

CosmicFish Implementation Notes: Developers Version

Marco Raveri^{1,2,3}, Matteo Martinelli^{4,5}, Gong-Bo Zhao^{6,7} and Yuting Wang^{6,7}

¹ *SISSA - International School for Advanced Studies, Via Bonomea 265, 34136, Trieste, Italy*

² *INFN, Sezione di Trieste, Via Valerio 2, I-34127 Trieste, Italy*

³ *INAF-Osservatorio Astronomico di Trieste, Via G.B. Tiepolo 11, I-34131 Trieste, Italy*

⁴ *Institute Lorentz, Leiden University, PO Box 9506, Leiden 2300 RA, The Netherlands*

⁵ *Institut für Theoretische Physik, Ruprecht-Karls-Universität Heidelberg, Philosophenweg 16, 69120 Heidelberg, Germany.*

⁶ *National Astronomy Observatories, Chinese Academy of Science, Beijing, 100012, P.R.China*

⁷ *Institute of Cosmology & Gravitation, University of Portsmouth,*

Dennis Sciama Building, Portsmouth, PO1 3FX, UK

(Dated: April 10, 2017)

CosmicFish is a publicly available library to perform Fisher matrix forecast for several cosmological observations. With the present implementation notes we provide a guide to the physical and technical details of the library. We reproduce here the details and all the relevant equations, as they appear in the code. We submit these notes to the arXiv to grant full and permanent access to this material which provides a useful guidance to forecasting and the use of CosmicFish code. We will update this set of notes when relevant modifications to the CosmicFish code will be released. The present version is based on CosmicFish Developers Version.

PACS numbers: 98.80

Contents

I. Introduction and overview	2
II. Contributing to the CosmicFish code development	2
III. Fisher forecasts	3
IV. Statistical analysis of Fisher matrices	3
A. Marginalized bounds	3
B. Marginal ellipses plotting	4
C. Information Gain	4
V. Fisher matrix for angular power spectra	5
A. CMB angular power spectra	6
B. LSS angular power spectra	7
1. Window functions	7
2. LSS Weak Lensing Noise	9
3. LSS Number Counts Noise	9
VI. Fisher matrix for Supernovae observations	9
A. SN survey parameters	10
VII. Fisher matrix for Redshift Drift measurements	11
VIII. Fisher matrix for derived parameters	11
A. Derived parameters	12
IX. Examples	12
Acknowledgments	13
References	14

I. INTRODUCTION AND OVERVIEW

These notes contain the CosmicFish implementation details of all the physical quantities relevant to the code. The core of the CosmicFish code consists of two libraries. The first one is a Fortran library that takes care of producing the Fisher matrices. This is interfaced with CAMB sources [1, 2], EFTCAMB sources [3, 4] and MGCAMB sources [5, 6] and is automatically compiled against these three Einstein-Boltzmann codes for maximum model coverage.

The second part consists of a Python library optimized to perform operations on Fisher matrices, once they are produced. This also contains a full set of plotting utilities. Both these two libraries have applications built with them. For the Fortran part the only application already present is one that computes Fisher matrices. Other applications are in the process of being developed. The Python part contains several plotting applications that produce 1D, 2D and triangular marginalized plots, an application producing tables with marginalized bounds on the parameters and an application that performs a complete analysis of a set of Fisher matrices. Both libraries are based on state of the art, optimized, core algorithms including precise derivatives calculators, spectral protection of Fisher matrices against degenerate, unconstrained, parameters, to name a few.

The CosmicFish code is publicly available at <http://cosmicfish.github.io>.

The code comes with two unit testing suites that are responsible for checking all the building blocks of the two libraries. These can be easily used to check whether the code is correctly deployed and functioning or if modification introduced to the code break the correct execution of previous features.

The two libraries constituting the CosmicFish code are supplied with a thorough automatic documentation that explain the interfaces and the purposes of all the functions in both codes. This documentation is accessible at <http://cosmicfish.github.io/documentation/CosmicFish/index.html> for the Fortran part and at <http://cosmicfish.github.io/documentation/CosmicFishPyLib/index.html> for the Python part.

The codes comes with two packages that we used to validate the code and to perform the analysis presented in [7]. These packages are built to fully exploit the power of the library and show an example of a work pipeline with the CosmicFish library. The structure of these packages is extremely flexible and they can be easily used to build packages with other applications of the code.

The CosmicFish code is also publicly released in a developers version containing the latest update and features that are not yet implemented in the stable version. Thanks to its flexible and documented structure the developers version of the CosmicFish code can be easily modified to implement new features that can be useful to the cosmological community. We encourage everyone that is interested in developing its own applications or modifications to the code to use this version of the code, exploit it for scientific purposes and then to upload modifications to it to join our efforts in pursuing cosmological forecasting as a community achievement.

We believe that, in its first release, the CosmicFish code stands as a flexible and powerful framework to produce cosmological forecasts. Its fully modular structure, modern design and utilities make it a perfect tool to produce transparent and accessible scientific results and represent a first step toward community wide forecasting efforts.

II. CONTRIBUTING TO THE COSMICFISH CODE DEVELOPMENT

Cosmological forecasting is intrinsically a complicated matter as it requires deep knowledge of many branches of cosmology. In particular performing useful forecast requires modeling extended physical theories, understanding systematic effects related to cosmological observables and experimental design. All this can hardly be mastered by a single individual. For this reason we believe that CosmicFish, as a forecasting code, has to be open to contributions from other people that will bring their expertise to the code.

The developers version of the CosmicFish code is available at <https://github.com/CosmicFish/CosmicFish> and we welcome anyone to submit contributions.

In an effort toward allowing scientific recognition of contributors to the code we shall use the following scheme.

All contributions, regardless of size, are recognized and acknowledged in one of the CosmicFish files, distributed with the main code and in the acknowledgement section of these notes.

Contributors to the code are further divided into three levels:

- *Contributors*: anyone contributing to the code;
- *Developers*: anyone contributing to the CosmicFish code in the long-term and that provided substantial additions to the code;
- *Principal Developers*: anyone that contributed to the code to the same extent of other Principal Developers and with a global overview of the code;

To help developers in justifying their effort in contributing to the CosmicFish code, as an academic undertaking, we shall prepare a scientific paper exploiting the new features of the code at any mayor release of its stable version. *Principal Developers* shall be invited to sign the paper, *Developers* shall be invited to contribute to the realization of the paper. This paper will accompany the release of the new version of the code and will be the paper for which the code license requires a citation by the users. In addition *Principal Developers* will be invited to sign the CosmicFish implementation notes.

III. FISHER FORECASTS

Given a data set D and a model \mathcal{M} that is described by a vector of parameters θ we call the likelihood of the data $\mathcal{L}(\theta) \equiv P(D|\theta|\mathcal{M})$ the probability of the data given a set of parameters and a model. The Fisher information matrix is then defined by:

$$F_{ij} \equiv \left\langle \left(\frac{\partial \ln \mathcal{L}}{\partial \theta_i} \right) \left(\frac{\partial \ln \mathcal{L}}{\partial \theta_j} \right) \right\rangle_D = \int \left(\frac{\partial \ln \mathcal{L}}{\partial \theta_i} \right) \left(\frac{\partial \ln \mathcal{L}}{\partial \theta_j} \right) \mathcal{L} dD \quad (1)$$

Where brackets denote average over data realizations. The Fisher matrix can also be written as:

$$F_{ij} \equiv - \left\langle \left(\frac{\partial^2 \ln \mathcal{L}}{\partial \theta_i \partial \theta_j} \right) \right\rangle_D = - \int \left(\frac{\partial^2 \ln \mathcal{L}}{\partial \theta_i \partial \theta_j} \right) \mathcal{L} dD \quad (2)$$

since the difference between the two is zero once averaged over the data D .

