A precise symbolic emulator of the linear matter power spectrum

Deaglan J. Bartlett, Lukas Kammerer, Gabriel Kronberger, Harry Desmond, Pedro G. Ferreira, Benjamin D. Wandelt, Bogdan Burlacu, David Alonso, Matteo Zennaro

A&A, 686, A209 (2024)
https://doi.org/10.1051/0004-6361/202348811

Astronomy & Astrophysics

A precise symbolic emulator of the linear matter power spectrum

Deaglan J. Bartlett[1], Lukas Kammerer[2], Gabriel Kronberger[2], Harry Desmond[3], Pedro G. Ferreira[4], Benjamin D. Wandelt[1,5], Bogdan Burlacu[2], David Alonso[4], and Matteo Zennaro[4]

[1] CNRS & Sorbonne Université, Institut d'Astrophysique de Paris (IAP), UMR 7095, 98 bis bd Arago, 75014 Paris, France
e-mail: deaglan.bartlett@iap.fr
[2] Heuristic and Evolutionary Algorithms Laboratory, University of Applied Sciences Upper Austria, Hagenberg, Austria
[3] Institute of Cosmology & Gravitation, University of Portsmouth, Dennis Sciama Building, Portsmouth PO1 3FX, UK
[4] Astrophysics, University of Oxford, Denys Wilkinson Building, Keble Road, Oxford OX1 3RH, UK
[5] Center for Computational Astrophysics, Flatiron Institute, 162 5th Avenue, New York, NY 10010, USA

Received 1 December 2023 / Accepted 3 April 2024

ABSTRACT

Context. Computing the matter power spectrum, $P(k)$, as a function of cosmological parameters can be prohibitively slow in cosmological analyses, hence emulating this calculation is desirable. Previous analytic approximations are insufficiently accurate for modern applications, so black-box, uninterpretable emulators are often used.
Aims. We aim to construct an efficient, differentiable, interpretable, symbolic emulator for the redshift zero linear matter power spectrum which achieves sub-percent level accuracy. We also wish to obtain a simple analytic expression to convert A_s to σ_8 given the other cosmological parameters.
Methods. We utilise an efficient genetic programming based symbolic regression framework to explore the space of potential mathematical expressions which can approximate the power spectrum and σ_8. We learn the ratio between an existing low-accuracy fitting function for $P(k)$ and that obtained by solving the Boltzmann equations and thus still incorporate the physics which motivated this earlier approximation.
Results. We obtain an analytic approximation to the linear power spectrum with a root mean squared fractional error of 0.2% between $k = 9 \times 10^{-3}$–$9\,h\,\mathrm{Mpc}^{-1}$ and across a wide range of cosmological parameters, and we provide physical interpretations for various terms in the expression. Our analytic approximation is 950 times faster to evaluate than CAMB and 36 times faster than the neural network based matter power spectrum emulator BACCO. We also provide a simple analytic approximation for σ_8 with a similar accuracy, with a root mean squared fractional error of just 0.1% when evaluated across the same range of cosmologies. This function is easily invertible to obtain A_s as a function of σ_8 and the other cosmological parameters, if preferred.
Conclusions. It is possible to obtain symbolic approximations to a seemingly complex function at a precision required for current and future cosmological analyses without resorting to deep-learning techniques, thus avoiding their black-box nature and large number of parameters. Our emulator will be usable long after the codes on which numerical approximations are built become outdated.

Key words. methods: numerical – cosmological parameters – cosmology: theory – large-scale structure of Universe

1. Introduction

Machine learning (ML) methods have great potential for simplifying and accelerating the analysis of astrophysical data sets. The primary focus has been on what one might dub "advanced numerical methods" using, for example, Gaussian processes or neural networks. In these cases, one tries to construct efficient algorithms which can be used to either infer specific physical properties from complex data sets or emulate complex processes which can then be extrapolated to new situations. Typically these methods involve constructing a set of pre-established basis functions and then inferring their weights, or building complex, expressible functions, with parameters that can be optimised via efficient gradient descent methods. These methods can be easily incorporated in Bayesian inference frameworks that have achieved significant success, becoming the standard practice in astrostatistics.

The drawback of the more traditional, numerical ML techniques is their opaqueness; it is not always clear what information is being used and how methods trained on (necessarily imperfect) simulations will perform when applied to real-world data. A somewhat overlooked branch of machine learning which has tremendous promise for the types of problems being considered in astrophysics is symbolic regression (SR). With SR one tries to infer the mathematical expressions that best capture the properties of the physical system one is trying to study. The process is an attempt to mimic and systematise the practice that physicists have always used: to infer simple physical laws (i.e., formulae) from data. The field of SR has developed over the years into a vibrant and active field of research in ML, typically associated with evolutionary methods such as Genetic Programming. It has been shown that it can be used to infer some well-established laws of physics from data and infer new ones (e.g., Lemos et al. 2023; Bartlett et al. 2023a,b; Desmond et al. 2023; Kamerkar et al. 2023; Sousa et al. 2024; Delgado et al. 2022; Miniati & Gregori 2022; Wadekar et al. 2023; Koksbang 2023a,b,c; Alestas et al. 2022; Lodha et al. 2024).

Within the field of cosmology, one often compresses observations from galaxy surveys into two-point correlation functions (or their Fourier transforms, power spectra), which are compared to theory through Markov chain Monte Carlo methods to constrain cosmological parameters. As cosmological

surveys become increasingly vast and precise, a fundamental limitation to the feasibility of such inferences has been the speed at which one can make this theoretical prediction, since it involves solving a complex set of coupled, highly non-linear differential equations. Recently, instead of directly solving these equations (Lewis et al. 2000; Blas et al. 2011; Hahn et al. 2023) and adding non-linear corrections (Smith et al. 2003), emulation techniques such as Neural Networks, Gaussian Processes or polynomial interpolation schemes have been used to accelerate these calculations to directly output the matter power spectrum as a function of cosmological parameters (Fendt & Wandelt 2007a,b; Heitmann et al. 2014; Winther et al. 2019; Angulo et al. 2021; Aricò et al. 2022; Euclid Collaboration 2021; Spurio Mancini et al. 2022; Mootoovaloo et al. 2022; Zennaro et al. 2023). These methods act as black boxes and require up to several hundreds of parameters to be optimised.

However, through perturbation theory, one knows analytic limits of the power spectrum and, through visual inspection, it does not appear to be an extremely complex function. As such, one wonders whether an analytic approximation exists. Indeed, for many years, the leading method of accelerating this calculation has been an analytic approximation (Eisenstein & Hu 1998, 1999), however it is insufficiently precise for modern experiments. Analytic approximations to beyond ΛCDM power spectra have been proposed in the context of modified gravity (Orjuela-Quintana et al. 2024), although these still only achieve a precision of between 1 and 2%.

Such an emulator has the advantage that it will not become deprecated when the codes on which current numerical methods are built become outdated, whereas other methods require the transfer of the inferred weights and biases as well as the model architecture, hindering longevity. Even in the short term, an analytic expression using standard operators is more portable, since it can be more easily be incorporated into the user's favourite programming language without the need to install or write wrappers for the model. Moreover, having an analytic expression allows one to interpret such a fit, and potentially identify physical processes which could lead to certain terms, contrary to the black-box numerical methods. Additionally, such expressions often contain fewer free parameters to optimise than numerical ML methods.

