# Fast transforms for Gaussian random fields

by

Paul G. Beckman

A dissertation submitted in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

Department of Mathematics

New York University

August, 2025

Dr. Michael O'Neil

i only wanted to look in the well

but ended up deeper than expected

the pinhole camera hiding a widening panorama.

i only waded in to expel

the holy pretext of windows and cellophane.

For Aleks.

# Acknowledgements

First and foremost, I would like to express my sincerest gratitude to Professor Mike O'Neil, whose unceasing generosity with his time, ideas, expertise, collaboration, opportunities, advice, and support made my time at Courant truly fulfilling and engaging.

I would also like to thank my committee — Professors Leslie Greengard, Alex Barnett, Georg Stadler, and Jonathan Weare — for their input and guidance throughout my graduate career; my collaborators Chris Geoga and Sam Potter for their trust and effort; and numerous faculty and researchers at New York University and the Flatiron Institute for fostering an enriching and supportive academic environment, including Aleks Donev, Antoine Cerfon, Manas Rachh, Dan Fortunato, and Phil Greengard.

I am fondly indebted to Sarah Bellavance and to my friends within and outside Courant for sharing all kinds of non-mathematical joy and care. Special thanks go to Xander Beberman for providing the photo which is included in the frontispiece to this dissertation.

Lastly, I would like to thank my family for their unconditional encouragement, and for sharing the enduring belief that laboring for and sharing understanding are their own rewards.

# Abstract

The spectral properties of functions and operators provide key insights into their structure, and form the basis for state-of-the-art computational methods for simulation, learning, and inference. By providing a fast transform between physical and frequency spaces, the fast Fourier transform revolutionized applications across computational mathematics. However, there remain many settings in which spectral methods cannot be efficiently applied because the geometric or analytic structure of the problem is not directly amenable to the fast Fourier transform. Motivated by parameter estimation and sampling problems in spatial statistics and uncertainty quantification, we develop numerical methods in three such settings.

First, we discuss an adaptive integration method for computing continuous Fourier transforms of singular functions which is accelerated by existing nonuniform fast Fourier transform algorithms. Next, we develop a nonuniform fast Hankel transform for computing Fourier transforms of radially symmetric functions in higher dimensions. Finally, we present a manifold harmonic transform for performing Fourier analysis on arbitrary smooth manifolds by leveraging a multilevel low-rank approximation known as a butterfly factorization. In each case, we show how these fast transforms yield scalable methods for Gaussian random fields, and we briefly comment on other applications including imaging, graphics, and numerical partial differential equations.

# Contents

**Dedication** **iv**

**Acknowledgments** **v**

**Abstract** **vi**

**List of Figures** **x**

**1 Introduction** **1**

1.1 Gaussian random fields and spectral density functions . . . . . . . . . . . . . . . . . 1

1.2 Existing methods . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 3

1.3 Overview of the dissertation . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 6

**2 Fast adaptive Fourier integration** **8**

2.1 Introduction . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 8

2.2 Method . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 9

2.2.1 Panel integration of the Fourier integral . . . . . . . . . . . . . . . . . . . . 10

2.2.2 Acceleration with the nonuniform fast Fourier transform . . . . . . . . . . 14

2.2.3 Error estimation . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 16

2.2.3.1 Quadrature error . . . . . . . . . . . . . . . . . . . . . . . . . . . 17

2.2.3.2 Truncation error . . . . . . . . . . . . . . . . . . . . . . . . . . . 18

2.2.4 Power law singularities at the origin . . . . . . . . . . . . . . . . . . . . 21
2.2.5 Computing derivatives of the kernel . . . . . . . . . . . . . . . . . . . . 22
2.3 Numerical results . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 24
2.3.1 Error estimation for Matérn model . . . . . . . . . . . . . . . . . . . . 24
2.3.2 Relative max-norm errors and the log-likelihood . . . . . . . . . . . . . . . 25
2.3.3 Designing new spectral densities . . . . . . . . . . . . . . . . . . . . . 26
2.3.4 Singular Matérn model . . . . . . . . . . . . . . . . . . . . . . . . . 28
2.3.5 Performance for dense covariance matrix construction . . . . . . . . . . . . 32
2.4 Application . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 35
2.4.1 Preliminary analysis . . . . . . . . . . . . . . . . . . . . . . . . . . 35
2.4.2 Parameter estimation methodology . . . . . . . . . . . . . . . . . . . . 36
2.4.3 Results . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 37
2.5 Discussion . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 39

**3 A nonuniform fast Hankel transform** **43**

3.1 Introduction . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 43
3.2 Overview of the algorithm . . . . . . . . . . . . . . . . . . . . . . . . 47
3.3 Bessel function approximations . . . . . . . . . . . . . . . . . . . . . . 49
3.3.1 The Wimp expansion . . . . . . . . . . . . . . . . . . . . . . . . . . 49
3.3.2 Hankel's expansion . . . . . . . . . . . . . . . . . . . . . . . . . . . 52
3.3.3 Determining order of expansions and crossover point . . . . . . . . . . . . 53
3.4 The Nonuniform Fast Hankel Transform . . . . . . . . . . . . . . . . . . 54
3.4.1 Subdividing the matrix into blocks by expansion . . . . . . . . . . . . . . 54
3.4.2 Complexity analysis . . . . . . . . . . . . . . . . . . . . . . . . . . 56
3.5 Numerical experiments . . . . . . . . . . . . . . . . . . . . . . . . . . 60
3.5.1 Comparison to direct evaluation . . . . . . . . . . . . . . . . . . . . . 61

3.5.1.1 Asymptotic scaling . . . . . . . . . . . . . . . . . . . . . . . . 61

3.5.1.2 Impact of the order and tolerance on runtime . . . . . . . . . . . 62

3.5.1.3 Approximation error . . . . . . . . . . . . . . . . . . . . . . . . 63

3.5.2 Computing Fourier transforms of radial functions . . . . . . . . . . . . . . . 64

3.5.3 A Helmholtz solver using Fourier-Bessel expansions . . . . . . . . . . . . . 66

3.6 Discussion . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 69

**4 A butterfly-accelerated manifold harmonic transform** **71**

4.1 Introduction . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 71

4.2 The butterfly factorization . . . . . . . . . . . . . . . . . . . . . . . . . . . 74

4.3 Fiedler trees . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 77

4.4 The butterfly-accelerated manifold harmonic transform . . . . . . . . . . . . 80

4.5 Rank and complexity analysis . . . . . . . . . . . . . . . . . . . . . . . . . 81

4.6 Numerical experiments . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 92

4.6.1 Gaussian random fields on manifolds . . . . . . . . . . . . . . . . . . . . 93

4.7 Discussion . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 94

**5 Conclusion** **96**

**Bibliography** **98**

# List of Figures

2.1 Example of panel integration and resulting errors . . . 13

2.2 Demonstration of Lemma 2.1 and Theorem 2.2 . . . 21

2.3 Pointwise errors and error estimates . . . 25

2.4 Log-likelihood relative errors . . . 26

2.5 Generalized Matérn model (2.31) . . . 27

2.6 "Oscillatory" Matérn (2.32) and semi-parametric long-memory (2.33) models . . . 28

2.7 Comparison of numerical methods for evaluating the singular Matérn covariance 29

2.8 Comparison of FAFI relative covariance matrix error with requested tolerance . . 33

2.9 Comparison of runtime cost using FAFI and the trapezoidal rule . . . 34

2.10 Doppler LIDAR data from 1400-1500 UTC on June 03, 2015, at an altitude of 240m 36

2.11 Whittle MLE for the singular Matérn model . . . 36

2.12 Fitted spectral densities and resulting covariance functions . . . 39

2.13 Sample paths with identical forcing for standard and singular Matérn MLEs . . . 40

2.14 "Oscillatory" Matérn model (2.32) in two dimensions . . . 42

3.1 Bessel function and pointwise relative error using local and asymptotic expansions 48

3.2 Splitting of Hankel transform matrix into local and asymptotic regions . . . 49

3.3 NUFHT scaling with $n$, $m$, and $p$ . . . 61

3.4 NUFHT scaling with $n$ for $p = \mathcal{O}(n)$ . . . 63

3.5 NUFHT relative 2-norm error as a function of tolerance . . . . . . . . . . . . . . . . 64

3.6 Quadrature nodes and runtime comparison between NUFHT and 2D NUFFT . . . 66

3.7 Fourier-Bessel solution to Helmholtz equation on a disk . . . . . . . . . . . . . . . . 69

4.1 Eigenfunctions of the Laplacian on a cow manifold . . . . . . . . . . . . . . . . . . 72

4.2 Row tree, column tree, and complementary blocks of a butterfly matrix . . . . . . 74

4.3 Cut-cell-free Neumann Fiedler tree on a cow manifold . . . . . . . . . . . . . . . . . 81

4.4 Schematic depiction of the streaming butterfly factorization . . . . . . . . . . . . 83

4.5 Diagram of the two-dimensional Fourier transform as a MHT . . . . . . . . . . . 89

4.6 Laplace-Beltrami eigenfunctions on deformed torus . . . . . . . . . . . . . . . . . . 93

4.7 BA-MHT scaling . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 93

4.8 Samples from GRFs on deformed torus . . . . . . . . . . . . . . . . . . . . . . . . . 94

# 1 Introduction

The fast Fourier transform (FFT) [28, 33] has revolutionized a wide range of applications across mathematics, statistics, and the physical sciences by accelerating the computation of Fourier sums between equispaced points and integer frequencies. In the last thirty years, nonuniform fast Fourier transform (NUFFT) algorithms [34, 51] have been developed to extend the scalability of the FFT to Fourier sums involving nonuniform points and frequencies. The FFT and NUFFT have had a particularly strong impact on spatiotemporal statistics, where they are central components in a wide variety of scalable algorithms for estimation, sampling, and regression [32, 39, 40, 54, 92, 116] . However, the FFT and NUFFT grew out of a need to perform discrete Fourier analysis in Cartesian coordinates, and can not be directly and efficiently applied in certain statistical settings due to the analytical or geometric structure of the problem. In this dissertation, we develop quadrature schemes and novel fast transforms for three such settings, focusing on applications to parameter estimation and sampling of Gaussian random fields (GRFs).

## 1.1 Gaussian random fields and spectral density functions

GRF models are ubiquitous in many statistical settings. They provide a flexible method of interpolating noisy data by estimating and incorporating dependence structure between measurements, and they yield model-implied second-order information that can be used for uncertainty quantification. A GRF $Z$ is defined by a mean function $\mu(\boldsymbol{x}) = \mathbb{E}Z(\boldsymbol{x})$ and a positive definite covariance function

$$\operatorname{Cov}\left(Z(\boldsymbol{x}), Z(\boldsymbol{x}')\right) = K_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{x}'), \tag{1.1}$$

where $K_{\boldsymbol{\theta}}$ is some parametric family of positive definite covariance functions indexed by parameters $\boldsymbol{\theta}$. Consider observing $Z$ at locations $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in \mathbb{R}^d$ corresponding to measurements $z_i := Z(\boldsymbol{x}_i)$. Then the vector of observations $\boldsymbol{z} := [z_1, \ldots, z_n] \in \mathbb{R}^n$ is normally distributed as $\boldsymbol{z} \sim N(\boldsymbol{\mu}, \boldsymbol{\Sigma}_{\boldsymbol{\theta}})$, where $\boldsymbol{\mu}_j := \mu(\boldsymbol{x}_j)$ and the covariance matrix is given entry-wise by $(\boldsymbol{\Sigma}_{\boldsymbol{\theta}})_{jk} := K_{\boldsymbol{\theta}}(\boldsymbol{x}_j, \boldsymbol{x}_k)$. For the remainder of this dissertation, we will assume that $\mu(x) \equiv 0$, as estimating a spatially-varying mean function and subtracting it from the data is typically a relatively straightforward modeling and computational task. Furthermore, for GRFs defined on $\mathbb{R}^d$, we will make the additional simplifying assumption that the process $Z$ is *stationary*, so that $K_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{x}') = K_{\boldsymbol{\theta}}(\boldsymbol{x} - \boldsymbol{x}') = K_{\boldsymbol{\theta}}(\boldsymbol{r})$, which implies that the process $Z$ is translation-invariant. This is a more severe assumption, which limits the types of dependence structure one can express in exchange for significant theoretical and computational benefits. We refer readers to [110] for a more detailed introduction and discussion of GRFs, covariance functions, and stationarity.

Once a parametric family of covariance models has been specified for some dataset $\boldsymbol{z}$, practitioners often need to fit the model parameters $\boldsymbol{\theta}$ by, for example, computing a maximum likelihood estimator (MLE) denoted $\hat{\boldsymbol{\theta}}$, which minimizes the Gaussian negative log-likelihood

$$-2\ell(\boldsymbol{\theta}) := \log|\boldsymbol{\Sigma}_{\boldsymbol{\theta}}| + \boldsymbol{z}^\top \boldsymbol{\Sigma}_{\boldsymbol{\theta}}^{-1} \boldsymbol{z} + n \log(2\pi). \tag{1.2}$$

Naturally, evaluating the log-likelihood in this form requires evaluation of $K_{\boldsymbol{\theta}}$ in order to construct $\boldsymbol{\Sigma}_{\boldsymbol{\theta}}$. If $K_{\boldsymbol{\theta}}$ is available in closed form, this is of course no issue. Common examples of covariance functions in this category are the isotropic Matérn model $K_{\boldsymbol{\theta}}(\boldsymbol{r}) \propto \|\boldsymbol{r}\|^{\nu} \mathcal{K}_{\nu}(\|\boldsymbol{r}\|)$, where $\mathcal{K}_{\nu}$ is the modified second-kind Bessel function [90], and its special cases $K_{\boldsymbol{\theta}}(\boldsymbol{r}) \propto e^{-\|\boldsymbol{r}\|}$ for $\nu = \frac{1}{2}$ and $K_{\boldsymbol{\theta}}(\boldsymbol{r}) \propto e^{-\|\boldsymbol{r}\|^2}$ for $\nu \to \infty$. After the parameters $\boldsymbol{\theta}$ have been estimated, one can then perform regression, sampling, and other downstream tasks using the distribution specified by the fitted parameters.

The specification of the parametric family of covariance functions $K_{\boldsymbol{\theta}}$ is of paramount importance when employing Gaussian process models, as this family should capture all of the relevant correlation structure present in the data. Unfortunately, positive definite covariance functions that can be expressed in closed form are fairly difficult to construct mathematically, and thus few are known and used in practice. This severely limits the expressiveness of Gaussian process models available to practitioners, sometimes preventing domain scientists from capturing scientifically interesting phenomena. For stationary processes, however, one can alternatively construct valid covariance models by specifying their Fourier transform. Bochner's theorem [100] states that

$$K_{\boldsymbol{\theta}}(\boldsymbol{x} - \boldsymbol{x}') = \int_{\mathbb{R}^d} S_{\boldsymbol{\theta}}(\boldsymbol{\omega})\, e^{2\pi i \boldsymbol{\omega}^\top (\boldsymbol{x} - \boldsymbol{x}')} \, \mathrm{d}\boldsymbol{\omega}. \tag{1.3}$$

is a positive definite function for *any* integrable positive function $S_{\boldsymbol{\theta}}$, which we refer to as a *spectral density*. Closed form positive spectral densities are much more easily constructed and parameterized than positive definite covariance functions. Therefore, from a modeling flexibility perspective it can be desirable to build and fit models via their spectral representations.

## 1.2 Existing methods

Due to these appealing aspects of spectral modeling, many computational methods for constructing GRF models on $\mathbb{R}^d$ in the spectral domain in a variety of special cases exist in the literature. For gridded data in one or two dimensions, Whittle-type methods [125] approximate the log-likelihood (1.2) directly from $S_{\boldsymbol{\theta}}$ using an FFT, but may introduce severe bias without special care [50, 114]. For spatiotemporal processes that are gridded in at least one dimension, “half-spectral" models have been used to generate flexible kernels [30, 42, 61, 111]. For spectral densities represented using spline bases at low frequencies and algebraic tails at higher frequencies, [63] evaluates the covariance function $K_{\boldsymbol{\theta}}$ at fully nonuniform locations using the relevant special functions.

Another popular class of GRF spectral methods is the “Fourier Features” paradigm, where the Fourier integral (1.3) is approximated using Monte Carlo integration [97], the trapezoidal rule [54, 59, 109], or Gaussian quadrature [53, 106]. This yields a global approximation to the covariance matrix

$$K_\theta(\boldsymbol{x} - \boldsymbol{x}') \approx \sum_{k=1}^{m} \gamma_k S_\theta(\boldsymbol{\omega}_k) e^{2\pi i \boldsymbol{\omega}_k^\top (\boldsymbol{x}-\boldsymbol{x}')} \quad \Longrightarrow \quad \boldsymbol{\Sigma}_\theta = \mathbf{\Phi}\mathbf{D}\mathbf{\Phi}^*, \tag{1.4}$$

where $\boldsymbol{\omega}_k$ and $\gamma_k$ are quadrature nodes and weights, $\mathbf{\Phi}_{jk} := e^{2\pi i \boldsymbol{\omega}_k^\top \boldsymbol{x}_j}$ is the feature matrix, and $\mathbf{D}_{kk} := \gamma_k S_\theta(\boldsymbol{\omega}_k)$. This approach has several advantageous qualities. First, the resulting covariance matrix approximation is positive definite by construction. In addition, for sufficiently smooth processes, $S_\theta$ decays rapidly, and thus this representation is low-rank to high accuracy. The log-likelihood (1.2) can then be efficiently evaluated using Sherman-Morrison-Woodbury identities. However, a large number of Fourier features are needed to accurately approximate models whose spectral densities decay very slowly, and thus these approaches can become expensive when modeling rough processes. We propose an alternative method which is better suited to the non-smooth regime in Chapters 2 and 3.

For GRFs on compact manifolds $\mathcal{M}$, specifying *any* valid covariance functions becomes challenging. The naïve approach of taking a positive definite function on $\mathbb{R}^d$ and replacing Euclidean distance with geodesic distance does *not* yield a positive definite function on $\mathcal{M}$ [17]. Thus a popular approach is to model the process $Z$ as the solution to a stochastic partial differential equation (SPDE) $\mathscr{L}Z = W$ on $\mathcal{M}$, where $\mathscr{L}$ is a differential operator and $W$ is Gaussian white noise [14, 76]. The SPDE framework can be made computationally efficient using a triangulation of $\mathcal{M}$, a finite element method (FEM) discretization of $\mathscr{L}$, and common FEM sparse matrix methods, although one must carefully monitor FEM discretization error [102]. However, this paradigm is typically limited to processes for which the relevant differential operator can be written in weak form and discretized using FEM. This limitation has restricted the set of models used in practice, just as we noted for covariance-space modeling. It is worth mentioning, however, that significant progress towards more expressive SPDE models has been made in recent years using polynomial and rational approximations to the spectrum [14, 15, 70].

In order to construct GRF models on compact manifolds $\mathcal{M}$ in spectral space as we did above in $\mathbb{R}^d$, one can apply an analogue to Bochner's theorem on $\mathcal{M}$. Note that the complex exponentials $e^{2\pi i \boldsymbol{\omega}^\top \boldsymbol{r}}$ that appear in (1.3) are exactly the eigenfunctions of the Laplacian on $\mathbb{R}^d$. It is therefore natural to consider an analogous construction using the eigenfunctions of the Laplace-Beltrami operator on an arbitrary manifolds in place of the complex exponentials. On compact manifolds $\mathcal{M}$ this collection of eigenfunctions is discrete, and thus the integral in (1.3) is replaced by a sum, and we can express covariances in terms of any positive spectral density $S_{\boldsymbol{\theta}}$ as

$$K_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{x}') = \sum_{k=1}^{\infty} S_{\boldsymbol{\theta}}(\lambda_k) \varphi_k(\boldsymbol{x}) \overline{\varphi_k(\boldsymbol{x}')} \tag{1.5}$$

where $(\lambda_k, \varphi_k)$ for $k = 1, \ldots, \infty$ are eigenvalues and corresponding eigenfunctions of the Laplace-Beltrami operator $\Delta_{\mathcal{M}}$. Expansions of this type are suggested in [17, 108] as natural generalizations of the Fourier features paradigm to the manifold setting, but only a small number of features $\varphi_k$ are used so that the resulting covariances remain low-rank. Similar spectral representations are used for expositional purposes in [70, 71], although SPDE-based computational tools are then used.

While the SPDE approach is typically viewed as a means of accelerating computation for standard covariance models like the Matérn, if one considers the spectrum of the differential operator $\mathscr{L}$, it can be seen as a version of the spectral modeling paradigm presented above and in Section 1.1. Consider, for example, the Matérn SPDE

$$(\rho^2 - \Delta)^{\frac{\nu}{2}+\frac{d}{4}} Z(\boldsymbol{x}) := \mathcal{F}^{-1}\left\{(\rho^2 + \lambda_k)^{\frac{\nu}{2}+\frac{d}{4}} (\mathcal{F}Z)(k)\right\}(\boldsymbol{x}) = W(\boldsymbol{x}), \tag{1.6}$$

where the Fourier transform of $Z$ is defined in a suitable weak sense. We see from this example that the differential operator $\mathscr{L}$ is defined simply from the inverse square root of the corresponding Matérn spectrum $S_{\boldsymbol{\theta}}(\lambda) = (\rho^2 + \lambda)^{-\nu-\frac{d}{2}}$. This equivalence is not specific to the Matérn, and we have $\mathscr{L}f = \mathcal{F}^{-1}\left\{S_{\boldsymbol{\theta}}(\lambda_k)^{-\frac{1}{2}}(\mathcal{F}f)(k)\right\}$ for any sufficiently smooth $f$ and sufficiently regular $S_{\boldsymbol{\theta}}$. Therefore essentially any GRF that can be expressed as an SPDE can be equivalently represented by its spectral density, and the difference between the approaches is purely in the choice of computational tools — weak form SPDEs and FEM discretizations versus spectral densities and oscillatory discrete transforms. [76, Appendix B.3.2] claims that this spectral definition is not computationally useful because linear combinations of Laplace-Beltrami eigenfunctions $\varphi_k$ cannot be efficiently computed. We introduce a method for accelerating precisely this computation in Chapter 4.

## 1.3 Overview of the dissertation

Although specifying GRFs by their spectral densities in these various geometric settings can significantly increase modeling flexibility, it presents certain computational challenges. In particular, one must efficiently evaluate (1.3) and its analogues to move from spectral space, in which the statistical model is defined, to physical space, in which the data are found. The development of quadrature schemes and fast algorithms for these purposes is precisely the focus of this dissertation.

In Chapter 2, we introduce a Gaussian panel quadrature framework for efficiently and accurately evaluating covariance functions and their derivatives at irregular locations in one dimension directly from any continuous, integrable spectral density $S_{\boldsymbol{\theta}}$. This method is fully adaptive and accelerated by the NUFFT.

In Chapter 3, we develop the fast transform necessary to extend the methods of Chapter 2 to higher dimensions. In particular, when working with an isotropic GRF one has $S_{\boldsymbol{\theta}}(\boldsymbol{\omega}) = S_{\boldsymbol{\theta}}(\|\boldsymbol{\omega}\|)$, and the $d$-dimensional Fourier integral (1.3) simplifies to a one-dimensional Hankel transform

$$K_{\boldsymbol{\theta}}(\boldsymbol{r}) = \frac{(2\pi)^{\frac{d}{2}}}{r^{\frac{d}{2}-1}} \int_0^\infty S_{\boldsymbol{\theta}}(\omega)\, J_{\frac{d}{2}-1}(\omega r)\, \omega^{\frac{d}{2}}\ \mathrm{d}\omega, \tag{1.7}$$

where $\omega := \|\boldsymbol{\omega}\|$ and $r := \|\boldsymbol{r}\|$. We use local and asymptotic expansions of the Bessel function $J_\nu$ to develop a nonuniform fast Hankel transform (NUFHT) to evaluate discretized forms of (1.7) given by

$$g_j = \sum_{k=1}^{m} c_k J_\nu(\omega_k r_j) \qquad \text{for } j = 1, \ldots, n \tag{1.8}$$

in $\mathcal{O}\big((m+n)\log\min(n,m)\big)$ time to any user-specified tolerance $\varepsilon$. We also discuss applications of the NUFHT beyond statistics to numerical PDE.

In Chapter 4, we propose a manifold harmonic transform (MHT) to accelerate sampling of GRFs on arbitrary compact manifolds $\mathcal{M}$ by evaluating sums of the form

$$g_j = \sum_{k=1}^{m} c_k \varphi_k(\boldsymbol{x}_j), \quad \text{for } j = 1, \ldots, n. \tag{1.9}$$

We build a compressed representation of the matrix $\boldsymbol{\Phi}_{jk} = \varphi_k(\boldsymbol{x}_j)$ using a multilevel low-rank factorization for oscillatory operators known as a butterfly factorization [75, 89]. For 2-manifolds in $\mathbb{R}^3$, we numerically demonstrate $\mathcal{O}(n + m^{3/2})$ asymptotic scaling for this method, and establish theoretical guarantees which form the foundation for a more rigorous proof of this complexity in a special case. We briefly mention applications of the butterfly-accelerated manifold harmonic transform (BA-MHT) to numerical PDE, graphics, and machine learning.

# 2 | Fast adaptive Fourier integration

## 2.1 Introduction

In this chapter, we develop a computational method for computing covariances $K_{\boldsymbol{\theta}}(r)$ from any integrable, positive spectral density $S_{\boldsymbol{\theta}}$ using adaptive Gaussian quadrature. By focusing on evaluating entries of the covariance matrix $\boldsymbol{\Sigma}_{\boldsymbol{\theta}}$, we provide a tool for performing maximum likelihood estimation from a general spectral density $S_{\boldsymbol{\theta}}$ with fully irregularly sampled data that is compatible with a variety of subsequent likelihood approximations. We demonstrate that panel Gaussian integration of the Fourier integral allows the accurate and efficient computation of covariances $K(r)$ and their derivatives $\frac{\partial}{\partial \theta_j} K_{\boldsymbol{\theta}}(r)$ from *any* continuous, integrable spectral density $S_{\boldsymbol{\theta}}$. This in turn facilitates gradient-based maximum likelihood estimation directly on any parameterization of the spectral density. By taking advantage of modern NUFFT and automatic differentiation (AD) [55] methods, along with careful analysis of the tail behavior of the spectral density, we can compute $K_{\boldsymbol{\theta}}(r)$ and its derivatives with respect to parameters to $\varepsilon = 10^{-12}$ accuracy even for slowly decaying spectral densities at millions of inter-observation distances $r$ in seconds on a standard laptop. Our free and open source `Julia` code is available at `https://github.com/pbeckman/SpectralKernels.jl`.

While the possibilities for functional forms of spectral densities are endless and the machinery described here is generally applicable, in this chapter we study in detail the simple extension given by incorporating an integrable singularity into a bounded spectral density $S_{\boldsymbol{\theta}}(\omega)$, resulting in the new model $|\omega|^{-\alpha} S_{\boldsymbol{\theta}}(\omega)$. Such extensions correspond to "long memory" processes, for which we derive some theoretical properties, overcome numerical challenges associated with evaluating their covariance functions, and demonstrate their practical value by fitting a singular model to Doppler LiDAR wind velocity profiles. For the remainder of this chapter we will often suppress the dependence of $\boldsymbol{\Sigma}$, $K$, and $S$ on $\boldsymbol{\theta}$ for notational clarity.

## 2.2 Method

We are concerned here only with real-valued covariance functions $K$, and therefore assume $S$ is an even function. This results in the simplification

$$K(r) = 2 \int_0^\infty S(\omega) \cos(2\pi\omega r) \, d\omega. \tag{2.1}$$

In order to evaluate $K(r)$ by directly computing (2.1), one must choose a quadrature rule. As the spectral density $S$ is assumed to be integrable, it must decay sufficiently fast for large $\omega$, so that one can truncate the infinite interval $[0, \infty)$ at some point $b$ and integrate only on the finite interval $[0, b]$. This gives

$$K(r) \approx 2 \int_0^b S(\omega) \cos(2\pi\omega r) \, d\omega \approx 2 \sum_{j=1}^{m} \gamma_j S(\omega_j) \cos(2\pi\omega_j r), \tag{2.2}$$

where $\omega_j$ and $\gamma_j$ are the nodes and weights of the chosen quadrature rule. The simplest choice of quadrature is the trapezoidal rule, which uses equispaced points $\omega_j = (j-1)h$ with weights $\gamma_1 = \gamma_m = \frac{h}{2}$ and $\gamma_j = h$ for $j = 2, \ldots, m-1$, where the grid spacing is $h := \frac{b}{m-1}$. The aliasing and truncation errors when using the trapezoidal rule are treated in detail for the squared exponential and Matérn models in [2].

For known spectral densities $S$ with sufficiently fast decay, this can be a highly accurate quadrature. However, it has two limitations. First, for more flexible and complex spectral densities $S$, the analysis used in [2] to choose the grid spacing $h$ necessary to resolve $S$ becomes difficult, and must be done for each new parametric family $S$. Second, for small $r$ we may need to take both a small $h$ to control the quadrature error, as well as a large $b$ to control the truncation error when integrating slowly decaying spectral densities $S$ such as the commonly used Matérn model [110]

$$S_{\boldsymbol{\theta}}(\omega) = \varphi^2(\rho^2 + \omega^2)^{-\nu-\frac{1}{2}}, \quad \boldsymbol{\theta} := \{\varphi, \rho, \nu\} \tag{2.3}$$

with small values of the smoothness parameter $\nu$, e.g. $\nu = 1/2$. This can result in a number of quadrature nodes $m$ which is prohibitively large from a computational standpoint. In the remainder of this section, we demonstrate that panel Gaussian integration of the Fourier integral allows us to overcome both of these limitations.