The relevance of this quantity comes from to the Cramér-Rao lower bound [14]. This states that if:

$$\left\langle \frac{\partial \ln \mathcal{L}}{\partial \theta_i} \right\rangle_D = \int \left(\frac{\partial \ln \mathcal{L}}{\partial \theta_i} \right) \mathcal{L} dD = 0 \quad \forall \theta_i, \quad (3)$$

the covariance C of any unbiased estimator $\hat{\theta}$ of the parameters satisfies:

$$C(\hat{\theta}) - F^{-1} \geq 0 \quad (4)$$

where ≥ 0 means that this quantity is positive semidefinite.

IV. STATISTICAL ANALYSIS OF FISHER MATRICES

In this section we discuss some of the statistical analyses that can be performed with Fisher matrices at hand. These are all implemented in the CosmicFish code and we refer to the code documentation, <http://cosmicfish.github.io/documentation/CosmicFish/index.html> and <http://cosmicfish.github.io/documentation/CosmicFishPyLib/index.html>, for the implementation details.

A. Marginalized bounds

Once Fisher matrices are produced we can use them to forecast marginal bounds on parameters. Given a Fisher matrix F , the Cramér-Rao lower bound (4) implies that:

$$\text{var}(\hat{\theta}_i) \geq (F^{-1})_{ii} \quad (5)$$

If we assume that the posterior of the considered parameter is Gaussian we can forecast the bound at different confidence levels:

$$\text{C.L. bound} = \sqrt{2} \text{Erfinv}(\text{C.L.}) \sqrt{(F^{-1})_{ii}} \quad \text{where} \quad 0 \leq \text{C.L.} \leq 1, \quad (6)$$

where Erfinv denotes the inverse Error function and C.L. is the desired confidence level.

B. Marginal ellipses plotting

Once we compute a Fisher matrix it might be useful to plot the marginalized joint posterior probability of two parameters (θ_i, θ_j) , assuming that it is Gaussian, starting from the Fisher matrix. This helps shedding light on the possible degeneracy between two parameters and how this degeneracy changes or breaks when considering different experiments. Let us denote:

$$\begin{aligned}\sigma_i^2 &= (F^{-1})_{ii} \\ \sigma_j^2 &= (F^{-1})_{jj} \\ \sigma_{ij}^2 &= (F^{-1})_{ij}\end{aligned}\tag{7}$$

we can immediately compute:

$$\begin{aligned}a &= \sqrt{\frac{1}{2}(\sigma_i^2 + \sigma_j^2)} + \sqrt{\frac{1}{4}(\sigma_i^2 - \sigma_j^2)^2 + \sigma_{ij}^4} \\ b &= \sqrt{\frac{1}{2}(\sigma_i^2 + \sigma_j^2)} - \sqrt{\frac{1}{4}(\sigma_i^2 - \sigma_j^2)^2 + \sigma_{ij}^4} \\ \phi_0 &= \frac{1}{2} \text{atan} \left(\frac{2\sigma_{ij}^2}{\sigma_i^2 - \sigma_j^2} \right)\end{aligned}\tag{8}$$

and we can write the parametric form of the Fisher ellipse in the θ_i, θ_j plane as:

$$\begin{aligned}\theta_i &= \alpha (a \cos(\phi) \cos(\phi_0) - b \sin(\phi) \sin(\phi_0)) + \theta_i^0 \\ \theta_j &= \alpha (a \cos(\phi) \sin(\phi_0) + b \sin(\phi) \cos(\phi_0)) + \theta_j^0 \quad \forall \phi \in [0, 2\pi]\end{aligned}\tag{9}$$

where (θ_i^0, θ_j^0) is the value of the fiducial parameters and α is a coefficient encoding the confidence level of the ellipse and is given by $\alpha = \sqrt{2} \text{Erfinv}(\text{C.L.})$ where Erfinv denotes the inverse Error function and C.L. is the desired confidence level $0 \leq \text{C.L.} \leq 1$.

C. Information Gain

We can use Fisher matrices to forecasts the information gain between different experiments [8] as we shall review in this section. The statistical tool that we shall use at this goal is the Kullback-Leibler divergence, also called relative entropy or information gain. Consider two probability density functions (PDF), P_1 and P_2 of a d dimensional random variable θ . The Kullback-Leibler (KL) divergence is defined by:

$$D(P_2||P_1) \equiv \int P_2(\theta) \log_2 \left(\frac{P_2(\theta)}{P_1(\theta)} \right) d\theta = \frac{1}{\ln(2)} \int P_2(\theta) \ln \left(\frac{P_2(\theta)}{P_1(\theta)} \right) d\theta \quad [\text{bits}]\tag{10}$$

and represents the information difference in going from P_1 to P_2 in bits. The KL divergence has some noticeable properties:

- *Positive definite:* $D(P_2||P_1) \geq 0$ and $D(P_2||P_1) = 0$ iff $P_1 = P_2$;
- *Not symmetric:* $D(P_2||P_1) \neq D(P_1||P_2)$;
- *Invariant under re-parametrizations:* given $Y(X)$, a non-singular re-parametrization, $D(P_2(Y)||P_1(Y)) = D(P_2(X)||P_1(X))$.

If we assume that P_1 and P_2 are multivariate Gaussian distributions with mean θ_1 and θ_2 and covariance Σ_1 and Σ_2 the KL divergence becomes:

$$D(P_2||P_1) = \frac{1}{2 \ln 2} (\theta_1 - \theta_2)^T \Sigma_1^{-1} (\theta_1 - \theta_2) + \frac{1}{2 \ln 2} \left[-\ln \frac{\det \Sigma_2}{\det \Sigma_1} - d + \text{Tr}(\Sigma_2 \Sigma_1^{-1}) \right]\tag{11}$$

We can consider the posterior of some experiments as the PDF in the KL divergence to quantify the information difference between the interpretation of two different experiments within a model. With reference to Eq. (10) we shall consider $P_1 = P(\theta|D_1, \mathcal{M}) = P(\theta|\mathcal{M})P(D_1|\theta, \mathcal{M}) \equiv P(\theta)\mathcal{L}(D_1, \theta)$ and $P_2 = P(\theta|D_2, \mathcal{M}) = P(\theta|\mathcal{M})P(D_2|\theta, \mathcal{M}) \equiv P(\theta)\mathcal{L}(D_2, \theta)$ where D_1 and D_2 are two data sets and \mathcal{M} is a specific model.

If we apply the KL divergence to the posterior of two different experiments sometimes it proves useful to compute the expectation and variance of the KL divergence over D_2 realizations:

$$\langle D(P_2||P_1) \rangle_{D_2} \equiv \int D(P_2||P_1) \mathcal{L}(D_2) dD_2 \quad (12)$$

$$\sigma^2(D) \equiv \int [D(P_2||P_1) - \langle D(P_2||P_1) \rangle]^2 \mathcal{L}(D_2) dD_2 \quad (13)$$

We shall now consider the case where the two posterior are Gaussian in the parameters and the data and have the same mean values. We shall denote the Fisher matrix of the two experiments as F_1 and F_2 and the prior Fisher matrix as F_p . With these assumptions and notation it can be easily shown [8] that:

$$D(P_2||P_1) = \frac{1}{2 \ln 2} \left[-\ln \frac{\det(F_1 + F_p)}{\det(F_2 + F_p)} - d + \text{Tr} \left[(F_2 + F_p)^{-1} (F_1 + F_p) \right] \right] \quad (14)$$

$$\begin{aligned} \langle D(P_2||P_1) \rangle &= \frac{1}{2 \ln 2} \left[-\ln \frac{\det(F_1 + F_p)}{\det(F_2 + F_p)} - d + \text{Tr} \left[(F_2 + F_p)^{-1} (F_1 + F_p) \right] \right] \\ &+ \frac{1}{2 \ln 2} \text{tr} \left[F_2 (F_p + F_2)^{-1} (F_p + F_1) (F_p + F_2)^{-1} \left(I + F_2 (F_p + F_1)^{-1} \right) \right] \end{aligned} \quad (15)$$

$$\sigma^2(D) = \frac{1}{2 \ln 2} \text{tr} \left\{ \left[F_2 (F_p + F_2)^{-1} (F_p + F_1) (F_p + F_2)^{-1} \left(I + F_2 (F_p + F_1)^{-1} \right) \right]^2 \right\} \quad (16)$$

Equations (14, 15, 16) are the results that are implemented in the CosmicFish code to perform information gain forecasts.

