = 0.2162$, $a_{MF} = 0.707$, $p = 0.3$ and $\nu = \delta_c(z)/\sigma(M, z)$. Note that ν also enters Eq.(38) through the normalized Fourier transfrom term $W^2(k, M)$ which is evaluated not in k, but in $k' = \nu^\eta k$ with $\eta = 0.603 - 0.3\sigma_8(z)$.

On the large scales, haloes are no more Poisson distributed and displacements between haloes must be accounted for introducing the two halo term $\Delta_{2H}(k)$. For matter perturbations, this can be straightforwardly evaluated since it is simply

$$\Delta^2_{2H}(k) = \Delta^2_{lin}(k) \tag{47}$$

so that one only needs the linear power spectrum. The total power spectrum should now be estimated as the sum of the two terms, but Mead et al. [?] have found that a much better agreement with simulations is obtained setting

$$\Delta^2(k) = \left[(\Delta'^2_{1H})^\alpha + (\Delta'^2_{2H})^\alpha \right]^{1/\alpha} \tag{48}$$

where the primed spectra are given by

$$\Delta'^2_{1H}(k) = \left\{1 - \exp\left[(-k/k_\star)^2\right]\right\} \Delta^2_{1H}(k) \; , \tag{49}$$

$$\Delta'^2_{2H}(k) = \left[1 - f_{2H} \tanh^2\left(k\sigma_v/\sqrt{f_{2H}}\right)\right] \Delta^2_{2H}(k) \; , \tag{50}$$

while $(\alpha, k_\star, f_{2H})$ are fitted to the simulations to get [**?**]

$$\alpha(z) = 3.24 \times 1.85^{n_{eff}(z)} \; , \tag{51}$$

$$k_\star(z) = 0.584/\sigma_v(z) \; , \tag{52}$$

$$f_{2H}(z) = 0.0095\, \sigma_{100}^{1.37}(z) \; . \tag{53}$$

Here, we have defined the following auxiliary functions

$$n_{eff}(z) = -3 - \left.\frac{d\ln\sigma^2(R,z)}{d\ln R}\right|_{\sigma=1} \; , \tag{54}$$

$$\sigma_v^2(R,z) = \frac{1}{3}\int_0^\infty \frac{\Delta^2_{lin}(k,z)}{k^2}\mathcal{T}^2(kR)d\ln k \tag{55}$$

$$\sigma_{100}(z) = \sigma_v(R = 100h^{-1}\ \mathrm{Mpc}, z) \; . \tag{56}$$

It is worth noting that a code to calclulate the power spectra implementing all the above formulae is available[2]. Originally conceived to fit the ΛCDM power spectrum taking care of nonlinearities and baryonic effects. However, it has been shown to work fine also for quintessence models with a CPL - like equation of state. Moreover, it can be easily extedend to models with massive neutrinos and modified gravity provided some fitting functions are accordingly changed (see [**?**] for the corresponding formulae).

[2] `https://github.com/alexander-mead/hmcode`