### 2.2.1 Panel integration of the Fourier integral

If one wishes to integrate (2.1) for a large range of $r$ with uniform accuracy, using a dense quadrature rule for $\omega \in [0, b]$ for some large $b$ might appear to be an unavoidable cost. However, as we demonstrate numerically in this section and theoretically in Section 2.2.3.2, truncation error in $K(r)$ is concentrated near the origin in $r$-space. Intuitively, this is because the high frequency information contained in the tails of the spectral density $S$ has a much greater impact on covariances between observations which are close together. This intuition will be made precise shortly by Theorem 2.2.

The localization of error near $r = 0$ suggests that we can compute the Fourier integral on the interval $\omega \in [0, b]$, then iteratively add the contribution of the Fourier integral on successive intervals in $\omega$-space for *only* those $r$'s nearest the origin in $r$-space. We will refer to each interval

$[a, b]$ in $\omega$-space as a *panel*. Define the exact and discretized panel integrals

$$I_{[a,b]}(r) := \int_a^b S(\omega)\cos(2\pi\omega r)\, d\omega \tag{2.4}$$

$$\tilde{I}^{(m)}_{[a,b]}(r) := \sum_{j=1}^{m} \gamma_j S(\omega_j)\cos(2\pi\omega_j r) \tag{2.5}$$

where $\omega_j$ and $\gamma_j$ are nodes and weights of an $m$-point quadrature rule on $[a, b]$. The error

$$E^{(m)}_{[a,b]}(r) := \left| I_{[a,b]}(r) - \tilde{I}^{(m)}_{[a,b]}(r) \right| \tag{2.6}$$

in the $m$-point trapezoidal rule is only order $O(m^{-2})$ when the integrand is non-periodic, as is the case in this panel integral setting. However, the $m$-point Gauss-Legendre rule, which integrates polynomials of degree $2m - 1$ exactly, is generally much more accurate for smooth, non-periodic functions. In particular, for $f \in C^{(s)}([a, b])$ with mild additional smoothness assumptions, an $m$-order Gauss-Legendre quadrature rule will have error of order $O(m^{-2s-1})$ [118, Theorem 19.4]. Gauss-Legendre quadrature is thus amenable to panel integration and adaptivity, which allows us to accurately discretize Fourier integrals for any continuous, integrable spectral density. Adaptive quadrature will be treated in greater detail in Section 2.2.3.1.

By the Nyquist-Shannon sampling theorem, one requires at least $m = 2r \cdot (b - a)$ values to completely determine a function $f(\omega)$ on the interval $\omega \in [a, b]$ if it has bandlimit $r$, that is, if $\mathcal{F}\{f\}(t) \equiv 0$ for all $|t| > r$. Therefore, for a desired accuracy $\delta$, a given $m$, and sufficiently smooth $S$, we expect an $O(m)$-node Gauss-Legendre rule to compute the Fourier integral with integrand $f(\omega) = S(\omega)\cos(2\pi\omega r)$ to within accuracy $\delta$ on any interval of length $b - a = \frac{m}{2r}$. We use this heuristic to choose the next panel $[a, b]$ in $\omega$-space, where $r$ is taken to be the largest distance for which $K(r)$ has not yet converged. Then all $r' < r$ result in less oscillatory integrands, which are therefore also accurately resolved by the $m$-point rule.

It is worth noting that this Nyquist-informed heuristic for panel selection is not tight in either direction. If $S$ contains sharp features, then the integrand $f(\omega) = S(\omega)\cos(2\pi\omega r)$ will not have bandlimit $r$, and thus more quadrature nodes may be required to achieve the desired accuracy $\delta$. Conversely, if $S$ is very smooth, then the effect of aliasing may be below the desired accuracy $\delta$, and thus fewer quadrature nodes may still yield a adequately accurate result despite not fully resolving the integrand. However, for sufficiently large $m$ we find that this Nyquist-based choice typically computes the Fourier integral to double precision with neither significant redundant oversampling nor the need for further refinement where $S$ is smooth.

The novelty and efficiency of our approach stems from the two related mechanisms discussed above. First, we can reduce the number of $r$ values to be computed after adding each panel Fourier integral, as the largest $r$'s have converged. This reduces the number of points $n$ at which we must evaluate the sum (2.5). Second, because $\cos(2\pi\omega r)$ is less oscillatory in $\omega$ for smaller $r$, as the largest $r$'s converge we can take increasingly large panels while keeping the number of oscillations per panels constant. Therefore we use the same $m$-point quadrature rule to accurately resolve the integrand. This can significantly reduce the total number of quadrature nodes used to evaluate the Fourier integral when compared to a more uniform quadrature scheme. Figure 2.1 provides a visual example of this panel growth as $K(r)$ is resolved for the largest $r$'s which reduces the highest remaining Nyquist frequency. In particular, note that the spacing between quadrature nodes increases by almost two orders of magnitude between the first and fourth panels, corresponding to a proportional reduction in computational effort relative to using, for example, the trapezoidal rule to integrate the same interval in $\omega$-space. This trend only continues as we progress in $\omega$-space, adaptively generating a highly non-uniform quadrature rule with increasingly sparse nodes as $\omega$ increases. The resulting quadrature provides orders-of-magnitude speedups over alternatives, as we demonstrate in Section 2.3.5.

Computing the sum (2.5) directly for $n$ distances with an $m$-point quadrature rule has $O(nm)$ complexity for each panel. This ostensibly introduces a major tension in the computation of these panel integrals: higher-order quadrature rules enable larger panels and faster convergence,

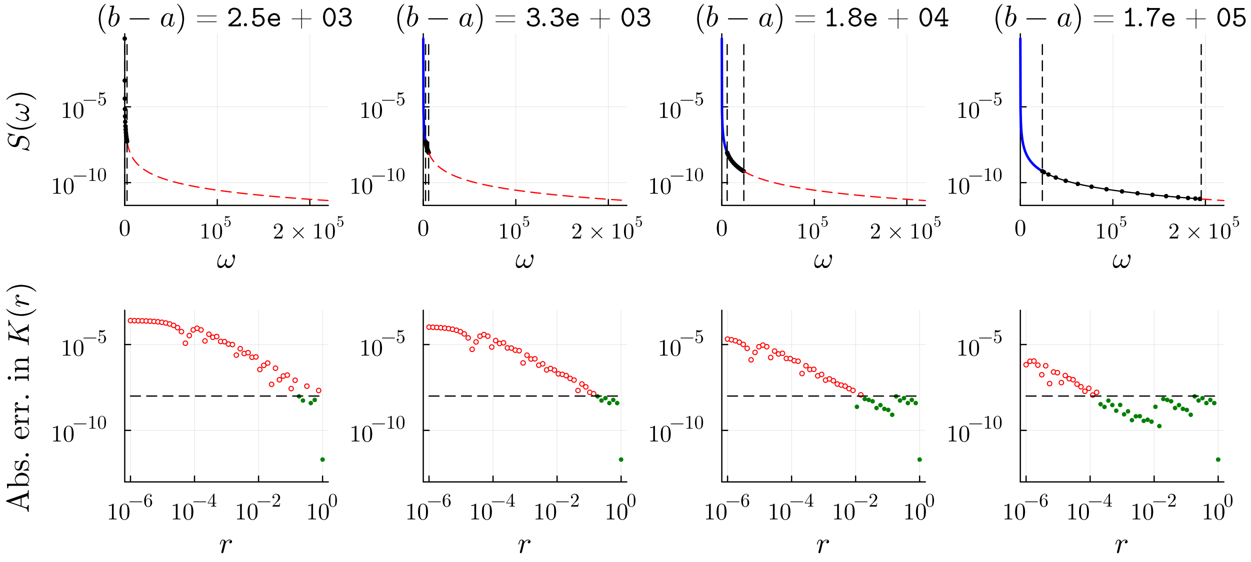


**Figure 2.1:** Panel integration of a Matérn spectral density with $\nu = \frac{1}{2}$, $\rho = 1$, and $\varphi$ chosen so that $K(0) = 1$. The top row shows various panels of the spectral density being integrated, and each corresponding plot in the bottom row shows the absolute truncation error in the kernel after the panel integral above has been added. A small subset of the quadrature nodes are shown in each panel. A tolerance of $\varepsilon = 10^{-8}$ and $m = 5{,}000$ nodes per panel are used.

but particularly for large data sizes they lead to enormously burdensome computations if done directly. In this next section, we discuss how the NUFFT can be used to relieve this tension by reducing the cost of each panel integral to $O(m + n \log n)$ — a speedup which is imperative to making Fourier integration computationally tractable.

**Remark 2.1.** The idea of using the same number of quadrature nodes $m$ to discretize Fourier integrals for which the space-frequency product $r \cdot (b - a)$ is constant is reminiscent of the complementary low-rank property used in butterfly algorithms [74, 89]. One could view the present method as a butterfly-like algorithm for which the error control is designed specifically for the problem of adaptively computing continuous Fourier transforms of slowly decaying functions.

**Remark 2.2.** While globally low-rank methods like EFGP are of the form

$$\widetilde{\Sigma} = \mathbf{\Phi} \mathbf{S} \mathbf{\Phi}^*, \tag{2.7}$$

panel integration yields

$$\widetilde{\Sigma} = \mathbf{\Phi}_0 \mathbf{S}_0 \mathbf{\Phi}_0^* + \boldsymbol{A}_1 \odot (\mathbf{\Phi}_1 \mathbf{S}_1 \mathbf{\Phi}_1^*) + \cdots + \boldsymbol{A}_p \odot (\mathbf{\Phi}_p \mathbf{S}_p \mathbf{\Phi}_p^*) \tag{2.8}$$

where $\odot$ is the pointwise or Hadamard product, and the matrix $\boldsymbol{A}_\ell \in \{0, 1\}^{n\times n}$ indicates to which entries $\Sigma_{jk} = K(x_j - x_k)$ the contribution of panel $\ell$ should be added. If $\boldsymbol{A}_\ell \succ 0$ for all $\ell = 1, \ldots, p$ then $\widetilde{\Sigma} \succ 0$, as Hadamard products and linear combinations both preserve positive definiteness. This is not true in general, and thus our method does not guarantee positive definiteness. However, because we can obtain very high accuracies, positive definiteness is typically not an issue in practice. Alternative choices of $\boldsymbol{A}_\ell$ which are positive definite, for example compactly-supported functions like those of [123], remain an interesting direction for future research.

### 2.2.2 Acceleration with the nonuniform fast Fourier transform

As each panel Fourier integral (2.5) is a sum of cosines with nonuniform frequencies $\omega_1, \ldots, \omega_m$ evaluated at nonuniform distances $r_1, \ldots, r_n$, it can be computed as the real part of the exponential sum

$$f_k = \sum_{j=1}^{m} \gamma_j S(\omega_j) e^{2\pi i \omega_j r_k} \qquad k = 1, \ldots, n \tag{2.9}$$

so that $\tilde{I}^{(m)}_{[a,b]}(r_k) = \mathrm{Re}(f_k)$. Equivalently, the computation of the Fourier integral for each panel $[a, b]$ in $\omega$-space at all unconverged distances $r$ can be viewed as a matrix-vector product of a dense $n \times m$ nonuniform discrete Fourier matrix with a vector of $m$ evaluations of the spectral density along with appropriate quadrature weights given by

$$n \text{ distances} \left\{ \underbrace{\begin{bmatrix} e^{2\pi i\omega_1 r_1} & \cdots & e^{2\pi i\omega_m r_1} \\ \vdots & \ddots & \vdots \\ e^{2\pi i\omega_1 r_n} & \cdots & e^{2\pi i\omega_m r_n} \end{bmatrix}}_{m \text{ quadrature nodes}} \right. \begin{bmatrix} \gamma_1 S(\omega_1) \\ \vdots \\ \gamma_m S(\omega_m) \end{bmatrix} = \begin{bmatrix} f_1 \\ \vdots \\ f_k \end{bmatrix}. \tag{2.10}$$

Computing this dense matrix-vector product directly has a cost of $O(nm)$. However, the sum (2.9) and the matrix-vector product (2.10) are equivalent views of exactly a "type 3" nonuniform to nonuniform discrete Fourier transform, which can be evaluated to accuracy $\delta$ in $O(m+n\log n)$ complexity using the NUFFT. Most existing NUFFT algorithms work by convolving the input data with some "spreading" function, performing an equispaced FFT on a fine grid, and deconvolving to obtain values at the desired output locations. The development of efficient NUFFT libraries with tunable accuracy guarantees has been the subject of significant research in the past few decades. See for example [3, 34, 52].

In the present context of computing pointwise covariances from the spectral density, the improved scaling of the NUFFT when compared to direct summation is essential to facilitating the use of larger, more accurate quadrature rules to compute the values of each panel integral at a greater number of distances. The dramatic computational impact of the NUFFT will be studied in greater detail in Section 2.3.5, where we observe orders of magnitude speedup for practical data sizes.

**Remark 2.3.** The number of points $m$ in each NUFFT can be tuned to improve performance. Generally, using large NUFFTs helps to amortize setup costs and take advantage of multi-threading. However, ill-conditioning and round-off errors limit the accuracy of the NUFFT to about $10^{-9}$ for large inputs [3, Remark 9]. So if higher accuracy is needed, smaller NUFFTs must be used at the expense of a larger pre-factor.

### 2.2.3 Error estimation

As we saw in Section 2.2.1, the key idea behind fast panel integration of Fourier integrals is to use panels of increasing size in $\omega$-space. This is made possible by choosing a working tolerance $\delta$ and sequentially integrating panels in $\omega$-space for *only* those distances $r$ at which $K(r)$ has not yet been resolved to within $\delta$ accuracy. Determining when $K(r)$ is adequately resolved for each $r$ requires a careful analysis of multiple sources of error, and naive stopping criteria often result in the loss of several digits of accuracy in the computed integral.

Recall that for any valid covariance function $K$, one has that $K(0) \geq |K(r)|$ for all $r > 0$. Then for any set of locations $x_1, \ldots, x_n$, computing each entry in an approximate covariance matrix $\widetilde{\mathbf{\Sigma}}_{ij} := \widetilde{K}(x_i - x_j)$ such that the pointwise error relative to $K(0)$ is controlled $|K(r) - \widetilde{K}(r)| \,/\, K(0) < \varepsilon$ for all $r$, consequently bounds the relative max-norm error in $\widetilde{\mathbf{\Sigma}}$

$$\frac{\|\mathbf{\Sigma} - \widetilde{\mathbf{\Sigma}}\|_{\max}}{\|\mathbf{\Sigma}\|_{\max}} = \frac{\max\limits_{1 \leq i,j \leq n} \left| K(x_i - x_j) - \widetilde{K}(x_i - x_j) \right|}{K(0)} < \varepsilon. \tag{2.11}$$

By the equivalence of norms, controlling the relative max-norm error also controls the relative Frobenius and spectral norm errors, up to constants which may depend on $n$.

We emphasize that the pointwise error relative to $K(0)$ is equivalent to neither the relative *nor* the absolute pointwise error in each $K(r)$. If $K(r) = 10^{-12}$, then *relative* error $|K(r) - \widetilde{K}(r)|/|K(r)| < 10^{-8}$ would require cancellation in the sum of panel integrals to 20 digits, which is impossible in general in double precision. Conversely, if $K(r) = \texttt{1e12}$, then *absolute* error $|K(r) - \widetilde{K}(r)| < 10^{-8}$ would require 20 correct digits, which is again impossible in double precision. These are standard and well-known limitations of adaptive integration in finite precision arithmetic, and neither bounding the relative nor absolute pointwise error by $\varepsilon$ are necessary conditions for bounding the resulting error in log-likelihood evaluation by $\varepsilon$.

As our final goal is to evaluate the log-likelihood, we argue that controlling relative norm errors in $\widetilde{\mathbf{\Sigma}}$ is a relevant metric, as small pointwise errors in $\mathbf{\Sigma}$ will result — at least asymptotically — in small log-likelihood errors because the log-likelihood is a smooth function of the entries of the covariance matrix $\mathbf{\Sigma}$ for fixed data $\boldsymbol{y}$. However, other choices of error metric are of course possible, and we emphasize that the connection between pointwise kernel accuracy and the accuracy of the resulting log-likelihood is not direct. If the true covariance matrix is poorly conditioned, then the log-likelihood may be computed to significantly fewer than 16 digits of accuracy purely due to fundamental limitations of double-precision arithmetic, *even if every kernel evaluation is accurate to machine precision.* See Section 2.3.2 for a brief numerical study of the impact of pointwise errors and ill-conditioning on log-likelihood accuracy. Since this chapter is focused on kernel evaluation, we leave a more thorough error analysis of the Gaussian log-likelihood to future work. Returning to the problem of error estimation within our method, we note that there are two sources of error at play which we seek to bound above by $\delta$ — quadrature error and truncation error — which we now treat individually.

#### 2.2.3.1 Quadrature error

To control the quadrature error, we use a straightforward adaptive approach. As is standard practice in adaptive integration, we estimate the error in the discretized integral by comparing the result to that obtained by a higher order quadrature rule. For this purpose we use a $2m$-point Gauss-Legendre rule. The error in the $m$-point rule on $[a, b]$ is then approximated as

$$\widetilde{E}^{(m)}_{[a,b]}(r) := \left|\tilde{I}^{(2m)}_{[a,b]}(r) - \tilde{I}^{(m)}_{[a,b]}(r)\right| \approx \left|I_{[a,b]}(r) - \tilde{I}^{(m)}_{[a,b]}(r)\right|. \tag{2.12}$$

For a given panel, if $\widetilde{E}^{(m)}_{[a,b]}(r) > \delta$ for any unconverged distance $r$, we divide $[a, b]$ in two and repeat this procedure separately on $[a, (a + b)/2]$ and $[(a + b)/2, b]$. We deem these subpanels converged when both $\widetilde{E}^{(m)}_{[a,(a+b)/2]}(r) < \delta/2$ and $\widetilde{E}^{(m)}_{[(a+b)/2,b]}(r) < \delta/2$, so that the total error on

$[a, b]$ remains bounded by $\delta$ for all unconverged $r$'s. If the error remains large for any $r$ in one or both of these subpanels, we continue this dyadic refinement and proportional tolerance reduction until we obtain an approximation $\tilde{I}_{[a,b]}(r)$ as a sum of subpanel integrals such that the total quadrature error is bounded by $\delta$. This is again standard practice for adaptive integration, and can be performed to high accuracy. See for example [49].

#### 2.2.3.2 Truncation error

Controlling the truncation error, given by

$$E_{\text{trunc}}(b, r) := \int_b^\infty S(\omega) \cos(2\pi\omega r)\, d\omega, \tag{2.13}$$

is a more subtle issue. As discussed in Section 2.2.1, we iteratively integrate panels from zero to infinity in spectral space. The remaining question is how to determine for each $r$ when we have integrated a sufficient interval $[0, b]$ in spectral space so that $|E_{\text{trunc}}(b, r)| < \delta$ and we can cease adding new panels.

One could check that the contribution of the current panel is less than the tolerance $\delta$, that is $\tilde{I}^{(m)}_{[a,b]}(r) < \delta$. This is a necessary but *not* a sufficient condition for $E_{\text{trunc}}(b, r) < \delta$. In practice, for spectral densities which decay exponentially this condition results in a negligible loss of accuracy. However, for spectral densities with slow algebraic decay, terminating integration when $\tilde{I}^{(m)}_{[a,b]}(r) < \delta$ may result in errors significantly greater than $\delta$, as one would be truncating many panels whose contributions would each be $\mathcal{O}(\delta)$.

Therefore, for better error control, we consider the exponent $\beta$ and constant $c$ in the tails of the spectral density such that $S(\omega) \sim c\omega^{-\beta}$ as $r \to \infty$. For many spectral densities $c$ and $\beta$ can be derived analytically, often by straightforward means. Otherwise, these coefficients can be estimated using linear least squares in log space. With these values, we can compute the truncation error analytically for the resulting power law tail, which gives

$$|E_{\text{trunc}}(b,r)| \approx \left| \int_b^\infty c\omega^{-\beta} \cos(2\pi\omega r)\, d\omega \right| \tag{2.14}$$

$$= -c(2\pi r)^{\beta-1} \operatorname{Re}\left( (-i)^{\beta-1} \Gamma(-\beta+1, -2\pi i b r) \right), \tag{2.15}$$

where $\Gamma(s,z) := \int_z^\infty t^{s-1}e^{-t}\, dt$, is the upper incomplete Gamma function [90]. There exist a number of libraries to numerically evaluate this special function. However, it is helpful to have a simple, tight, and easily invertible algebraic upper bound on this truncation error. We now derive such a bound as a consequence of the following lemma.

**Lemma 2.1.** *For any $s, y > 0$,*

$$|\Gamma(-s, iy)| \leq \min\left( y^{-s-1}, \frac{y^{-s}}{s} \right). \tag{2.16}$$

*Proof.* Using the integral representation 8.6.4 from [90],

$$|\Gamma(-s, iy)| = \left| \frac{(iy)^{-s}e^{-iy}}{\Gamma(s+1)} \int_0^\infty \frac{t^s e^{-t}}{t+iy}\, dt \right| \tag{2.17}$$

$$\leq \frac{y^{-s}}{\Gamma(s+1)} \int_0^\infty \frac{t^s e^{-t}}{|t+iy|}\, dt \tag{2.18}$$

$$\leq \frac{y^{-s}}{\Gamma(s+1)} \int_0^\infty \frac{t^s e^{-t}}{y}\, dt \tag{2.19}$$

$$= \frac{y^{-s-1}}{\Gamma(s+1)} \int_0^\infty t^s e^{-t}\, dt \tag{2.20}$$

$$= y^{-s-1}, \tag{2.21}$$

where $\Gamma(z) := \Gamma(0, z)$ is the usual gamma function. By a similar argument, we obtain

$$|\Gamma(-s, iy)| \leq \frac{y^{-s}}{\Gamma(s+1)} \int_0^\infty \frac{t^s e^{-t}}{t}\, dt \tag{2.22}$$

$$= \frac{y^{-s}}{\Gamma(s+1)} \Gamma(s) \tag{2.23}$$

$$= \frac{y^{-s}}{s} \tag{2.24}$$

using the fundamental property of the Gamma function $\Gamma(z+1) = z\Gamma(z)$. □

The above lemma leads immediately to the following result.

**Theorem 2.2.** *For any $\beta > 1$ and any $b, r > 0$,*

$$\left| \int_b^\infty \omega^{-\beta} e^{-2\pi i \omega r}\, d\omega \right| \leq \min\left( \frac{1}{\beta - 1} b^{-\beta+1}, \frac{1}{2\pi r} b^{-\beta} \right), \tag{2.25}$$

*where the first term gives a tighter bound when $br \leq \frac{\beta-1}{2\pi}$, and the second term otherwise.*

*Proof.* Taking the change of variables $t = 2\pi i \omega r$, we obtain

$$\left| \int_b^\infty \omega^{-\beta} e^{-2\pi i \omega r}\, d\omega \right| = \left| (2\pi i r)^{\beta-1} \int_{2\pi i b r}^\infty t^d e^{-t}\, dt \right| = (2\pi r)^{\beta-1} \left| \Gamma(-\beta + 1, 2\pi i b r) \right|.$$

Applying Lemma 2.1 gives the desired result. □

This result indicates that when the product $br$ is small (for close together observations or limited integration domains in spectral space), the truncation error in the Fourier integral behaves like the truncation error in the non-oscillatory integral. But when $br$ is large (for far apart observations or large integration domains in spectral space), the truncation error decays faster by an additional power of $b$, with a constant that decreases for larger $r$. This makes precise the intuition of Figure 2.1 that for a fixed integration domain, the truncation error for larger distances $r$ is lower.

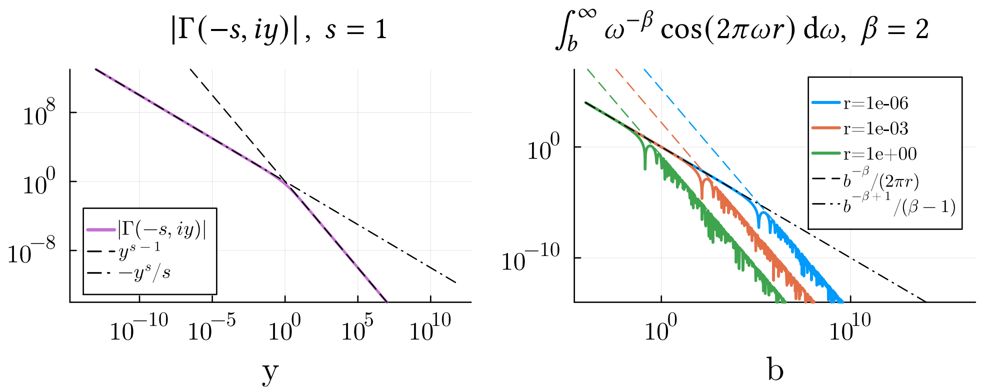


**Figure 2.2:** The incomplete Gamma function bound from Lemma 2.1 (left) and corresponding bound on the power law truncation error given by Theorem 2.2 for various $r$ (right).

One can therefore use the analytic formula (2.14) or the simpler algebraic upper bound (2.25) to control the truncation error. In conjunction with the adaptive integration and quadrature error estimation described in Section 2.2.3.1, we obtain accurate estimates of the total error in the computed Fourier integral.

### 2.2.4 Power law singularities at the origin

Long-memory Gaussian processes can be characterized by having an integrable singularity in their spectral density $S$ at the origin. Such models typically lack closed form expressions for the resulting kernel $K$, and even when such expressions are available, they are often difficult to compute numerically. We will focus here on a modification of the Matérn family that we will call a “singular" Matérn model, which was first proposed in [96] and corresponds to a spectral density given by

$$S_{\boldsymbol{\theta}}(\omega) = \varphi^2|\omega|^{-\alpha}(\rho^2 + \omega^2)^{-\nu-\frac{1}{2}}, \qquad \boldsymbol{\theta} := \{\varphi, \alpha, \rho, \nu\} \tag{2.26}$$

with $0 \leq \alpha < 1$. As will be discussed in detail in the next section, the corresponding covariance function does technically have a closed form representation, but it is exceptionally challenging to evaluate numerically.

In contrast, such spectral densities with power law singularities are treated easily by the panel Gaussian quadrature framework presented here. While directly applying a Gauss-Legendre quadrature to the integral

$$K(r) = 2 \int_0^b \omega^{-\alpha} S(\omega) \cos(2\pi\omega r) \, d\omega \tag{2.27}$$

results in low accuracy, Gauss-Jacobi quadratures are designed specifically to treat the singularity $\omega^{-\alpha}$ accurately, and can be computed quickly and accurately [47, 57]. Simply using a Gauss-Jacobi rule on the first panel followed by Gauss-Legendre rules on all remaining panels allows accurate computation of singular Fourier integrals of this form. For panels that do not contain the origin, the $\omega^{-\alpha}$ term may safely be included in the Fourier integrand, and the standard Gauss-Legendre rule gives high accuracy.

### 2.2.5 Computing derivatives of the kernel

In order to take advantage of gradient-based optimizers for maximum likelihood estimation, one must compute derivatives $\frac{\partial}{\partial \theta_j} K(r)$. Assuming that the parametric family $S_{\boldsymbol{\theta}}$ is sufficiently well-behaved as to allow exchanging differentiation and integration, we have

$$\frac{\partial K_{\boldsymbol{\theta}}(r)}{\partial \theta_j} = 2 \int_0^\infty \omega^{-\alpha} \frac{\partial S(\omega)}{\partial \theta_j} \cos(2\pi\omega r) \, d\omega, \tag{2.28}$$

which can be computed using the same framework that is used to evaluate $K_{\boldsymbol{\theta}}$ itself.

While differentiating even standard covariance functions such as the Matérn with respect to kernel parameters can be challenging [45], most common spectral densities are very simple to differentiate. The computation of the partial derivatives $\frac{\partial}{\partial \theta_j} S_{\boldsymbol{\theta}}(\omega)$ can be done easily using automatic differentiation (AD) [55], which operates at the code level to programmatically generate derivatives of a given function. As a result, end users can simply write any parametric spectral density $S_{\boldsymbol{\theta}}$ that they would like, and the derivatives of $K_{\boldsymbol{\theta}}$ will be obtained automatically using our software.