In Sect. 2 we briefly describe the matter power spectrum and the Eisenstein & Hu approximation, and in Sect. 3 we detail the SR method we use in this work. We present an analytic emulator for σ_8 as a function of other cosmological parameters in Sect. 4 (which is easily invertible to obtain A_s as a function of cosmological parameters), and in Sect. 5 we give our emulator for the linear matter power spectrum. The main results of this paper are given in Eqs. (4) and (6). We conclude and discuss future work in Sect. 6. Throughout this paper 'log' denotes the natural logarithm.

2. The matter power spectrum

2.1. Definition

We would like to construct an efficient, differentiable and (if at all possible) interpretable emulator for the power spectrum of the matter distribution in the Universe, $P(k;\boldsymbol{\theta})$, for wavenumber k and cosmological parameters $\boldsymbol{\theta}$.

The power spectrum is defined as follows: the matter density of the Universe, $\rho(\boldsymbol{x})$ can be decomposed into a constant (in space) background density, $\bar{\rho}$, and a density contrast, $\delta(\boldsymbol{x})$ such that $\rho(\boldsymbol{x}) = \bar{\rho}[1 + \delta(\boldsymbol{x})]$. If $\tilde{\delta}(\boldsymbol{k})$ is the Fourier Transform of $\delta(\boldsymbol{x})$, and the matter distribution is statistically homogeneous and isotropic, we have that

$$(2\pi)^3 P(k;\boldsymbol{\theta})\delta^{\mathrm{D}}(\boldsymbol{k}-\boldsymbol{k}') \equiv \langle\tilde{\delta}(\boldsymbol{k})\tilde{\delta}^*(\boldsymbol{k}')\rangle, \tag{1}$$

where $\langle\cdots\rangle$ denotes an ensemble average and δ^{D} is the Dirac delta function.

From observations of the cosmic microwave background (CMB; Planck Collaboration VI 2020), it is known that the density fluctuations at early times were approximately Gaussian and thus fully described by $P(k;\boldsymbol{\theta})$. At these early times, the power spectrum of the comoving curvature perturbations is proportional to $A_s k^{n_s-4}$, where $n_s \approx 0.9665$ (Planck Collaboration VI 2020). Although structure formation through gravity makes the present-day density field non-Gaussian (e.g., the intricate structure of the cosmic web is typically associated with higher order statistics), the power spectrum still holds a central role in modern cosmological analyses.

The current cosmological model is described by only six parameters: the baryonic, Ω_b, and total matter, Ω_m, density parameters, the Hubble constant, $H_0 = 100\,h\,\mathrm{km\,s^{-1}\,Mpc^{-1}}$, the scalar spectral index, n_s, the curvature fluctuation amplitude, A_s, and the reionisation optical depth, τ. All other parameters can be derived from these six, and thus sometimes a different set of parameters is chosen. For example, instead of A_s, one often quotes σ_8 which is the root-mean-square density fluctuation when the linearly evolved field is smoothed with a top-hat filter of radius $8\,h^{-1}$ Mpc. Specifically, one defines for a top-hat of radius R

$$\sigma_{\mathrm{R}}^2 = \int \mathrm{d}k\,\frac{k^2}{2\pi^2}P(k;\boldsymbol{\theta})\,|W(k,R)|^2\,, \tag{2}$$

where $\boldsymbol{\theta}$ is the set of cosmological parameters and the Fourier transfer of the top-hat filter is

$$W(k,R) = \frac{3}{(kR)^3}\left(\sin(kR) - kR\cos(kR)\right), \tag{3}$$

and σ_8 is simply σ_R for $R = 8\,h^{-1}$ Mpc. Throughout this paper we ignore the small dependence of the power spectrum on the reionisation optical depth parameter, and focus on the remaining five parameters. We set the neutrino mass to zero in all calculations.

2.2. Eisenstein & Hu approximation

Since each evaluation of a Boltzmann solver to compute $P(k;\boldsymbol{\theta})$ can be expensive, the ability to emulate this procedure and replace this solver with a surrogate model has long been desirable. The most notable attempt to do this in an analytic manner is given in a series of papers by Eisenstein & Hu (1998, 1999). In these works, an approximation is constructed based on physical arguments including baryonic acoustic oscillations (BAO), Compton drag, velocity overshoot, baryon infall, adiabatic damping, Silk damping, and cold dark matter (CDM) growth suppression. Rather than repeat their findings, we refer the reader to these papers to inspect the structure of the equations. Such a model is accurate to a few percent which, although invaluable at the time of writing, is insufficiently accurate for modern cosmological analyses. It is thus the goal of this work to build upon this analytic emulator to provide sub-percent level predictions. We note that alternative symbolic approximations also exist to $P(k)$, such as the earlier, less accurate approximation by Bardeen et al. (1986; BBKS).

More recently, Orjuela-Quintana et al. (2023) found simple expressions using genetic programming which can achieve similar accuracy to the Eisenstein & Hu (1998) expression, but we choose to use Eisenstein & Hu (1998)'s approximation due to its physical motivation and widespread use.

3. Symbolic regression

To extract analytic approximations from sampled data, we use the symbolic regression package OPERON[1] (Burlacu et al. 2020). This package leverages the most popular (e.g., Lemos et al. 2023; Cranmer et al. 2020; Cranmer 2020, 2023; Schmidt & Lipson 2009; Schmidt et al. 2011; Virgolin et al. 2021; de Franca & Aldeia 2021; La Cava et al. 2019; Kommenda et al. 2020; Arnaldo et al. 2014) approach to SR, namely genetic programming (Turing 1950; David 1989; Haupt & Haupt 2004). Genetic programming describes the evolution of "computer programs", in our case mathematical expressions encoded as expression trees. Following the principle of natural selection, over several iterations the worst performing equations (given some fitness metric) are discarded and new equations are produced by combining sub-expressions of the current population (crossover) or by randomly inserting, replacing or deleting a subtree in an expression (mutation). Over the course of several generations, the expectation is that the population of equations evolve to become fitter and thus we obtain increasingly accurate analytic expressions.

We note that many other techniques exist for SR, such as supervised or reinforcement learning with neural networks (Petersen et al. 2021; Landajuela et al. 2022; Tenachi et al. 2023; Biggio et al. 2021), deterministic approaches (Worm & Chiu 2013; Kammerer et al. 2021; Rivero et al. 2022; McConaghy 2011), Markov chain Monte Carlo (Jin et al. 2019), physics-inspired searches (Udrescu & Tegmark 2020; Udrescu et al. 2020; René Broløs et al. 2021), and exhaustive searches (Bartlett et al. 2023a). However, we choose OPERON and thus genetic programming due to its speed, high memory efficiency and its strong performance in benchmark studies (Cava et al. 2021; Burlacu 2023).

To improve the search, every time a terminal node appears in an expression tree (i.e., k or one of the cosmological parameters), a scaling parameter is introduced, which is then optimised (Kommenda et al. 2020) using the Levenberg–Marquardt algorithm (Levenberg 1944; Marquardt 1963). We denote the total number of nodes in the expression excluding the scaling as the "length" of the model, and the "complexity" refers to the total number of nodes, including these.

When comparing objective values during non-dominated sorting (NSGA2), OPERON implements the concept of ϵ-dominance (Laumanns et al. 2002), where the parameter ϵ is defined such that two objective values which are within ϵ of each other are considered equal. This parameter therefore affects the number of duplicate equations in the population and is designed to promote convergence to a representative well distributed approximation of the global Pareto front: the set of solutions which cannot be made more accurate without being made more complex. We choose different values for this parameter when searching for our two emulators, and these were found after some experimentation with different values to find settings which produced accurate yet compact models.

