

This is the accepted manuscript made available via CHORUS. The article has been published as:

## Looking for non-Gaussianity in all the right places: A new basis for nonseparable bispectra

Joyce Byun, Nishant Agarwal, Rachel Bean, and Richard Holman

Phys. Rev. D **91**, 123518 — Published 12 June 2015

DOI: [10.1103/PhysRevD.91.123518](https://doi.org/10.1103/PhysRevD.91.123518)

# Looking for non-Gaussianity in all the right places: A new basis for non-separable bispectra

Joyce Byun,<sup>1,\*</sup> Nishant Agarwal,<sup>2,3,†</sup> Rachel Bean,<sup>1,‡</sup> and Richard Holman<sup>4,§</sup>

<sup>1</sup>*Department of Astronomy, Cornell University, Ithaca, NY 14853, USA*

<sup>2</sup>*Department of Astronomy and Astrophysics, The Pennsylvania State University, University Park, PA 16802, USA*

<sup>3</sup>*Institute for Gravitation and the Cosmos, The Pennsylvania State University, University Park, PA 16802, USA*

<sup>4</sup>*Department of Physics, Carnegie Mellon University, Pittsburgh, PA 15213, USA*

Non-Gaussianity in the distribution of inflationary perturbations, measurable in statistics of the cosmic microwave background (CMB) and large scale structure fluctuations, can be used to probe non-trivial initial quantum states for these perturbations. The bispectrum shapes predicted for generic non-Bunch-Davies initial states are non-factorizable (“non-separable”) and are highly oscillatory functions of the three constituent wavenumbers. This can make the computation of CMB bispectra, in particular, computationally intractable. To efficiently compare with CMB data one needs to construct a separable template that has a significant similarity with the actual shape in momentum space. In this paper we consider a variety of inflationary scenarios, with different non-standard initial conditions, and how best to construct viable template matches. In addition to implementing commonly used separable polynomial and Fourier bases, we introduce a basis of localized piecewise spline functions. The spline basis is naturally nearly orthogonal, making it easy to implement and to extend to many modes. We show that, in comparison to existing techniques, the spline basis can provide better fits to the true bispectrum, as measured by the cosine between shapes, for sectors of the theory space of general initial states. As such, it offers a useful approach to investigate non-trivial features generated by fundamental properties of the inflationary Universe.

## I. INTRODUCTION

We are fortunate to live in a time when cosmological datasets can probe the Universe in exquisite detail. In particular, the cosmic microwave background (CMB) provides a rich source of information about the very early Universe, and is an especially precise probe of the inflationary paradigm. An important question that we are now in a position to probe, more thoroughly than ever, is: what is the initial quantum state of inflationary fluctuations? It is usually taken to be the Bunch-Davies state, but from the point of view of treating inflation as an effective theory it is not unreasonable to consider the choice of state to be open, subject to the conditions that it allow for inflation to occur and that it be consistent with field theoretic precepts. Explicit examples of scenarios that give rise to non-Bunch-Davies initial conditions for inflation can be found in [1–6]. Assuming that the initial state is more general than the free vacuum, such as a Bogoliubov transform of the Bunch-Davies state or even a mixed state, we can calculate its imprint on cosmological observables like the power spectrum and bispectrum of inflationary perturbations, and in turn those of CMB temperature anisotropies [2, 3, 7–22]. Whether these effects can actually be observed in cosmological data depends on the extent of departure from a Bunch-Davies state, the number of e-folds of in-

flation beyond the minimum required, and of course the sensitivity of our experiments [23–25].

Given these choices, how can we narrow down the possibilities? The initial state of perturbations will leave its imprints on their correlation functions, for example, logarithmic oscillatory modulations in the CMB power spectrum [26]. Higher order correlators, such as the bispectrum [27, 28], are extremely sensitive to deviations from the Bunch-Davies state. The bispectrum carries information both about the amplitude of the correlations, typically encoded in  $f_{\text{NL}}$ , as well as of preferred configurations in momentum space; the three momenta must form a triangle, but the shape of the triangle is sensitive to both the interactions of the inflaton, including the mixing with gravity, as well as the initial state.

The bispectrum for general initial states is highly oscillatory and cannot be written in a separable form, i.e. as a product of separate functions of the three momentum modes. This makes the study of these states via the bispectrum computationally difficult. For such shapes we usually construct a basis of separable functions and rewrite the desired shape as a sum over many such basis functions, for example, using a polynomial basis [29], Fourier basis [30], or divergent basis [31].<sup>1</sup> As long as the original and reconstructed shapes are very similar and have a significant overlap, or in other words have a cosine close to unity, we can look for signatures of the

---

\*Electronic address: [byun@astro.cornell.edu](mailto:byun@astro.cornell.edu)

†Electronic address: [nua11@psu.edu](mailto:nua11@psu.edu)

‡Electronic address: [rbean@astro.cornell.edu](mailto:rbean@astro.cornell.edu)

§Electronic address: [rh4a@andrew.cmu.edu](mailto:rh4a@andrew.cmu.edu)

---

<sup>1</sup> In special shape-specific cases, other basis sets can be used: for example, feature or resonant models exhibiting linear or logarithmic oscillations, respectively, can be efficiently reconstructed through a one-dimensional (1D) expansion in the sum of wavenumbers,  $k_1 + k_2 + k_3$  [32, 33].

reconstructed shape in the CMB bispectrum and be assured that the result will be an accurate reflection of what we would have obtained for the real shape. Non-Bunch-Davies shapes, however, can be difficult to efficiently describe with existing bases [28–31]. This leaves open the possibility that signatures of general initial states could still be present, yet undetected, in the CMB data.

In this paper we describe a new basis of piecewise spline functions and use it to fit non-Bunch-Davies shapes of the bispectrum. The *spline* basis consists of polynomial functions defined locally, between various points called “knots” in 3D space [34, 35]. This makes it particularly suitable for smoothing and interpolating data with complex patterns, and in our case, for capturing localized features of any complicated shape of the bispectrum. Another immediate advantage of defining localized functions is that the basis functions are orthogonal to a good approximation, and there is no need to perform a Gram-Schmidt-like orthogonalization process on the basis functions. Instead, the basis-fitting algorithm we adopt does not rely on having an orthonormal basis. This makes the spline basis easy to implement and to extend, using a large number of mode functions to capture fine features in the bispectrum. We find that the spline basis performs at least as well as the polynomial and Fourier bases in describing most non-Bunch-Davies shapes, and for many shapes offers significant improvements (in the cosines) over existing techniques.

The remaining paper is organized as follows. In section II we briefly review how we define the initial conditions for the perturbations and obtain the bispectrum for general initial states. We describe the spline basis in section III and use it to fit non-Bunch-Davies shapes in section IV. We conclude in section V with a summary and discussion on the scope of this work. Appendices A and B contain details on calculations of the correlation functions for general initial states, and appendix C describes our numerical implementation of the spline basis.

## II. NON-BUNCH-DAVIES SHAPES

We usually describe primordial correlation functions in terms of the curvature perturbation  $\zeta(t, \vec{x})$ , since this quantity is conserved outside the horizon [36]. It is defined as the perturbation in the local scale factor  $a(t)$ ; the metric perturbation  $h_{ij}(t, \vec{x})$  is then written as  $h_{ij} = a^2 e^{2\zeta} \delta_{ij}$ . In an effective field theory setting,  $\zeta(t, \vec{x})$  is related to the Goldstone mode of time reparameterization symmetry breaking, usually denoted as  $\pi(t, \vec{x})$  [37, 38]. In this section we describe how the choice of initial state for the perturbation  $\zeta(t, \vec{x})$  (or equivalently  $\pi(t, \vec{x})$ ) affects the primordial bispectrum  $B_\zeta(k_1, k_2, k_3)$  defined via  $\langle \zeta_{\vec{k}_1} \zeta_{\vec{k}_2} \zeta_{\vec{k}_3} \rangle = (2\pi)^3 \delta^3(\vec{k}_1 + \vec{k}_2 + \vec{k}_3) B_\zeta(k_1, k_2, k_3)$ .

Starting with the Einstein-Hilbert action, we can calculate the action for scalar perturbations directly in the  $\zeta$ -gauge. Writing down the most general Lorentz-invariant scalar-tensor theory with second order equa-

tions of motion results in the Horndeski action for the perturbations [39–43].<sup>2</sup> For  $P(X, \phi)$  models of inflation, with  $X \equiv -g_{\mu\nu} \partial^\mu \phi \partial^\nu \phi$ ,<sup>3</sup> the leading order in slow-roll Horndeski action at cubic order in the perturbations is given by

$$S = \int d^3x dt a^3 \left\{ \frac{\varepsilon}{c_s^2} \left[ \dot{\zeta}^2 - \frac{c_s^2}{a^2} (\partial_i \zeta)^2 \right] + \Lambda_1 \dot{\zeta}^3 + \Lambda_2 \zeta \dot{\zeta}^2 + \frac{\Lambda_3}{a^2} \zeta (\partial_i \zeta)^2 \right\} + S_{\text{boundary}}. \quad (1)$$

Here we have set  $M_{\text{Pl}} = 1$ ,  $\varepsilon \equiv -\dot{H}/H^2$  is the slow-roll parameter,  $c_s$  is the sound speed for perturbations, and the couplings  $\Lambda_1 - \Lambda_3$  are given by

$$\Lambda_1 = \frac{\varepsilon}{H c_s^4} \left( 1 - c_s^2 - 2 \frac{\lambda c_s^2}{\Sigma} \right), \quad (2)$$

$$\Lambda_2 = -3 \frac{\varepsilon}{c_s^4} (1 - c_s^2), \quad (3)$$

$$\Lambda_3 = \frac{\varepsilon}{c_s^2} (1 - c_s^2), \quad (4)$$

with  $\lambda \equiv X^2 \frac{\partial^2 P}{\partial X^2} + \frac{2}{3} X^3 \frac{\partial^3 P}{\partial X^3}$  and  $\Sigma \equiv X \frac{\partial P}{\partial X} + 2 X^2 \frac{\partial^2 P}{\partial X^2} = \frac{H^2 \varepsilon}{c_s^2}$ . The boundary terms in eq. (1) are important and in general do contribute to the bispectrum when the initial state is different from the Bunch-Davies vacuum; here we assume for simplicity that the initial state for  $\zeta(t, \vec{x})$  is defined in such a way as to cancel these boundary terms. It is also worth noting that the action in eq. (1) is equivalent to that obtained using an effective field theory approach [37, 38] up to boundary terms, and the bispectrum for general initial states agrees between the two actions, as shown in [44].

