



<b>Publication Year</b>	2019
<b>Acceptance in OA</b>	2021-02-17T11:17:10Z
<b>Title</b>	Euclid preparation: II. The EuclidEmulator - A tool to compute the cosmology dependence of the nonlinear matter power spectrum
<b>Authors</b>	Euclid Collaboration, Mischa Knabenhans, Joachim Stadel, Stefano Marelli, Doug Potter, Romain Teyssier, Laurent Legrand, Aurel Schneider, Bruno Sudret, Linda Blot, Saeeda Awan, BURIGANA, CARLO, Carla Sofia Carvalho, Hannu Kurki-Suonio, Gabriele Sirri
<b>Publisher's version (DOI)</b>	10.1093/mnras/stz197
<b>Handle</b>	<a href="http://hdl.handle.net/20.500.12386/30424">http://hdl.handle.net/20.500.12386/30424</a>
<b>Journal</b>	MONTHLY NOTICES OF THE ROYAL ASTRONOMICAL SOCIETY
<b>Volume</b>	484

# *Euclid* preparation: II. The EUCLIDEMULATOR – a tool to compute the cosmology dependence of the nonlinear matter power spectrum

Euclid Collaboration, Mischa Knabenhans<sup>1</sup>,<sup>\*</sup> Joachim Stadel<sup>1</sup>, Stefano Marelli<sup>2</sup>,<sup>\*</sup> Doug Potter<sup>1</sup>, Romain Teyssier<sup>1</sup>, Laurent Legrand<sup>3</sup>, Aurel Schneider<sup>4</sup>, Bruno Sudret<sup>2</sup>, Linda Blot<sup>5,6</sup>, Saeeda Awan<sup>7</sup>, Carlo Burigana<sup>8,9,10</sup>, Carla Sofia Carvalho<sup>11</sup> and Hannu Kurki-Suonio<sup>12</sup> Gabriele Sirri<sup>13</sup>

*Affiliations are listed at the end of the paper*

Accepted 2019 January 15. Received 2019 January 2; in original form 2018 September 12

## ABSTRACT

We present a new power spectrum emulator named *EuclidEmulator* that estimates the nonlinear correction to the linear dark matter power spectrum depending on the six cosmological parameters  $\omega_b$ ,  $\omega_m$ ,  $n_s$ ,  $h$ ,  $w_0$ , and  $\sigma_8$ . It is constructed using the uncertainty quantification software *UQLab* using a spectral decomposition method called polynomial chaos expansion. All steps in its construction have been tested and optimized: the large high-resolution  $N$ -body simulations carried out with *PKDGRAV3* were validated using a simulation from the *Euclid* Flagship campaign and demonstrated to have converged up to wavenumbers  $k \approx 5 h \text{ Mpc}^{-1}$  for redshifts  $z \leq 5$ . The emulator is based on 100 input cosmologies simulated in boxes of  $(1250 \text{ Mpc}/h)^3$  using  $2048^3$  particles. We show that by creating mock emulators it is possible to successfully predict and optimize the performance of the final emulator prior to performing any  $N$ -body simulations. The absolute accuracy of the final nonlinear power spectrum is as good as one obtained with  $N$ -body simulations, conservatively,  $\sim 1$  per cent for  $k \lesssim 1 h \text{ Mpc}^{-1}$  and  $z \lesssim 1$ . This enables efficient forward modelling in the nonlinear regime, allowing for estimation of cosmological parameters using Markov Chain Monte Carlo methods. *EuclidEmulator* has been compared to *HALOFIT*, *CosmicEmu*, and *NGenHalofit*, and shown to be more accurate than these other approaches. This work paves a new way for optimal construction of future emulators that also consider other cosmological observables, use higher resolution input simulations, and investigate higher dimensional cosmological parameter spaces.

**Key words:** methods: numerical – methods: statistical – cosmological parameters – large-scale structure of Universe.

## 1 INTRODUCTION

Next generation cosmological surveys of large-scale structure such as DES<sup>1</sup> (The Dark Energy Survey Collaboration 2005), Euclid<sup>2</sup> (Laureijs et al. 2011), LSST<sup>3</sup> (LSST Science Collaboration 2009), and WFIRST<sup>4</sup> (Green et al. 2012) will exploit the highly nonlinear domain in order to vastly improve upon current precision estimates of cosmological parameters coming from cosmic microwave background (CMB) experiments such as Planck (Tauber et al. 2010;

Planck Collaboration XIII 2015) and WMAP (Bennett et al. 2003). *Euclid*, to be launched by ESA in 2021, will measure the matter distribution in the Universe over most of its cosmic history (up to a redshift  $z \approx 2.3$ ). Dark matter (DM), dark energy, and neutrino mass are currently the biggest challenges to modern physics. *Euclid* will be one of the first missions to shed light on this dark sector, provided it manages to fully exploit the highly nonlinear scales of this large-scale structure. It is not just an observing challenge, but also a theory challenge that is laid down by these new large-scale structure surveys.

The theory delving into this highly nonlinear domain is extremely complex and computationally expensive as the desired level of accuracy is currently only achieved by cosmological  $N$ -body simulations. Such simulations are very expensive since both large simulation volumes and large numbers of particles are needed

\* E-mail: [mischa.knabenhans@uzh.ch](mailto:mischa.knabenhans@uzh.ch)

<sup>1</sup> [www.darkenergysurvey.org](http://www.darkenergysurvey.org)

<sup>2</sup> [sci.esa.int/euclid](http://sci.esa.int/euclid)

<sup>3</sup> [www.lsst.org/lsst](http://www.lsst.org/lsst)

<sup>4</sup> [wfirst.gsfc.nasa.gov](http://wfirst.gsfc.nasa.gov)

to reach the required precision. It is therefore mandatory to have theoretical tools able to much more rapidly predict cosmological observables on these small, highly nonlinear scales at an accuracy level of better than 1 per cent (Huterer & Takada 2005). Even elaborate perturbation theory techniques break down below scales of  $x \lesssim 10 h^{-1} \text{Mpc}$  or  $k \gtrsim 0.6 h \text{Mpc}^{-1}$  (Carrasco et al. 2014). Fast, accurate, and easy-to-use emulators like `EuclidEmulator` presented in this paper are critical to the success of large-scale structure surveys.

Cosmic emulators provide a fast alternative to reliably predict cosmological observables, needing only a very small number of high-precision  $N$ -body simulations during their construction. Recent examples include `FRANKENEMU`, based on the Coyote Universe simulations presented by Heitmann et al. in Heitmann et al. (2010a,b); Lawrence et al. (2010b); Heitmann et al. (2014); `CosmicEmu` (Lawrence et al. 2010a), based on the Mira-Titan simulation suite discussed in Heitmann et al. (2016); Lawrence et al. (2017); and the *Aemulus* project introduced by DeRose, McClintock, Zhai et al. in DeRose et al. (2018); McClintock et al. (2018); Zhai et al. (2018). Emulation makes use of pre-evaluated simulations for a relatively small set of cosmologies in a given parameter space. Having these data available, a surrogate model for a desired cosmological observable can be computed. This surrogate model computes the desired quantity for a given input cosmology within fractions of a second on a usual desktop machine.

Applications, such as Monte Carlo approaches for parameter space searches and forward modelling of cosmological observations, become feasible. This then also allows for likelihood sampling and thus for forecasting of Fisher matrices and Kullback–Leibler divergences (Kullback & Leibler 1951; Amendola et al. 2018). Cosmological emulators can hence be used to accurately estimate the tightness of an error ellipsoid (referred to as ‘Figure of Merit’), and thus are an important tool to maximize the science output of such large-scale projects.

Baryonic effects, such as cooling and feedback, complicate the study of matter clustering at medium and small scales because so far there is no self-consistent treatment of the relevant processes in the cosmological context. Recent hydrodynamical simulations report a suppression of power of the order of 10–30 per cent at medium scales ( $0.2 < k < 10 h \text{Mpc}^{-1}$ ) followed by a strong enhancement at very small scales ( $k > 10 h \text{Mpc}^{-1}$ ) (Van Daalen et al. 2011); the latter is a consequence of baryon cooling and star formation in the halo centres. While most simulations reproduce this general trend, there is currently no agreement at the quantitative level. Some simulations predict a relatively small suppression affecting modes above  $k > 1 h \text{Mpc}^{-1}$  only (Hellwing et al. 2016; Chisari et al. 2018; Springel et al. 2018), others show a much stronger effect impacting modes above  $k > 0.1 h \text{Mpc}^{-1}$  (Van Daalen et al. 2011; Vogelsberger et al. 2014; Mummery et al. 2017).

The lack of agreement between different hydrodynamical simulations poses a serious challenge for future weak lensing and galaxy clustering surveys. Only if all baryonic effects can be controlled at the level of a few per cent will it be possible to fully exploit the potential of future galaxy surveys like *Euclid*. Recently, it has been shown that the amplitude of the baryon power suppression can be constrained with X-ray and Sunyaev–Zel’dovich observations of gas around galaxy groups and clusters (Schneider & Teyssier 2015; McCarthy et al. 2017; Mummery et al. 2017). This means that it is possible to come up with models to parametrize baryonic effects and calibrate them against observations (Semboloni et al. 2011; Zentner et al. 2013; Schneider & Teyssier 2015; McCarthy et al.

2017). These models can be encoded in a baryonic correction to the nonlinear power spectrum (sometimes referred to as the baryonic boost factor) that we hope to add to the analysis at a later stage.

Davis & Peebles (1983), Kaiser (1984), Bardeen et al. (1986), and others have shown that galaxies cluster significantly differently than DM and hence a thorough understanding of this so-called galaxy bias is crucial in order to compare observations to theoretical predictions based on DM simulations. While this bias is not part of the work presented in this publication, in the third paper of the *Aemulus* project series (Zhai et al. 2018) an emulation approach for the galaxy correlation function (and accordingly for the galaxy bias) is presented. They show that these quantities can be emulated by adding the relevant parameters to the cosmological parameter space, assuming that the halo occupation distribution (HOD) approach is sufficient to model the galaxy bias.

In this paper we present a new cosmic emulator for the nonlinear correction factor, i.e. the ratio between the nonlinear and the linear contribution of the matter power spectrum. This quantity is advantageous for three reasons: first, emulating the nonlinear correction factor is more accurate than emulating the power spectrum directly. Recall that the linear power spectrum can be computed exactly using Boltzmann solvers like `CAMB` (Lewis, Challinor & Lasenby 2000) or `CLASS` (Blas, Lesgourgues & Tram 2011) and hence the product of such a linear power spectrum and an emulated nonlinear correction is more accurate than a directly emulated nonlinear power spectrum. Secondly, full transparency for all steps involved in the power spectrum estimation is maintained, as both the linear power spectrum and its nonlinear correction are accessible in the entire emulation process. Thirdly, as the nonlinear correction factor is multiplied by a linear power spectrum, the latter may feature additional physics that is not included in the nonlinear correction. As an example, a nonlinear correction factor emulated based on the six-parameter model (as laid out in this paper) still allows for a final nonlinear power spectrum that includes neutrino physics or general relativistic effects to linear order. Furthermore, the nonlinear correction factor approach provides a framework that can be easily extended at a later stage. For example, an additional correction describing the aforementioned baryon effects could be readily added to a future version of the emulator. For now we focus on the six-parameter model inspired by Planck2015 (Donzelli, Dor & Gregorio 2016) including the baryon density  $\omega_b$ , the matter density  $\omega_m$ , the spectral index  $n_s$ , the reduced Hubble parameter  $h$ , the dark energy (DE) equation of state (EoS) parameter  $w_0$  and the variance  $\sigma_8$  in a first step. We leave further parameters that quantify mostly deviations from standard  $\Lambda$ CDM (Lambda cold dark matter) models (as e.g. time varying DE EoS  $w_a$ , neutrino density  $\omega_\nu$ , or primordial non-Gaussianity of the local type  $f_{\text{NL}}$ ) to subsequent studies.

In contrast to prior emulators (Heitmann et al. 2010a,b; Lawrence et al. 2010b; Heitmann et al. 2014, 2016; Lawrence et al. 2017; DeRose et al. 2018; McClintock et al. 2018; Zhai et al. 2018) that use *Kriging* (Santner, Williams & Notz 2013), a Gaussian process interpolation technique, we use regression between the sample cosmologies using sparse polynomial chaos expansion (SPCE), discussed e.g. in Blatman & Sudret (2011). Choosing this emulation technique we decrease the global maximal error of our emulator compared to a *Kriging* emulator. As we will find in Section 3, a sample of the cosmological parameter space [which in the field of uncertainty quantification, from now on abbreviated as UQ, is commonly referred to as the *experimental design* (ED)] with 100 points being enough to achieve a global maximal emulation-only error (EOE; i.e. the relative error between the emulated nonlinear correction spectrum and the nonlinear

correction spectrum computed from a full  $N$ -body simulation) below 1 per cent. In order to assess how the uncertainties on the input parameters affect the output observables, we use a state-of-the-art uncertainty quantification software called UQLab (Marelli & Sudret 2014).

Further, we are the first to apply pairing-and-fixing techniques (Angulo & Pontzen 2016) together with an extension of the algorithm presented in Jing (2005) on piece-wise cubic spline (PCS) mass assignment (Sefusatti et al. 2016) to pre-process the input cosmological simulations. This strategy allows us to drastically reduce numerical effects such as computational cosmic variance in the low  $k$  regime and aliasing effects near the sampling Nyquist frequency.

In this work we mainly focus on the emulation strategy and how it can be optimized. As the power spectrum is a very fundamental quantity and because it is very natural to emulate, we choose it as our observable of interest. Emulation of other observables can and will be investigated in subsequent work.

This paper is structured as follows: in Section 2 the input simulations of the emulator and the applied optimization techniques are discussed. Then, in Section 3, we investigate the actual construction and calibration of the emulator whose performance is assessed in Section 4. We summarize and conclude in Section 5. We list the codes and acronyms used in this work together with short explanations in the ‘Glossary’ section.

## 2 INPUT COSMOLOGICAL SIMULATIONS

For the construction of an emulator, a full suite of high-quality cosmological simulations serves as the input data set. As will be discussed in Section 4, in our approach, the simulation errors are the dominant contribution to the uncertainties in the final emulated nonlinear correction. As a consequence, the production of these data is not only very expensive but also challenging considering the tight bounds of 1 per cent on the power spectrum estimation set by the *Euclid* mission. Here, we describe a number of applied optimization techniques that allow us to reduce the computational time by roughly a factor of 5 compared to a standard  $N$ -body simulation approach without any decrease in the quality of the data.

The *EuclidEmulator* predicts the nonlinear correction  $B(k, z)$  of the DM power spectrum defined as

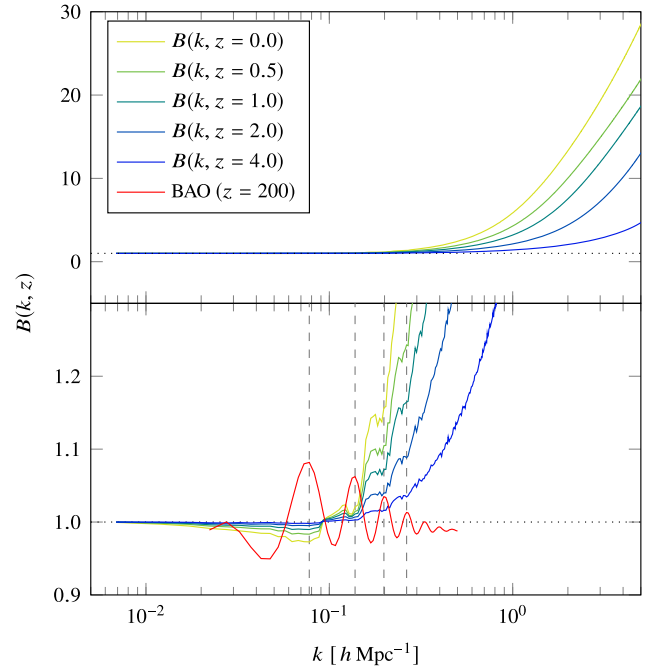
$$B(k, z) := \frac{P_{\text{nl}}(k, z)}{P_{\text{lin}}(k, z)}, \quad (1)$$

which divides the nonlinear by the linear DM power spectrum. An example of nonlinear correction is shown in Fig. 1, where the expected (Eisenstein et al. 2005; Crocce & Scoccimarro 2008) damping and broadening of the Baryon acoustic oscillation (BAO) wiggles are evident. On  $k \lesssim 0.1 \text{ h Mpc}^{-1}$  there is a clear nonlinear suppression of power corresponding to pre-virialization (Davis & Peebles 1977; Peebles 1990; Jain & Bertschinger 1994), which can also be understood as the nonlinear growth of voids at these scales.

The quality and performance of the emulator are highly dependent on the sampling of the cosmologies for which the  $N$ -body simulations are run. This sample of input cosmologies is called the ED. In this section the simulation strategies for the computation of the ED are explained.

### 2.1 Simulation of the ED

The ED was computed performing  $N$ -body simulations of the nonlinear matter power spectrum for a sample of 100 input cosmologies



**Figure 1.** The nonlinear correction to the power spectrum of the Euclid reference cosmology at different redshifts. While the overall behaviour of the nonlinear correction is smooth (top panel), a zoom-in (bottom panel) reveals clear structure like the pre-virialization dip and the nonlinear suppression of the linear baryon acoustic oscillations (BAOs; indicated by the red line in the bottom panel). In the bottom it is nicely visible how the nonlinear features become more and more distinct over time. The grey dashed vertical lines make clear that the minima of the nonlinear damping features coincide perfectly with the BAO maxima.

