Lawrence Berkeley National Laboratory
Recent Work

Title
Rotation method for accelerating multiple-spherical Bessel function integrals against a numerical source function

Permalink
https://escholarship.org/uc/item/1x24325s

Authors
Slepian, Zachary
Li, Yin
Schmittfull, Marcel
et al.

Publication Date
2019-11-29

Peer reviewed

Rotation method for accelerating multiple-spherical Bessel function integrals against a numerical source function

Zachary Slepian,[1,2,3]⋆ Yin Li,[3,4,7]† Marcel Schmittfull,[5] & Zvonimir Vlah[6]

[1]*Department of Astronomy, University of Florida, 211 Bryant Space Science Center, Gainesville, FL 32611-2055, USA*
[2]*Lawrence Berkeley National Laboratory, 1 Cyclotron Road, Berkeley, CA 94720, USA*
[3]*Berkeley Center for Cosmological Physics, University of California, Berkeley, Berkeley, CA 94720, USA*
[4]*Kavli Institute for Physic and Mathematics of the Universe, 5-1-5 Kashiwanoha, Kashiwa, Chiba Prefecture 277-8583, Japan*
[5]*Institute for Advanced Study, 1 Einstein Drive, Princeton, NJ 08540, USA*
[6]*Centre Européen pour Recherche Nucleaire, Espl. des Particules 1, 1217 Meyrin, Switzerland*
[7]*Center for Computational Astrophysics & Center for Computational Mathematics, Flatiron Institute, New York, NY, 10010, USA*

ABSTRACT
A common problem in cosmology is to integrate the product of two or more spherical Bessel functions (sBFs) with different configuration-space arguments against the power spectrum or its square, weighted by powers of wavenumber. These integrals generically emerge when correlation functions are evaluated at a displacement from the origin of coordinates. Naively computing them scales as $N_{\rm g}^{p+1}$ with p the number of configuration space arguments and $N_{\rm g}$ the grid size, and they cannot be done with Fast Fourier Transforms (FFTs). Here we show that by rewriting the sBFs as sums of products of sine and cosine and then using the product to sum identities, these integrals can then be performed using 1-D FFTs with $N_{\rm g} \log N_{\rm g}$ scaling. This refactorization is a $45°$ rotation from integrals onto pairs of configuration space arguments (e.g. (a,b)) to sums of integrals onto *single* differences $a-b$ and $a+b$. Hence we call it "the rotation method." It has the potential to accelerate significantly a number of calculations in cosmology, such as perturbation theory predictions of loop integrals, higher order correlation functions, and analytic templates for correlation function covariance matrices. We implement this approach numerically both in a free-standing, publicly-available PYTHON code and within the larger, publicly-available package `mcfit`. The rotation method evaluated with direct integrations already offers a factor of 6-10× speed-up over the naive approach in our test cases. Using FFTs, which the rotation method enables, then further improves this to a speed-up of $\sim 1000 - 3000×$ over the naive approach. The rotation method should be useful in light of upcoming large datasets such as DESI or LSST. In analysing these datasets recomputation of these integrals a substantial number of times, for instance to update perturbation theory predictions or covariance matrices as the input linear power spectrum is changed, will be one piece in a Monte Carlo Markov Chain cosmological parameter search: thus the overall savings from our method should be significant. We make our PYTHON code publicly available at https://github.com/eelregit/sbf_rotation

Key words: methods – cosmology: theory

1 INTRODUCTION

Fourier space and configuration space are the two complementary bases for most problems in cosmology. With the exception of radio, most observations are done in configuration space; for example, a map of the 3-D positions of galaxies or of Cosmic Microwave Background (CMB) temperature as a function of angle on the sky. However, many of the calculations are simpler in Fourier space. Advantages include that the Fourier Transform (FT) converts spatial derivatives into multiplication by powers of the wavevector, and that translation invariance is built in since for plane

---

⋆ E-mail: zslepian@ufl.edu (ZS)
† E-mail: yinli@flatironinstitute.org (YL)

waves it introduces only a trivial phase factor.[1] Moreover, for a Gaussian Random Field quantities like the power spectrum are diagonal in the wave-vector magnitude $|\vec{k}|$, and the covariance matrix is often simpler in Fourier space.[2] Further, at the level of linear perturbation theory, all Fourier modes evolve independently (e.g. Bernardeau et al. 2002).

Given these complementary advantages to the two spaces, one often needs to convert between them, which can be computationally expensive in some cases, e.g. for higher-order correlation functions and covariances. An added complication is that, while Fourier modes are well-suited to a Cartesian grid, problems in cosmology often invoke a degree of isotropy about the observer, making spherical coordinates a more natural basis than Cartesian.[3] Thus not only does one require an efficient way to transform between Fourier and configuration spaces, but one would like to end up in spherical coordinates.

For Fourier transforms from Cartesian wavevectors to Cartesian positions, Fast Fourier Transform (FFT) algorithms have long been available that reduce what is naively (for a 1-D transform) an N_g^2 problem to $N_g \log N_g$, where N_g is the number of grid points in configuration space or Fourier space (Cooley & Tukey 1965). The simplest versions of these algorithms use the Danielson-Lanczos lemma (Danielson & Lanczos 1942) to bisect the transform into even and odd pieces of length $N_g/2$, meaning the full result is given by a single complex transform of length $N_g/2$. The bisection can be continued until one has expressed the desired transform as a sum over c transforms of length unity, where c is the number of bisections. The factor of $\log N_g$ thus comes from the number of bisections N_b required, given by $2^{N_b} = N_g \rightarrow N_b \sim \log N_g$. This is why the simplest FFT requires that the input be a power of 2. Modern implementations relax this restriction.

For the spherical analog, one can use the plane wave expansion into spherical Bessel functions and spherical harmonics to convert the plane waves into spherical coordinates (e.g. Arfken et al. 2013 equation 16.61). Often the angular integrals can then be performed analytically; many problems in cosmology have relatively simple angular structure or indeed have full spherical symmetry. This integration leaves a 1-D integral from a 3-D wave vector's magnitude $k = |\vec{k}|$ to radius r against a spherical Bessel function $j_\ell(kr)$. This integral is essentially a Hankel transform up to details of the integration measure, which can be absorbed into the function being transformed if desired.

Many fast algorithms exist for the Hankel transform. Most exploit the change of variables of Siegman (1977) where

one sets $kr = \exp[\ln k + \ln r]$ to render the transform a convolution in $\ln k$ or $\ln r$ that can then be performed using an FFT. These algorithms are further discussed in Hamilton (2000).

However, these algorithms treat only a bijective mapping from k to r. Yet in cosmology an additional case that often arises is a 1-D integral over k involving a product of spherical Bessel functions with differing configuration space arguments, e.g. $(2\pi)^{-2}\int k^2 dk\, P(k) j_\ell(kr) j_{\ell'}(kr')$, where $P(k)$ is the function being transformed, often the power spectrum, and j_ℓ is a spherical Bessel function of order ℓ. It is not obvious how to evaluate this integral using 1-D FFTs.

In this work, we show how to reduce this computation to a sum over 1-D FFTs. Indeed, we show that the method we present for a two spherical-Bessel-function integral generalizes to integrals over an arbitrary number of products of spherical Bessel functions, and we include examples for the case of three spherical Bessel functions as well.

Our approach should substantially accelerate calculations involving these integrals, which are often a computational bottleneck. A number of examples exist: exact projection of the 3-D power spectrum to the angular power spectrum without invoking the Limber approximation (Limber 1953), computation of loop integrals for the correlation function or power spectrum in cosmological perturbation theory (PT) (Schmittfull & Vlah 2016, Slepian 2018), computation of predictions for the tree-level isotropically-averaged redshift-space 3-point correlation function (3PCF) as well as the anisotropic analog; computation of the 2PCF, 3PCF, and higher correlation functions' covariances in the Gaussian-Random-Field limit (Xu et al. 2013, Slepian & Eisenstein 2015b, Slepian & Eisenstein 2018b, Slepian, Cahn & Eisenstein in prep.).

Developing new, faster tools for these kinds of calculations is particularly vital in the context of the wealth of current and emerging large-scale structure datasets with millions to billions of objects. These datasets will offer unprecedented precision, frequently sub-percent on any given parameter, and thus higher precision predictions and weighting of the data will be critical. In particular, relevant current and upcoming efforts are the Sloan Digital Sky Survey (SDSS; e.g. Alam et al. 2017), WiggleZ (Parkinson et al. 2012), and Vipers (Scodeggio et al. 2018), as well as upcoming, such as extended Baryon Oscillation Spectroscopic Survey (eBOSS; Dawson et al. 2016), Dark Energy Spectroscopic Instrument (DESI[4]; DESI Collaboration et al. 2016) and Euclid[5] (Laureijs et al. 2011). There are also significant photometric datasets, both now available, such as Dark Energy Survey (DES[6]; The Dark Energy Survey Collaboration, 2005), and upcoming, such as Large Synoptic Survey Telescope (LSST[7]; LSST Dark Energy Science Collaboration 2012). There will also be much intensity mapping data, e.g. from CHIME[8] (Bandura et al. 2014), SKA[9] (Group et al. 2018), and HIRAX[10] (Newburgh et al. 2016) along with envi-

---

[1] This is just Bloch's theorem. Furthermore, we note that plane waves are eigenstates of momentum, the Noether's theorem conserved quantity associated with translation invariance.

[2] By Gaussian Random Field we mean that the real and imaginary parts of each Fourier-space mode are drawn from a Gaussian at each k with variance given by the power spectrum $P(k)$. This results in the complex phase's being uniformly distributed from 0 to 2π.

[3] We write "degree" of isotropy because when working in redshift-space one does not have full rotational symmetry, though one retains azimuthal symmetry about the line of sight. Furthermore, the RSD are in fact isotropic about the point from which the galaxy survey is observed, just not any other point.

[4] https://www.desi.lbl.gov/
[5] https://www.euclid-ec.org/
[6] https://www.darkenergysurvey.org/
[7] https://www.lsst.org/
[8] https://chime-experiment.ca/
[9] https://www.skatelescope.org/
[10] https://hirax.ukzn.ac.za/

sioned future experiments such as PUMA[11] (Bandura et al. 2019) and others (Ansari et al. 2018).

To fully exploit these datasets, clustering analyses will need to be faster than ever, and explore a larger space of variations, such as in the cosmological parameters. In particular, a common recent approach is to do Monte Carlo Markov Chain (MCMC) exploration over several cosmological parameters at a time, requiring a potentially substantial number of these integrals (see e.g. Cataneo et al. 2017 for discussion of how to do this with a Taylor series about a given cosmology). Being able to evaluate theory predictions quickly will enable this exploration. In particular, fast computation of PT loop integrals will allow exploiting these datasets down to smaller scales, important for constraining modified gravity models (e.g. DESI collaboration Modified Gravity whitepaper, in prep.) and galaxy biasing models (McDonald & Roy 2009, Assassi et al. 2014, Senatore 2015, Mirbabayi et al. 2015, Angulo et al. 2015; for a review, Desjacques et al. 2018).

Further, efficient analytic evaluation of covariance matrices will be important for optimal weighting of these datasets while avoiding obtaining the covariance from a large number of mocks, which is computationally expensive (Percival et al. 2014), although alternatives to reduce the cost have been proposed (Xu et al. 2013, Slepian & Eisenstein 2015b, Padmanabhan et al. 2016, O'Connell et al. 2016, Mohammed et al. 2017, Howlett & Percival 2017, Slepian & Eisenstein 2018a, Li et al. 2018). Many of these alternatives require many of the integrals we treat in the current work.

This paper is laid out as follows. In §2, we present the simplest case in which double sBF integrals of the power spectrum emerge, from multipole expansion of correlation functions evaluated with both endpoints offset from the origin. In §3 we present a number of the cosmological use cases where they enter, and focus on presenting the anisotropic 2-Point Correlation Function (2PCF) covariance in more detail in §4. In §5 we outline our technique for evaluating these integrals. §6 generalizes the technique to integrals against three or more spherical Bessel functions. In §7 we discuss a further acceleration idea generalizing recent work by Assassi et al. (2017), Grasshorn Gebhardt & Jeong (2018), and Simonović et al. (2018). We present numerical tests of these ideas in §8. We conclude in §9.

2 HOW THESE INTEGRALS EMERGE

Consider the usual transformation between the power spectrum $P(k)$ and the correlation function $\xi(r)$. Our convention will be to use a minus sign in the exponential for a 3-D inverse FT and normalize it as $d^3\vec{k}/(2\pi)^3$. We have

$$\xi(r) = \int \frac{d\Omega_r}{4\pi} \int \frac{d^3\vec{k}}{(2\pi)^3} e^{-i\vec{k}\cdot\vec{r}} P(k), \tag{1}$$

with $r \equiv |\vec{r}|$. We rewrite the plane wave in terms of spherical Bessel functions j_ℓ and spherical harmonics $Y_{\ell m}$ using the plane wave expansion (e.g. Arfken et al. 2013 equation

16.61),

$$e^{-i\vec{k}\cdot\vec{r}} = 4\pi \sum_{\ell m} (-i)^\ell j_\ell(kr) Y_{\ell m}(\hat{k}) Y_{\ell m}^*(\hat{r}). \tag{2}$$

We can integrate over $d\Omega_k$ since the power spectrum has no angular dependence. This sets $\ell m = 00$ by orthogonality of the spherical harmonics. We also perform the integral over $d\Omega_r$. Our integral becomes

$$\xi(r) = \int \frac{k^2 dk}{2\pi^2} j_0(kr) P(k). \tag{3}$$

Naively, computing this integral will scale as $N_r N_k$ if one uses N_r sample points in r and N_k sample points in k. This becomes order-$N_g{}^2$ if one uses an equal number of grid points, N_g, in each.

However, this integral can be accelerated using a 1-D FFT by setting $kr = \exp[\ln k + \ln r]$, a transformation first presented in Siegman (1977). Defining $u = \ln k$ and $v = \ln r$ yields

$$\xi(r) = \int \frac{e^{3u} du}{2\pi^2} j_0(e^{u+v}) P(e^u) = \left\{ \left[e^{3u} P(e^u) \right] \star \left[j_0(e^u) \right] \right\}(v(r)), \tag{4}$$

where "star" means convolution. The change of variables to $d\ln k$ required one extra power of e^u in the Jacobian: $k^2 dk = (e^u)^2 e^u du$ since $du = dk/k$. Our notation in the second equality reflects that the convolution is an integral of the convolvands over the dummy variable u offset from each other by the free variable v, which depends on r. By the Convolution Theorem equation (4) becomes

$$\left\{ \left[e^{3u} P(e^u) \right] \star \left[j_0(e^u) \right] \right\}(v) =$$
$$\mathrm{FT}^{-1}\left\{ \mathrm{FT}\left[e^{3u} P(e^u) \right](s) \mathrm{FT}\left[j_0(e^u) \right](s) \right\}(v), \tag{5}$$

where our notation is that the FT is an integral over the dummy variable u evaluated at the free variable s, and then the inverse FT integrates over s to be evaluated at v. Equation (5) allows use of FFTs to obtain the results at all v in $N_g \log N_g$ time. We note that this approach appears to require a one-to-one match between configuration space and Fourier space variables: there is just one k and one r, or equivalently u and v.