[1] https://github.com/heal-research/operon

Model selection is an essential part of any SR search. Since one optimises both accuracy and simplicity during the search, SR is often a Pareto-optimisation problem. In the presence of statistical errors, one can combine simplicity and accuracy in a principled, information theory motivated way into a single objective to optimise under the minimum description length principle (Bartlett et al. 2023a). In this case, the task of picking the optimum function is unambiguous, and one can incorporate prior information into the functional form using language models (Bartlett et al. 2023b) to obtain more physically motivated functions. In our problem, however, we do not have noise in our data and thus have to rely on heuristic methods.

To choose a model, we first generate the Pareto front of candidate expressions. We then consider only those models for which the loss is below some predefined level (to ensure sufficiently accurate solutions for our applications) and those for which the loss on the training and validation sets do not differ significantly (and indicator of over-fitting). At this point one could automate model selection; for example, in the code PYSR (Cranmer et al. 2020; Cranmer 2020) the best model is the one with the best "score", which is the one with largest negative of the derivative of the loss with respect to complexity. However, given we wish to have interpretable and physically-reasonable functions, we instead visually inspect the most accurate solution found for each model length and make a qualitative judgement as to the function which is sufficiently compact to be interpretable yet is accurate enough for our applications.

Further details are given in Sects. 4 and 5.

4. Analytic emulator for σ_8

We begin by considering the simplest emulator one may want for power spectrum related quantities: an emulator for σ_8 as a function of other cosmological parameters ($A_s, \Omega_b, \Omega_m, h, n_s$) or, equivalently, an emulator for A_s given σ_8 and the other cosmological parameters. Although the set of neural network emulators BACCO contains a function to do this (Aricò et al. 2022), to the best of the authors' knowledge an analytic approximation is not currently in common use. The standard approach is to compute the linear matter power spectrum with a Boltzmann code assuming some initial guess of A_s, then compute the integral in Eq. (2) to obtain σ_8. For a target σ_8 of σ_8', one should then use $A_s' = (\sigma_8'/\sigma_8)^2 A_s$.

We wish to accelerate this process with a symbolic emulator. To compute this, we constructed a Latin hypercube (LH) of 100 sets of cosmological parameters, using uniform priors in the ranges given in Table 1, which are the same as those used in Euclid Collaboration (2021). We constructed a second LH of 100 points to be used for validation. For these parameters, we computed σ_8 using CAMB (Lewis et al. 2000) and attempted to learn this mapping using a mean squared error loss function with OPERON.

For the equation search, we used a population size of 1000 with a brood size of 10 and tournament size of 5, optimising both the mean squared error and the length of the expression simultaneously, with $\epsilon = 10^{-6}$ (see Sect. 3). From Eq. (2), one would expect that A_s only appears in the expression for σ_8 as $\sigma_8 \propto \sqrt{A_s}$ since A_s linearly scales the power spectrum. As such, we chose to fit for $\sigma_8/\sqrt{10^9 A_s}$, where we use $10^9 A_s$ instead of A_s so that all cosmological parameters and the target variable are $O(1)$. Parameters were optimised during the search using a nonlinear least squared optimiser with up to 1000 iterations per optimisation attempt. We set the maximum allowed model length to 40 and

Table 1. Cosmological parameters used for analytic emulators.

Parameter	Minimum	Maximum
$10^9 A_s$	1.7	2.5
Ω_m	0.24	0.40
Ω_b	0.04	0.06
h	0.61	0.73
n_s	0.92	1.00

Notes. We sample all parameters independently and uniformly in the range between the minimum and maximum values given.

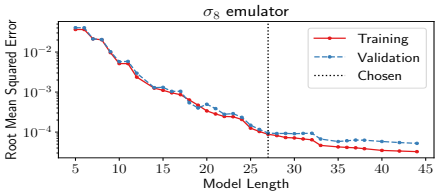

Fig. 1. Pareto front of solutions obtained using OPERON when fitting $\sigma_8/\sqrt{10^9 A_s}$ as a function of Ω_b, Ω_m, h and n_s. We plot the root mean squared error as a function of model length from the training and validation sets separately. The model in Eq. (4) has a model length of 27.

maximum number of iterations to 10^8, although we found that both of these are much larger than the required values needed to converge to a desirable solution.

The candidate expressions were comprised of standard arithmetic operations (addition, subtraction, multiplication), as well as the natural logarithm, square and square root operators. It is somewhat difficult to predict the exact effect the function set has on the quality of the results. The efficiency of the algorithm is only affected insofar as the transcendental functions (sin, cos, log, exp, $\sqrt{\cdot}$, etc.) are slower to evaluate than arithmetic operators. The effect is minor, however. Increasing the number of basis functions inflates the total search space, making it potentially less likely that well-fitting expressions are found, yet too small a basis set could remove compact approximations to the functions of interest. For both this section and Sect. 5 we experimented with alternative basis sets and found that those chosen gave accurate yet compact expressions within a reasonable run time.

After 2 min of operation on one node of 56 cores, we found the Pareto front of expressions given in Fig. 1. We see that the training and validation losses are comparable at all model lengths, reaching a root mean squared error of around 10^{-3} by a model length of 14. We see that the difference between the training and validation losses increases after the model of length 27, so we take this model as our fiducial result. It is given by

$$\frac{\sigma_8}{\sqrt{10^9 A_s}} \approx a_0\Omega_m + a_1 h + a_2\left(\Omega_m - a_3\Omega_b\right)\left(\log(a_4\Omega_m) - a_5 n_s\right) \times \left(n_s + a_6 h\left(a_7\Omega_b - a_8 n_s + \log(a_9 h)\right)\right), \quad (4)$$

where the optimised parameters are $\boldsymbol{a} = [0.51172,\ 0.04593,\ 0.73983,\ 1.56738,\ 1.16846,\ 0.59348,\ 0.19994,\ 25.09218,\ 9.36909,\ 0.00011]$. We note that we have removed a final additive term produced by OPERON since this has a value of 3×10^{-6} and is thus much smaller than the error in the fit so can be safely neglected.

We note several important features of this equation which make it desirable. First, we find that it is a highly accurate approximation, with a root mean squared fractional error on the validation set of only 0.1% which is far smaller than the precision to which one can measure this number with cosmological experiments. Second, one sees that A_s (by design) only appears once in this equation and as a multiplicative term. Thus, it is trivial to invert this equation to obtain A_s as a function of the other cosmological parameters, as is often needed.