**Table 1.** Cosmological parameters of the Euclid reference cosmology.

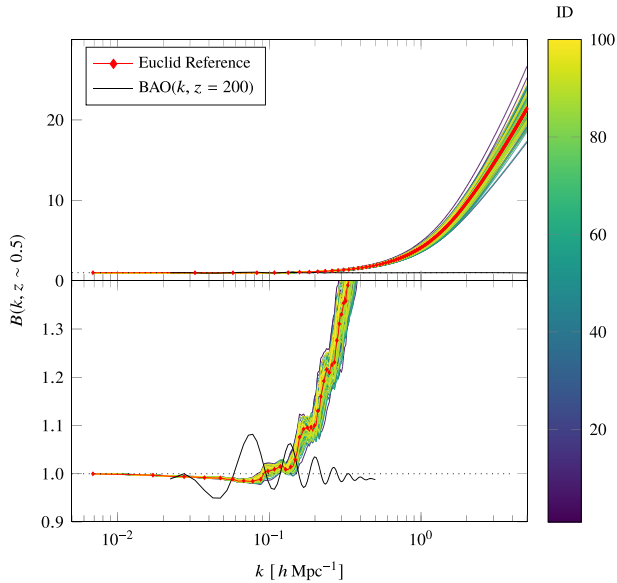
$\Omega_b$	$\Omega_m$	$n_s$	$h$	$w_0$	$\sigma_8$
0.049	0.319	0.96	0.67	-1.0	0.83

using the code PKDGRAV3 (Stadel 2001; Potter & Stadel 2016). Each simulation started at the initial redshift  $z_i = 200$  and evolved up to the final redshift  $z_f = 0$  in 100 base time-steps (smaller individual substeps are also used). Further details about the simulations will be discussed in Section 2.2. As our surrogate model emulates the nonlinear correction, the last step of the process in building the ED is to compute the nonlinear correction for each simulated cosmology (further explanations in Section 2.2.4).

Convergence testing of the power spectrum (see Appendix C) was performed on the Euclid reference cosmology (Table 1) for which we had available a much higher resolution simulation (part of the Euclid Flagship simulation campaign, see Potter, Stadel & Teyssier 2017). However, the results at this particular reference cosmology are *not* included in the ED. In Fig. 2 the set of 100 + 1 nonlinear correction curves (including the nonlinear correction of the Euclid reference cosmology) corresponding to the 60th time-step (equivalent to a redshift  $z \sim 0.5$ ) is shown.

### 2.2 Cosmological $N$ -body simulation

The matter power spectrum is well understood to linear order, i.e. in the regime where the DM overdensities in the Universe are



**Figure 2.** The ED used as input for the construction of the *EuclidEmulator*: nonlinear correction curves for each input cosmology and the Euclid reference cosmology (red line with diamond markers) at output step 60 (roughly  $z \sim 0.5$ , depending slightly on the cosmology). The colour bar labels the 100 different cosmologies in the ED via their IDs. It is clearly visible that the Euclid reference cosmology is reasonably near the centre of the parameter space as all nonlinear corrections of the input cosmologies scatter nicely around it. The black oscillatory curve in the lower panel indicates the BAO at high redshift, as in Fig. 1.

small enough to allow a valid description in terms of linearized fluid equations. Apart from higher order perturbative approaches, there are no precise analytical means to calculate the nonlinear power spectrum. Several codes provide fast computations of these higher order corrections (Crocco, Scoccimarro & Bernardeau 2012; McEwen et al. 2016; Fang et al. 2017) but all of them break down in the weakly nonlinear regime (Carlson, White & Padmanabhan 2009). This is where  $N$ -body simulations come in. These codes are a direct first principles approach for simulating the process of cosmic structure formation by numerically evolving the density field.

### 2.2.1 The PKDGRAV3 $N$ -body code

The main cost of producing an emulator of the nonlinear power spectrum is in performing the needed simulations over the cosmological parameter space. For this reason, it is important to have available a fast and accurate  $N$ -body code. We have used the publicly available (Potter & Stadel 2016)  $N$ -body code PKDGRAV3 by Potter et al. (2017) and Stadel (2001). PKDGRAV3 is a parallel fast multipole method tree code, which uses a block-step multiple time-stepping scheme for the integration of the equations of motion for the particles. It uses 5th order multipole expansions of the potential in calculating the force due to all the other particles, as well as for the calculation of periodic boundary conditions.

PKDGRAV3 has been validated (Schneider et al. 2016) against two other well-established  $N$ -body codes, namely GADGET3 (for an older version of the code, see Springel 2005) and RAMSES (Teyssier 2010). From this comparison, we know that the absolute accuracy of power spectra generated with PKDGRAV3 is better than 1 per cent for  $k \lesssim 1 h \text{Mpc}^{-1}$  and  $z \lesssim 1$  (at  $z = 0$ , 1 per cent accuracy is achieved up to  $6 h \text{Mpc}^{-1}$ ). PKDGRAV3 is very memory efficient,

allowing for large simulations to fit on a relatively small number of nodes. In our case the  $2048^3$  simulations fit comfortably on 16 nodes. Each simulation on 16 nodes (each node having 64 GB of RAM, 16 cores, and no GPU) took almost exactly 3 days to complete. This adds up to 190 000 node hours to complete all  $2 \times 100$  simulations used as input for our emulator (the factor of 2 comes from the pairing-and-fixing described in Section 2.2.2).

### 2.2.2 Pairing-and-fixing of initial conditions

A common issue in power spectrum estimations in numerical simulations is the computational cosmic variance arising from the finiteness of the simulation box: very small Fourier modes (or equivalently very large physical distances) are undersampled. This leads to a loss of information or, put differently, to a large variance in statistical quantities like the power spectrum. We do not want to include any contributions due to this effect within our emulator.

We have used two techniques to reduce contributions from this sampling variance for the suite of input simulations. First, it is possible to reduce these contributions at the linear level by computing the nonlinear correction in a specific way, namely by dividing the nonlinear power spectrum at redshift  $z$  by the properly rescaled initial output power spectrum of the very same simulation. Here, a properly rescaled initial power spectrum is obtained by taking the linear power spectrum at redshift  $z = 0$  and scaling it back to the initial redshift  $z_i$  using the linear growth factor. However, due to mode coupling in the nonlinear evolution, the sampling variance still propagates somewhat from larger to smaller scales. This phenomenon adds a sample variance contribution to the power spectrum that remains despite the described division procedure. Secondly, to further improve on this, we apply the method of phase pairing and power spectrum amplitude fixing (hereafter ‘pairing-and-fixing’) described in Angulo & Pontzen (2016). This method is able to drastically reduce the computational cosmic variance and shall briefly be reviewed here: we use a fixed, linear input power spectrum  $P_i$  (computed, e.g. with CLASS) and draw the initial overdensity fields  $\delta_{i,\text{lin}}$  that can be decomposed into a magnitude  $|\delta_{i,\text{lin}}|$  and a phase  $\theta_i$  according to the probability distribution function given by

$$P(|\delta_{i,\text{lin}}|, \theta_i) = \prod_i \frac{1}{2\pi} \delta_{\text{D}}(|\delta_{i,\text{lin}}| - \sqrt{P_i}), \quad (2)$$

with  $\delta_{\text{D}}$  being the Dirac delta function and the index  $i$  labels the Fourier modes. This probability distribution function identifies uniquely the magnitude of  $\delta_{i,\text{lin}}$  (it is ‘fixed’) while the phase is still uniformly random between 0 and  $2\pi$  such that one obtains

$$\delta_{i,\text{lin}} = \sqrt{P_i} e^{i\theta_i}. \quad (3)$$

For a comparison between paired-and-fixed simulations against traditional Gaussian random initial condition-based simulations, we refer the reader to Appendix D. Following this algorithm, we generate two initial conditions per set of cosmological parameters, both having the same magnitudes  $|\delta_{i,\text{lin}}| = \sqrt{P_i}$  but the phases being shifted by  $\pi$  with respect to each other, i.e. we draw the first phase  $\theta_{i,1}$  randomly and set  $\theta_{i,2} = \theta_{i,1} + \pi$  for the second initial condition. For the generation of each initial condition, we use the transfer function at  $z = 0$  (from CLASS) and scale it back to high redshift ( $z_i = 200$ ). Particle displacements are then set using the Zel’dovich approximation (ZA). We then perform a simulation for each of the two initial conditions and measure the power spectra.

ZA was chosen over 2LPT for computing the displacement field due to the fact that a version of PKDGRAV3 that correctly accounted

for relativistic fluids with 2LPT was not available at the time. Current development versions of PKDGRAV3 address this and will allow us to avoid the very high redshift starts using ZA, thereby minimizing systematic effects due to discreteness without loss of accuracy. In principle, 2LPT starts at lower redshift are favoured and will be considered in future work.

The resulting power spectra are then averaged (‘paired’) and the nonlinear correction is subsequently computed from the paired power spectra (for a deeper discussion of this nonlinear correction computation, see Section 2.2.4). We find that, on large and intermediate scales where computational cosmic variance poses a problem, a nonlinear correction computed with this algorithm is comparable to a nonlinear correction coming from a power spectrum ensemble averaged over 10 realizations. For a more detailed analysis of this algorithm and its performance, we refer to Angulo & Pontzen (2016) and Pontzen et al. (2016). We found that we could reduce the computational effort by at least a factor of 5 using this method of pairing fixed simulations over conventional ensemble averaging.

### 2.2.3 4th-order mass assignment

As the code PKDGRAV3 evolves particles in a tree, their mass needs to be assigned to a grid whenever the power spectrum is computed. The mass assignment scheme has a non-negligible impact on the quality of the power spectrum, particularly on nonlinear scales. While 2nd- and 3rd-order (cloud-in-cell and triangular shaped cloud) mass assignment schemes are widely used in simulations, we use 4th-order PCS mass assignment as in Sefusatti et al. (2016). Although the time required for the mass assignment with this technique is increased, the errors in the power spectrum are substantially reduced.

### 2.2.4 Post-processing: computing the nonlinear correction

The main advantage of emulating the nonlinear correction over full power spectrum emulation is that  $B(k, z) = 1$  on linear scales for all redshifts. This allows one to multiply it by a linear power spectrum that includes more physics on these large scales than can be explained by the relatively limited six-parameter model used for the nonlinear correction computation itself (a prominent example is given by the superhorizon damping of the matter power spectrum captured, e.g. by the Boltzmann codes CAMB and CLASS). An added benefit is that emulating the logarithm of the nonlinear correction appears to be almost an order of magnitude more precise than emulating the raw power spectrum, as is shown in Fig. 3 and in Section 3.

Having access to both the linear power spectrum from Boltzmann solvers like CAMB or CLASS and the power spectra from  $N$ -body simulations at all time-steps, there are two different possible ways to compute the nonlinear correction:

- (i) Take the nonlinear power spectrum simulated by the  $N$ -body code and divide it by the linear input power spectrum computed with a Boltzmann solver like CAMB or CLASS.
- (ii) Divide the nonlinear power spectrum at redshift  $z$  by the properly rescaled quasi-linear power spectrum at the initial redshift  $z_i$  of the  $N$ -body code.

We follow the second approach for two reasons: first, this is the only approach where the nonlinear correction is actually equal to 1 (as visible in Figs 1 and 2) for low  $k$ -values as required by the argument stated above. This would not be achieved if one

divided by the linear power spectrum computed with a Boltzmann solver. Secondly, as mentioned in Section 2.2.2, the former division already cancels out a considerable amount of computational cosmic variance.

We show in Section 3.4 that only 100 cosmologies need to be simulated to achieve a maximal error of less than 1 per cent over the  $k$ -range of interest  $0.01 h \text{ Mpc}^{-1} \leq k \leq 5 h \text{ Mpc}^{-1}$ . We run two simulations per ED sampling point in a  $1250 h^{-1} \text{ Mpc}$  box with  $2048^3$  particles, each with fixed initial conditions starting at a redshift  $z_i = 200$  and evolving to the present day ( $z_f = 0$ ). We produce nonlinear 1D power spectrum outputs for 100 time-steps (equidistantly spaced in time) along the way. In a next step we average the power spectra over each pair of simulations ( $P_1$  and  $P_2$ ) and subsequently compute the nonlinear correction spectrum at a certain redshift  $z$  by dividing the averaged nonlinear power spectrum at redshift  $z$  by the averaged nonlinear power spectrum at initial redshift. We thus compute

$$B(k, z) = \frac{\frac{1}{2}[P_1(k, z) + P_2(k, z)]}{\frac{1}{2}[P_1(k, z_i) + P_2(k, z_i)]} \left( \frac{D_{\text{ILPT}}(z_i)}{D_{\text{ILPT}}(z)} \right)^2 \quad (4)$$

instead of averaging the nonlinear corrections themselves (in this equation,  $D_{\text{ILPT}}$  denotes the scale-independent ILPT growth factor). In a comparison of these two calculation strategies, they turned out to agree almost perfectly (to within less than 0.1 per cent over all wavenumbers of interest). Now we have an ED of  $n_{\text{ED}} = 100$  nonlinear correction spectrum sets each with  $n_z = 100$  different nonlinear correction spectra (one for each redshift output step in the simulations) measured at  $n_k = 2000$  different linearly spaced  $k$ -points.

## 2.3 Convergence of simulations

As will be discussed below, the main contribution to the overall emulation error is due to the underlying simulations. We have performed a convergence test using different box sizes with an edge length  $L$  between 480 and  $1920 h^{-1} \text{ Mpc}$ , with different particle numbers  $N^3$  ranging from  $1024^3$  to  $2048^3$  and with different grid resolutions (once, twice, or four times as many grid points as particles per dimension). For reference, two simulations have been used: a large volume simulation with a  $(4000 h^{-1} \text{ Mpc})^3$  box with  $4096^3$  particles for assessing the minimally required simulation volume and a high-resolution run with  $8000^3$  particles in a  $(1920 h^{-1} \text{ Mpc})^3$  box to find the minimal mass resolution. We found that simulations with  $L^3 = (1250 h^{-1} \text{ Mpc})^3$  and  $N^3 = 2048^3$  particles (corresponding to a mass resolution of roughly  $2 \times 10^{10} h^{-1} M_{\odot}$  per particle) have converged to the level of accuracy required, if a power spectrum measurement grid with roughly double this resolution is used. Using these specifications, we find that the simulated nonlinear correction spectra have converged up to  $k_{\text{max}} = 5.48 h \text{ Mpc}^{-1}$  for all redshifts  $z \leq 5$  (reducing the number of  $k$ -points to 1100). For further details about the convergence tests, please refer to Appendix C.

## 3 EMULATOR CONSTRUCTION AND CONFIGURATION

The emulated data are supposed to approximate simulations as accurately as possible. Accuracy, however, comes at the expense of higher cost in the construction of the emulator, or can result in an increase of the time and resources needed in the use of the emulator. In this section we will highlight the important aspects that influence

the performance and the efficiency of the emulator and discuss how the EOE can be reduced while keeping the overall costs for the construction of the emulator manageable. We define the EOE as follows:

$$\text{EOE}_{\mathbf{c}}(k, z) = \frac{B_{\mathbf{c}}^{\text{emulated}}(k, z)}{B_{\mathbf{c}}^{\text{simulated}}(k, z)} - 1, \quad (5)$$

where  $k$  is the wavenumber,  $z$  is the redshift, and  $\mathbf{c}$  stands for a cosmology for which the nonlinear correction is evaluated. The steps involved in the construction of `EuclidEmulator` are as follows:

- (i) Definition of the cosmological parameter space, in our case, a six-dimensional box over which a uniform prior is assumed,
- (ii) Latin hypercube sampling (LHS) of the parameter space,
- (iii)  $N$ -body simulation of all cosmologies in the Latin hypercube (LH) sample,
- (iv) Computation of the nonlinear correction spectra (this data set in its entirety is called the ED),
- (v) principal component analysis (PCA) of the nonlinear corrections,
- (vi) polynomial chaos expansion (PCE) of each individual eigenvalue of the PCA, neglecting polynomial terms based on the sparsity-of-effects principle (hence SPCE using UQLab),
- (vii) Recombination of the principal components in a single emulator (using UQLab or our own stripped down C code).

For actually using `EuclidEmulator` to produce nonlinear power spectra, one only needs to combine step (vii) with a linear power spectrum generated by the `CLASS` or `CAMB` Boltzmann codes.

Redshift is not an emulated parameter and the ED data matrix  $\mathbf{D}$  contains a specific set of 100 nonlinear correction spectra at different, cosmology dependent, redshifts (one for each output step of the simulations). To allow the computation of the nonlinear correction at any requested  $z$ -value, we linearly interpolate between two adjacent nonlinear correction spectra that bracket this redshift. By doing so, we commit the biggest error at the maximal redshift (because the input simulations are distributed less densely in redshift space towards higher redshifts) and maximal  $k$ -mode (as the change in nonlinear correction is larger per  $z$ -interval for larger  $k$ -values) allowed by the emulator. We have tested that this maximal error is  $\sim 0.6$  per cent. For all smaller  $k$ -modes and redshift values, the error due to linear interpolation is smaller. Higher order interpolation over the data matrix  $\mathbf{D}$  would make such errors at high  $z$  and  $k$  negligible, but since this consideration lies outside of the emulation strategy, we do not further consider it here.

### 3.1 ED sampling

The performance of the emulator crucially depends on how the ED is constructed (Blatman & Sudret 2011). The construction of the ED involves steps (i)–(iii). In this subsection these three phases shall be explained in more detail.