Now consider a correlation function offset from the origin by some vector $\vec{r}'$:

$$\xi(|\vec{r} + \vec{r}'|) = \int \frac{d^3\vec{k}}{(2\pi)^3} e^{-i\vec{k}\cdot(\vec{r} + \vec{r}')} P(k). \tag{6}$$

There are two ways to expand this into separated radial and angular integrals. The first is to treat $\vec{r} + \vec{r}'$ as a unit and use equation (3) with $r \to |\vec{r} + \vec{r}'|$:

$$\xi\left(|\vec{r} + \vec{r}'|\right) = \int \frac{k^2 dk}{2\pi^2} j_0(k|\vec{r} + \vec{r}'|) P(k). \tag{7}$$

This seems to significantly increase the computational burden because now one has three relevant parameters: r, r', and $\hat{r}\cdot\hat{r}'$, as $|\vec{r} + \vec{r}'| = \sqrt{r^2 + r'^2 + 2rr'\hat{r}\cdot\hat{r}'}$.

The second option is to factor the exponential as

$$e^{-i\vec{k}\cdot(\vec{r} + \vec{r}')} = e^{-i\vec{k}\cdot\vec{r}} e^{-i\vec{k}\cdot\vec{r}'} \tag{8}$$

and then expand each exponential using the plane wave expansion (2) twice. Doing so we obtain

$$\xi\left(|\vec{r}+\vec{r}'|\right) = (4\pi)^2 \sum_{\ell\ell'} (-i)^{\ell+\ell'} \sum_{mm'} Y_{\ell m}(\hat{r}) Y^*_{\ell'm'}(\hat{r}')$$

$$\times \int \frac{k^2 dk}{2\pi^2} j_\ell(kr) j_{\ell'}(kr') P(k) \int \frac{d\Omega_k}{4\pi} Y^*_{\ell m}(\hat{k}) Y_{\ell'm'}(\hat{k}). \tag{9}$$

Using orthogonality to perform the angular integral over $d\Omega_k$ we find

$$\xi\left(|\vec{r}+\vec{r}'|\right) = 4\pi \sum_\ell (-1)^\ell \sum_m Y_{\ell m}(\hat{r}) Y^*_{\ell m}(\hat{r}')$$

$$\times \int \frac{k^2 dk}{2\pi^2} j_\ell(kr) j_\ell(kr') P(k)$$

$$= \sum_\ell (-1)^\ell (2\ell+1) \mathscr{L}_\ell(\hat{r}\cdot\hat{r}') \int \frac{k^2 dk}{2\pi^2} j_\ell(kr) j_\ell(kr') P(k), \tag{10}$$

where $\mathscr{L}_\ell$ is a Legendre polynomial of degree ℓ, and we used the spherical harmonic addition theorem (e.g. Arfken et al. 2013 equation 16.57)

$$\mathscr{L}_\ell(\hat{r}\cdot\hat{r}') = \frac{4\pi}{2\ell+1} \sum_{m=-\ell}^{\ell} Y_{\ell m}(\hat{r}) Y^*_{\ell m}(\hat{r}') \tag{11}$$

to resum the spherical harmonics over spins m and obtain the second equality in equation (10). As expected, the result (10) depends on three parameters: the magnitudes of the two vectors r and r' and the angle between them $\hat{r}\cdot\hat{r}' \equiv \mu$. Before (equation 7) the integral was over a 3-D grid in r, r' and angle cosine $\mu = \hat{r}\cdot\hat{r}'$; now we have traded this for an infinite sum of 2-D-grid integrals onto r and r'. We note also that a given number N_μ of grid points in angle cosine would track a maximum frequency N_μ corresponding to a maximal multipole $\ell_{\max} = N_\mu$ because $\mathscr{L}_\ell \propto \cos^\ell \theta \propto \cos \ell\theta$—so the trade is a fair one. Rather than tracking N_μ angle cosines over which to integrate, we must track this number of multipole coefficients in our series for the correlation function.

Moving forward, we define the integral in equation (10) as the $\ell = \ell'$, $n = 0$ limit of the more general

$$f^n_{\ell\ell'}(r, r') = \int \frac{k^2 dk}{2\pi^2} k^n j_\ell(kr) j_{\ell'}(kr') P(k). \tag{12}$$

We see from equation (10) that the physical interpretation of these integrals is as the radial expansion coefficients in a multipole series for the correlation function shifted from the origin by $\vec{r}'$. If there is an external angular momentum in the problem, such as the quadrupolar modulations that linear-theory redshift space distortions (RSD) induce (e.g. Hamilton 1992), then an f-tensor with three sBFs may also arise, defined as

$$f^n_{\ell\ell'\ell''}(r, r', r'') = \int \frac{k^2 dk}{2\pi^2} k^n j_\ell(kr) j_{\ell'}(kr') j_{\ell''}(kr'') P(k). \tag{13}$$

The 2-D integral representation involving $j_0(k|\vec{r}+\vec{r}'|)$ (equation 7) and the infinite sum representation (equation 10) are equivalent in principle. However as examples in the following section will show, one often needs only a finite, small set of these tensors. Consequently in practice, for such applications the second, factorized representation (10) offers a considerable computational acceleration over the first, unfactored representation.

We note that multipole coefficients could also be obtained after the fact from the first approach equation (7),

but the total work would be larger than in the second approach equation (10). The first would require an $N_\mu \times N_r \times N_{r'}$ integral and then 1-D projection integrals onto $\mathscr{L}_\ell(\hat{r}\cdot\hat{r}')$ for, say, N_ℓ desired ℓ, making the total work scale as $N_\mu^2 N_r N_{r'} N_\ell$. In contrast, in the second approach one needs to do N_ℓ 2-D integrals on a grid $N_r \times N_{r'}$, for total work scaling as $N_\ell N_r N_{r'}$. The fact that the work using these two approaches to obtain multipole coefficients differs by a factor of N_μ is sensible: the advantage of the second method is that it performs all the angular integrals analytically. This of course also confers an improvement in the expected precision.

3 EXAMPLES

In this section we present a number of cosmological use-cases for the integrals treated in this paper.

3.1 Exact Angular Power Spectra

Angular power spectra of galaxy clustering or lensing measurements can be computed by projecting the 3-D matter power spectrum $P(k, z)$ onto the sky, where z is the redshift. This 3-D to 2-D projection yields a line-of-sight integral over two spherical Bessel functions, two redshift kernels, and the 3-D matter power spectrum (e.g. Assassi et al. 2017; Grasshorn Gebhardt & Jeong 2018).

Usually, these integrals are simplified using the Limber approximation (Limber 1953), but this breaks down on large scales, corresponding to low multipoles $\ell \lesssim 10 - 20$. When modeling angular statistics on such large scales one therefore needs to perform full line-of-sight integrals with two spherical Bessel functions, corresponding to the general type of integral we are evaluating in this paper. Our method at present can already be used for the first few beyond-Limber terms, where the correction is most important. In principle, it can handle the higher-ℓ corrections as well. However in practice, due to numerical challenges in canceling divergences in our splitting of the integrals, our current implementation does not extend to the $\ell \sim 10 - 20$ ultimately desirable for beyond-Limber.

3.2 Loop Corrections to the Power Spectrum

In Eulerian Standard Perturbation Theory (SPT) the leading order correction to the linear power spectrum arises from 1-loop integrals. These can be evaluated using a truncated Fourier series representation of the power spectrum (McEwen et al. 2016) or direct representation in terms of 1-D spherical Bessel integrals (Schmittfull et al. 2016). The next order in perturbation theory involves two-loop integrals over two 3-D wavevectors. While terms with at most one inverse Laplacian ∇^{-1} can again be reduced to 1-D spherical Bessel integrals, this is no longer true for more complicated terms that arise from multiple inverse Laplacians, which in Fourier space each look like $1/|\vec{k}+\vec{q}|$. One approach (Schmittfull & Vlah 2016) to evaluate these integrals is to express them as

$$\Xi^{(i+j)}_{nn'}(\vec{r}, \vec{k}) \equiv \int \frac{d^3\vec{q}}{(2\pi)^3} \frac{e^{i\vec{q}\cdot\vec{r}}}{q^{2n}|\vec{k}+\vec{q}|^{2n'}} P^i(q) P^j(|\vec{k}+\vec{q}|) \tag{14}$$

where i, j take on the values 0 or 1 and P is the linear power spectrum. These fundamental integrals can be reduced by expanding in Legendre polynomials in $\hat{r} \cdot \hat{k}$. The coefficients in this expansion are then

$$
\begin{aligned}
Q_{nn',L}^{(N)}(r,k) = 4\pi \int ds\, s^2 j_L(ks) R_{n'}^{(N)}(s) \int \frac{dq}{2\pi^2} \\
\times q^{2-2n} j_L(qs) j_L(qr) \tilde{R}^{(N)}(q),
\end{aligned}
\tag{15}
$$

where R and $\tilde{R}$ are defined in Schmittfull & Vlah (2016). The inner integral over q involves two spherical Bessel functions. Thus the acceleration scheme for these integrals presented in the current work would accelerate evaluating the 2-loop SPT power spectrum.[12]

Recent work (Slepian 2018) shows how these 2-loop PT integrals may be evaluated as a sum of serial back and forth 1-D FFTs. This work also uses decomposition into f-tensors. While the transforms there are already more or less 1-D, the 1-D transforms are done over a number of discrete integer arguments for second sBFs in each integration. Treating these integer arguments as additional free variables thus renders the 1-D transforms 2-D, and they could then be accelerated with the ideas presented in this work. This might ultimately be faster than doing serial 1-D transforms at many fixed integer values of the argument of the second sBF.

3.3 Cyclically Summing the Perturbation Theory 3PCF Model

The tree-level 3PCF in Standard Perturbation Theory involves two linear density fields and a second-order density field, and its calculation can be simplified by assuming that the second order density field sits at the origin of coordinates. A multipole expansion of the 3PCF can then be made by projecting its triangle-opening-angle dependence onto Legendre polynomials of the cosine of the angle enclosed by two sides extending from the origin, only $\ell = 0, 1$, and 2 modes are required to describe the opening-angle-dependence (Szapudi 2004, Slepian & Eisenstein 2015a).

However, the 3PCF we then observe in this basis does not necessarily place the second-order density field at the origin from which the triangle opening angle is also measured. Thus to connect the PT model with the observable 3PCF we must cyclically sum over the three possible locations for the second-order density field. Two terms in the cyclic sum will be the same because once the second-order field is moved away from the origin, there is a labeling switch symmetry.

A naive way to obtain the full cyclic sum's multipole expansion is simply to solve directly for the angles and side lengths that enter and then reproject onto Legendre polynomials. However, as shown in Slepian & Eisenstein (2017) §3, the cyclic sum's multipole moments may be more efficiently obtained if all of the angular integrations are performed an-

alytically instead. That work found that the additional moments generated by cyclic summing take the form

$$
\xi^{[L]}(r_1) \xi^{[L]}(r_3) \mathscr{L}_L(\hat{r}_1 \cdot \hat{r}_3) = \sum_\ell \zeta_\ell(r_1, r_2) \mathscr{L}_\ell(\hat{r}_1 \cdot \hat{r}_2).
\tag{16}
$$

r_1 and r_2 enclose the angle with respect to which we measure the 3PCF multipole coefficients ζ_ℓ, and the lefthand side is the simple-looking term that appears in the cyclic sum but which actually needs to be reprojected onto multipoles with respect to $\hat{r}_1 \cdot \hat{r}_2$.

We have defined

$$
\begin{aligned}
\zeta_l(r_1, r_2) = \frac{4\pi}{2L+1} \xi^{[L]}(r_1) \sum_{L_1} i^{l+L-L_1} \mathscr{C}_{LlL_1}^2 \begin{pmatrix} L & l & L_1 \\ 0 & 0 & 0 \end{pmatrix}^2 \\
\times \int \frac{k^2 dk}{2\pi^2} P(k) j_{L_1}(kr_1) j_l(kr_2),
\end{aligned}
\tag{17}
$$

where the 2×3 matrix is a Wigner 3-j symbol (Arfken et al. 2013, Varshalovich et al. 1988),

$$
\mathscr{C}_{LlL_1} = \sqrt{\frac{(2L+1)(2l+1)(2L_1+1)}{4\pi}}
\tag{18}
$$

and

$$
\xi_L(r) \equiv \int \frac{k^2 dk}{2\pi^2} j_L(kr) P(k).
\tag{19}
$$

Thus we see that accelerating the computation of the integral on the righthand side of equation (17) would render cyclically summing the PT prediction for the 3PCF more efficient. This in turn would be useful in the context of embedding a 3PCF analysis in an MCMC parameter estimation routine where model predictions must be computed many times as the cosmology is varied.

3.4 Evaluating the Gaussian Random Field Covariance of the 3PCF

Another context in which the integrals discussed in this work appear is the covariance matrix of the 3PCF. In particular, the leading-order covariance of the large-scale 3PCF stems from the Gaussian Random Field (GRF) component of the 6-point function, i.e. the disconnected piece. This point is also true for the 2PCF covariance, but we defer discussion of that to §4. This holds both for the covariance of the isotropic 3PCF and that of the anisotropic 3PCF. This latter generalizes the isotropic covariance by promoting the isotropic power spectrum to a multipole series depending on the angle to the line of sight of the different wave vectors entering the calculation.

As shown in Slepian & Eisenstein (2015b), the isotropic 3PCF covariance between 3PCF multipoles ℓ and ℓ' and tri-

---

[12] Schmittfull & Vlah (2016) also presents an alternative expansion of these integrals as an infinite sum of 1-D integrals that can be evaluated using FFTs (their equation 51). This latter expansion is actually used to evaluate their example integrals in their Figure 2, since in practice the infinite sum is relatively well-converged after a modest number of terms.

angle sides, respectively, r_1, r_2 and r'_1, r'_2, takes the form

$$
\text{Cov}_{ll'}(r_1, r_2; r'_1, r'_2) = \frac{4\pi}{V}(2l+1)(2l'+1)(-1)^{l+l'}
$$

$$
\times \int r^2 dr \sum_{l_2} (2l_2+1) \begin{pmatrix} l & l' & l_2 \\ 0 & 0 & 0 \end{pmatrix}^2
$$

$$
\times \Bigg\{ (-1)^{l_2} \xi_0(r) \Big[f_{l_2 ll'}(r; r_1, r'_1) f_{l_2 ll'}(r; r_2, r'_2)
$$

$$
+ f_{l_2 ll'}(r; r_2, r'_1) f_{l_2 ll'}(r; r_1, r'_2) \Big] + (-1)^{(l+l'+l_2)/2}
$$

$$
\times \Big[f_{ll}(r; r_1) f_{l'l'}(r; r'_1) f_{l_2 ll'}(r; r_2, r'_2)
$$

$$
+ f_{ll}(r; r_1) f_{l'l'}(r; r'_2) f_{l_2 ll'}(r; r_2, r'_1)
$$

$$
+ f_{ll}(r; r_2) f_{l'l'}(r; r'_1) f_{l_2 ll'}(r; r_1, r'_2)
$$

$$
+ f_{ll}(r; r_2) f_{l'l'}(r; r'_2) f_{l_2 ll'}(r; r_1, r'_1) \Big] \Bigg\}. \tag{20}
$$