This ease of differentiation can be extended to singular spectral densities. To compute the derivative of the kernel $K$ with respect to the singularity parameter $\alpha$, given by

$$\frac{\partial K(r)}{\partial \alpha} = -2 \int_0^\infty \omega^{-\alpha} \log(\omega) S(\omega) \cos(2\pi\omega r) \, d\omega, \tag{2.29}$$

we require a method for accurately integrating the $\omega^{-\alpha}\log(\omega)$ singularity on the panel $[0, b]$ containing the origin. Ignoring the $\log(\omega)$ singularity and applying a Gauss-Jacobi rule results in low accuracy, especially as $\alpha$ approaches 1 and the weights become relatively large and positive near zero. However, if we apply integration by parts

$$\begin{aligned}(1-\alpha)\int_0^b \omega^{-\alpha}\log(\omega)S(\omega)\cos(2\pi\omega r)\,d\omega = {} & b^{1-\alpha}\log(b)S(b)\cos(2\pi b r) \\ & - \int_0^b \omega^{-\alpha}\Big(S(\omega) + \omega\log(\omega)S'(\omega)\Big)\cos(2\pi\omega r)\,d\omega \\ & + 2\pi r \int_0^b \omega^{1-\alpha}\log(\omega)S(\omega)\sin(2\pi\omega r)\,d\omega,\end{aligned} \tag{2.30}$$

we see that the resulting integrals involve only the power law singularity $\omega^{-\alpha}$, and that the log singularity has been removed. As $S$ is a closed form function provided by the user, $S'(\omega)$ can be obtained analytically or using AD, and we can evaluate the above expression using the same Gauss-Jacobi rule that is employed to compute kernel values. This requires only two NUFFTs, and avoids any additional cost to compute specialized quadratures for the $\omega^{-\alpha}\log(\omega)$ singularity. For panels which do not contain the origin, the $\omega^{-\alpha}\log\omega$ term can be included in the integrand, and the Gauss-Legendre rule gives high accuracy as before. Using this strategy and incorporating it into a custom rule for the AD engine again means that this derivative can be obtained entirely programmatically without any end-user intervention.

## 2.3 Numerical results

Before employing the adaptive integration method described above to compute kernel values in a maximum likelihood estimation context with real data, we provide several demonstrations. First, we provide numerical experiments with simulated data to validate our error estimation techniques and illustrate their impact on log-likelihood accuracy. We continue with examples of statistically interesting models which can be written easily in the spectral domain but for which no closed form expression exists for the corresponding covariance function. Next, we show that our method can accurately evaluate the singular Matérn covariance function even when existing alternative numerical methods fail. We close with a runtime comparison which illustrates that our adaptive NUFFT-accelerated Gaussian quadrature scheme is necessary to efficiently obtain high accuracy kernel evaluations.

### 2.3.1 Error estimation for Matérn model

As a first test to validate the error estimation strategies described in Section 2.2.3, we compare it to analytical evaluation of the Matérn covariance $K$ in the challenging regime $\nu = 0.51$, where the spectral density $S$ decays slowly. We choose a moderate range parameter $\rho = 1$ and chose $\varphi$ so that $K(0) = 1$. For various tolerances $\varepsilon$, we compute 100 values of $K(r)$ for $r \in [\texttt{1e-8}, 1]$, along with the corresponding error estimate for each $r$, given by the sum of the quadrature and truncation error estimates discussed in Section 2.2.3. See Figure 2.3. Note that for all tested tolerances $\varepsilon$, all kernel values are computed to within $\varepsilon$ of the true value. In addition, the error estimates are generally accurate indicators of the true errors, or at least act as upper bounds, which is to be expected from the use of the truncation error bound (2.25).

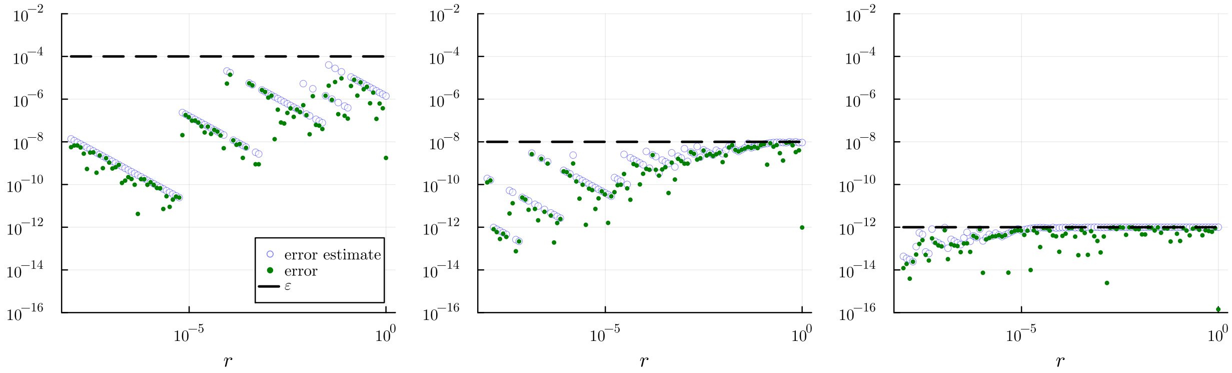


**Figure 2.3:** Pointwise errors and error estimates for various tolerances $\varepsilon$ using a Matérn model with $\nu = 0.51, \rho = 1$, and $\varphi$ chosen so that $K(0) = 1$ with $m = 256$ quadrature nodes per panel, demonstrating the sharpness of the error control methods discussed in the previous section.

### 2.3.2 Relative max-norm errors and the log-likelihood

To support the arguments presented in Section 2.2.3 that the relative max norm error in the covariance matrix is a meaningful error metric for maximum likelihood estimation, we perform a simple numerical study. We take 100 equispaced locations in the interval [0,1] and construct dense covariance matrices using closed form entry evaluation and our scheme with relative max norm tolerance $\varepsilon = 1\text{e}{-8}$ for Matérn models with $\nu = \frac{1}{2}, \frac{5}{2}$ and $\rho$ ranging from 1e−8 to 1e4, with $\varphi$ chosen so that $K(0) = 1$. Using data $\boldsymbol{z}$ simulated using the entry evaluation covariance matrix, we then compute the same log-likelihoods in arbitrary precision as reference and plot the relative error in the log-likelihood $|\ell(\boldsymbol{\Sigma}) - \ell(\tilde{\boldsymbol{\Sigma}})|/|\ell(\boldsymbol{\Sigma})|$ as a function of the standard 2-norm condition number $\kappa(\boldsymbol{\Sigma})$: For well-conditioned $\boldsymbol{\Sigma}$ we see that the number of correct max norm digits in $\boldsymbol{\Sigma}$ agrees with the number of correct digits in the log-likelihood — 8 for our method and roughly 16 for direct kernel evaluation. However, as we increase the lengthscale, the conditioning of $\boldsymbol{\Sigma}$ degrades and we lose accuracy in the log-likelihood *even for double precision closed form entry evaluation.* The conclusion is thus that for well conditioned covariance matrices, controlling the relative max norm does in practice control the log-likelihood error. However, there is no way to circumnavigate ill-conditioning; if $\boldsymbol{\Sigma}$ is ill-conditioned, the log-likelihood cannot be accurately

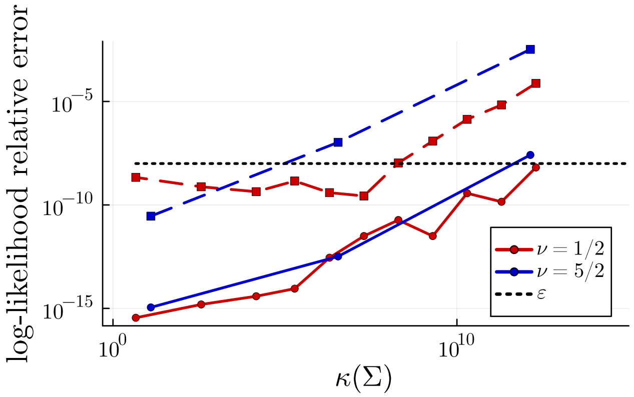


**Figure 2.4:** Log-likelihood relative error as a function of covariance matrix condition number. Solid lines are double precision evaluation of the closed form Matérn covariance, and dashed lines are from our method with $\varepsilon$ = 1e-8.

evaluated in double precision regardless of how the entries of $\mathbf{\Sigma}$ are computed.

### 2.3.3 Designing new spectral densities

As a motivating example to illustrate how easily practitioners can write models in the spectral domain, we study a generalization of the standard Matérn model $K(r) \propto r^{\nu}\mathcal{K}_{\nu}(r)$. Two known limitations of the Matérn model are its exponential decay as $r \to \infty$ for any finite $\nu$, and its inability to take on negative values. While loosening these restrictions in "kernel space" is challenging, it is trivial to write spectral densities whose corresponding kernels move beyond these limitations. Consider, for example, the model

$$S_{\boldsymbol{\theta}}(\omega) = \varphi^2\left(\lambda + (1-\lambda)\,|\omega|^{\gamma}\right)\left(\rho^2 + |\omega|^{\tau}\right)^{-\nu-\frac{1}{2}}, \quad \boldsymbol{\theta} := \{\varphi, \lambda, \gamma, \rho, \tau, \nu\}, \tag{2.31}$$

with $\lambda \in [0,1]$, $\tau \in (0,2]$, and the constraint $\tau(\nu + 1/2) - \gamma > 1$ for integrability. In the case $\tau = 2$ this is precisely a standard Matérn model plus a fractional derivative of a standard Matérn model. For $\tau < 2$ the function $S_{\boldsymbol{\theta}}$ will not be smooth at the origin, so the corresponding kernel will decay more slowly than the exponential rate of the standard Matérn. For small values of $\lambda$ and large enough values of $\gamma$, the corresponding kernel can also take negative values. Figure 2.5

shows an example of this model.

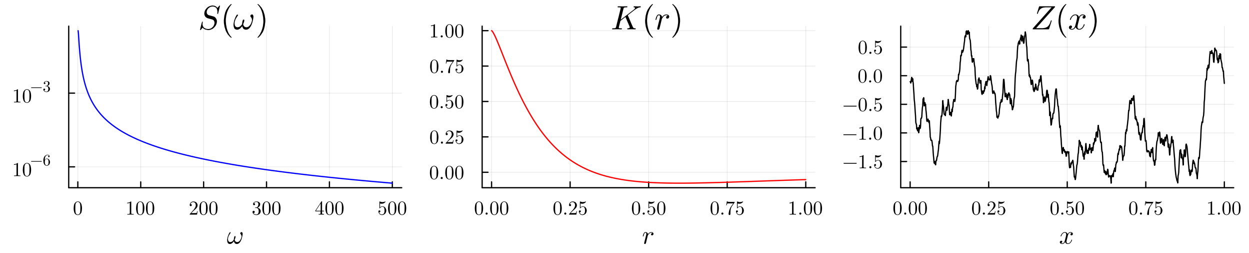


**Figure 2.5:** The generalized Matérn spectral density (2.31) (left), its corresponding covariance function (center), and a sample path from the process (right).

To further highlight the broad class of novel models which can be specified through their spectral densities and fit using our method, we consider two more examples. The first is an “oscillatory" Matérn spectral density given by

$$S_{\boldsymbol{\theta}}(\omega) = \varphi^2(\rho^2 + \omega^2)^{-\nu-\frac{1}{2}}\Big(1 - e^{-\lambda|\omega|}\sin(\gamma|\omega|)\Big), \quad \boldsymbol{\theta} := \{\varphi, \rho, \nu, \lambda, \gamma\}. \tag{2.32}$$

This model promotes oscillatory behavior through strong negative then positive kernel values near the origin, while still maintaining full control over the mean-square differentiability of the process through the parameter $\nu$.

The final model we consider here is a semi-parametric long-memory model given by

$$S_{\boldsymbol{\theta}}(\omega) = \varphi^2|\omega|^{-\alpha}\exp\left\{-\lambda|\omega| + \sum_{k=0}^{K} c_k T_k\left(\frac{|\omega| - \rho}{|\omega| + \rho}\right)\right\}, \quad \boldsymbol{\theta} := \{\varphi, \alpha, \lambda, \rho, c_0, \dots, c_K\}, \tag{2.33}$$

where $T_k$ is the Chebyshev polynomial of order $k$. This model is in the spirit of semi-parametric ideas like the popular *spectral mixture kernel* [126] in that it avoids directly specifying the functional form of the spectral density. The singularity parameter here again gives the model the flexibility to capture slowly decaying tails in the covariance function, and the arbitrary number of orthogonal polynomials affecting frequencies near the origin can yield flexible kernels. Both of these models are shown in the spectral domain, covariance domain, and with an example sample

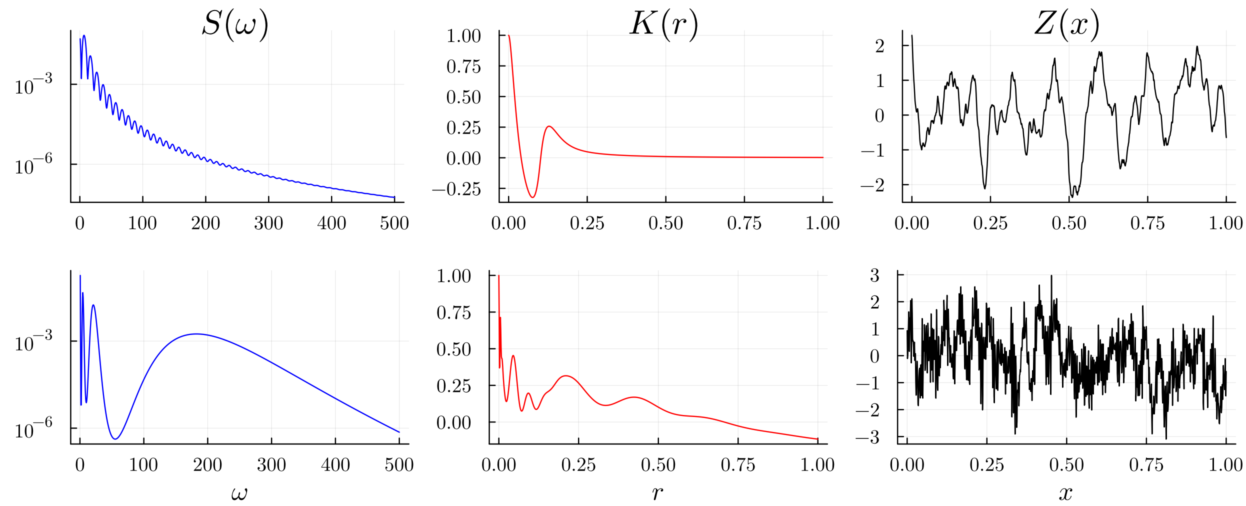


**Figure 2.6:** The top row shows the “oscillatory” Matérn spectral density (2.32) (left), its corresponding covariance function (center), and a sample path from the process (right). The bottom row displays analogous quantities for the semi-parametric long-memory model (2.33).

path in Figure 2.6.

### 2.3.4 Singular Matérn model

As previously mentioned, the singular Matérn model (2.26) is a useful tool for validating our approach and demonstrating its effectiveness for spectral densities with origin singularities. The Fourier transform of (2.26) is given by

$$\begin{aligned} K(r) &= 2\int_0^\infty |\omega|^{-\alpha}\varphi(\rho^2+\omega^2)^{-\nu-1/2}\cos(2\pi\omega r)\,\mathrm{d}\omega \qquad (2.34)\\ &= \frac{\varphi}{\rho^{2\nu}\,(2\pi r)^{-\alpha}\,\Gamma(\nu+\frac{1}{2})}\Bigg[(2\pi r\rho)^{-\alpha}\,\Gamma(\tfrac{2\nu+\alpha}{2})\Gamma(\tfrac{1-\alpha}{2})\,{}_1F_2\Big(\tfrac{1-\alpha}{2},\{\tfrac{1}{2},\tfrac{2-2\nu-\alpha}{2}\},(\rho\pi r)^2\Big)\\ &\quad+2(2\pi r\rho)^{2\nu}\cos(\tfrac{(2\nu+\alpha)\pi}{2})\Gamma(\tfrac{2\nu+1}{2})\Gamma(-\alpha-2\nu)\,{}_1F_2\Big(\tfrac{2\nu+1}{2},\{\tfrac{1+2\nu+\alpha}{2},\tfrac{2+2\nu+\alpha}{2}\},(\rho\pi r)^2\Big)\Bigg], \end{aligned}$$

where $\Gamma$ is the Gamma function and ${}_1F_2$ is a generalized hypergeometric function [90]. Evaluating this $K(r)$ accurately in double precision is extremely challenging, as the two terms inside the brackets are rapidly growing numbers of opposite signs which approach each other in absolute value and cancel as $r$ grows. In exact arithmetic this is not a problem, but it poses a serious numerical issue in finite precision. Consider the case where both of the above terms have absolute value of order `1e20`, and a routine for evaluating any of the constituent special functions incurs a numerical relative error of even $10^{-15}$. Such small relative errors are inevitable, as double precision floating point numbers store only roughly 16 relative digits. Then an addition which should result in exact cancellation to zero could give a value of size $10^{-15} \times 10^{20} \approx 10^5$ instead. Considering how quickly the inner terms grow, one reaches this regime for small $r$ even with non-pathological choices of $\varphi$, $\rho$, and $\nu$.

To illustrate the challenge of evaluating the singular Matérn covariance function (2.34), we compare three methods for computing $K(r)$: one using a double-precision library for all special functions, one using the extended precision mathematical library `Arb` [65] for all special functions with 10,000 bits of precision, and our method.

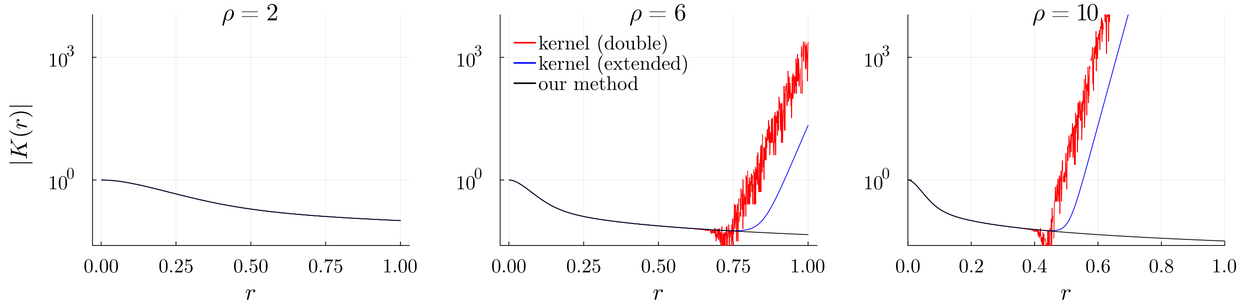


**Figure 2.7:** A comparison of the three methods for evaluating the covariance function given in Equation 2.34: the double-precision direct kernel routine (red), the extended precision direct kernel routine (blue), and our Fourier quadrature routine (black).

Figure 2.7 shows the evaluated kernel on a regular grid of points in $[0, 1]$ for a choice of $\nu = 2.1$, $\alpha = 0.3$, several $\rho$ values, and in all cases a $\varphi$ such that $K(0) = 1$, which is a "best case" numerical choice. For the case $\rho = 2$, Figure 2.7 shows good visual agreement between the three methods. But as $\rho$ is increased, we can see that both direct evaluation methods clearly fail to achieve even a single correct digit by $r = 0.8$. For $\rho = 10$, which is again still not a pathological parameter choice, we see that both direct evaluation methods are so affected by roundoff errors ruining cancellation that they provide $K(1)$ to be on the order of 1e14.

The difficulty of accurately evaluating $K(r)$ for large $r$ is especially problematic because long-memory models are most commonly applied to data which are strongly dependent on long time horizons — and it is precisely these cases in which evaluating the covariance function for large $r$ is most relevant. This is because singular spectral densities can be used to build covariance functions that decay exceptionally slowly. As the following theorem demonstrates, singular spectral densities can provide covariance functions that are square-integrable but not integrable, or for $\alpha > \frac{1}{2}$ not even square-integrable.

**Theorem 2.3.** *Let $S \in C^2(\mathbb{R}) \cap L^1(\mathbb{R})$ be a bounded, symmetric, positive spectral density with $S', S'' \in L^1(\mathbb{R})$. Then for all $\alpha \in [0, 1)$*

$$K_\alpha(r) := 2\int_0^\infty |\omega|^{-\alpha} S(\omega)\cos(2\pi\omega r)\,\mathrm{d}\omega \sim r^{-1+\alpha} \quad \textit{as } r \to \infty. \tag{2.35}$$

*Proof.* Define the covariance function $K := \mathcal{F}^{-1}\{S\}$ corresponding to $S$. As $S$ is integrable, $K$ is well defined and bounded

$$|K(r)| \leq \int_{-\infty}^\infty S(\omega)\,|\cos(2\pi\omega r)|\,\mathrm{d}\omega = \|S\|_{L^1(\mathbb{R})} < \infty. \tag{2.36}$$

As $S \in C^2(\mathbb{R})$ and $S', S'' \in L^1(\mathbb{R})$, integrating by parts twice and applying the Riemann-Lebesgue lemma gives the standard decay rate $K(r) = o(r^{-2})$ for $r \to \infty$. Therefore $K$ is integrable with

$$M := \int_{-\infty}^\infty K(t)dt < \infty. \tag{2.37}$$

Using the fact that

$$\mathcal{F}^{-1}\{\,|\cdot|^{-\alpha}\,\}(r) = (2\pi)^\alpha \frac{\Gamma(1-\alpha)}{\pi} \sin\left(\frac{\pi\alpha}{2}\right)|r|^{-1+\alpha} \tag{2.38}$$

in the distributional sense and applying the convolution theorem, we obtain

$$\begin{aligned} K_\alpha(r) &= \mathcal{F}^{-1}\Big\{ |\cdot|^{-\alpha}\, S(\cdot)\Big\}(r) && (2.39)\\ &= \Big(\mathcal{F}^{-1}\big\{ |\cdot|^{-\alpha} \big\} * \mathcal{F}^{-1}\big\{S\big\}\Big)(r) && (2.40)\\ &\propto \Big( |\cdot|^{-1+\alpha} * K\Big)(r) && (2.41)\\ &= \int_{-\infty}^{\infty} |t|^{-1+\alpha}\, K(r-t)\,\mathrm{d}t. && (2.42) \end{aligned}$$

Take $0 < c < (1-\alpha)^{\frac{1}{2-\alpha}} < 1$. First consider the upper tail of the integral (2.42). Taylor expanding around $t = r$ gives

$$\int_{cr}^{\infty} t^{-1+\alpha} K(r-t)\,\mathrm{d}t = \int_{cr}^{\infty} \Big(r^{-1+\alpha} - (1-\alpha)\xi^{-2+\alpha}(t-r)\Big)K(r-t)\,\mathrm{d}t \tag{2.43}$$

for some $\xi$ between $r$ and $t$. This results in two terms. As $K$ is integrable, the first term gives the desired asymptotic behavior

$$r^{-1+\alpha}\int_{cr}^{\infty} K(r-t)\,\mathrm{d}t = r^{-1+\alpha}\Big(M - \int_{(1-c)r}^{\infty} K(u)\,\mathrm{d}u\Big) \sim r^{-1+\alpha}. \tag{2.44}$$

As $\xi \geq cr$ and $K(r) = o(r^{-2})$ for $r \longrightarrow \infty$, the second term decays with at least this asymptotic rate

$$(1-\alpha)\int_{cr}^{\infty} \xi^{-2+\alpha}(t-r)K(r-t)\,\mathrm{d}t \lesssim r^{-1+\alpha} \tag{2.45}$$

and by our choice of $c$, is strictly smaller in magnitude than (2.44), which avoids cancellation. Next consider the lower tail of (2.42)

$$\int_{-\infty}^{-cr} (-t)^{-1+\alpha}K(r-t)\,\mathrm{d}t \leq r^{-1+\alpha}\int_{(1+c)r}^{\infty} K(u)\,\mathrm{d}u \lesssim r^{-2+\alpha} \tag{2.46}$$

again due to the fact that $K(r) = o(r^{-2})$ for $r \longrightarrow \infty$. Finally consider the central term in (2.42). Define the interval $I_r := [(1-c)r, (1+c)r]$. Then we have

$$\int_{-cr}^{cr} t^{-1+\alpha} K(r-t)\,\mathrm{d}t \leq \|K\|_{L^\infty(I_r)} \frac{2}{\alpha}(cr)^\alpha \lesssim r^{-2+\alpha} \tag{2.47}$$

as $\|K\|_{L^\infty(I_r)} = o(r^{-2})$ for $r \longrightarrow \infty$. □

We note that the requirement $S \in C^2(\mathbb{R})$ is far from a necessary condition, and this asymptotic behavior is a very general phenomenon. For example $S(\omega) = e^{-|\omega|}$ yields the covariance function

$$K_\alpha(r) = \frac{2}{\Gamma(1-\alpha)} \left(1 + (2\pi r)^2\right)^{-\frac{1}{2}(1-\alpha)} \cos\left((1-\alpha)\tan^{-1}(2\pi\,|r|)\right) \sim r^{-1+a} \tag{2.48}$$

as $r \longrightarrow \infty$ despite the fact that $S$ is not even $C^1(\mathbb{R})$. In such cases where $S$ is monotonic on $[0, \infty)$, one can apply more general Tauberian results from the theory of slowly varying functions to obtain the same asymptotic decay [12, Theorem 4.10.3].

**Remark 2.4.** It is possible that one could use an expansion of the hypergeometric function in conjunction with relevant special function identities to derive an evaluation scheme for (2.34) which avoids the numerical instability of adding very large floats. However, doing so efficiently and accurately would represent a significant research endeavor, and our method can evaluate a broad class of such covariance functions without requiring further kernel-specific effort.

### 2.3.5 Performance for dense covariance matrix construction

As discussed in Section 2.2.3, by controlling the pointwise error relative to $K(0)$ we have bounded the relative max-norm error in any covariance matrix with entries computed using our method. In Figure 2.8, we demonstrate agreement between the user-specified tolerance $\varepsilon$ and the relative error in $\mathbf{\Sigma}$ in various matrix norms with $N = 1000$ random observation locations $x_1, \ldots, x_N \overset{\text{i.i.d.}}{\sim} \text{Unif}([0,1])$ for a slowly-decaying singular Matérn model with $\nu = 0.51, \rho = 0.5$,

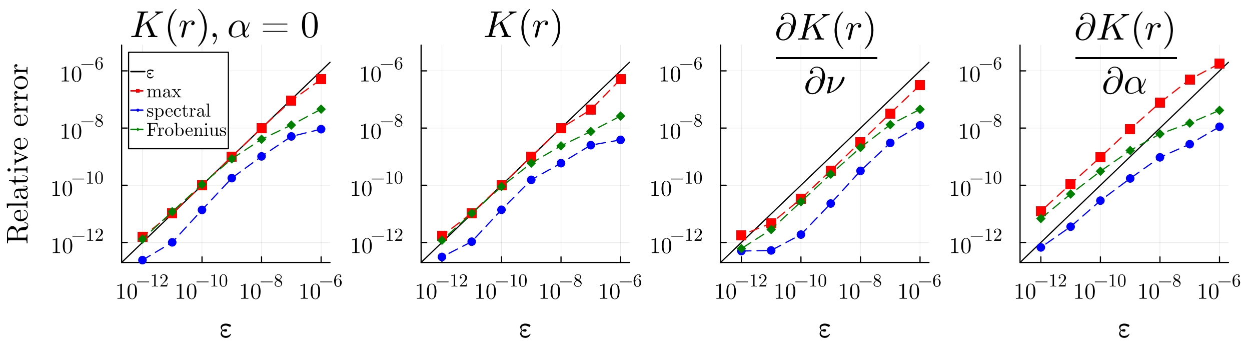


**Figure 2.8:** Comparison of the relative error $\|\boldsymbol{A} - \tilde{\boldsymbol{A}}\|/\|\boldsymbol{A}\|$ with the requested tolerance $\varepsilon$ for a variety of matrix norms, where $\boldsymbol{A} = \boldsymbol{\Sigma}$ (first and second plots) or $\boldsymbol{A} = \frac{\partial}{\partial \theta}\boldsymbol{\Sigma}$ (third and fourth plots). In the first plot, $\alpha$ is set to zero to test the non-singular case, and all other parameters remain unchanged.

and $\alpha = 0.1$, with $\varphi$ chosen so that $K(0) = 1$. We also show analogous errors in the derivative matrices $\frac{\partial}{\partial \theta}\boldsymbol{\Sigma}$ computed using AD for parameters $\theta \in \{\nu, \alpha\}$, for which the derivatives are most numerically challenging.

Having confirmed the accuracy of our scheme, we now study its computational cost, and demonstrate that both the NUFFT and our Nyquist-based heuristic for increasing panel length are necessary for computational efficiency. For various $N$ ranging from 10 to 10,000, we form the dense covariance matrix for $N$ uniform random observation locations in $[0, 1]$ as above and time three approaches. First, we run our adaptive Gauss-Legendre method and time only the evaluation of the necessary NUFFTs, excluding the computation of error estimates. This provides a fair comparison with the trapezoidal rule, which is non-adaptive. Second, we repeat this with direct Fourier sums in place of the NUFFT. Finally, we use [2, Corollary 6] to determine the grid spacing $h$ and number of quadrature nodes $m$ in a trapezoidal rule necessary to obtain each tolerance, and time the “type 2” NUFFT needed to compute kernel values from this quadrature rule. Figure 2.9 shows the resulting runtimes using an 8 core Apple M1 Pro CPU with 32GB of memory.