#### 3.1.1 Definition of the parameter space

Similar to Lawrence et al. (2010b), our emulator is built upon the six-parameter model including the following cosmological parameters:

- (i) baryonic matter density parameter in the Universe,  $\omega_b = \Omega_b h^2$ ,
- (ii) total matter density parameter in the Universe,  $\omega_m = \Omega_m h^2$ ,
- (iii) reduced Hubble parameter  $h$ ,
- (iv) spectral index  $n_s$ ,

- (v) EoS parameter of dark energy  $w_0$ ,
- (vi) power spectrum normalization  $\sigma_8$ ,

where we assume a flat geometry of the Universe throughout ( $\Omega_k = 1$ ) such that the dark energy density parameter  $\Omega_{\text{DE}}$  is uniquely defined by the relation

$$\Omega_m + \Omega_{\text{rad}} + \Omega_{\text{DE}} = 1, \quad (6)$$

where  $\Omega_{\text{rad}}$  has a constant non-zero value for all cosmologies. These parameters are a subset of the parameters of the base  $\Lambda$ CDM cosmology from Planck (Donzelli et al. 2016). A key goal of the *Euclid* mission is to further constrain the DE EoS (Amendola et al. 2018). For this reason,  $w_0$  has been added to the investigated parameter space. Further important physical processes relevant for power spectrum measurement are, amongst many others, the effect of neutrinos on DM clustering or the impact of a time-dependent DE EoS. Corresponding parameters have been included in the Mira-Titan Universe-based `CosmicEmu` (Heitmann et al. 2016; Lawrence et al. 2017) and will be included in future versions of `EuclidEmulator`.

We base our parameter box ranges on the Planck2015 best-fitting values mentioned in table 4 in Donzelli et al. (2016). For the parameters  $\omega_m$ ,  $n_s$ ,  $h$ , and  $\sigma_8$ , we use Planck-only data. However, as the constraining power of Planck for  $\omega_b$  and  $w_0$  is significantly improved by combining it with external data, we use the combined best-fitting values for bounding the ranges of these two parameters. The upper and lower bounds are defined by  $\mu \pm \Delta$ , where  $\Delta$  corresponds to  $6\sigma$  quoted in Donzelli et al. (2016) for all cosmological parameters but  $w_0$  ( $\Delta_{w_0} = 3.5\sigma$ ). The parameter ‘box’ thus is finally defined as follows:

$$\begin{aligned} \omega_b &\in [0.0215, 0.0235], \\ \omega_m &\in [0.1306, 0.1546], \\ n_s &\in [0.9283, 1.0027], \\ h &\in [0.6155, 0.7307], \\ w_0 &\in [-1.30, -0.70], \\ \sigma_8 &\in [0.7591, 0.8707]. \end{aligned} \quad (7)$$

We assume massless neutrinos for all cosmologies and take their energy contribution into account as a component of  $\omega_{\text{rad}}$ . The photonic contribution to it is related to the CMB temperature, which we set to  $T_{\text{CMB}} = 2.7255$  K. Our final  $\omega_{\text{rad}}$ , including both photons and massless neutrinos, is cosmology dependent (via  $h$ ). Since `CLASS` calculates this internally, we use its value for each `PKDGRAV3` simulation.

#### 3.1.2 Sampling

The parameter space constructed above then has to be sampled in such a way that on the one hand one ends up with an ED containing only a relatively small number of points (otherwise the computational cost to produce the corresponding simulations explodes) and on the other hand the emulator built on top of these simulations must return highly accurate results. For now we assume no preliminary knowledge about the behaviour of the emulated observable depending on the point in the parameter space. It is hence standard to use LHS (McKay, Beckman & Conover 1979; Tang 1993), which provides a fairly uniform coverage of the parameter space. For further discussion of statistical sampling techniques and their properties, see Heitmann et al. (2010a), section 2.1 and references therein.

In order to perform an LHS, one needs to define the number of sampling points in advance. As we describe in Section 3.4, 100 sampling points are enough in order to construct an emulator that reaches the required accuracy in the output quantities. This step is performed using the statistics and uncertainty quantification software UQLab<sup>5</sup> (Marelli & Sudret 2014). As the construction of such a sample is a random process and not unique at all, we add an optimization step by generating  $10^5$  different samples and choosing the realization for which the minimal distance (in Euclidean metric) between the sampling points is maximized (a classical maximin criterion, see e.g. Johnson, Moore & Ylvisaker 1990). By doing so we, e.g., avoid the unlikely event of all sampling points being aligned along the diagonal of the parameter space.

### 3.2 Principal component analysis

Simulation data are usually noisy and suffer from non-physical, spurious numerical signals. We want the emulated data to be free from these problems, which can be achieved by de-noising the input simulation data of the ED using PCA. The entire ED nonlinear correction spectrum data set  $\mathbf{D}$  can be represented as a  $n_{\text{ED}} \times (n_z \cdot n_k)$  matrix, where  $n_{\text{ED}}$  is the number of sampling points in the ED,  $n_z$  is the number of output steps per simulation, and  $n_k$  is the number of wavenumbers considered for the power spectrum measurement. It turns out that the overall EOE (the  $L_\infty$  norm over all  $k$  and  $z$ ) is drastically reduced (by roughly an order of magnitude) if we do not store the nonlinear correction values themselves into the data matrix  $\mathbf{D}$  but rather use the logarithm thereof. The data matrix  $\mathbf{D}$  is next decomposed into its  $n_{\text{ED}}$  principal components

$$\mathbf{D} = \sum_{i=1}^{n_{\text{ED}}} \lambda_i(\omega_b, \omega_m, n_s, h, w_0, \sigma_8) \text{PC}_i(k, z), \quad (8)$$

where the  $\lambda_i$ 's are the eigenvalues of the covariance matrix  $\text{cov}(\mathbf{D})$  and the principal component (PC)'s are its eigenmodes (vectors of length  $n_z \cdot n_k$ ).

### 3.3 The surrogate model

The main goal of emulation is to produce data in a simultaneously fast and precise way for all possible inputs. Therefore we choose to use PCE (a spectral representation on an orthonormal polynomial basis, see Ghanem & Spanos 2003; Xiu & Karniadakis 2006; Xiu 2010; Blatman & Sudret 2011) in contrast to Gaussian process modelling (aka Kriging) as done e.g. by Heitmann et al. (2016), Lawrence et al. (2017) and also by Zhai et al. (2018). This strategy minimizes the global error, but comes at the expense of not exactly retrieving the simulation data at the input cosmologies.

Since according to equation (8) all information about the cosmological parameters is stored in the eigenvalues  $\lambda_i$ , one only needs to find a surrogate for them. We therefore create a PCE of each component separately. In the case of EuclidEmulator, the expansion reads

$$\lambda_i(\omega_b, \omega_m, n_s, h, w_0, \sigma_8) \approx \sum_{\alpha \in \mathcal{A}} \eta_\alpha \Psi_\alpha(\mathbf{x}), \quad (9)$$

where  $\alpha = (\alpha_1, \dots, \alpha_6)$  denotes a multi-index,  $\Psi_\alpha$  the polynomial basis element, and  $\eta_\alpha$  the corresponding coefficient. Here

$\mathbf{x} = (x_1, \dots, x_6)$  is the vector of the six cosmological parameters, each mapped to the  $[-1, 1]$  interval. In practice the sum is truncated, making  $\mathcal{A}$  finite and the above equation an approximation. The multivariate basis functions can be expressed in terms of normalized Legendre polynomials like so:

$$\Psi_\alpha(\mathbf{x}) = \phi_{\alpha_1}(x_1) \phi_{\alpha_2}(x_2) \dots \phi_{\alpha_6}(x_6) = \prod_{l=1}^6 \sqrt{2\alpha_l + 1} P_{\alpha_l}(x_l). \quad (10)$$

The PCE coefficients  $\eta_\alpha$  are computed with the least-angle regression (LARS) algorithm (Efron et al. 2004; Blatman & Sudret 2011). This algorithm considers a set of candidate multivariate basis functions  $\mathcal{A}_{\text{cand}}$  defined by criteria related to the maximal total degree of polynomials  $p$ , the maximum interaction  $r$  (the number of non-zero values in the  $\alpha$  vector of size 6), and a sparsity-inducing  $q$ -norm as follows:

$$\mathcal{A}_{\text{cand}} = \left\{ \alpha : \left( \sum_{i=1}^6 \alpha_i^q \right)^{1/q} \leq p, \sum_{i=1}^6 1_{\alpha_i \neq 0} \leq r \right\}. \quad (11)$$

The least-angle regression then determines an optimal sparse set of polynomials  $\mathcal{A} \subset \mathcal{A}_{\text{cand}}$  such that a built-in error estimator on the truncated series equation (9) is minimized. For a deeper discussion of the least-angle regression-based selection (LARS) algorithm, we refer to Appendix B.2.2 or to the literature mentioned above. The final performance of the emulator is tightly related to how the series is truncated, which terms are taken into account, and which ones are dropped. The applied truncation scheme is a hybrid of hyperbolic and so-called maximum interaction truncation (Marelli & Sudret 2017). For more elaborate instructions about how to compute a PCE, we refer to the referenced literature.

### 3.4 Optimizing and projecting emulator performance

A proper configuration of the emulator is key for good performance. While a misconfigured emulator can introduce EOE's much larger than the simulation uncertainties (basically defeating the purpose of the emulator), a carefully configured surrogate model is able to introduce EOE's so small that they are negligible compared to simulation errors. Such an emulator is thus capable of producing effectively simulation-quality results (but at much lower cost).

#### 3.4.1 The configuration space

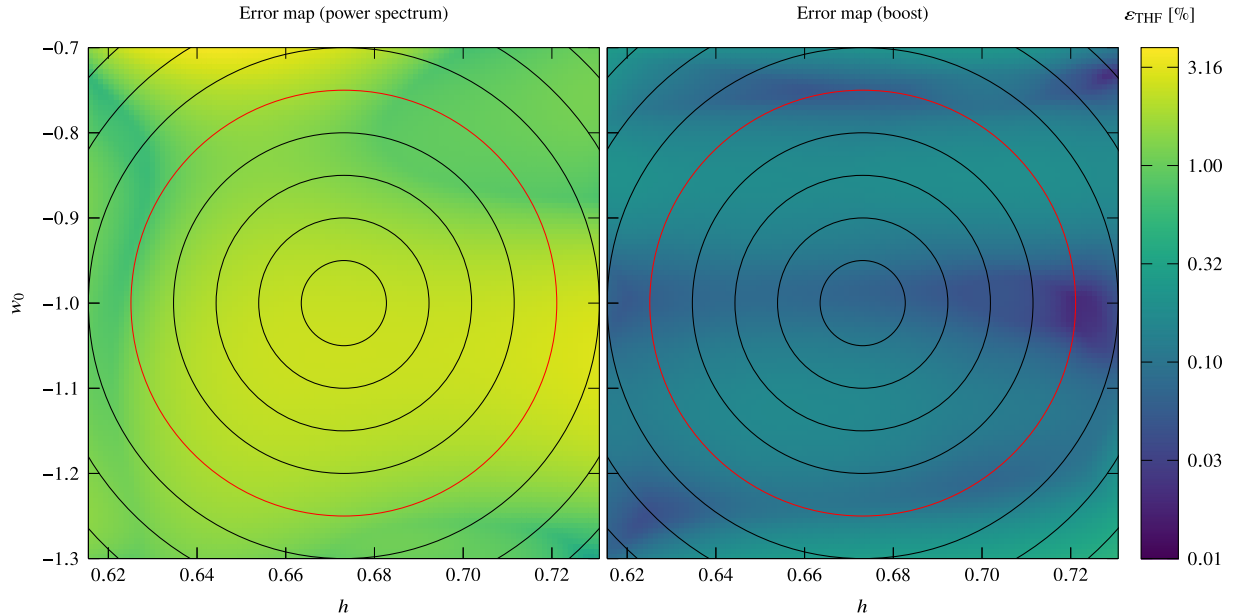
As we have discussed in the previous sections, the emulator construction process depends on various degrees of freedom (for a deeper discussion, we refer to Appendix B):

- (i) the number of sampling points  $n_{\text{ED}}$  in the ED,
- (ii) the truncation parameters  $p$ ,  $q$ , and  $r$  characterizing the multi-index set  $\mathcal{A}$ ,
- (iii) the accuracy parameter  $a$  defined as the fraction of the total variance captured by the principal components taken into account with respect to the total variance of the data. This is directly related to the number of principal components  $n_{\text{PCA}}$  taken into account.<sup>6</sup>

As briefly described in Appendix B, the maximal polynomial order  $p$  can be found following an iterative approach. This is done

<sup>6</sup>Notice that the accuracy parameter  $a$  is more fundamental than  $n_{\text{PCA}}$  since the latter additionally depends on further quantities as, e.g. the size of the data set.

<sup>5</sup><http://www.uqlab.com>



**Figure 3.** Error maps of the  $(h, w_0)$  plane based on an ED with  $n_{\text{ED}} = 100$  for direct emulation of the power spectrum (left-hand panel, similar to Heitmann et al. 2010b and Lawrence et al. 2017) and the logarithm of the nonlinear correction (right-hand panel). We stress that logarithms are just used to construct the emulator and the errors shown here in both panels are based on comparisons of the full nonlinear power spectrum/nonlinear correction. For the nonlinear correction emulation, the maximal error over all cosmologies is of the order of 0.5 per cent (in contrast to 3.7 per cent for direct power spectrum emulation) but if restricted to the  $0.83\Delta$ -ellipsoid (red ellipse), the maximal EOE drops to below 0.2 per cent (direct power spectrum emulation: 2.75 per cent). The black circles indicate the different  $\Delta$ -ellipsoids (the innermost ellipse corresponds to the  $0.17\Delta$ , the second to the  $0.33\Delta$ , etc.).

automatically by UQLab for every principal component separately and hence does not form part of the subsequent analysis. The remaining four parameters, though, need to be tuned carefully in order to optimize the emulator’s final performance (i.e. balance its accuracy against its efficiency). We investigated this 4D parameter space on a grid given by

$$\begin{aligned} n_{\text{ED}} &= 10, 25, 50, 75, 100, 250, \\ a &= 1 - 10^{-\kappa} \text{ with } \kappa = 1, 2, \dots, 10, \\ q &= 0.1, 0.2, \dots, 1.0, \\ r &= 2, 3, 4, 5, 6. \end{aligned} \quad (12)$$

For each of these 3000 grid points, an emulator was constructed and used to make predictions that in turn were tested against a comparison data set. The relative error between prediction and comparison data was recorded. The most precise of all these 3000 emulators has then been filtered out under the requirement that it also be computationally efficient, i.e. under 0.1 s on a single CPU core (comparable to the best Boltzmann codes).

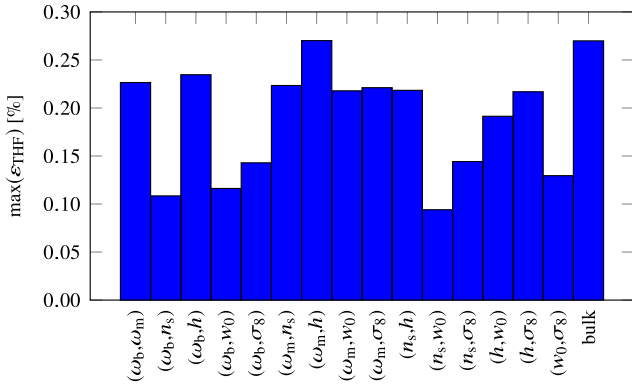
### 3.4.2 Comparison data set

Notice that since we are interested in studying the EOE, it is mandatory that both the ED and the comparison data set are computed the same way (as otherwise differences in the computation strategy could contaminate the EOE). As will become clear, this investigation requires more data than could be produced with  $N$ -body simulations. This is why we map out the error committed by emulating the nonlinear correction using CLASS (version 2.6.3) and Takahashi’s extension of HALOFIT (Smith et al. 2003; Takahashi et al. 2012) (hereafter abbreviated as THF) as an alternative surrogate technique. For each cosmological parameter, we have chosen 100 values equidistantly spread over the respective

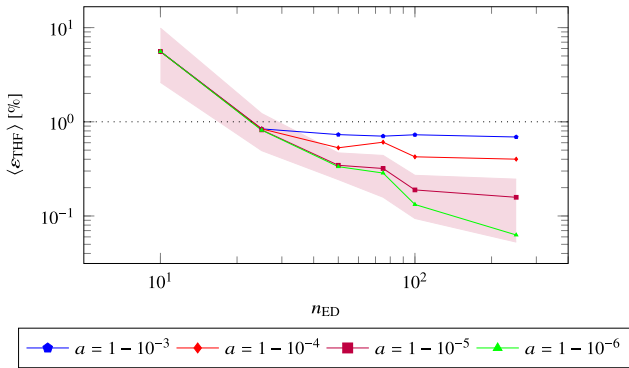
range resulting in a six-dimensional lattice with  $10^{12}$  points. As a computation of the nonlinear correction spectrum for 1 trillion cosmologies is not feasible even with the halo model, we restrict the further analysis to the 15 coordinate planes of the parameter space (i.e. the planes of pairs of cosmological parameters), each of which is sampled by  $10^4$  points, and a random sample of  $10^4$  cosmologies in the bulk of the entire six-dimensional space. For the resulting  $16 \times 10^4$  cosmologies, we compute the nonlinear correction curves for redshifts  $z = 10^{-16}, 1, 2,$  and  $5$  in the interval  $10^{-5} h \text{ Mpc}^{-1} \leq k \leq 10 h \text{ Mpc}^{-1}$ .