The usual method to obtain correlation functions of  $\zeta(t, \vec{x})$  is to use in-in perturbation theory [45–52]. The initial state can be input as a density matrix defined at the initial time  $t_0$  [20]; for simplicity we will assume that the initial density matrix is pure and Gaussian.<sup>4</sup> Doubling the fields on the plus and minus branches of the in-in contour, we can calculate the Green’s function corresponding to the quadratic (or “free”) part of the action in eq. (1), including the effect of a general initial state;

<sup>2</sup> The Horndeski action does not describe ghost inflation, however, which can be included in an effective field theory setting. Another example outside the Horndeski domain is Horava-Lifshitz gravity, in which Lorentz invariance is explicitly broken.

<sup>3</sup> The action for many single scalar field models of inflation can be written as  $S = \frac{1}{2} \int d^4x \sqrt{-g} [R + 2P(X, \phi)]$ , with  $\phi$  controlling the dynamics of both the background and perturbations.

<sup>4</sup> Here by “pure” we are distinguishing between pure and mixed quantum states. Mathematically speaking,  $\text{Tr}(\rho^2) = 1$  for pure states, while  $\text{Tr}(\rho^2) < 1$  for mixed states,  $\rho$  being the density matrix. Relaxing the pure state assumption leads to qualitatively very similar results to what we discuss here. By “Gaussian” we mean that the action describing the initial state is quadratic.

details of this calculation can be found in the appendix of [20]. The Green's function for  $\zeta(t, \vec{x})$  is found to be

$$\mathcal{G}_k^\zeta(\eta, \eta') = \frac{c_s^2}{2\varepsilon} \frac{1}{a(\eta)a(\eta')} \begin{pmatrix} G_k^{++}(\eta, \eta') & G_k^{+-}(\eta, \eta') \\ G_k^{-+}(\eta, \eta') & G_k^{--}(\eta, \eta') \end{pmatrix}, \quad (5)$$

where  $\eta$  is the conformal time defined as  $\eta = \int dt/a$  and the factor out front comes from rewriting the quadratic action in terms of the canonically rescaled field  $\chi(t, \vec{x})$ ,  $\zeta = \frac{1}{\sqrt{2\varepsilon}} \frac{c_s}{a} \chi$ . The functions  $G_k^{\pm, \pm}(\eta, \eta')$  are given by

$$G_k^{++}(\eta, \eta') = f_k^>(\eta) f_k^<(\eta') \theta(\eta - \eta') + f_k^<(\eta) f_k^>(\eta') \theta(\eta' - \eta), \quad (6)$$

$$G_k^{+-}(\eta, \eta') = f_k^<(\eta) f_k^>(\eta'), \quad (7)$$

$$G_k^{-+}(\eta, \eta') = G_k^{+-*}(\eta, \eta'), \quad (8)$$

$$G_k^{--}(\eta, \eta') = G_k^{++*}(\eta, \eta'), \quad (9)$$

and the mode functions  $f_k^{\gtrless}(\eta)$  are solutions to the second order differential equation resulting from the Green's function equation,

$$f_k^{\gtrless}(\eta) = \alpha_k^{\gtrless} h_k(\eta) + \beta_k^{\gtrless} h_k^*(\eta) - 2f_k^{\gtrless}(\eta_0) A_k g_k(\eta) \theta(\eta_0 - \eta). \quad (10)$$

The last term above is an additional contribution from the initial density matrix,  $A_k$  being the kernel that multiplies the  $\zeta_k^+ \zeta_{-k}^+$  term in the initial state action [20]; it does not, however, contribute to the bispectrum, and so we will ignore it in our discussion. The Bogoliubov coefficients  $\alpha_k^{\gtrless}$ ,  $\beta_k^{\gtrless}$  are functions of kernels in the initial density matrix, the mode functions, and time derivatives of the mode functions, all at the initial time, and satisfy  $\alpha_k^< = \beta_k^{>*}$ ,  $\beta_k^< = \alpha_k^{>*}$ ,  $|\alpha_k^>|^2 - |\beta_k^>|^2 = 1$ . Finally, the function  $h_k(\eta)$  is defined at leading order in slow-roll as

$$h_k(\eta) = -\frac{1}{2} (-\pi\eta)^{1/2} H_{3/2}^{(1)}(-c_s k \eta), \quad (11)$$

where  $H_{3/2}^{(1)}$  is a Hankel function. The Bunch-Davies choice consists of setting the initial time  $\eta_0 \rightarrow -\infty$ , the initial density matrix to unity, and additionally  $\alpha_k^> = 1$ ,  $\beta_k^> = 0$  so that the mode function  $f_k^>(\eta)$  picks out the positive frequency solution proportional to  $e^{-ic_s k \eta}$ .

The time  $\eta_0$  at which the initial conditions are set can be taken to be a constant time in the past at the onset of inflation, or can be considered as a scale-dependent quantity,  $\eta_0(k)$ . In the latter case, the initial conditions for each  $k$  mode are set at the time when the physical momentum corresponding to this mode  $c_s k/a(\eta_0)$  (with  $a(\eta_0) = -1/(\eta_0 H)$  during inflation, at leading order) crosses a fixed energy scale  $\Lambda$  of new physics. The Bogoliubov transform in eq. (10) is correct for either choice of  $\eta_0$ . In appendix A we show that for a scale-dependent initial time, this solution leads to the well-known oscillations in the late-time power spectrum [10, 11]. We use

both choices of initial time in section IV when we apply the spline basis to non-Bunch-Davies shapes of the bispectrum.

Let us now discuss how general initial states modify the bispectrum. Observables, such as the bispectrum, can be calculated using any combination of plus and minus fields on the in-in contour. For Gaussian initial states, we can calculate the three-point function in the perturbations as

$$\begin{aligned} \langle \zeta_{k_1}^+ \zeta_{k_2}^+ \zeta_{k_3}^+ \rangle(\eta) &= \langle \zeta_{k_1}^+(\eta) \zeta_{k_2}^+(\eta) \zeta_{k_3}^+(\eta) \\ &\times \exp \left[ i \left( S^{(3)+} - S^{(3)-} \right) \right] \rangle_G, \end{aligned} \quad (12)$$

where  $S^{(3)}$  is the cubic part of the action in eq. (1) written in momentum space, with conformal time derivatives, and with the time integral running from  $\eta_0$  to  $\eta$ . The subscript ‘‘G’’ indicates that Wick contractions on the right are carried out using the Gaussian theory. At leading order in slow-roll, only the three operators  $\zeta^3$ ,  $\zeta \dot{\zeta}^2$ , and  $\zeta (\partial_i \zeta)^2$  contribute, and we can write the three-point function at late times  $\langle \zeta_{k_1}^+ \zeta_{k_2}^+ \zeta_{k_3}^+ \rangle(\eta)|_{\eta \rightarrow 0^-}$  as a sum of contributions from each of these three operators. To calculate the three-point function in eq. (12) at late times we need the function  $G_k^{\zeta, ++}(0, \eta')$  and its derivative  $\partial_{\eta'} G_k^{\zeta, ++}(0, \eta')$  for  $\eta' \geq \eta_0$ ; using eqs. (6), (10) (discarding the  $\theta(\eta_0 - \eta)$  term), and (11) these are given by

$$G_k^{\zeta, ++}(0, \eta') = \frac{H^2}{4\varepsilon c_s k^3} \left[ a_k (1 - ic_s k \eta') e^{ic_s k \eta'} + b_k (1 + ic_s k \eta') e^{-ic_s k \eta'} \right] \quad (13)$$

and

$$\partial_{\eta'} G_k^{\zeta, ++}(0, \eta') = \frac{H^2 c_s \eta'}{4\varepsilon k} \left( a_k e^{ic_s k \eta'} + b_k e^{-ic_s k \eta'} \right), \quad (14)$$

where we have defined the functions

$$a_k = (\alpha_k^> - \beta_k^>) \alpha_k^{>*}, \quad (15)$$

$$b_k = -(\alpha_k^> - \beta_k^>) \beta_k^{>*}. \quad (16)$$

Using these in eq. (12) and performing the time integrals we find that the contributions to the bispectrum from the three operators are given by

$$\begin{aligned} \langle \zeta_{k_1}^+ \zeta_{k_2}^+ \zeta_{k_3}^+ \rangle_{\zeta^3}(\eta)|_{\eta \rightarrow 0^-} &= -\frac{3}{32} (2\pi)^3 \delta^3 \left( \sum \vec{k}_i \right) \\ &\times \Lambda_1 \frac{H^5}{\varepsilon^3} \left[ \sum_{l, m, n=0}^1 c_{k_1}^{(l)} c_{k_2}^{(m)} c_{k_3}^{(n)} \right. \\ &\times \mathcal{F}_{\zeta^3} \left( (-1)^l k_1, (-1)^m k_2, (-1)^n k_3, \eta_0 \right) \left. \right] \\ &+ \text{c.c.}, \end{aligned} \quad (17)$$

$$\begin{aligned}
& \left\langle \zeta_{k_1}^+ \zeta_{k_2}^+ \zeta_{k_3}^+ \right\rangle_{\zeta \dot{\zeta}^2} (\eta) \Big|_{\eta \rightarrow 0^-} = \frac{1}{32} (2\pi)^3 \delta^3 \left( \sum \vec{k}_i \right) \\
& \times \Lambda_2 \frac{H^4}{\varepsilon^3} \left[ \sum_{l,m,n=0}^1 c_{k_1}^{(l)} c_{k_2}^{(m)} c_{k_3}^{(n)} \right. \\
& \times \mathcal{F}_{\zeta \dot{\zeta}^2} \left( (-1)^l k_1, (-1)^m k_2, (-1)^n k_3, \eta_0 \right) \Big] \\
& + \text{c.c.}, \tag{18}
\end{aligned}$$

and

$$\begin{aligned}
& \left\langle \zeta_{k_1}^+ \zeta_{k_2}^+ \zeta_{k_3}^+ \right\rangle_{\zeta (\partial \zeta)^2} (\eta) \Big|_{\eta \rightarrow 0^-} = \frac{1}{64} (2\pi)^3 \delta^3 \left( \sum \vec{k}_i \right) \\
& \times \Lambda_3 \frac{H^4}{c_s^2 \varepsilon^3} \left[ \sum_{l,m,n=0}^1 c_{k_1}^{(l)} c_{k_2}^{(m)} c_{k_3}^{(n)} \right. \\
& \times \mathcal{F}_{\zeta (\partial \zeta)^2} \left( (-1)^l k_1, (-1)^m k_2, (-1)^n k_3, \eta_0 \right) \Big] \\
& + \text{c.c.}, \tag{19}
\end{aligned}$$

where “c.c.” denotes complex conjugate and

$$c_k^{(i)} = \begin{cases} a_k & i = 0 \\ b_k & i = 1. \end{cases} \tag{20}$$

The functions  $\mathcal{F}_{\dot{\zeta}^3}$ ,  $\mathcal{F}_{\zeta \dot{\zeta}^2}$ , and  $\mathcal{F}_{\zeta (\partial \zeta)^2}$  are written out explicitly in appendix B. The above equations give us the leading order result for the bispectrum for general initial states. The non-Bunch-Davies contributions to the bispectrum are strongly peaked in the flattened limit  $k_1 \approx k_2 + k_3$  (assuming that  $k_1$  is the largest momentum mode) and in the squeezed limit  $k_3 \ll k_1 \approx k_2$ . (We show these enhancements and discuss apparent divergences in both limits in appendix B.) Further, these shapes are highly oscillatory, which makes them even harder to constrain using the CMB bispectrum. In the next section we discuss the spline basis that we use to rewrite these shapes as a sum of separable functions.