Some properties of (14, 15, 16) are worth noticing:

- *KL divergence with uninformative priors*: if the prior is less informative than the Fisher matrix, i.e. $F_1, F_2 \gg F_p$ then:

$$\begin{aligned} D(P_2||P_1) &= \frac{1}{2 \ln 2} \left[-\ln \frac{\det F_1}{\det F_2} - d + \text{Tr} [F_2^{-1} F_1] \right] \\ \langle D(P_2||P_1) \rangle &= \frac{1}{2 \ln 2} \left[-\ln \frac{\det F_1}{\det F_2} + 2 \text{Tr} [F_2^{-1} F_1] \right] \\ \sigma^2(D) &= \frac{1}{2 \ln 2} \text{tr} \left[F_1 F_2^{-1} F_1 F_2^{-1} + 2 F_1 F_2^{-1} \right] + \frac{d}{2 \ln 2} \end{aligned} \quad (17)$$

- *KL divergence with strong prior*: in the limit where the prior is much more informative than the data, i.e. $F_p \gg F_1, F_2$ we have:

$$\lim_{F_p \gg F_1, F_2} D(P_2||P_1) \rightarrow 0 ; \quad \langle D(P_2||P_1) \rangle \rightarrow 0 ; \quad \sigma^2(D) \rightarrow 0 \quad (18)$$

that is, if the prior are too strong the information that we can gain is reduced to zero.

V. FISHER MATRIX FOR ANGULAR POWER SPECTRA

In this section we shall describe how we obtain the Fisher matrix for angular power spectra [13].

Consider N fields measured on the sky. These will constitute a vector \vec{T} of angle (θ) dependent measurements that we can decompose into spherical harmonics:

$$\vec{T}(\theta) = \sum_{l,m} \vec{a}_{lm} Y_{lm}(\theta) \quad \text{with} \quad \dim(\vec{a}_{lm}) = N \quad (19)$$

We shall assume that the \vec{a}_{lm} are Gaussian distributed with zero mean and covariance $C_l^{XY} = \langle |a_{lm}^X a_{lm}^{Y+}| \rangle$ where X and Y denote two of the observables in \vec{T} . This covariance constitutes our theoretical prediction and we shall compare it to the data estimate of the covariance:

$$\hat{C}_l = \frac{1}{2l+1} \sum_{m=-l}^l \vec{a}_{lm} \vec{a}_{lm}^+ \quad (20)$$

At a fixed m and l the probability of \vec{a}_{lm} can be written as:

$$P(\vec{a}_{lm}|C_l) = (2\pi)^{-N/2} |C_l^{XY}|^{-1/2} \exp \left[-\frac{1}{2} \vec{a}_{lm}^T (C_l^{XY})^{-1} \vec{a}_{lm} \right] \quad (21)$$

where we used the notation $\det(\cdot) = |\cdot|$. The joint probability distribution of the $2l+1$ independent \vec{a}_{lm} is then given by:

$$\begin{aligned} P(\vec{a}_{lm}|C_l) &= \prod_{m=-l}^l \left[(2\pi)^{-N/2} |C_l^{XY}|^{-1/2} \exp \left[-\frac{1}{2} \vec{a}_{lm}^T (C_l^{XY})^{-1} \vec{a}_{lm} \right] \right] \\ &= (2\pi)^{-N(2l+1)/2} |C_l^{XY}|^{-(2l+1)/2} \exp \left[-\frac{2l+1}{2} \sum_{XY} \hat{C}_l^{XY} (C_l^{XY})^{-1} \right] \end{aligned} \quad (22)$$

An experiment measures the C_l 's from l_{\min} to l_{\max} so that the joint probability distribution of the full measurements is given by:

$$\begin{aligned} \ln \mathcal{L} = \ln P(\hat{C}_l|C_l) &= -\frac{1}{2} N \log(2\pi) (1 + l_{\max} - l_{\min}) (1 + l_{\max} + l_{\min}) \\ &\quad - \frac{1}{2} \sum_{l=l_{\min}}^{l_{\max}} (2l+1) \left[\log |C_l^{XY}| + \text{Tr} \left[\hat{C}_l^{XY} (C_l^{XY})^{-1} \right] \right] \end{aligned} \quad (23)$$

With this at hand we can immediately compute the Fisher matrix for angular power spectra as:

$$F_{ij} = \sum_{l=l_{\min}}^{l_{\max}} \frac{2l+1}{2} \sum_{\alpha\beta\gamma\delta=1}^N \frac{\partial C_l^{\alpha\beta}}{\partial \theta_i} (\hat{C}_l^{\beta\gamma})^{-1} \frac{\partial C_l^{\gamma\delta}}{\partial \theta_j} (\hat{C}_l^{\delta\alpha})^{-1} \quad (24)$$

Here we model the observed \hat{C}_l^{XY} as:

$$\hat{C}_l^{XY} = \frac{(C_l^{XY} + N_l^{XY})}{\sqrt[4]{f_{\text{sky}}^X f_{\text{sky}}^Y}} \quad (25)$$

The angular power spectra that are already implemented in the CosmicFish code are:

- CMB temperature, E and B mode polarization, CMB lensing as discussed in Section V A.
- Galaxy weak lensing, as discussed in Section V B;
- Galaxy number counts fluctuations, as discussed in Section V B;

To compute the Fisher matrix for angular power spectra the parameter flag `cosmicfish_want_cls` has to be turned to true. By default the cross correlation between all the angular power spectra is considered. To remove cross-correlation from the Fisher matrix calculation the user should act on the flag `Fisher_want_XC`.

A. CMB angular power spectra

We implement four different CMB angular power spectra accounting for their cross correlation. In particular we have CMB Temperature spectrum, CMB E-mode polarization spectrum, CMB B-mode polarization spectrum and CMB lensing angular spectrum. Since we consider only statistical noise these are not correlated and the noise in the

cross-correlation is zero. In the case of the Temperature spectrum we consider different frequency channels ν so that the total statistical noise is given by:

$$\begin{aligned} N_\ell^{TT} &= \left(\sum_\nu \frac{1}{\text{var}(\nu)} \exp \left(-\ell(\ell+1)\sigma^2(\nu) \right) \right)^{-1} \\ \text{var}(\nu) &= \left(S_T(\nu)\theta(\nu) \frac{\pi T_{\text{CMB}}}{180 \times 60} \right)^2 \\ \sigma^2(\nu) &= \left(\frac{\pi \theta(\nu)}{180 \times 60 \sqrt{8 \log 2}} \right)^2 \end{aligned} \quad (26)$$

where $\theta(\nu)$ is the beam FWHM in arcmin and $S_T(\nu) \equiv \Delta T/T$ is temperature sensitivity. In the case of E-B mode polarization the noise is given by the same Eq. 26 with $S_T(\nu)$ replaced by $S_P(\nu) \equiv \Delta P/T$ as polarization sensitivity. In the case of CMB lensing noise is given by [9] and computed with the code developed in [10]. Different CMB spectra can be included in the Fisher matrix calculation by acting on the flags: **Fisher_want_CMB_T**, **Fisher_want_CMB_E**, **Fisher_want_CMB_B** and **Fisher_want_CMB_lensing**. The relevant parameters for CMB experiments are:

- **CMB_n_channels**: number of frequency channels of the CMB experiment;
- **CMB_TT_fsky**, **CMB_EE_fsky** and **CMB_BB_fsky** to select f_{sky} for different observables;
- **l_max_TT**, **l_max_EE** and **l_max_BB** to select the maximum multipole for a given observable;
- **CMB_temp_sens(i)** to select temperature sensitivity ($\Delta T/T$) in the *i*-th frequency channel;
- **CMB_pol_sens(i)** to select polarization sensitivity ($\Delta P/T$) in the *i*-th frequency channel;
- **CMB_fwhm(i)** to select the beam FWHM in arcmin for the *i*-th frequency channel;

B. LSS angular power spectra

We implement in the CosmicFish code the possibility to include Large Scale Structure (LSS) angular power spectra in the Fisher matrix forecast. In particular, based on the output of CAMB sources, we implement galaxy Weak Lensing (WL) and tracers number counts fluctuations, hereafter GC.