ξ_0 is defined by setting $L = 0$ in equation (19). We note that l_2 is bounded by triangle inequalities once l and l' are fixed. The anisotropic covariance, between 3PCF coefficients at momenta l and l' and shared spin m, with triangle sides as in the isotropic case, is

$$
\text{Cov}_{l_1 l_2 m, l'_1 l'_2 m'}(r_1, r_2; r'_1, r'_2) = \frac{(4\pi)^{3/2}}{V}(-1)^{m+m'}(-i)^{l_1+l_2+l'_1+l'_2}
$$

$$
\times \int r^2 dr \sum_{l_q l_p l_k} \frac{1}{\sqrt{(2l_q+1)(2l_p+1)(2l_k+1)}}
$$

$$
\times \sum_{J_1 J_2 J_3} \mathscr{D}_{J_1 J_2 J_3} \mathscr{C}_{J_1 J_2 J_3} \begin{pmatrix} J_1 & J_2 & J_3 \\ 0 & 0 & 0 \end{pmatrix}
$$

$$
\times \Bigg\{ \xi_{l_k}(r) \Big[w_1 f_{J_1 l_1 l'_1}^{l_q}(r; r_1, r'_1) f_{J_2 l_2 l'_2}^{l_p}(r; r_2, r'_2)
$$

$$
+ w_2 f_{J_1 l_1 l'_2}^{l_q}(r; r_1, r'_2) f_{J_2 l_2 l'_1}^{l_p}(r; r_2, r'_1) \Big] + \begin{pmatrix} J_1 & J_2 & J_3 \\ S_1 & S_2 & S_3 \end{pmatrix}
$$

$$
\times \Bigg\{ f_{J_1 l_1}^{l_q}(r; r_1) \Big[w_3 f_{J_2 l_2 l'_2}^{l_p}(r; r_2, r'_2) f_{J_3 l'_1}^{l_k}(r; r'_1) \delta_{S_1-m, S_3-m'}^{K}
$$

$$
+ w_4 f_{J_2 l_2 l'_1}^{l_p}(r; r_2, r'_1) f_{J_3 l'_2}^{l_k}(r; r'_2) \delta_{S_1-m, S_3 m'}^{K} \Big]
$$

$$
+ f_{J_2 l_2}^{l_p}(r; r_2) \Big[w_5 f_{J_1 l_1 l'_2}^{l_q}(r; r_1, r'_2) f_{J_3 l'_1}^{l_k}(r; r'_1) \delta_{S_2 m, S_3-m'}^{K}
$$

$$
+ w_6 f_{J_1 l_1 l'_1}^{l_q}(r; r_1, r'_1) f_{J_3 l'_2}^{l_k}(r; r'_2) \delta_{S_2 m, S_3 m'}^{K} \Big] \Bigg\} \Bigg\}. \tag{21}
$$

$\mathscr{C}$ is defined in equation (18), ξ_{l_k} is defined in equation (19), $\mathscr{D}_{J_1 J_2 J_3} \equiv i^{J_1+J_2+J_3}$, δ^{K} is a Kronecker delta, unity if its arguments are equal and zero otherwise. The w_i are angular momentum coupling weights involving integrals of three and four spherical harmonics, i.e. Gaunt integrals and products of Gaunt integrals, or equivalently, Wigner 3j-symbols.

The main cost of evaluating these covariance expressions is obtaining the f-tensors (defined in equation 12; see also equation 45 of Slepian & Eisenstein 2018a), and again one might wish to compute them many times if the 3PCF were used as part of an MCMC cosmological parameter analysis. A fast means of computing them is thus highly enabling for a 3PCF analysis where cosmological parameters are varied.

3.5 Gaussian Random Field Contribution to N-Point Correlation Functions

In the future it is likely that we will measure higher-point statistics of galaxy clustering. All of the even correlation functions (e.g. 4PCF, 6PCF, 8PCF) will have contributions from the Gaussian Random Field component of the overall density field. However, this component simply duplicates information already available in the 2PCF. Thus removing it before fitting models to the higher-order (even) correlation functions is desirable. It must therefore be calculated.

Several recent works have discussed measurement of higher N-point functions. Sabiu et al. (2019) uses a graph spatial database to obtain NPCFs, and presents the first measurement of the 4PCF of Luminous Red Galaxies in the Sloan Digital Sky Survey Baryon Oscillation Spectroscopic Survey (SDSS BOSS) CMASS sample. Tomlinson et al. (2019) outlines how to measure higher-order polyspectra, the Fourier-space analogs of the NPCFs, averaging over internal angles of the Fourier-space polyhedron. Slepian, Cahn & Eisenstein (in prep.) outlines how to measure higher-order correlation functions in the basis of rotation-invariant combinations of spherical harmonic coefficients, building on Slepian & Eisenstein (2015b), Slepian & Eisenstein (2016a), Slepian & Eisenstein (2018b). This will then require removing remove the contributions of the GRF piece in the harmonic basis. Here we sketch results in configuration space as these give rise to the double-Bessel transforms of the power spectrum that are our focus in this work.

For an even N-point function, the Gaussian random field contribution will be the cyclic sum of products of 2-point functions with all possible vector argument differences (shown in Slepian, Cahn & Eisenstein in prep.). For instance, for the 4PCF, denoted $\zeta^{(4)}$, we have

$$
\zeta_{\text{GRF}}^{(4)}(\vec{r}_1, \vec{r}_2, \vec{r}_3, \vec{r}_4) = \xi(|\vec{r}_2 - \vec{r}_1|)\xi(|\vec{r}_4 - \vec{r}_3|) + \text{cyc.} \tag{22}
$$

For the 6PCF, we have

$$
\zeta_{\text{GRF}}^{(6)}(\vec{r}_1, \vec{r}_2, \vec{r}_3, \vec{r}_4, \vec{r}_5, \vec{r}_6) =
$$
$$
\xi(|\vec{r}_2 - \vec{r}_1|)\xi(|\vec{r}_4 - \vec{r}_3|)\xi(|\vec{r}_6 - \vec{r}_5|) + \text{cyc.} \tag{23}
$$

To compute the GRF piece of an even NPCF projected onto the basis of spherical harmonics, we may do so by writing each correlation function as an inverse Fourier Transform of the power spectrum, and then projecting the plane waves therein onto spherical harmonics using the plane wave expansion and then orthogonality. This will leave f-tensors of exactly the form given in equation (12). Note that we may indeed have $\ell \neq \ell'$. The only constraint on each term in the cyclic sum for an even NPCF is that the parity of the summed angular momenta in a given term be even. This is required by rotation-invariance: each spherical harmonic picks up $(-1)^\ell$ as it rotates, we need the sum of the ℓ to be even.

We also note that in analysing the results of an NPCF computation, covariance matrices will be required for optimally weighting the data. These quickly become enormous matrices and for generic NPCFs would require a prohibitive number of mock catalogs if one wished to estimate them from mocks. Instead, Xu et al. (2013), Slepian & Eisenstein (2015b), Slepian et al. (2016), Slepian & Eisenstein (2018b), and Slepian, Cahn, & Eisenstein (in prep.) advocate using a template covariance matrix; the last work referenced ex-

tends this to general NPCFs and shows that the template can be cast solely in terms of the integrals (12). Thus speeding up the evaluation of these will aid computation of the NPCF covariance matrix templates.

4 MAIN EXAMPLE: GAUSSIAN RANDOM FIELD 2PCF COVARIANCE MATRIX

The main numerical example we will present in this work is the covariance matrix of the anisotropic 2PCF multipoles. It can be shown (e.g. Xu et al. 2013) that this matrix is given by

$$\text{Cov}\left(\xi_\ell(r), \xi_{\ell'}(r')\right) = \int \frac{k^2 dk}{2\pi^2} \, j_\ell(kr) j_{\ell'}(kr') P^2(k) \tag{24}$$

with P the linear power spectrum. If one wishes to include shot noise one can promote P to $P + 1/n$ with n the survey number density, and often one also uses a "template" power spectrum where the BAO features have been smoothed, usually by a Gaussian kernel of width $\sim 7 - 8$ Mpc, which is the root-mean-square displacement in the Zel'dovich approximation. One also generally smoothes the whole power spectrum by a Gaussian of width 1 Mpc to avoid ringing due to the truncation of the Fourier-space grid. Generally the radial variables in the covariance are binned to match the fact that the 2PCF itself is measured on radial bins.

Here we will simply do the full, unbinned calculation, since certainly if that is accurate, the binned version will end up more so, as binning effectively smoothes the spherical Bessel functions, making the numerical accuracy requirements for their integration less stringent. One could also imagine replacing the spherical Bessel functions with their binned analogs before performing the integration over k, as for instance Xu et al. (2013) does (their equations 8-11). That would still be amenable to the techniques outlined in this work, since due to their recursion relations, the integral of a spherical Bessel functions yields a sum of two others, weighted by powers of the argument (e.g. Olver et al. (2010) 10.51.1 and 10.51.2 can be manipulated to show this). Consequently even after "analytic" binning one still has integrals over k of the type we focus on performing here.

The monopole and quadrupole have been measured by BOSS, and DESI will measure them to high precision. DESI will not achieve very high precision ($\sim 10\%$) on the hexadecapole $\ell = 4$ (DESI Collaboration et al. 2016), so we further focus our numerical work here down to the monopole ($\ell = 0$) and quadrupole ($\ell = 2$) and their cross-covariances as well.

In particular, the use case numerical-results we present (i.e. those where we have used the true power spectrum, rather than some test power spectrum for simply verifying the accuracy of our integrations) will be doing the integrals for $\text{Cov}(\zeta_0, \zeta_0)$ and $\text{Cov}(\zeta_2, \zeta_2)$. We do not perform tests for $\text{Cov}(\zeta_2, \zeta_0)$ because if we are able to get the other two matrix elements right, we expect that the numerical performance will be sufficient for this cross term.

5 ROTATION METHOD: FROM 2-D TO 1-D

5.1 Basic Idea

We now show how to reduce $f_{\ell\ell'}$ (equation 12) to a sum of 1-D integrals. All spherical Bessel functions $j_\ell(x)$ are of the form

$$j_\ell(x) = \frac{C_\ell(x)}{x^\ell} \cos x + \frac{S_\ell(x)}{x^{\ell+1}} \sin x, \tag{25}$$

where $C_\ell(x)$ and $S_\ell(x)$ are polynomials with a finite, small number of terms ("C" denotes the polynomial multiplying the cosine, "S" that multiplying the sine). Writing further that

$$C_\ell(x) = \sum_{n=0}^{n_{max}} c_{\ell n} x^n \tag{26}$$

and

$$S_\ell(x) = \sum_{m=0}^{m_{max}} s_{\ell m} x^m, \tag{27}$$

where $n_{max} \leq \ell - 1$ for odd ℓ and $n_{max} \leq \ell - 2$ for even ℓ and $m_{max} \leq \ell - 1$ for odd ℓ and $m_{max} \leq \ell$ for even ℓ. Exact forms for these relations are presented in Appendix A. We then see that a product of two spherical Bessel functions with arguments ka and kb will have the form

$$j_\ell(ka) j_{\ell'}(kb) = \tag{28}$$
$$\cos ka \cos kb \left(k^{-\ell-\ell'} a^{-\ell} b^{-\ell'} \right) \sum_{nn'} c_{\ell n} c_{\ell' n'} k^{n+n'} a^n b^{n'}$$
$$+ \cos ka \sin kb \left(k^{-\ell-\ell'-1} a^{-\ell} b^{-\ell'-1} \right) \sum_{nm'} c_{\ell n} s_{\ell' m'} k^{n+m'} a^n b^{m'}$$
$$+ \sin ka \cos kb \left(k^{-\ell-1-\ell'} a^{-\ell-1} b^{-\ell'} \right) \sum_{mn'} s_{\ell m} c_{\ell' n'} k^{m+n'} a^m b^{n'}$$
$$+ \sin ka \sin kb \left(k^{-\ell-\ell'-2} a^{-\ell-1} b^{-\ell'-1} \right) \sum_{mm'} s_{\ell m} s_{\ell' m'} k^{m+m'} a^m b^{m'}.$$

$c_{\ell n}$, $c_{\ell' n'}$, $s_{\ell m}$, and $s_{\ell' m'}$ are constants; the argument b always matches to primed indices. It is clear that the powers of a and b can be taken outside any integrals over k. We note that since all spherical Bessel functions have asymptotic behavior $j_\ell(x) \to 1/x$ for large x, their product as above must fall as $1/k^2$. This can be directly verified case by case (product of even and odd, two even, or two odd spherical Bessel functions) and term-by-term above using the maximal values of n, m, n' and m' noted. Importantly, this holds individually for each of the four terms in the overall sum above. This large-k behavior cancels the k^2 in the integration measure, and assures that the integrals $f_{\ell\ell'}$ will not be UV-divergent as long as $P(k) \to 0$ as $k \to \infty$.

For small values of their argument, $j_\ell(x) \to x^\ell$ so their product as above scales as $k^{\ell+\ell'}$. As long as the power spectrum satisfies $P(k) \propto k^n$ as $k \to 0$ with $n \geq -(\ell + \ell' + 2)$ the integrals $f_{\ell\ell'}$ will not be IR-divergent.

However, we note that the small-argument scaling of the spherical Bessel functions relies on cancellation of terms in a Taylor series, and that the splitting above, into an overall sum over four terms, does not retain this cancellation on a term-by-term basis. For instance, the last term above will scale as $k^{-\ell-\ell'}$ as $k \to 0$. Consequently, if each of the four terms above is integrated individually against the power spectrum, we are not guaranteed to avoid IR-divergence. However, numerical integration over a finite range effectively regularizes this divergence, as we discuss further in §8.
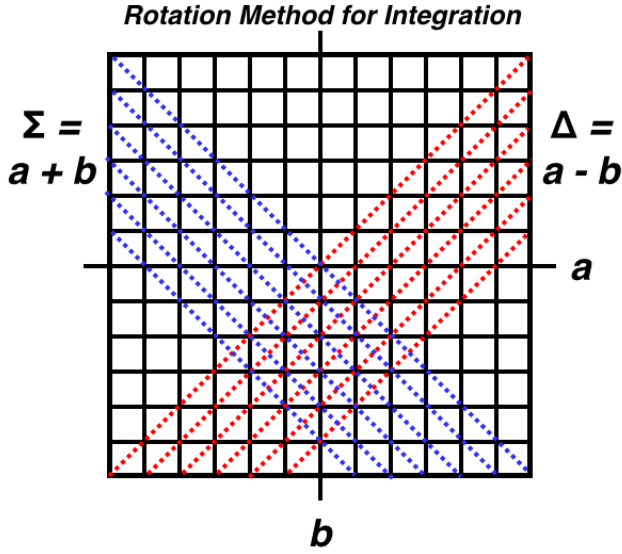
Figure 1. Here we illustrate how the "rotation" method works, for the case of a two-sBF integral. Σ and Δ correspond to rotations of 45 degrees, which essentially decouple the 2-D integrals over (a,b) into sums and differences of 1-D integrals over Δ and Σ. This decoupling is analogous to diagonalizing a matrix.

Overall, we see that if $\sin ka \cos kb$ and the other relevant combinations of trigonometric functions entering can be factorized so as to eliminate the a and b dependence from the k integral, we will have fully reduced the seemingly 2-D problem of computing $f_{\ell\ell'}$ to a sum over 1-D integrals.

Defining $\Delta = a - b$ and $\Sigma = a + b$ and employing the identities

$$\cos ka \cos kb = \frac{1}{2}\left[\cos k\Delta + \cos k\Sigma\right]$$

$$\cos ka \sin kb = \frac{1}{2}\left[\sin k\Sigma - \sin k\Delta\right]$$

$$\sin ka \sin kb = \frac{1}{2}\left[\cos k\Delta - \cos k\Sigma\right] \tag{29}$$

we see that each of the four terms in equation (28) will lead to a sum over integrals that each look like

$$I_C^q(u) = \int \frac{k^2 dk}{2\pi^2} P(k) k^q \cos ku \tag{30}$$

and

$$I_S^p(u) = \int \frac{k^2 dk}{2\pi^2} P(k) k^p \sin ku \tag{31}$$

with $u \to \Sigma$ or Δ. I is for "integral", C for "cosine", S for "sine", and the superscript p indicates the power of k weighting the integrand.

From these variables we can see that writing the products of trigonometric functions as a sum of trigonometric functions effectively performed a 2-D, 45 degree rotation from the a,b to the Σ, Δ plane. This is shown schematically in Figure 1.