### III. THE SPLINE BASIS

B-splines, short for “basis splines”, are a well-established, and conceptually simple, mathematical formalism for curve fitting, using a set of piecewise polynomial functions [34, 35]. For a basis in one dimension, one chooses a set of “knots”,  $\{x_0, x_1, \dots, x_N\}$ , representing the points at which the polynomial function pieces are joined, and the degree  $q$  of the polynomials. For example, a 1D spline basis spanning  $0 \leq x \leq 1$  with a set of six piecewise cubic polynomials is shown in fig. 1. As an explicit example of the functional form of the basis, one of the basis functions shown in the figure is

$$B_1(x) = \begin{cases} \frac{9}{4} (4x - 18x^2 + 21x^3) & 0 \leq x < \frac{1}{3} \\ \frac{1}{4} (8 - 36x + 54x^2 - 27x^3) & \frac{1}{3} \leq x \leq \frac{2}{3}. \end{cases} \tag{21}$$

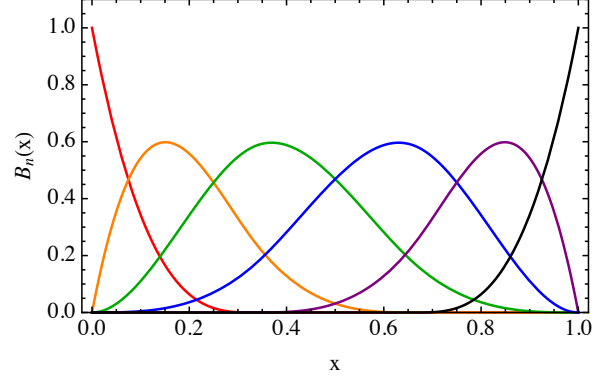


FIG. 1: Example spline basis generated from knots at  $x = \{0, 0, 0, 0, 1/3, 2/3, 1, 1, 1, 1\}$  and polynomials of degree  $q = 3$ .

The numerical coefficients defining the spline basis for input knots and  $q$  are easily generated using existing software libraries in many languages.

A general 1D function,  $f(x)$ , can be expanded, and approximated, using the spline basis,

$$f'(x) = \sum_{n=0}^{N-1} \alpha_n B_n(x). \tag{22}$$

The expansion coefficients are computed by first tabulating a sample of  $(x_i, f_i)$  values, where  $f_i \equiv f(x_i)$ , corresponding to  $M$  data points, and finding the values of  $\{\alpha_n\}$  that minimize the least-squares function,

$$\text{LS} = \sum_{i=1}^M \left( f_i - \sum_{n=0}^{N-1} \alpha_n B_n(x_i) \right)^2. \tag{23}$$

In practice, this requires solving the linear system of equations given by  $B^T B \vec{\alpha} = B^T \vec{f}$ , where  $B$  here is an  $M \times N$  matrix containing the values of  $B_n(x_i)$ .

Similarly, we can generate a higher-dimensional basis by multiplying 1D b-splines together. For example, a 3D b-spline fit for a shape<sup>5</sup> can be expressed as

$$\begin{aligned}
S'(k_1, k_2, k_3) = & \sum_{k=0}^{N-1} \sum_{j=0}^k \sum_{i=0}^j A_{ijk} [B_i(k_1) B_j(k_2) B_k(k_3) \\
& + \text{perms}], \tag{24}
\end{aligned}$$

where “perms” refers to permutations of  $k_1, k_2$ , and  $k_3$ . We note that the number of relevant  $(i, j, k)$  combinations in such a fit is not  $N^3$ , because each spline mode is symmetric in the three wavenumbers, and not all  $(i, j, k)$  combinations will correspond to a spline mode that is

<sup>5</sup> A shape function is usually denoted by  $S(k_1, k_2, k_3)$ , not to be confused with the action  $S$ .

non-zero for wavenumber combinations that can form a closed triangle. In the 2D and 3D cases, efficient algorithms solving for the expansion coefficients  $A_{ijk}$  already exist in the literature [53]. We illustrate how to use the spline basis further through explicit examples and include snippets of our numerical codes in appendix C.

In contrast to other bases used to create non-Gaussian templates made of globally varying functions, such as polynomial, Fourier, and divergent functions, the spline basis consists of a set of localized modes, each of which describes only a small region of the allowed  $k$ -space. This makes the spline basis very well-suited to describing shapes such as those of non-Bunch-Davies models, that are characterized by highly-peaked features concentrated on very flattened triangles. Because of the localization of spline modes, the modes are by construction nearly orthogonal, and no orthogonalization procedure (such as Gram-Schmidt) is used in our implementation. Avoiding any explicit orthogonalization is an advantage, as orthogonalizing the polynomial/Fourier basis sets via Gram-Schmidt is numerically unstable, and requires very high precision throughout to create a basis with a large number of modes.<sup>6</sup>

#### IV. FITTING NON-BUNCH-DAVIES SHAPES

In this section we consider two non-Bunch-Davies shapes included in Planck's analysis that were not well-reconstructed in multipole-space using the polynomial modal expansion,  $S_{\text{NBD1}}$  and  $S_{\text{NBD2}}$  [28].<sup>7</sup> We also consider a set of generalized non-Bunch-Davies shapes,  $S_{\zeta^3}$ ,  $S_{\zeta^2}$ , and  $S_{\zeta(\partial\zeta)^2}$ ,<sup>8</sup> with different assumptions for the initial time boundary  $c_s\eta_0$  and its potential wavenumber dependence,  $c_s|\eta_0| = 10^3$  Mpc,  $(\Lambda/H)/(k_1 + k_2 + k_3)$ , and  $(\Lambda/H)/k_1$ , where we allow  $\Lambda/H$  to be 10 or  $10^3$ , representing a physically-motivated range of values. These shapes span a wide variety of possible non-Bunch-Davies features in the bispectrum, and allow us to compare our results with previous analyses.

<sup>6</sup> The numerical instability of the classical Gram-Schmidt algorithm can be partially mitigated by instead adopting the modified Gram-Schmidt algorithm, which is what we have implemented in generating the polynomial and Fourier modes.

<sup>7</sup> In the notation of [28],  $S_{\text{NBD}i}$  here is equal to  $(k_1 k_2 k_3)^2 B_{\Phi}^{\text{NBD}i} / (2A^2 f_{\text{NL}}^{\text{NBD}i})$ , where  $i = 1, 2$ .

<sup>8</sup> The shapes  $S_{\zeta^3}$ ,  $S_{\zeta^2}$ , and  $S_{\zeta(\partial\zeta)^2}$  here are related to corresponding leading order non-Bunch-Davies corrections to the bispectrum; for example,  $S_{\zeta^3}(k_1, k_2, k_3) = (k_1 k_2 k_3)^2 [\mathcal{F}_{\zeta^3}(-k_1, k_2, k_3, \eta_0) + \mathcal{F}_{\zeta^3}(k_1, -k_2, k_3, \eta_0) + \mathcal{F}_{\zeta^3}(k_1, k_2, -k_3, \eta_0) + \text{c.c.}]$ , where we have set  $a_{k_i} = 1$  and  $b_{k_i} = 0.01$ . In general  $b_{k_i}$  can have some scale-dependence as long as it does not spoil constraints from backreaction of the energy density in the initial state. Whatever choice is made for  $b_{k_i}$  should be applied consistently in the case of a joint analysis of the power spectrum and bispectrum.

For any non-separable primordial shape  $S$ , and given a choice of basis  $\{M_n\}$ , one can compute a separable fit  $S'$  that approximates  $S$  as a linear combination of separable basis functions,

$$S'(k_1, k_2, k_3) = \sum_n \alpha_n \mathcal{M}_n(k_1, k_2, k_3). \quad (25)$$

The similarity between the original shape and its fit is quantified by a cosine,

$$\cos(S, S') \equiv \frac{\langle S, S' \rangle}{\sqrt{\langle S, S \rangle \langle S', S' \rangle}}, \quad (26)$$

where the inner product is defined in Fourier space with a choice of weighting,

$$\langle S, S' \rangle \equiv \int d\mathcal{V}_T S(k_1, k_2, k_3) S'(k_1, k_2, k_3) w(k_1, k_2, k_3). \quad (27)$$

The volume  $\mathcal{V}_T$  includes only those combinations of  $k_1$ ,  $k_2$ , and  $k_3$  that can form a closed triangle, with each wavenumber satisfying  $k_{\min} \leq k_1, k_2, k_3 \leq k_{\max}$ , where  $k_{\min} = 10^{-3} \text{ Mpc}^{-1}$  and  $k_{\max} = 0.1 \text{ Mpc}^{-1}$ . The weight  $w(k_1, k_2, k_3)$  is typically taken to be either unity or  $1/(k_1 + k_2 + k_3)$ , where the latter choice is meant to represent a more accurate reflection of the scaling of the covariance of the CMB bispectrum, such that the Fourier-space cosine is closer to the multipole-space cosine between the CMB bispectra corresponding to the shapes  $S$  and  $S'$ . However, in general we find that both choices result in similar cosines. In our analysis, we use a unit weight for the 3D Fourier basis fits and all 2D fits, and a weight of  $1/(k_1 + k_2 + k_3)$  for the remaining fits.