As with CMB spectra, by default the code includes in the forecast all the cross correlation between different LSS observables and also the cross correlation between CMB observables and LSS observables.

In this section we discuss the details of the implementation of LSS forecast.

1. Window functions

For LSS observables we need to specify the corresponding window functions. We implement several choices that we shall discuss in this section.

- *Gaussian window function*: the Gaussian window function is specified as:

$$W(z) = \frac{1}{\sqrt{2\pi}\sigma} \exp \left(-\frac{1}{2} \frac{(z - z_0)^2}{\sigma^2} \right) \quad (27)$$

where z_0 indicates the window mean redshift and σ denotes the window spread. This window can be selected with the parameter **window_type=1**. The parameters of this window can be specified to the code by: **redshift(i)**= z_0 in the *i*-th window; **redshift_sigma(i)**= σ in the *i*-th window.

- *Binned window function*: the binned window function is specified by a global window function that encodes the information about all the objects found by a survey. We implement:

$$\bar{N}_{\text{survey}}(z) = \frac{\beta}{z_0^{(1+\alpha)} \Gamma \left(\frac{1+\alpha}{\beta} \right)} z^\alpha \exp \left[- \left(\frac{z}{z_0} \right)^\beta \right] \quad (28)$$

where α , β and z_0 are the parameters specifying the window function. In particular α and β are two parameters defining the shape of the window function, Γ is the Euler gamma function and z_0 is the window median redshift. This global window function has several properties:

$$\begin{aligned} \int_{z_1}^{z_2} \bar{N}_{\text{survey}}(z) dz &= \frac{1}{\Gamma\left(\frac{1+\alpha}{\beta}\right)} \left[\Gamma\left(\frac{1+\alpha}{\beta}, \left(\frac{z_1}{z_0}\right)^\beta\right) - \Gamma\left(\frac{1+\alpha}{\beta}, \left(\frac{z_2}{z_0}\right)^\beta\right) \right] \\ \int_0^\infty \bar{N}_{\text{survey}}(z) dz &= 1 \end{aligned} \quad (29)$$

Photometric redshift determinations have always some errors. To account for them the window function is usually smoothed. If we consider a redshift bin going from z_1 to z_2 the Gaussian smoothed binned window function becomes:

$$\begin{aligned} W(z) &= \int_{z_1}^{z_2} \bar{N}_{\text{survey}}(z) \exp\left[-\frac{(\tilde{z}-z)^2}{2\sigma_z^2(z)}\right] d\tilde{z} = \bar{N}_{\text{survey}}(z) \int_{z_1}^{z_2} \exp\left[-\frac{(\tilde{z}-z)^2}{2\sigma_z^2(z)}\right] d\tilde{z} \\ &= \frac{\bar{N}_{\text{survey}}(z)}{2} \left[\text{Erf}\left(\frac{z-z_1}{\sqrt{2}\sigma_z(z)}\right) - \text{Erf}\left(\frac{z-z_2}{\sqrt{2}\sigma_z(z)}\right) \right] \\ &= \frac{\bar{N}_{\text{survey}}(z)}{2} \left[\text{Erf}\left(\frac{z-z_1}{\sqrt{2}\sigma_z(z)}\right) - \text{Erf}\left(\frac{z-(z_1+\Delta z)}{\sqrt{2}\sigma_z(z)}\right) \right] \end{aligned} \quad (30)$$

Where $\sigma_z^2(z)$ is the model for the photometric redshift error and Erf is the error function. We implement a simple model for σ_z that is specified by $\sigma_z(z) = \sigma_0(1+z)$. The parameters for this window function can be specified to the code by:

- `window_type=2` to select this kind of window;
- `window_alpha= α` ;
- `window_beta= β` ;
- `redshift_zero= z_0` ;
- `photoz_error= σ_0` ;
- `redshift(i)= z_1` in the i-th window;
- `redshift_sigma(i)= Δz` in the i-th window;

- *Flat window function*: this window function is specified by:

$$W(z) = \begin{cases} 1 & z_1 \leq z \leq z_1 + \Delta z \\ 0 & \text{otherwise} \end{cases} \quad (31)$$

This window function can be selected by setting `window_type=3` and the parameters: `redshift(i)= z_1` in the i-th window; `redshift_sigma(i)= Δz` in the i-th window.

- *Smoothed flat window function*: this window function is equivalent to the flat window function but accounting for photo-z error Gaussian smoothing. It is specified by:

$$W(z) = \frac{\bar{N}_{\text{survey}}(z)}{2} \left[\text{Erf}\left(\frac{z-z_1}{\sqrt{2}\sigma_z(z)}\right) - \text{Erf}\left(\frac{z-(z_1+\Delta z)}{\sqrt{2}\sigma_z(z)}\right) \right] \quad (32)$$

where:

$$\bar{N}_{\text{survey}}(z) = \begin{cases} 1 & z_1 \leq z \leq z_1 + \Delta z \\ 0 & \text{otherwise} \end{cases} \quad (33)$$

This window function can be selected by setting `window_type=4` and the parameters: `redshift(i)= z_1` in the i-th window; `redshift_sigma(i)= Δz` in the i-th window. We implement $\sigma_z(z) = \sigma_0(1+z)$ where $\sigma_0 = \text{photoz_error}$.

Notice that all these window functions are automatically normalized to unity by the CAMB code so we do not need to specify the normalization.

These window functions can be used for both GC and WL. In its current version, the code does not allow to use two different functional forms of window functions at the same time. This is likely to be upgraded in the future.

All the LSS windows have also the possibility of having a different f_{sky} and a different l_{max} by setting: `LSS_fsky(i)` for the sky fraction and `LSS_lmax(i)` for the maximum multipole, in the i-th window.

2. LSS Weak Lensing Noise

In this section we shall discuss the experimental noise that is added to the predicted C_l . In the case of WL we consider a noise given by:

$$N_\ell^{\text{WL}_i \text{WL}_i} = \frac{\gamma_{\text{rms}}^2}{N_i} \quad (34)$$

where is the rms shear stemming from the intrinsic ellipticity of the galaxies and N_i is the number of galaxies in the window. In code notation: `LSS_intrinsic_ellipticity(i) = γ_{rms}` and `LSS_num_galaxies(i) = N_i` in the i -th window.

3. LSS Number Counts Noise

In this section we shall discuss the experimental noise that we consider for number counts fluctuations. This is given by:

$$N_\ell^{\text{GC}_i \text{GC}_i} = \frac{1}{N_i} \quad (35)$$

where N_i is the number of galaxies in the window. In code notation: `LSS_num_galaxies(i) = N_i` in the i -th window.

For GC we need to specify bias. In the present version of the code this is treated as a constant, scale independent, possibly different in all redshift windows. The value of bias in the i -th window is specified by: `redshift_bias(i)`. It is possible to add these values of the bias to the Fisher matrix calculation by setting `param[bias]` to true. More complicated bias models will be added in the near future.