These integrals are simply the Fourier cosine and sine transforms of $P(k) k^{2+q}$ and $P(k) k^{2+p}$ respectively, and are easily evaluated by a standard FFT package (for a linearly-spaced grid) or FFTLog package (for a logarithmically-spaced grid; e.g. Hamilton 2000).

5.2 Examples and Automatic Generation of the Required Forms

The complexity required to make our notation general may obscure that for reasonably small ℓ and ℓ', explicit computation shows that the required number of terms is small. For instance, for $\ell = 0 = \ell'$, we have

$$j_0(ka) j_0(kb) = \frac{1}{2k^2 ab}\left\{\cos k(a-b) - \cos k(a+b)\right\} \tag{32}$$

so that

$$f_{00}(a,b) = \frac{1}{ab}\left\{I_C^{-2}(\Delta) - I_C^{-2}(\Sigma)\right\}. \tag{33}$$

Only one fundamental integral is required for this simple case above, as the integral once obtained may be evaluated at the required arguments Σ and Δ as long as one has computed it over a large enough range. Furthermore, we emphasize that if one uses the same grid in a and b, of length N_a, there are not N_a^2 unique values of $a \pm b$, but rather $3N_a$; we can subtract all elements b from the $a = 0$ element, generating a negative $a - b$ grid of length N_a, and we can add all elements of the b grid to the maximal a element, generating an additional positive grid from $a_{\max}$ to $2a_{\max}$. But we need not evaluate the integrals at all $3N_a$ of these points; rather one can use the parity properties of the Fourier sine and cosine transform to avoid evaluating on the negative portion of the net grid, reducing the work to $2N_a$. In our numerical work we did not implement this latter trick to avoid the need for carefully tracking signs, but adding it would doubtless further accelerate the calculation, likely by roughly 30% as it takes the work from $3N_a$ to $2N_a$.

For $\ell = 1 = \ell'$, we have

$$f_{11}(a,b) = \frac{1}{2ab}\left\{\frac{1}{ab}I_C^{-4}(\Delta) + I_C^{-2}(\Delta) - \frac{1}{ab}I_C^{-4}(\Sigma) + I_C^{-2}(\Sigma)\right.$$

$$\left. + \frac{1}{b}I_S^{-3}(\Delta) - \frac{1}{a}I_S^{-3}(\Delta) - \frac{1}{b}I_S^{-3}(\Sigma) - \frac{1}{a}I_S^{-3}(\Sigma)\right\}. \tag{34}$$

Here, only three fundamental integrals are required: I_C^{-4}, I_C^{-2}, and I_S^{-3}. For $\ell = 2 = \ell'$ we find

$$f_{22}(a,b) = \frac{9}{2a^3b^3}\left(I_C^{-6}(\Delta) - I_C^{-6}(\Sigma)\right) -$$

$$\frac{1}{2}\left(\frac{3}{ab^3} - \frac{9}{a^2b^2} + \frac{3}{a^3b}\right)I_C^{-4}(\Delta) + \frac{1}{2}\left(\frac{3}{ab^3} + \frac{9}{a^2b^2} + \frac{3}{a^3b}\right)I_C^{-4}(\Sigma)$$

$$+ \frac{1}{2ab}\left(I_C^{-2}(\Delta) - I_C^{-2}(\Sigma)\right) + \frac{9}{2}\left(\frac{1}{a^2b^3} - \frac{1}{a^3b^2}\right)I_S^{-5}(\Delta) -$$

$$\frac{9}{2}\left(\frac{1}{a^2b^3} + \frac{1}{a^3b^2}\right)I_S^{-5}(\Sigma) + \frac{3}{2}\left(\frac{1}{ab^2} - \frac{1}{a^2b}\right)I_S^{-3}(\Delta)$$

$$+ \frac{3}{2}\left(\frac{1}{ab^2} + \frac{1}{a^2b}\right)I_S^{-3}(\Sigma). \tag{35}$$

Although the expression is long, only five fundamental integrals are required: I_C^{-6}, I_C^{-4}, I_C^{-2}, I_S^{-5}, and I_S^{-3}.

For an "off-diagonal" example, i.e. with $\ell \neq \ell'$, we consider

$$f_{10}(a,b) = \frac{1}{2a^2b}\left\{I_C^{-3}(\Delta) - I_C^{-3}(\Sigma) + aI_S^{-2}(\Delta) - aI_S^{-2}(\Sigma)\right\}. \tag{36}$$

In this off-diagonal example the difference in indices ($\ell = 1, \ell' = 0$) breaks the switch symmetry between a and b, so as we might expect the final expression is not symmetric under $a \leftrightarrow b$. We also highlight that just two fundamental integrals are required.

Another example, relevant for the cross covariance of the monopole and quadrupole of the anisotropic 2PCF, is

$$f_{02}(a,b) = \frac{1}{2ab} \left\{ \frac{3}{b^2} \left(I_C^{-4}(\Delta) - I_C^{-4}(\Sigma) \right) - \left(I_C^{-2}(\Delta) + I_C^{-2}(\Sigma) \right) \right.$$
$$\left. - \frac{3}{b} \left(I_S^{-3}(\Delta) + I_S^{-3}(\Sigma) \right) \right\}. \quad (37)$$

Above only three fundamental integrals are required.

A third example is

$$f_{13}(a,b) = \frac{15}{2a^2b^4} \left[I_C^{-6}(\Delta) + I_C^{-6}(\Sigma) \right]$$
$$+ \left(\frac{15}{2ab^3} - \frac{3}{a^2b^2} \right) \left[I_C^{-4}(\Delta) + I_C^{-4}(\Sigma) \right]$$
$$- \frac{1}{2ab} \left[I_C^{-2}(\Delta) + I_C^{-2}(\Sigma) \right]$$
$$+ \frac{15}{2} \left(\frac{1}{ab^4} - \frac{1}{a^2b^3} \right) \left[I_S^{-5}(\Delta) + I_S^{-5}(\Sigma) \right]$$
$$- \left(\frac{3}{ab^2} - \frac{1}{2a^2b} \right) I_S^{-3}(\Delta) + \left(\frac{3}{ab^2} + \frac{1}{2a^2b} \right) I_S^{-3}(\Sigma). \quad (38)$$

Here only five fundamental integrals are required.

In general these expressions can be easily generated using MATHEMATICA[13] with the command

TrigReduce[FunctionExpand[SphericalBesselJ[l, kr]
$\times$ SphericalBesselJ[l', kr']]]. (39)

As noted earlier, a general form for explicit expressions for the coefficients for arbitrary ℓ and ℓ' (i.e. the two-sBF case) is given in Appendix A. We also investigated using PYTHON's *sympy* symbolic manipulation package to generate these forms. However for the relatively modest-ℓ cases we focus on here this was not necessary, so we defer full development of that code to future work.

6 GENERALIZATION TO THREE OR MORE SPHERICAL BESSEL FUNCTIONS

It is clear that all we require to generalize the approach of §5 to integrals of three or more spherical Bessel functions is the analogous trigonometric identities for converting products to sums with three or more trigonometric functions. Rewriting the trigonometric functions as complex exponentials, multiplying them, refactoring into combinations that depend on $a \pm b \pm c$, and resumming to trigonometric functions yields these integrals. MATHEMATICA can again be used to obtain them quickly with a command analogous to that in equation (39).

We note that the number of identities needed is not large because one can always switch $a \leftrightarrow b$, $b \leftrightarrow c$, etc. so that only the number of cosines and sines entering the product matters, not which argument is in which function. For three spherical Bessel functions, one therefore needs only four identities, and for products of four spherical Bessel functions, five identities. In general the number of combinations is simply the number of possible factors of sine that can appear, which runs from zero up to the number of spherical Bessel functions being considered.

[13] We make some example MATHEMATICA code available at https://github.com/eelregit/sbf_rotation

The simplest example for three spherical Bessel functions is

$$f_{000}(a,b,c) = \frac{1}{4abc} \left\{ -I_S^{-3}(\chi_{--}) + I_S^{-3}(\chi_{+-}) \right.$$
$$\left. + I_S^{-3}(\chi_{-+}) - I_S^{-3}(\chi_{++}) \right\} \quad (40)$$

with

$$\chi_{\pm\pm} = a \pm b \pm c \quad (41)$$

the generalization of Δ and Σ to three variables instead of two. The ith subscripted sign denotes the sign preceding the $(i+1)^{th}$ variable.

We emphasize that the possible accelerations presented in the next section (§7) will also apply to integrals against products of three or more spherical Bessel functions, since after use of the trigonometric identities we will again only have 1-D integrals of sine and cosine against the power spectrum.

7 POSSIBLE ACCELERATIONS

We now present a complementary technique for the particular case of an integral of two spherical Bessel functions against the power spectrum, similar to the idea presented in Assassi et al. (2017) and Grasshorn Gebhardt & Jeong (2018). If the power spectrum can be expanded into a sum over (possibly complex) power laws as

$$P(k) = \sum_n c_n k^n, \quad (42)$$

then we have

$$f_{\ell\ell'}(a,b) = \sum_n c_n \int \frac{k^2 dk}{2\pi^2} k^n j_\ell(ka) j_{\ell'}(kb). \quad (43)$$

Using the identity that

$$j_\ell(x) = \sqrt{\frac{\pi}{2x}} J_{\ell+1/2}(x), \quad (44)$$

where $J_{\ell+1/2}$ is a Bessel function, we find

$$f_{\ell\ell'}(a,b) = \frac{1}{4\pi\sqrt{ab}} \sum_n c_n \int k dk \, k^n J_{\ell+1/2}(ka) J_{\ell'+1/2}(kb). \quad (45)$$

Using Gradshteyn et al. (2007) 6.574.1-3, the integral (45) can be evaluated explicitly in terms of Gamma functions Γ and hypergeometric functions F. We have the restrictions that $\ell + \ell' + 2 > -n$ and $n < 1$. The argument of the hypergeometric function depends on the ratio (a/b) or (b/a), whichever is less than unity (6.574.1 for $a < b$, 6.574.3 for $b < a$, 6.574.2 for $a = b$). For grids in a and b of length N_a, we will thus require N_a^2 evaluations of the hypergeometric function. However fast implementations of these functions exist and this should not be a significant computational burden. This approach was used for the case where $\ell = \ell'$ in Assassi et al. (2017), with further discussion in Grasshorn Gebhardt & Jeong (2018). The latter focuses on a stable, accurate, and fast method for evaluating the required hypergeometric functions.

We can use this approach to evaluate an integral over

three spherical Bessel functions as well, a problem not considered in previous literature. With the power spectrum again expanded as in equation (43) we have

$$f_{\ell\ell'\ell''}(a,b,c) = \frac{1}{2^{5/2}\pi^{1/2}abc}$$
$$\times \sum_n c_n \int k\,dk\, k^n J_{\ell+1/2}(ka)J_{\ell'+1/2}(kb)J_{\ell''+1/2}(kc) \quad (46)$$

which can be evaluated in terms of an Appell hypergeometric function using Gradshteyn et al. (2007) 6.578.1.

In general, the approach we have outlined above can be extended to obtain integral-free forms for an integral of the power spectrum against an arbitrary number of spherical Bessel functions. This can be done using the technique outlined in Fabrikant (2003) and Fabrikant (2013) for computing integrals of an arbitrary number of spherical Bessel functions against power laws. For instance Fabrikant (2013) equations (16) and (19) can be combined to provide the result for the case of four spherical Bessel functions. Fonseca de la Bella et al. (2017) provides a publicly-available MATHEMATICA notebook implementing Fabrikant's procedure for three spherical Bessel functions.[14]

8 NUMERICAL WORK

Here we briefly outline our numerical tests of these ideas. Along with this paper we make publicly available a simple PYTHON code implementing the rotation method (but with direct integration, not FFTLog) and the naive method for comparison.[15] This version uses only simple functions such as sine and cosine and Romberg integration, so it is backwards-compatible with all PYTHON versions and might be well-suited for integration into a larger package that focuses on back-compatibility, such as COLOSSUS (Diemer 2018). We also make available through `mcfit` an implementation of the rotation method that further exploits FFTLog to render the integrals even faster, as discussed at the end of §5.1.[16] We recommend this latter be used for "production" analyses, with the former mainly of use for understanding the ideas here as well as offering a simple implementation that requires no external libraries or packages save Romberg integration.

For most of our numerical work, we focused on the cases $j_0(ka)j_0(kb)$ ("00"), $j_1(ka)j_1(kb)$ ("11"), $j_2(ka)j_2(kb)$ ("22"), and $j_0(ka)j_0(kb)j_0(kc)$ ("000"), on grids where a, b, and if relevant c were linearly spaced from $0-100$ in steps of unity. The 00 and 22 cases are relevant for the anisotropic 2PCF covariance matrix. We did the 11 case for completeness and the 000 case to show how our method extends to triple-sBF integrals. The anisotropic 2PCF covariance matrix can also have cross terms (e.g. 02); we do not investigate these numerically because the issues should be just the same as for the

auto terms. In particular, 02 will have less stringent numerical requirements than 22 because the divergences we have to track go up with higher total ℓ. Finally, we performed tests using a case where the answer is exactly known analytically over a much larger range in a and b, from 10^{-4} to 10^2 in each.

8.1 Overview

We compared three approaches. For this section only, to make our plot titles complete yet concise, we adopt a shorter notation for our integrals than the f-tensor notation of earlier sections. We only consider a weight of k^2, so we do not need the upper index of the f tensors. However, we consider integrals of both P and P^2, so we need an index for that as well as two or three for the orders of each sBF integrated over. Finally, we suppress the arguments of the integral since that will be evident from the plot axis titles. We thus use the notation

$$I_{\ell\ell'}^{[n]} \equiv \int \frac{k^2 dk}{2\pi^2}\, j_\ell(ka)j_{\ell'}(kb)P^n(k). \quad (47)$$

We will also always weight our results by $(ab/100^2)^2$ as this is the weight that each pixel of the result would have on a spherical shell; this is relevant if one were to further integrate these results onto spherical bins.

8.1.1 Naive approach: $N_g{}^3$

First, we evaluated these integrals using a naive approach scaling as $N_a N_b N_k$. Here, $N_a = N_b$ is the number of grid points in a and b, and N_k is the number of points in k used for the integration. The scaling is $N_a N_b N_k$ because at each point in the (a,b) plane, we need to evaluate the integrand at each of the N_k points in k and then sum (or use some other integration scheme of choice). As a shorthand we will sometimes refer to this as the $N_g{}^3$ approach, which would be the case if one used equal grids in a, b, and k of length N_g each. In actual practice, for us $N_k \gg N_a$.

For the case of $j_0(ka)j_0(kb)$, the result was not sensitive to whether we used e.g. a Romberg integration method or a simple sum over the sample points. For the other cases (higher-ℓ), we used Romberg as it seemed a simple sum was not sufficiently precise. Furthermore, we examined both fixed-grid and adaptive Romberg. The fixed grid tests were all on a linearly-spaced grid of 4097 points from $k = 2 \times 10^{-4}$ to 2 h/Mpc. We used 4097 points because the PYTHON (`scipy`) implementation of Romberg integration we used required a grid of length $2^p + 1$, with p a natural number. For the adaptive Romberg tests we explored several different maximal k; we discuss this in greater detail for in the sections on the specific tests where it became relevant.

8.1.2 Rotation method with direct integration: $N_g{}^2$

Our second implementation was the rotation method but with direct integration. It scales as $N_a N_k$, and as a shorthand we will sometimes refer to it as the $N_g{}^2$ approach, which would be the case if one used equal grids in a, b, and k. In this second implementation, we simply employed Romberg integration to obtain each of the required 2-D integrals from k to all Δ or all Σ (e.g. equations 30 and 31).