### 3.4.3 Comparisons and tests

The EDs are as well computed using CLASS and Takahashi’s halo model, one for each of the  $n_{\text{ED}}$  values, at the same four redshifts and for the same wavenumber intervals like in the comparison data set. We constructed emulators based on the logarithm of these nonlinear correction data. To predict the EOE (labelled as  $\varepsilon_{\text{THF}}$  to emphasize that it is based on the THF), we then evaluate the emulator at each lattice point of the cosmological coordinate planes, take the exponential of the emulation result in order to undo that logarithm and compare it to the data obtained directly from CLASS/THF. The relative error was subsequently computed and maximized over all redshifts and wavenumber values, and plotted as a density plot as shown in the example plot Fig. 3. We thus get one such density plot for each coordinate plane. At this point we emphasize that while emulating  $\log(B)$  instead of the nonlinear correction itself is an essential technical step improving the final accuracy of the emulator dramatically, it has no effect on how the comparison is presented. For the comparison of directly computed and emulated nonlinear correction spectra, the logarithm has always been undone. Since the largest emulation errors are often found close to the boundaries of the parameter box, we exclude these by restricting the emulator to



**Figure 4.** Maximal errors over all cosmologies in all coordinate planes and in the bulk of the space (restricted to the  $0.83\Delta$  region). This bar chart shows that a comparison of error maps as in Fig. 3 looks similar for all coordinate planes: the maximal error of EuclidEmulator over all coordinate planes is about 0.27 per cent in contrast to roughly 1 per cent of the FRANKENEMU (Heitmann et al. 2010b) and 4 per cent of the COSMICEMU (Lawrence et al. 2017).



**Figure 5.** Dependence of the relative error on the size of the ED and the accuracy parameter  $a$ . The truncation parameters  $r$  and  $q$  have been set to 2 and 0.5, respectively, as these values are used for the construction of the actual EuclidEmulator. While the curves represent the mean of the maximal error (i.e. the error maximized over all  $k$  and  $z$  and then averaged over all cosmologies in the 15 coordinate planes and the bulk), the red shaded region additionally indicates the maximal and minimal maximal errors measured for  $a = 1 - 10^{-5}$ . The curves plateau when the errors due to omission of terms dominate the errors coming from the sampling size.

lie within the inner  $0.83\Delta$ -ellipsoid ( $\Delta$  defined in Section 3.1.1). The region outside of this is no longer considered in what follows.

Fig. 4 shows the maximum error found within each of the 15 possible parameter planes, as well as over the entire parameter box (labelled *bulk*). Clearly, all  $\max(\epsilon_{\text{THF}})$  are comparable and considerably smaller than the maximal error reported in Heitmann et al. (2010b) and Lawrence et al. (2017).

### 3.4.4 Cardinality of the ED

We expect the maximal error to decrease as more cosmologies are included in the ED. This expectation has been tested over the range  $10 \leq n_{\text{ED}} \leq 250$  keeping all the other parameters fixed. The result is shown in Fig. 5. Notice that this plot suggests that as few as 50 cosmologies in the ED are enough to bring the maximal error within the  $0.83\Delta$ -ellipsoid down to below 1 per cent given

the configuration used ( $a = 1 - 10^{-5}$ ). However, unlike the halo model, whose data are smooth, a  $N$ -body simulation will produce data with some noise. Techniques, such as PCA, can reduce but not eliminate this noise contribution. Thus the overall maximal error of  $N$ -body simulation-based emulators can be larger than predicted by halo model-based emulators. For this reason, we decided to build the EuclidEmulator on a conservative ED containing 100 cosmologies, reducing the expected relative error again by roughly a factor of 2.

It is important to notice that the relation between the parameter  $a$  and the number of principal components taken into account is non-trivial and depends on  $n_{\text{ED}}$  as well as the nature of the data in the ED itself (e.g. are the data noisy or not). As a result, the number of principal components is not constant along the curves in Fig. 5: the larger  $n_{\text{ED}}$ , the more principal components are considered for a given parameter value for  $a$ .

### 3.4.5 Truncation and accuracy parameters

The truncation parameters are  $q \in [0, 1]$  and  $r \in \mathbb{N}$  (characterizing the multi-index set  $\mathcal{A}$ ) together with the accuracy parameter  $a \in [0, 1]$  (or alternatively the number of principal components). We have constructed an emulator for each set  $(a, q, r)$  in the grid defined in equation (12) using  $N = 100$  cosmologies in the ED and computed the maximal error as explained above. The goal now is to find the set  $(a, q, r)$  that includes the least number of terms in the PC expansion while keeping the EOE low.

As this emulator shall be capable of evaluating many nonlinear correction spectra within a second, we try to identify the most efficient emulator. We find that the accuracy parameter  $a$  is the most dominant of those three parameters and that changing  $q$  and  $r$  does not have a significant effect as long as  $r \geq 2$  and  $q \geq 0.5$ . We thus report the subsequent results always for  $n_{\text{ED}} = 100$ ,  $r = 2$ , and  $q = 0.5$  and only investigate the dependence on  $a$  and  $n_{\text{PCA}}$ , respectively.

We find that the smallest number of PCs that have to be taken into account is  $n_{\text{PCA}} = 4$  (corresponding to  $a = 1 - 10^{-3}$ ) as this leads to a maximal EOE of just about 1 per cent. It is also possible to identify the emulator minimizing the maximal error, which is achieved by setting  $n_{\text{PCA}} = 26$  ( $a = 1 - 10^{-6}$ ) with  $\max(\epsilon_{\text{THF}}) \approx 0.1$  per cent, which is an order of magnitude smaller than the simulation uncertainty. Notice, though, that increasing  $n_{\text{ED}}$  further does not automatically improve the result: taking all principal components into account leads to an enhanced final error, hinting at the fact that there is an optimal number of PCs that can be taken into account.

Including more principal components will decrease the emulator performance as more terms have to be computed and the amount of input data for the emulator increases considerably. It is thus desirable to find a configuration that keeps the maximal EOE well within the 1 per cent region but still leads to an efficient emulator.

We have chosen to configure EuclidEmulator with the parameters

$$\begin{aligned} n_{\text{ED}} &= 100, \\ n_{\text{PCA}} &= 11 \quad (a = 1 - 10^{-5}), \\ q &= 0.5, \\ r &= 2. \end{aligned} \tag{13}$$

The conclusion of our Takahashi HALOFIT modelling is that our final EuclidEmulator should achieve a maximal error of 0.27 per cent using only 100 different cosmologies, or 200 paired-and-fixed  $N$ -body simulations. We show in the next section

that this appears to be confirmed for the final simulation-based emulator. Thus we can quickly and reliably predict the performance and minimize the computational cost of any future emulator, thereby maximizing the return of the entire  $N$ -body simulation campaign.

## 4 EMULATOR PERFORMANCE, ERRORS, AND SENSITIVITY TO PARAMETERS

### 4.1 End-to-end tests of the EOE

The predictions we have obtained in the previous section now have to be tested in an end-to-end manner for the actual `EuclidEmulator` based on real simulations. The test is performed along the six coordinate axes (varying only one parameter at the time). For each of them, a  $N$ -body simulation of the cosmologies at  $\pm 1\sigma$ ,  $\pm 3\sigma$ , and  $\pm 5\sigma$  (for  $w_0$  we similarly chose  $\pm 0.58\sigma$ ,  $\pm 1.75\sigma$ , and  $\pm 2.92\sigma$ ) from the centre of the parameter range is run resulting in a test set of 36 reference simulations outside the ED used for the construction of the emulator. Then the emulator is executed at the very same cosmologies and compared to the simulations. In Table 2 we report the relative errors maximized over the entire  $k$ -range, the redshift range, and the  $\sigma$ -set (i.e. we report  $\max[\text{EOE}(n\sigma), \text{EOE}(-n\sigma)]$ ): the overall maximal EOE found is 0.145 per cent and thus much better than the error coming from the simulations and within the limit predicted by the `HALOFIT`-based error map. In Fig. 6 we explicitly compare simulated and emulated nonlinear correction curves for six different cosmologies along the  $h$ -axis for redshift  $z = 0$  (plotted are the corrections relative to the Euclid reference cosmology). As will be established in the following Section 4.2, the Hubble parameter  $h$  is one of the parameters the emulator is most sensitive to, even for higher order principal components, and thus its variation should have a non-negligible effect. This is actually true as the six different cases are clearly distinguishable in the figure: While varying  $h$  has almost no effect on linear scales, the curves corresponding to these six cases clearly deviate from one another on small scales.<sup>7</sup> Yet, the emulated nonlinear corrections coincide almost perfectly with the simulated ones that are based on paired-and-fixed initial conditions. Though the relative differences (lower panel) do show a systematic around BAO scale, these differences are negligible and the emulated data are effectively of simulation-quality. Note that the simulation of one of these nonlinear correction curves takes about 2000 node hours while the corresponding emulated curve is computed within less than 50 ms on a usual laptop. `EuclidEmulator` thus speeds up the data generation process by more than seven orders of magnitude compared to a classic  $N$ -body simulation with essentially no additional uncertainty due to emulation.

### 4.2 Sensitivity analysis

Sobol' indices (Sobol' 1993, 2001) measure how sensitive the eigenvalues  $\lambda_i(\omega_b, \omega_m, n_s, h, w_0, \sigma_8)$  (introduced in equation 9) are to each single input parameter as well as to any of their interactions. For an introduction into Sobol' sensitivity analysis, we refer to the previous references or to Marelli et al. (2017) and Le Gratiet, Marelli & Sudret (2016).

<sup>7</sup>This makes it obvious, why it is important for surveys like Euclid to investigate the small scales. There is tremendous leverage on cosmological parameters in this regime.

Sobol' indices are based on the Hoeffding-Sobol' decomposition, which states that any square-integrable function over a hypercube input parameter space can be cast as a sum of a constant, a set of univariate functions of each input parameter, another set of bivariate functions, etc. This decomposition is unique and the various terms are orthogonal with each other (with respect to the uniform probability measure over the hypercube). The variance of the output can then be apportioned to each input parameter, each pair, triplets, etc.: these contributions are called Sobol' sensitivity indices  $S$ . Although their classical estimation relies on costly Monte Carlo simulations, Sobol' indices can be computed analytically from a PCE as in equation (9), see Sudret (2008). In our case, we get one such Sobol' expansion for each principal component. Each Sobol' index corresponds to the fraction of the total variance of the respective eigenvalue that is caused by the parameter(s) under consideration. The bigger this number, the more  $\lambda_i$  depends on the considered set of input parameters. Fig. 7 shows the first-order (no interactions) Sobol' index plots for the first and fifth principal components (see Fig. A1 in Appendix A). It is interesting to notice that the input parameter  $\omega_b$  has no leading-order effect on the output nonlinear correction. The nonlinear correction starts to show non-vanishing sensitivity to this parameter only at fifth order and higher. This is explained by the fact that baryons are treated as a background quantity that only come into the simulation data via the transfer function. As they are not directly evolved themselves in `PKDGRAV3`, their nonlinear contribution is only due to their mass, which is taken into account in the  $\omega_m$  parameter.

### 4.3 Performance of EuclidEmulator

For the current implementation of `EuclidEmulator`, we have measured the execution times in three different set-ups (all times quoted were measured using one computing node):

(i) the emulation of the full nonlinear power spectrum using the python wrapper of `EuclidEmulator` (called `e2py`), which in turn calls the python wrapper of `CLASS` (called `classy`) to compute the linear power spectrum. In this set-up, a wall-time of 0.37 s was measured.

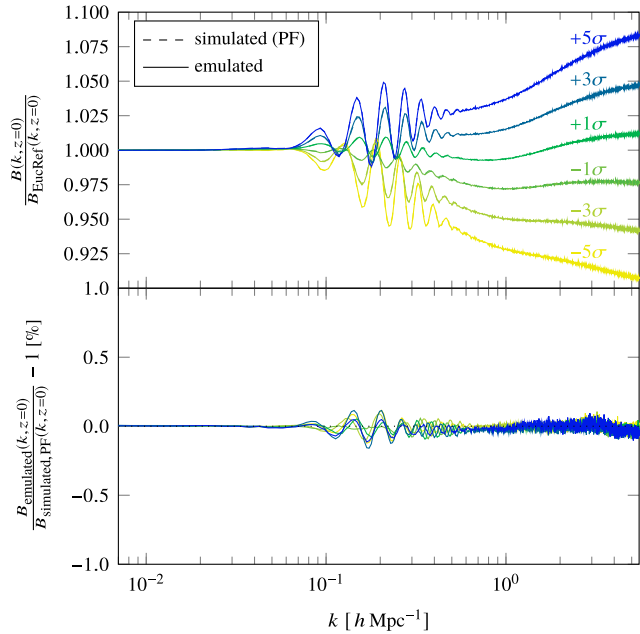
(ii) the emulation of the nonlinear correction only using the C code. For this task, we measured a wall-time of 6 ms. This time includes loading the information from the data table, calculation of cosmological quantities (e.g. the conversion from expansion factor to time), redshift interpolation, and printing the results.

(iii) in a 'Monte Carlo set-up' (not yet available in the currently public version), i.e. the set-up that would be used to actually perform a Markov Chain Monte Carlo (MCMC) search of the parameter space. In this scenario one would load the data table only once and pre-compute the needed parts (dependent on the redshift) of the output data space  $\mathbf{D}$  for the interpolation. This leaves calculating the PCE and assembling the principal components for each MCMC step. In this case we measure an evaluation time of less than 5  $\mu\text{s}$  (for  $n_{\text{PCA}} = 11$ ). Notice that only in this setting we were able to measure the difference in wall-time between emulators taking different numbers of principal components into account as in the two previous cases this difference was unmeasurable compared to the total runtime. If we reduce  $n_{\text{PCA}}$  to 8, we measured 2.92  $\mu\text{s}$  and 1.72  $\mu\text{s}$  for  $n_{\text{PCA}} = 2$ .

We stress that the current implementation of the code is not particularly optimized and any optimization at this point would be premature as clearly the biggest part of the calculation is spent in the computation of the linear power spectrum. This motivates the need

**Table 2.** In this table we compare nonlinear correction spectra computed with *EuclidEmulator* (EE) on the one hand and with an emulator based on Takahashi’s *HALOFIT* (THF) on the other hand to nonlinear correction spectra of full  $N$ -body simulations for 36 cosmologies outside the ED over the entire  $k$  and  $z$  range. The numbers in the table correspond to the relative errors in per cent between the emulated and simulated nonlinear corrections. We find that the errors predicted with the THF emulator are broadly consistent with the EE errors. This implies that the configuration of the EE based on a *HALOFIT* emulator is actually valid and that there is no need to make a huge investment to run  $N$ -body simulations just for the configuration study of an emulator. This result is particularly important for finding the size of the ED (see Fig. 5).

	$\omega_b$		$\omega_m$		$n_s$		$h$		$w_0$		$\sigma_8$	
	EE	THF	EE	THF	EE	THF	EE	THF	EE	THF	EE	THF
$\pm 1\sigma$	0.081	0.081	0.093	0.091	0.082	0.076	0.078	0.119	0.082	0.076	0.086	0.097
$\pm 3\sigma$	0.081	0.098	0.105	0.124	0.085	0.064	0.114	0.181	0.076	0.080	0.089	0.067
$\pm 5\sigma$	0.073	0.062	0.166	0.220	0.121	0.069	0.145	0.127	0.081	0.062	0.089	0.096

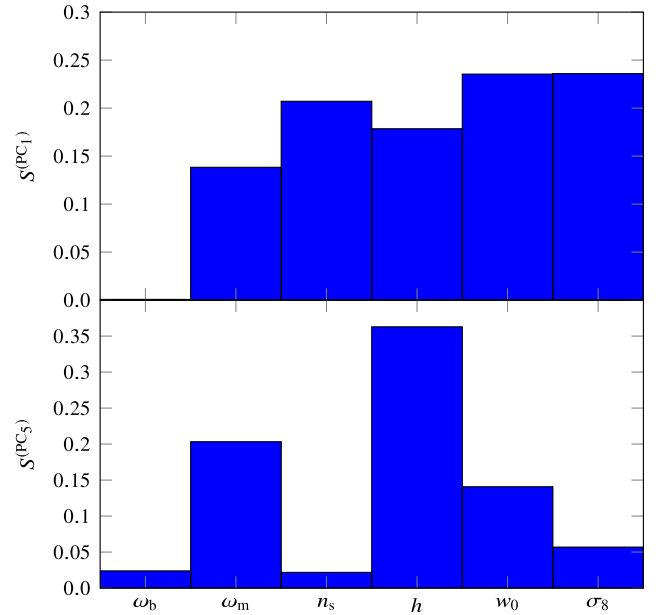


**Figure 6.** Comparison of the emulated (solid lines) and the paired-and-fixed (PF) initial condition-based simulated (dashed lines) nonlinear corrections for six different cosmologies at  $\pm 1\sigma$ ,  $\pm 3\sigma$ , and  $\pm 5\sigma$  away from the centre of the  $h$ -parameter axis (see column 7 of Table 2). In the upper panel, corrections of the nonlinear corrections themselves relative to Euclid reference (‘EuclidRef’) cosmology are shown. No difference between the emulated and the simulated curves is visible. In the lower panel, the relative differences between emulated and PF-simulated nonlinear corrections are plotted. The entire y-range of the lower subplot corresponds to the accuracy tolerance regime of  $\pm 1$  per cent, while the maximal error is roughly 6 times smaller than this upper limit.

for a comparably fast method to estimate the linear component, e.g. with a (separate) emulator. Clearly, this approach makes an MCMC search of the parameter space very efficient.