In this section we implement the existing polynomial and Fourier basis methods, and additionally our new basis of piecewise splines, to obtain separable fits to non-Bunch-Davies shapes. In each case, we quantify the performance of the basis by computing the cosine as a function of an increasing number of modes used in the fit.

We use the polynomial basis described in [29] and the Fourier basis described in [30]. In each case, three 1D functions based on either polynomials or sines/cosines of  $k_i$  are multiplied together to form 3D separable functions, that are then orthogonalized using a Gram-Schmidt algorithm to create a basis of 3D orthonormalized and separable functions, called  $\{\mathcal{R}_n\}$  for polynomials and  $\{\mathcal{F}_n\}$  for Fourier modes. Since the basis functions are orthonormal in each case, the expansion coefficients  $\{\alpha_n\}$  can be computed through inner products between  $S$  and the basis functions,  $\alpha_n = \langle S, \mathcal{R}_n \rangle$  or  $\alpha_n = \langle S, \mathcal{F}_n \rangle$ .

Alternatively, in the spline basis expansion, three 1D piecewise spline functions are multiplied together to form a basis of 3D separable spline functions  $\{B_n\}$ . In this case, each mode is highly localized in a region of Fourier space, so any two modes are orthogonal by construction unless they have peaks that overlap. While the lack of strict orthogonality means that the expansion coefficients

in the spline basis cannot be computed using simple inner products, existing algorithms can solve for the coefficients efficiently [53] (also see appendix C).

In the subsections that follow, we present polynomial, Fourier, and spline basis fits to a variety of non-Bunch-Davies shapes.

### A. $S_{\text{NBD1}}$ and $S_{\text{NBD2}}$

The cosines for polynomial, Fourier, and spline fits to these shapes are shown in the upper left panel of fig. 2. We find that the polynomial expansion produces a better fit than the Fourier basis, and polynomial cosines typically increase slowly beyond about 100 modes for  $S_{\text{NBD1}}$  and 50 modes for  $S_{\text{NBD2}}$ , indicating that lower order modes contribute most to the fits. While it is possible that increasing the number of polynomial or Fourier modes will increase the cosines further, during our analysis we found that generating large polynomial and Fourier basis sets is computationally very demanding. The separable modes are orthogonalized using a Gram-Schmidt algorithm, which is known to be numerically unstable, and the instabilities become more severe as higher order polynomials are used. On the other hand, the spline basis expansion as we have implemented it does not require orthogonalization, so in comparison a large number of modes can easily be generated and used in the separable fits. In the case of  $S_{\text{NBD1}}$  and  $S_{\text{NBD2}}$ , we find that the spline expansion performs similarly well as the polynomial expansion when 200 modes are used, and better fits can be achieved by using a larger number of modes.

### B. $S_{\zeta^3}$ , $S_{\zeta^2}$ , and $S_{\zeta(\partial\zeta)^2}$ with $k$ -independent $c_s\eta_0$

These shapes have oscillatory features that grow with the size of the triangle (i.e. with  $k_1$ ) and in the flattened and squeezed limits. With  $c_s|\eta_0| = 10^3$  Mpc, the set of 200 polynomial modes produces a cosine of 0.52 for  $S_{\zeta^3}$  and 0.80 for  $S_{\zeta^2}$ , as shown in the upper right panel of fig. 2.<sup>9</sup> While these modes produce a higher cosine than a set of 200 spline functions, better fits can be generated with a larger set of splines.

### C. $S_{\zeta^3}$ , $S_{\zeta^2}$ , and $S_{\zeta(\partial\zeta)^2}$ with $k$ -dependent $c_s\eta_0$

For small  $\Lambda/H$ , we find that the shapes are easily reconstructed with both the polynomial and spline methods using  $\lesssim 200$  modes, as shown in lower left panel

of fig. 2. The oscillatory features are of low enough frequency that lower order polynomials are sufficient to capture most of the features of these shapes, and the spline functions also do not need to have a very fine resolution. If the initial conditions are set at  $10/k_1$ , rather than  $10/(k_1 + k_2 + k_3)$ , then the oscillatory features are of somewhat higher frequency; for either shape, however, the large global features still allow both the polynomial and spline bases to efficiently reconstruct these shapes.

For cases with  $k$ -dependent  $c_s\eta_0$  and larger values of  $\Lambda/H$ , the oscillations have a much higher frequency than what can be captured by polynomials, and we find that the spline reconstructions perform similarly poorly. However, the  $k$ -dependence of  $c_s\eta_0$ , whether it be  $(\Lambda/H)/(k_1 + k_2 + k_3)$  or  $(\Lambda/H)/k_1$ , allows the shapes to be rewritten as functions of only two free parameters: the ratios  $x \equiv k_3/k_1$  and  $y \equiv k_2/k_1$ . In terms of  $x$  and  $y$ , the oscillation frequency does not increase drastically throughout the allowed parameter space, which makes it easier to generate good 2D fits. The advantage of rewriting scale-invariant shapes such as these in terms of two free parameters for computing  $k$ -space cosines and CMB bispectra has been discussed in earlier works [54].

We generate the 2D fits by defining a 2D analogue of the polynomial/spline basis sets and the cosine. We find, however, that to generate the same number of polynomial modes as in the 3D case (200) requires using higher order 1D polynomials, which exacerbates the numerical issues that we encountered in the 3D case, so for the 2D fits we only use 100 polynomial modes with unit weight. We do not encounter similar numerical issues in the spline fits.

We show the results for the 2D fits in the  $c_s|\eta_0| = 10^3/(k_1 + k_2 + k_3)$  case in the lower right panel of fig. 2; for  $c_s|\eta_0| = 10^3/k_1$  the shapes have very similar features, with the oscillations being slightly more rapid and more difficult to represent in the latter case. 2D spline fits achieve similar cosines as 2D polynomial fits using the same number of modes ( $\sim 100$ ), while also being able to produce higher cosines through the addition of more modes without running into numerical issues, as shown in fig. 3.

## V. DISCUSSION

If the CMB is to be used to its full potential to elucidate the details of inflation, we need to be able to both effectively characterize a wide variety of potential inflationary signatures while also ensuring that this information is accessed in a timely manner. For non-Gaussian signatures, present in the CMB bispectrum, a potential bottleneck in this process is the production of separable templates that accurately match the characteristic properties of the underlying shape. This issue is particularly acute for initial states that differ from the Bunch-Davies state, which can generically have highly oscillatory features whose characteristic scale can vary over the bispectrum configuration space.

<sup>9</sup> The shape for  $S_{\zeta(\partial\zeta)^2}$  is sufficiently similar to  $S_{\zeta^2}$  that its fits are not shown separately in the figures.

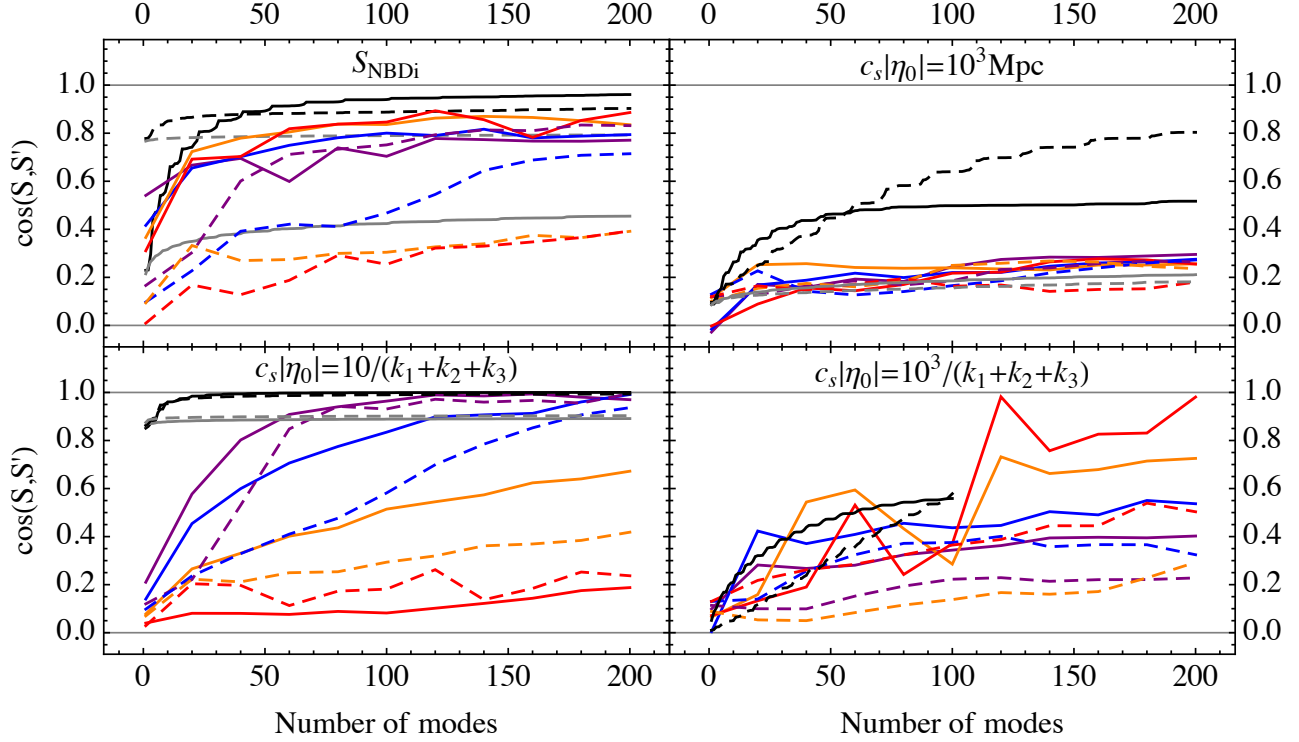


FIG. 2: Comparisons of shape reconstructions for  $S_{\text{NBD1}}$  (solid) and  $S_{\text{NBD2}}$  (dashed) [upper left panel] or  $S_{\xi_3}$  (solid) and  $S_{\xi_2}$  (dashed) [other three panels] using the polynomial (black), Fourier (gray), and b-spline methods. *Top row and lower left panels:* The 3D spline basis reconstructions correspond to the colored curves, where the purple, blue, orange, and red colors correspond to spline basis sets derived from using 10, 14, 22, and 40 1D spline functions in each dimension. *Lower right panel:* The 2D spline basis reconstructions correspond to the colored curves, where the purple, blue, orange, and red colors correspond to spline basis sets derived from using 50, 100, 200, and 300 1D spline functions in each dimension.