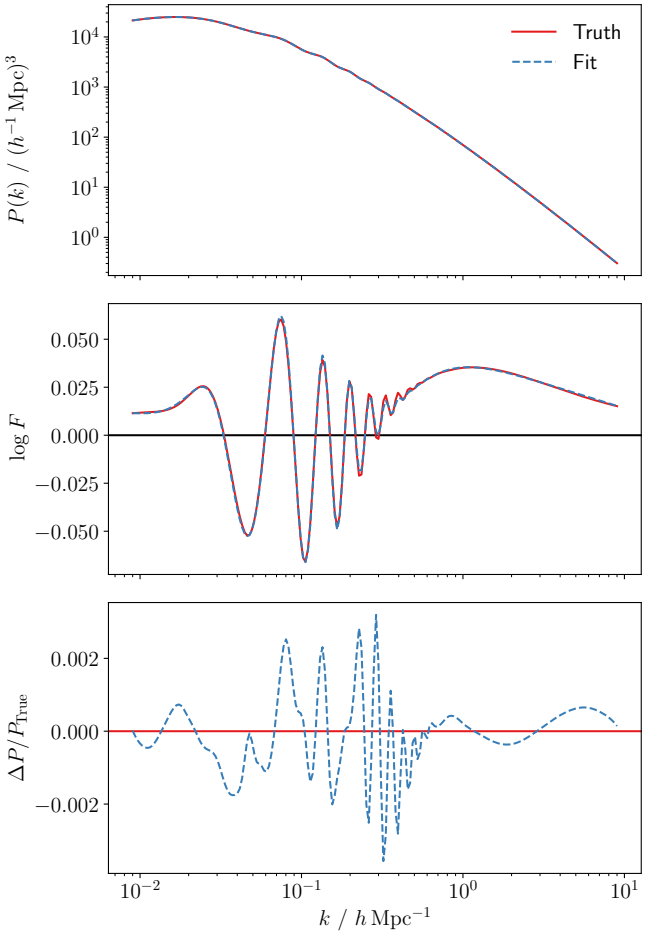

Fig. 2. Linear matter power spectrum (upper), the residuals Eq. (5) from the Eisenstein & Hu fit without baryons (middle), and the fractional residuals on $P(k)$ compared to the truth for the *Planck* 2018 (Planck Collaboration VI 2020) cosmology. In all panels we plot the truth computed with CAMB with solid red lines, and the analytic fit Eq. (6) obtained in this paper with dashed blue lines. We see that the fit is accurate within 0.3% across all k considered.

5. Analytic emulator for the linear power spectrum

We now move on to the more challenging task of producing an analytic emulator for the linear matter power spectrum. Given the previous success of Eisenstein & Hu (1998, 1999), we believe it is sensible to build upon this work, not least due to the physically-motivated terms included in their fit and so that we must only have to fit a small residual (of the order of a few percent). Thus, instead of directly fitting for $P(k, \boldsymbol{\theta})$, we define

$$P(k;\boldsymbol{\theta}) \equiv P_{\mathrm{EH}}(k;\boldsymbol{\theta})F(k;\boldsymbol{\theta}), \quad (5)$$

where $P_{\mathrm{EH}}(k;\boldsymbol{\theta})$ is the zero-baryon fit of Eisenstein & Hu (1998), which does not include an attempt to fit the BAO. We plot both $P(k;\boldsymbol{\theta})$ and $\log F(k;\boldsymbol{\theta})$ in Fig. 2 for the best-fit cosmology obtained by *Planck* (Planck Collaboration VI 2020), where

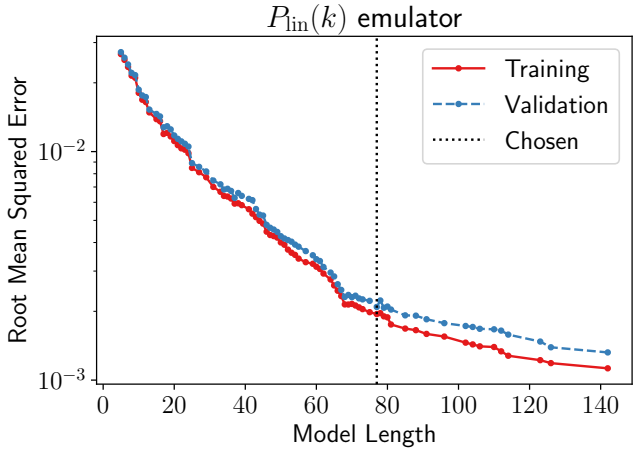

Fig. 3. Pareto front of solutions obtained using OPERON when fitting the linear matter power spectrum as a function of σ_8, $\Omega_{\rm b}$, $\Omega_{\rm m}$, h and $n_{\rm s}$. We plot the root mean squared error on $\log F$ as a function of model length for the training and validation sets separately. The model given in Eq. (6) has a model length of 77, as indicated by the dotted line.

we see that dividing out the Eisenstein & Hu term retains the BAO part of the power spectrum and reduces the dynamic range required for the fit.

As before, we obtained 100 sets of cosmological parameters on a LH using the priors in Table 1 and computed both $P(k;\boldsymbol{\theta})$ with CAMB and $P_{\rm EH}(k;\boldsymbol{\theta})$ with the COLOSSUS (Diemer 2018) implementation, using 200 logarithmically spaced values of k in the range 9×10^{-3}–$9\,h\,{\rm Mpc}^{-1}$. We note that this is an extremely small training set compared to many power spectrum emulators, but we find that it is sufficient to obtain sub-percent level fits.

We chose to symbolically regress $\log F(k;\boldsymbol{\theta})$ using a mean squared error loss function, and thus wish to minimise the fractional error on this residual. We chose to fit for $\log F$ as this ensures that our final estimate of $P(k;\boldsymbol{\theta})$ is positive, as guaranteed by exponentiation, which is physically required. Additionally, we first multiplied $\log F$ by 100 so that the target was $O(1)$. We used a further 100 sets of cosmological parameters, also arranged on a LH, for validation. We chose to fit using the cosmological parameters σ_8, $\Omega_{\rm b}$, $\Omega_{\rm m}$, h and $n_{\rm s}$. We used the same settings for OPERON as in Sect. 4, except we chose $\epsilon = 10^{-3}$, terminated our search after 10^8 function evaluations and used a basis set comprising of addition, subtraction, multiplication, natural logarithm, cosine, power and analytic quotient operators $({\rm aq}(x,y)\equiv x/\sqrt{1+y^2})$.

The root mean squared error for the best function found at each model length is given in Fig. 3, where we see that we are able to achieve values of $O(10^{-3})$ for $\log F$. Unlike for the σ_8 emulator, we obtain slightly worse losses for the validation set compared to training, however always by less than a factor of two.

Given this set of candidate solutions, we wish to choose one which is sufficiently accurate for current applications yet is sufficiently compact to be interpretable. In Fig. 3, one observes a plateau in accuracy between model lengths ∼65–80 and thus it seems reasonable to choose a solution in this regime, since doubling the model length only achieves approximately a factor of two improvement in fit beyond this point. Moreover, beyond this point the training and validation curves begin to deviate, suggesting a degree of overfitting.

Table 2. Best-fit parameters for the linear matter power spectrum emulator given in Eq. (6).

Parameter	Value	Parameter	Value
b_0	0.0545	b_{19}	0.0111
b_1	0.0038	b_{20}	5.35
b_2	0.0397	b_{21}	6.421
b_3	0.1277	b_{22}	134.309
b_4	1.35	b_{23}	5.324
b_5	4.0535	b_{24}	21.532
b_6	0.0008	b_{25}	4.742
b_7	1.8852	b_{26}	16.6872
b_8	0.1142	b_{27}	3.078
b_9	3.798	b_{28}	16.987
b_{10}	14.909	b_{29}	0.0588
b_{11}	5.56	b_{30}	0.0007
b_{12}	15.8274	b_{31}	195.498
b_{13}	0.0231	b_{32}	0.0038
b_{14}	0.8653	b_{33}	0.2767
b_{15}	0.8425	b_{34}	7.385
b_{16}	4.554	b_{35}	12.3961
b_{17}	5.117	b_{36}	0.0134
b_{18}	70.0234		

Notes. Although units are excluded in this table, the units for each parameter are easily obtained by noting that these are defined assuming that k is measured in $h\,{\rm Mpc}^{-1}$ in Eq. (6).