#### 4.4 Comparison to other fast prediction techniques

We compare *EuclidEmulator* against two well-known alternative surrogate modelling tools: Takahashi’s extension of *HALOFIT* and the *CosmicEmu* code based on the Mira-Titan Universe suite of simulations (Lawrence et al. 2017) produced with the  $N$ -body code HACC described in Habib et al. (2016). Moreover, we also compare *EuclidEmulator* against the very recent *NGenHalofit* (Smith & Angulo 2018). For these comparisons, we use the Euclid Reference cosmology (i.e. the comparisons are out-of-sample tests) and they are performed on the level of

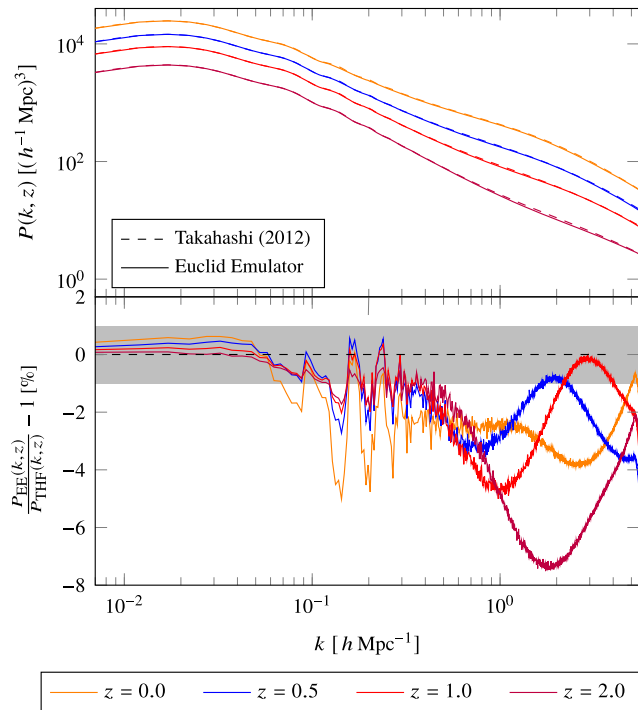


**Figure 7.** Sobol’ index analysis plots. The bar plots show the individual first-order Sobol’ indices for each cosmological parameter for the first (upper panel) and fifth principal components (lower panel). It is clearly visible that the baryon density parameter  $\omega_b$  has only a higher order effect on the nonlinear correction spectrum.

power spectra, i.e. the nonlinear correction curves computed by *EuclidEmulator* were multiplied with a linear power spectrum generated with the Boltzmann code *CLASS*. The result is then compared to the data from the other two predictors.

##### 4.4.1 Takahashi extension of *HALOFIT*

It becomes evident that *EuclidEmulator* is indeed able to correctly reproduce the linear regime of the power spectra (see Fig. 8). This is a big advantage of nonlinear correction emulation over direct emulation of power spectra (as is clear from the comparison to the *CosmicEmu*, see Section 4.4.2). Further, one can see the distinct systematic wiggles at BAO scales. They come from the fact that *HALOFIT* does not capture the nonlinear evolution of the BAOs very well (Heitmann et al. 2010b). On even smaller scales, there is a clear disagreement between the Takahashi model and *EuclidEmulator* at the level of several percentages. The differences, however, obey the uncertainty limits quoted in Takahashi et al. (2012).

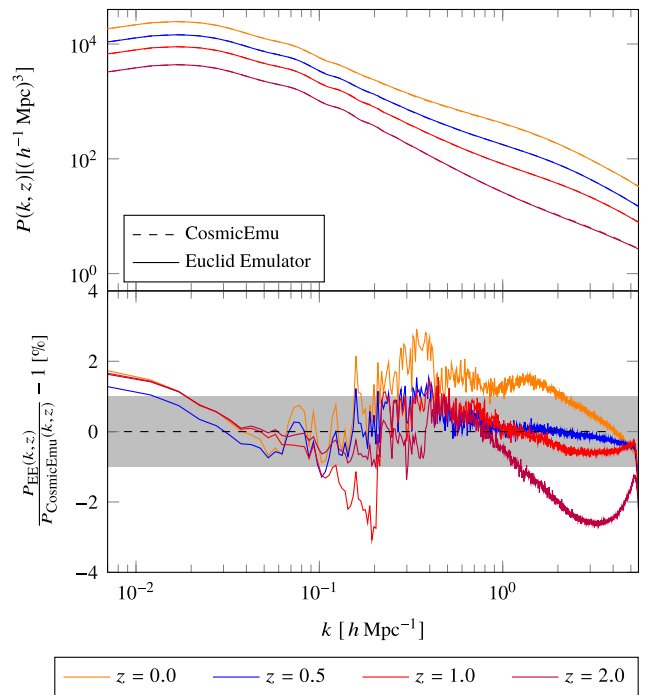


**Figure 8.** Comparison of nonlinear power spectra computed using `EuclidEmulator` (upper panel, solid curves) and the Takahashi model (upper panel, dashed curves). In the upper panel the absolute power spectra are shown while in the lower panel the relative error curves between the power spectra computed by the two different means are plotted. The agreement on large, linear scales is nearly perfect. On intermediate scales a distinct systematic coming from the BAO signal is observed for all tested redshifts. This shows that the Takahashi model does not capture the BAOs at the desired level of accuracy. The disagreement on the smallest scales is of the order of several percentages over all redshifts.

#### 4.4.2 `CosmicEmu` (*Mira-Titan emulator*)

One observes a disagreement up to  $\sim 2$  per cent between `CosmicEmu` and `EuclidEmulator` at linear scales (see Fig. 9). This is partially explained by the fact that `COSMICEMU` (see glossary) emulates the nonlinear power spectrum directly, which introduces an error on all scales. As `EuclidEmulator` nonlinear correction curves have to be multiplied with a linear power spectrum, the resulting nonlinear power spectrum matches linear theory on large scales by construction. As we use `CLASS` for the computation of the linear power spectrum, the resulting nonlinear curve does also contain the general theory of relativity (GR) corrections to the level given by `CLASS`. This is one of the biggest advantages of `EuclidEmulator`, but also comes at the expense of speed. Since direct emulation of the power spectrum circumvents the need for a Boltzmann solver, `CosmicEmu` is substantially faster: It takes `CosmicEmu` roughly 20 ms to compute the nonlinear power spectrum.

On intermediate and small scales the disagreement between `CosmicEmu` and `EuclidEmulator` is at most 3 per cent and thus consistent with the uncertainty bounds reported in Lawrence et al. (2017). Summarizing, one can say that on large scales ( $k < 0.06 h \text{ Mpc}^{-1}$ ) where the DM clustering nicely follows linear theory, `EuclidEmulator` can be used to produce power spectra consistent with `HALOFIT` and Takahashi’s extension well within the 1 per cent region. `CosmicEmu`, however, does deviate

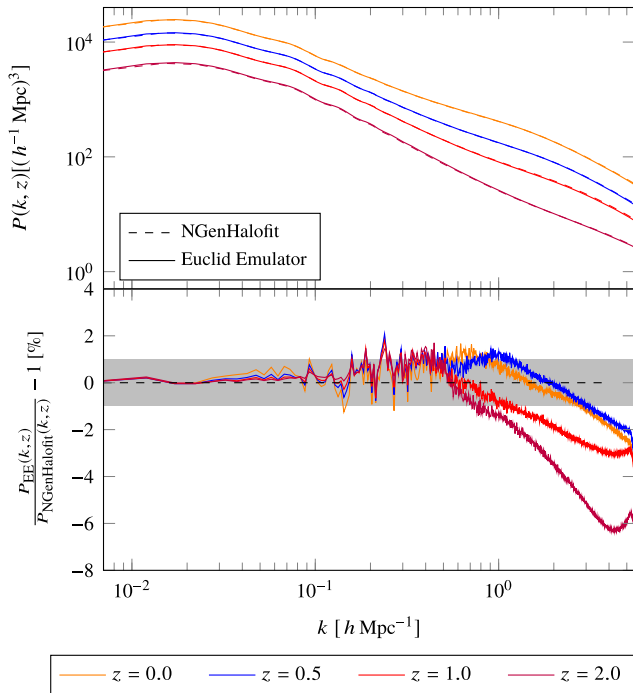


**Figure 9.** Comparison of nonlinear power spectra computed using `EuclidEmulator` (upper panel, solid curves) and the `CosmicEmu` code (upper panel, dashed curves). The relative errors between the power spectra computed with the two different approaches are again shown in the lower panel. On the largest scales, there is a systematic disagreement originating in the `CosmicEmu` data (remember that on these scales `EuclidEmulator` is correct by construction). While on intermediate scales one observes disagreements of the order of a few per cent as well, the agreement on small scales is very good over all redshifts.

from the Takahashi model by a few per cent on these scales; a consequence of the emulation strategy. On mildly nonlinear scales ( $0.06 < k < 0.5 h \text{ Mpc}^{-1}$ ), there is a certain disagreement between `EuclidEmulator` and both the Takahashi model and `CosmicEmu`, but of an entirely differing nature. While the deviation of Takahashi’s model is systematic and correlated with the BAO signal, the few per cent differences between `EuclidEmulator` and `CosmicEmu` show an overall offset with redshift over these intermediate scales. On small scales ( $k > 0.5 h \text{ Mpc}^{-1}$ ), `CosmicEmu` and `EuclidEmulator` are largely consistent in contrast to Takahashi’s `HALOFIT`, which systematically overestimates the nonlinear power by 4–8 per cent, depending on redshift. These observed discrepancies are broadly consistent with the ones shown in fig. 5 of Schneider et al. (2016). Only on the smallest scales ( $k > 3 h \text{ Mpc}^{-1}$ ) and redshifts  $z \geq 1$ , there is also a mismatch in the comparison, which can be explained by the different mass resolution considered in that figure.

#### 4.4.3 `NGenHalofit`

The agreement between `EuclidEmulator` and `NGenHalofit` is nearly perfect for large scales with  $k \leq 0.1 h \text{ Mpc}^{-1}$  for all tested redshifts, as can be seen in Fig. 10. On intermediate scales the agreement is slightly above the 1 per cent level, which is better than the corresponding results from the comparison to `CosmicEmu` or Takahashi’s `HALOFIT`. On small scales, however, we observe a mismatch of up to  $\sim 6$  per cent (at  $z = 2$ ), which



**Figure 10.** Comparison of nonlinear power spectra computed using *EuclidEmulator* (upper panel, solid curves) and the *NGenHalofit* code (upper panel, dashed curves). The agreement on linear and intermediate scales is very good with relative errors above 1 per cent at only a few  $k$ -modes around the BAO scale. However, there are substantial deviations on small scales that grow as one goes back in redshift.

is outside the bounds reported in Smith & Angulo (2018). This disagreement may be explained by the fact that the *Dämmerung* simulation suite used to build *NGenHalofit* uses 2LPT initial conditions generated at redshift  $z = 49$  while the simulations used to construct *EuclidEmulator* are based on ZA initial conditions (see the discussion in Section 2.2.2). We performed a comparison between ZA-based simulations ( $z_{\text{ZA}} = 200$ ) including radiation (the *EuclidEmulator* simulations) and 2LPT simulations ( $z_{\text{2LPT}} = 49$ ) without radiation (like the *Dämmerung* simulations). At  $z \sim 2$  and  $k \sim 5 h \text{ Mpc}^{-1}$ , we find an underestimation of power in the ZA case compared to the 2LPT data at the level of roughly 3 per cent. While the two approaches agree perfectly on linear scales, the disagreement only becomes significant towards higher redshifts and higher  $k$ -modes (the agreement between ZA and 2LPT is better than 1 per cent for all  $z \lesssim 1$ ). The exact  $k$ -mode at which the maximal mismatch is located is resolution dependent. This finding explains the excess mismatch we find in Fig. 10. This topic has also been discussed in Garrison et al. (2016), where the authors find that the 2LPT approach is the more accurate one. The agreement is at the 3 per cent level (or better) up to  $z \sim 1$  and out to  $k \sim 5 h \text{ Mpc}^{-1}$ .

## 5 CONCLUSION

Efficient and at the same time accurate estimation of nonlinear matter power spectra is crucial in order to exploit the full potential of cosmological surveys such as Euclid, DES, LSST, and WFIRST. The Boltzmann solvers *CAMB* and *CLASS* are well established as numerical tools to compute the linear matter power spectra. In this paper we hence focused on the nonlinear correction in order to combine the strengths of the Boltzmann solvers and  $N$ -body

simulation codes: the former include much more physics (such as GR and baryonic physics) than any contemporary  $N$ -body code efficient enough to produce simulations of the size and resolution needed for current and upcoming surveys. The latter, however, are the only means by which structure growth on highly nonlinear scales can be studied. *EuclidEmulator*, presented in this paper, is a numerical tool that estimates the nonlinear correction spectra of an input cosmology (respecting the predefined parameter boundaries of  $0.83\Delta$  around the Planck2015 best-fitting cosmology) at any redshift  $z \leq 5$  with an overall accuracy far better than 1 per cent based on only 100 pre-evaluated DM-only simulations performed with the  $N$ -body code *PKDGRAV3*. The EOE is of the order of a fraction of a per cent and is hence dominated by the expected simulation errors (of the order of  $\sim 1$  per cent up to  $k \sim 1 h \text{ Mpc}^{-1}$ ).

The accuracy of the emulation could be achieved by using well-tested statistical techniques from the field of uncertainty quantification: like Heitmann et al. (2010b), we have used a special sampling technique called LHS in order to guarantee that the resulting ED of input cosmologies covers the cosmological parameter space in a statistically uniform way. We then simulate the corresponding nonlinear responses with *PKDGRAV3*, using paired-and-fixed (Angulo & Pontzen 2016) initial conditions that drastically reduce computational cosmic variance in the simulations. In contrast to the Coyote Universe, the Mira-Titan Universe and the Aemulus project emulators by Heitmann et al. (2010b), Lawrence et al. (2017), and Zhai et al. (2018), respectively, we employ a regression strategy called sparse PCE in order to surrogate model the nonlinear correction spectra. Our nonlinear correction approach leads to very accurate emulation of the nonlinear matter power spectrum, but additionally requires the linear power spectrum, calculated from *CLASS* or *CAMB*.

The emulator itself depends on a set of numerical parameters that need to be configured properly. To perform this configuration, we predict the emulator performance for a given set of emulation parameters (such as the size of the ED, the number of principal components taken into account, the truncation of the polynomial chaos series) using *HALOFIT* input data. We LH sample EDs of different sizes and compute mock emulators based on *HALOFIT/CLASS* nonlinear correction spectra. Doing so we identify the optimal emulator configuration leading to a maximal emulation-only error of 0.27 per cent within the  $0.83\Delta$  region of the parameter space. *EuclidEmulator*, constructed from 200  $N$ -body simulations with this optimal configuration, almost perfectly reproduces the results of  $N$ -body simulated power spectra at the 0.3 per cent level within 50 ms. Due to possible numerical systematics in the  $N$ -body simulations themselves, the absolute accuracy of nonlinear power spectra generated with *EuclidEmulator* is bounded by  $\pm 1$  per cent up to  $k \sim 6 h \text{ Mpc}^{-1}$  at  $z = 0$  while at  $z \sim 1$  this only holds up to  $k \sim 1 h \text{ Mpc}^{-1}$  (Schneider et al. 2016). In order to reduce uncertainties due to DM physics that potentially contaminate studies of baryonic effects that are dominant at these scales, it is hence vital in the future to further improve our confidence in the  $N$ -body simulations in the interval  $1 h \text{ Mpc}^{-1} \leq k \leq 10 h \text{ Mpc}^{-1}$  for redshifts up to  $z \sim 3$  or higher. Once this can be achieved, thanks to the emulator strategy presented in this paper, the same accuracy will be reflected by the emulated power spectra. For Euclid these scales are important to assess the constraining power of the mission.

Our modelling approach will allow us to optimize the configuration of future emulators for further observables such as the bispectrum and the halo mass function, projecting their end-to-end accuracy, prior to running any simulations. We will also optimize

emulation over a widened parameter space, adding neutrino mass, dark energy EoS evolution, and primordial non-Gaussianity. These shall be included self-consistently within a future set of  $N$ -body simulations, with a mass resolution comparable to the Euclid Flagship simulation ( $2.5 \times 10^9 h^{-1} M_{\odot}$ ).

We have shown in Section 4.3 that the run time of an emulation of a fully nonlinear power spectrum with `EuclidEmulator` is highly dominated by the evaluation of the linear part with the Boltzmann solver. This motivates the need for a future, separate emulator of the linear power spectrum in order to speed up the entire process. Two separate emulators for the linear power spectrum and its nonlinear correction are expected to perform better than one emulator for the nonlinear power spectrum, as the separation approach allows a denser sampling of the parameter space in the construction of the ED leading to more accurate results.

`EuclidEmulator` can be downloaded from GitHub (<https://github.com/miknab/EuclidEmulator>). The repository contains the main C source code together with a python wrapper, CMAKE files, scripts, and parameter files. Executing these scripts, the user can create a fully nonlinear power spectrum using `CLASS`<sup>8</sup> and `EuclidEmulator`.