This approach scales as $N_\Delta N_k$ (or $N_\Sigma N_k$), and $N_\Delta = N_\Sigma = 2N_a$. We explored Romberg both on a fixed *a priori* grid and with adaptive sampling. The latter required defining an interpolating function for the integrand and allowing the integration routine to call it. This latter approach is slower than simply using a fixed grid, as each call to the interpolant adds some time. However the adaptive method is more accurate. The fixed-grid Romberg was again on a linearly-spaced grid with the same parameters as in §8.1.1.

We compared adaptive-to-adaptive for the naive and rotation methods, and within the "adaptive" tests we also explored how much the maximal number of sub-divisions we permit the Romberg algorithm to make impacts the comparison.

To attempt to reduce the dynamic range of the integrals before adding them up and canceling regularized-divergent terms,[17] we also explored integrating on shells in k, i.e. evaluating each 1-D integral on a set of bins in k. We could then add up all of the 1-D integrals on each bin, and then finally add over bins. The aim of this scheme was to reduce the dynamic range one must track before getting to cancellations. We hoped that the cancellations on a given bin would place less stringent demands on the numerical precision. This point probably does hold true, but unfortunately an integral that is performed by being split into bins, and then doing a Romberg integration on each bin, is no longer globally Romberg. This is because Romberg weights depend on the whole range being integrated over. Consequently the Romberg weights set up over just one bin do not match the Romberg weights one would use on integration points in that bin if one were performing a Romberg integration on the whole range covered by all the bins. The loss of precision from failure to remain globally Romberg turned out to outweigh the possible gain due to performing the cancellations bin by bin. We therefore summarize this line of inquiry as a cautionary note only, but do not show detailed results from it.

8.1.3 Rotation method with FFTLog-integration

Finally, we produced an implementation of the rotation method using FFTLog (e.g. Hamilton 2000) to obtain the integrals we required. This approach scales as $N_k \log N_k$ (and requires that $N_k = N_\Sigma = N_\Delta$), and so is more efficient than the rotation method implementation with direct integration. It also has the advantage that it naturally handles logarithmically-spaced inputs, allowing much more dynamic range in e.g. the power spectrum, and matching the typical output format of linear Boltzmann solvers such as CAMB (Lewis et al. 2000).

8.2 Brief Summary

We now briefly summarize our numerical test results, focusing on precision and timing. For readers who simply want to apply our method to the low-ℓ cases we have worked out, this subsection should be enough. For readers interested in

the details of our numerical tests, two subsections following this one provide more extensive discussion (§8.3 and §8.4).

8.2.1 Precision

Overall, we were able to obtain percent to sub-percent agreement between the results of the 3 methods. For most applications this should be sufficient. Doubtless for further, application-dependent precision needs, one could refine the details of the integration approaches and parameters, e.g. using smoothing, using a finer k-grid, requiring higher precision in the integration, using a different k-range, using different numerical types. For our *numerical* test cases (power spectrum, power spectrum squared), we caution that if one wished to push beyond the precision found here, it would as a first step be worth checking that the fiducial case of the naive, $N_g{}^3$ implementation even is itself accurate to that level. Of course this question is well-posed in our analytic test case, where the exact answer is known. Finally, testing the $N_g{}^2$ method against the $N_g \log N_g$ method required some iteration to sufficiently match the dynamic range accessible to the latter with the former. We found that $k_{\max} \simeq 8\,h/\text{Mpc}$ was enough to obtain very good agreement over the range of a and b investigated for P, and $k_{\max} \sim 4\,h/\text{Mpc}$ for P^2. As further detailed in §8.4, going to higher $k_{\max}$ in the $N_g{}^3$ or $N_g{}^2$ methods actually hurt the agreement with the $N_g \log N_g$ method, likely due to details of how PYTHON's Romberg integration routine handles the high-k oscillations of the sines and cosines.

We further note that the areas of largest fractional disagreement between methods tended to be those where the integrals had smallest absolute value. It is very easy to get a numerical calculation wrong with a high fractional error if the value being computed itself flirts with the bounds of machine precision. In turn, these small absolute values of the integral typically correspond to regions where either a or b (or both) are very small. Since for many applications the integrals would be binned in a and b (in 3-D) after computation, the spherical-shell weights of $a^2 b^2$ would greatly suppress the importance of any region of the integral our algorithm gets wrong. So we do not believe this to be a significant limitation in practice.

The main source of disagreement between the naive method and the rotation method seemed to be the large dynamic range of the 1-D integrands combined with the need for rather precise cancellation of a number of divergent such integrals to a finite sum. When one uses the rotation method, one has several 1-D integrals that have rather high powers of k in the denominator (generically, they can be at most $\sim k^{-\ell-\ell'}$). Thus even a small integration interval in k is converted into a large dynamic range of the integrand. We considered using a higher-precision numerical type (through PYTHON's `mpmath` library). However there are not many integration methods available in PYTHON for this type, and type conversion to it would increase runtimes as well. Consequently we deferred detailed investigation of its use. For the test cases we explored, the precision achieved already seemed sufficient for practical use.

The overall comparisons for the precision and accuracy of the different methods can be read off from our Figures, which show the results with the naive method and then the differences between the $N_g \log N_g$ and $N_g{}^2$ methods; since

---

[17] Some of the integrals do in fact diverge if taken from $k = 0 \to \infty$, but of course numerically integrating them over a finite range regularizes this divergence.

the $N_g{}^3$ and $N_g{}^2$ methods typically agreed much better than $N_g \log N_g$ and $N_g{}^2$, the $N_g \log N_g$ differences from $N_g{}^2$ can also be taken to indicate those between $N_g \log N_g$ and $N_g{}^3$. Figures 2 and 3 show the value and these differences for $I_{00}^{[1]}$; Figures 4 and 5 show the value and these differences for $I_{00}^{[2]}$. Figures 6 and 7 show the value and these differences for $I_{11}^{[1]}$; Figures 8 and 9 show the value and these differences for $I_{11}^{[2]}$. Finally, Figures 10 and 11 show the value and differences for $I_{000}^{[1]}$. We do not show plots for the $j_2 j_2$ integrals because our results for them are extremely similar to those for the j_0 and j_1 cases.

8.2.2 Timings

Regarding timings, we performed repeated tests of the $N_g{}^2$ rotation method implementation with the Romberg method and on the fixed grid. We saw, averaged over 10 or so iterations, an acceleration of order $10-30\times$ over the naive, $N_g{}^3$ method, with variation of about 10%. This acceleration factor should not be over-interpreted: the scaling of each method with the number of integration points is fundamentally different, so for different problems (e.g. a different number of a and b being integrated at, a different number of k being integrated over), one would likely see different acceleration multipliers. This comment also holds true when considering the acceleration we found for the $N_g \log N_g$ over the $N_g{}^2$ and $N_g{}^3$. We typically found accelerations of $100\times$ for the $N_g \log N_g$ over the $N_g{}^2$, and more like $1000-3000\times$ over the $N_g{}^3$. We detail the absolute timings in the following subsections, but emphasize that the absolute timings will be hardware-dependent. Again we caution that these factors are problem-specific. However, the problems we evaluated are realistic use cases, so these factors likely do offer an accurate sense for the impact of our algorithm on e.g. DESI analytic covariance evaluations. We note that the more challenging the problem (i.e. the more (a,b) or k points needed), the more our FFTLog-based rotation-method implementation ($N_g \log N_g$) will outperform the others: its scaling with the number of input points is simply fundamentally better. This point should be borne in mind when choosing which algorithm to build into code that may then be scaled up.

8.3 Unsmoothed power spectrum

We now go into greater detail on the numerical test cases we investigated. The primary use case we focused on testing in this work was the double-Bessel integrals relevant for the Gaussian Random Field contribution to the covariance matrix of the anisotropic 2PCF multipoles (e.g. Xu et al. 2013), as might be used in a DESI analysis. Taking $P \rightarrow P + 1/n$ and then expanding the integral in equation (24), we find three terms: one proportional to P^2, a second proportional to $P \times (1/n)$, and a third proportional to $(1/n)^2$, with n the number density of the galaxy survey. The third may be done analytically, since $1/n$ is k-independent. The second and first must be done numerically, but of course the factor of $1/n$ in the second may also be pulled out.

So in short, for the covariance matrix, we will have numerical double-sBF integrals against P and P^2. In a typical use-case, several additional modifications would be
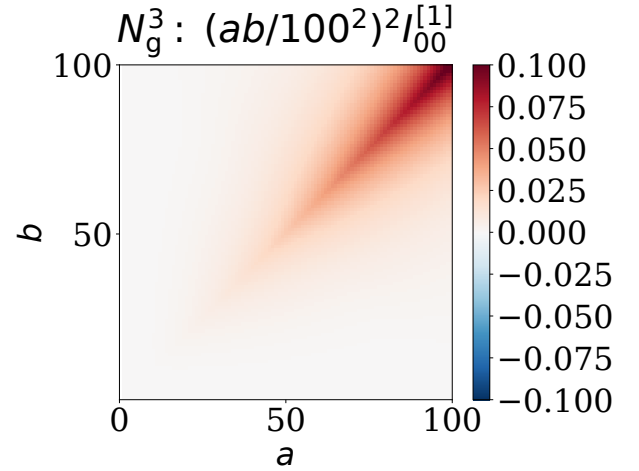
Figure 2. Here we show the weighted integral of $j_0(ka) j_0(kb)$ against $P(k)$ evaluated with the naive, $N_g{}^3$ method. The weights correspond to those one would use when binning in a and b on spherical shells; the volume of a shell in a,b scales as the bin width squared times $a^2 b^2$. Our purpose here is simply to show the actual values to which subsequent plots of differences between the different methods should be compared.
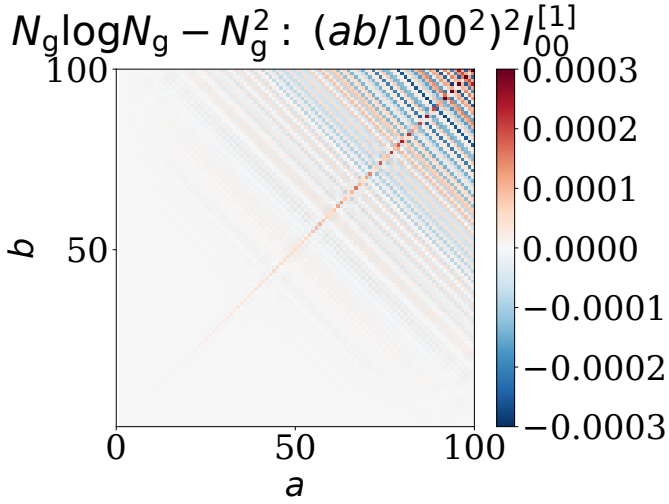
made relative to the test cases we examine.

i) Smoothing of BAO in the power spectrum.
ii) Smoothing of the Fourier transform as we go to configuration space to avoid ringing.
iii) Binning over separations.

We have already discussed these in more detail in §4. We simply recapitulate them now to highlight that our test case, an "unsmoothed" power spectrum with i) no suppression of BAO ii) no smoothing of the FT, and iii) no binning, is a more stringent numerical test than the use cases that would likely occur. So the agreement we achieve here between the three methods tested should be taken as a conservative lower bound on the true agreement likely to occur in applications.

For our tests, we used a numerical power spectrum from CAMB (Lewis et al. 2000) with a cosmology roughly that of Planck 2016 (Planck Collaboration et al. 2016). We also used this for the P^2 tests, described in §8.4. For all of our test cases, we achieved very good agreement between the different implementations. For the $j_0 j_0$ test, the typical agreement was at the level of 10^{-16} between the naive and $N_g{}^2$-rotation method, and the values were of order 10^{-1}. The agreement between $N_g{}^2$ and $N_g \log N_g$ rotation methods was at the level of 3×10^{-4}. For the $j_1 j_1$ test, the differences between naive and $N_g{}^2$ were at the level of 10^{-9}, with the values of the integrals of order 10^{-1} to 1; differences between $N_g{}^2$ and $N_g \log N_g$ were of order 4×10^{-4}. The agreement between the three implementations was less good for the $j_2 j_2$ and $j_0 j_0 j_0$ tests but still typically better than 10^{-4} fractionally of the integrals' values.

For $I_{00}^{[1]}$, going from naive to $N_g{}^2$ saw a factor of $30\times$ speedup (9,200 s vs. 360 s for both integrals done with adap-

$$N_{\mathrm{g}}\log N_{\mathrm{g}} - N_{\mathrm{g}}^2 : (ab/100^2)^2 I_{00}^{[1]}$$

$$N_{\mathrm{g}}\log N_{\mathrm{g}} - N_{\mathrm{g}}^2 : (ab/100^2)^2 I_{00}^{[2]}$$

Figure 3. Here we show the weighted difference between the $N_{\mathrm{g}}\log N_{\mathrm{g}}$ and N_{g}^2 methods. This plot should be compared to Figure 2; one can see that the differences between the two methods here are of order 1/300 the integral's value. We note that N_{g}^2 and N_{g}^3 agree far better than this, though we do not show that plot for brevity. But as a consequence one can take it that this Figure also well-represents the difference between $N_{\mathrm{g}}\log N_{\mathrm{g}}$ and N_{g}^3, as N_{g}^2 can be taken as a proxy for the latter.
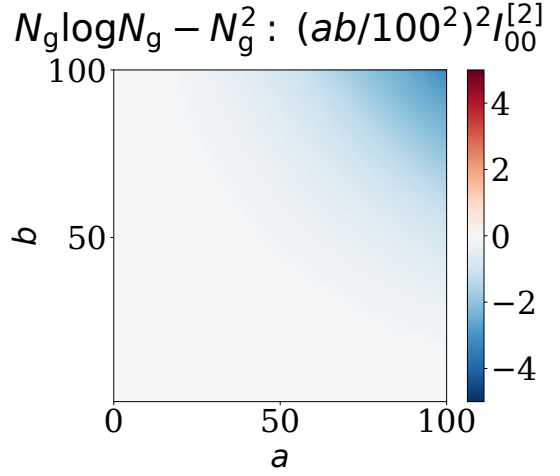
Figure 5. Here we show the weighted difference between the $N_{\mathrm{g}}\log N_{\mathrm{g}}$ and N_{g}^2 methods for the integral of $j_0(ka)j_0(kb)$ against P^2. This plot should be compared to Figure 4; one can see that the differences between the two methods here are of order 1/250 the integral's value. We note that N_{g}^2 and N_{g}^3 agree far better than this, though we do not show that plot for brevity. But as a consequence one can take it that this Figure also well-represents the difference between $N_{\mathrm{g}}\log N_{\mathrm{g}}$ and N_{g}^3, as N_{g}^2 can be taken as a proxy for the latter.