VI. FISHER MATRIX FOR SUPERNOVAE OBSERVATIONS

In this section we discuss how we obtain Fisher matrix forecast for Supernovae observations. We observe the magnitude of a supernova at a certain redshift and we model it as:

$$m(z_i) = 5 \log_{10} (D_L(z_i)) - \alpha X_1 + \beta C + M_0 - 25 \quad (36)$$

where X_1 is the stretch of the supernova, C its color and M_B the intrinsic luminosity. If we measure the luminosity distance, redshift, color and stretch of a set of N supernovae and we assume that all these measurements are Gaussian distributed with mean $m(z_i)$ and covariance Σ we can immediately write the likelihood as:

$$\mathcal{L} = (2\pi)^{-N/2} \det(\Sigma)^{-1/2} \exp \left(-\frac{1}{2} (\vec{m}_{\text{obs}} - \vec{m})^T \Sigma^{-1} (\vec{m}_{\text{obs}} - \vec{m}) \right) \quad (37)$$

We model the covariance as diagonal and:

$$\Sigma = \sigma_m^2 + \sigma_{\text{lens}}(z)^2 + \sigma_z(z)^2 + \beta^2 \sigma_C^2(z) + \alpha^2 \sigma_{X_1}^2(z) + 2\alpha \sigma_{mX_1}(z) - 2\beta \sigma_{Cm}(z) - 2\alpha\beta \sigma_{X_1C}(z) \quad (38)$$

where: σ_m is the error in determining the SN magnitude; σ_{lens} is the error due to the lensing of the SN and we model its redshift dependence as $\sigma_{\text{lens}}(z) = \sigma_{\text{lens}} z$; σ_z is the error in the determination of the SN redshift whose redshift dependence is modeled by $\sigma_z(z) = \frac{5\sigma_z}{z \ln(10)}$; σ_C is the error in the determination of SN color and we model its redshift dependence as $\sigma_C(z) = \sigma_{C0} + \sigma_{C2} z^2$; σ_{X_1} is the error in the determination of the SN stretch and we model its redshift dependence as $\sigma_{X_1}(z) = \sigma_{X_10} + \sigma_{X_12} z^2$; σ_{mX_1} is the error in the joint determination of stretch and magnitude of a supernova and we model its redshift dependence as $\sigma_{mX_1}(z) = \sigma_{mX_10} + \sigma_{mX_12} z^2$; σ_{Cm} is the error in the joint determination of the color and magnitude of a supernova and we model its redshift dependence as $\sigma_{Cm}(z) = \sigma_{Cm0} + \sigma_{Cm2} z^2$; σ_{X_1C} is the error in the joint determination of the supernova stretch and color and we model its redshift dependence as $\sigma_{X_1C}(z) = \sigma_{X_1C0} + \sigma_{X_1C2} z^2$.

With this at hand we can compute the Fisher matrix that becomes:

$$F_{ab} = \left\langle \frac{\partial m_i}{\partial p_a} \Sigma_{ij}^{-1} \frac{\partial m_j}{\partial p_b} + \frac{1}{2} \text{Tr} \left(\Sigma^{-1} \frac{\partial \Sigma}{\partial p_a} \Sigma^{-1} \frac{\partial \Sigma}{\partial p_b} \right) \right\rangle_D \quad (39)$$

because the covariance matrix depends on the parameters α and β . Notice that this quantity needs to be averaged over the data. In particular the average over the observed magnitude has already been taken into account while the average over redshift, color and stretch has to be performed. This turns out to be challenging to do analytically. We thus allow the option to perform a Monte-Carlo average of this quantity considering color and stretch to be Gaussian distributed with zero mean and user defined spread.

In the present version of the code the term depending on the covariance derivative is not implemented. It will be implemented in a future code release.

A. SN survey parameters

In this section we review the parameters that can be used to tell the code the specifications of a SN survey.

- `SN_Fisher_MC_samples`: number of Monte Carlo Fisher matrix samples;
- `alpha_SN`: fiducial value of the stretch coefficient. This can be added as a parameter of the Fisher matrix by setting `param[alpha_SN]` to true;
- `beta_SN`: fiducial value of the color coefficient. This can be added as a parameter of the Fisher matrix by setting `param[beta_SN]` to true;
- `M0_SN`: fiducial value of the SN intrinsic luminosity. This can be added as a parameter of the Fisher matrix by setting `param[M0_SN]` to true;
- `color_dispersion`: dispersion in the colors. This fixes the variance in color of the generated mock SN catalog;
- `stretch_dispersion`: dispersion in the stretch. This fixes the variance in color of the generated mock SN catalog;
- `magnitude_sigma` = σ_m ;
- `c_sigmaz` = σ_z ;
- `sigma_lens_0` = σ_{lens} ;
- `dcolor_offset` = σ_{C0} ;
- `dcolor_zcorr` = σ_{C2} ;
- `dshape_offset` = σ_{X10} ;
- `dshape_zcorr` = σ_{X12} ;
- `cov_ms_offset` = σ_{mX10} ;
- `cov_ms_zcorr` = σ_{mX12} ;
- `cov_mc_offset` = σ_{Cm0} ;
- `cov_mc_zcorr` = σ_{Cm2} ;
- `cov_sc_offset` = σ_{X1C0} ;
- `cov_sc_zcorr` = σ_{X1C2} ;
- `number_SN_windows`: number of redshift bin in the SN redshift distribution;
- `SN_redshift_start(i)`: lower redshift if the i-th SN redshift distribution bin;
- `SN_redshift_end(i)`: higher redshift if the i-th SN redshift distribution bin;
- `SN_number(i)`: number of SN in the i-th SN redshift distribution bin;

VII. FISHER MATRIX FOR REDSHIFT DRIFT MEASUREMENTS

In this section we describe how we obtain the Fisher matrix for Redshift Drift (RD) measurements. The theoretical shift in the spectroscopic velocity of a source after a time interval Δt can be modelled as

$$\Delta v(z) = cH_0\Delta t \left[1 - \frac{E(z)}{1+z} \right] \quad (40)$$

where $E(z) \equiv H(z)/H_0$ and H_0 is expressed in s^{-1} .

The error on this quantity is strongly dependent on the measurement strategy adopted and on the observed sources; at the moment CosmicFish follows the modelling of the uncertainties adopted by the European Extremely Large Telescope (E-ELT), expected to produce measurements of this effect observing QSO absorption systems. Given this setup, the uncertainties on the measurements can be modelled as:

$$\sigma_{\Delta v} = 1.35 \frac{2370}{S/N} \sqrt{\frac{30}{N_{QSO}}} \left(\frac{5}{1+z} \right)^x \quad (41)$$

where N_{QSO} is the number of observed system in the redshift bin centered in z , S/N is the signal-to-noise ratio of the observations and $x = 1.7$ for $z \leq 4$ and $x = 0.9$ for $z > 4$.

Defining the likelihood for a set of N measurements at redshifts z_k as

$$\mathcal{L} \propto \exp \left[- \sum_{k=1}^N \frac{(\Delta v_{\text{obs}}(z_k) - \Delta v(z_k))^2}{2\sigma_{\Delta v}^2} \right] \quad (42)$$

Given this shape of the likelihood function, the Fisher matrix can be written as

$$F_{ij} = - \frac{\partial^2 \ln \mathcal{L}(\vec{\theta})}{\partial \theta_i \partial \theta_j} = \sum_{k=1}^N \frac{1}{\sigma_k^2} \frac{\partial \Delta v_k}{\partial \theta_i} \frac{\partial \Delta v_k}{\partial \theta_j} \quad (43)$$

The parameters that can be used to build the redshift drift data set are

- **RD_exptype**: selects between different error modelization (only 1 available in the present version);
- **number_RD_redshifts**: number of redshift bins;
- **delta_time**: time interval Δt (in years) for which to compute the shift;
- **RD_sig_to_noise**: signal to noise ratio S/N ;
- **RD_source_number(i)**: number of observed sources in the i -th redshift bin;
- **RD_redshift(i)**: central redshift value of the i -th redshift bin.