## GLOSSARY

### Codes:

CAMB	Code for anisotropies in the microwave background 3, 5, 6, 14, 15
CLASS	Cosmological linear anisotropy solving system 3, 5–7, 9, 12–16
CosmicEmu	Cosmic emulator based on the Mira-Titan cosmological simulation suite (successor of FrankenEmu based on the Coyote simulation suite). 1, 2, 7, 10, 13, 14, 18
EuclidEmulator	Emulator code to emulate nonlinear corrections to DM power spectra 1–4, 6–15, 17–23
HACC	Hardware/Hybrid Accelerated Cosmology Code 13
HALOFIT	Analytical code to produce nonlinear power spectra 1, 9, 11–15
NGenHalofit	Code to produce nonlinear power spectra using a semi-analytical approach for large and a smoothing-spline-t model for small scales 1, 13, 14
PKDGRAV3	Parallel k-D tree gravity code (version 3); Cosmological $N$ -body tree code 1, 3–5, 7, 12, 14, 15, 20, 23
UQLab	Matlab-based uncertainty quantification framework 1, 3, 6–8
e2py	classy Python wrapper for CLASS 12 Python wrapper for EuclidEmulator 12

### Acronyms:

BAO	Baryon acoustic oscillations 3, 4, 11, 13
CMB	Cosmic microwave background 2, 7

DE	Dark energy 3, 7
DM	Dark matter 13
ED	Experimental design 3, 4, 6, 7, 9, 10, 18, 21, 23
EE	EuclidEmulator 12
EFHR	Euclid Flagship High Resolution 20, 21
EOE	Emulation-only error 6, 8–11, 17
EoS	Equation of state 3, 7
GR	General theory of relativity 13, 14
HOD	Halo occupation distribution 2
LARS	Least-angle regression-based selection 8, 18
LH	Latin hypercube 6, 15
LHS	Latin hypercube sampling 6
LV	Large volume 20–22
PC	Principal component 8, 9, 17
PCA	Principal component analysis 6, 7, 9
PCE	Polynomial chaos expansion 6, 8, 12, 18, 20
PCS	Piece-wise cubic spline 3, 5
SPCE	Sparse polynomial chaos expansion 3, 6
ZA	Zel'dovich approximation 5, 14

## ACKNOWLEDGEMENTS

We express our gratitude to Julien Lesgourgues, whose feedback and assistance in the `CLASS` studies were highly appreciated. MK acknowledges support from the Swiss National Science Foundation (SNF) grant 200020\_149848. LL acknowledges support from the ERC starting grant ERCStg-717001. Simulations were performed on the zBox4 + cluster at the University of Zurich. The Euclid Consortium acknowledges the European Space Agency and the support of a number of agencies and institutes that have supported the development of Euclid. A detailed complete list is available on the Euclid web site (<http://www.euclid-ec.org>). In particular the Agenzia Spaziale Italiana, the Centre National d'Etudes Spatiales, the Deutsches Zentrum für Luftund Raumfahrt, the Danish Space Research Institute, the Fundação para a Ciência e a Tecnologia, the Ministerio de Economía y Competitividad, the National Aeronautics and Space Administration, the Nederlandse Onderzoekschool Voor Astronomie, the Norwegian Space Center, the Romanian Space Agency, the State Secretariat for Education, Research and Innovation (SERI) at the Swiss Space Office (SSO), the United Kingdom Space Agency, and the University of Helsinki.

## REFERENCES

- Amendola L. et al., 2018, *Living Rev. Relativ.*, 21, 2  
 Angulo R. E., Pontzen A., 2016, *MNRAS*, 462, L1  
 Bardeen J. M., Bond J. R., Kaiser N., Szalay A. S., 1986, *ApJ*, 304, 15  
 Bennett C. L. et al., 2003, *ApJ*, 583, 1  
 Blas D., Lesgourgues J., Tram T., 2011, *J. Cosmol. Astropart. Phys.*, 07, 34  
 Blatman G., 2009, PhD thesis, Université de Clermont-Ferrand  
 Blatman G., Sudret B., 2009, in 19ème Congrès Français de Mécanique. Association Française de Mécanique, Maison de la Mécanique, Courbevoie, France, p. 01, available at: <http://hdl.handle.net/2042/37252>  
 Blatman G., Sudret B., 2010, *Probab. Eng. Mech.*, 25, 183  
 Blatman G., Sudret B., 2011, *J. Comput. Phys.*, 230, 2345  
 Carlson J., White M., Padmanabhan N., 2009, *Phys. Rev. D*, 80, 043531  
 Carrasco J. M., Foreman S., Green D., Senatore L., 2014, *J. Cosmol. Astropart. Phys.*, 1  
 Chisari N. E. et al., 2018, *MNRAS*, 480, 3962  
 Crocce M., Scoccimarro R., 2008, *Phys. Rev. D*, 77, 23533

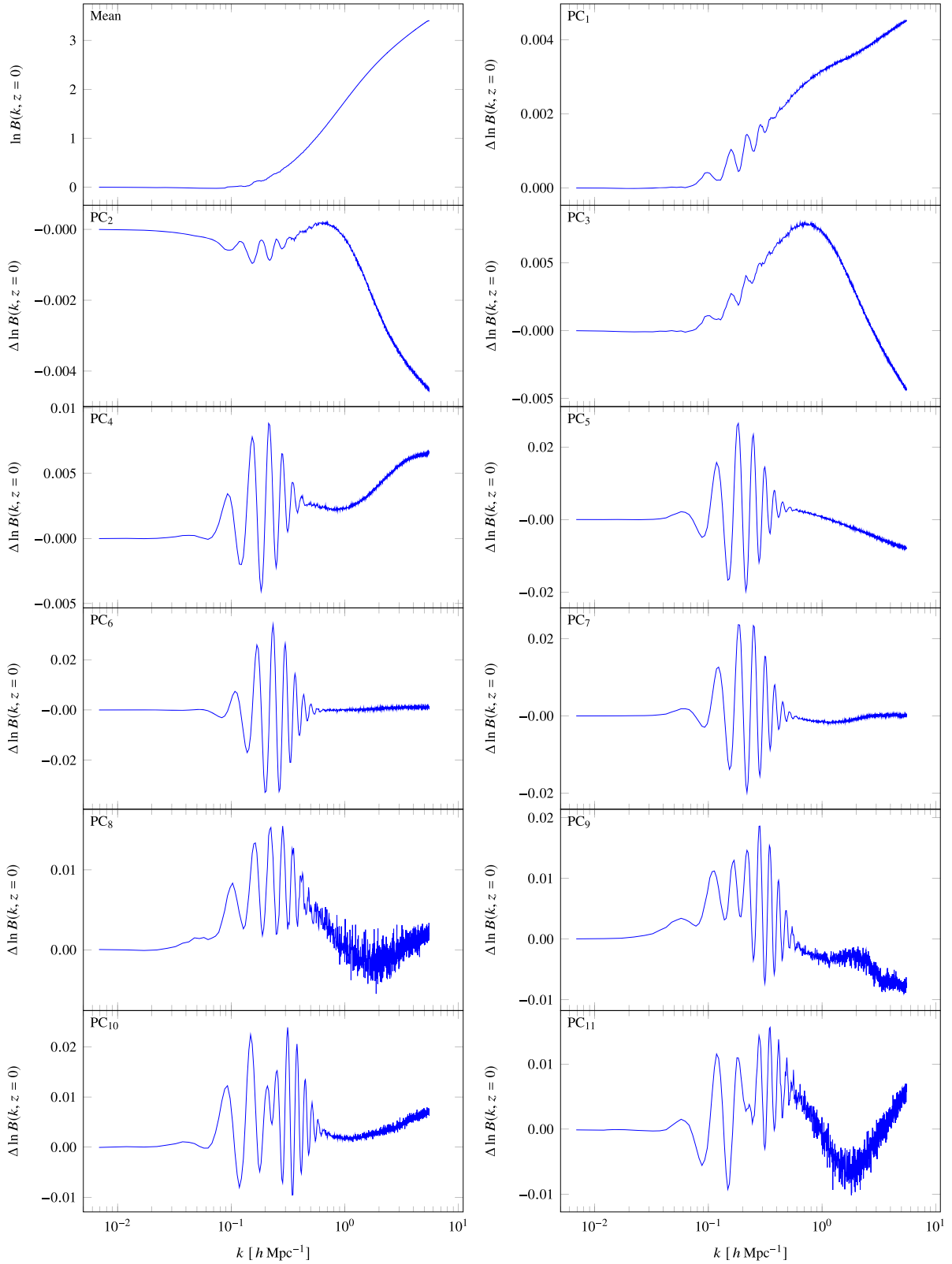
<sup>8</sup>The `CLASS` code has to be installed separately. It can be downloaded from <http://class-code.net>.

- Crocce M., Scoccimarro R., Bernardeau F., 2012, *MNRAS*, 427, 2537
- Davis M., Peebles P. J. E., 1977, *ApJ*, 34, 425
- Davis M., Peebles P. J. E., 1983, *ApJ*, 267, 465
- DeRose J. et al., 2018, preprint ([arXiv:1804.05865](https://arxiv.org/abs/1804.05865))
- Donzelli S., Dor O., Gregorio A., 2016, *A&A*, 594, 1
- Efron B., Hastie T., Johnstone I., Tibshirani R., 2004, *Ann. Stat.*, 32, 407
- Eisenstein D. J. et al., 2005, *ApJ*, 633, 560
- Fang X., Blazek J., McEwen J., Hirata C., 2017, *J. Cosmol. Astropart. Phys.*, 2, 030
- Garrison L. H., Eisenstein D. J., Ferrer D., Metchnik M. V., Pinto P. A., 2016, *MNRAS*, 461, 4125
- Ghanem R., Spanos P., 2003, *Stochastic Finite Elements: A Spectral Approach*, 2nd edn. Courier Dover Publications, Mineola, New York
- Green J. et al., 2012, preprint ([arXiv:1208.4012](https://arxiv.org/abs/1208.4012))
- Habib S. et al., 2016, *New Astron.*, 42, 49
- Heitmann K. et al., 2016, *ApJ*, 820, 108
- Heitmann K., Higdon D., White M., Habib S., Williams B. J., Lawrence E., Wagner C., 2010a, *ApJ*, 705, 156
- Heitmann K., White M., Wagner C., Habib S., Higdon D., 2010b, *ApJ*, 715, 104
- Heitmann K., Lawrence E., Kwan J., Habib S., Higdon D., 2014, *ApJ*, 780, 111
- Hellwing W. A., Schaller M., Frenk C. S., Theuns T., Schaye J., Bower R. G., Crain R. A., 2016, *MNRAS*, 461, L11
- Huterer D., Takada M., 2005, *Astropart. Phys.*, 23, 369
- Jain B., Bertschinger E., 1994, *ApJ*, 431, 495
- Jing Y. P., 2005, *ApJ*, 620, 559
- Johnson M. E., Moore L. M., Ylvisaker D., 1990, *J. Stat. Plan. Inference*, 26, 131
- Kaiser N., 1984, *ApJ*, 284, L9
- Klypin A., Prada F., 2018, *MNRAS*, 478, 4602
- Kullback S., Leibler R. A., 1951, *Ann. Math. Stat.*, 22, 79
- Laureijs R. et al., 2011, preprint ([arXiv:1110.3193](https://arxiv.org/abs/1110.3193))
- Lawrence E. et al., 2017, *ApJ*, 847, 50
- Lawrence E., Heitmann K., White M., Higdon D., Wagner C., Habib S., Williams B., 2010a, *CosmicEmu: Cosmic Emulator for the Dark Matter Power Spectrum*, Astrophysics Source Code Library, record ascl:1010.030, available at: <http://adsabs.harvard.edu/abs/2010ascl.soft10030L>
- Lawrence E., Heitmann K., White M., Higdon D., Wagner C., Habib S., Williams B., 2010b, *ApJ*, 713, 1322
- Lewis A., Challinor A., Lasenby A., 2000, *ApJ*, 538, 473
- Le Gratiet L., Marelli S., Sudret B., 2016, in Ghanem R., Higdon D., Owhadi H., eds, *Handbook on Uncertainty Quantification*. Springer International Publishing, Cham, Switzerland, p. 1289
- LSST Science Collaboration, 2009, preprint ([arXiv:0912.0201](https://arxiv.org/abs/0912.0201))
- Marelli S., Sudret B., 2014, in *Vulnerability, Uncertainty, and Risk* (Proc. 2nd Int. Conf. on Vulnerability, Risk Analysis and Management (ICVRAM2014), Liverpool, American Society of Civil Engineers, United Kingdom, p. 2554
- Marelli S., Sudret B., 2017, Technical Report, UQLab user manual - Polynomial Chaos Expansion. Chair of Risk, Safety & Uncertainty Quantification. ETH Zurich, Zurich, Switzerland
- Marelli S., Lamas C., Sudret B., Konakli K., 2017, Technical Report, UQLab user manual - Sensitivity analysis. Chair of Risk, Safety & Uncertainty Quantification. ETH Zurich, Zurich, Switzerland
- McCarthy I. G., Schaye J., Bird S., Le Brun A. M. C., 2017, *MNRAS*, 465, 2936
- McClintock T. et al., 2018, preprint ([arXiv:1804.05866](https://arxiv.org/abs/1804.05866))
- McEwen J., Fang X., Hirata C., Blazek J., 2016, *J. Cosmol. Astropart. Phys.*, 09, 015
- McKay M. D., Beckman R. J., Conover W. J., 1979, *Technometrics*, 21, 239
- Mummery B. O., McCarthy I. G., Bird S., Schaye J., 2017, *MNRAS*, 471, 227
- Nyquist H., 1928, *Proc. IEEE*, 47, 617
- Peebles P. J. E., 1990, *ApJ*, 365, 27
- Planck Collaboration XIII, 2015, *A&A*, 594, A13
- Pontzen A., Slosar A., Roth N., Peiris H. V., 2016, *Phys. Rev. D*, 93, 103519
- Potter D., Stadel J. G., 2016, PKDGRAV3: Parallel Gravity Code, Astrophysics Source Code Library, record ascl:1609.016, available at: <http://adsabs.harvard.edu/abs/2016ascl.soft09016P>
- Potter D., Stadel J., Teyssier R., 2017, *Comput. Astrophys. Cosmol.*, 4, 2
- Santner T. J., Williams B. J., Notz W. I., 2013, *The Design and Analysis of Computer Experiments*, Springer Science & Business Media, New York
- Schneider A. et al., 2016, *J. Cosmol. Astropart. Phys.*, 04, 047
- Schneider A., Teyssier R., 2015, *J. Cosmol. Astropart. Phys.*, 2015, 049
- Sefusatti E., Crocce M., Scoccimarro R., Couchman H. M. P., 2016, *MNRAS*, 460, 3624
- Semboni E., Hoekstra H., Schaye J., van Daalen M. P., McCarthy I. G., 2011, *MNRAS*, 417, 2020
- Smith R. E., Angulo R. E., 2018, preprint ([arXiv:1807.00040](https://arxiv.org/abs/1807.00040))
- Smith R. E. et al., 2003, *MNRAS*, 341, 1311
- Sobol' I. M., 1993, *Math. Modeling Comp. Exp.*, 1, 407
- Sobol' I. M., 2001, *Math. Comput. Simul.*, 55, 271
- Springel V. et al., 2018, *MNRAS*, 475, 676
- Springel V., 2005, *MNRAS*, 364, 1105
- Stadel J. G., 2001, PhD thesis, Univ. Washington
- Sudret B., 2008, *Reliab. Eng. Syst. Saf.*, 93, 964
- Takahashi R., Sato M., Nishimichi T., Taruya A., Oguri M., 2012, *ApJ*, 761, 152
- Tang B., 1993, *J. Am. Stat. Assoc.*, 88, 1392
- Tauber J. A. et al., 2010, *A&A*, 520, 1
- Teyssier R., 2010, RAMSES: A New N-body and Hydrodynamical Code, Astrophysics Source Code Library, record ascl:1011.007, available at: <http://adsabs.harvard.edu/abs/2010ascl.soft11007T>
- The Dark Energy Survey Collaboration, 2005, preprint ([astro-ph/0510346](https://arxiv.org/abs/astro-ph/0510346))
- van Daalen M. P., Schaye J., Booth C. M., Dalla Vecchia C., 2011, *MNRAS*, 415, 3649
- Vapnik V., 1998, *Statistical Learning Theory*. New York
- Villaescusa-Navarro F. et al., 2018, *ApJ*, 867, 24
- Vogelsberger M. et al., 2014, *Nature*, 509, 177
- Wiener N., 1938, *Am. J. Math.*, 60, 897
- Xiu D., 2010, *Numerical Methods For Stochastic Computations – A Spectral Method Approach*. Princeton Univ. Press, Princeton, NJ
- Xiu D., Karniadakis G. E., 2006, *SIAM J. Sci. Comput.*, 24, 619
- Zentner A. R., Semboni E., Dodelson S., Eifler T., Krause E., Hearin A. P., 2013, *Phys. Rev. D*, 87, 043509
- Zhai Z. et al., 2018, preprint ([arXiv:1804.05867](https://arxiv.org/abs/1804.05867))

## APPENDIX A: PRINCIPAL COMPONENTS OF THE EXPERIMENTAL DESIGN

In contrast to Heitmann et al. (2010b), we find that 11 principal components should be taken into account in order to bring the EOE to a sub-per cent level (see the discussion in Section 3.4.5). In Fig. A1 we plot the mean and the first 11 principal components (at  $z = 0$ ) of the nonlinear correction spectra used for the construction of the EuclidEmulator. Notice that the emulation is performed using the logarithm of the nonlinear correction. This is why we report the mean and the corrections to the mean of  $\ln(B)$ . Recall further that the data presented are normalized.

For each principal component, a Sobol' index analysis (see Section 4.2) can be performed. The results for the first and the fifth PC are shown in Fig. 7, which tells us that PC<sub>1</sub> is sensitive to all cosmological parameters but  $\omega_b$ , while PC<sub>5</sub> mostly depends on  $\omega_m$  and  $h$ .



**Figure A1.** The mean and the first five principal components of the ED response data (logarithm of the nonlinear correction spectra) used to construct the *EuclidEmulator*. Each of these curves is multiplied with its eigenvalue  $\lambda$ , which is the actual output of the SPCE emulator. The sum of these then produces the final nonlinear correction spectrum. Principal components PC<sub>6</sub> to PC<sub>11</sub> of the ED response data (logarithm of the nonlinear correction spectra) used to construct the *EuclidEmulator*. PC<sub>11</sub> is the highest order principal component used in the *EuclidEmulator*.