There are a two significant conclusions to be gathered from these results. The most obvious is the paramount importance of the NUFFT. The difference between $O(nm)$ and $O(m+n \log n)$ com-

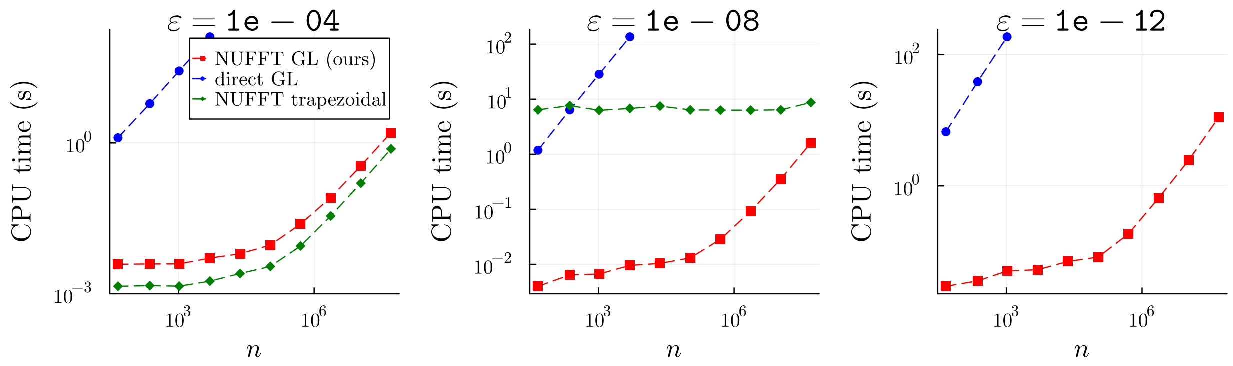


**Figure 2.9:** A comparison of the runtime cost of computing kernel values from a Matérn model with $\nu = 0.55$, $\alpha = 0.5$, and $\varphi$ chosen so that $K(0) = 1$. For three tolerances $\varepsilon$ we plot Gauss-Legendre quadrature with direct summation (blue), global trapezoidal quadrature with the NUFFT (green), and Gauss-Legendre quadrature with the NUFFT (red).

plexity for direct summation and the NUFFT respectively gives an orders of magnitude speedup, without which one can afford to compute only very few kernel values. The other conclusion is the necessity of our Nyquist-based heuristic for obtaining high accuracy kernel values. For tolerance $\varepsilon = 10^{-8}$, the trapezoidal rule requires both small $h$ in order to resolve the spectral density near the origin, as well as large $m$ to control the truncation error. This results in a very large NUFFT of size $m \approx$ 1e8. The $O(m)$ “spreading” step in the NUFFT thus becomes the dominant cost for all tested $n$, leading to a nearly constant cost which is orders of magnitude slower than our adaptive quadrature for moderate $n$. For $\varepsilon = 10^{-12}$, the necessary number of equispaced trapezoidal nodes is $m \approx$ 1e12, and results in memory issues. As our adaptive scheme uses panels with $m = 2^{16}$ nodes, the $O(m)$ “spreading” step is only the bottleneck up to $n =$ 1e5 or so, after which the $O(n \log n)$ equispaced FFT which is evaluated within the NUFFT routine becomes the dominant cost, and we see quasilinear scaling with $n$. Therefore our method still runs in seconds, even when computing the kernel at approximately 50 million distances to 12 digit accuracy.

## 2.4 Application

In this section we demonstrate the practical value of our framework with an application to high-frequency vertical wind profiles. The US Department of Energy's Atmospheric Radiation Measurement (ARM) program offers a large collection of freely available measurements collected at field sites across the country, and in this chapter we look at the Doppler LiDAR-based vertical wind profiles made at the main field site in the Southern Great Plains (SGP) collection [84, 86]. These measurements of vertical profiles are made at a temporal resolution of approximately one second and a spatial resolution of 30m, thus providing particularly high resolution in time. However, aside from some special small segments across the several years' worth of data, the measurements have frequent interruptions due to horizontal sweeps made by the sensor, occasional long pauses for various reasons, or other momentary instrument-based delays. For this reason, unless one focuses on very narrow time intervals of approximately $12 - 13$ minutes or is uninterested in studying high-frequency structure of the process, approaches like ours for continuous-time models which are applicable to irregularly sampled data become necessary.

### 2.4.1 Preliminary analysis

We investigate here the same six days that were studied in [42], but by lifting the limitation of gridded data, we can examine a full hour of data on each of the days. Figure 2.10 shows an example of one hour of measurements made at the altitude of 240m.

A practitioner looking to model this data may first compute a Whittle-type estimator [124] as an exploratory tool. Figure 2.11 shows the result of Whittle estimation that is performed by treating each of the six one-hour segments of data as i.i.d. samples, breaking them into four segments based on the largest gaps and ignoring the smaller irregularities, and computing one averaged periodogram per day. The Whittle MLE for this data implies a strong singularity of $\hat{\alpha} \approx 0.62$, corresponding to a process with significant long-range dependence even when segments

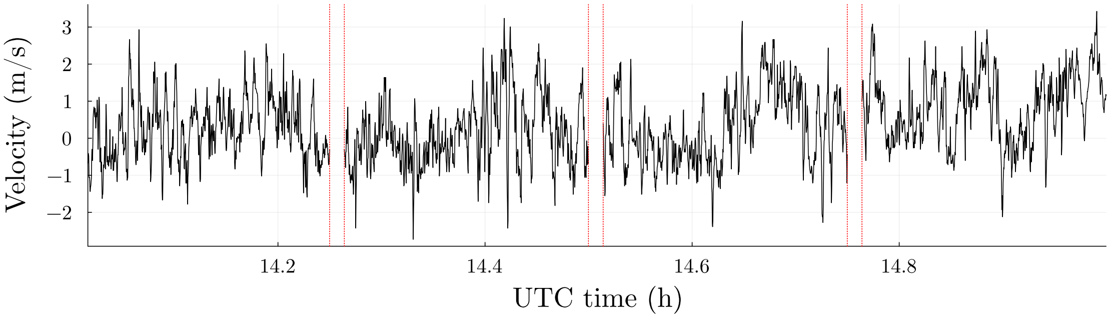


**Figure 2.10:** The Doppler LIDAR data from 1400-1500 UTC on June 03, 2015, at an altitude of 240m. Vertical lines are given to emphasize the largest three gaps in measurements.

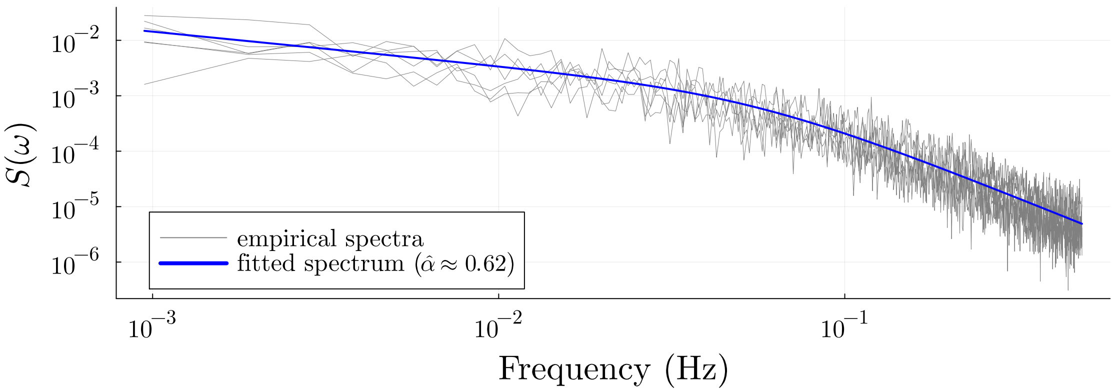


**Figure 2.11:** Whittle MLE for the singular Matérn model computed treating the six days of LiDAR data as i.i.d. samples and doing rudimentary gap and sample irregularity handling.

are constrained to 12 − 13 minutes, which limits the lowest observable frequencies.

### 2.4.2 Parameter estimation methodology

Motivated by this exploratory analysis, in this section we individually fit each of the six days of data as a continuous-time process using the singular Matérn model. In all cases, the MLE indicates a strong singularity comparable to the one obtained by the Whittle estimator. We additionally fit a standard Matérn model to compare log-likelihoods, and in all cases the log-likelihood of the data is materially improved by adding the singularity.

All estimates discussed below were obtained using the Artelys KNITRO optimizer [20] with the sequential quadratic programming (SQP) algorithm. In this spirit of Fisher scoring, we use the *expected* Fisher information matrix, given by

$$\mathcal{I}(\boldsymbol{\theta})_{jk} = \frac{1}{2}\mathrm{tr}\left(\boldsymbol{\Sigma}_{\boldsymbol{\theta}}^{-1}\left[\frac{\partial}{\partial\theta_j}\boldsymbol{\Sigma}_{\boldsymbol{\theta}}\right]\boldsymbol{\Sigma}_{\boldsymbol{\theta}}^{-1}\left[\frac{\partial}{\partial\theta_k}\boldsymbol{\Sigma}_{\boldsymbol{\theta}}\right]\right), \tag{2.49}$$

as a proxy for the Hessian of the log-likelihood. This matrix has the benefit of being computable with only first derivatives of the covariance function, and has meaningfully better performance than a general-purpose BFGS approximation [6, 42, 43, 56].

### 2.4.3 Results

We now discuss in detail the estimation results for the data on June 03, one of the six days that was studied. The results from fitting singular and standard Matérn models with a nugget are summarized by point estimates and terminal likelihood values in Table 2.1. Along with point estimates, this table provides standard deviations implied by the expected Fisher information matrix. These implied uncertainties are often informative, but should be interpreted with care because the regularity conditions under which the expected Fisher matrix converges to the asymptotic precision of the MLE do not hold in many spatial modeling settings [110].

| | fixed $\alpha = 0$ | fitted $\alpha$ |
|---|---|---|
| $\ell(\hat{\boldsymbol{\theta}})$ | -1898.42 | -1948.35 |
| $\varphi$ | 25.72 (4.182) | 205.3 (95.429) |
| $\rho$ | 100.0 (7.854) | 351.2 (45.705) |
| $\nu$ | 0.7947 (0.023) | 0.6908 (0.058) |
| $\alpha$ | — | 0.775 (0.072) |

**Table 2.1:** MLE estimates and terminal negative log-likelihoods for the standard Matérn (fixed $\alpha = 0$) and the singular Matérn model for data on June 03. When possible, expected Fisher matrix-implied standard deviations are provided in parentheses.

The most obvious observation from Table 2.1 is that the singular Matérn model has a materially better log-likelihood than the standard Matérn model. A more scientifically interesting observation pertains to the implied smoothness of the process: noting that

$$\varphi^2|\omega|^{-\alpha}(\rho^2+\omega^2)^{-\nu-\frac{1}{2}} \sim \omega^{-\alpha-2\nu-1} \tag{2.50}$$

at high frequencies, we see that the implied decay rate of the singular Matérn spectral density using the estimated $(\hat{\nu}, \hat{\alpha}) \approx (0.69, 0.78)$ is $\hat{\beta} = \hat{\alpha} + 2\hat{\nu} + 1 \approx 3.06$. Recall that a process with spectral density $S$ is mean-square differentiable if and only if $\int \omega^2 S(\omega) < \infty$, which in this notation is equivalent to $\beta > 3$. Therefore, under the singular model where $\alpha$ is estimated, the process is mean-square differentiable, whereas in the standard Matérn model the estimate of $\hat{\nu} = 0.79$ gives the decay $\hat{\beta} = 2\hat{\nu} + 1 \approx 2.58$ which implies that it is not. The differentiability of these measurements below the atmospheric boundary layer (ABL) where convective forces are dominant and the process exhibits chaotic behavior has been a question of interest in several prior applications [42, 44], with parameter estimates often being very borderline. The differentiability under the singular Matérn model is an indication that the long-range dependence parameter $\alpha$ may be valuable in disentangling low-frequency and high-frequency behavior in such processes, and agrees with the more recent work that uses continuous-time models in kernel-space [44].

A related observation that is particularly interesting from a theoretical perspective is that, to the degree that the expected Fisher information matrices can be trusted to serve as proxies for the precision of the MLE, the smoothness and singularity parameters $\nu$ and $\alpha$ are jointly resolved reasonably well. Table 2.1 shows that the implied uncertainties for both of those parameters are small, whereas the parameters $\varphi$ and $\rho$ which now most affect moderate frequencies are much less well-resolved. It is well-known that in most fixed-domain asymptotic regimes in fewer than four dimensions, no individual parameter besides the smoothness $\nu$ in the Matérn model can be estimated consistently [62, 110, 128], and in light of this observation the high uncertainty in $\varphi$ and $\rho$ is not surprising. For the singular Matérn model in particular, however, $\rho$ serves the important purpose of making $S$ bounded at the origin, so that the singularity parameter $\alpha$ can be disentangled from the effect of $\nu$ on the tail decay in the model $|\omega|^{-\alpha}S(\omega)$.

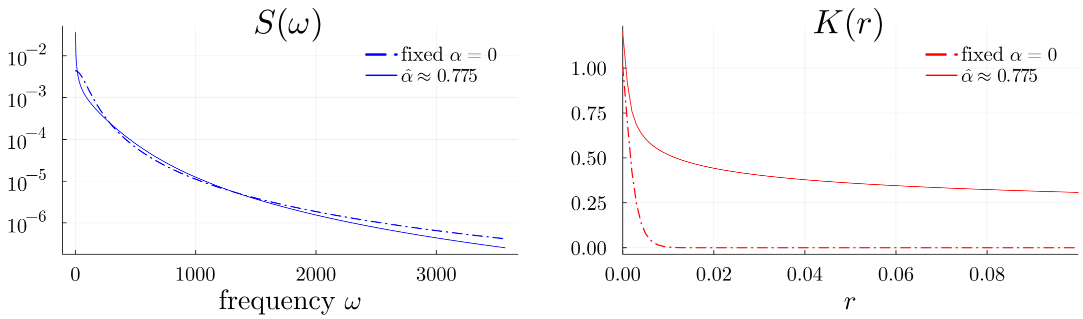


**Figure 2.12:** Left: Model-implied spectral densities for the data on June 03 and the parameter estimates in Table 2.1. Right: the corresponding model-implied kernels.

To understand what these estimates imply about the spectral densities and covariances of the processes, Figure 2.12 shows the MLE-implied spectra and kernel values from both models. As can be seen, the singular model moves much of the spectral mass into the singularity near the origin, and the more rapid decay in the tails of the singular Matérn spectral density is clearly visible. The plot of the implied kernel values in the center of Figure 2.12 shows a more dramatic difference, as the estimated standard Matérn kernel is highly concentrated at the origin, while the singular Matérn kernel displays slow decay. Finally, Figure 2.13 shows sample paths from the two MLE-implied models computed with the same white noise forcing. As Table 2.1 and Figure 2.12 would suggest, the additional high-frequency information in the standard Matérn model is quite prominent.

## 2.5 Discussion

In this chapter we introduce a numerical method for efficiently and accurately evaluating the Fourier transform of spectral densities, even those that are barely integrable due to singularities at the origin or slowly decaying tails. Making this strategy computational practical requires several technical observations. The first and most crucial is that high-order quadrature rules can significantly accelerate convergence. The use of the NUFFT is critical for this purpose, as

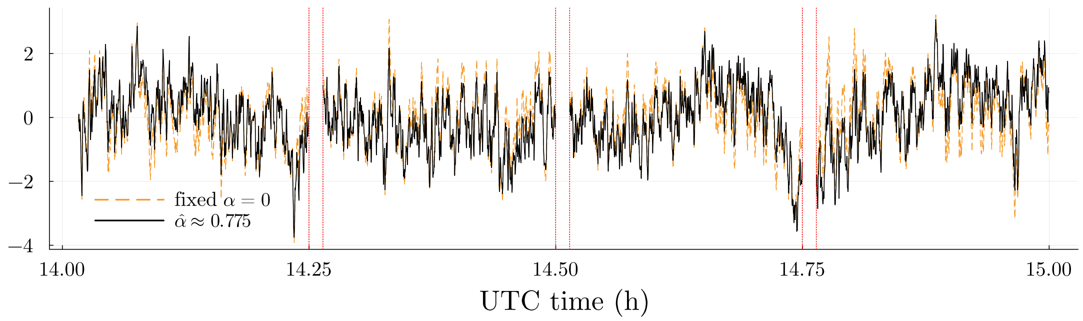


**Figure 2.13:** Sample paths with identical white noise forcing of the processes implied by the standard Matérn MLE (orange dashes) and the singular Matérn MLE (solid black), showing the noticeable affect of the more rapid spectral decay implied by the singular Matérn model.

it reduces the computational cost to quasilinear complexity in both the number of quadrature nodes $m$ and the number of inter-observation distances $n$. The second vital observation is that the covariance function $K$ is generally resolved to a given tolerance $\varepsilon$ more quickly at distances $r$ that are well-separated from the origin. Therefore one can adaptively increase the width of panels being integrated as one reduces the highest frequency oscillations in the integrands remaining to be resolved. This Nyquist-based observation is essential to overcoming the difficulties of slowly decaying spectral densities, which may take orders of magnitude longer to converge if one were to use a more uniform quadrature rule. Finally, the design of an efficient but precise stopping criterion based on both truncation error *and* panel contribution is imperative to keeping the routine performant but accurate for a wide variety of spectral densities. Combining all of these observations and the additional technical discussions provided in Section 2.2, one can evaluate covariance functions specified by these spectral densities at tens of millions of locations in seconds on a laptop.

The cost of using this method to compute kernel values will naturally be significantly higher than direct kernel evaluation when a simple closed form exists, for example the standard Matérn model. Although the evaluation of $\mathcal{K}_\nu$ makes the Matérn covariance numerically challenging, it can be computed to nearly machine precision in less than a microsecond due to advances in modern special function libraries [45]. Therefore, in such cases where efficient special function implementations can be used, direct kernel evaluation will likely be both faster and more accurate than even the $\varepsilon = 10^{-12}$ given by our method. Yet for the vast majority of spectral densities whose Fourier transforms are completely unavailable in closed form, our method is a robust, broadly applicable option for obtaining direct kernel evaluations for irregularly sampled data which can be composed with a practitioner's estimation method of choice. The code and details of the integration strategy can surely be improved, further reducing the overall cost of this approach—but even now it makes many modeling choices available that were previously effectively impossible. While the LiDAR application in this chapter is focused on the singular Matérn model, we again remind the reader that the objective of this framework is to empower practitioners to work with *any* continuous, integrable spectral density. The extremely slow decay of a long-memory covariance function is of course not always desirable, and there are many other new ways to add valuable degrees of freedom to parametric families of spectral densities. It is our hope that this method and the accompanying software will inspire others to explore and discover new parametric families of models useful to their own contexts.

We emphasize that our framework can be used in conjunction with a number of existing methods for large scale and nonstationary GPs. First, because it requires only a spectral density and a list of distances $r_j$ at which to evaluate the corresponding covariance function, our method can be used with any approximation strategy for which these distances are specified in advance. Examples include Vecchia approximation methods [68, 112, 121] and entry-based rank-structured covariance matrix approximations [6, 25, 43]. Second, we note that while our approach is of course intrinsically limited to evaluating stationary covariance functions, it is compatible with a number of methods which construct nonstationary models from stationary ones, including warping [101] and convolutional frameworks [93].

Finally, it is natural to ask how this methodology can be extended to multiple dimensions. For any isotropic spectral density $S(\boldsymbol{\omega}) = S(\|\boldsymbol{\omega}\|)$ in $\mathbb{R}^d$ one can integrate out the radial variables,

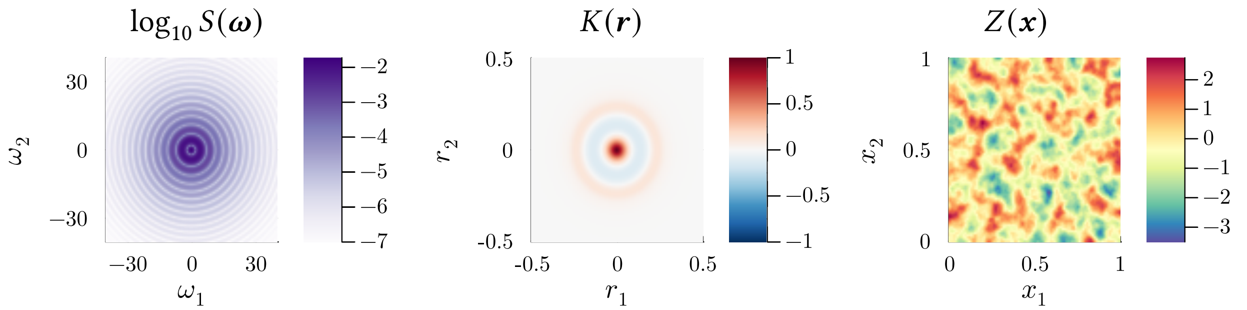


**Figure 2.14:** Log of spectral density (left), corresponding covariance function (center), and sample from the process (right) using the isotropic "oscillatory" Matérn model (2.32) in two dimensions.

resulting in the covariance function

$$K(\boldsymbol{r}) = \int_{\mathbb{R}^d} S(\|\boldsymbol{\omega}\|) e^{2\pi i \boldsymbol{\omega}^\top \boldsymbol{r}} \, \mathrm{d}\boldsymbol{\omega} = \frac{2\pi}{r^{\frac{d}{2}-1}} \int_0^\infty S(\omega) J_{\frac{d}{2}-1}(2\pi\omega \|\boldsymbol{r}\|) \, \omega^{\frac{d}{2}} \, \mathrm{d}\omega \tag{2.51}$$

where $J_\nu$ is the Bessel function of the first kind of order $\nu$. We can compute this integral using the one-dimensional adaptive integration framework just described, with the only modification being that the NUFFT is replaced by a nonuniform fast Hankel transform. While there exist a number of fast Hankel transform algorithms [29, 117], we are not aware of an existing fully nonuniform method in which $r$ and $\omega$ can be chosen freely to accommodate the use of Gaussian quadrature rules for irregularly sampled data. Without this fast transform, we are restricted to direct $O(nm)$ summation to compute each panel integral. Figure 2.14 shows an example of a 2D isotropic kernel computed with direct summation. In the following chapter, we develop the necessary NUFHT to accelerate these panel integral computations in higher dimensions.

# 3 A nonuniform fast Hankel transform

## 3.1 Introduction

In this chapter — motivated by the panel integration framework detailed in the previous chapter — we describe a fast algorithm for computing discrete Hankel transforms of moderate orders from $n$ nonuniform points to $m$ nonuniform frequencies in $O\big((m+n)\log\min(n,m)\big)$ operations. Our approach combines local and asymptotic Bessel function expansions with nonuniform fast Fourier transforms. The order of each expansion is adjusted automatically according to error analysis to obtain any desired precision $\varepsilon$. Several numerical examples are provided which demonstrate the speed and accuracy of the algorithm in multiple regimes and applications.

The problem of computing Fourier transforms of radially symmetric functions in dimensions $d \geq 2$, as seen in the GRF context in (2.51), is common across a variety of applications from electromagnetics to imaging. To agree with the dominant convention in the literature in those fields, we exchange the roles of $\omega$ and $r$ used in Chapter 2 for the duration of this chapter. In this notation, we will study here the problem of, for example, computing the Fourier transform of a function $f$ in two dimensions given by

$$g(\omega_1,\omega_2) = \frac{1}{4\pi^2}\iint_{\mathbb{R}^2} f(x_1,x_2)\, e^{-i(\omega_1 x_1+\omega_2 x_2)}\, dx_1\, dx_2. \tag{3.1}$$

Transforming to polar coordinates $(\omega_1, \omega_2) \mapsto (\omega, \alpha)$ and $(x_1, x_2) \mapsto (r, \theta)$ the above expression becomes

$$\begin{aligned} g(\omega, \alpha) &= \frac{1}{4\pi^2} \int_0^{2\pi} \int_0^{\infty} f(r, \theta)\, e^{-i\omega r(\cos\alpha\cos\theta + \sin\alpha\sin\theta)}\, r\, dr\, d\theta \\ &= \frac{1}{4\pi^2} \int_0^{2\pi} \int_0^{\infty} f(r, \theta)\, e^{-i\omega r\cos(\alpha - \theta)}\, r\, dr\, d\theta. \end{aligned} \tag{3.2}$$

Furthermore, if $f$ is radially symmetric, i.e. $f(r, \theta) = f(r)$, then the above transform can be written as

$$\begin{aligned} g(\omega, \alpha) &= \frac{1}{4\pi^2} \int_0^{\infty} f(r)\, r \int_0^{2\pi} e^{-i\omega r\cos(\alpha - \theta)}\, d\theta\, dr \\ &= \frac{1}{2\pi} \int_0^{\infty} f(r)\, J_0(\omega r)\, r\, dr, \end{aligned} \tag{3.3}$$

where we have used the integral representation of the zeroth-order Bessel function [90]

$$J_0(x) = \frac{1}{\pi} \int_0^{\pi} \cos\left(x\cos\theta\right)\, d\theta. \tag{3.4}$$

The final integral involving $J_0$ in equation (3.3) is known as a *Hankel Transform* of order $0$ — usually referred to simply as a Hankel Transform.

In higher ambient dimensions, the Fourier transform of radially symmetric functions reduces to a Hankel transform of higher order. Similarly, if the function $f$ in (3.2) has a particular periodic dependence in $\theta$ so that $f(r, \theta) = f(r)e^{i\nu\theta}$ with $\nu \in \mathbb{Z}$, then we have

$$\begin{aligned} g(\omega, \alpha) &= \frac{1}{4\pi^2} \int_0^{\infty} f(r)\, r \int_0^{2\pi} e^{-i\omega r\cos(\alpha - \theta)}\, e^{i\nu\theta}\, d\theta\, dr \\ &= \frac{i^{\nu}}{2\pi} \int_0^{\infty} f(r)\, r\, J_{\nu}(\omega r)\, dr, \end{aligned} \tag{3.5}$$

where, again, we have invoked an integral representation for $J_\nu$ [90].

In order to numerically compute $g$ in (3.3) or (3.5) at a collection of $m$ “frequencies” $\omega_j$, the Hankel transform must be discretized using an appropriate quadrature rule with nodes $r_k$ and weights $w_k$ which depend on the particular class of $f$ for which the integral is desired. In general this results in the need for computing

$$\begin{aligned} g(\omega_j) \approx g_j &:= \sum_{k=1}^{n} w_k \, f(r_k) \, r_k \, J_\nu(\omega_j r_k) \\ &= \sum_{k=1}^{n} c_k \, J_\nu(\omega_j r_k) \qquad \text{for } j = 1, \ldots, m. \end{aligned} \tag{3.6}$$

The above sum will be referred to as the Discrete Hankel Transform (DHT) of order $\nu$.

In our motivating example — computing the continuous Fourier transform — the DHT arises from the discretization of the radially symmetric Fourier integral. The DHT also appears in a wide range of applications including imaging [60, 81, 129], statistics [41, 79], and separation of variables methods in partial differential equations [1, 13, 130]. In many such applications, a fully nonuniform DHT is desired, as the relevant frequencies $\omega_j$ may not be equispaced, and the most efficient quadrature rule for discretizing (3.3) may have nodes $r_k$ which are also not equispaced.

The algorithm of this work allows for arbitrary selection of the frequencies $\omega_j$ and nodes $r_k$, in contrast to other algorithms which require some structure to their location (e.g. equispaced or exponentially distributed). There are a few types of commonly encountered DHTs, all of which our algorithm can address. Schlömilch expansions [77, 117] take frequencies $\omega_j = j\pi$. Fourier-Bessel expansions — often used in separation of variables calculations for PDEs — take frequencies $\omega_j = \xi_{\nu,j}$, where $\xi_{\nu,j}$ denotes the $j^{th}$ root of $J_\nu$. In the most restrictive cases [66], one fixes both $\omega_j = \xi_{\nu,j}$ and $r_k = \xi_{\nu,k}/\xi_{\nu,k+1}$.

## Existing methods

A number of methods exist in the literature to evaluate (3.3) and (3.6). These include series expansion methods [19, 22, 80], convolutional approaches [64, 78, 83, 107], and projection-slice or Abel transform-based methods [58, 67, 91]. See [29] for a review of many of these early computational approaches. Unfortunately, these existing methods are either not applicable to the discrete case, require a particular choice of $\omega_j$ or $r_k$ due to the constraints of interpolation or quadrature subroutines, or suffer from low accuracy as a result of intermediate approximations. Therefore, extending these schemes to compute the fully nonuniform DHT with controllable accuracy is not straightforward.

A notable contribution is [78], which describes a fully nonuniform fast Hankel transform. This work takes the popular convolutional approach, using a change of variables to reformulate the Hankel transform as a convolution with a kernel whose Fourier transform is known. However, its accuracy is limited by the need for a quadrature rule on the nonuniform points $r_k$. The authors use an irregular trapezoidal rule for this purpose, which is not high-order accurate. This method also requires the computation of the inverse NUFFT using conjugate gradients. For even moderately clustered points or frequencies, this inverse problem is extremely ill-conditioned, and thus the number of required iterations can be prohibitive. This method is therefore suitable for "quasi-equispaced" points and frequencies, but is not tractable in general.