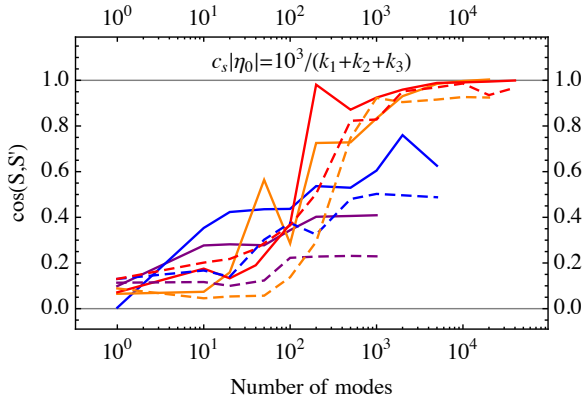


FIG. 3: 2D b-spline fits to  $S_{\xi_3}$  (solid) and  $S_{\xi_2}$  (dashed), where the purple, blue, orange, and red colors correspond to spline basis sets derived from using 50, 100, 200, and 300 1D spline functions in each dimension.

In this work we have analyzed a variety of functions to

generate templates for bispectrum shapes arising in a variety of non-Bunch-Davies scenarios. We have quantified how well different choices of separable basis functions, derived from polynomials, Fourier functions, and b-splines, can reconstruct the original non-Bunch-Davies shapes. The spline expansion method is a new alternative choice of basis, that we implement here for the first time, and can be used at a reasonable computational cost to obtain cosines that can be very close to unity.

We find that the polynomial basis is good for describing some non-Bunch-Davies shapes with large features, and generally performs better than the Fourier basis. For the rest of the shapes we considered, assuming the low cosines will steadily increase with the addition of more modes, we find that numerical difficulties prevent us from generating enough modes to see higher cosines. For most shapes, the spline basis performs as well as the polynomial one when equal numbers of modes are chosen. The spline basis is both numerically simpler to compute, and does not require orthogonalization due to its localized nature. This allows a larger number of modes to be efficiently calculated to improve the match between the



templates and the actual shapes.

The spline basis expansion method is very flexible, and there are other ways to adapt a spline basis to target specific shapes better than we have not explored here. For example, while our spline bases are derived from equally spaced knots, we note that one can create a basis using unequally-spaced knots, such that different  $k$ -space regions are sampled more finely than others. Non-Bunch-Davies shapes, where the features are sharply peaked and localized near flattened and squeezed configurations, can potentially be probed more efficiently by optimizing the spline basis in this way.

Finally, while our analysis takes place in primordial  $k$ -space, the flexibility and computational simplicity of the spline basis approach translates directly to multipole-space, where it complements existing approaches, such as the polynomial basis. Utilizing a variety of basis expansion techniques ensures that the exquisite CMB data available now, and in the future, can be used efficiently to explore the full theory space of viable inflationary scenarios.

### Acknowledgments

It is a pleasure to thank Donghui Jeong, Daan Meerburg, Raquel Ribeiro, and Sarah Shandera for very useful discussions. The work of J. B. and R. B. is supported by NASA ATP grants NNX11AI95G and NNX14AH53G, NASA ROSES grant 12-EUCLID12-0004, NSF CAREER grant 0844825 and DoE grant DE-SC0011838. J. B. would also like to thank the Cornell Graduate School for a travel grant that facilitated the conception of this paper. R. H. was supported in part by DOE grant DE-FG03-91-ER40682. He would also like to thank the Perimeter Institute for hospitality while this work was being completed.

### Appendix A: Power spectrum for general initial states

In this appendix we show that the result for the Green's function for general initial states in section II with a scale-dependent initial time  $\eta_0(k)$  gives the correct form of the late-time power spectrum in [10, 11]. We start with developing a precise definition for the adiabatic vacuum at  $\eta_0 \rightarrow -\infty$  or the Bunch-Davies vacuum. Let us write the field  $\chi_{\vec{k}}(\eta)$  (defined via  $\zeta = \frac{1}{\sqrt{2\varepsilon}} \frac{c_s}{a} \chi$ ) in terms of annihilation and creation operators at the time  $\eta_0$ ,  $a_{\vec{k}}(\eta_0)$  and  $a_{-\vec{k}}^\dagger(\eta_0)$ , and the mode functions  $f_k^{\gtrless}(\eta)$ , as

$$\chi_{\vec{k}}(\eta) = a_{\vec{k}}(\eta_0) f_k^>(\eta) + a_{-\vec{k}}^\dagger(\eta_0) f_k^<(\eta). \quad (\text{A1})$$

The conjugate momentum,  $\pi_{\vec{k}}(\eta)$  can be obtained from the quadratic Lagrangian for  $\chi_{\vec{k}}(\eta)$ ; at leading order this

is given by

$$\pi_{\vec{k}}(\eta) = \frac{\partial \mathcal{L}^{(2)}}{\partial \dot{\chi}} = \frac{d\chi_{\vec{k}}}{d\eta} - \frac{1}{a} \frac{da}{d\eta} \chi_{\vec{k}}, \quad (\text{A2})$$

We can write  $\pi_{\vec{k}}(\eta)$  in terms of its corresponding mode functions  $g_k^{\gtrless}(\eta)$  as

$$\pi_{\vec{k}}(\eta) = -i \left( a_{\vec{k}}(\eta_0) g_k^>(\eta) - a_{-\vec{k}}^\dagger(\eta_0) g_k^<(\eta) \right). \quad (\text{A3})$$

For Bunch-Davies modes we choose the positive frequency solution at early times, and using  $a(\eta) = -1/(\eta H)$  at leading order during inflation, the mode functions are given by

$$f_k^>(\eta) = \frac{1}{\sqrt{2c_s k}} e^{-ic_s k \eta} \left( 1 - \frac{i}{c_s k \eta} \right), \quad (\text{A4})$$

$$g_k^>(\eta) = \sqrt{\frac{c_s k}{2}} e^{-ic_s k \eta}. \quad (\text{A5})$$

The choice of Bunch-Davies vacuum can then be expressed as the following relationship between the field and its conjugate momentum in the infinite past,

$$\pi_{\vec{k}}(\eta_0) = (-ic_s k) \chi_{\vec{k}}(\eta_0). \quad (\text{A6})$$

Note that this does not imply that the position and momentum operators commute at all times; it is merely a statement of how they are related at  $\eta_0 \rightarrow -\infty$ . Equivalently, in terms of the mode function  $f_k^>(\eta)$  we can write

$$\left. \frac{df_k^>}{d\eta} \right|_{\eta=\eta_0} - \frac{1}{a} \frac{da}{d\eta} f_k^>(\eta_0) = (-ic_s k) f_k^>(\eta_0), \quad (\text{A7})$$

for  $\eta_0 \rightarrow -\infty$ . The above condition is the definition of the adiabatic vacuum in the infinite past, or equivalently the Bunch-Davies vacuum. As shown in [10, 55] this choice corresponds to a minimum uncertainty state.

The prescription to choose an adiabatic vacuum at a finite initial time is to enforce the same condition in eq. (A7) at a given  $\eta_0$ . This corresponds to a state which minimizes the uncertainty at  $\eta = \eta_0$ . Choosing the initial density matrix to be unity, but still having a non-Bunch-Davies initial state by allowing  $\beta_k^> \neq 0$ , i.e. with  $f_k^>(\eta)$  given by eq. (10) with  $A_k = 0$ , this leads to the following relation between the Bogoliubov coefficients,

$$\frac{\beta_k^>}{\alpha_k^>} = \left( \frac{i}{2c_s k \eta_0 + i} \right) e^{-2ic_s k \eta_0}. \quad (\text{A8})$$

Combining this with the usual condition  $|\alpha_k^>|^2 - |\beta_k^>|^2 = 1$  we find that

$$|\alpha_k^>|^2 = \frac{4c_s^2 k^2 \eta_0^2 + 1}{4c_s^2 k^2 \eta_0^2}. \quad (\text{A9})$$

Let us now choose  $\eta_0$  to be a function of  $k$  such that the physical momentum  $c_s k/a(\eta_0)$  crosses some fixed high energy scale  $\Lambda$  of new physics at  $\eta_0$ , then

$$\eta_0 = -\frac{\Lambda}{H c_s k}. \quad (\text{A10})$$

With the above equations we can write the following solution for  $\alpha_k^>$  and  $\beta_k^>$  (note that we are free to choose any overall phase),

$$\alpha_k^> = \left( \frac{2\Lambda/H - i}{2\Lambda/H} \right) e^{-i\Lambda/H}, \quad (\text{A11})$$

$$\beta_k^> = - \left( \frac{i}{2\Lambda/H} \right) e^{i\Lambda/H}. \quad (\text{A12})$$

For  $\Lambda/H \gg 1$  we can now write the late-time power spectrum as

$$\begin{aligned} \mathcal{P}_\zeta(k) &= \frac{k^3}{2\pi^2} \frac{c_s^2}{2\varepsilon} \frac{1}{a^2(\eta)} f_k^>(\eta) f_k^<(\eta) \Big|_{\eta \rightarrow 0^-} \\ &= \frac{H^2}{8\pi^2 \varepsilon c_s} \left[ 1 - \frac{H}{\Lambda} \sin\left(\frac{2\Lambda}{H}\right) \right] \Big|_{c_s k = aH}, \end{aligned} \quad (\text{A13})$$

which includes a scale-dependent oscillatory term. The argument of the oscillations is usually written as being proportional to  $\ln(k/k_p)$ , where  $k_p$  is some fixed pivot scale. This can be seen by expanding  $H$  around the pivot scale so that  $H(k) \approx H(k_p) [1 - \varepsilon(N - N_p) + \dots] \approx H(k_p) \left[ 1 - \varepsilon \ln \frac{k}{k_p} + \dots \right]$ , leading to logarithmic oscillations in the power spectrum. Note that this expansion only holds for small  $\varepsilon$  and a reasonable range of scales.