VIII. FISHER MATRIX FOR DERIVED PARAMETERS

In this section we discuss how we obtain the Jacobian matrix that is needed to propagate the bounds from a set of parameters to another. We can easily derive the transformation of the Fisher matrix under a change in the parameters. If we denote a set of parameters by θ_i and another set of parameters by $\tilde{\theta}_i$ and taking equation (1) we immediately have:

$$\tilde{F}_{ij} = \left\langle \left(\frac{\partial \ln \mathcal{L}}{\partial \tilde{\theta}_i} \right) \left(\frac{\partial \ln \mathcal{L}}{\partial \tilde{\theta}_j} \right) \right\rangle_D = \frac{\partial \theta_k}{\partial \tilde{\theta}_i} \left\langle \left(\frac{\partial \ln \mathcal{L}}{\partial \theta_k} \right) \left(\frac{\partial \ln \mathcal{L}}{\partial \theta_m} \right) \right\rangle_D \frac{\partial \theta_m}{\partial \tilde{\theta}_j} = \frac{\partial \theta_k}{\partial \tilde{\theta}_i} F_{km} \frac{\partial \theta_m}{\partial \tilde{\theta}_j} \quad (44)$$

similarly we have:

$$F_{km} = \frac{\partial \tilde{\theta}_i}{\partial \theta_k} \tilde{F}_{ij} \frac{\partial \tilde{\theta}_j}{\partial \theta_m} \quad (45)$$

The CosmicFish code then computes numerically $J \equiv \frac{\partial \tilde{\theta}_i}{\partial \theta_m}$ and the Fisher matrix with the derived parameters can be computed as:

$$\tilde{F} = (J^T F^{-1} J)^{-1} \quad (46)$$

Notice that the derived Fisher matrix satisfies the Cramér-Rao lower bound for the derived parameters [14]. The derived parameters Jacobian matrix is computed by the CosmicFish code if `cosmicfish_want_derived` is set to true.

A. Derived parameters

The CosmicFish code allows to get derived bounds on several derived parameters that we list here as a reference:

- The present day baryon relative density Ω_b . Active if `param[omegab]` is true;
- The present day CDM relative density Ω_c . Active if `param[omegac]` is true;
- The present day neutrino relative density Ω_ν . Active if `param[omegan]` is true;
- The present day DE relative density Ω_Λ . Active if `param[omegav]` is true;
- The present day curvature relative density Ω_k . Active if `param[omegak]` is true;
- The present day total matter relative density Ω_m . Active if `param[omegam]` is true;
- θ_{CMB} that measures the sound horizon at last scattering. Active if `param[theta]` is true;
- Total mass of massive neutrinos m_ν . Active if `param[mnu]` is true;
- The reionization redshift z_{re} . Active if `param[zre]` is true;
- Effective number of relativistic species n_{eff} . Active if `param[neff]` is true;

In addition there are a couple of tomographic derived parameters, that can be obtained at several different redshifts. These are:

- $\sigma_8(z)$ the amplitude of scalar perturbations on the scale of $8 h^{-1} \text{Mpc}$. Active if `param[sigma8]` is true;
- $\log \mathcal{H}(z)$ the logarithm of the Hubble parameter at user specified redshifts. Active if `param[loghubble]` is true;
- $\log D_A(z)$ the logarithm of the angular diameter distance at user specified redshifts. Active if `param[logDA]` is true;

Tomographic derived parameters are active if `FD_num_redshift` is set greater than zero. If this is the case `FD_redshift(i)` specifies the *i*-th redshift of the tomographic derived parameter. These constitute a basic set of derived parameters. It should be easy for the user to implement new ones.

IX. EXAMPLES

In this section we showcase some example usage of the CosmicFish code. All the results presented here are exactly the ones that can be obtained by running the example code distributed with the library. To do so the user needs to go to the `examples` directory and issue the command `make all`, after compiling successfully the library. This will produce all the Fisher matrices needed for these examples, by means of the Fortran CosmicFish library and then produce these plots with the Python CosmicFish library.

We choose to show, as examples, some of the forecast results for a *Planck* Blue Book (BB) CMB satellite, as taken from [11], and a DES-like Galaxy Clustering survey with specifications taken from [12].

In Figure 1 we show the marginalized 1D plots on three cosmological parameters while in Figure 2 we show their 2D joint forecasted PDF. In Figure 3 we show the triangular plot of these three parameters while Table IX shows the 68% C.L. confidence forecasted bounds. In all figures we show the *Planck* BB constraints and the *Planck*+DES constraints to show an example of a plot displaying a the results from a single Fisher matrix and multiple ones. Notice that the code automatically maintains color consistency throughout all the plots, within a single execution of the program.

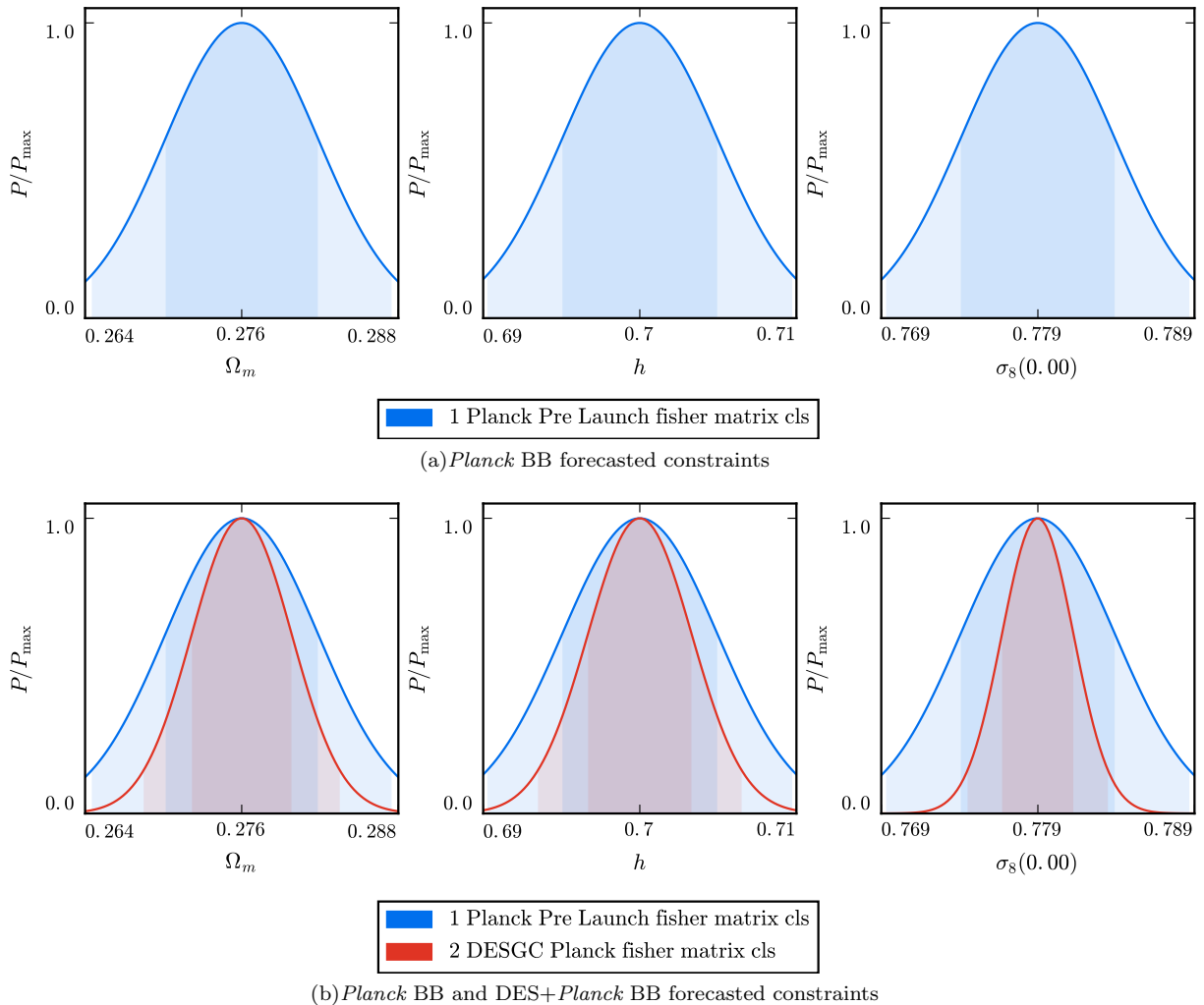


FIG. 1: The marginalized 1D forecasted constraints on Ω_m , h and σ_8 . Different colors correspond to different combinations of experiments, as shown in legend. The darker and lighter color shades indicate the forecasted 68% C.L. and 95% C.L. bounds.