More recently, butterfly algorithms [75, 89, 94] were introduced as a broadly applicable methodology for rapidly computing oscillatory transforms including the nonuniform DHT. However, these algorithms require a precomputation or factorization stage for each new set of $\omega_j$ and $r_k$. Such precomputations can, unfortunately, be a bottleneck for applications in which these evaluation points change with each iteration or application of the transform. In addition, storing the butterfly factorization in memory can be prohibitive for very large transforms at high accuracies. In order to provide a precomputation-free, low-memory DHT, [117] employs a combination of asymptotic expansions and Bessel function identities evaluated using the equispaced FFT. The resulting scheme is applicable to equispaced or perturbed "quasi-equispaced" grids in space and frequency, for example $\omega_j = \xi_{0,j}$ and $r_k = \xi_{0,k}/\xi_{0,n+1}$.

### Novelty of this work

We describe here a precomputation-free nonuniform fast Hankel transform (NUFHT) which generalizes [117] to the fully nonuniform setting in a number of ways. First, we employ an adaptive partitioning scheme which, for any choice of $\omega_j$ and $r_k$, subdivides the matrix with entries $J_\nu(\omega_j r_k)$ into blocks for which matrix-vector products can be evaluated efficiently. Second, we use the NUFFT to evaluate asymptotic expansions for nonuniform $r_k$ and $\omega_j$. Finally, we utilize the low-rank expansion of $J_\nu$ given in [127] in the local regime where asymptotic expansions are not applicable. We derive error bounds for this low-rank expansion, allowing us to choose all approximation parameters automatically by analysis which guarantees that the resulting error is bounded by the user-specified tolerance $\varepsilon$.

### Outline of the chapter

The chapter is organized as follows. In Section 3.2 we give a high level view of our algorithm, omitting technical details. Then in Section 3.3 we study the local and asymptotic expansions of Bessel functions which serve as the key building blocks of the algorithm. Afterward, in Section 3.4, we provide a detailed description of the algorithm and its associated complexity. Various numerical examples are provided in Section 3.5, and we conclude with some additional discussion in Section 3.6.

## 3.2 Overview of the algorithm

To more concisely describe our approach, we write the DHT (3.6) as the equivalent matrix-vector product with $\mathbf{A} \in \mathbb{R}^{m \times n}$

$$\mathbf{g} = \mathbf{A}\mathbf{f}, \qquad \mathbf{A}(j,k) = J_\nu(\omega_j r_k). \tag{3.7}$$

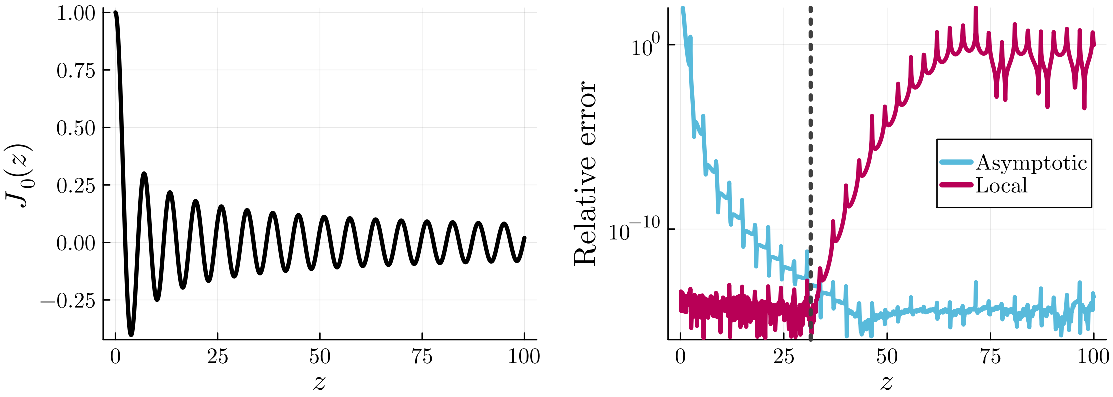


**Figure 3.1:** Bessel function $J_0(z)$ and pointwise relative error in approximating $J_0(z)$ using 31-term local and 4-term asymptotic expansions. Dotted vertical line shows crossover point where both expansions are accurate to $\varepsilon = 10^{-12}$.

The matrix **A** is in general full rank and possesses complex oscillatory structure. As a result, no straightforward fast algorithm exists to apply the full matrix **A** to a vector. However, we design an NUFHT by noting that certain *blocks* $\mathbf{A}(j_0 : j_1, k_0 : k_1)$ are able to be applied to a vector rapidly using analytical expansions of the underlying Bessel function $J_\nu$.

When the argument $\omega_j r_k$ is small, $J_\nu$ is smooth and essentially non-oscillatory, and we use a closed-form local expansion which approximates $J_\nu$ in terms of Chebyshev polynomials, yielding a low-rank approximation to various matrix blocks that can be applied to a vector in linear time. When the argument $\omega_j r_k$ is large, we use a classical asymptotic expansion which expresses $J_\nu$ as a sum of a small number of decaying sinusoids, and can therefore be applied to a vector in quasilinear time using the NUFFT. Figure 3.1 shows the oscillatory behavior of $J_0$, as well as the accuracy of these local and asymptotic expansions.

By analyzing the error in these two expansions, we can choose a crossover point $z$ such that an $L$-term local expansion and an $M$-term asymptotic expansion are both guaranteed to be accurate to the desired tolerance $\varepsilon$ in the regions $\omega_j r_k \leq z$ and $\omega_j r_k > z$ respectively. Next, we adaptively subdivide **A** into disjoint blocks so that either $\omega_j r_k \leq z$ or $\omega_j r_k > z$ for all $\omega_j$ and all $r_k$ in each block. This leaves only a few small blocks with $\omega_j r_k \approx z$ whose entries can be directly computed,

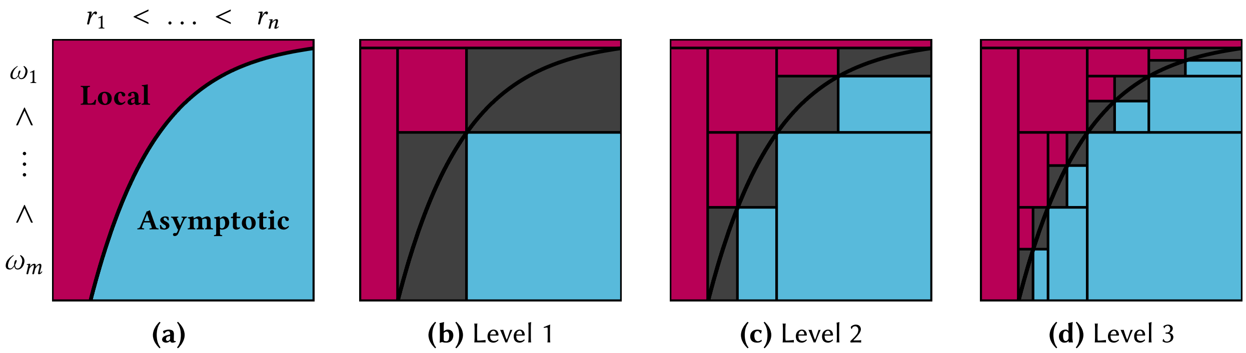


**Figure 3.2:** Splitting of Hankel transform matrix **A** along the curve $\omega r = z$ into local and asymptotic regions. Adaptive subdivision of **A** into corresponding local (red), asymptotic (blue), and mixed (gray) sub-blocks at various levels.

and which can be directly applied. Figure 3.2 shows a Hankel transform matrix **A** divided into local and asymptotic entries along the curve $\omega r = z$, as well as the corresponding adaptive subdivision of the matrix into blocks which can be rapidly applied. Following the subdivision step, all that remains is to apply each of the disjoint blocks of **A** to **f** using the corresponding fast method.

## 3.3 Bessel function approximations

We now describe local and asymptotic expansions of the Bessel function $J_\nu(\omega r)$, and provide error analysis by which one can select the number of terms needed in each expansion to assure $\varepsilon$ accuracy in both regimes.

### 3.3.1 The Wimp expansion

Near the origin, $J_\nu$ is a smooth and essentially non-oscillatory function. As a result, $J_\nu(\omega r)$ is a numerically low-rank function of all sufficiently small inputs $\omega$ and $r$. Fortuitously, one such low-rank expansion — which we refer to as the *Wimp expansion* — is available in closed form for integer $\nu$ [127]. In the case that $\nu$ is even, for all $|r| \leq 1$ we have

$$\begin{aligned} J_\nu(\omega r) &= \sum_{\ell=0}^{\infty} C_\ell(\omega)\, T_{2\ell}(r) \\ C_\ell(\omega) &= \delta_\ell\, J_{\frac{\nu}{2}+\ell}(\omega)\, J_{\frac{\nu}{2}-\ell}(\omega), \end{aligned} \tag{3.8}$$

where $\delta_0 = 1$ and $\delta_\ell = 2$ for all $\ell \neq 0$, $T_{2\ell}$ is the Chebyshev polynomial of the first kind of order $2\ell$. A similar expansion exists for $\nu$ odd [127, p. 2.23].

In order to employ the Wimp expansion to compute local terms within the Hankel transform, we must determine the number of terms $L$ needed to construct an $\varepsilon$-accurate approximation to $J_\nu(\omega r)$ on a given rectangle $(\omega, r) \in [0, \Omega] \times [0, R]$. The following lemma provides a bound on the induced truncation error in the Wimp expansion as a function of the order $\nu$, the space-frequency product $\Omega R$, and the number of retained terms $L$.

**Lemma 3.1.** *Truncating the Wimp expansion after L terms gives*

$$\left| J_\nu(\omega r) - \sum_{\ell=0}^{L-1} C_\ell(\omega R) T_{2\ell}\left(\frac{r}{R}\right) \right| \leq \frac{2\exp\left\{\frac{\nu}{2}(\beta-\gamma) + L(\beta+\gamma)\right\}}{1 - e^{\beta+\gamma}} =: B^{\text{LOC}}_{\nu,L}(\Omega R) \tag{3.9}$$

*for all* $\omega \in [0, \Omega], r \in [0, R]$, *where*

$$\psi(p) := \log p + \sqrt{1-p^2} - \log\left(1 + \sqrt{1-p^2}\right) \tag{3.10}$$

$$\beta := \psi\left(\frac{\Omega R}{2L+\nu}\right) \tag{3.11}$$

$$\gamma := \begin{cases} \psi\left(\frac{\Omega R}{2L-\nu}\right) & L > \frac{\nu}{2} \\ 0 & \text{otherwise} \end{cases} \tag{3.12}$$

*Proof.* For $\nu$ even, the truncation error after $L$ terms is bounded by

$$\left|\sum_{\ell=L}^{\infty} C_\ell(\omega R) T_{2\ell}\left(\frac{r}{R}\right)\right| \le 2\sum_{\ell=L}^{\infty}\left|J_{\frac{\nu}{2}+\ell}\left(\frac{\omega R}{2}\right)\right|\left|J_{\frac{\nu}{2}-\ell}\left(\frac{\omega R}{2}\right)\right|. \tag{3.13}$$

Define $p_\ell(\omega) := \omega R/(\nu + 2\ell)$. Then by Siegel's bound [90, p. 10.14.5] we have

$$\left|J_{\frac{\nu}{2}+\ell}\left(\frac{\omega R}{2}\right)\right| = \left|J_{\frac{\nu}{2}+\ell}\left(\left(\frac{\nu}{2}+\ell\right)p_\ell(\omega)\right)\right| \tag{3.14}$$

$$\le \exp\left\{\left(\frac{\nu}{2}+\ell\right)\psi\big(p_\ell(\omega)\big)\right\} \tag{3.15}$$

$$\le \exp\left\{\left(\frac{\nu}{2}+\ell\right)\beta\right\}, \tag{3.16}$$

where the last inequality follows from the fact that $\psi$ is an increasing function on $(0,1)$, and thus $\psi\big(p_\ell(\omega)\big) \le \beta < 0$ for all $\ell \ge L+1$ and all $\omega \in [0,\Omega]$.

If $L > \frac{\nu}{2}$, we define $q_\ell(\omega) := \omega R/(2\ell - \nu)$ and apply Siegel's bound again to obtain

$$\left|J_{\frac{\nu}{2}-\ell}\left(\frac{\omega R}{2}\right)\right| = \left|J_{\ell-\frac{\nu}{2}}\left(\left(\ell-\frac{\nu}{2}\right)q_\ell(\omega)\right)\right| \le \exp\left\{\left(\ell-\frac{\nu}{2}\right)\gamma\right\}. \tag{3.17}$$

If $L \le \frac{\nu}{2}$, Siegel's bound does not apply and we use instead the simple bound $\left|J_{\frac{\nu}{2}-\ell}\left(\frac{\omega R}{2}\right)\right| \le 1$, which is equivalent to taking $\gamma = 0$.

All that remains is to apply a geometric series argument

$$\left|J_\nu(\omega r) - \sum_{\ell=0}^{L-1} C_\ell(\omega R) T_{2\ell}\left(\frac{r}{R}\right)\right| \le 2\sum_{\ell=L}^{\infty}\exp\left\{\left(\frac{\nu}{2}+\ell\right)\beta + \left(\ell-\frac{\nu}{2}\right)\gamma\right\} \tag{3.18}$$

$$= 2\exp\left\{\frac{\nu}{2}(\beta-\gamma)\right\}\sum_{\ell=L}^{\infty}\left(e^{\beta+\gamma}\right)^\ell \tag{3.19}$$

$$= \frac{2\exp\left\{\frac{\nu}{2}(\beta-\gamma) + L(\beta+\gamma)\right\}}{1-e^{\beta+\gamma}} \tag{3.20}$$

A similar calculation can be carried out for $\nu$ odd. □

Lemma 3.1 is rather opaque regarding the impact of the various parameters on the error because we have not utilized any simplifying bounds on the function $\psi$, as done in [98, Lemma 1] for large $\nu$. However, our analysis takes into account the decay in both $J_{\frac{\nu}{2}+\ell}$ *and* $J_{\frac{\nu}{2}-\ell}$, thus remaining relatively tight for small $\nu$. It is therefore well-suited to our purposes because, given $z, L > 0$, it provides a bound $B^{\text{loc}}_{\nu,L}(z)$ on the pointwise error in approximating any block of the matrix $J_\nu(\omega_j r_k)$ for which $\omega r \leq z$ using the $L$-term Wimp expansion.

This expansion is highly beneficial from a computational perspective, as it yields an analytical rank-$L$ approximation to any block of $\boldsymbol{A}$ for which $\omega_j r_k$ is sufficiently small

$$\mathbf{A}(j_0 : j_1, k_0 : k_1) \approx \mathbf{C}\mathbf{T}^\top \tag{3.21}$$

where $\mathbf{C} \in \mathbb{R}^{(j_1-j_0+1)\times L}$ and $\mathbf{T} \in \mathbb{R}^{(k_1-k_0+1)\times L}$ with entries

$$\mathbf{C}(j,\ell) = C_{\ell-1}(\omega_j r_{k_1}) \qquad \text{and} \qquad \mathbf{T}(k,\ell) = T_{2\ell-2}\left(\frac{r_k}{r_{k_1}}\right), \tag{3.22}$$

where we have scaled $\omega_j$ and $r_k$ by the largest distance $r_{k_1}$ to ensure that the argument of the Chebyshev polynomial is at most one. For a block of $\mathbf{A}$ of size $m_b \times n_b$, the low-rank approximation given by the Wimp expansion can be applied to a vector in $O\big(L(m_b + n_b)\big)$ time by first applying $\boldsymbol{T}^\top$ then applying $\boldsymbol{C}$.

### 3.3.2 Hankel's expansion

Away from the origin, $J_\nu$ exhibits essentially sinusoidal oscillation with period $2\pi$. This statement is made precise by Hankel's asymptotic expansion, which states that for argument $\omega r \to \infty$

$$J_\nu(\omega r) \sim \sqrt{\frac{2}{\pi \omega r}} \left( \cos\left(\omega r + \varphi\right) \sum_{\ell=0}^{\infty} \frac{(-1)^\ell a_{2\ell}(\nu)}{(\omega r)^{2\ell}} - \sin\left(\omega r + \varphi\right) \sum_{\ell=0}^{\infty} \frac{(-1)^\ell a_{2\ell+1}(\nu)}{(\omega r)^{2\ell+1}} \right) \tag{3.23}$$

where $\varphi := -\frac{(2\nu+1)\pi}{4}$ and

$$a_\ell(\nu) := \frac{(4\nu^2-1)(4\nu^2-3)\dots(4\nu^2-(2\ell-1)^2)}{\ell!\,8^\ell}. \tag{3.24}$$

Rearranging this expansion, we obtain an expansion which can be evaluated using two NUFFTs and diagonal scalings, and whose remainder is bounded by the size of the first neglected terms [122, Section 7.3]

$$\begin{aligned}\left|J_\nu(\omega r) - \sqrt{\frac{2}{\pi}} \sum_{\ell=0}^{M-1} \left[ \frac{(-1)^\ell a_{2\ell}(\nu)}{\omega^{2\ell+\frac{1}{2}}} \operatorname{Re}\left(\frac{e^{i(\omega r+\varphi)}}{r^{2\ell+\frac{1}{2}}}\right) - \frac{(-1)^\ell a_{2\ell+1}(\nu)}{\omega^{2\ell+\frac{3}{2}}} \operatorname{Im}\left(\frac{e^{i(\omega r+\varphi)}}{r^{2\ell+\frac{3}{2}}}\right)\right]\right| \\ \leq \sqrt{\frac{2}{\pi}} \left(\frac{|a_{2M}(\nu)|}{z^{2M+\frac{1}{2}}} + \frac{|a_{2M+1}(\nu)|}{z^{2M+\frac{3}{2}}}\right) =: B^{\textsc{asy}}_{\nu,M}(z)\end{aligned} \tag{3.25}$$

for all $\omega r \geq z$.

The computational advantage of this expansion is that the $2M$-term asymptotic expansion of any block of $\boldsymbol{A}$ can be rapidly applied to a vector $\mathbf{c}$ using $2M$ Type-III NUFFTs

$$\begin{aligned}\mathsf{A}(j_0:j_1,k_0:k_1)\,\mathbf{c} \approx \sqrt{\frac{2}{\pi}} \sum_{\ell=0}^{M-1} (-1)^\ell \Bigg[ a_{2\ell}(\nu) \mathbf{D}_\omega^{-2\ell-\frac{1}{2}} \operatorname{Re}\left(e^{i\varphi} \mathbf{F} \mathbf{D}_r^{-2\ell-\frac{1}{2}} \mathbf{c}\right) \\ - a_{2\ell+1}(\nu) \mathbf{D}_\omega^{-2\ell-\frac{3}{2}} \operatorname{Im}\left(e^{i\varphi} \mathbf{F} \mathbf{D}_r^{-2\ell-\frac{3}{2}} \mathbf{c}\right)\Bigg]\end{aligned} \tag{3.26}$$

where $\mathbf{F} \in \mathbb{C}^{(j_1-j_0+1)\times(k_1-k_0+1)}$ is the Type-III nonuniform DFT matrix with entries $\mathbf{F}_{jk} := e^{i\omega_{(j_0+j-1)} r_{(k_0+k-1)}}$, and the diagonal scaling matrices are given by $\mathbf{D}_\omega := \operatorname{diag}(\omega_{j_0}, \dots, \omega_{j_1})$, and $\mathbf{D}_r := \operatorname{diag}(r_{k_0}, \dots, r_{k_1})$.

### 3.3.3 Determining order of expansions and crossover point

With these error bounds in hand, we precompute the parameters $z^M_{\nu,\varepsilon}$ and $L^M_{\nu,\varepsilon}$ for tolerances $\varepsilon = 10^{-4}, \dots, 10^{-15}$, orders $\nu = 1, \dots, 100$, and number of asymptotic expansion terms $M = 1, \dots, 20$:

- $z^M_{\nu,\varepsilon}$ such that $M$-term Hankel expansion of $J_\nu(\omega r)$ is $\varepsilon$-accurate $\forall\ \omega r > z^M_{\nu,\varepsilon}$,

- $L^M_{\nu,\varepsilon}$ such that $L^M_{\nu,\varepsilon}$-term Wimp expansion of $J_\nu(\omega r)$ is $\varepsilon$-accurate $\forall\ \omega r \leq z^M_{\nu,\varepsilon}$.

First, the crossover points $z^M_{\nu,\varepsilon}$ are computed using Newton's method on the function $\zeta(z) := B^{\text{ASY}}_{\nu,M}(z) - \varepsilon$. Then the number of local expansion terms $L^M_{\nu,\varepsilon}$ are taken to be the smallest integer such that $B^{\text{LOC}}_{\nu,L}\left(z^M_{\nu,\varepsilon}\right) < \varepsilon$. These tables are precomputed once when the library is installed, and even this precomputation requires only a few seconds on a laptop.

With these tables stored, for any order $\nu$ we can look up a pair of complementary local and asymptotic expansions with error everywhere bounded by the requested tolerance $\varepsilon$. The only remaining free parameter is the number of asymptotic terms $M$. This parameter is selected based on various numerical experiments which maximize speed by balancing the cost of the local, asymptotic, and direct evaluations. In our implementation, we use the heuristic

$$M_{\nu,\varepsilon} = \min\left(\left\lfloor 1 + \frac{\nu}{5} - \frac{\log_{10}(\varepsilon)}{4}\right\rfloor, 20\right). \tag{3.27}$$

## 3.4 The Nonuniform Fast Hankel Transform

We now describe our NUFHT algorithm in detail, emphasizing the process by which $\mathbf{A}$ is adaptively subdivided into blocks using the results of the above error analysis.

### 3.4.1 Subdividing the matrix into blocks by expansion

Having established error bounds which allow us to automatically select the number of asymptotic terms $M$, local terms $L$, and crossover point $z$ given a tolerance $\varepsilon$ and order $\nu$, we subdivide the matrix $\mathbf{A}$ into three sets of blocks, each of which can be efficiently applied to a vector as described above:

- Local blocks $\mathscr{L} = \left\{\mathbf{A}(j_0 : j_1, k_0 : k_1) \mid \omega_j r_k \leq z \;\forall\; j_0 \leq j \leq j_1,\ k_0 \leq k \leq k_1\right\}$
- Asymptotic blocks $\mathscr{A} = \left\{\mathbf{A}(j_0 : j_1, k_0 : k_1) \mid \omega_j r_k > z \;\forall\; j_0 \leq j \leq j_1,\ k_0 \leq k \leq k_1\right\}$

- Direct blocks $\mathscr{D}$ which are small enough that no fast expansion is needed

In order to determine a subdivision of **A** into blocks of these three types, we initialize a set of *mixed* blocks $\mathscr{M} = \{(1:m, 1:n)\}$, each of which contains a mix of local and asymptotic entries. We then chose an index pair $(j,k)$ such that $\omega_j r_k \approx z$. This index subdivides the block into four new sub-blocks with $(j,k)$ at the center, so that the upper left block can be applied using the local expansion and is appended to $\mathscr{L}$, and the lower right block using the asymptotic expansion and is appended to $\mathscr{A}$.

The remaining lower left and upper right blocks each still contain a mix of local and asymptotic entries. If they are of sufficiently small size $m_b \times n_b$ with $m_b n_b < \texttt{min\_size}$ — a user-defined parameter which is taken to be 1024 by default — they can be evaluated directly and are appended to $\mathscr{D}$. Otherwise they are appended back to $\mathscr{M}$, and we continue the subdivision process recursively.

This method yields a valid partition for any choice of $(j,k)$, but for efficiency these indices are chosen to maximize the number of matrix entries which can be applied using a fast expansion, i.e. the sizes of the upper left and lower right blocks. This is done by solving the following constrained optimization problem

$$
\begin{aligned}
(j,k) &= \textsc{SplitIndices}(r_1,\dots,r_n,\omega_1,\dots,\omega_m,z) && (3.28)\\
&:= \begin{cases} \underset{j,k\in\mathbb{Z}}{\arg\max} & (j-j_0)(k_1-k)+(j_1-j)(k-k_0) \\ \text{subject to} & j_0 \le j \le j_1 \\ & k_0 \le k \le k_1 \\ & \omega_j r_k \le z \end{cases} && (3.29)
\end{aligned}
$$

This problem can be solved exactly in $O(j_1-j_0+k_1-k_0)$ time. However, computing the exact optimal splitting indices for every box gives a negligible speedup to the overarching Hankel transform compared to a simpler, quasi-optimal scheme. In practice it is sufficient to choose a small number

**Algorithm 1:** Block subdivision of Hankel transform matrix

```
Subdivide($\{r_k\}_{k=1}^{n}$, $\{\omega_j\}_{j=1}^{m}$, $z$, min_size):
    $\mathscr{L} = \mathscr{A} = \mathscr{D} = \emptyset$
    $\mathscr{M} = \{(1:m, 1:n)\}$
    while $\mathscr{M} \neq \emptyset$ do
        Pop a submatrix $\mathbf{A}(j_0:j_1, k_0:k_1)$ from $\mathscr{M}$
        $(j, k) = $ SplitIndices($\{r_k\}_{k=k_0}^{k_1}$, $\{\omega_j\}_{j=j_0}^{j_1}$, $z$)
        Append $\mathbf{A}(j_0:j, k_0:k)$ to $\mathscr{L}$
        Append $\mathbf{A}(j{+}1:j_1, k{+}1:k_1)$ to $\mathscr{A}$
        Append $\mathbf{A}(j_0:j, k{+}1:k_1)$ to $\mathscr{M}$ if $(j{-}j_0{+}1)(k_1{-}k) >$ min_size else $\mathscr{D}$
        Append $\mathbf{A}(j{+}1:j_1, k_0:k)$ to $\mathscr{M}$ if $(j_1{-}j)(k_1{-}k{+}1) >$ min_size else $\mathscr{D}$
    end
    return $(\mathscr{L}, \mathscr{A}, \mathscr{D})$
```

of equispaced indices $j \in \{j_0, \ldots, j_1\}$, compute the corresponding $k = \arg\max\{k \mid r_k \leq \frac{z}{\omega_j}\}$ for each $j$, and choose $(j, k)$ as the pair which minimizes the objective function of (3.29) among this small collection.

### 3.4.2 Complexity analysis

We now analyze the computational complexity of the proposed approach. In order to do so, we must first comment on the complexity of the NUFFT, which is an important subroutine in our method. Most analysis-based NUFFT codes — including the `FINUFFT` library [3] which we use in our NUFHT implementation — consist of three steps. First, delta masses centered at each non-uniform point are convolved with a *spreading function* which smears them onto a fine $N$-point uniform grid. Then, a standard equispaced FFT is computed on the fine grid. Finally, a diagonal de-convolution with the Fourier transform of the spreading function is applied to reverse the effect of the original smearing. For a more complete description of this NUFFT method, see [3, 34, 51]. For $n$ points $r_k$ and $m$ frequencies $\omega_j$, spreading the input points to a finer grid is $O(n)$, the FFT on the finer grid is $O(N \log N)$, and the global deconvolution at the output frequencies is $O(m)$. For the Type-III NUFFT, the size $N$ of the fine grid typically scales linearly with the

**Algorithm 2:** Nonuniform fast Hankel transform

```
NUFHT($\nu, \varepsilon, \{r_k\}_{k=1}^n, \{c_k\}_{k=1}^n, \{\omega_j\}_{j=1}^m$):
    $\boldsymbol{g} = \mathbf{0} \in \mathbb{C}^m$
    Choose $M_{\nu,\varepsilon}$ using (3.27)
    Look up $L_{\nu,\varepsilon}^M$ and $z_{\nu,\varepsilon}^M$ from pre-computed tables
    Set min_size
    $(\mathscr{L}, \mathscr{A}, \mathscr{D})$ = Subdivide($\{r_k\}_{k=1}^n, \{\omega_j\}_{j=1}^m, z_{\nu,\varepsilon}^M$, min_size)
    for $\mathscr{B} \in (\mathscr{L}, \mathscr{A}, \mathscr{D})$ do
        for $\mathbf{A}(j_0 : j_1, k_0 : k_1) \in \mathscr{B}$ do
            $\boldsymbol{g}(j_0 : j_1)$ += $\boldsymbol{A}(j_0 : j_1, k_0 : k_1)\, \boldsymbol{c}(k_0 : k_1)$ using expansion (3.21) for $\mathscr{L}$, (3.26)
              for $\mathscr{A}$, or direct evaluation for $\mathscr{D}$
        end
    end
    return $\boldsymbol{g}$
```

space-frequency product $p := (\omega_m - \omega_1)(r_n - r_1)$ [3, 51]. Therefore the total cost of the NUFFT is $O(n + m + p \log p)$. Applying this fact in each asymptotic block in the Hankel transform matrix, and adding the cost of applying local and direct blocks, we can now analyze the complexity of the entire NUFHT method shown in Algorithm 2.

**Theorem 3.2.** *Take $0 \le \omega_1 < \cdots < \omega_m \le \Omega$ and $0 \le r_1 < \cdots < r_n \le R$ and define the space-frequency product $p := \Omega R$. Then the complexity of computing the NUFHT of order $\nu$ to tolerance $\varepsilon$ using Algorithm 2 is*

$$O\Big((L + M)(m + n) \log \min(n, m) + Mp \log p\Big),$$

*where $L$ and $M$ are the number of local and asymptotic terms respectively chosen according to $\nu$ and $\varepsilon$.*

*Proof.* For notational clarity we suppress the dependence of $z_{\nu,\varepsilon}^M$ on its parameters and simply denote it as $z$. If $\omega_j r_k \le z$ for all $j = 1, \ldots, n$ and $k = 1, \ldots, m$ then only the $L$-term low-rank local expansion is used, which can be applied in $O(L(m+n))$ time. If instead $\omega_j r_k > z$ everywhere, then only the $M$-term asymptotic expansion is used, which can be applied using the Type-III NUFFT in $O(M(m + n + p \log p))$ complexity.