## Appendix B: Bispectrum for general initial states

The functions  $\mathcal{F}_{\zeta^3}$ ,  $\mathcal{F}_{\zeta^2\zeta}$ , and  $\mathcal{F}_{\zeta(\partial\zeta)^2}$  in eqs. (17) - (19) are given by

$$\mathcal{F}_{\zeta^3}(p_1, p_2, p_3, \eta_0) = \frac{1}{k_1 k_2 k_3} \left[ -\frac{2}{K_1^3} + \frac{e^{ic_s K_1 \eta_0}}{K_1} \left( \frac{2}{K_1^2} - \frac{2ic_s \eta_0}{K_1} - c_s^2 \eta_0^2 \right) \right], \quad (\text{B1})$$

$$\begin{aligned} \mathcal{F}_{\zeta^2\zeta}(p_1, p_2, p_3, \eta_0) &= \frac{1}{k_1 k_2 k_3} \left[ \frac{-2K_1^3 K_3^3 + K_1^2 K_2^4 + K_1 K_2^2 K_3^3}{K_1^3 K_3^6} \right. \\ &\quad \left. + \frac{e^{ic_s K_1 \eta_0}}{K_1^3 K_3^6} (2K_1^3 K_3^3 - K_1^2 K_2^4 - K_1 K_2^2 K_3^3 + ic_s K_1^2 K_2^2 K_3^3 \eta_0) \right], \end{aligned} \quad (\text{B2})$$

and

$$\begin{aligned} \mathcal{F}_{\zeta(\partial\zeta)^2}(p_1, p_2, p_3, \eta_0) &= \frac{1}{(k_1 k_2 k_3)^3} \left[ \frac{K_1^6 - 3K_1^4 K_2^2 - K_1^3 K_3^3 + 2K_1^2 K_2^4 + 2K_1 K_2^2 K_3^3}{K_1^3} \right. \\ &\quad \left. + \frac{e^{ic_s K_1 \eta_0}}{c_s K_1^3 \eta_0} \left\{ i(K_1^5 - 2K_1^3 K_2^2) + c_s (K_1^4 K_2^2 + K_1^3 K_3^3 - 2K_1^2 K_2^4 - 2K_1 K_2^2 K_3^3) \eta_0 \right. \right. \\ &\quad \left. \left. - ic_s^2 (K_1^4 K_3^3 - 2K_1^2 K_2^2 K_3^3) \eta_0^2 \right\} \right], \end{aligned} \quad (\text{B3})$$

where for brevity of notation we have suppressed the explicit momentum dependence of the functions  $K_1$ ,  $K_2$ , and  $K_3$ ,

$$K_1(p_1, p_2, p_3) = p_1 + p_2 + p_3, \quad (\text{B4})$$

$$K_2(p_1, p_2, p_3) = (p_1 p_2 + p_2 p_3 + p_3 p_1)^{1/2}, \quad (\text{B5})$$

$$K_3(p_1, p_2, p_3) = (p_1 p_2 p_3)^{1/3}. \quad (\text{B6})$$

In the next two subsections we show how one obtains the flattened and squeezed enhancements from the functions  $\mathcal{F}$  (also see [56]). We work with the simplest function  $\mathcal{F}_{\zeta^3}$ , though the results are similar for the other two functions (or at least for their appropriate sum) as well.

### 1. Flattened limit

The  $b_{k_1} a_{k_2} a_{k_3}$  term leads to an enhanced flattened limit ( $k_1 \approx k_2 + k_3$ ,  $k_1$  being the largest momentum mode) bispectrum. Let us first assume that  $b_{k_1}$  is purely imaginary, so that the bispectrum is proportional to the imaginary part of  $\mathcal{F}$ . The corresponding  $\mathcal{F}$  function we consider is,

$$\begin{aligned} \mathcal{F}_{\zeta^3}(-k_1, k_2, k_3, \eta_0) &= -\frac{2}{\tilde{k}_1^3 k_1 k_2 k_3} \\ &\quad + \frac{e^{ic_s \tilde{k}_1 \eta_0}}{\tilde{k}_1 k_1 k_2 k_3} \left( \frac{2}{\tilde{k}_1^2} - \frac{2ic_s \eta_0}{\tilde{k}_1} - c_s^2 \eta_0^2 \right), \end{aligned} \quad (\text{B7})$$

where  $\tilde{k}_1 \equiv -k_1 + k_2 + k_3$ . In the limit of  $\tilde{k}_1 \rightarrow 0$ , the exponential can be expanded as

$$\lim_{\tilde{k}_1 \rightarrow 0} e^{ic_s \tilde{k}_1 \eta_0} = 1 + ic_s \tilde{k}_1 \eta_0 - \frac{c_s^2 \tilde{k}_1^2 \eta_0^2}{2} - \frac{ic_s^3 \tilde{k}_1^3 \eta_0^3}{6} + \dots \quad (\text{B8})$$

Using this in eq. (B7), and noticing that any term with  $\tilde{k}_1$  in the numerator goes to zero, we find that

$$\lim_{\tilde{k}_1 \rightarrow 0} \mathcal{F}_{\zeta^3}(-k_1, k_2, k_3, \eta_0) = -\frac{i}{3(k_2 + k_3)k_2 k_3} c_s^3 \eta_0^3. \quad (\text{B9})$$

For fixed  $\eta_0$ , we can set  $c_s |\eta_0| = 1/k_*$ . The above limit of the three-point function is therefore enhanced (though not divergent) in the flattened limit. For  $\eta_0(k)$  with large  $\Lambda/H$  as well we see an enhancement in the flattened limit. If we instead assume that  $b_{k_1}$  is real, then the bispectrum is still enhanced, though not in the exactly flattened limit but in a near-flattened limit.

## 2. Squeezed limit

Let us now look at the  $a_{k_1} b_{k_2} a_{k_3}$  piece,

$$\mathcal{F}_{\zeta^3}(k_1, -k_2, k_3, \eta_0) = -\frac{2}{\tilde{k}_2^3 k_1 k_2 k_3} + \frac{e^{ic_s \tilde{k}_2 \eta_0}}{\tilde{k}_2 k_1 k_2 k_3} \left( \frac{2}{\tilde{k}_2^2} - \frac{2ic_s \eta_0}{\tilde{k}_2} - c_s^2 \eta_0^2 \right), \quad (\text{B10})$$

where  $\tilde{k}_2 \equiv k_1 - k_2 + k_3$ . In the squeezed limit ( $k_3 \ll k_1 \approx k_2$ ) we have  $\tilde{k}_2 \rightarrow k_{\min}$ , where  $k_{\min}$  is the smallest momentum mode observable today. Using this in eq. (B10) we find that

$$\lim_{\tilde{k}_2 \rightarrow k_{\min}} \mathcal{F}_{\zeta^3}(k_1, -k_1, k_3, \eta_0) = -\frac{2}{k_1^2 k_{\min}^4} + \frac{e^{ic_s k_{\min} \eta_0}}{k_1^2 k_{\min}^2} \left( \frac{2}{k_{\min}^2} - \frac{2ic_s \eta_0}{k_{\min}} - c_s^2 \eta_0^2 \right). \quad (\text{B11})$$

For fixed  $\eta_0$  and in the limit of  $k_{\min} \gg k_*$ , the  $c_s^2 \eta_0^2$  term gives the largest contribution. This term is multiplied with a highly oscillatory function though, and averaging over the large argument of the cosine (real part of the exponential) we expect its contribution to vanish. The leading order contribution is then proportional to  $1/(k_1^2 k_{\min}^4)$ , which shows a strong squeezed limit enhancement. In the limit of  $k_{\min} \gtrsim k_*$  or for  $\eta_0(k)$  this argument no longer holds and we may or may not see enhancements.<sup>10</sup>

## Appendix C: B-splines fitting algorithm

In this appendix, we build on the discussion of section III to describe in more detail the b-spline fitting algorithms we have used, and illustrate b-spline fitting examples for a simple 1D function and a 2D representation of the scale-invariant enfolded template. Snippets of Mathematica 10 codes we have implemented are shown here, and the same algorithms were first constructed and explicitly shown as Matlab code in [53].

Our first example considers a spline fit to a simple 1D function  $f(x)$ , with  $0 \leq x \leq 1$ . To generate a fit, we first make two choices: a choice of basis and a choice of data points to fit. The spline basis is determined by a choice of  $k+1$  equidistant knots,  $\{x_0, x_1, \dots, x_k\}$ , at which each basis function's degree  $q$  polynomial pieces will be joined. The b-splines in general do not have to be generated using constant knot intervals, but for simplicity we always start with equidistant knots, and additionally include  $q$  extra knots at each of  $x=0$  and  $x=1$  to produce a ‘‘clamped’’ basis. Without these extra knots, the generated basis sets do not have splines with non-zero amplitudes at  $x=0$  and  $x=1$ , and it will be difficult to fit functions that are non-zero at the endpoints.

Given these inputs, existing codes, such as Mathematica's `BSplineBasis` function, can recursively generate a basis of  $k+q$  b-splines such that each one is spanned by  $q+2$  knots, made up of  $q+1$  polynomial pieces, with derivatives continuous up to order  $q-1$ . In addition, the sum of all b-spline amplitudes at any  $x$  is unity. For example, the b-splines in fig. 1 are easily generated by choosing  $q=3$  and  $k=3$  such that the knots vector is `knots = {0,0,0,0,1/3,2/3,1,1,1,1}`, and executing `BSplineBasis[{q,knots},i,x]`, where  $i$  is an integer  $0 \leq i \leq k+q-1$  identifying each particular b-spline. In this work, we vary the number and widths of the b-splines by varying  $k$ , but always keep the degree fixed to  $q=3$ .

Next, the choice of data points depends on how finely we wish to sample  $f(x)$ . We would like to choose a data set consisting of  $M$  data pairs,  $(x_i, f_i)$ , where  $f_i \equiv f(x_i)$ , such that our data resolves any potentially fine features in the function we would like to fit, without including so many extra data points that our numerical calculation becomes intractable. The final fit should ultimately be insensitive to the sampling we have chosen. Again, for simplicity, we always use equidistant sampling points  $x_i$  in our analysis, but vary the density of sampling points by changing  $M$ .

The spline fit then approximates the original function as

$$f'(x) = \sum_{n=0}^{N-1} \alpha_n B_n(x), \quad (\text{C1})$$

where the expansion coefficients  $\{\alpha_n\}$  are solved for by

<sup>10</sup> In our fits with fixed  $\eta_0$  in section IV we took  $k_*$  to be similar to  $k_{\min}$  ( $\sim 10^{-3}$  Mpc $^{-1}$ ) which is what the Planck team had used in [28].

minimizing the least-squares function,

$$\text{LS} = \sum_{i=1}^M \left( f_i - \sum_{n=0}^{N-1} \alpha_n B_n(x_i) \right)^2. \quad (\text{C2})$$

This requires solving the linear system of equations given by  $B^T B \vec{\alpha} = B^T \vec{f}$ , where  $B$  is an  $M \times N$  matrix containing the values of  $B_n(x_i)$ , and is easily performed with algorithms such as Mathematica's `LinearSolve`.