We choose to report the model of length 77, as indicated by the dotted line in Fig. 3, since this provided one of the most interpretable solutions, and achieved a sub-percent error for 95% (2σ) of the cosmological parameters considered, for both the training and validation set. After some simplification, this can be written as

$$\begin{aligned}\log F \approx\ & b_0 h - b_1 \\ & + \left(\frac{b_2\Omega_{\rm b}}{\sqrt{h^2+b_3}}\right)^{b_4\Omega_{\rm m}}\left[\frac{b_5 k-\Omega_{\rm b}}{\sqrt{b_6+(\Omega_{\rm b}-b_7k)^2}} b_8(b_9k)^{-b_{10}k}\cos\left(b_{11}\Omega_{\rm m}\right.\right. \\ & \left.\left. -\frac{b_{12}k}{\sqrt{b_{13}+\Omega_{\rm b}^2}}\right) - b_{14}\left(\frac{b_{15}k}{\sqrt{1+b_{16}k^2}}-\Omega_{\rm m}\right)\cos\left(\frac{b_{17}h}{\sqrt{1+b_{18}k^2}}\right)\right] \\ & + b_{19}(b_{20}\Omega_{\rm m}+b_{21}h-\log(b_{22}k)+(b_{23}k)^{-b_{24}k})\cos\left(\frac{b_{25}}{\sqrt{1+b_{26}k^2}}\right) \\ & + (b_{27}k)^{-b_{28}k}\left(b_{29}k-\frac{b_{30}\log(b_{31}k)}{\sqrt{b_{32}+(\Omega_{\rm m}-b_{33}h)^2}}\right)\cos\left(b_{34}\Omega_{\rm m}-\frac{b_{35}k}{\sqrt{b_{36}+\Omega_{\rm b}^2}}\right),\end{aligned} \tag{6}$$

where the best-fit parameters for this function are given in Table 2. We find that there are 37 different parameters required for this fit, far fewer than would be used if one were to emulate this with a neural network. We note that, if we used a method based on the "score" approach of PYSR (Cranmer et al. 2020; Cranmer 2023; see Sect. 3) to choose our model – defining the loss to be the mean squared error on the training set and computing the derivative with respect to model length – then we would have chosen the model of length 80. This has an almost identical functional form to Eq. (6), so our results are not sensitive to the exact model selection method.

We plot this fit and the residuals compared to CAMB for the *Planck* 2018 cosmology in Fig. 2, which we note is not included in either our training or validation sets. One can see that the difference between the true power spectrum and our analytic fit is

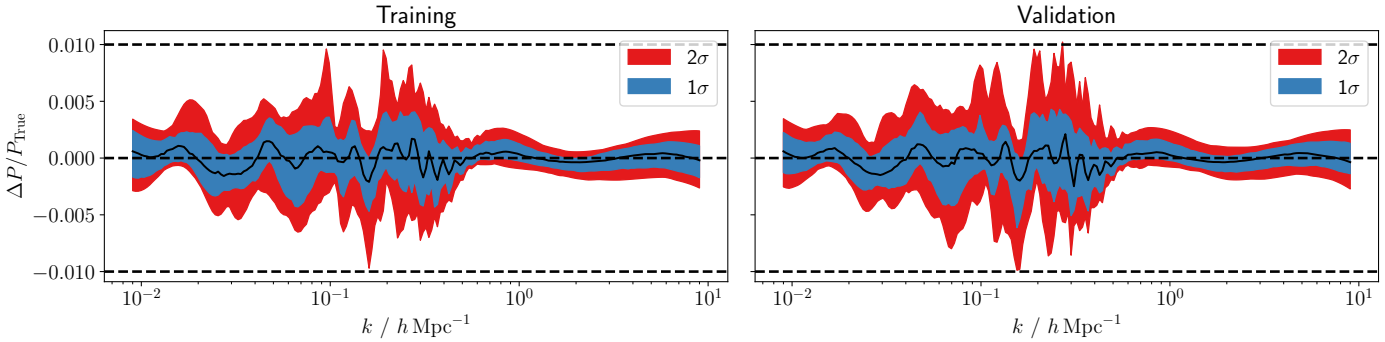

Fig. 4. Distribution of fractional errors as a function of k on the linear matter power spectrum across all cosmologies in the training and validation sets, as compared to the predictions of CAMB. The bands give the 1 and 2σ values. The dotted line corresponds to a 1% error, and we see that our expression achieves this for all cosmologies and values of k considered, with a root mean squared fractional error of 0.2%.

almost imperceptible, and in the residuals plot we see that for all k considered, the fractional error does not exceed 0.3%. This is smaller than the error on $\log F$ given in Fig. 3, since we compare at the level of the full $P(k;\boldsymbol{\theta})$, such that a moderate error on $\log F$ becomes very small once substituted into Eq. (5). This is shown in Fig. 4, where we plot the distribution of fractional residuals in $P(k;\boldsymbol{\theta})$ for all the cosmologies in the training and validations sets. We obtain sub-percent level predictions for all cosmologies and values of k considered, with a root mean squared fractional error of 0.2%.

Part of the appeal of a symbolic emulator is the possibility for interpretability and to easily identify what information used in the input is used to make the prediction. To begin, we note that, although we obtained our emulator by varying σ_8, Ω_b, Ω_m, h and n_s, we see that Eq. (6) contains neither σ_8 nor n_s. For the linear matter power spectrum, one expects that A_s and n_s only appear as a multiplicative factor of $A_\mathrm{s}k^{n_\mathrm{s}-1}$, with all other terms independent of these parameters. Given that the Eisenstein & Hu (1998) term already contains this expression, it is unsurprising that $\log F$ is independent of n_s. Indeed, if it did appear, this would indicate a degree of overfitting. Since σ_8 is not proportional to A_s (see Eq. (4)), we cannot use the same argument to explain the lack of its appearance in our expression, but can conclude that a combination of the Eisenstein & Hu (1998) term and the first line of Eq. (6) can sufficiently approximate A_s, since this line is k independent and thus contributes to an overall offset for the emulator.

Turning to the remaining lines of Eq. (6), we observe that each term contains an oscillation modulated by a k- and cosmology-dependent damping. Despite there being four such terms across the remaining three lines, we find that we can split these into two pairs with the same structure of the oscillations. Firstly, we have cosines with an argument proportional to $1/\sqrt{1+bk^2}$, for some constant b. This functional form $(x/\sqrt{1+y^2})$ arises due to the inclusion of the analytic quotient operator, which also explains why the constant 1 appears multiple times in Eq. (6). These terms give oscillations which vary slowly as a function of k. In particular, as plotted in Fig. 5, the third line of Eq. (6) contains approximately one cycle of oscillation across the range of k considered, with a minimum during the BAO part of the power spectrum, and a maximum just afterwards. Beyond this point, this term fits the non-oscillatory, decaying part of the residual beyond $k \sim 1\,h\,\mathrm{Mpc}^{-1}$ (compare the middle panel of Fig. 2 to the third term plotted in Fig. 5).