**APPENDIX B: SURROGATE MODELLING VIA SPCE**

**B1 Introduction to surrogate modelling**

The ability to make predictions and to properly propagate the input uncertainties to the output response vector in cases of complex systems is of prime interest in numerous situations. Yet, it is infeasible to perform expensive large-scale experiments or simulations for many input parameter sets to study the system’s behaviour in detail. In such cases a surrogate model (or ‘emulator’) can be computed that establishes a (model) relation between input and output. This means that the surrogate is not the ‘true’ relation but, depending on the computational resources available, it is able to capture the main features of this ‘true’ relation up to a required accuracy. The uncertainties can be kept under control using well-known techniques from statistical uncertainty quantification.

Mathematically we can formulate the problem of emulating a black-box model as follows: consider a set of input parameters as a random vector  $\mathbf{X} \in \mathbb{R}^d$

$$\mathbf{X} := \{X_i | i \in 1, \dots, d; d \in \mathbb{N}\},$$

where  $d$  parametrizes the dimensionality of the parameter space under consideration. The probability distributions of each of the independent components of  $\mathbf{X}$  are given by:  $X_i \sim F_{X_i}(x_i)$ . These parameters are mapped by a black-box relation  $\mathcal{M}$  (the computational model) to a quantity of interest  $Y$ :

$$Y := \mathcal{M}(\mathbf{X})$$

with  $Y \in \mathbb{R}$ . Due to the uncertainty in the input vector  $\mathbf{X}$ ,  $Y$  is also a random variable.

The goal is to find a surrogate model  $\mathcal{S}$  relating  $\mathbf{X}$  to  $Y$  based on a small set of model evaluations known as the ED  $\mathcal{X} = \{\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(n_{ED})}\}$  and the corresponding model responses  $\mathcal{Y} = \{y^{(1)} = \mathcal{M}(\mathbf{x}^{(1)}), \dots, y^{(n_{ED})} = \mathcal{M}(\mathbf{x}^{(n_{ED})})\}$ .

In realistic scenarios, the computational budget to create an ED is limited and this limitation puts a constraint on the amount of information one can use to construct the surrogate model. Further we add the requirement that the surrogate model  $\mathcal{S}$  must be as accurate as possible throughout the entire parameter space spanned by  $\mathbf{X}$ .

In this section we will focus on a specific type of surrogate model, polynomial chaos expansions (Wiener 1938), in contrast to Heitmann et al. (2010b) and Lawrence et al. (2017) where they use Kriging for both the FRANKENEMU and the COSMICEMU. We have chosen the PCE approach to construct EUCLIDEMULATOR (see glossary) because globally the errors are expected to be smaller than in the case of Kriging. Further, a PCE approach allows us to relax assumptions about the noise properties of the input model: Kriging can indeed deal with noise, but a very severe assumption on the noise distribution is to be made; the noise is assumed to be Gaussian. In a linear regression setting such as PCE, the only assumption is that the noise is unbiased (see e.g. Vapnik 1998).

**B2 Theory of PCEs**

The concept of PCE based on LARS shall quickly be reviewed here. For a deeper discussion of this topic, we refer to Blatman (2009), Blatman & Sudret (2009, 2010, 2011), and references therein.

Let us take  $\mathbf{X} = \{X_1, \dots, X_d\}^\top \in \mathbb{R}^d$  to be a random input vector with a joint probability density function (pdf)  $f_{\mathbf{X}}(\mathbf{x})$  and a finite variance model  $\mathcal{M}$  mapping  $\mathbf{X}$  to the response  $Y$  via  $Y := \mathcal{M}(\mathbf{X})$ ,

i.e.

$$\mathbb{E}[Y^2] = \int_{\mathcal{D}_{\mathbf{X}}} \mathcal{M}^2(\mathbf{x}) f_{\mathbf{X}}(\mathbf{x}) d\mathbf{x} < \infty, \tag{B1}$$

where  $\mathcal{D}_{\mathbf{X}}$  is the domain of the random input vector. Then  $\mathcal{M}(\mathbf{X})$  is an element of the stochastic Hilbert space  $\mathcal{H}$  of finite variance functions endowed with the inner product

$$\langle g, h \rangle := \mathbb{E} [g(\mathbf{X}) \cdot h(\mathbf{X})] = \int_{\mathcal{D}_{\mathbf{X}}} g(\mathbf{X})h(\mathbf{X}) f_{\mathbf{X}}(\mathbf{x}) d\mathbf{x}. \tag{B2}$$

Then the following spectral representation, known as PCE, holds:

$$Y = \mathcal{M}(\mathbf{X}) = \sum_{\alpha \in \mathbb{N}^d} \eta_{\alpha} \Psi_{\alpha}(\mathbf{X}), \tag{B3}$$

where  $\alpha = \{\alpha_1, \dots, \alpha_d\}$  is a multi-index,  $\Psi_{\alpha}$  is an element of a multivariate orthonormal polynomial basis of  $\mathcal{H}$ , and  $\eta_{\alpha}$  is the corresponding coefficient (coordinate). The multivariate polynomials  $\Psi_{\alpha}$  are constructed by tensor products of univariate orthonormal polynomials w.r.t. the input random variables:

$$\Psi_{\alpha}(\mathbf{X}) = \prod_{i=1}^d \phi_{\alpha_i}^{(i)}(x_i), \tag{B4}$$

where  $\phi_{\alpha_i}^{(i)}(x_i)$  is a polynomial of degree  $\alpha_i$  in  $x_i$  and orthonormal w.r.t. the pdf of  $f_{X_i}(x_i)$ . In other words:

$$\langle \phi_{\alpha}, \phi_{\beta} \rangle = \delta_{\alpha\beta}. \tag{B5}$$

From equation (B4), it follows that the total degree of the basis element  $\Psi_{\alpha}(\mathbf{X})$  is  $p = \|\alpha\|_1 = \sum_{i=1}^d \alpha_i$ , while from equation (B5) it follows that for an input random vector with independent components  $\mathbf{X}$ :

$$\langle \Psi_{\alpha}, \Psi_{\beta} \rangle = \delta_{\alpha\beta}. \tag{B6}$$

In the present case, EuclidEmulator is built to be consistently accurate across predefined parameter intervals. Therefore, their input distributions are considered uniform between the given bounds. Prior to the expansion, each parameter is linearly rescaled to the interval  $[-1, 1]$ , so that the polynomials  $\phi_{\alpha_i}^{(i)}$  used in the expansions belong to the Legendre family (Ghanem & Spanos 2003; Xiu & Karniadakis 2006).

*B2.1 Truncation of the polynomial basis*

For practical computational purposes, the expansion in equation (B3) needs to be truncated to a finite number of terms:

$$Y = \mathcal{M}(\mathbf{X}) \approx \sum_{\alpha \in \mathcal{A}} \eta_{\alpha} \Psi_{\alpha}(\mathbf{X}), \tag{B7}$$

where  $\mathcal{A}$  is a truncation with cardinality  $P := \text{card}(\mathcal{A}) < \infty$ .

Several strategies are available to define a suitable truncation set  $\mathcal{A}$  in the literature. The scheme applied in the construction of the EuclidEmulator is a combination of the so-called *maximum interaction* and *hyperbolic truncation* introduced in Blatman & Sudret (2011) and Marelli & Sudret (2017).

The *standard truncation scheme* is given by retaining only basis functions up to a specific total degree  $p$  such that

$$\mathcal{A}^{d,p} = \{\alpha \in \mathbb{N}^d, \|\alpha\|_1 \leq p\}. \tag{B8}$$

The cardinality of such a set is:

$$\text{card}(\mathcal{A}^{d,p}) = \binom{d+p}{p}, \tag{B9}$$

which is a polynomially increasing quantity. A significant reduction of the number of basis elements is to impose bounds on the maximum number of non-zero elements in  $\alpha_i$  to a desired  $r \leq d$ :

$$\mathcal{A}^{d,p,r} = \{\alpha \in \mathcal{A}^{d,p}, \|\alpha\|_0 < r\}, \quad (\text{B10})$$

where

$$\|\alpha\|_0 := \sum_{i=1}^d \mathbb{1}_{\alpha_i > 0} \quad (\text{B11})$$

is the rank of the multi-index. The effect of this is that in each multivariate polynomial chaos basis function only  $r$  or less univariate factors are not constant and hence  $r$  or less input parameters interact with each other (a ‘maximum interaction’ is defined).

We reduce the number of terms taken into account once more by applying hyperbolic truncation. This is closely related to the standard truncation scheme with the difference that instead of the 1-norm a more general  $q$ -norm is used with  $q \in (0, 1]$ :

$$\mathcal{A}^{d,p,q} = \{\alpha \in \mathcal{A}^{d,p}, \|\alpha\|_q \leq p\}, \quad (\text{B12})$$

where

$$\|\alpha\|_q := \left( \sum_{i=1}^d \alpha_i^q \right)^{1/q}. \quad (\text{B13})$$

Hence, for a hyperbolic and maximum interaction limited truncation we get:

$$\mathcal{A}^{d,p,q,r} = \{\alpha \in \mathcal{A}^{d,p}, \|\alpha\|_q \leq p \text{ and } \|\alpha\|_0 < r\}. \quad (\text{B14})$$

Notice that only  $d$  is specified as it is the dimension of the input random vector  $\mathbf{X}$ . The maximal polynomial order  $p$  can be found automatically following the iterative approach described in detail in Blatman & Sudret (2011). Finding the optimal values for  $q$  and  $r$ , on the other hand, requires a dedicated parametric study, discussed in Section 3.4.

### B2.2 Calculating the PCE coefficients with sparse regression

Once the polynomial basis has been constructed, the expansion coefficients  $\eta_\alpha$  need to be calculated. Given the high computational costs of the computational model, `EuclidEmulator` employs the sparse-regression approach in Blatman & Sudret (2011), based on the well-known *least-angle regression* technique first introduced in Efron et al. (2004). This approach has been widely demonstrated to be highly efficient even in the presence of high dimensional or highly nonlinear models, as it favours highly sparse models so as to avoid overfitting in the presence of small EDs.

### B2.3 A note on the extrapolation properties of PCE

As mentioned already above, PCE is not an interpolant but a regression technique. This means that a PCE-based surrogate model is able to accurately estimate the response of the input model not just near the positions of the ED points but also further away from them (on a global scale). However, due to the fact that the cosmological parameters have to be mapped to the interval  $[-1, 1]$  in order to be evaluated by the Legendre polynomials (see the explanations in Section 3.3), the regression only works within the predefined parameter bounds. If one wants to predict the

response for a cosmology outside the input bounds, a new emulator has to be trained. This will result in different basis functions and coefficients.

## APPENDIX C: SIMULATIONS AND CONVERGENCE TESTS

Since there is no analytical way to compute a ‘true’ nonlinear power spectrum, a convergence test for the power spectrum is necessary. In Table C1 we list all simulations we have used in this work together with their specifications and the required runtime  $\Delta T$  in node hours. We assign a unique label to each simulation that we use for reference in the text below. We define  $L$  to be the length of a simulation box edge in units of  $h^{-1}$  Mpc,  $N$  denotes the number of particles per dimension used in a simulation to create the initial conditions, and  $R_{\text{grid}} = N_{\text{ma}}/N$  is the ratio between the number of cells  $N_{\text{ma}}$  used for the mass assignment and the number of particles  $N$ .

### C1 Simulation parameters

The goal in this work is to find the minimal volume, the number of particles, and mass assignment grid size that allows us to achieve the required 1 per cent accuracy over the  $k$ -range of interest. A number of further parameters like softening and time-stepping have already been assessed in Schneider et al. (2016). They report that varying the time-stepping and softening parameters has a sub-per cent effect over all  $k$ -scales of interest. We use the PKDGRAV3-default values that have been shown to be reasonable choices: the softening is given by  $\varepsilon = 0.02 \ell_{\text{mean}}$  with  $\ell_{\text{mean}}$  being the mean inter-particle distance. The time-stepping parameter  $\eta = 0.2$  controls each individual particle’s time-step via  $\Delta T = \eta \sqrt{\varepsilon/a}$ , with  $a$  being the gravitational acceleration of the particle.

In what follows we focus on the box volume and mass resolution (i.e. particle number) as well as the size of the mass assignment grid. We perform the convergence test in three steps: first, we determine the minimal simulation box volume by comparing to a paired-and-fixed simulation in LV of  $(4000 h^{-1} \text{Mpc})^3$  with  $4096^3$  particles. Secondly, we find the minimal mass resolution by converging towards an extreme high resolution run (EFHR) with  $8000^3$  particles in a  $(1920 h^{-1} \text{Mpc})^3$  box ( $N_{\text{ma}}^{\text{EFHR}} = 8000$ ), and thirdly, the minimal size of the mass assignment grid is assessed.

For the minimal volume measurement, we compare five paired-and-fixed runs

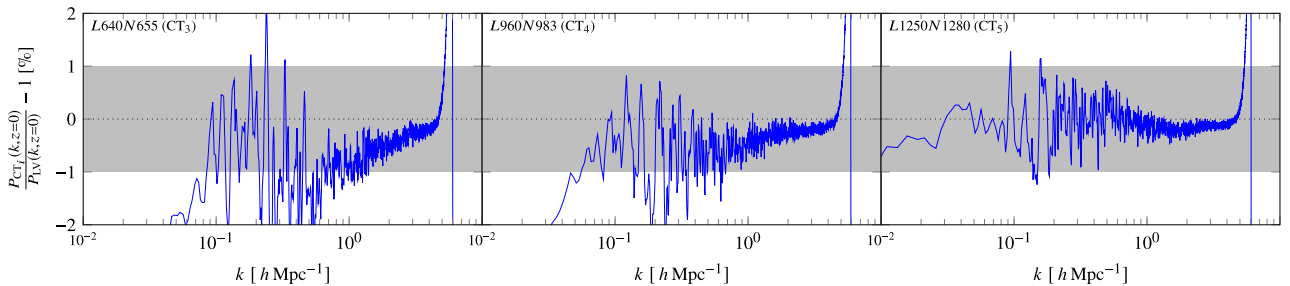
- (i)  $L = 256 h^{-1} \text{Mpc}$  and  $N = 262$  (CT<sub>1</sub>),
- (ii)  $L = 512 h^{-1} \text{Mpc}$  and  $N = 524$  (CT<sub>2</sub>),
- (iii)  $L = 640 h^{-1} \text{Mpc}$  and  $N = 655$  (CT<sub>3</sub>),
- (iv)  $L = 960 h^{-1} \text{Mpc}$  and  $N = 983$  (CT<sub>4</sub>), and
- (v)  $L = 1250 h^{-1} \text{Mpc}$  and  $N = 1280$  (CT<sub>5</sub>)

against the LV-simulation (notice that the mass resolution is the same for all these simulations). According to Fig. C1, we find that a minimal simulation box volume of  $L^3 = (1250 h^{-1} \text{Mpc})^3$  is necessary for the power spectrum to converge to the LV-power spectrum to within 1 per cent at large scales. This result is consistent with the very recent paper Klypin & Prada (2018) and updates the conclusion drawn in Schneider et al. (2016) where they claim a lower bound for cosmological simulation box sizes of only  $500 h^{-1} \text{Mpc}$ . A potential reason for this underestimation is that their reference simulation volume is only  $1024 h^{-1} \text{Mpc}$  and hence most likely too small.

**Table C1.** List of simulations used in this paper. For each simulation, its unique ID, as well as its specifications, are listed. The specifications consist of the box size ( $L$ ), the number of particles per side length ( $N$ ), whether it is a paired-and-fixed run (PF yes/no), and the runtime in node hours. The simulation 000 is a EuclidEmulator-like simulation but run based on the Euclid reference cosmology, which is not part of the ED. Simulations 001 to 100 are the runs that form the actual EuclidEmulator-ED while 101 to 136 were used for the end-to-end test reported in Section 4.1. Euclid Flagship High Resolution (EFHR) and Large Volume (LV) denote the two reference simulations for the convergence tests against which the simulations CT<sub>1</sub> to CT<sub>20</sub> were compared. For all CT-runs with  $N = 1024$ , the power spectra were measured with three different mass assignment grids (see Fig. C3), indicated by the three labels a, b, and c (this did not require separate simulations). Note that the EFHR simulation was run with GPUs. The total run time for all simulations sums up to over 380 000 node hours.

Simulation identifier	$L$ ( $h^{-1}$ Mpc)	$N$	PF	The number of runs	Total runtime (node hours)
000	1250	2048	yes	2	1904
001–100	1250	2048	yes	200	190 200
101–136	1250	2048	yes	72	68 472
Euclid Flagship High Resolution (EFHR)	1920	8000	no	1	93 600*
Large Volume (LV)	4000	4096	yes	2	14 696
Convergence Test 1 (CT <sub>1</sub> )	256	262	yes	2	23
CT <sub>2</sub>	512	524	yes	2	50
CT <sub>3</sub>	640	655	yes	2	80
CT <sub>4</sub>	960	983	yes	2	236
CT <sub>5</sub>	1250	1280	yes	2	508
CT <sub>6abc</sub> (a: $R_{\text{grid}} = 1$ , b: $R_{\text{grid}} = 2$ , c: $R_{\text{grid}} = 7$ )	480	1024	yes	2	378
CT <sub>7</sub>	480	1536	yes	2	1402
CT <sub>8abc</sub>	640	1024	yes	2	308
CT <sub>9</sub>	640	1536	yes	2	1204
CT <sub>10abc</sub>	960	1024	yes	2	240
CT <sub>11</sub>	960	1536	yes	2	896
CT <sub>12</sub>	960	2048	yes	2	2274
CT <sub>13abc</sub>	1440	1024	yes	2	209
CT <sub>14</sub>	1440	1536	yes	2	752
CT <sub>15</sub>	1440	1920	yes	2	1953
CT <sub>16abc</sub>	1920	1024	yes	2	184
CT <sub>17</sub>	1920	1536	yes	2	633
CT <sub>18</sub>	1920	1920	yes	2	1529

Note. \* with GPUs



**Figure C1.** Comparison of three different runs with different box sizes but equal mass resolution  $\ell^{-1} = 1.024 h \text{ Mpc}^{-1}$  against an LV reference simulation with a box of  $L = 4000 h^{-1} \text{ Mpc}$  side length and  $N = 4096$  particles per dimension. Only simulation volumes of at least  $L^3 = (1250 h^{-1} \text{ Mpc})^3$  (right-hand panel) allow a power spectrum measurement that agrees with the one in a large volume at the 1 per cent level. Smaller simulation volumes lead to a power deficit on large scales (left-hand and middle panel) and increasingly larger deviations from the reference on mildly nonlinear scales.