Otherwise consider the case where $\mathbf{A}$ contains both local and asymptotic entries. First, note that the number of levels $N_{\text{level}}$ scales like $\mathcal{O}(\log\min(n,m))$. The cost of determining the splitting indices $(j,k)$ for each box $\mathbf{A}(j_0 : j_1, k_0 : k_1)$ is $\mathcal{O}(j_1 - j_0 + k_1 - k_0)$, and thus the total cost of subdivision at each level is $\mathcal{O}(m+n)$. Therefore the total cost of subdividing $\mathbf{A}$ is $\mathcal{O}((m+n)\log\min(n,m))$.

Now, without loss of generality, assume $\omega_1 \le z/r_n < \omega_2$ and $r_1 \le z/\omega_m < r_2$. If this were not the case, we would have blocks which can be evaluated using a single expansion as described above without affecting the complexity. After step $\ell$ of subdividing every mixed block, we obtain $2^\ell$ new mixed blocks, $2^{\ell-1}$ new local blocks, and $2^{\ell-1}$ new asymptotic blocks. Let the local blocks be of size $m_{\ell,b}^{(\text{loc})} \times n_{\ell,b}^{(\text{loc})}$ for $b = 1, \dots, 2^{\ell-1}$. Then,

$$\sum_{b=1}^{2^{\ell-1}} m_{\ell,b}^{(\text{loc})} \le m, \qquad \text{and} \qquad \sum_{b=1}^{2^{\ell-1}} n_{\ell,b}^{(\text{loc})} \le n. \tag{3.30}$$

An analogous fact holds for the asymptotic blocks.

Therefore, the total cost of local evaluation is

$$\begin{aligned}\sum_{\ell=1}^{N_{\text{level}}} \sum_{b=1}^{2^{\ell-1}} \mathcal{O}\left(L\left(m_{\ell,b}^{(\text{loc})} + n_{\ell,b}^{(\text{loc})}\right)\right) &= \sum_{\ell=1}^{N_{\text{level}}} \mathcal{O}(L(m+n)) \\ &= \mathcal{O}\big(L(m+n)\log\min(n,m)\big).\end{aligned} \tag{3.31}$$

Let $p_{\ell,b}$ be the space-frequency product of box $b$ at level $\ell$. The total space frequency product $p$ is the area of the rectangle $R := [\omega_1, \omega_m] \times [r_1, r_n]$, and all asymptotic boxes occupy disjoint sub-rectangles of $R$. Therefore the sum of their areas is bounded by the area of $R$, so that

$$\sum_{\ell=1}^{N_{\text{level}}} \sum_{b=1}^{2^{\ell-1}} p_{\ell,b} \le p.$$

Then by Hölder's inequality we obtain

$$\sum_{\ell=1}^{N_{\text{level}}} \sum_{b=1}^{2^{\ell-1}} p_{\ell,b} \log p_{\ell,b} \leq \left( \sum_{\ell=1}^{N_{\text{level}}} \sum_{b=1}^{2^{\ell-1}} p_{\ell,b} \right) \left( \max_{\ell,b} \log p_{\ell,b} \right) \leq p \log p. \tag{3.32}$$

The total cost of asymptotic evaluation via the Type-III NUFFT is therefore

$$\begin{aligned} &\sum_{\ell=1}^{N_{\text{level}}} \sum_{b=1}^{2^{\ell-1}} \mathcal{O}\left( M \left( m_{\ell,b}^{(\text{asy})} + n_{\ell,b}^{(\text{asy})} + p_{\ell,b}^{(\text{asy})} \log p_{\ell,b}^{(\text{asy})} \right) \right) \\ &\qquad = \sum_{\ell=1}^{N_{\text{level}}} \mathcal{O}(M(m+n)) + \sum_{\ell=1}^{N_{\text{level}}} \sum_{b=1}^{2^{\ell-1}} \mathcal{O}\Big( M \Big( p_{\ell,b}^{(\text{asy})} \log p_{\ell,b}^{(\text{asy})} \Big) \Big) \\ &\qquad = \mathcal{O}\big( M(m+n) \log \min(n,m) + Mp \log p \big). \end{aligned} \tag{3.33}$$

We subdivide until all direct blocks are all of size $m_b \times n_b$ with $m_b n_b = \mathcal{O}(1)$. Thus the cost of computing the dense matvec with each direct block is $\mathcal{O}(1)$, and the number of direct blocks is $\mathcal{O}(m+n)$. Therefore the total direct evaluation cost is $\mathcal{O}(m+n)$. Summing the cost of matrix subdivision, as well as local, asymptotic, and direct evaluation gives the result. □

In typical applications the maximum point $r_n$ is fixed by, for example, the support of the function $f$ whose Fourier transform is desired, and the maximum frequency $\omega_m$ at which the transform is computed grows linearly with $n$. The following corollary studies this common scenario, which includes Schlömilch expansions and Fourier-Bessel series. For notational conciseness, we consider the number of terms $L$ and $M$ in each expansion as constants here.

**Corollary 3.3.** *Take $0 \leq \omega_1 < \cdots < \omega_n$ and $0 \leq r_1 < \cdots < r_n$ such that the space-frequency product $\omega_n r_n = \mathcal{O}(n)$. Then the complexity of computing the NUFHT using Algorithm 2 is $\mathcal{O}(n \log n)$.*

**Remark 3.1.** If $r_1 \gg 0$, the space-frequency product $p = \Omega R$ may be large while the recentered space-frequency product $\tilde{p} := \Omega(r_n - r_1)$ is small. One can then use the Neumann addition formula [90, p. 10.23.2]

$$J_\nu(\omega_j r_k) = \sum_{\ell=-\infty}^{\infty} J_{\nu-\ell}(\omega_j r_1) J_\ell(\omega_j r_k - \omega_j r_1) \tag{3.34}$$

so that the Hankel transform can be approximated by

$$\sum_{k=1}^{m} c_k J_\nu(\omega_j r_k) \approx \sum_{\ell=-N}^{N} J_{\nu-\ell}(\omega_j r_1) \sum_{k=1}^{m} c_k J_\ell\Big(\omega_j (r_k - r_1)\Big). \tag{3.35}$$

Each inner sum above can be computed with an NUFHT of order $\ell$ with space-frequency product $\tilde{p} \ll p$. The outer sum converges rapidly in $N$ for $N > \tilde{p}$, and thus the $O(\tilde{p}^2 \log \tilde{p})$ cost of evaluating (3.35) may be smaller than the $O(p \log p)$ cost of using Algorithm 2 directly. The same procedure can be used in $\omega$ if $\omega_1 \gg 0$.

**Remark 3.2.** There exist butterfly factorization-based NUFFT methods that could be used to remove the dependence on the space-frequency product $p$ in Theorem 3.2 using linear algebraic approximations [94]. However, we find that the asymptotic dependence on $p$ is generally seen only in pathological cases, and thus choose to avoid the precomputations and memory requirements associated with butterfly methods.

## 3.5 Numerical experiments

In the following section, we perform a number of numerical experiments to validate the accuracy and complexity of our method. We close with two applications from Fourier analysis and numerical PDEs.

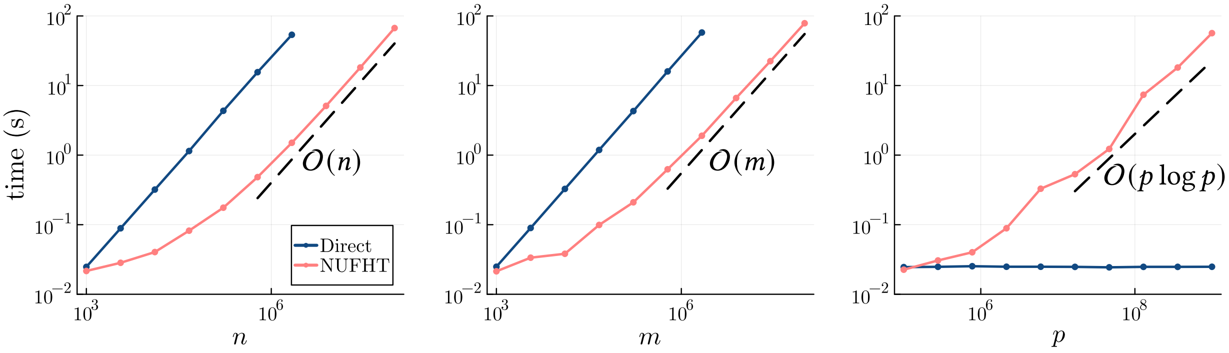


**Figure 3.3:** Scaling with $n$, $m$, and $p$ respectively, with the other variables held constant.

### 3.5.1 Comparison to direct evaluation

We start by empirically verifying the error analysis in Sections 3.3.1 and 3.3.2, and the asymptotic scaling analysis in Section 3.4.2 by comparing to direct evaluation of the Hankel transform.

#### 3.5.1.1 Asymptotic scaling

In order to study the impact of each of the relevant parameters in the scaling analysis of Theorem 3.2 independently, we take $n$ equispaced points $r_k$ in the interval $[0, \sqrt{10^5}]$ and $m$ equispaced frequencies $\omega_j$ in the interval $[0, p/\sqrt{10^5}]$. First, we fix $m = 10^3$ and $p = 10^5$ while increasing $n$. Then, we fix $n = 10^3$ and $p = 10^5$, this time increasing $m$. Finally, we fix both $n = m = 10^3$ while increasing $p$. Figure 3.3 shows the CPU time for the NUFHT as well as for direct summation in each of these scenarios. We observe the linear or quasilinear scaling expected from Theorem 3.2 with each of $n, m$, and $p$. Note in particular that the NUFHT scales with $p$ while direct summation does not. Therefore, if a DHT is desired with relatively few points with a very large space-frequency product, direct summation may give superior performance, although such circumstances are rare in practice.

Next, we study the more typical scenario where the space-frequency product $p$ grows linearly with $n$, as discussed in Corollary 3.3. Here we study two cases. First, we consider the Fourier-Bessel expansion where $\omega_j = \xi_{\nu,j}$ and $r_k = \xi_{\nu,k}/\xi_{\nu,n+1}$ with $n = m$. This is the direct analogue of the discrete Fourier transform as the points and frequencies are the scaled roots of the basis, and the resulting points and frequencies are quasi-equispaced for small to moderate $\nu$.

We also consider the case of exponentially distributed points and frequencies $\omega_j = r_j = 10^{\log_{10}(j)-\log_{10}(n)/2}$ with $n = m$. This is a somewhat pathological worst case scenario for our algorithm, as the simple calculation

$$\sqrt{\frac{\Omega z}{R}} = \underset{\frac{z}{R} \leq \omega \leq \Omega}{\arg\max} \, (\Omega - \omega) \left( R - \frac{z}{\omega} \right) \tag{3.36}$$

shows that if we subdivide a block with space frequency product $\Omega R$ at a point $(\omega, r)$ which lies on the curve $\omega r = z$, then the largest possible space-frequency product $p$ for the resulting lower right asymptotic block is achieved by taking $\omega$ to be the mid-point of $[z/R, \Omega]$ on a log scale. In other words, points and frequencies which are exponentially distributed result in the highest possible space-frequency product $p$ for every asymptotic block at every level. From Theorem 3.2, maximizing $p$ drives the cost of the NUFHT. This distribution of points and frequencies is also challenging because it leads to equally-sized square blocks at every level, which guarantees that all blocks are subdivided the maximum number of times before yielding sufficiently small direct blocks.

Figure 3.4 shows the CPU time needed to evaluate the NUFHT in the Fourier-Bessel and exponentially-distributed cases with $\nu = 0$ and $\varepsilon = 10^{-8}$. Both cases eventually demonstrate the expected $\mathcal{O}(n \log n)$ scaling. As a result of the challenges just discussed for the exponentially-distributed case, its runtime is up to an order of magnitude slower than the Fourier-Bessel series.

#### 3.5.1.2 Impact of the order and tolerance on runtime

As the order $\nu$ increases or the tolerance $\varepsilon$ decreases, the number of necessary terms $L$ and $M$ in the local and asymptotic expansions, respectively, both grow. From Theorem 3.2, we expect the runtime to grow linearly with $L + M$. Figure 3.4 shows the runtime of our method for various

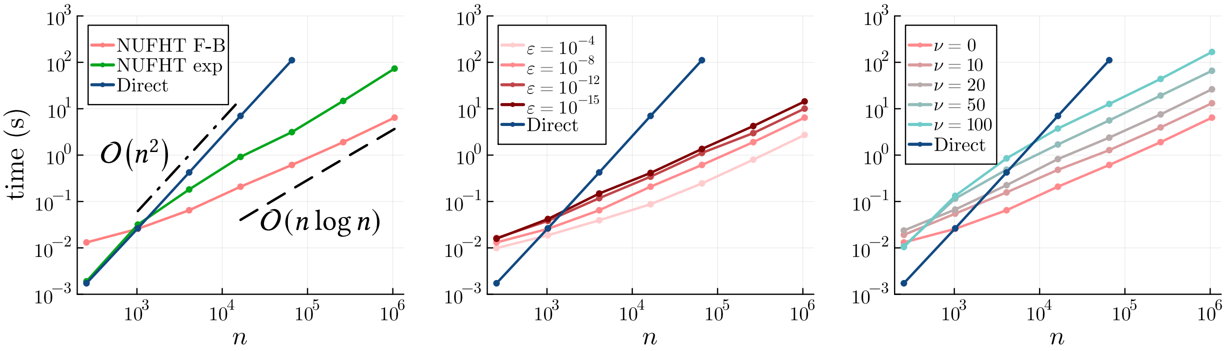


**Figure 3.4:** Scaling with $n$ for $p = O(n)$ test cases. In the first plot, we fix $\nu = 0, \varepsilon = 10^{-8}$ and time the NUFHT for both the Fourier-Bessel and exponentially distributed cases. In the second and third plots, we consider the Fourier-Bessel series only, and fix one of the parameters $\nu = 0$ and $\varepsilon = 10^{-8}$ while varying the other. The timings of direct summation and Fourier-Bessel series from the first plot are repeated in the other two plots for reference.

$\varepsilon$ with $\nu = 0$ held constant, as well as for multiple $\nu$ with $\varepsilon = 10^{-8}$ fixed. The $O(n \log n)$ scaling of the algorithm is similar in all cases, while the prefactors vary; a transform with $\varepsilon = 10^{-15}$ is about an order of magnitude slower than using $\varepsilon = 10^{-4}$, and an order $\nu = 100$ transform is almost two orders of magnitude slower than the order $\nu = 0$ equivalent.

### 3.5.1.3 Approximation error

Finally, we study the relative error in the output $\mathbf{g}$ as a function of the desired tolerance $\varepsilon$. To do this, we fix $n$ and form a sparse vector $\mathbf{f} \in \mathbb{R}^n$ with 1000 nonzero entries whose indices are selected at random and whose values are independent standard Gaussian. We evaluate the Fourier-Bessel series using the NUFHT with the full vector $\mathbf{f}$ as input, and denote the output as $\tilde{\mathbf{g}}$. We then use direct summation on only the nonzero entries to generate a reference result $\mathbf{g}$. Figure 3.5 shows the 2-norm relative error $\|\mathbf{g} - \tilde{\mathbf{g}}\|_2 / \|\mathbf{g}\|_2$ between the NUFHT and the reference. For small transforms with $n = 10^3$, the relative error demonstrates excellent agreement with the tolerance $\varepsilon$ down to $\varepsilon = 10^{-14}$ or so. This suggests that the analysis used in Section 3.3 to determine the necessary number of local and asymptotic terms is fairly tight. For larger transforms, however, the error saturates, and regardless of the tolerance $\varepsilon$ our method gives at most 9 digits

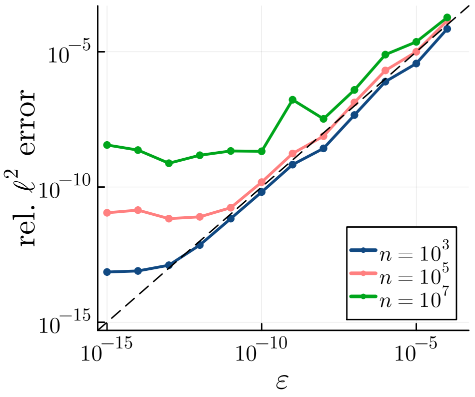


**Figure 3.5:** Relative 2-norm error $\|\mathbf{g} - \tilde{\mathbf{g}}\|_2 / \|\mathbf{g}\|_2$ as a function of tolerance $\varepsilon$ for a NUFHT of order $\nu = 0$ for various $n$.

of accuracy for transforms of size $n = 10^7$. This is a well-known limitation of existing NUFFT methods, for which the error generally scales like $n$ times machine precision [3, Remark 9].

### 3.5.2 Computing Fourier transforms of radial functions

For radial functions $f(\boldsymbol{r}) = f(\|\boldsymbol{r}\|)$ in $\mathbb{R}^d$, one can integrate out the radial variables analytically, reducing the $d$-dimensional Fourier integral to a single Hankel transform

$$\hat{f}(\boldsymbol{\omega}) = \int_{\mathbb{R}^d} f(\|\boldsymbol{r}\|)\, e^{i\boldsymbol{\omega}^\top \boldsymbol{r}}\, \mathrm{d}\boldsymbol{r} = \frac{(2\pi)^{\frac{d}{2}}}{\omega^{\frac{d}{2}-1}} \int_0^\infty f(r)\, J_{\frac{d}{2}-1}(\omega r)\, r^{\frac{d}{2}}\, \mathrm{d}r. \tag{3.37}$$

As discussed in Chapter 2, this situation is of particular relevance in spatial statistics and kernel learning problems (with the roles of $r$ and $\omega$ reversed) [63, 97, 126]. We consider the isotropic “singular Matérn” model proposed in [96] and fitted in Chapter 2, which we parameterize in two dimensions as

$$f(\boldsymbol{r}) := \|\boldsymbol{r}\|^{-\alpha} \left(\rho^2 + \|\boldsymbol{r}\|^2\right)^{-\nu-1} =: |r|^{-\alpha} f_0(r) \tag{3.38}$$

The parameters $\rho$, $\nu$, and $\alpha$ control the lengthscale, smoothness, and degree of “long memory” (i.e. slow decay in $\hat{f}$) respectively. The bounds $0 \leq \alpha < 2$ and $\nu > -\frac{\alpha}{2}$ guarantee that $f$ is integrable.

This function $f$ lacks a closed form Fourier transform which can be simply and stably evaluated, making numerical quadrature a promising approach, as illustrated in one dimension in [4].

We take $\alpha = 1.4, \rho = 0.01, \nu = 2$ so that we can consider $f$ to be compactly supported on $[0, 1]$ up to tolerance $\varepsilon = 10^{-8}$. We then compare two methods of computing $\hat{f}$. First, we use a Gauss-Jacobi quadrature rule on $[0, 1]$ with nodes $r_k$ and weights $w_k$. We utilize the NUFHT to compute the resulting sum

$$\begin{aligned} \hat{f}(\omega) &= 2\pi \int_0^1 r^{-\alpha} f_0(r)\, J_0(\omega r)\ \mathrm{d}r \\ &\approx 2\pi \sum_{k=1}^{n} w_k\, f_0(r_k)\, J_0(\omega r_k) \end{aligned} \tag{3.39}$$

doubling the number of nodes $m$ until the error in the computed integral is less than $\varepsilon$. Second, we build a two-dimensional quadrature rule in polar coordinates, using the same $m$-point Gauss-Jacobi rule in $r$ and a $t_k$-node trapezoidal rule in $\theta$ on each circle of radius $r_k$. We double the number of trapezoidal nodes $t_k$ in each circle until the error in the corresponding radial integral is less than $\varepsilon$. We then utilize the 2D NUFFT to compute the resulting double sum

$$\hat{f}(\omega) = \frac{1}{4\pi^2} \int_0^{2\pi} \int_0^1 r^{-\alpha} f_0(r)\, e^{-i\omega r \cos\theta}\ \mathrm{d}r\, \mathrm{d}\theta \tag{3.40}$$

$$\approx \frac{1}{4\pi^2} \sum_{k=1}^{n} w_k\, f_0(r_k)\, \frac{2\pi}{t_k} \sum_{s=1}^{t_k} \exp\left\{-i\omega r_k\ \cos\left(\frac{2\pi s}{t_k}\right)\right\}. \tag{3.41}$$

For both methods, we use tolerance $\varepsilon = 10^{-8}$ and evaluate the Fourier transform $\hat{f}$ at $m$ equispaced points $\omega_j \in [0, \omega_{\max}]$.

If only low frequencies $\omega$ are desired, e.g. $\omega_{\max} = 64$, the integrands are only mildly oscillatory and few trapezoidal nodes are required. In combination with the relative ease of amortizing costs in the NUFFT, the two-dimensional transform is often faster than the NUFHT. However, for larger $\omega_{\max}$ the integrands become more oscillatory, and in two dimensions $n = O(\omega_{\max}^2)$ nodes

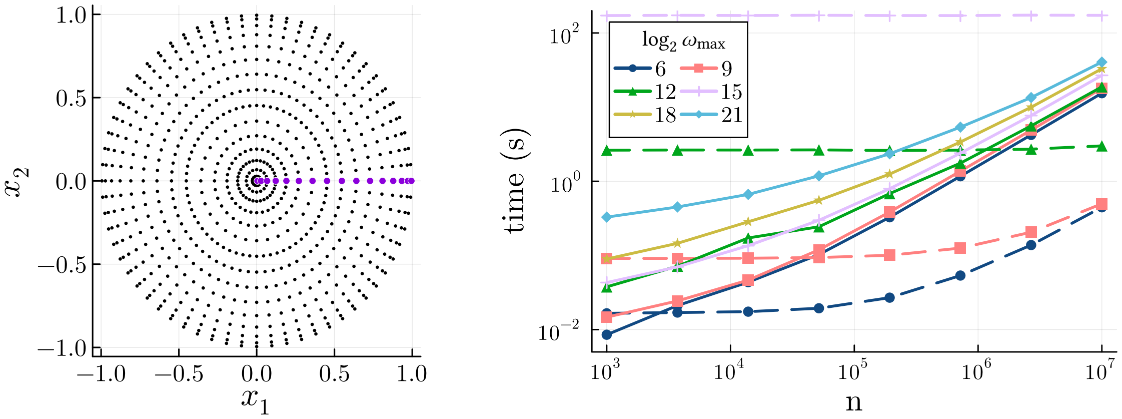


**Figure 3.6:** Example two-dimensional quadrature nodes for the NUFFT, with one-dimensional radial Gauss-Legendre quadrature on $[0, 1]$ for the NUFHT emphasized. Runtime comparison between NUFHT and 2D NUFFT for various choices of the maximum frequency $\omega_{\max}$ and the number of evaluation points $n$. Solid lines indicate the NUFHT, and the corresponding dashed lines indicate the 2D NUFFT.

are needed to resolve these oscillations. Therefore the $O(n)$ spreading step in the NUFFT becomes prohibitively expensive. However, by using radial symmetry to reduce to a one-dimensional integral, the NUFHT requires only $O(\omega_{\max})$ quadrature nodes, avoiding the curse of dimensionality. Figure 3.6 shows an example quadrature and runtimes for both the NUFFT and NUFHT approaches. Note that for $\omega_{\max} = 2^{15}$ the 2D NUFFT is orders of magnitude slower than the NUFHT for most $m$, and for even larger $\omega_{\max}$ the quadratic scaling of the 2D NUFFT with frequency makes the computation intractable on a laptop, while the NUFHT's linear scaling with frequency allows evaluation of the Fourier transform at significantly higher frequencies at an only moderately increased cost.

### 3.5.3 A Helmholtz solver using Fourier-Bessel expansions

Finally, we demonstrate the application of the nonuniform Hankel transform to solving partial differential equations on the disk using Fourier-Bessel expansions. Consider the following inhomogeneous Helmholtz problem on the unit disk $D$

$$\begin{aligned} (\Delta + \kappa^2)u(r,\theta) &= f(r,\theta), && \text{for } r \in [0,1), \quad \theta \in [0,2\pi), \\ u(1,\theta) &= 0 && \text{for } \theta \in [0,2\pi). \end{aligned} \tag{3.42}$$

Note that the functions $\psi_{j\ell}(r,\theta) := J_\ell(\xi_{\ell,j} r)e^{i\ell\theta}$ are the eigenfunctions of the Laplacian on the unit disk with homogeneous Dirichlet boundary condition, so that

$$\Delta\psi_{j\ell}(r,\theta) = \lambda_{j\ell}\psi_{j\ell}(r,\theta), \tag{3.43}$$

where $\lambda_{j\ell} = -\xi_{\ell,j}^2$ [18, 122]. Therefore, writing the forcing function $f$ and solution $u$ in terms of their respective Fourier-Bessel expansions

$$f(r,\theta) = \sum_{\ell=-\infty}^{\infty} \sum_{j=1}^{\infty} \alpha_{j\ell}\, J_\ell(\xi_{\ell,j} r)\, e^{i\ell\theta}, \qquad u(r,\theta) = \sum_{\ell=-\infty}^{\infty} \sum_{j=1}^{\infty} \beta_{j\ell}\, J_\ell(\xi_{\ell,j} r)\, e^{i\ell\theta} \tag{3.44}$$

decouples (3.42) into a system of diagonal equations resulting in an explicit formula for the coefficients $\beta_{j\ell}$:

$$\beta_{j\ell} = \frac{\alpha_{j\ell}}{\lambda_{j\ell} + \kappa^2}. \tag{3.45}$$

Due to the orthogonality of the Bessel functions $J_\ell$, the Fourier-Bessel coefficients of the forcing $f$ can be computed as:

$$\alpha_{j\ell} = \frac{2}{J_{\ell+1}(\xi_{\ell,j})^2} \int_0^{2\pi} \int_0^1 f(r,\theta)\, J_\ell(\xi_{\ell,j} r)\, e^{-i\ell\theta}\, r \,\mathrm{d}r \,\mathrm{d}\theta, \tag{3.46}$$

and the Fourier-Bessel expansion of the solution $u$ can then be written explicitly

$$u(r,\theta) = \sum_{\ell=-\infty}^{\infty} \sum_{j=1}^{\infty} \frac{\alpha_{j\ell}}{\lambda_{j\ell} + \kappa^2}\, J_\ell(\xi_{\ell,j} r)\, e^{i\ell\theta}. \tag{3.47}$$

By diagonalizing the Laplacian, this Fourier-Bessel solver thus provides a direct analogue in the Dirichlet disk setting of spectral methods on a periodic rectangle using bivariate Fourier expansions, and inherits many of the merits of spectral methods. First, if $f$ and all its derivatives go to zero at $r = 1$ and $f$ is smooth in the interior of $D$, then $\left|\alpha_{j\ell}\right| \to 0$ exponentially fast in both $j$ and $\ell$ [18]. In addition, solutions for arbitrary $\kappa$ can be evaluated without additional computations involving $f$, assuming that $\kappa^2$ is not itself a Dirichlet eigenvalue of the Laplacian on $D$.

To compute the Fourier-Bessel coefficients $\alpha_{j\ell}$ of $f$ using (3.46), we use an $m$-point Gauss-Legendre rule in $r$ and a $t$-point trapezoidal rule in $\theta$. We iteratively double the number of nodes in each rule until the relative norm difference in computed coefficients between iterations is less than $\varepsilon$ (controlling the discretization error) and the relative norm of the coefficients appended in the last iteration is less than $\varepsilon$ (controlling the truncation error). Computing all $\alpha_{j\ell}$ at each iteration requires $t$ NUFHTs of size $m$ and $m$ FFTs of size $t$, resulting in $O(tm\log m + mt\log t)$ total complexity. Figure 3.7 shows an example random forcing $f$, the magnitude of its Fourier-Bessel coefficients $\alpha_{j\ell}$, and the corresponding solution $u$ to the Helmholtz equation (3.42) compute to relative precision $\varepsilon = 10^{-8}$.

This approach does, however, have two main limitations. First is that the coefficients of $f$ decrease only algebraically in $j$ if $f$ has nonzero derivatives at $r = 1$. More precisely, if $\Delta^q f(r)|_{r=1} = 0$ for all $0 \leq q \leq p-1$, then $\left|\alpha_{j\ell}\right| \sim j^{-2p-\frac{1}{2}}$, with exponential convergence only possible if $\Delta^q f(r)|_{r=1} = 0$ for all integer $q$ [18]. This is a fundamental property of the Fourier-Bessel expansion, and does not depend on the numerical method used to evaluate the Hankel transform. The second limitation is the increase in computational cost of our NUFHT with the order $\ell$, as demonstrated in Figure 3.4. As $\alpha_{j\ell}$ decrease spectrally in $\ell$ for smooth functions $f$, very large $\ell$ are not often needed. However, as in any spectral method, functions with sharp features or discontinuous derivatives will yield only algebraic decay in $\ell$, requiring more Fourier bases. In such cases the corresponding high order NUFHTs become intractable using the method described here.