The remaining oscillatory terms are of the form $\cos(\omega k + \phi)$. The phase, ϕ, of these oscillations is proportional to the total

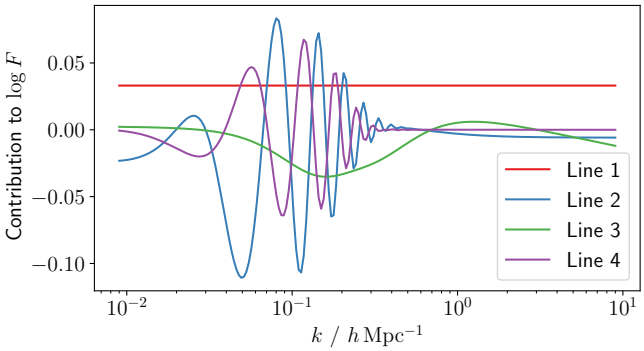

Fig. 5. Contributions to $\log F$ from our emulator as a function of k for the *Planck* 2018 cosmology. The line numbers indicated in the legend correspond to the line in Eq. (6). One sees that the first term provides an overall offset, the second and fourth capture the BAO signal, and the third term contains a broad oscillation and then matches on to the decaying residual at high k.

matter density, Ω_m, such that changing this parameter at fixed Ω_b shifts the BAOs to peak at different values of k. The frequency of these oscillations is $\omega \propto 1/\sqrt{b+\Omega_\mathrm{b}^2}$ for some parameter b, such that cosmologies with a higher fraction of baryons have many more cycles of BAO in a given range of k, as one would physically expect. From Fig. 5, one can see how the second and fourth lines of Eq. (6) capture the BAO signal with opposite signs, such that they combine to give the familiar damped oscillatory feature. Using $\Omega_\mathrm{b}h^2 = 0.02242$ and $h = 0.6766$, as appropriate for the *Planck* 2018 cosmology (Planck Collaboration VI 2020), the frequency of the oscillations are $b_{12}/\left(h\sqrt{b_{13}+\Omega_\mathrm{b}^2}\right) = 146.5\,\mathrm{Mpc}$ and $b_{35}/\left(h\sqrt{b_{36}+\Omega_\mathrm{b}^2}\right) = 145.8\,\mathrm{Mpc}$, and are thus approximately equal to the sound horizon, which is $r_* = 144.6\,\mathrm{Mpc}$ for this cosmology. One can therefore view these frequencies as symbolic approximations to the sound horizon, although we refer the reader to Aizpuru et al. (2021) for alternative SR fits.

Thus, although we did not enforce physically motivated terms in the equation search, we see that simple oscillatory contributions for the BAOs have emerged and thus our symbolic emulator is not merely a high order series expansion, but contains terms which are both compact and interpretable. We find that such terms exist in many functions given in Fig. 3, however we find that using shorter run times for OPERON of only 2–4 h

(compared to approximately 24 h on a single node of 128 cores for our fiducial analysis) do not provide as interpretable expressions as Eq. (6).

As a note of caution, one can identify a few terms in Eq. (6) which will become problematic if extrapolated to values of k much smaller than those used to train the emulator, namely those containing $\log(k)$ and k raised to a power proportional to k. For $k \lesssim 10^{-3}\,h\,\mathrm{Mpc}^{-1}$ this can lead to an error on $P(k)$ of more than one percent. Although one is likely cosmic-variance dominated in this regime so such errors should not be problematic, we know that the Eisenstein & Hu (1998) provide a very good approximation, and thus we suggest that Eq. (6) is included in a piece-wise fit, such that it is only used approximately in the range of k which were used to obtain it. A similar effect is seen if one extrapolates to higher k than considered here. Although this is far beyond the validity of the linear approximation, we caution that applying any parameterisation of the non-linear power spectrum that depends on the linear one may suffer from potentially catastrophic extrapolation failures at high k if used beyond the k range considered here. Again, it is potentially advisable to just use the Eisenstein & Hu (1998) fit in this regime.

In light of the potential for significant efficiency improvements in cosmological analyses through the use of symbolic approximations, it is informative to compare the run times of our approach to the standard computation of the linear matter power spectrum. To do this, we evaluated the redshift-zero linear matter power spectrum 1000 times on an Intel Xeon E5-4650 CPU at the *Planck* 2018 cosmology (Planck Collaboration VI 2020) using CAMB (Lewis et al. 2000), the BACCO (Angulo et al. 2021) neural network emulator and using our formulae, where we considered both a PYTHON3 and FORTRAN90 implementation, demonstrating the ease at which one can change programming language when using symbolic emulators. We found that CAMB takes an average of 0.18 s to evaluate $P(k)$, which is significantly slower than the BACCO emulator, which requires just 6.9 ms. However, our approach is even faster, requiring just 850 µs when written in PYTHON3 and 190 µs in FORTRAN90. This is approximately 950 times faster than CAMB and 36 times faster than BACCO.

If the reader wishes to use a more accurate, yet less interpretable, emulator, we provide the most accurate equation found in Appendix A, which has a model length of 142, with 73 parameters and yields a root mean of squared fractional errors on $P(k)$ of 0.1% for both the training and validation sets.

6. Discussion and conclusion

In this paper we have found analytic approximations to σ_8 Eq. (4) and the linear matter power spectrum Eq. (6) as a function of cosmological parameters which are accurate to sub-percent levels. In the case of σ_8, the simple yet accurate expression we have identified can be easily inverted to obtain A_s as a function of σ_8 and the other cosmological parameters. Our approximation to $P(k)$ is built by fitting the residuals between the output of a Boltzmann solved (CAMB) and the physics-inspired approximation of Eisenstein & Hu (1998). As such, unlike neural network or Gaussian process based approaches, our expression explicitly captures many physical processes (and is thus interpretable) whilst still achieving sub-percent accuracy.

This work is the first step in a programme of work dedicated to obtaining analytic approximations to $P(k)$ which can be used in current and future cosmological analyses. In this paper we have focused on the linear $P(k)$, i.e., the power spectrum of the linearly evolved density fluctuations. Although this approach is valid on large scales, the real Universe is non-linear, such that non-linear corrections are required at $k \gtrsim 10^{-1}\,h$ Mpc^{-1} to accurately model the observed matter power spectrum across a wider range of scales. In Bartlett et al. (2024) we extend our framework to capture such non-linear physics and to include redshift dependence in our emulator. Finally, in our emulator we have considered a ΛCDM Universe with massless neutrinos. In the future we will add corrections to the expressions found in this work to incorporate the effects of massive neutrinos and include beyond ΛCDM effects, such as a $w_0 - w_a$ parametrisation of dark energy.

We have demonstrated that, despite the temptation to blindly apply black-box methods such as neural networks to approximate physically useful functions, even in ostensibly challenging situations such as the matter power spectrum, one can achieve the required precision with relatively simple analytic fits. Given the unknown lifetime of current codes upon which numerical ML approximations are built and the ease of copying a few mathematical functions into your favourite programming language, finding analytic expressions allows one to more easily future-proof such emulators and should therefore be encouraged wherever possible.

Acknowledgements. We thank Bartolomeo Fiorini for useful comments and suggestions. DJB is supported by the Simons Collaboration on "Learning the Universe". LK was supported by a Balzan Fellowship. HD is supported by a Royal Society University Research Fellowship (grant no. 211046). PGF acknowledges support from STFC and the Beecroft Trust. BDW acknowledges support from the Simons Foundation. DA acknowledges support from the Beecroft Trust, and from the Science and Technology Facilities Council through an Ernest Rutherford Fellowship, grant reference ST/P004474/1. MZ is supported by STFC. We made extensive use of computational resources at the University of Oxford Department of Physics, funded by the John Fell Oxford University Press Research Fund, and at the Institut d'Astrophysique de Paris. For the purposes of open access, the authors have applied a Creative Commons Attribution (CC BY) licence to any Author Accepted Manuscript version arising. The data underlying this article will be shared on reasonable request to the corresponding author. We provide PYTHON3 and FORTRAN90 implementations of Eqs. (4) to (6) and (A.1) at `https://github.com/DeaglanBartlett/symbolic_pofk`.