In the second step, we determine the minimal mass resolution, which we measure in terms of  $\ell^{-1} := N/L$  corresponding to the inverse of the mean inter-particle separation. We define a discrete parameter space by

$$L \in \{480, 640, 960, 1440, 1920\} h^{-1} \text{ Mpc},$$

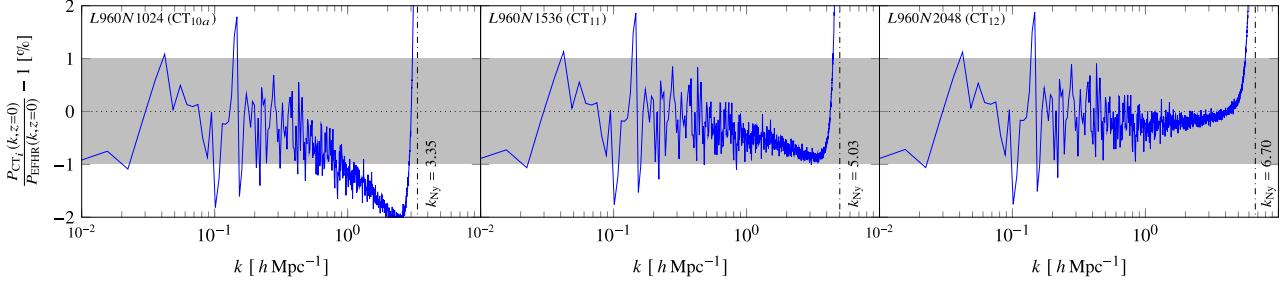
$$N \in \{1024, 1536, 2048\},$$

$$R_{\text{grid}} \in \{1, 2, 4\}$$

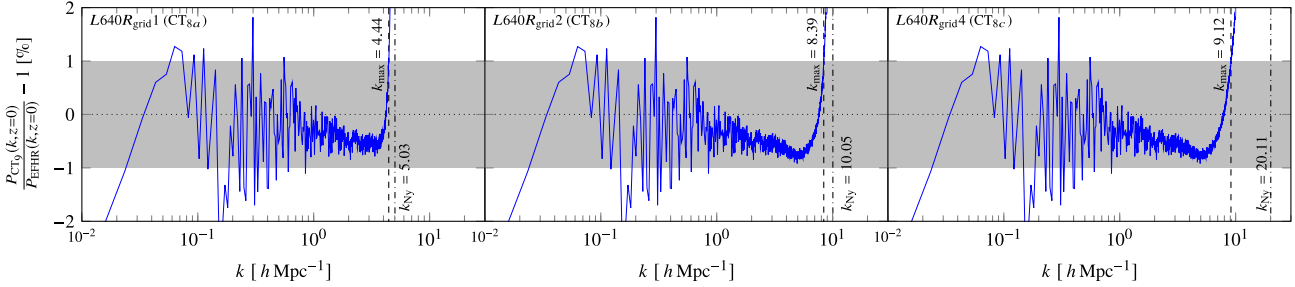
and run  $N$ -body simulations for three selected planes in this space:

- (i) the  $L$ - $N$  plane with  $R_{\text{grid}} = 1$ ,
- (ii) the  $L$ - $R_{\text{grid}}$  plane with  $N = 1024$ , and
- (iii) the  $N$ - $R_{\text{grid}}$  plane with  $L = 960 h^{-1} \text{ Mpc}$ .

We perform this convergence test against the EFHR simulation ( $L = 1920 h^{-1} \text{ Mpc}$  and  $N = 8000$ ,  $\ell_{\text{EFHR}}^{-1} = 4.167$ ). Increasing the number of particles  $N$  used in a simulation of a given volume, we reduce aliasing that comes from discreteness. This is expected because in the limit  $N \rightarrow \infty$  we approximate the real fluid case we are actually interested in. We observe that a resolution parameter of  $\ell^{-1} = 2048/960 = 2.13 h \text{ Mpc}^{-1}$  (Fig. C2, right-hand panel) yields



**Figure C2.** Effect of measuring the power spectrum with different mass resolutions and with different number of particles ( $R_{\text{grid}} = 1$ ). Shown is a subset of the  $L$ – $N$  plane with a fixed  $L$ : all the panels belong to the same box size  $L = 960 h^{-1} \text{ Mpc}$ . From left to right,  $N = 1024, 1536$ , and  $2048$  particles per dimension were used. We thus span an inverse mean inter-particle distance range of  $1.07 h \text{ Mpc}^{-1} \leq \ell^{-1} \leq 2.13 h \text{ Mpc}^{-1}$ . The turn-up of the error curve at high  $k$ -modes indicates the location of the Nyquist frequency (dash-dotted line) of the mass assignment grid.



**Figure C3.** Effect of measuring the power spectrum using different grid sizes, i.e. in the  $L$ – $R_{\text{grid}}$  plane. From left to right, the columns refer the number of sampling grid points being equal to once, twice, and four times the number of points in the particle grid per dimension, respectively. The number of particles is  $N = 1024$  in all subplots. The dashed vertical lines indicate  $k_{\text{max}}$  (the frequency at which the relative error becomes larger than 1 per cent) while the dash-dotted lines again show the Nyquist frequency of the mass assignment grid.

almost perfect results on highly nonlinear scales (up to the point where the Nyquist effect from the mass assignment grid kicks in) while a resolution of only  $\ell^{-1} = 1536/960 = 1.6 h \text{ Mpc}^{-1}$  (Fig. C2, middle panel) leads to an aliasing artefact that only just stays within the 1 per cent region. From this we conclude that in order to meet the 1 per cent accuracy level required by Euclid over all scales of interest we need a resolution parameter  $\ell^{-1} \geq 1.6 h \text{ Mpc}^{-1}$ .

We are then left with assessing how small the mass assignment cells have to be in order to reach the desired precision. We performed another set of simulations for the five different box sizes where the mass assignment cells are either 1, 1/8, or 1/64 times the volume of the particle grid cells, respectively (cf. Fig. C3). The Nyquist frequency  $f_{\text{Ny}}$  of the mass assignment grid itself is linearly proportional to  $N_{\text{ma}}$  (and thus also to  $R_{\text{grid}}$  for a given  $N$ ) according to the Shannon–Nyquist theorem (Nyquist 1928)

$$f_{\text{Ny}} = \frac{1}{2} N_{\text{ma}} \frac{2\pi}{L} = \frac{1}{2} R_{\text{grid}} N \frac{2\pi}{L}, \quad (\text{C1})$$

where  $2\pi/L$  corresponds to the canonical inter-particle scale  $\Delta k$  used by the fast Fourier transform (FFT) such that  $N_{\text{ma}} \Delta k$  equals the maximal Fourier mode for the power spectrum measurement. It is not a priori clear, though, by what factor an increasing value of  $R_{\text{grid}}$  increases the  $k$ -interval within which the error curve remains bounded by 1 per cent. Let us define  $k_{\text{max}}$  to be the maximal  $k$ -value such that

$$100 \left( \frac{P_{\text{CT}_i}(k, z=0)}{P_{\text{EFHR}}(k, z=0)} - 1 \right) > 1, \quad \forall k > k_{\text{max}}. \quad (\text{C2})$$

We find that the proportionality given in equation (C1) translates rather well to a proportionality between  $k_{\text{max}}$  and  $R_{\text{grid}}$  for low-resolution parameters and small values of  $R_{\text{grid}}$  while for high  $\ell^{-1}$

and larger values of  $R_{\text{grid}}$  it breaks down as the slope of the Nyquist turn-up becomes more and more shallow for larger values of  $R_{\text{grid}}$ . For instance, while we can essentially double  $k_{\text{max}}$  in an  $L640N1024$  simulation by going from  $k_{\text{max}}(R_{\text{grid}} = 1) = 4.4 h \text{ Mpc}^{-1}$  (Fig. C3, left-hand panel) to  $k_{\text{max}}(R_{\text{grid}} = 2) = 8.39 h \text{ Mpc}^{-1}$  (Fig. C3, centre panel), we cannot do so again by increasing  $R_{\text{grid}}$  from 2 to 4 (Fig. C3, right-hand panel). In this figure, data from the  $L640N1024$  simulation is shown, which fulfils the resolution condition of  $\ell^{-1} = 1.6 h \text{ Mpc}^{-1}$  found above. One can see that the error stays within 1 per cent up to  $k_{\text{max}}(R_{\text{grid}} = 1) = 4.44 h \text{ Mpc}^{-1}$ ,  $k_{\text{max}}(R_{\text{grid}} = 2) = 8.39 h \text{ Mpc}^{-1}$ , and  $k_{\text{max}}(R_{\text{grid}} = 4) = 9.12 h \text{ Mpc}^{-1}$ . Taking into account that increasing  $R_{\text{grid}}$  leads to a non-negligible increase in required computational resources, we conclude that  $R_{\text{grid}} = 2$  is a reasonable choice.

We thus end up with the following lower bounds:

$$\begin{aligned} \ell^{-1} &\geq 1.6 h \text{ Mpc}^{-1}, \\ R_{\text{grid}} &\geq 2. \end{aligned} \quad (\text{C3})$$

Summarizing the results of our convergence tests, we decide to use the following specifications for the construction of the EuclidEmulator ED:

$$\begin{aligned} L &= 1250 h^{-1} \text{ Mpc}, \\ N &= 2048, \\ R_{\text{grid}} &= 2. \end{aligned} \quad (\text{C4})$$

Notice that this choice obeys the constraint put on  $\ell^{-1}$  as here  $\ell^{-1} = 1.638 h \text{ Mpc}^{-1}$ . This choice of simulation volume and particle number corresponds to a mass resolution of roughly  $2 \times 10^{10} h^{-1} M_{\odot}$  per particle.

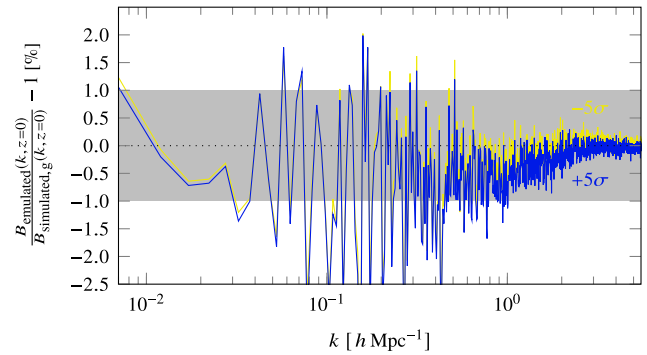
## C2 Redshift dependence of $k_{\max}$

The chosen configuration for the ED simulations suggests that at  $z = 0$  the simulated nonlinear power spectrum can be trusted up to  $k_{\max} \approx 8 h \text{ Mpc}^{-1}$ . Of course, in order to be able to produce a reliable nonlinear correction prediction using the EuclidEmulator at a certain redshift, it is of utmost importance to know how  $k_{\max}$  changes with redshift. We have found that the initial power spectrum at  $z = 200$  measured by PKDGRAV3 has converged to linear theory (as computed by CLASS) up to  $k_{\max} = 5.48 h \text{ Mpc}^{-1}$ . The convergence test in the previous section suggests that the nonlinear power spectrum at lower  $z$  converges up to larger  $k_{\max}$  (for  $z = 0$ , we find  $k_{\max} \approx 8 h \text{ Mpc}^{-1}$ ). The division by the initial power spectrum in the computation of the nonlinear correction renders the latter to be converged up to  $k_{\max} = 5.48 h \text{ Mpc}^{-1}$  for all redshifts up to  $z = 5$ . Based on this, the allowed redshift and  $k$  range for emulation with the EuclidEmulator is set to  $0 \leq z \leq 5$  and  $0.01 h \text{ Mpc}^{-1} \leq k \leq 5 h \text{ Mpc}^{-1}$ , respectively. For  $z > 5$ , we found a non-trivial dependence of  $k_{\max}$  on the redshift. However, it is not clear yet to what extent this functional relation is influenced by numerical artefacts (like e.g. aliasing or transients) and to what extent it is physical.

## APPENDIX D: PAIRING-AND-FIXING VERSUS GAUSSIAN INITIAL CONDITION-BASED SIMULATIONS

In this appendix, we address potential issues of pairing-and-fixing (introduced in Pontzen et al. 2016) and compare emulated nonlinear corrections to nonlinear corrections coming from a traditional, Gaussian initial condition sample. In Angulo & Pontzen (2016) it is explained that fixing the power spectrum amplitude in the initial conditions allows us to approximate the ensemble mean of a set of power spectra with Gaussian initial conditions with no variance at the cost of introducing some non-Gaussianity into the initial conditions. As is shown in the lower panel of fig. 2 in Angulo & Pontzen (2016), a deviation of the PF mean from the ensemble mean of power spectra can be observed at high  $k$  but it stays inside the 0.1 per cent region up to  $k \gtrsim 1 h \text{ Mpc}^{-1}$ .

Fig. D1 is a plot similar to Fig. 6 with the difference that here we compare to the nonlinear correction of a single run with Gaussian random field initial conditions (for the ‘+5 $\sigma$ ’ and ‘-5 $\sigma$ ’ cases; we use the same notation as was used in Fig. 6). We find that on large scales the computational cosmic variance does not play a big role. This is due to the fact that in order to compute the nonlinear correction, one divides the power spectrum at a given redshift  $z$  by the initial condition of the simulation and this cancels out most of the variance. However, the biggest deviations are observed on mildly nonlinear scales. On these intermediate scales, the variation is initially small but is amplified nonlinearly during the evolution. Hence, division by initial condition is not enough to efficiently cancel it. Pairing-and-fixing does decrease the cosmic variance on these scales to some degree. Villaescusa-Navarro et al. (2018) have studied the  $k$ -dependence on how much paired-and-fixed simulations feature less computational cosmic



**Figure D1.** A similar comparison as in Fig. 6. Shown is the relative difference between the emulated nonlinear corrections and those computed from Gaussian random field initial condition-based simulations (for the two extreme cases ‘+5 $\sigma$ ’ and ‘-5 $\sigma$ ’). The high-frequency errors are due to computational cosmic variance that (in our case) mainly has an effect on intermediate scales.

variance compared to usual Gaussian random field simulations. They find that towards smaller scales, the improvement brought by pairing-and-fixing degrades to the point where the pairing-and-fixing approach performs equally well as the classical Gaussian random field approach. In the context of our work, however, this is not a problem as on smaller scales cosmic variance does not play an important role in the first place.

<sup>1</sup>Institute for Computational Science, University of Zurich, Winterthurerstrasse 190, CH-8057 Zurich, Switzerland

<sup>2</sup>Chair of Risk, Safety and Uncertainty Quantification, Department of Civil Engineering, ETH Zurich, Stefano-Francini-Platz 5, CH-8093 Zurich, Switzerland

<sup>3</sup>Kapteyn Astronomical Institute, University of Groningen, Landleven 12, NL-9747 Groningen, the Netherlands

<sup>4</sup>Institute for Particle Physics and Astrophysics, Department of Physics, ETH Zurich, Wolfgang-Pauli-Strasse 27, CH-8093 Zurich, Switzerland

<sup>5</sup>Institute of Space Sciences (ICE, CSIC), Campus UAB, Carrer de Can Magrans, s/n, E-08193 Barcelona, Spain

<sup>6</sup>Institut d’Estudis Espacials de Catalunya (IEEC), E-08034 Barcelona, Spain

<sup>7</sup>Mullard Space Science Laboratory, University College London, Holmbury St. Mary, Dorking RH5 6NT, UK

<sup>8</sup>INAF, Istituto di Radioastronomia, Via Piero Gobetti 101, I-40129 Bologna, Italy

<sup>9</sup>Dipartimento di Fisica e Scienze della Terra, Università di Ferrara, Via Giuseppe Saragat 1, I-44122 Ferrara, Italy

<sup>10</sup>Istituto Nazionale di Fisica Nucleare, Sezione di Bologna, Via Irnerio 46, I-40126 Bologna, Italy

<sup>11</sup>Instituto de Astrofísica e Ciências do Espaço, Faculdade de Ciências, Universidade de Lisboa, Tapada da Ajuda, P-PT-1349-018 Lisboa, Portugal

<sup>12</sup>Department of Physics and Helsinki Institute of Physics, Gustaf Hällströmin katu 2, FI-00014 University of Helsinki, Finland

<sup>13</sup>INFN – Bologna, Istituto Nazionale di Fisica Nucleare, V.le Berti Pichat 6/2, I-40127 Bologna, Italy

This paper has been typeset from a  $\text{\TeX}/\text{\LaTeX}$  file prepared by the author.