We note one extension of b-splines, called p-splines, that aims to avoid overfitting a set of input data by imposing smoothness on the resulting fit. Short for “penalty b-splines”, in the p-spline method, the fit's expansion coefficients are determined by both the choice of basis and the choice of input data, plus a choice of penalty function that generally disfavors fits with large differences between coefficients of neighboring b-splines [57]. In this context one would instead minimize

$$\text{LS} = \sum_{i=1}^M \left( f_i - \sum_{n=0}^{N-1} \alpha_n B_n(x_i) \right)^2 + \lambda \sum_{j=k}^{N-1} (\Delta^k \alpha_j)^2, \quad (\text{C3})$$

where  $\lambda$  is a constant that controls the smoothness of the fit and  $k$  is the order of the penalty, a typical choice being  $k = 2$ , such that  $\Delta^2 \alpha_j = \alpha_j - 2\alpha_{j-1} + \alpha_{j-2}$ . The use of a penalty is optional, and its main purpose in the context of data fitting is to avoid fitting any noisy features in the data. Further, if there are not sufficiently many data points sampling  $f(x)$ , with many more splines than data points, then without a penalty the fits may display spurious features, as we will deliberately try to show in the 1D example that follows.

In fig. 4 we show fits to  $f(x) = \sin(10x)/(10x)$  with different choices of data points and smoothing parameter  $\lambda$ . We have fixed the knots at  $\{0, 0, 0, 0, 0.1, 0.2, \dots, 0.8, 0.9, 1, 1, 1, 1\}$ , yielding a basis of 13 b-splines. We find, as illustrated in the figure, that we can achieve good fits without introducing extra smoothing through a non-zero value of  $\lambda$ , as long as we fit to enough data points. So we now continue to an example of fitting a primordial shape in two dimensions, without a penalty.

To illustrate the b-spline fitting algorithm in two dimensions, we construct a basis of 2D b-splines and use it to fit the scale-invariant enfolded template,

$$S_{\text{enf}}(x, y) = \frac{1}{xy} (1 - x - y - x^2 - y^2 + x^3 + y^3 - x^2y - xy^2 + 3xy), \quad (\text{C4})$$

where  $x \equiv k_3/k_1$  and  $y \equiv k_2/k_1$ . As in the 1D case, the inputs to the fitting algorithms are made up of a choice of basis and a set of data points. The basis is specified by a choice of polynomial degree and a sequence of knots in each of the two dimensions,  $x$  and  $y$ . In our particular application, since we are aiming to fit shape functions that

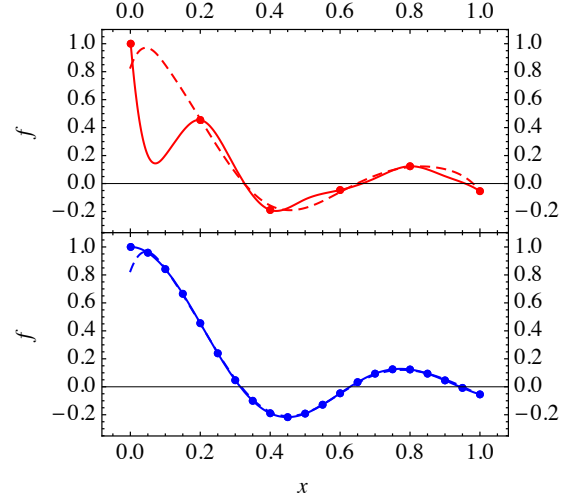


FIG. 4: Example 1D b-spline fits to  $f(x) = \sin(10x)/(10x)$ . All fits shown have used a basis of 13 splines. In the upper panel, the fits have been computed using 6 data points, sampled at  $x_i = \{0, 0.2, 0.4, 0.6, 0.8, 1\}$ . Attempting to construct such a fit without smoothing [solid red] causes the splines to produce spurious features, especially at smaller values of  $x$ , while introducing a penalty and a small amount of smoothing [dashed red],  $\lambda = 0.1$ , restores the fit to a reasonable representation of the true function. The lower panel is the same fit, except with 21 data points, sampled at  $x_i = \{0, 0.05, 0.1, \dots, 0.9, 0.95, 1\}$ . In this case, the fits with [dashed blue] and without [solid blue] smoothing are very similar.

are symmetric in their wavenumber arguments, we only specify the knots and degree in one dimension, and use the same b-spline basis for the additional second dimension. The 2D b-splines are then made of products of any two 1D splines, for example,  $B_n(x)B_m(y)$ . The data are given by  $(x_i, y_j, S(x_i, y_j))$  and stored in  $Y_{ij} = S(x_i, y_j)$ , where again some care must be taken in the choice of sampling, which must be dense enough to capture any small features such as oscillations that we would like to capture in the resulting fit.

The 2D analogue of the least-squares function in eq. (C2) is

$$\text{LS} = \sum_{i=1}^M \sum_{j=1}^M \left( Y_{ij} - \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} \alpha_{mn} B_m(x_i) B_n(y_j) \right)^2. \quad (\text{C5})$$

To turn the problem of solving for the expansion coefficients  $\alpha_{mn}$  into a linear system, we create a regression basis  $C$  from the  $M \times N$  b-spline basis matrices in each dimension,  $B_1$  and  $B_2$ , which in our case are equal. We define

$$C = (B_2 \otimes e_L^T) \odot (e_K^T \otimes B_1) \equiv B_1 \square B_2, \quad (\text{C6})$$

where  $\otimes$  is the Kronecker product,  $\odot$  is an element-by-element multiplication, the second equality defines the  $\square$  operation, and  $e_L$  and  $e_K$  are vectors of 1's with length  $L$  each. In Mathematica, we define the  $\square$  operation as box:

```
box[B1_, B2_] := Module[{K, L, eK, eL},
  K = Length[B1[[1, All]]];
  L = Length[B2[[1, All]]];
  eK = ConstantArray[1, {K}];
  eL = ConstantArray[1, {L}];
  KroneckerProduct[B2, {eL}]
  * KroneckerProduct[{eK}, B1]
]
```

Then by stacking the columns of the coefficients array  $\alpha_{mn}$  and the data array  $Y_{ij}$  to get vectors  $\vec{\beta}$  and  $\vec{y}$  respectively, the task of finding a solution for the coefficients is once again reduced to solving a linear system of equations given by  $C^T W C \vec{\beta} = C^T W \vec{y}$ . Here  $W$  is a matrix containing weights, which may be different for each data point, but for simplicity we restrict ourselves to using a weight of unity for all of our data.

For modest amounts of data and numbers of b-splines, one can quickly solve for the coefficients in this straightforward way. For large data sets and numbers of b-splines, however, this approach becomes computationally cumbersome due to the large size of  $C$ . While it is still possible to numerically solve for the coefficients  $\alpha_{mn}$  using a low-level language like C(++) or Fortran, we have instead used algorithms developed for higher level languages, such as Matlab in [53], using only vector and matrix operations. This has the benefit of being easier to implement, while still being able to sidestep much of the memory storage and speed issues typical of a more brute-force approach in a high-level language. Instead of starting with a calculation of  $C$  in the brute-force approach, the algorithm from [53] that we have adopted computes  $C^T W C$  and  $C^T W \vec{y}$  using only  $B$ . We refer the reader to [53] for a detailed discussion of how the method itself is devised and constructed, or to see the equivalent Matlab code, and present here an implementation of the b-spline fitting algorithms in Mathematica.

The normal equations can be efficiently constructed and solved, given an input of data in  $Y$  and information about the data sampling and b-spline basis in  $B$ :

```
get2dfit[Y_, B_] := Module[{m, n, W, R, r,
  F, a, A},
  m = Length[Y[[1, All]]];
  n = Length[B[[1, All]]];
  W = ConstantArray[1, {m, m}];
  R = Transpose[B].(W*Y).B;
  r = ArrayReshape[R, {n*n, 1}];
  F = Transpose[box[B, B]].W.box[B, B];
  F = ArrayReshape[F, {n, n, n, n}];
  F = TensorTranspose[F, Cycles{{3, 2}}];
  F = ArrayReshape[F, {n*n, n*n}];
  a = LinearSolve[F, r];
  A = ArrayReshape[a, {n, n}]
]
```

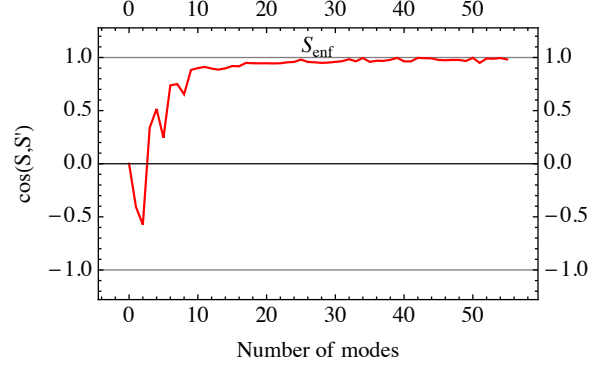


FIG. 5: Cosines between the enfolded shape and the 2D b-spline fits generated by a basis with 10 splines per dimension.

]

After the matrix of coefficients is solved for, we construct the final fit through two steps. First, we map the coefficients output as  $A$  from `get2dfit` to a new set of coefficients that corresponds to a 2D basis of splines which is symmetric in its two arguments, so that each 2D basis mode is a sum of up to two terms:  $B_i(x)B_j(y) + B_j(x)B_i(y)$ . Second, in building up the fit, mode by mode, we start with the modes that contribute most to the fit. Since the b-splines in a choice of basis have similar shapes and amplitudes, we use the magnitude of the  $A_{ij}$  coefficient as a proxy for gauging how much any particular b-spline contributes to a fit's overall cosine with the original shape. This motivates building up a fit by adding in modes, starting with those that have the largest  $|A_{ij}|$ . The cosine is then computed in the usual way, through an inner product over  $(x, y)$ -space between the original shape  $S_{\text{enf}}$  and the fit.

As an example, we use a basis of 10 splines in each dimension constructed by choosing  $k = 7$  and  $q = 3$ , and use as our data set a grid of uniformly spaced  $(x, y)$  values from taking 50 samples in each dimension, to compute the matrix of coefficients  $A_{ij}$ , using the algorithms `box` and `get2dfit` above. The total number of symmetric modes is then 55, and the modes are ordered by their corresponding values of largest to smallest  $|A_{ij}|$  to produce the cosines in fig. 5. A visual comparison of the full fit using 55 modes and the original enfolded template is given in fig. 6.