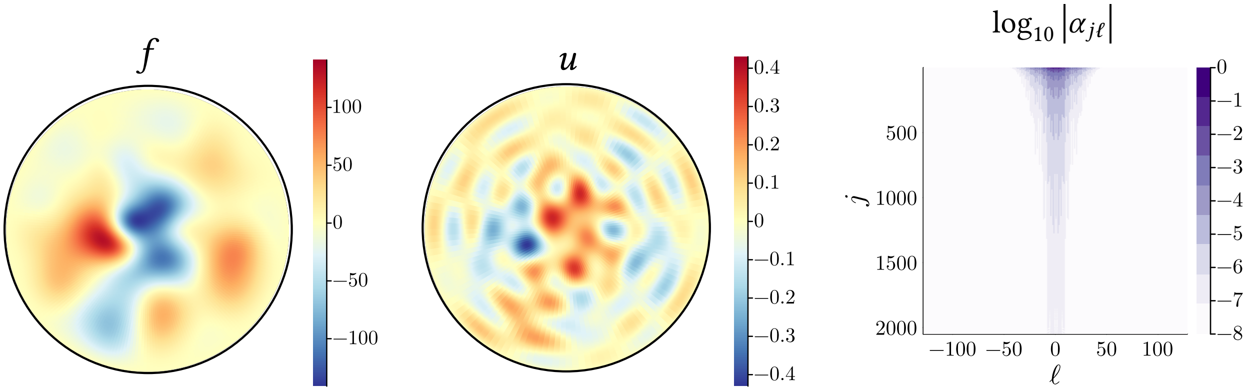


**Figure 3.7:** Forcing $f$, solution $u$, and log magnitude of Fourier-Bessel expansion coefficients $\alpha_{j\ell}$ for (3.42) with $\kappa = 25$.

## 3.6 Discussion

In this chapter we have presented a fast algorithm for computing discrete Hankel transforms of moderate orders from $n$ nonuniform points to $m$ nonuniform frequencies in $\mathcal{O}\big((m+n)\log\min(n,m)\big)$ operations. The algorithm relies on a careful space-frequency analysis of the Bessel function kernel, judicious use of small-argument series expansions and large-argument asymptotic expansions, as well as a small number of direct calculations. The algorithm makes no assumptions on the distribution of points in space and frequency — it applies to the fully nonuniform case — and can be used for Hankel transforms of higher order with a modest increase in computational cost. More importantly, the algorithm does not require any precomputation, in contrast to algorithms based on butterfly factorizations of the Hankel transform matrix. Significant speedups over the direct calculation have been demonstrated, as well as asymptotic scaling of the computational complexity. An implementation of the algorithm of this chapter is available as an open-source Julia package at `github.com/pbeckman/FastHankelTransform.jl`.

In order to efficiently extend our algorithm to compute arbitrarily high-order Hankel transforms which are needed for higher-order Fourier-Bessel expansions and in various high-dimensional statistical settings [79, 87], alternative expansions and asymptotics of $J_\nu$ need to be used or derived. This is the focus of ongoing research.

# 4 | A butterfly-accelerated manifold harmonic transform

## 4.1 Introduction

In this chapter, we pivot away from the entry evaluation schemes of Chapters 2 and 3, and develop a "Fourier feature"-type method for GRFs on compact manifolds. In order to motivate the transform that we will study in this chapter, consider the two-dimensional type-II nonuniform Fourier transform

$$g(\boldsymbol{x}_j) = \sum_{\boldsymbol{k} \in [-\frac{\sqrt{m}}{2}, \frac{\sqrt{m}}{2}-1]^2} f_{\boldsymbol{k}}\, e^{i\boldsymbol{k}^\top \boldsymbol{x}_j}, \quad \text{for } j = 1, \ldots, n \tag{4.1}$$

from $m$ uniform frequencies $\boldsymbol{k}$ to $n$ nonuniform spatial locations $\boldsymbol{x}_j \in [-\pi, \pi]^2$. Noting that $e^{i\boldsymbol{k}^\top \boldsymbol{x}}$ for $\boldsymbol{k} \in \mathbb{Z}^2$ are exactly the eigenfunctions of the Laplacian $\Delta$ on $[-\pi, \pi]^2$ under periodic boundary conditions, it is natural to consider analogous Fourier-like transforms which can be used in spectral methods and to perform harmonic analysis on domains other than the flat square.

For a general compact manifold $\mathcal{M}$ with or without a boundary $\partial\mathcal{M}$, consider the Laplace-

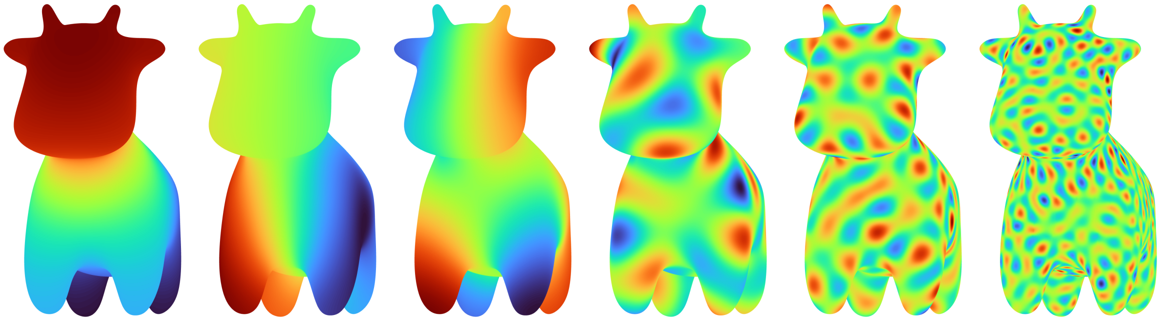


**Figure 4.1:** Eigenfunctions $\varphi_k$ of the Laplacian on a cow manifold for $k = 2, 4, 8, 78, 340, 1350$.

Beltrami eigenproblem

$$\begin{cases} \Delta_{\mathcal{M}}\varphi = \lambda\varphi, & \text{in } \mathcal{M}, \\ h\left(\varphi, \dfrac{\partial\varphi}{\partial n}\right) = 0 & \text{on } \partial\mathcal{M}, \end{cases} \tag{4.2}$$

where $h(\varphi, \frac{\partial\varphi}{\partial n})$ represents any suitable set of boundary conditions, e.g. Dirichlet, Neumann, Robin, or a mixture thereof. This eigenproblem yields a set of nonnegative eigenvalues $0 \leq \lambda_1 \leq \lambda_2 \leq \ldots$ whose corresponding eigenfunctions $\{\varphi_k\}_{k=1}^{\infty}$ form an orthonormal basis for $L^2(\mathcal{M})$ [23]. A generalization of the Fourier transform to the manifold $\mathcal{M}$ is then given by the *manifold harmonic transform* (MHT)

$$g(\boldsymbol{x}_j) = \sum_{k=1}^{m} f_k \varphi_k(\boldsymbol{x}_j), \quad \text{for } j = 1, \ldots, n, \tag{4.3}$$

which we write in the equivalent matrix form $\mathbf{g} = \mathbf{\Phi}\mathbf{f}$ where $\mathbf{\Phi} \in \mathbb{C}^{n\times m}, \mathbf{\Phi}_{jk} = \varphi_k(\boldsymbol{x}_j)$, and $\mathbf{g}_j = g(\boldsymbol{x}_j)$. Figure 4.1 shows numerically computed Laplace-Beltrami eigenfunctions $\varphi_k$ on an example manifold $\mathcal{M}$, each of which serves as a column of the matrix $\mathbf{\Phi}$. Note that as we increase the order $k$, the eigenfunctions exhibit oscillations on decreasing lengthscales, providing a natural multiscale Fourier-like basis on $\mathcal{M}$.

The MHT and related computations appear in applications to computer graphics and meshing [72, 95, 99, 104, 120]; surface partial differential equations (PDEs) [16, 35]; statistics [17, 70]; as well as data assimilation and kernel learning settings [8, 46]. A number of compressed representations for the lowest Laplace-Beltrami eigenfunctions have been proposed in the literature [10, 27, 85]. However, these works are focused mainly on scalably computing these eigenfunctions, and do not explicitly investigate the possibility of a fast transform for rapidly evaluating linear combinations of many eigenfunctions $\{\varphi_k\}_{k=1}^m$ at a large number of target points $\{\boldsymbol{x}_j\}_{j=1}^n \subset \mathcal{M}$.

For other oscillatory integral transforms such as the nonuniform Fourier, Hankel, and spherical harmonic transforms, one can leverage various symmetries and analytical expansions of the eigenfunctions $\varphi_k$ to develop fast algorithms [5, 34, 82, 117]. However, no such analytical structure is available a priori for the MHT on general manifolds $\mathcal{M}$. Instead, we must rely on purely linear-algebraic structure to compress the matrix $\mathbf{\Phi}$. One such compressed linear-algebraic representation of oscillatory operators is given by the *butterfly factorization* [75, 89], which we apply and analyze here in the MHT context. The butterfly factorization is a generalization of the hierarchical algebraic structure of the FFT algorithm — in which interactions between each level are exactly rank-one operations — to more general operators in which interactions between scales have small $\varepsilon$-rank. Butterfly factorizations have been successfully applied to compress Fourier integral operators [21, 74, 75], Hankel transforms [75, 89], and spherical harmonic transforms [103, 119], among others.

In this chapter, we introduce a discretization-agnostic butterfly-accelerated manifold harmonic transform (BA-MHT) which reduces the cost to store and apply the MHT on a 2-manifold $\mathcal{M} \subset \mathbb{R}^3$ from $\mathcal{O}(nm)$ to $\mathcal{O}(n + m^{3/2})$ after an $\mathcal{O}(nm)$ precomputation to evaluate and compress the relevant eigenfunctions.

Notation Throughout this chapter, we use Golub and Van Loan [48] i.e. MATLAB-style submatrix notation, so that given a matrix $\mathbf{\Phi} \in \mathbb{C}^{n \times m}$ and index sets $\gamma := [j_1, j_2, \ldots, j_p]$ and $\sigma :=$

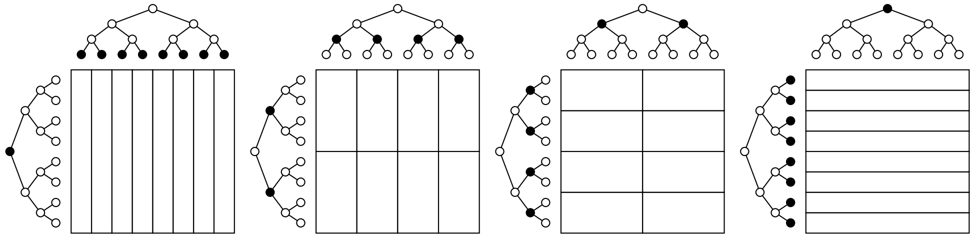


**Figure 4.2:** Row tree, column tree, and resulting complementary blocks of $\mathbf{\Phi}$ at levels $\ell = 0, \ldots, 3$.

$[k_1, k_2, \ldots, k_q]$, we denote the corresponding $p \times q$ submatrix by

$$\mathbf{\Phi}(\gamma, \sigma) = \begin{bmatrix} \mathbf{\Phi}(j_1, k_1) & \cdots & \mathbf{\Phi}(j_1, k_q) \\ \vdots & \ddots & \vdots \\ \mathbf{\Phi}(j_p, k_1) & \cdots & \mathbf{\Phi}(j_p, k_q) \end{bmatrix}. \tag{4.4}$$

We denote $\mathbf{\Phi}(\gamma, :) = \mathbf{\Phi}(\gamma, [1, \ldots, m])$ and $\mathbf{\Phi}(:, \sigma) = \mathbf{\Phi}([1, \ldots, n], \sigma)$. We use $|\gamma| = p$ to denote the size of an index set.

## 4.2 The butterfly factorization

Discretizing oscillatory operators such as the Fourier transform generally results in full-rank matrix representations $\mathbf{\Phi}$. Nevertheless, many such matrices exhibit a *complementary low-rank property*, so that all blocks $\mathbf{\Phi}(\gamma, \sigma)$ with the same number of entries $|\gamma|\cdot|\sigma|$ have constant or nearly-constant ranks. Butterfly algorithms result from building trees on the column and row indices and considering blocks of the matrix with the corresponding indices. One can then move down the row tree (halving the number of rows) and simultaneously move up the column tree (doubling the number of columns). The number of block entries at each level is thus held constant, and the complementary low-rank property can be exploited in a nested fashion on each level. Figure 4.2 provides a graphical representation of the row and column trees and the corresponding low-rank matrix blocks at each level.

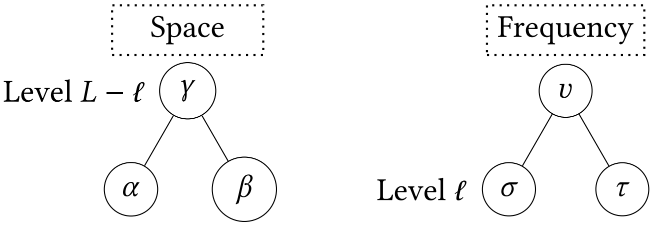
We now describe the butterfly factorization algorithm of [89]. To align with Fourier transform and MHT context of Section 4.1, we refer to the first dimension (the row index) as representing space, and the second dimension (the column index) as representing frequency. We assume binary trees $T_x$ and $T_\lambda$ have been constructed whose nodes are index sets on the rows and columns of $\mathbf{\Phi}$ respectively. The precise choice of trees will affect the level of resulting compression, but does not alter the algorithm, and this construction generalizes in a straightforward way to two or three dimensions using quadtrees or octrees.

To begin, at level $\ell = 0$ we compute a low-rank factorization of the block column

$$\mathbf{\Phi}(:,\sigma) = \mathbf{U}_{:\sigma}\mathbf{V}_{:\sigma}^{*} \tag{4.5}$$

for each leaf node $\sigma$ in the frequency tree. We note that the butterfly algorithm is agnostic to the particular type of low-rank factorization used here and throughout, and many are suitable, e.g. SVD, QR, interpolative decomposition, etc.

At levels $\ell = 1, \ldots, L-1$ we use the low-rank factors from the previous level to compute a low-rank factorization of each complementary block at the current level. Let $\gamma$ be a node at level $L-\ell$ of the space tree with children $\alpha, \beta$. Let $\sigma, \tau$ be nodes at level $\ell$ of the frequency tree with parent $\upsilon$.



Assume we have computed low-rank factorizations of $\mathbf{\Phi}(\gamma,\sigma)$ and $\mathbf{\Phi}(\gamma,\tau)$. We can then move down the space tree and up the frequency tree one level, writing $\mathbf{\Phi}(\alpha,\upsilon)$ in terms of the previously computed factors

$$\begin{aligned}\mathbf{\Phi}(\alpha, \upsilon) &= \begin{bmatrix}\mathbf{\Phi}(\gamma, \sigma) & \mathbf{\Phi}(\gamma, \tau)\end{bmatrix}(\alpha, :) && (4.6)\\ &= \begin{bmatrix}\mathbf{U}_{\gamma\sigma}\mathbf{V}_{\gamma\sigma}^{*} & \mathbf{U}_{\gamma\tau}\mathbf{V}_{\gamma\tau}^{*}\end{bmatrix}(\alpha, :) && (4.7)\\ &= \begin{bmatrix}\mathbf{U}_{\gamma\sigma}(\alpha, :) & \mathbf{U}_{\gamma\tau}(\alpha, :)\end{bmatrix}\begin{bmatrix}\mathbf{V}_{\gamma\sigma}^{*} & \\ & \mathbf{V}_{\gamma\tau}^{*}\end{bmatrix}. && (4.8)\end{aligned}$$

We then compute a low-rank factorization of the first matrix above

$$\begin{bmatrix}\mathbf{U}_{\gamma\sigma}(\alpha, :) & \mathbf{U}_{\gamma\tau}(\alpha, :)\end{bmatrix} = \mathbf{U}_{\alpha\upsilon}\mathbf{R}_{\alpha\upsilon}. \tag{4.9}$$

This yields a low-rank factorization of the desired matrix block

$$\mathbf{\Phi}(\alpha, \upsilon) = \mathbf{U}_{\alpha\upsilon}\mathbf{R}_{\alpha\upsilon}\begin{bmatrix}\mathbf{V}_{\gamma\sigma}^{*} & \\ & \mathbf{V}_{\gamma\tau}^{*}\end{bmatrix} = \mathbf{U}_{\alpha\upsilon}\mathbf{V}_{\alpha\upsilon}^{*} \tag{4.10}$$

where $\mathbf{R}_{\alpha\upsilon}$ is the *transfer matrix* which constructs the row basis $\mathbf{V}_{\alpha\upsilon}$ for the current level implicitly using the row bases $\mathbf{V}_{\gamma\sigma}, \mathbf{V}_{\gamma\tau}$ from the previous level. An analogous process is carried out for $\mathbf{\Phi}(\beta, \upsilon)$, after which we can discard the column bases $\mathbf{U}_{\gamma\sigma}$ and $\mathbf{U}_{\gamma\tau}$ from the last level. We repeat this computation for each block at each level $\ell = 1, \ldots, L$.

Let $r_{\varepsilon}(\gamma, \sigma)$ denote the $\varepsilon$-rank of the block $\mathbf{\Phi}(\gamma, \sigma)$. Then each transfer matrix is of size $r_{\varepsilon}(\alpha, \upsilon) \times \big(r_{\varepsilon}(\gamma, \sigma) + r_{\varepsilon}(\gamma, \tau)\big)$. If all complementary blocks have $\varepsilon$-rank exactly by $r$, then all transfer matrices are $r \times 2r$. For a matrix $\mathbf{\Phi} \in \mathbb{C}^{n \times m}$ with $m \leq n$, we store $O(m)$ transfer matrices at each of the $O(\log m)$ levels, along with column basis matrices at level $L$. This yields a $O\big(rn + r^2 m \log m\big)$ storage and application cost. However, depending on the rank structure of $\mathbf{\Phi}$, the ranks of complementary blocks may grow with $n$ or $m$, and as a result, the storage and application cost may exceed the quasilinear cost of the fixed-rank case. This is true for the MHT, as we will see in Section 4.5.

We turn now to the cost of the necessary precomputations for the BA-MHT. The butterfly factorization cost is dominated by the first level, where we must compute low-rank factorizations of $O(m)$ matrices, each of size $n \times O(1)$. This leads to an $O(nm)$ factorization cost. A number of approaches have been proposed to reduce the cost of this precomputation, including linear cost interpolative decompositions [26, 36, 94] and residual phase functions [21, 31]. Unfortunately, these approaches rely on difficult-to-verify incoherence properties of the submatrices of $\mathbf{\Phi}$, require the physical locations $\{\boldsymbol{x}_j\}_{j=1}^n$ to lie on specific tensor product grids, or necessitate knowledge of a phase function $\psi$ for which $K(\boldsymbol{\omega}, \boldsymbol{x}) \approx e^{i\psi(\boldsymbol{\omega},\boldsymbol{x})}$. Thus none of these techniques are directly applicable to the BA-MHT. Furthermore, computing $m$ Laplace-Beltrami eigenfunctions generally requires at least $O(nm)$ operations, and is thus the bottleneck relative to the factorization step in practice. Therefore, we do not make any attempt here to accelerate the butterfly factorization step in the MHT.

**Remark 4.1.** By arranging the transfer matrices $\mathbf{R}_{\alpha v}$ at each level into a matrix, the butterfly factorization can be equivalently written as a product of $O(\log n)$ sparse matrices each with $O(r^2 n)$ entries. This can be a useful viewpoint for various computational tasks. However, we find the sparsity structure of these factor matrices to be less intuitive for our purposes, and thus focus on the tree-based construction given above.

## 4.3 Fiedler trees

In order to compute a butterfly factorization of the MHT, we must first construct a row tree on the spatial domain $\mathcal{M}$ and a column tree on the frequency domain $[0, \lambda_m]$. Forming a binary tree on the frequency interval is straightforward. However, building a suitable tree on the manifold $\mathcal{M}$ is more subtle. In some simple cases like the sphere, torus, or deformations thereof, one can use an underlying two-dimensional parameterization of the surface to build a tree whose elements

**Algorithm 3:** Butterfly factorization

```
ButterflyFactorization($\mathbf{\Phi}$, $T_x$, $T_\lambda$):
    for leaf node $\sigma$ in $T_\lambda$ do
        $\mathbf{U}_{:\sigma}\mathbf{V}_{:\sigma}^{*} = \mathbf{\Phi}(:, \sigma)$
    end
    for level $\ell = 1, \ldots, L-1$ do
        for node $\sigma$ at level $\ell$ of $T_\lambda$ do
            Let $\sigma$ and $\tau$ be the children of $v$
            for node $\gamma$ at level $L-\ell$ of $T_x$ do
                Let $\alpha$ and $\beta$ be the children of $\gamma$
                $\begin{bmatrix}\mathbf{U}_{\gamma\sigma}(\alpha,:) & \mathbf{U}_{\gamma\tau}(\alpha,:)\end{bmatrix} = \mathbf{U}_{\alpha v}\mathbf{R}_{\alpha v}$
                $\begin{bmatrix}\mathbf{U}_{\gamma\sigma}(\beta,:) & \mathbf{U}_{\gamma\tau}(\beta,:)\end{bmatrix} = \mathbf{U}_{\beta v}\mathbf{R}_{\beta v}$
                Free memory associated with $\mathbf{U}_{\gamma\sigma}$ and $\mathbf{U}_{\gamma\tau}$
            end
        end
    end
    return $\left\{\begin{matrix}\textit{Row basis } \mathbf{V}_{:\sigma} \textit{ for each leaf node } \sigma \textit{ in } T_\lambda \\ \textit{Transfer matrix } \mathbf{R}_{\alpha v} \textit{ for each block at levels } \ell = 1, \ldots, L-1 \\ \textit{Column basis } \mathbf{U}_{\alpha:} \textit{ for each leaf node } \alpha \textit{ in } T_x\end{matrix}\right\}$
```

at each level have approximately equal areas.

For general surfaces, one simple approach is to subdivide $\mathcal{M}$ according to an octree in the ambient space. For simple convex domains this approach may be sufficient in practice. However, it often results in octree boxes containing submanifolds of $\mathcal{M}$ which are close in the extrinsic Euclidean metric but may be arbitrarily far apart in geodesic distance. As a result, eigenfunctions $\varphi_k$ of similar orders $k$ within a given octree box may have highly dissimilar structures, reducing compressibility and causing the block ranks within the butterfly factorization to grow unnecessarily quickly.

To avoid such issues, we instead use a *Fiedler tree* [11, 115], which respects the intrinsic geometry of $\mathcal{M}$ relevant to the MHT. The construction of a Fiedler tree takes advantage of two essential observations. The first is that by the Courant Nodal Domain Theorem [23], the second eigenfunction $\varphi_2$ of the Laplace-Beltrami operator divides $\mathcal{M}$ into exactly two nodal domains —

one on which $\varphi_2 \leq 0$ and one on which $\varphi_2 > 0$. If $\mathcal{M}$ is a manifold with boundary, then this fact holds for any suitable choice of boundary conditions, e.g. Dirichlet, Neumann, or mixed. The second observation is that partitioning a manifold according to the value of $\varphi_2$ provides an approximation to the cut of $\mathcal{M}$ into submanifolds which minimizes the ratio of the length of the cut to the volume of each of the resulting submanifolds [24, 37, 105]. In other words, $\varphi_2$ divides $\mathcal{M}$ into two pieces with approximately equal surface areas and little connection between the pieces. We then apply this reasoning recursively; we compute $\varphi_2$ on the original manifold, divide it into two submanifolds according to the computed values of $\varphi_2$, then compute a new eigenfunction $\varphi_2$ on each of the submanifolds, divide each one into two new submanfiolds using the values of their respective eigenfunctions $\varphi_2$, and so on. Repeating this partitioning process until $\mathcal{O}(1)$ points $\boldsymbol{x}_j$ lie in each submanifold yields a Fiedler tree on $\mathcal{M}$.

Several technical choices must now be made. First, we must choose a boundary condition to enforce in the eigenproblem on each submanifold. For this, we use a homogeneous Neumann condition and solve

$$\begin{cases} \Delta_{\mathcal{M}}\varphi = \lambda\varphi, & \text{in } \mathcal{M}, \\ \dfrac{\partial\varphi}{\partial n} = 0 & \text{on } \partial\mathcal{M}. \end{cases} \tag{4.11}$$

Other choices of boundary condition are possible and will result in slightly different trees. Next, we must determine how to partition $\mathcal{M}$ given $\varphi_2$. [11] uses a linear approximation to the level set $\varphi_2 = 0$ on each triangular patch $P$, introducing new cut cells into a linear finite element discretization of $\mathcal{M}$. However, because we require only a rough partitioning of $\mathcal{M}$ and wish to remain agnostic to the surface discretization method, we forgo cut cells and instead subdivide $\mathcal{M}$ according to the sign of the integral $\int_P \varphi_2(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}$ over each element $P$. This approach is simple to implement, and can be used regardless of whether the elements $P$ are a low- or high-order, triangular or quadrilateral. If $\mathcal{M}$ is instead defined in an implicit or *mesh-free* way as the collection of points $\{\boldsymbol{x}_j\}_{j=1}^n$, and the eigenfunctions are computed using a kernel-based approach as in the

**Algorithm 4:** Construct Fiedler tree

```
FiedlerTree($\mathcal{M}$, $\{\boldsymbol{x}_j\}_{j=1}^n$):
    if $n <$ min_size then
        return $\{\boldsymbol{x}_j\}_{j=1}^n$
    Compute second eigenfunction $\varphi_2$ of $\Delta_{\mathcal{M}}$
    if Discretization of $\mathcal{M}$ consists of patches $P$ then
        $\mathcal{M}_1 := \left\{P \in \mathcal{M} \,:\, \int_P \varphi_2(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} \leq 0\right\}$
    else if Representation of $\mathcal{M}$ is mesh-free then
        $\mathcal{M}_1 := \left\{1 \leq j \leq n \,:\, \varphi_2(\boldsymbol{x}_j) \leq 0\right\}$
    $\mathcal{M}_2 := \mathcal{M} \setminus \mathcal{M}_1$
    $X_1 := \left\{\boldsymbol{x}_j \text{ for } j = 1, \ldots, n \,:\, \boldsymbol{x}_j \in \mathcal{M}_1\right\}$
    $X_2 := \{\boldsymbol{x}_j\}_{j=1}^n \setminus X_1$
    return $\left\{\text{FiedlerTree}(\mathcal{M}_1, X_1), \text{FiedlerTree}(\mathcal{M}_2, X_2)\right\}$
```

"Laplacian eigenmaps" paradigm [7, 8, 73], then one can simply partition $\mathcal{M}$ by the sign of $\varphi_2(\boldsymbol{x}_j)$ at each point. Algorithm 4 gives pseudocode for the recursive construction of a Fiedler tree, and Figure 4.3 illustrates the partitioning process on an example high-order quadrilateral mesh, plotting the eigenfunctions $\varphi_2$ used on various submanifolds at each level.

## 4.4 The butterfly-accelerated manifold harmonic transform

Having developed a tree structure on $\mathcal{M}$, we now have all the necessary tools to apply the butterfly factorization to the MHT. However, we note that applying Algorithm 3 directly requires that we first compute column bases $\mathbf{U}_{:\sigma}$ for every leaf node $\sigma$ in $T_\lambda$. At this point in the algorithm, the necessary working memory is $O(nm)$ in general. As our objective is to compute a compressed representation of $\boldsymbol{\Phi}$ when the dense matrix may not fit in working memory, this step may be infeasible.

In order to reduce the memory requirements of the algorithm so that the necessary memory

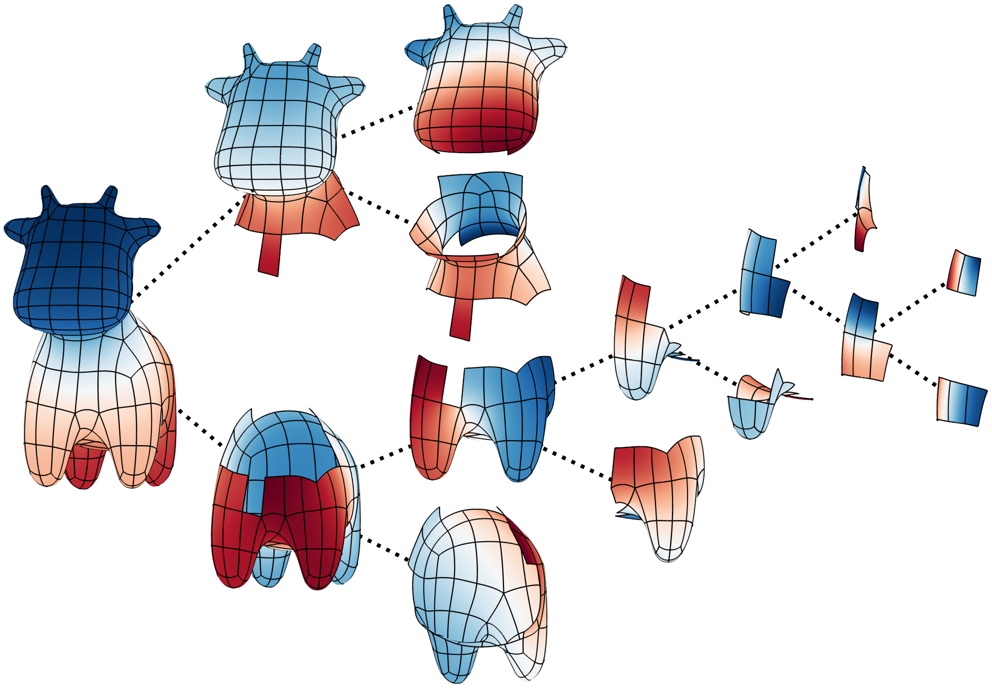


**Figure 4.3:** A subset of the nodes of a cut-cell-free Neumann Fiedler tree on the cow mesh, with the second Neumann eigenfunction $\varphi_2$ plotted on each submanifold.

is bounded at all times by the size of the final compressed representation up to a small constant factor, we reorganize the factorization process using a post-order traversal of the nodes in $T_\lambda$ as suggested in [119]. The post-order traversal begins at the left-most leaf node and proceeds in a "left, right, parent" fashion. We can thus compute a small number of Laplace-Beltrami eigenfunctions $\varphi_k$ corresponding to a leaf node in the frequency tree $T_\lambda$, and compress this block of eigenfunctions as much as possible before computing the next set of columns. Algorithm 5 and Figure 4.4 provide pseudocode and a graphical representation for this streaming BA-MHT algorithm.