References

Aizpuru, A., Arjona, R., & Nesseris, S. 2021, Phys. Rev. D, 104, 043521
Alestas, G., Kazantzidis, L., & Nesseris, S. 2022, Phys. Rev. D, 106, 103519
Angulo, R. E., Zennaro, M., Contreras, S., et al. 2021, MNRAS, 507, 5869
Aricò, G., Angulo, R. E., & Zennaro, M. 2022, Open Res Europe, 1:152
Arnaldo, I., Krawiec, K., & O'Reilly, U. M. 2014, Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation, GECCO '14 (New York: Association for Computing Machinery), 879
Bardeen, J. M., Bond, J. R., Kaiser, N., & Szalay, A. S. 1986, ApJ, 304, 15
Bartlett, D. J., Desmond, H., & Ferreira, P. G. 2023a, IEEE Transactions on Evolutionary Computation, 1
Bartlett, D. J., Desmond, H., & Ferreira, P. G. 2023b, The Genetic and Evolutionary Computation Conference 2023
Bartlett, D. J., Wandelt, B. D., Zennaro, M., Ferreira, P. G., & Desmond, H. 2024, A&A, 686, A150
Biggio, L., Bendinelli, T., Neitz, A., Lucchi, A., & Parascandolo, G. 2021, in Proceedings of the 38th International Conference on Machine Learning, eds. M. Meila, & T. Zhang, Proc. Mach. Learn. Res., 139, 936
Blas, D., Lesgourgues, J., & Tram, T. 2011, J. Cosmol. Astropart. Phys., 2011, 034
Burlacu, B. 2023, Proceedings of the Companion Conference on Genetic and Evolutionary Computation, GECCO '23 Companion (New York: Association for Computing Machinery), 2412
Burlacu, B., Kronberger, G., & Kommenda, M. 2020, Proceedings of the 2020 Genetic and Evolutionary Computation Conference Companion, GECCO '20 (New York: Association for Computing Machinery), 1562
Cava, W. G. L., Orzechowski, P., Burlacu, B., et al. 2021, arXiv e-prints [arXiv:2107.14351]

Cranmer, M. 2020, https://doi.org/10.5281/zenodo.4041459
Cranmer, M. 2023, arXiv e-prints [arXiv:2305.01582]
Cranmer, M., Sanchez Gonzalez, A., Battaglia, P., et al. 2020, arXiv e-prints [arXiv:2006.11287]
David, E. 1989, Genetic Algorithms in Search, Optimization and Machine Learning (Addison-Wesley)
de Franca, F. O., & Aldeia, G. S. I. 2021, Evolu. Comput., 29, 367
de Franca, F. O., & Kronberger, G. 2023, in Proceedings of the Genetic and Evolutionary Computation Conference, GECCO '23 (New York: Association for Computing Machinery), 1064
Delgado, A. M., Wadekar, D., Hadzhiyska, B., et al. 2022, MNRAS, 515, 2733
Desmond, H., Bartlett, D. J., & Ferreira, P. G. 2023, MNRAS, 521, 1817
Diemer, B. 2018, ApJS, 239, 35
Eisenstein, D. J., & Hu, W. 1998, ApJ, 496, 605
Eisenstein, D. J., & Hu, W. 1999, ApJ, 511, 5
Euclid Collaboration (Knabenhans, M., et al.) 2021, MNRAS, 505, 2840
Fendt, W. A., & Wandelt, B. D. 2007a, ApJ, submitted [arXiv:0712.0194]
Fendt, W. A., & Wandelt, B. D. 2007b, ApJ, 654, 2
Hahn, O., List, F., & Porqueres, N. 2023, JCAP, submitted [arXiv:2311.03291]
Haupt, R., & Haupt, S. 2004, Practical Genetic Algorithms, 2nd edn. (Wyley)
Heitmann, K., Lawrence, E., Kwan, J., Habib, S., & Higdon, D. 2014, ApJ, 780, 111
Jin, Y., Fu, W., Kang, J., & Guo, J. 2019, arXiv e-prints [arXiv:1910.08892]
Kamerkar, A., Nesseris, S., & Pinol, L. 2023, Phys. Rev. D, 108, 043509
Kammerer, L., Kronberger, G., Burlacu, B., et al. 2021, arXiv e-prints [arXiv:2109.13895]
Koksbang, S. M. 2023a, Phys. Rev. D, 107, 103522
Koksbang, S. M. 2023b, Phys. Rev. D, 108, 043539
Koksbang, S. M. 2023c, Phys. Rev. Lett., 130, 201003
Kommenda, M., Burlacu, B., Kronberger, G., & Affenzeller, M. 2020, Genet. Program. Evol. Mach., 21, 471
La Cava, W., Helmuth, T., Spector, L., & Moore, J. H. 2019, Evolu. Comput., 27, 377
Landajuela, M., Lee, C. S., Yang, J., et al. 2022, A Unified Framework for Deep Symbolic Regression, 36th Conference on Neural Information Processing Systems
Laumanns, M., Thiele, L., Deb, K., & Zitzler, E. 2002, Evol. Comput., 10, 263
Lemos, P., Jeffrey, N., Cranmer, M., Ho, S., & Battaglia, P. 2023, Mach. Learn. Sci. Technol., 4, 045002
Levenberg, K. 1944, Quart. Appl. Math., 2, 164
Lewis, A., Challinor, A., & Lasenby, A. 2000, ApJ, 538, 473
Lodha, K., Pinol, L., Nesseris, S., et al. 2024, MNRAS, 530, 1424
Marquardt, D. W. 1963, J. Soc. Indust. Appl. Math., 11, 431
McConaghy, T. 2011, in FFX: Fast, Scalable, Deterministic Symbolic Regression Technology, eds. R. Riolo, E. Vladislavleva, & J. H. Moore (New York: Springer), 235
Miniati, F., & Gregori, G. 2022, Sci. Rep., 12, 11709
Mootoovaloo, A., Jaffe, A. H., Heavens, A. F., & Leclercq, F. 2022, Astron. Comput., 38, 100508
Orjuela-Quintana, J. B., Nesseris, S., & Cardona, W. 2023, Phys. Rev. D, 107, 083520
Orjuela-Quintana, J. B., Nesseris, S., & Sapone, D. 2024, Phys. Rev. D, 109, 063511
Petersen, B. K., Larma, M. L., Mundhenk, T. N., et al. 2021, International Conference on Learning Representations
Planck Collaboration VI. 2020, A&A, 641, A6
René Broløs, K., Vieira Machado, M., Cave, C., et al. 2021, arXiv e-prints [arXiv:2104.05417]
Rivero, D., Fernandez-Blanco, E., & Pazos, A. 2022, Exp. Syst. Appl., 198, 116712
Schmidt, M., & Lipson, H. 2009, Science, 324, 81
Schmidt, M., & Lipson, H. 2011, in Age-Fitness Pareto Optimization (New York: Springer), eds. R. Riolo, T. McConaghy, & E. Vladislavleva, 129
Smith, R. E., Peacock, J. A., Jenkins, A., et al. 2003, MNRAS, 341, 1311
Sousa, T., Bartlett, D. J., Desmond, H., & Ferreira, P. G. 2024, Phys. Rev. D, 109, 083524
Spurio Mancini, A., Piras, D., Alsing, J., Joachimi, B., & Hobson, M. P. 2022, MNRAS, 511, 1771
Tenachi, W., Ibata, R., & Diakogiannis, F. I. 2023, ApJ, 959, 99
Turing, A.M. 1950, Mind, LIX, 433
Udrescu, S. M., & Tegmark, M. 2020, Sci. Adv., 6, eaay2631
Udrescu, S. M., Tan, A., Feng, J., et al. 2020, AI Feynman 2.0: Pareto-optimal Symbolic Regression Exploiting Graph Modularity, 34th Conference on Neural Information Processing Systems
Virgolin, M., Alderliesten, T., Witteveen, C., & Bosman, P. A. N. 2021, Evolu. Comput., 29, 211
Wadekar, D., Thiele, L., Hill, J. C., et al. 2023, MNRAS, 522, 2628
Winther, H. A., Casas, S., Baldi, M., et al. 2019, Phys. Rev. D, 100, 123540
Worm, T., & Chiu, K. 2013, Proceedings of the 15th Annual Conference on Genetic and Evolutionary Computation, GECCO '13 (New York: Association for Computing Machinery), 1021
Zennaro, M., Angulo, R. E., Pellejero-Ibáñez, M., et al. 2023, MNRAS, 524, 2407