1 Planck Pre Launch fisher matrix cls marginal			
$\Omega_b h^2 = 0.0226 \pm 0.0001$	$h = 0.7 \pm 0.005$	$n_s = 0.96 \pm 0.003$	$\Omega_m = 0.276 \pm 0.006$
$\Omega_c h^2 = 0.112 \pm 0.001$	$\ln(10^{10} A_s) = 3.045 \pm 0.007$	$\tau = 0.09 \pm 0.004$	$\sigma_8(0.00) = 0.779 \pm 0.005$
2 DESGC Planck fisher matrix cls marginal			
$\Omega_b h^2 = 0.0226 \pm 0.0001$	$h = 0.7 \pm 0.003$	$n_s = 0.96 \pm 0.003$	$\Omega_m = 0.276 \pm 0.004$
$\Omega_c h^2 = 0.112 \pm 0.0007$	$\ln(10^{10} A_s) = 3.045 \pm 0.005$	$\tau = 0.09 \pm 0.003$	$\sigma_8(0.00) = 0.779 \pm 0.002$

TABLE I: The 68% confidence level bounds on cosmological parameters obtained with *Planck* BB and DES forecasts.

Acknowledgments

We are grateful to Ana Achúcarro, Carlo Baccigalupi, Erminia Calabrese, Stefano Camera, Luigi Danese, Giulio Fabbian, Noemi Frusciante, Bin Hu, Valeria Pettorino, Levon Pogorian, Giuseppe Puglisi and Alessandra Silvestri for useful and helpful discussions on the subject. We are indebted to Luca Heltai for help with numerical algorithms. MM is supported by the Foundation for Fundamental Research on Matter (FOM) and the Netherlands Organization for Scientific Research / Ministry of Science and Education (NWO/OCW). MM was also supported by the DFG TransRegio TRR33 grant on The Dark Universe during the preparation of this work. MR acknowledges partial support by the Italian Space Agency through the ASI contracts Euclid-IC (I/031/10/0) and the INFN-INDARK

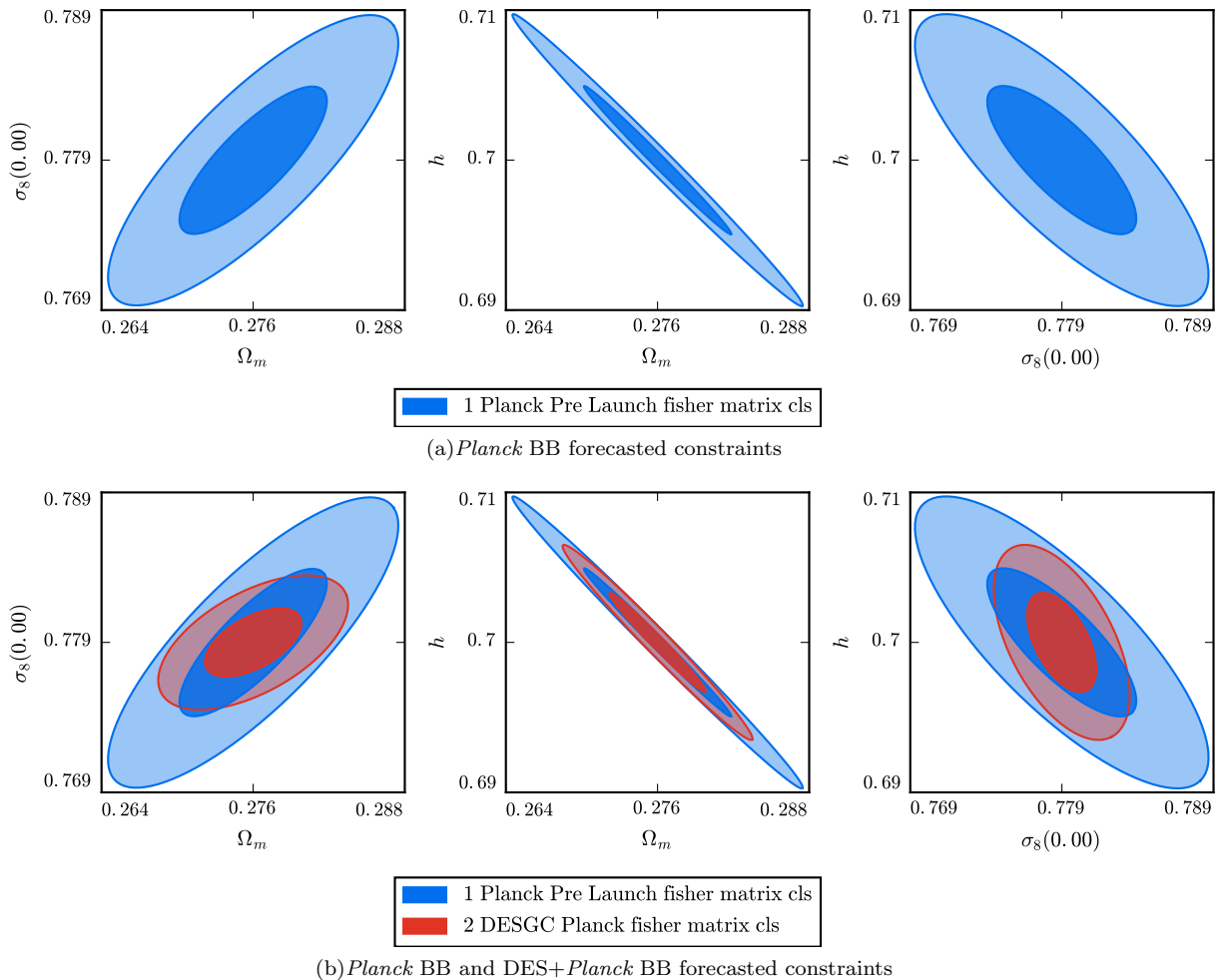


FIG. 2: The marginalized 2D forecasted constraints on Ω_m , h and σ_8 . Different colors correspond to different combinations of experiments, as shown in legend. The darker and lighter color shades indicate the forecasted 68% C.L. and 95% C.L. bounds.

initiative. MR acknowledges the joint SISSA/ICTP Master in High Performance Computing for support during the development of this work. MR thanks the National Astronomical Observatories, Chinese Academy of Science for the hospitality during the initial phases of development of this work. MR and MM thank the Galileo Galilei Institute for Theoretical Physics for the hospitality and the INFN for partial support during the completion of this work. GBZ and YW are supported by the Strategic Priority Research Program “The Emergence of Cosmological Structures” of the Chinese Academy of Sciences Grant No. XDB09000000, and by University of Portsmouth. YW is supported by the NSFC grant No. 11403034.