$$N_{\mathrm{g}}^3 : (ab/100^2)^2 I_{00}^{[2]}$$
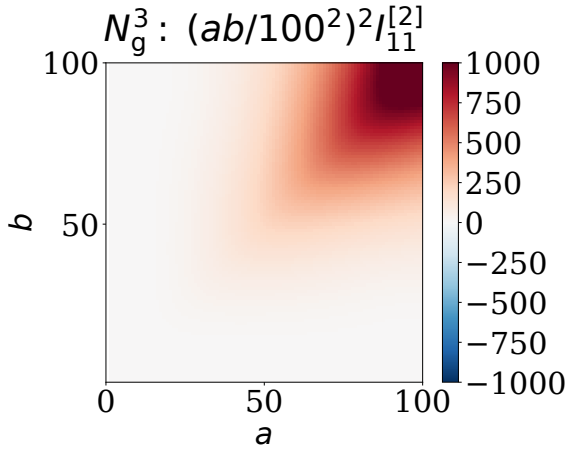
$$N_{\mathrm{g}}^3 : (ab/100^2)^2 I_{11}^{[1]}$$
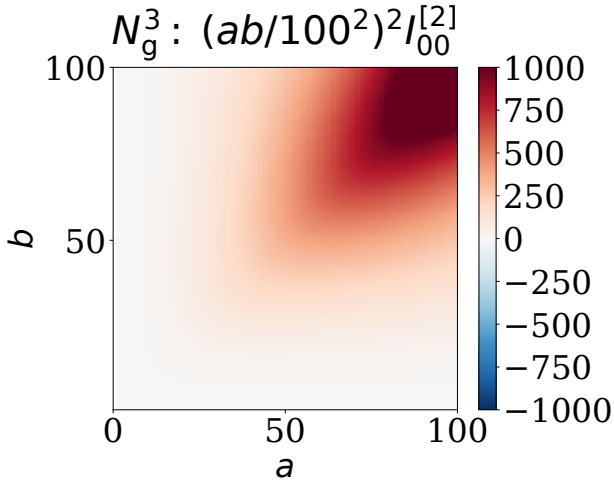
Figure 4. Here we show the weighted integral of $j_0(ka)j_0(kb)$ against $P^2(k)$ evaluated with the naive, N_{g}^3 method. Our purpose here is simply to show the actual values to which subsequent plots of differences between the different methods should be compared. One can also compare this Figure to Figure 2 to get a sense for how squaring the power spectrum alters the integral; essentially it extends the region of strong support of the result away from the diagonal. This makes sense as squaring the power spectrum is convolving the correlation function with itself. This is a smoothing operation and hence will make the off-diagonal contributions higher.
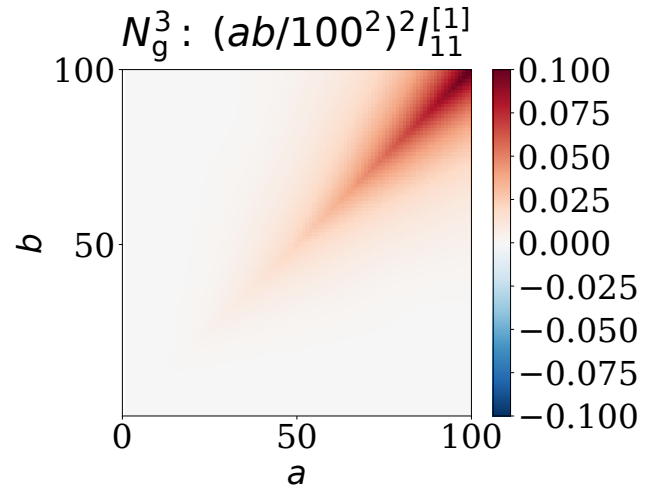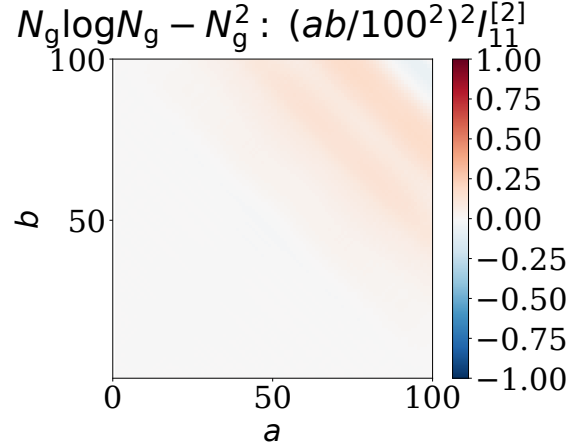
Figure 6. Here we show the weighted integral of $j_1(ka)j_1(kb)$ against $P(k)$ evaluated with the naive, N_{g}^3 method. Our purpose here is simply to show the actual values to which subsequent plots of differences between the different methods should be compared. One can also compare this Figure to Figure 2 to get a sense for how raising the order of each spherical Bessel function alters the integral; essentially it does not do so greatly. The structure remains primarily on the diagonal.
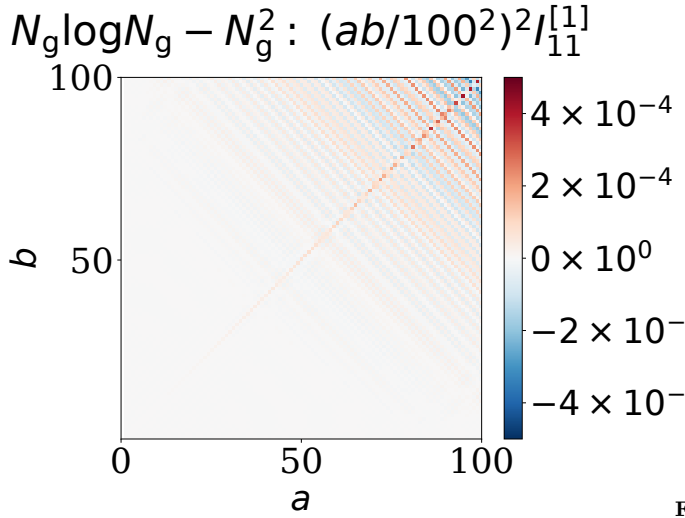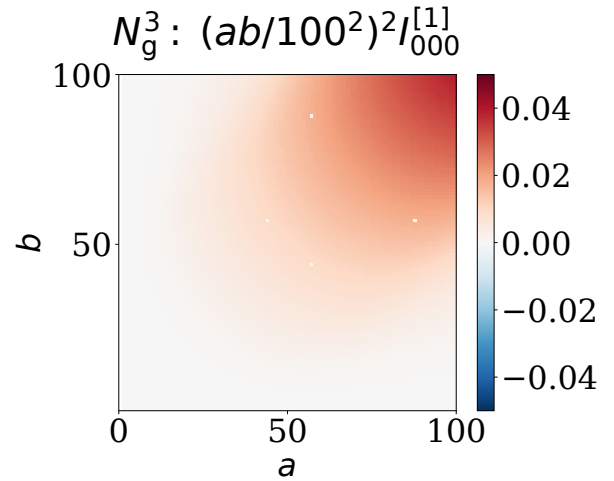
Figure 7. Here we show the weighted difference between the $N_{\rm g}\log N_{\rm g}$ and $N_{\rm g}^2$ methods for the integral of $j_1(ka)j_1(kb)$ against P. This plot should be compared to Figure 6; one can see that the differences between the two methods here are of order $1/250$ the integral's value. We note that $N_{\rm g}^2$ and $N_{\rm g}^3$ agree far better than this, though we do not show that plot for brevity. But as a consequence one can take it that this Figure also well-represents the difference between $N_{\rm g}\log N_{\rm g}$ and $N_{\rm g}^3$, as $N_{\rm g}^2$ can be taken as a proxy for the latter.

Figure 8. Here we show the weighted integral of $j_1(ka)j_1(kb)$ against $P^2(k)$ evaluated with the naive, $N_{\rm g}^3$ method. Our purpose here is simply to show the actual values to which subsequent plots of differences between the different methods should be compared. One can also compare this Figure to Figure 6 to get a sense for how squaring the power spectrum alters the result; the take-away is the same as that discussed in the analogous case for $j_0(ka)j_0(kb)$ (Figure 4).
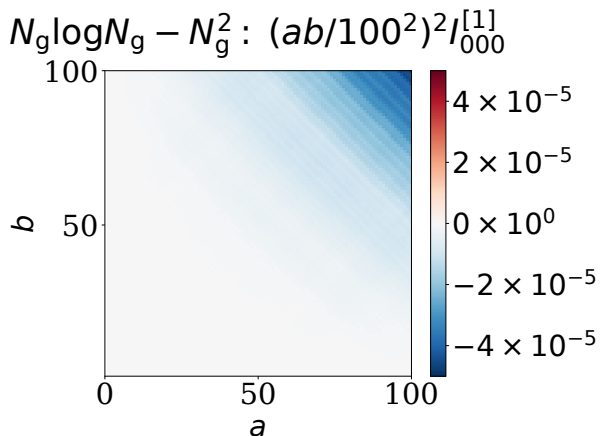

Figure 9. Here we show the weighted difference between the $N_{\rm g}\log N_{\rm g}$ and $N_{\rm g}^2$ methods for the integral of $j_1(ka)j_1(kb)$ against $P^2(k)$. This plot should be compared to Figure 8; one can see that the differences between the two methods here are of order $1/1000$ the integral's value. We note that $N_{\rm g}^2$ and $N_{\rm g}^3$ agree far better than this, though we do not show that plot for brevity. But as a consequence one can take it that this Figure also well-represents the difference between $N_{\rm g}\log N_{\rm g}$ and $N_{\rm g}^3$, as $N_{\rm g}^2$ can be taken as a proxy for the latter.

Figure 10. Here we show the weighted integral of $j_0(ka)j_0(kb)j_0(kc)$ against $P(k)$ evaluated with the naive, $N_{\rm g}^3$ method and displayed on a slice of constant $c = 50$. Our purpose here is simply to show the actual values to which subsequent plots of differences between the different methods should be compared. One can also observe that the structure is much more off diagonal than the results for the two-sBF integrals of both P and P^2.

$N_g \log N_g - N_g^2 : (ab/100^2)^2 I_{000}^{[1]}$

Figure 11. Here we show the weighted difference between the $N_g \log N_g$ and N_g^2 methods for the integral of $j_0(ka)j_0(kb)$ against $P(k)$ and displayed on a slice of constant $c = 50$. This plot should be compared to Figure 10; one can see that the differences between the two methods here are of order $1/1000$ the integral's value. We note that N_g^2 and N_g^3 agree far better than this, though we do not show that plot for brevity. But as a consequence one can take it that this Figure also well-represents the difference between $N_g \log N_g$ and N_g^3, as N_g^2 can be taken as a proxy for the latter.

tive Romberg and a maximum of 15 refinements, 310 s vs. 10 s for both integrals done with adaptive Romberg and a maximum of 10 refinements). For $I_{11}^{[1]}$, the speedup was $10\times$ (30 s vs. 3 s for fixed grid, 370 vs. 32 s for adaptive). For $I_{22}^{[1]}$, the speedup was $7\times$ (390 s vs. 60 s, with adaptive Romberg). Going from N_g^2 to $N_g \log N_g$, runtimes dropped to typically 0.2 s; this corresponds to a factor of $100\times$ speed-up relative to the N_g^2 and $1000 - 3000\times$ relative to the naive. The tests of the naive and N_g^2 methods were run on a 2013 MacBook Air with Intel i7 processor and 8 GB 1600 MHz DDR3 RAM. The tests of the $N_g \log N_g$ method were run on a Lenovo Yoga 2 Pro with the same processor and amount of RAM, so the results can reasonably be cross-compared.

8.4 Squared unsmoothed power spectrum

Squaring the power spectrum will more sharply concentrate it about its peak. This can also be intuitively seen since in configuration space, squaring would correspond to convolving the linear correlation function ξ_0 with itself. On the very simplistic cartoon model that ξ_0 is a Gaussian, convolving it with itself would increase the width by $\sqrt{2}$. By the uncertainty relation, we see that this would then shrink the width of the power spectrum by the same factor. This means that the range of k over which P^2 is appreciable is smaller than that for P. Reducing this range should make our integration approach even more accurate. Indeed, the main source of inaccuracy is the high powers of k in the denominators, where one needs to track them accurately to get good cancellation when they are summed up at the end of the rotation method. Reducing the range in k over which the integrand is non-negligible will tend to make this issue better.

We tested for agreement between the three methods for

P^2. We found sub-percent agreement as long as we used $k_{max} = 4 \, h/\text{Mpc}$ for the N_g^3 and N_g^2 methods (the k grid for the $N_g \log N_g$ method was logarithmically-spaced from $10^{-4} \, h/\text{Mpc}$ to $100 \, h/\text{Mpc}$, with 8192 points. We did not extrapolate as a power-law outside either end of this range, which is also possible in this approach.

We believe the need to truncate at a different k_{max} to find good agreement between the methods is driven by details of how the N_g^3 and N_g^2 methods handle the integration. Both are done with adaptive Romberg, and the Romberg probably breaks down in accuracy when attempting to deal with the high-k points where the sBFs (or sines and cosines) greatly oscillate. This likely drives it to spend most of its sampling points there, even though these high-k regions do not matter for the value of the integrals. With adaptive Romberg, there is no way to control where it samples, save for the crude method of simply truncating at lower k_{max}. With Romberg on a fixed grid, the grid must be linearly spaced, not log spaced, so using that approach is not an obvious fix. One could imagine remapping the integrand to be in terms of $u = \ln k$, and then the integrand would be linearly spaced in u for a logarithmic k grid. However, we did not pursue this, as we felt sub-percent agreement between the methods was sufficient. In any case, we advocate the $N_g \log N_g$ method for production analyses, and it is not affected by these points.

We note that the DESI statistical error bars on the signal should be of order 1% in of order ten redshift bins (DESI Collaboration et al. 2016), for a total error of 0.1% on the BAO scale when these are combined. We emphasize that these percentages are relative to the signal. Making a fractional error of 10^{-4} in the covariance (which by Taylor series translates to $\sim 1/2 \times 10^{-4}$ in the error bar itself) would then constitute a fraction $1/2 \times 10^{-4} \times 1\%$ of the DESI signal. This level of error will be irrelevant to the derived likelihood.

8.5 Analytic test case

For additional confirmation of our method and implementation, we also evaluated a test case where the exact answer is known analytically. This test we focused on the FFTLog implementation of the rotation method, as that is the one we advocate putting into practical use as it is the fastest and has the best handling of dynamic range. Figures 12-14 show that the method performs quite well over a large range of a and b. While it initially appears that the relative error becomes large in some regions, this is simply because machine precision sets a floor on the absolute error, while the absolute value of the integral becomes quite small. Thus the relative error goes up.

In practice, though, this rise in relative error will not matter. As we have already discussed, typically one bins the covariance (e.g. Xu et al. 2013, equations 6-9), and the points where the absolute value of the integral is tiny will contribute very little to the binned result. Furthermore, many of the areas where the relative error is noticeable correspond to small a or b, and when binning one weights by the bin volume, scaling as $a^2 b^2$. This weight will further reduce the contribution these regions of higher relative error make to the binned covariance.

Finally, we note that the analytic test case is likely more stringent than the power spectrum (or its square) that might

be typical use cases. The analytic test case is less sharply peaked than the power spectrum. The power spectrum scales roughly as $k/[1+(k/k_{eq})^2]^2$ and is usually smoothed by a Gaussian as $\exp[-k^2\sigma^2]$ with $\sigma \simeq 1$ Mpc.[18] The power spectrum scaling is thus, quite roughly, the square of our analytic test case, making the region where the power spectrum is non-negligible rather narrower than that for the analytic test case. Thus for the actual power spectrum, we need to resolve less dynamic range in k than we do for the analytic test case, making this latter a more stringent test of our method.