Appendix A: Most accurate analytic expression found for linear power spectrum

The expression we report for an analytic approximation for the linear matter power spectrum Eq. (6) is not the most accurate one found, but the one which we deemed to appropriately balance accuracy, simplicity, and interpretability. It may be desirable to have a more accurate symbolic expression if interpretability is not a concern. In this case one may wish to use the most accurate equation found, which is

$$
\begin{aligned}
100 \log F \approx\ & c_0 k + c_1 \left(\Omega_\mathrm{b} c_2 - \frac{c_3 k}{\sqrt{c_4 + k^2}}\right)\left(\frac{c_{34}\,(c_{35}k)^{-c_{36}k}}{\sqrt{c_{39} + (-\Omega_\mathrm{b} + \Omega_\mathrm{m} c_{37} - c_{38} h)^2}} - \cos\left(\Omega_\mathrm{m} c_{32} - c_{33} k\right)\right) \\
& \times \left(\frac{c_{17}\,(c_{25}k)^{-c_{26}k}\left((\Omega_\mathrm{b} c_{18} + \Omega_\mathrm{m} c_{19} - c_{20} h)\cos\left(\Omega_\mathrm{m} c_{21} - c_{22} k\right) + \cos\left(c_{23} k - c_{24}\right)\right)}{\sqrt{c_{31} + \left(\frac{c_{27}\left(-\Omega_\mathrm{m} c_{28} + c_{29} k\right)}{\sqrt{c_{30} + k^2}} - k\right)^2}}\right. \\
& \left. - \frac{c_5\,(\Omega_\mathrm{m} c_{12} + c_{13} k)^{-c_{14}k}\left(\Omega_\mathrm{m} c_6 - c_7 k + (\Omega_\mathrm{b} c_8 - c_9 k)\cos\left(\Omega_\mathrm{m} c_{10} - c_{11} k\right)\right)}{\sqrt{c_{16} + (\Omega_\mathrm{b} c_{15} + k)^2}}\right) \\
& - c_{40}\left(\Omega_\mathrm{m} c_{41} - c_{42} h + c_{43} k + \frac{c_{44} k}{\sqrt{c_{45} + k^2}\sqrt{c_{47} + (-\Omega_\mathrm{m} - c_{46} h)^2}} - \frac{c_{48}\left(\Omega_\mathrm{m} c_{49} + c_{50} k\right)}{\sqrt{c_{51} + k^2}}\right)\cos\left(\frac{c_{52} k}{\sqrt{c_{53} + k^2}\sqrt{c_{55} + (\Omega_\mathrm{m} c_{54} - k)^2}}\right) \\
& - c_{56} - \frac{c_{57}\,(\Omega_\mathrm{m} c_{67} + c_{68} k)^{-c_{69}k}\left(\Omega_\mathrm{m} c_{58} - c_{59} k + (-\Omega_\mathrm{b} c_{60} - \Omega_\mathrm{m} c_{61} + c_{62} h)\cos\left(\Omega_\mathrm{m} c_{63} - c_{64} k\right) + \cos\left(c_{65} k - c_{66}\right)\right)}{\sqrt{\frac{c_{70}\left(\Omega_\mathrm{b} + \frac{c_{71} h}{\left(c_{72} + k^2\right)^{0.5}}\right)^2}{c_{73} + k^2}} + 1.0}.
\end{aligned}
\tag{A.1}
$$

This equation has 73 parameters, which is approximately twice as many as Eq. (6), yet one only gains a factor of two in the fractional root mean squared error. The best-fit parameter values are reported in Table A.1. We note that this function is the direct output of OPERON and is thus over-parameterised so that some simplification could be applied. For example, one only needs two of c_1, c_2 and c_3 as these only appear as $c_1 c_2$ and $c_1 c_3$. Since we only provide this expression as a precise emulator and do not attempt to interpret its terms, we choose not to apply any simplifications (although see de Franca & Kronberger (2023) for an automated method to do this).

Table A.1. Best-fit parameters for the most accurate linear matter power spectrum emulator found, reported in Appendix A.

Parameter	Value	Parameter	Value	Parameter	Value	Parameter	Value
c_0	5.1439	c_{19}	19.855	c_{38}	0.0177	c_{57}	0.867
c_1	0.867	c_{20}	15.939	c_{39}	0.0146	c_{58}	2.618
c_2	8.52	c_{21}	9.547	c_{40}	0.867	c_{59}	2.1
c_3	0.2920	c_{22}	97.34	c_{41}	32.371	c_{60}	114.391
c_4	0.0310	c_{23}	94.83	c_{42}	7.058	c_{61}	13.968
c_5	0.0033	c_{24}	1.881	c_{43}	6.075	c_{62}	11.133
c_6	240.234	c_{25}	3.945	c_{44}	16.311	c_{63}	4.205
c_7	682.449	c_{26}	11.151	c_{45}	0.0025	c_{64}	100.376
c_8	2061.023	c_{27}	0.0004	c_{46}	0.1632	c_{65}	106.993
c_9	6769.493	c_{28}	26.822	c_{47}	0.0771	c_{66}	3.359
c_{10}	7.125	c_{29}	230.12	c_{48}	0.0522	c_{67}	1.539
c_{11}	108.136	c_{30}	0.0009	c_{49}	22.722	c_{68}	1.773
c_{12}	6.2	c_{31}	1.0796×10^{-5}	c_{50}	774.688	c_{69}	18.983
c_{13}	2.882	c_{32}	3.162	c_{51}	0.0027	c_{70}	0.3838
c_{14}	59.585	c_{33}	99.918	c_{52}	1.0337	c_{71}	0.0024
c_{15}	0.1384	c_{34}	0.1210	c_{53}	0.0058	c_{72}	1.2865×10^{-7}
c_{16}	1.0825×10^{-5}	c_{35}	0.495	c_{54}	0.2472	c_{73}	2.0482×10^{-8}
c_{17}	0.0033	c_{36}	12.091	c_{55}	0.29		
c_{18}	85.791	c_{37}	0.6070	c_{56}	0.241		

Notes. Although units are excluded in this table, the units for each parameter are easily obtained by noting that these are defined assuming that k is measured in $h\,\mathrm{Mpc}^{-1}$.