For our 3D fits, the fitting method is the same: a choice of b-spline basis and data set make up the inputs to the algorithm, which returns an array  $A_{mnp}$  containing the expansion coefficients that approximate the input shape as  $S'(k_1, k_2, k_3) = \sum_{mnp} A_{mnp} B_m(k_1) B_n(k_2) B_p(k_3)$ . However, due to the higher dimensionality of the problem, we must introduce a new function, `rho`, to generalize the matrix product to the product of a matrix and a 3D array. Below, we show Mathematica code for `rho[A, B, p]`, which computes the normal matrix product between rows of  $A$  and the  $p^{\text{th}}$  column of  $B$ , resulting in

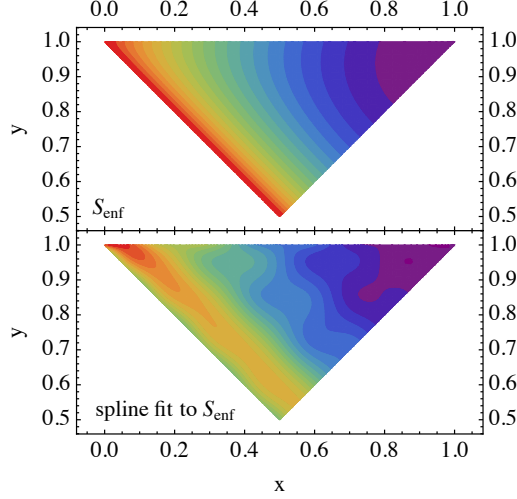


FIG. 6: The original template  $S_{\text{enf}}(x, y)$  [upper panel] and its b-spline fit using a set of 55 modes [lower panel]. The amplitudes range from 0 (violet) to 1 (red) in both panels.

a product,  $C$ , which has the same dimensions as  $B$ :

```
rho[A_, B_, p_] := Module[{sa, sb, n, ip,
  cycles, sbip, prodsbip, tempB, C},
  sa = Dimensions[A];
  sb = Dimensions[B];
  n = Length[sb];
  ip = Join[Range[p + 1, n], Range[1, p - 1]];
  Which[
    p == 1, cycles = Cycles[{}],
    p == 2, cycles = Cycles[{{1, 3, 2}}],
```

```
    p == 3, cycles = Cycles[{{1, 2, 3}}]];
  tempB = TensorTranspose[B, cycles];
  sbip = sb[[ip]];
  prodsbip = Product[sbip[[i]],
    {i, 1, Length[sbip]}];
  tempB = ArrayReshape[tempB, {sb[[p]],
    prodsbip}];
  C = Transpose[A].tempB;
  C = ArrayReshape[C, Join[{sa[[2]]},
    sb[[ip]]]];
  C = TensorTranspose[C,
    InversePermutation[cycles]]
]
```

Given this definition of  $\rho$ , the 3D b-spline fit coefficients are calculated using `get3dfit`:

```
get3dfit[Y_, B_] := Module[{m, n, W, F, R, A},
  m = Length[Y[[1, 1, All]]];
  n = Length[B[[1, All]]];
  W = ConstantArray[1, {m, m, m}];
  F = rho[box[B, B], W, 1];
  F = rho[box[B, B], F, 2];
  F = rho[box[B, B], F, 3];
  R = rho[B, Y * W, 1];
  R = rho[B, R, 2];
  R = rho[B, R, 3];
  F = ArrayReshape[F, {n, n, n, n, n}];
  F = TensorTranspose[F,
    Cycles[{{3, 2, 4, 5}}]];
  F = ArrayReshape[F, {n^3, n^3}];
  A = LinearSolve[F,
    ArrayReshape[R, {n^3, 1}]];
  A = ArrayReshape[A, {n, n, n}]
]
```

- 
- [1] A. Vilenkin and L. H. Ford, Phys.Rev. **D26**, 1231 (1982).
  - [2] A. Dey and S. Paban, JCAP **1204**, 039 (2012), 1106.5840.
  - [3] A. Dey, E. Kovetz, and S. Paban, JCAP **1210**, 055 (2012), 1205.2758.
  - [4] I. Agullo, A. Ashtekar, and W. Nelson, Class.Quant.Grav. **30**, 085014 (2013), 1302.0254.
  - [5] L. Lello, D. Boyanovsky, and R. Holman, Phys.Rev. **D89**, 063533 (2014), 1307.4066.
  - [6] K. Sugimura and E. Komatsu, JCAP **1311**, 065 (2013), 1309.1579.
  - [7] C. R. Contaldi, R. Bean, and J. Magueijo, Phys.Lett. **B468**, 189 (1999), astro-ph/9910309.
  - [8] J. Martin and R. H. Brandenberger, Phys.Rev. **D63**, 123501 (2001), hep-th/0005209.
  - [9] N. Kaloper, M. Kleban, A. E. Lawrence, and S. Shenker, Phys.Rev. **D66**, 123510 (2002), hep-th/0201158.
  - [10] U. H. Danielsson, Phys.Rev. **D66**, 023511 (2002), hep-th/0203198.
  - [11] R. Easther, B. R. Greene, W. H. Kinney, and G. Shiu, Phys.Rev. **D66**, 023518 (2002), hep-th/0204129.
  - [12] X. Chen, M.-x. Huang, S. Kachru, and G. Shiu, JCAP **0701**, 002 (2007), hep-th/0605045.
  - [13] R. Holman and A. J. Tolley, JCAP **0805**, 001 (2008), 0710.1302.
  - [14] P. D. Meerburg, J. P. van der Schaar, and P. S. Corasaniti, JCAP **0905**, 018 (2009), 0901.4044.
  - [15] I. Agullo and L. Parker, Phys.Rev. **D83**, 063526 (2011), 1010.5766.
  - [16] A. Ashoorioon and G. Shiu, JCAP **1103**, 025 (2011), 1012.3392.
  - [17] J. Ganc, Phys. Rev. **D84**, 063514 (2011), 1104.0244.
  - [18] D. Chialva, JCAP **1210**, 037 (2012), 1108.4203.
  - [19] S. Kundu, JCAP **1202**, 005 (2012), 1110.4688.
  - [20] N. Agarwal, R. Holman, A. J. Tolley, and J. Lin, JHEP **1305**, 085 (2013), 1212.1172.
  - [21] S. Brahma, E. Nelson, and S. Shandera, Phys.Rev. **D89**, 023507 (2014), 1310.0471.
  - [22] S. Bahrami and E. E. Flanagan, JCAP **1410**, 010 (2014), 1310.4482.

- [23] R. Flauger, D. Green, and R. A. Porto, JCAP **1308**, 032 (2013), 1303.1430.
- [24] A. Aravind, D. Lorshbough, and S. Paban, JHEP **1307**, 076 (2013), 1303.1440.
- [25] A. Ashoorioon, K. Dimopoulos, M. M. Sheikh-Jabbari, and G. Shiu, JCAP **1402**, 025 (2014), 1306.4914.
- [26] P. A. R. Ade et al. (Planck collaboration) (2015), 1502.02114.
- [27] C. L. Bennett et al. (WMAP collaboration), Astrophys.J.Suppl. **208**, 20 (2013), 1212.5225.
- [28] P. A. R. Ade et al. (Planck collaboration), Astron.Astrophys. **571**, A24 (2014), 1303.5084.
- [29] J. R. Fergusson, M. Liguori, and E. P. S. Shellard, Phys.Rev. **D82**, 023502 (2010), 0912.5516.
- [30] P. D. Meerburg, Phys.Rev. **D82**, 063517 (2010), 1006.2771.
- [31] J. Byun and R. Bean, JCAP **1309**, 026 (2013), 1303.3050.
- [32] M. Munchmeyer, F. Bouchet, M. G. Jackson, and B. Wandelt, Astron.Astrophys. **570**, A94 (2014), 1405.2550.
- [33] M. Munchmeyer, P. D. Meerburg, and B. D. Wandelt, Phys.Rev. **D91**, 043534 (2015), 1412.3461.
- [34] C. de Boer, Springer (2001).
- [35] P. Dierckx, Oxford University Press (1995).
- [36] J. M. Maldacena, JHEP **0305**, 013 (2003), astro-ph/0210603.
- [37] C. Cheung, P. Creminelli, A. L. Fitzpatrick, J. Kaplan, and L. Senatore, JHEP **0803**, 014 (2008), 0709.0293.
- [38] C. Cheung, A. L. Fitzpatrick, J. Kaplan, and L. Senatore, JCAP **0802**, 021 (2008), 0709.0295.
- [39] G. W. Horndeski, Int.J.Theor.Phys. **10**, 363 (1974).
- [40] X. Gao and D. A. Steer, JCAP **1112**, 019 (2011), 1107.2642.
- [41] A. De Felice and S. Tsujikawa, Phys.Rev. **D84**, 083504 (2011), 1107.3917.
- [42] C. Burrage, R. H. Ribeiro, and D. Seery, JCAP **1107**, 032 (2011), 1103.4126.
- [43] R. H. Ribeiro, JCAP **1205**, 037 (2012), 1202.4453.
- [44] N. Agarwal, R. H. Ribeiro, and R. Holman, JCAP **1406**, 016 (2014), 1311.0869.
- [45] J. S. Schwinger, J.Math.Phys. **2**, 407 (1961).
- [46] K. T. Mahanthappa, Phys.Rev. **126**, 329 (1962).
- [47] P. M. Bakshi and K. T. Mahanthappa, J.Math.Phys. **4**, 1 (1963).
- [48] L. P. Kadanoff and G. Baym, New York (1962).
- [49] P. M. Bakshi and K. T. Mahanthappa, J.Math.Phys. **4**, 12 (1963).
- [50] L. V. Keldysh, Zh.Eksp.Teor.Fiz. **47**, 1515 (1964).
- [51] R. D. Jordan, Phys.Rev. **D33**, 444 (1986).
- [52] E. Calzetta and B. L. Hu, Phys.Rev. **D35**, 495 (1987).
- [53] P. Eilers, I. Currie, and M. Durban, Computational Statistics and Data Analysis **50**, 61 (2006).
- [54] J. R. Fergusson and E. P. S. Shellard, Phys.Rev. **D76**, 083523 (2007), astro-ph/0612713.
- [55] D. Polarski and A. A. Starobinsky, Class.Quant.Grav. **13**, 377 (1996), gr-qc/9504030.
- [56] I. Agullo and S. Shandera, JCAP **1209**, 007 (2012), 1204.4409.
- [57] P. Eilers and B. Marx, Statistical Science **11**, 89 (1996).