We now detail our analytic test case. It is Gradshteyn et al. (2007) 6.577.1,

$$\int_0^\infty x^{\nu-\mu+1+2n} J_\mu(ax) J_\nu(bx) \frac{dx}{x^2+c^2} \tag{48}$$
$$= (-1)^n c^{\nu-\mu+2n} I_\mu(ac) K_\nu(bc), \quad \text{for}$$

$b > a > 0, \mathrm{Re}\, c > 0, 2 + \mathrm{Re}\,\mu - 2n > \mathrm{Re}\,\nu > -1-n, 0 \le n \in \mathbb{Z},$

with the following special case when $n = 0$, $\mu = 3/2$, and $\nu = 3/2$:

$$\int_0^\infty j_1(ax) j_1(bx) \frac{x^2\,\mathrm{d}x}{1+x^2} = \frac{\pi}{2a^2 b^2}$$
$$\times \begin{cases} (1+b)e^{-b}(a\cosh a - \sinh a), & a \le b, \\ (1+a)e^{-a}(b\cosh b - \sinh b), & a > b. \end{cases} \tag{49}$$

Expanding the lefthand side of equation (49), we find eight terms (as displayed in equation 34, and setting $\Sigma = a+b$ and $\Delta = a-b$). Two out of these terms are proportional to

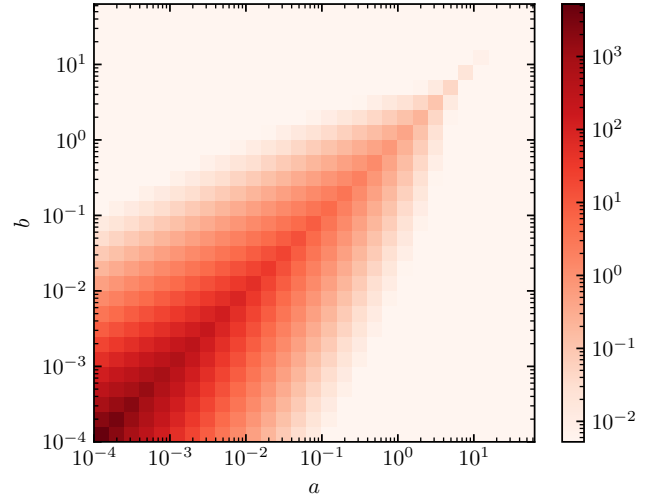$$\int_0^\infty dx \frac{\cos[(a \pm b)x]}{x^2(1+x^2)}$$

which diverges in the infrared ($k \to 0$). However their sum converges. We will use `mcfit` to compute an incorrect finite result for each of the divergent terms and expect they will sum up to the correct one (as the true, divergent results would also do).

`mcfit` is a PYTHON package that implements the FFT-Log algorithm for fast Hankel transforms (Talman 1978, Hamilton 2000). The algorithm assumes that the input function is sampled on a logarithmically spaced grid. It then approximates the input function (of logarithmic variable) with a truncated Fourier series, and evaluates the exact Hankel transform of each Fourier mode, before adding them up. The analytic treatment avoids numerically computing the oscillatory integrals involving Bessel functions, therefore leading to more accurate and efficient results. We refer the readers to the above references for more details.

We use `mcfit` to integrate each term of the expansion equation (34). Each term can be computed to give a finite value, even the seemingly divergent ones. This is because

Figure 12. Here we show the integral for our analytic test case equation (49), for comparison with the plots of the error in our method we then display in Figure 14. This plot is just meant to give a sense for the structure of this integral to then aid in interpreting Figure 14.
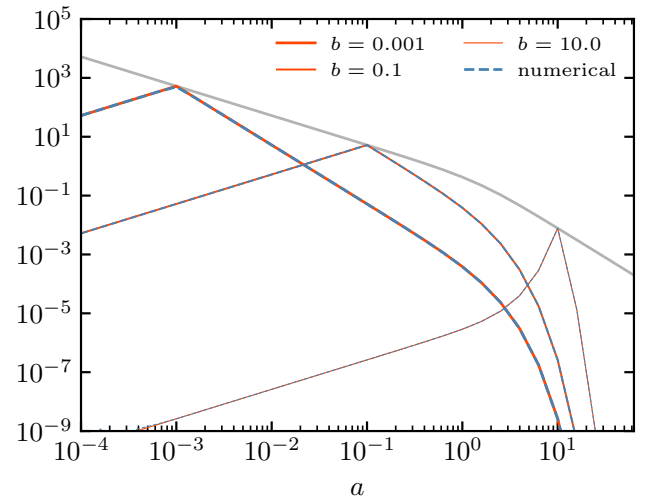

Figure 13. Comparison between numerical (dashed blue) and analytic (solid red) results, shown as functions of a at the three different fixed values of b indicated in the legend and displayed in lines of varying width. The grey line gives the peak amplitudes on the trace $a = b$ through the (a,b) plane, i.e. the diagonal of Fig. 12, for comparison. Of course the blue and red curves meet the grey when $a = b$; generally the amplitudes drop sharply away from the diagonal. Overall the agreement is quite good over a very large dynamic range of both a, b and values of the integral.

the exact Hankel transform of the Fourier modes is in a form whose divergence can be removed by analytic continuation.[19]

The rotation method requires a linear grid, which can be obtained by interpolation from the logarithmically-spaced

---

[18] k_{eq} is the wavenumber corresponding to the scale of the horizon at matter-radiation equality, and this term gives the turnover of the matter transfer function for modes that entered the horizon during radiation-domination and thus had their growth suppressed as the potential evolved. This was first noted in Mészáros (1974); a configuration-space picture is presented in Slepian & Eisenstein (2016b). Also in the numerator we took it that the scalar spectral tilt $n_s \simeq 1$; in reality it is measured to be $n_s = 0.965 \pm 0.004$ from *Planck* data (Collaboration et al. 2018).

[19] The explicit form is equation (B18) of Hamilton (2000), where divergences arise at the poles of the Gamma function.
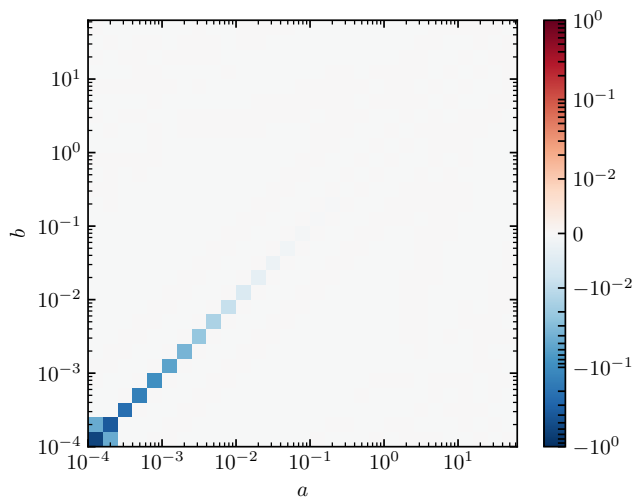
Figure 14. Numerical error, normalized by the square root of the product of the diagonal of Figure 12, $\sqrt{I_{aa}I_{bb}}$. The residual at the lower left corner is at the 10% level, consistent with Fig. 13 given the large range of the vertical axis there.

output of `mcfit`. However, interpolations, typically with cubic spline, are prone to numerical errors, especially so when one output variable is much larger than the other. We define C_n and S_n as follows:

$$C_n(y) = \int \frac{x^2\,dx}{1+x^2}\frac{\cos(xy)}{x^n}, \tag{50}$$

$$S_n(y) = \int \frac{x^2\,dx}{1+x^2}\frac{\sin(xy)}{x^n}. \tag{51}$$

If $a \gg b$, combinations like $C_n(a+b) - C_n(a-b)$ would produce $2bC_n'(a)$ at leading order, which involves the derivative of the spline. As a numerical derivative, this will generally be less accurate than the spline interpolation itself. To solve this problem, we adopt a Hermite spline in which the derivatives are explicitly supplied alongside the function values. With a Hermite spline, the piecewise polynomials are determined locally by the function's values, its derivatives, and the end points. This is in contrast with the (cubic) interpolating spline, whose piecewise polynomials are determined globally by all the function values. Specifially for the test case we use the quintic Hermite spline that contructs piecewise polynomials with both first and second derivatives. The derivatives of C_n and S_n form a mutual recursion

$$\frac{dS_n}{dy} = C_{n-1},$$
$$\frac{dC_n}{dy} = -S_{n-1}. \tag{52}$$

where on the righthand side the derivatives are terms that also appear in the rotation method expansion. This allows us to reuse those numerical computations in contructing the Hermite splines. We have verified that the quintic Hermite spline gives more accurate results than the cubic interpolating spline.

To compute each C_n or S_n, we sample $1/(1+x^2)$ from $k = 10^{-2}$ h/Mpc to 10^4 h/Mpc at 300 logarithmically-spaced points, before extending them on both sides using power-law extrapolations from the respective end segments. The ex-

trapolation is implemented in `mcfit` as a default option, and the total number of data points including the extrapolation is 8192. `mcfit` then performs Fourier sine or cosine tranforms to compute S_n or C_n, before adding them up to obtain the result. On a single core Intel i7-4510U CPU (2.00 GHz), the numerical computation with quintic Hermite interpolation takes about 200 ms. However, this is obviously limited and dominated by the `scipy` implementation of Hermite spline (`scipy.interpolate.BPoly.from_derivatives`), given that computation with a cubic spline only takes about 30 ms.

9 CONCLUSION & DISCUSSION

In this work, we have presented a new way of evaluating integrals of two or more spherical Bessel functions against a numerical source function such as the matter power spectrum. This is a common problem in cosmology, as these integrals generically emerge when correlation functions with both points displaced from the origin are written as a multipole series. Our method works by rewriting the spherical Bessel functions as sines and cosines weighted by powerlaws, multiplying them out, and recoupling them into sines and cosines of the sum and difference of frequencies. This corresponds to rotating 45 degrees in the plane of free frequencies (i.e. not the integration variable), and essentially decouples the problem so that 3-D integrals become a sum of 2-D integrals. The method also trivially extends to integrals of three or more sBFs, as we show with the example of $j_0(ka)j_0(kb)j_0(kc)$. While the method costs more integrals, they are all 2-D, and this results in a signficant computational cost savings over the naive approach to this computation. In particular, if FFTLog is used to evaluate the required 2-D integrals, then they become even faster than 2-D. They then scale as $N_g \log N_g$ as compared to N_g^2 for the "rotation" approach without FFTLog and N_g^3 for the "naive" approach. In our implementation of a typical use case, the integral of two sBFs against the power spectrum, this yields a roughly $1000 - 3000\times$ speed-up (depending on the specific combination of sBF indices and other details). Unlike other work, our method does not require any expansion or approximation of the power spectrum (e.g. into complex power laws). The direct (N_g^2) version of our method can also be used on any desired numerical grid, whereas the power law method must use a regular grid for its FT. Our method is also the first to show how to perform these integrals for mismatched indices (i.e. $\ell \neq \ell'$ for the sBFs) and for three or more sBFs.

The most closely related works to the present are Assassi et al. (2017) and Grasshorn Gebhardt & Jeong (2018), also discussed in §7. These works expand the power spectrum in complex power laws (which can capture oscillatory behavior) and use analytic results for the integrals of 2 sBFs against a power law to perform the integrations. The key requirement of those works is that the power spectrum can be expanded as a sum of complex power laws; this is just the condition that it has an inverse Fourier Transform, as the expansion coefficients are in fact its discrete inverse FT. This should hold for most standard power spectra but will fail for ones that diverge in the IR faster than k^{-2}. Local non-Gaussianity generates a nearly-IR-divergent

scale-dependent bias as k^{-2} (Dalal et al. 2008), so we speculate that more exotic non-Gaussianity models might diverge faster. They would therefore not interact well with the power-law method. However, they might still convergently be integrated against double sBFs with ℓ or ℓ' greater than zero because these will cancel the divergence. Our method could thus be used in this case even though the power-law method would break down.

Regarding computational efficiency, the power-law method requires one inverse FT to get the power spectrum's coefficients, and can then use a pre-computed lookup table of the analytic results for the power-law-double-sBF integrals and perform a matrix multiply with the coefficients. Creating the lookup table would be expensive but need only be done once. The lookup table would then need to be loaded from disk, which might be slow depending on the read rate and size of the table. In contrast, our method does not require any lookup tables, but computes a small number of FTs on the fly, all of which can remain in RAM during the computation. The speed comparison between the power law method and ours would thus be machine-dependent because each uses different pieces of the machine.

The approach presented here should be highly useful for problems such as computing the analytic, Gaussian Random Field template for the anisotropic 2PCF multipoles' covariance matrix, such as might be required in analyzing large redshift-survey data. In particular, future surveys are likely to be analyzed using MCMC over cosmological parameters and one would like to recompute the covariance matrix at each set of parameters. Thus one might be computing the covariance matrix millions of times, making efficient evaluation of it highly desirable. We have tested all the numerical cases required for the covariance matrix of the monopole and quadrupole, as well as cross-terms. We find agreement between our new method and the naive approach sufficient so that any differences are a negligible source of systematic error in e.g. a DESI analysis.

The approach presented here needs more work to extend to higher sBF index (ℓ). This is because the recoupled, 2-D integrals it uses have large powers of k in their denominators, which produce substantial dynamic range in the integrand and thereby strain the bounds of numerical precision. In particular, the powers of k grow with the maximal ℓ used in the computation, and thus a small input range of k maps to a larger and larger range for the integrand. Some of the integrals entering the expansion become formally divergent, though these divergences of course cancel when the integrals are summed in the end. The divergences do not appear numerically because we always integrate over a finite range. However in practice one still ends up needing quite precise cancellations between large values to get a smaller end result. We considered several schemes to address this issue, but none worked satisfactorily (and indeed, none were needed for the cases on which we focused). Future work might explore if splitting the broadband behavior of the power spectrum from the BAO wiggles helps alleviate this issue. The BAO wiggles are only present over roughly two decades in k, and one could imagine treating the broadband as a power law over much of its range, performing those integrals analytically, and using our scheme only for the BAO piece. Restricting the input range would of course reduce the output dynamic range in the integrand. However,

even this approach would only go so far as ℓ rises, since even two decades produces an intractable dynamic range if raised to a sufficiently high power.

Nonetheless, overall this paper represents a new, independent, highly efficient method for evaluating the types of double and triple-sBF integrals ubiquitous in cosmology, and for many use-cases the ℓ we have already implemented are sufficient. We hope this algorithm will be of use in analyzing data from the next generation of large surveys.

REFERENCES

Alam S., et al., 2017, MNRAS, 470, 2617
Angulo R., Fasiello M., Senatore L., Vlah Z., 2015, Journal of Cosmology and Astro-Particle Physics, p. 029
Ansari R., et al., 2018, Cosmic Visions 21 cm Collaboration: Inflation and Early Dark Energy with a Stage II Hydrogen Intensity Mapping Experiment (arXiv:1810.09572)
Arfken G. B., Weber H. J., Harris F. E., 2013, Mathematical Methods for Physicists, 7th edn. Academic Press, Boston
Assassi V., Baumann D., Green D., Zaldarriaga M., 2014, Journal of Cosmology and Astro-Particle Physics, 2014, 056
Assassi V., Simonović M., Zaldarriaga M., 2017, J. Cosmology Astropart. Phys., 11, 054
Bandura K., et al., 2014, Ground-based and Airborne Telescopes V
Bandura K., et al., 2019, Packed Ultra-wideband Mapping Array (PUMA): A Radio Telescope for Cosmology and Transients (arXiv:1907.12559)
Bernardeau F., Colombi S., Gaztañaga E., Scoccimarro R., 2002, Phys. Rep., 367, 1
Cataneo M., Foreman S., Senatore L., 2017, Journal of Cosmology and Astro-Particle Physics, 2017, 026
Collaboration P., et al., 2018, Planck 2018 results. VI. Cosmological parameters (arXiv:1807.06209)
Cooley J. W., Tukey J. W., 1965, Mathematics of Computation, 19, 297
DESI Collaboration et al., 2016, preprint, (arXiv:1611.00036)
Dalal N., Dorá O., Huterer D., Shirokov A., 2008, Physical Review D, 77

Danielson G., Lanczos C., 1942, Journal of the Franklin Institute, 233, 365

Dawson K. S., et al., 2016, AJ, 151, 44

Desjacques V., Jeong D., Schmidt F., 2018, Phys. Rep., 733, 1

Diemer B., 2018, The Astrophysical Journal Supplement Series, 239, 35

Fabrikant V. I., 2003, Zeitschrift Angewandte Mathematik und Mechanik, 83, 363

Fabrikant V. I., 2013, Quarterly of Applied Mathematics, LXXI, 3, 573

Fonseca de la Bella L., Regan D., Seery D., Hotchkiss S., 2017, J. Cosmology Astropart. Phys., 11, 039

Gradshteyn I. S., Ryzhik I. M., Jeffrey A., Zwillinger D., 2007, Table of Integrals, Series, and Products, 7$^{\text{th}}$ edn. Academic Press, Boston.