-
- [1] A. Lewis, A. Challinor and A. Lasenby, *Astrophys. J.* **538**, 473 (2000) doi:10.1086/309179 [astro-ph/9911177].
 - [2] A. Challinor and A. Lewis, *Phys. Rev. D* **84**, 043516 (2011) doi:10.1103/PhysRevD.84.043516 [arXiv:1105.5292 [astro-ph.CO]].
 - [3] B. Hu, M. Raveri, N. Frusciante and A. Silvestri, *Phys. Rev. D* **89**, no. 10, 103530 (2014) doi:10.1103/PhysRevD.89.103530 [arXiv:1312.5742 [astro-ph.CO]].
 - [4] M. Raveri, B. Hu, N. Frusciante and A. Silvestri, *Phys. Rev. D* **90**, no. 4, 043513 (2014) doi:10.1103/PhysRevD.90.043513 [arXiv:1405.1022 [astro-ph.CO]].
 - [5] G. B. Zhao, L. Pogosian, A. Silvestri and J. Zylberberg, *Phys. Rev. D* **79**, 083513 (2009) doi:10.1103/PhysRevD.79.083513 [arXiv:0809.3791 [astro-ph]].
 - [6] A. Hojjati, L. Pogosian and G. B. Zhao, *JCAP* **1108**, 005 (2011) doi:10.1088/1475-7516/2011/08/005 [arXiv:1106.4543 [astro-ph.CO]].
 - [7] M. Raveri and M. Martinelli, *Information Gain in Cosmology: From the Discovery of Expansion to Future Surveys*6 [arXiv:1606.XXXX [astro-ph.CO]].
 - [8] S. Seehars, A. Amara, A. Refregier, A. Paranjape and J. Akeret, *Phys. Rev. D* **90**, no. 2, 023533 (2014)

- doi:10.1103/PhysRevD.90.023533 [arXiv:1402.3593 [astro-ph.CO]].
- [9] W. Hu and T. Okamoto, *Astrophys. J.* **574**, 566 (2002) doi:10.1086/341110 [astro-ph/0111606].
 - [10] L. Perotto, J. Lesgourgues, S. Hannestad, H. Tu and Y. Y. Y. Wong, *JCAP* **0610**, 013 (2006) doi:10.1088/1475-7516/2006/10/013 [astro-ph/0606227].
 - [11] Planck - The Scientific Programme <http://sci.esa.int/jump.cfm?oid=47334>
 - [12] O. Lahav, A. Kiakotou, F. B. Abdalla and C. Blake, *Mon. Not. Roy. Astron. Soc.* **405**, 168 (2010) doi:10.1111/j.1365-2966.2010.16472.x [arXiv:0910.4714 [astro-ph.CO]].
 - [13] M. Tegmark, A. Taylor and A. Heavens, *Astrophys. J.* **480**, 22 (1997) doi:10.1086/303939 [astro-ph/9603021].
 - [14] S. M. Kay, “Fundamentals of statistical signal processing: estimation theory” Prentice Hall; 1 edition (April 5, 1993)

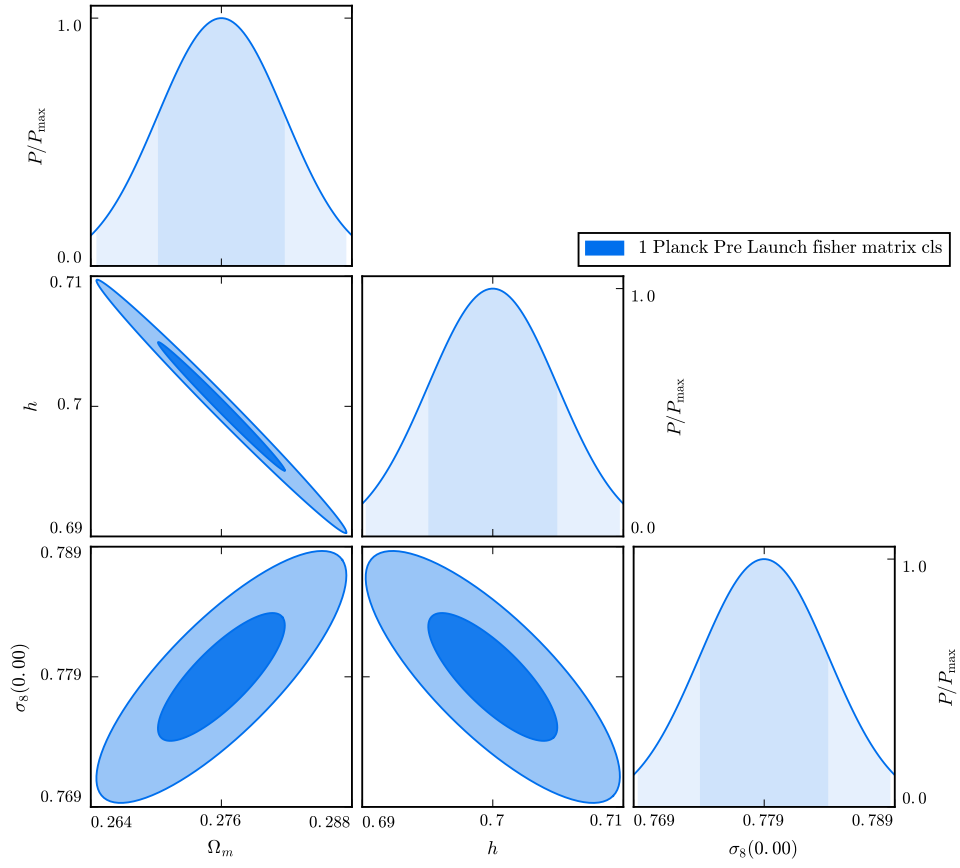
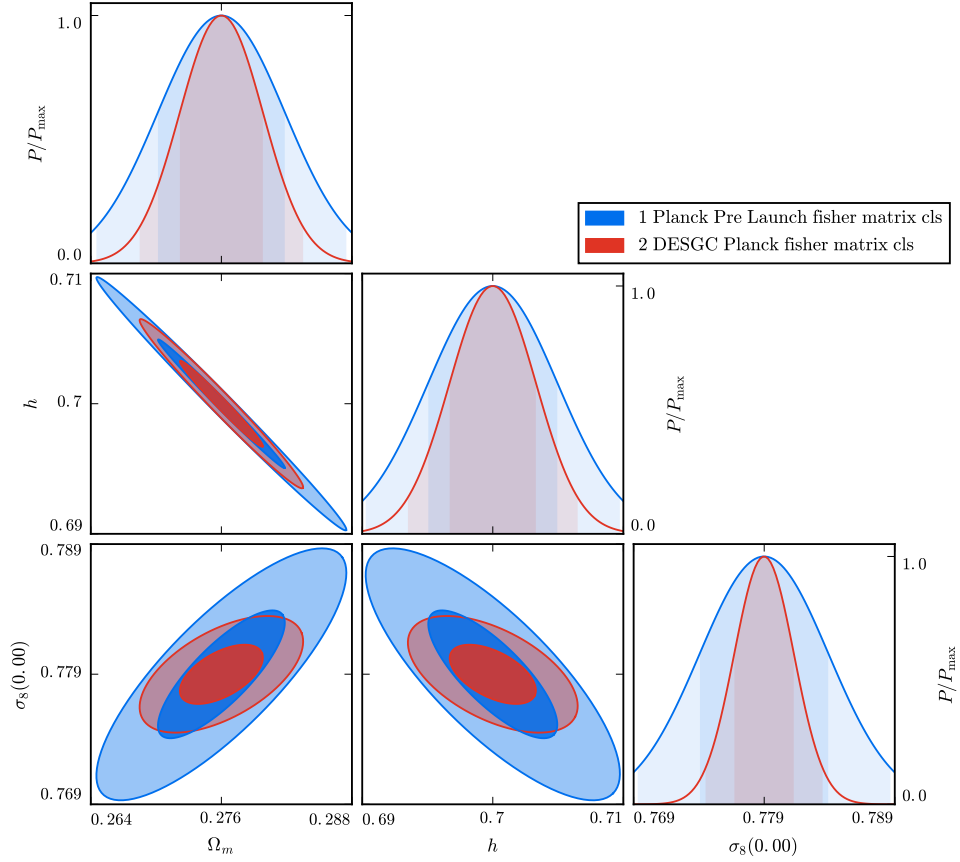
(a) *Planck* BB forecasted constraints(b) *Planck* BB and DES+*Planck* BB forecasted constraints

FIG. 3: The triangular plot of the marginalized 2D and 1D forecasted constraints on Ω_m , h and σ_8 . Different colors correspond to different combinations of experiments, as shown in legend. The darker and lighter color shades indicate the forecasted 68% C.L. and 95% C.L. bounds.