Grasshorn Gebhardt H. S., Jeong D., 2018, Phys. Rev. D, 97, 023504

Group S. K. A. C. S. W., et al., 2018, Cosmology with Phase 1 of the Square Kilometre Array; Red Book 2018: Technical specifications and performance forecasts (arXiv:1811.02743)

Hamilton A. J. S., 1992, ApJ, 385, L5

Hamilton A. J. S., 2000, MNRAS, 312, 257

Howlett C., Percival W. J., 2017, MNRAS, 472, 4935

LSST Dark Energy Science Collaboration 2012, preprint, (arXiv:1211.0310)

Laureijs R., et al., 2011, preprint, (arXiv:1110.3193)

Lewis A., Challinor A., Lasenby A., 2000, ApJ, 538, 473

Li Y., Singh S., Yu B., Feng Y., Seljak U., 2018, preprint, (arXiv:1811.05714)

Limber D. N., 1953, ApJ, 117, 134

McDonald P., Roy A., 2009, Journal of Cosmology and Astro-Particle Physics, 2009, 020

McEwen J. E., Fang X., Hirata C. M., Blazek J. A., 2016, Journal of Cosmology and Astro-Particle Physics, 2016, 015

Mészáros P., 1974, A&A, 37, 225

Mirbabayi M., Schmidt F., Zaldarriaga M., 2015, Journal of Cosmology and Astro-Particle Physics, 2015, 030

Mohammed I., Seljak U., Vlah Z., 2017, MNRAS, 466, 780

Newburgh L. B., et al., 2016, Ground-based and Airborne Telescopes VI

O'Connell R., Eisenstein D., Vargas M., Ho S., Padmanabhan N., 2016, MNRAS, 462, 2681

Olver F. W. J., Lozier D. W., Boisvert R. F., Clark C. W., 2010, NIST Handbook of Mathematical Functions. Cambridge University Press, New York

Padmanabhan N., White M., Zhou H. H., O'Connell R., 2016, MNRAS, 460, 1567

Parkinson D., et al., 2012, Phys. Rev. D, 86, 103518

Percival W. J., et al., 2014, MNRAS, 439, 2531

Planck Collaboration et al., 2016, A&A, 594, A13

Sabiu C. G., Hoyle B., Kim J., Li X.-D., 2019, preprint, (arXiv:1901.00296)

Schmittfull M., Vlah Z., 2016, Phys. Rev. D, 94, 103530

Schmittfull M., Vlah Z., McDonald P., 2016, Phys. Rev. D, 93, 103528

Scodeggio M., et al., 2018, A&A, 609, A84

Senatore L., 2015, Journal of Cosmology and Astro-Particle Physics, 2015, 007

Siegman A. E., 1977, Opt. Lett., 1, 13

Simonović M., Baldauf T., Zaldarriaga M., Carrasco J. J., Kollmeier J. A., 2018, J. Cosmology Astropart. Phys., 4, 030

Slepian Z., 2018, preprint, (arXiv:1812.02728)

Slepian Z., Eisenstein D. J., 2015a, MNRAS, 448, 9

Slepian Z., Eisenstein D. J., 2015b, MNRAS, 454, 4142

Slepian Z., Eisenstein D. J., 2016a, MNRAS, 455, L31

Slepian Z., Eisenstein D. J., 2016b, MNRAS, 457, 24

Slepian Z., Eisenstein D. J., 2017, MNRAS, 469, 2059

Slepian Z., Eisenstein D. J., 2018a, MNRAS, 478, 1468

Slepian Z., Eisenstein D. J., 2018b, MNRAS, 478, 1468

Slepian Z., et al., 2016, preprint, (arXiv:1607.06098)

Szapudi I., 2004, ApJ, 605, L89

Talman J. D., 1978, Journal of Computational Physics, 29, 35

The Dark Energy Survey Collaboration, 2005, preprint, (arXiv:astro-ph/0510346)

Tomlinson J., Jeong D., Kim J., 2019, preprint, (arXiv:1904.11055)

Varshalovich D. A., Moskalev A. N., Khersonskii V. K., 1988, Quantum Theory of Angular Momentum. World Scientific Publishing Co., Singapore

Xu X., Cuesta A. J., Padmanabhan N., Eisenstein D. J., McBride C. K., 2013, MNRAS, 431, 2834

APPENDIX A: PRODUCT OF TWO SPHERICAL BESSEL FUNCTIONS IN TERMS OF TRIGONOMETRIC FUNCTIONS

From Olver et al. (2010) 10.49.2 we have

$$j_\ell(kr) = \sin\left(kr - \frac{\ell\pi}{2}\right) \sum_{n=0}^{\lfloor \ell/2 \rfloor} (-1)^n \frac{a_{2n}(\ell+\frac{1}{2})}{(kr)^{2n+1}} + \cos\left(kr - \frac{\ell\pi}{2}\right) \sum_{m=0}^{\lfloor (\ell-1)/2 \rfloor} (-1)^m \frac{a_{2m+1}(\ell+\frac{1}{2})}{(kr)^{2m+2}}, \tag{A1}$$

where

$$a_w(\ell+\tfrac{1}{2}) = \frac{[4(\ell+\frac{1}{2})^2 - 1^2][4(\ell+\frac{1}{2})^2 - 3^2]\cdots[4(\ell+\frac{1}{2})^2 - (2w-1)^2]}{w!\,8^w} \tag{A2}$$

is a polynomial in ℓ.

The product of two spherical Bessel functions is then

$$\begin{aligned}
j_\ell(kr) j_{\ell'}(kr') = &\sin\left(kr - \frac{\ell\pi}{2}\right)\sin\left(kr' - \frac{\ell'\pi}{2}\right) \sum_{n=0}^{\lfloor \ell/2 \rfloor} \sum_{n'=0}^{\lfloor \ell'/2 \rfloor} (-1)^{n+n'} \frac{a_{2n}(\ell+\frac{1}{2})}{(kr)^{2n+1}} \frac{a_{2n'}(\ell'+\frac{1}{2})}{(kr')^{2n'+1}} \\
&+ \sin\left(kr - \frac{\ell\pi}{2}\right)\cos\left(kr' - \frac{\ell'\pi}{2}\right) \sum_{n=0}^{\lfloor \ell/2 \rfloor} \sum_{m'=0}^{\lfloor (\ell'-1)/2 \rfloor} (-1)^{n+m'} \frac{a_{2n}(\ell+\frac{1}{2})}{(kr)^{2n+1}} \frac{a_{2m'+1}(\ell'+\frac{1}{2})}{(kr')^{2m'+2}} \\
&+ \cos\left(kr - \frac{\ell\pi}{2}\right)\sin\left(kr' - \frac{\ell'\pi}{2}\right) \sum_{m=0}^{\lfloor (\ell-1)/2 \rfloor} \sum_{n'=0}^{\lfloor \ell'/2 \rfloor} (-1)^{m+n'} \frac{a_{2m+1}(\ell+\frac{1}{2})}{(kr)^{2m+2}} \frac{a_{2n'}(\ell'+\frac{1}{2})}{(kr')^{2n'+1}} \\
&+ \cos\left(kr - \frac{\ell\pi}{2}\right)\cos\left(kr' - \frac{\ell'\pi}{2}\right) \sum_{m=0}^{\lfloor (\ell-1)/2 \rfloor} \sum_{m'=0}^{\lfloor (\ell'-1)/2 \rfloor} (-1)^{m+m'} \frac{a_{2m+1}(\ell+\frac{1}{2})}{(kr)^{2m+2}} \frac{a_{2m'+1}(\ell'+\frac{1}{2})}{(kr')^{2m'+2}}.
\end{aligned} \tag{A3}$$

We can manipulate these relations to obtain explicit reduction formulae for an arbitrary product of any number of spherical Bessel functions. Here we obtain results for products of two; it is trivial to pursue the same approach for products of three if desired.

We first take each trigonometric function and use the angle sum formulae to rewrite it as a sum of products of trigonometric functions of kr, $\ell\pi/2$, kr' and $\ell'\pi/2$. For instance, for the first term we find

$$\begin{aligned}
\sin\left(kr - \frac{\ell\pi}{2}\right)\sin\left(kr' - \frac{\ell'\pi}{2}\right) = &\sin kr \sin kr' \cos\frac{\ell\pi}{2}\cos\frac{\ell'\pi}{2} - \sin kr \cos kr' \cos\frac{\ell\pi}{2}\sin\frac{\ell'\pi}{2} - \cos kr \sin kr' \sin\frac{\ell\pi}{2}\cos\frac{\ell'\pi}{2} \\
&+ \cos kr \cos kr' \sin\frac{\ell\pi}{2}\sin\frac{\ell'\pi}{2}.
\end{aligned} \tag{A4}$$

We now notice that

$$\cos\frac{\ell\pi}{2} = (1 - \ell \bmod 2)(-1)^{\ell/2} \equiv K_\ell^{\rm c}. \tag{A5}$$

We then have the product

$$\cos\frac{\ell\pi}{2}\cos\frac{\ell'\pi}{2} \equiv K_{\ell\ell'}^{\rm cc} = K_\ell^{\rm c} K_{\ell'}^{\rm c}. \tag{A6}$$

This notation enables us to write

$$\sin\left(kr - \frac{\ell\pi}{2}\right)\sin\left(kr' - \frac{\ell'\pi}{2}\right) = \frac{1}{2}\left\{ [\cos k\Delta - \cos k\Sigma] K_{\ell\ell'}^{\rm cc} - [\sin k\Sigma + \sin k\Delta] K_{\ell\ell'}^{\rm cs} - [\sin k\Sigma - \sin k\Delta] K_{\ell\ell'}^{\rm sc} - [\cos k\Delta + \cos k\Sigma] K_{\ell\ell'}^{\rm ss} \right\}. \tag{A7}$$

We may obtain similar results for the other terms in equation (A3). We find

$$\sin\left(kr - \frac{\ell\pi}{2}\right)\cos\left(kr' - \frac{\ell'\pi}{2}\right) = \frac{1}{2}\left\{ [\sin k\Sigma + \sin k\Delta] K_{\ell\ell'}^{\rm cc} + [\cos k\Delta - \cos k\Sigma] K_{\ell\ell'}^{\rm cs} - [\cos k\Delta + \cos k\Sigma] K_{\ell\ell'}^{\rm sc} - [\sin k\Sigma - \sin k\Delta] K_{\ell\ell'}^{\rm ss} \right\}. \tag{A8}$$

The third term in equation (A3) is just the second term with $r \leftrightarrow r'$ and $\ell \leftrightarrow \ell'$, so we have $\Sigma \to \Sigma$, $\Delta \to -\Delta$, $\ell \to \ell'$, $\ell' \to \ell$ in equation (A8), yielding

$$\cos\left(kr - \frac{\ell\pi}{2}\right)\sin\left(kr' - \frac{\ell'\pi}{2}\right) = \frac{1}{2}\left\{ [\sin k\Sigma - \sin k\Delta] K_{\ell\ell'}^{\rm cc} + [\cos k\Delta - \cos k\Sigma] K_{\ell\ell'}^{\rm cs} - [\cos k\Delta + \cos k\Sigma] K_{\ell\ell'}^{\rm sc} - [\sin k\Sigma + \sin k\Delta] K_{\ell\ell'}^{\rm ss} \right\}. \tag{A9}$$

Finally, for the last term in equation (A3) we find

$$\cos\left(kr - \frac{\ell\pi}{2}\right)\cos\left(kr' - \frac{\ell'\pi}{2}\right) = \frac{1}{2}\left\{ [\cos k\Delta + \cos k\Sigma] K_{\ell\ell'}^{\rm cc} + [\sin k\Sigma - \sin k\Delta] K_{\ell\ell'}^{\rm cs} + [\sin k\Sigma + \sin k\Delta] K_{\ell\ell'}^{\rm sc} + [\cos k\Delta - \cos k\Sigma] K_{\ell\ell'}^{\rm ss} \right\}. \tag{A10}$$

We now want to sum the four terms, each weighted by the double sum given in equation (A3). Denoting the double sums in

equation (A3) by w_1 through w_4 we may co-add all terms in each trigonometric function of $k\Delta$ and $k\Sigma$, finding

$$
\begin{aligned}
j_\ell(kr)j_{\ell'}(kr') = \frac{1}{2}\Bigg\{ & \cos k\Delta \Big[w_1(K^{\mathrm{cc}}_{\ell\ell'} + K^{\mathrm{ss}}_{\ell\ell'}) + w_2(K^{\mathrm{cs}}_{\ell\ell'} - K^{\mathrm{sc}}_{\ell\ell'}) + w_3(K^{\mathrm{cs}}_{\ell\ell'} - K^{\mathrm{sc}}_{\ell\ell'}) + w_4(K^{\mathrm{cc}}_{\ell\ell'} + K^{\mathrm{ss}}_{\ell\ell'}) \Big] \\
+ & \cos k\Sigma \Big[w_1(-K^{\mathrm{cc}}_{\ell\ell'} + K^{\mathrm{ss}}_{\ell\ell'}) + w_2(-K^{\mathrm{cs}}_{\ell\ell'} - K^{\mathrm{sc}}_{\ell\ell'}) + w_3(-K^{\mathrm{cs}}_{\ell\ell'} - K^{\mathrm{sc}}_{\ell\ell'}) + w_4(K^{\mathrm{cc}}_{\ell\ell'} - K^{\mathrm{ss}}_{\ell\ell'}) \Big] \\
+ & \sin k\Delta \Big[w_1(-K^{\mathrm{cc}}_{\ell\ell'} + K^{\mathrm{ss}}_{\ell\ell'}) + w_2(K^{\mathrm{cs}}_{\ell\ell'} + K^{\mathrm{sc}}_{\ell\ell'}) + w_3(-K^{\mathrm{cs}}_{\ell\ell'} - K^{\mathrm{sc}}_{\ell\ell'}) + w_4(-K^{\mathrm{cc}}_{\ell\ell'} + K^{\mathrm{ss}}_{\ell\ell'}) \Big] \\
+ & \sin k\Sigma \Big[w_1(-K^{\mathrm{cc}}_{\ell\ell'} - K^{\mathrm{ss}}_{\ell\ell'}) + w_2(K^{\mathrm{cs}}_{\ell\ell'} - K^{\mathrm{sc}}_{\ell\ell'}) + w_3(K^{\mathrm{cs}}_{\ell\ell'} - K^{\mathrm{sc}}_{\ell\ell'}) + w_4(K^{\mathrm{cc}}_{\ell\ell'} + K^{\mathrm{ss}}_{\ell\ell'}) \Big] \Bigg\}.
\end{aligned}
\tag{A11}
$$

Using the notation we have set up it would be straightforward, albeit algebraically involved, to obtain the analogous expression for a product of three or more spherical Bessel functions.

This paper has been typeset from a TeX/LaTeX file prepared by the author.