## 4.5 Rank and complexity analysis

We now establish asymptotic storage and complexity results for our algorithm to illuminate various aspects of the BA-MHT's performance which will be observed in numerical examples

**Algorithm 5:** Streaming butterfly factorization

```
StreamingButterflyFactorization($k \to \boldsymbol{\varphi}_k$, $T_x$, $T_\lambda$):
    for node $v$ in a post-order traversal of $T_\lambda$ do
        if $v$ is a leaf node then
            Compute $\mathbf{\Phi}(:, v)$
            $\mathbf{U}_{:v}\mathbf{V}_{:v}^* = \mathbf{\Phi}(:, v)$
        else if $v$ is a node at level $\ell - 1$ then
            Let $\sigma$ and $\tau$ be the children of $v$
            for node $\gamma$ at level $L - \ell$ of $T_x$ do
                Let $\alpha$ and $\beta$ be the children of $\gamma$
                $\begin{bmatrix}\mathbf{U}_{\gamma\sigma}(\alpha, :) & \mathbf{U}_{\gamma\tau}(\alpha, :)\end{bmatrix} = \mathbf{U}_{\alpha v}\mathbf{R}_{\alpha v}$
                $\begin{bmatrix}\mathbf{U}_{\gamma\sigma}(\beta, :) & \mathbf{U}_{\gamma\tau}(\beta, :)\end{bmatrix} = \mathbf{U}_{\beta v}\mathbf{R}_{\beta v}$
                Free memory associated with $\mathbf{U}_{\gamma\sigma}$ and $\mathbf{U}_{\gamma\tau}$
            end
    end
    return $\left\{\begin{array}{l}\textit{Row basis } \mathbf{V}_{:\sigma} \textit{ for each leaf node } \sigma \textit{ in } T_\lambda \\ \textit{Transfer matrix } \mathbf{R}_{\alpha v} \textit{ for each block at levels } \ell = 1, \ldots, L-1 \\ \textit{Column basis } \mathbf{U}_{\alpha:} \textit{ for each leaf node } \alpha \textit{ in } T_x\end{array}\right\}$
```

and applications in the following sections. Analyzing the rank of blocks of the manifold harmonic transform in the general case is challenging because the eigenfunctions and eigenvalues are not known in closed form. However, for the flat torus $\mathcal{M} = [-\pi, \pi]^2$, the manifold harmonic transform is exactly the two-dimensional Fourier transform with eigenvalues $\lambda_k = k_1^2 + k_2^2$ and eigenfunctions $\varphi_k(\boldsymbol{x}) = e^{i(k_1 x_1 + k_2 x_2)}$ for $k_1, k_2 \in \mathbb{Z}$. Therefore, we develop here explicit bounds on the $\varepsilon$-rank of blocks of the two-dimensional Fourier transform. We will see that these results agree with numerical experiments on other manifolds, and we will thus conjecture that they extend to the general case.

To begin, we require a simple bound which will be used extensively in the proofs that follow.

**Lemma 4.1.** *Take $k \in \mathbb{Z}$ with $k > 0$ and $x \in \mathbb{R}$ with $x > 0$. Then we have*

$$|J_k(x)| \le \frac{(x/2)^k}{k!} \le e^{\xi}\left(\frac{x}{2\xi}\right)^k. \tag{4.12}$$

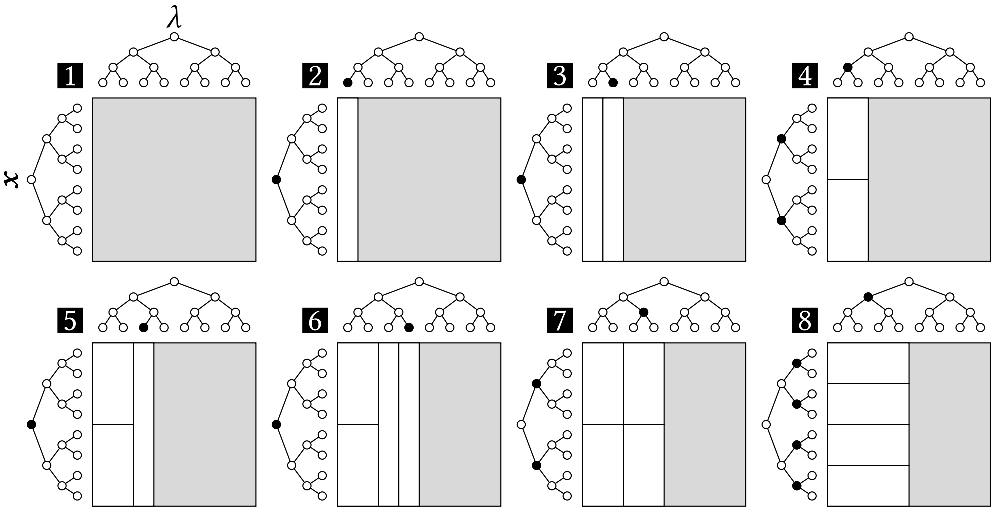


**Figure 4.4:** A schematic depiction of the streaming butterfly factorization. The nodes of the column (frequency) tree are traversed using a post-order traversal. When leaf nodes are visited, new bands of eigenvectors are streamed. When an internal node is visited, the partial factors corresponding to that node's children are merged and split. Eight steps of the algorithm are shown in sequence, with the block boxes indicating the order of the steps. Note that the nodes in the row and column trees being visited at each step are marked in the corresponding trees.

*For any $\xi > 0$. In particular, taking $\xi = \frac{ex}{2}$ gives*

$$|J_k(x)| \le e^{\frac{ex}{2}-k}. \tag{4.13}$$

*Proof.* The first inequality is [90, p. 10.14.4], and Stirling's formula gives the second. □

Consider the Fourier kernel $K(\boldsymbol{\omega}, \boldsymbol{x}) = e^{i\boldsymbol{\omega}^\top \boldsymbol{x}}$ in two dimensions. Given access to the two-dimensional frequencies $\boldsymbol{\omega}$, one could build a butterfly factorization for the two-dimensional Fourier transform using quadtrees in space and frequency. The following theorem demonstrates that the rank of the two-dimensional Fourier kernel from frequencies $\boldsymbol{\omega} \in B_b$ to points $\boldsymbol{x} \in B_R$ is of size $r_\varepsilon = O(b^2 R^2)$, nd thus the $\varepsilon$-ranks of all compressed blocks remain bounded by a fixed $r$ at all levels, resulting in an $O(r^2 n \log n)$ algorithm.

**Theorem 4.2.** *The $\varepsilon$-rank of the Fourier kernel $K(\boldsymbol{\omega}, \boldsymbol{x}) = e^{i\boldsymbol{\omega}^\top \boldsymbol{x}}$ for $\boldsymbol{\omega} \in B_b$ and $\boldsymbol{x} \in B_R$ is bounded* *by*

$$r_\varepsilon \le \frac{1}{2}\left\lceil ebR + \log 2 + \log(\varepsilon^{-1})\right\rceil^2. \tag{4.14}$$

*Proof.* Writing $\boldsymbol{\omega} \mapsto (\rho, \alpha)$ and $\boldsymbol{x} \mapsto (r, \theta)$ in polar coordinates gives

$$k(\boldsymbol{\omega}, \boldsymbol{x}) := e^{i\rho r \cos(\alpha - \theta)}. \tag{4.15}$$

Using [88, Lemma 3.2], we obtain

$$k(\boldsymbol{\omega}, \boldsymbol{x}) = \sum_{\ell=0}^{\infty} c_\ell J_\ell(\rho r)\Big(\cos(\ell\alpha)\cos(\ell\theta) + \sin(\ell\alpha)\sin(\ell\theta)\Big) \tag{4.16}$$

where $c_0 := 1$ and $c_\ell := 2i^\ell$ for all $\ell > 0$. Applying [90, p. 10.23.2] then gives the expansion

$$k(\boldsymbol{\omega}, \boldsymbol{x}) = \sum_{\ell=0}^{\infty} c_\ell\Big(\cos(\ell\alpha)\cos(\ell\theta) + \sin(\ell\alpha)\sin(\ell\theta)\Big)\rho^\ell \sum_{k=0}^{\infty} \frac{(-1)^k(\rho^2-1)^k(\frac{r}{2})^k}{k!} J_{\ell+k}(r), \tag{4.17}$$

which is separable into products of terms involving $(r, \theta)$ and $(\rho, \alpha)$ respectively. Assume without loss of generality that $\rho > 1$. Then using the fact that $|J_k(x)| \le \frac{1}{k!}(\frac{x}{2})^k$, we have

$$\left| c_\ell \cos(\ell\alpha)\cos(\ell\theta)\rho^\ell \frac{(-1)^k(\rho^2-1)^k(\frac{x}{2})^k}{k!} J_{\ell+k}(r)\right| \le \frac{2}{k!}\rho^\ell(\rho^2-1)^k\left(\frac{r}{2}\right)^k |J_{\ell+k}(r)|\,. \tag{4.18}$$

$$\le \frac{2}{(\ell+k)!k!}\left(\frac{\rho r}{2}\right)^{\ell+2k}, \tag{4.19}$$

with a similar bound for the sine term. Applying Lemma 4.1 and the fact that $\rho \le b$, we have

$$\frac{2}{(\ell+k)!k!}\left(\frac{\rho r}{2}\right)^{\ell+2k} \le 2e^{ebR-\ell-2k}. \tag{4.20}$$

Assuring this expression is bounded by $\varepsilon$, we obtain

$$\ell + 2k < ebR + \log 2 + \log(\varepsilon^{-1}). \tag{4.21}$$

Counting the number of integer pairs $(k, \ell)$ satisfying the above, and taking into account that each $\ell$ contributes two terms in (4.17) — one with cosines and one with sines — gives the result. □

However, in the case of the general MHT, no such two-dimensional frequency $\boldsymbol{\omega}$ is well-defined. Instead, we have access only to the linear ordering of eigenvalues $\lambda_k$ of the Laplacian. We now consider the two-dimensional Fourier transform as special case of the MHT, and study the rank of the Fourier kernel from eigenvalues with $\sqrt{\lambda} := \|\omega\| \in [a, b]$ to points $\boldsymbol{x} \in B_R$. This corresponds to considering frequencies in the annulus $\boldsymbol{\omega} \in A_a^b := \{\boldsymbol{\omega} \ : \ a \leq \|\boldsymbol{\omega}\| \leq b\}$.

One-dimensional intuition and the result of Theorem 4.2 that the $\varepsilon$-rank of the Fourier kernel from $\boldsymbol{\omega} \in B_b$ to $\boldsymbol{x} \in B_R$ scales like $r_\varepsilon = \mathcal{O}(\mathrm{vol}(B_b)\,\mathrm{vol}(B_R)) = \mathcal{O}(b^2R^2)$ both suggest that for $\boldsymbol{\omega} \in A_a^b$, we should have $r_\varepsilon = \mathcal{O}(\mathrm{vol}(A_a^b)\,\mathrm{vol}(B_R)) = \mathcal{O}((b^2 - a^2)R^2)$. In other words, that the rank of the Fourier kernel should depend only on the *phase space volume*, i.e. the product of the volumes of the physical and frequency domains. However, this turns out to be far from the case, which we will see is the driving factor in the computational complexity of the MHT.

Consider first the case of $\|\boldsymbol{\omega}\| = b$. Then the Jacobi-Anger expansion gives the Fourier coefficients of the Fourier kernel explicitly

$$K(\boldsymbol{\omega}, \boldsymbol{x}) = e^{ibr\cos(\alpha-\theta)} = \sum_{k=-\infty}^{\infty} i^k J_k(br) e^{ik(\alpha-\theta)}. \tag{4.22}$$

As $|J_k(bR)|$ is $\mathcal{O}(1)$ for $k < bR$ and the Fourier bases are orthogonal in $L^2([0, 2\pi])$, one cannot generally hope to obtain an expansion of $K$ with $\varepsilon$-rank less than $bR$ which is valid for $\boldsymbol{\omega}$ in any domain containing the circle of radius $b$. In particular, this implies that the $\varepsilon$-rank of the Fourier kernel between an annulus $\boldsymbol{\omega} \in A_a^b$ and a disk $\boldsymbol{x} \in B_R$ must grow at least as fast as the outer radius $b$ of the annulus, *even* if the phase space volume is held constant.

Before providing a bound on the rank of the annulus-to-disk Fourier kernel, we derive an explicit formula for the Chebyshev expansion coefficients of the Bessel function $J_k(\rho r)$ for $\rho$ in any interval $[a, b]$ in terms of sums of products of Bessel functions. This expansion is central to the proof that follows, and is of independent interest in other applications which require a local approximation of the Bessel function $J_k$.

**Lemma 4.3.** *Take $k \in \mathbb{Z}$ with $k \geq 0$. Then for all $r > 0$ and $a \leq \rho \leq b$,*

$$\begin{aligned} J_k(\rho r) &= \frac{c_{k0}(r)}{2} + \sum_{\ell=1}^{\infty} c_{k\ell}(r) T_\ell\left(\tfrac{2}{b-a}(\rho - a) - 1\right) && (4.23) \\ c_{k\ell}(r) &= \sum_{q=0}^{\infty} \begin{Bmatrix} c_{k\ell}^{2q}(r) & \ell \text{ even} \\ c_{k\ell}^{2q+1}(r) & \ell \text{ odd} \end{Bmatrix} \\ c_{k\ell}^{p}(r) &= 2\eta_p J_{\frac{p+\ell}{2}}\left(\tfrac{(b-a)r}{4}\right) J_{\frac{p-\ell}{2}}\left(\tfrac{(b-a)r}{4}\right)\left[J_{k-p}\left(\tfrac{(b+a)r}{2}\right) + (-1)^p J_{k+p}\left(\tfrac{(b+a)r}{2}\right)\right] \end{aligned}$$

*where $\eta_0 := \frac{1}{2}$ and $\eta_p := 1$ for all $p \neq 0$.*

*Proof.* Let $\rho = \frac{(b-a)}{2}(\cos\tau + 1) + a$. The Chebyshev coefficients are then given by

$$\begin{aligned} c_{k\ell} &= \frac{2}{\pi}\int_0^\pi J_k\left(\tfrac{(b-a)r}{2}\cos\tau + \tfrac{(b+a)r}{2}\right)\cos(\ell\tau)\,d\tau && (4.24) \\ &= \frac{2}{\pi}\int_0^\pi \left[\sum_{p=-\infty}^{\infty} J_p\left(\tfrac{(b-a)r}{2}\cos\tau\right) J_{k-p}\left(\tfrac{(b+a)r}{2}\right)\right]\cos(\ell\tau)\,d\tau && (4.25) \end{aligned}$$

where we have applied the Neumann addition formula [90, p. 10.23.2]. Exchanging the order of integration and summation, we obtain integrals of the form

$$\int_0^\pi J_p\left(\tfrac{(b-a)r}{2}\cos\tau\right)\cos(\ell\tau)\,d\tau = \begin{cases} \pi J_{\frac{p+\ell}{2}}\left(\tfrac{(b-a)r}{4}\right) J_{\frac{p-\ell}{2}}\left(\tfrac{(b-a)r}{4}\right) & p+\ell \text{ is even} \\ 0 & \text{otherwise.} \end{cases} \qquad (4.26)$$

The above equality follows by noting that the integrand is symmetric about $\frac{\pi}{2}$ when $p+\ell$ is even, in which case we apply [90, p. 10.22.13]. If $p+\ell$ is odd, the integrand is antisymmetric about $\frac{\pi}{2}$ and thus the integral is zero. Plugging this in to (4.25) gives the result. □

We can now use this formula for the Chebyshev coefficients of the Bessel function to bound the rank of the Bessel kernel $K_k(\rho, r) = J_k(\rho r)$.

**Lemma 4.4.** *Take $k \in \mathbb{Z}$. The $\varepsilon$-rank of $K_k(\rho, r) := J_k(\rho r)$ for $a \leq \rho \leq b$ and $0 \leq r \leq R$ is bounded by*

$$r_\varepsilon \leq \left\lceil \frac{2\xi + 2\log 4 + \log(\varepsilon^{-2})}{\log\left(\frac{8\xi}{(b-a)R}\right)} \right\rceil \tag{4.27}$$

*for any $\xi > \frac{(b-a)R}{8}$.*

*Proof.* Assume $\ell$ is even. Applying Lemma 4.3, bounding the absolute values of the Bessel functions of orders $\frac{p-\ell}{2}$, $k-p$, and $k+p$ by one, and applying Lemma 4.1 to the remaining term yields the bound

$$|c_{k\ell}(r)| \leq 4\sum_{q=0}^{\infty}\left|J_{q+\frac{\ell}{2}}\left(\frac{(b-a)r}{4}\right)\right| \tag{4.28}$$

$$\leq 4\sum_{q=0}^{\infty} e^{\xi}\left(\frac{(b-a)r}{8\xi}\right)^{q+\frac{\ell}{2}} \tag{4.29}$$

$$= \frac{4e^{\xi}\left(\frac{(b-a)r}{8\xi}\right)^{\frac{\ell}{2}}}{1-\frac{(b-a)r}{8\xi}}, \tag{4.30}$$

which is less than $\varepsilon$ for all

$$\frac{\ell}{2} > \frac{\xi - \log\left(1-\frac{(b-a)r}{8\xi}\right) + \log 4 + \log(\varepsilon^{-1})}{\log\left(\frac{8\xi}{(b-a)r}\right)} \tag{4.31}$$

for all $\xi > \frac{(b-a)r}{8}$. Neglecting the second term in the numerator gives the result. $\square$

We are ready to state and prove our desired rank bound on the annulus-to-disk Fourier transform. In essence, we show that the Fourier kernel mapping frequencies $\boldsymbol{\omega} \in A_a^b$ to points $\boldsymbol{x} \in B_R$ has $\varepsilon$-rank $\tilde{O}(bR)$ up to log factors in $(b-a)R$ if this quantity is sufficiently small.

**Theorem 4.5.** *The $\varepsilon$-rank of the Fourier kernel $K(\boldsymbol{\omega}, \boldsymbol{x}) := e^{i\boldsymbol{\omega}^\top \boldsymbol{x}}$ for $a \leq \|\boldsymbol{\omega}\| \leq b$ and $\boldsymbol{x} \in B_R$ is bounded by*

$$r_\varepsilon \leq \left\lceil \frac{5 + \log(\varepsilon^{-2})}{\log\left(\frac{8}{(b-a)R}\right)} \right\rceil \cdot \left\lceil \frac{e}{2} bR + \log(\varepsilon^{-1}) \right\rceil \tag{4.32}$$

*for any $(b-a)R < 8$.*

*Proof.* We first expand $K$ in a Fourier series in $\alpha - \theta$, giving

$$K(\boldsymbol{\omega}, \boldsymbol{x}) = e^{i\rho r\cos(\alpha-\theta)} = \sum_{k=-\infty}^{\infty} i^k J_k(\rho r) e^{ik(\alpha-\theta)}. \tag{4.33}$$

Applying Lemma 4.3 then yields

$$K(\boldsymbol{\omega}, \boldsymbol{x}) = \sum_{k=-\infty}^{\infty} \sum_{\ell=0}^{\infty} i^k c_{k\ell}(r) T_\ell\left(\tfrac{2}{b-a}(\rho - a) - 1\right) e^{ik(\alpha-\theta)}. \tag{4.34}$$

Applying Lemma 4.1 to $J_k(\rho r)$ gives

$$|c_{k\ell}(r)| \leq |J_k(\rho r)| \leq e^{\frac{e\rho r}{2} - k}, \tag{4.35}$$

and thus $|c_{k\ell}(r)| < \varepsilon$ for all

$$k > \frac{e\rho r}{2} + \log(\varepsilon^{-1}). \tag{4.36}$$

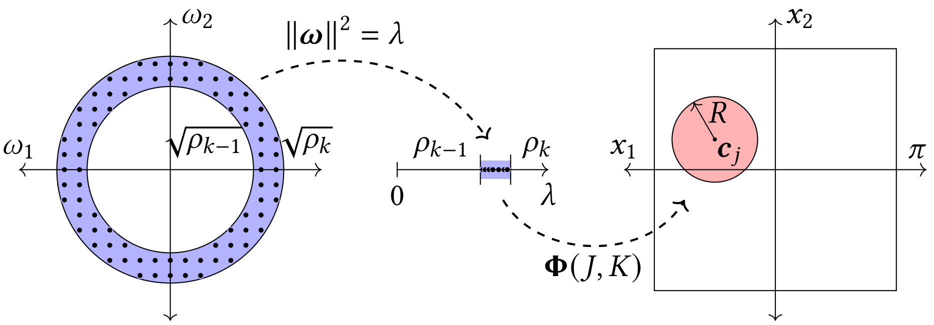


**Figure 4.5:** Diagram of the two-dimensional Fourier transform as a MHT. The annulus $A_{\sqrt{\rho_{k-1}}}^{\sqrt{\rho_k}}$ (left) contains eigenfrequencies $\boldsymbol{\omega} = [k_1, k_2]$ for $k_1, k_2 \in \mathbb{Z}$, denoted by black dots. These eigenfrequencies are mapped by $\|\boldsymbol{\omega}\|^2 = \lambda$ to the corresponding eigenvalues $\lambda_k$ of $\Delta_{[-\pi,\pi]^2}$ (center). Each block $\boldsymbol{\Phi}(J, K)$ of the MHT maps coefficients of eigenfunctions with $\lambda \in [\rho_{k-1}, \rho_k]$ to values in a disk $\boldsymbol{x} \in B_R(\boldsymbol{c}_j)$.

Combining this with Lemma 4.4 for $\xi = 1$, using the fact that $2 + 2\log 4 < 5$, and counting the number of pairs $(k, \ell)$ for which $|c_{k\ell}(r)|$ may exceed $\varepsilon$ gives the result. □

We now employ this rank bound for each compressed block of the BA-MHT to analyze the complexity of our algorithm applied to the flat torus. Figure 4.5 shows a diagram of the two-dimensional Fourier transform as a MHT and its relation to the annulus-to-disk Fourier kernel. The following result agrees with the numerical experiments of Section 4.6, but we will neglect certain logarithmic terms and will assume — as we observe in practice — that the storage and application complexity of the MHT is dominated by the middle level $\ell = \frac{L}{2}$ of the butterfly factorization. Therefore, the following sketch gives the necessary intuition bridging the above theoretical results to the numerical experiments of the next section, but lacks the formal details required to constitute a complete proof. A rigorous version of this result will appear in an forthcoming publication.

**Claim 4.6.** *Let $\mathcal{M} = [-\pi, \pi]^2$. The BA-MHT mapping $m$ coefficients to a linear combination of eigenfunctions at $n$ points is $O(n + m^{3/2})$ complexity to store and apply.*

*Proof sketch.* Assume $m \leq n$. In addition, assume that $m = 4^L$ for ease of notation. At each level $\ell = 0, \ldots, L$ and each box index $k = 1, \ldots, 4^\ell$ and $j = 1, \ldots, 4^{L-\ell}$, we apply a transfer matrix corresponding to $(\boldsymbol{\omega}, \boldsymbol{x}) \in A_{\sqrt{\rho_{k-1}}}^{\sqrt{\rho_k}} \times B_{\sqrt{2}\pi/2^\ell}(\boldsymbol{c}_j)$, where $\rho_k := k\lambda_m/4^{L-\ell}$ is the upper boundary of box $k$ on level $\ell$ of a 4-ary tree on $[0, \lambda_m]$, and $\boldsymbol{c}_j$ is the center of box $j$ on level $L - \ell$ of a quadtree on $[-\pi, \pi]^2$.

Take $a = \sqrt{\rho_{k-1}}$, $b = \sqrt{\rho_k}$, and $R = \sqrt{2}\pi/2^\ell$. In this proof sketch, we neglect the term involving $\log\big((b - a)R\big)$ in Theorem 4.5. Combining this theorem and the fact that Weyl's law [23] gives $\lambda_m \sim m/\pi$, we see that the $\varepsilon$-rank of block $(j, k)$ at level $\ell$ is of size

$$r_\varepsilon(\ell, j, k) = O\left(\sqrt{\frac{k\lambda_m}{4^{L-\ell}}}\frac{1}{2^\ell}\right) = O(\sqrt{k}) \tag{4.37}$$

At the same time, by Weyl's law the number of eigenvalues in the interval $[\rho_{k-1}, \rho_k]$, and therefore an upper bound on the size and rank of the corresponding MHT block is

$$r_\varepsilon(\ell, j, k) = O\left(\frac{k\lambda_m}{4^{L-\ell}} - \frac{(k-1)\lambda_m}{4^{L-\ell}}\right) = O(4^\ell) \tag{4.38}$$

Therefore, in early levels of the factorization there are very few eigenvalues in each interval $[\rho_{k-1}, \rho_k]$ and thus $r_\varepsilon$ is bounded by $O(4^\ell)$ for each block independent of $k$, despite the fact that the continuous rank of the corresponding annulus-to-disk Fourier kernel may be much larger. In contrast, in later levels the number of eigenvalues in each interval $[\rho_{k-1}, \rho_k]$ is larger than the $O(\sqrt{k})$ continuous rank of the Fourier kernel, and thus compression occurs.

Let $p_j$ denote the parent of block $j$ at level $\ell$ in the space tree, and let $c_k$ denote a child of block $k$ in the frequency tree. For levels $\ell = 0, \ldots, L - 1$, we store and apply transfer matrices which map the row bases at level $\ell$ to row bases at level $\ell + 1$, resulting in a cost of

$$\sum_{\ell=0}^{L-1} \sum_{j=1}^{4^\ell} \sum_{k=1}^{4^{L-\ell}} O\Big(r_\varepsilon(\ell, p_j, c_k) \cdot r_\varepsilon(\ell + 1, j, k)\Big). \tag{4.39}$$

The two numerical ranks appearing above are within a factor of four of each other because even if no compression occurs, moving one level up the quadtree can at most quadruple the size of the basis. Empirically, the transition between $\mathcal{O}(4^\ell)$ and $\mathcal{O}(\sqrt{k})$ ranks occurs at the middle level $\ell = \frac{L}{2}$. This remains to be rigorously established using Theorem 4.5. If this is taken for granted for the purpose of this proof sketch, however, then the complexity is dominated by the middle level, where $r_\varepsilon(\ell, p_j, c_k)$ and $r_\varepsilon(\ell + 1, j, k)$ are both $\mathcal{O}(\sqrt{k})$. This gives

$$\sum_{j=1}^{\sqrt{m}} \sum_{k=1}^{\sqrt{m}} \mathcal{O}(k) = \mathcal{O}(m^{3/2}). \tag{4.40}$$

All that remains is to determine the complexity of storing and applying the last level, where compressed blocks are mapped to values at point in physical space. Let $n_j$ be the number of points in box $j$ on the leaf level $L$ of a quadtree on $[-\pi, \pi]^2$. At this level $\ell = L$, we store and apply column basis matrices for each block row, yielding a cost of

$$\sum_{j=1}^{m} \mathcal{O}\Big(n_j \cdot r_\varepsilon(L, j, 1)\Big) = \sum_{j=1}^{m} \mathcal{O}(n_j) = \mathcal{O}(n) \tag{4.41}$$

This gives the result. □

In common application scenarios, one typically increases the number of desired eigenfunctions $m$ proportionally with the number of physical points $n$. In the case of the Fourier transform this corresponds to keeping the number of points per wavelength constant. Similar considerations must be made if the eigenfunctions $\varphi_k$ are computed numerically in order to assure their sufficient accuracy. The following simple corollary treats this natural scenario.

**Corollary 4.7.** *Let $\mathcal{M} = [-\pi, \pi]^2$. For $m = \mathcal{O}(n)$, the BA-MHT is $\mathcal{O}(n^{3/2})$ complexity to store and apply.*

We will see in the following section that the BA-MHT is largely geometry-agnostic, performing similarly on the flat torus, the sphere, and more generic compact manifolds. Thus we make the following conjecture generalizing Claim 4.6 to general manifolds $\mathcal{M}$.

**Conjecture 4.8.** *The BA-MHT is $O\big(n + m^{\frac{d+1}{2}}\big)$ to store and apply on any compact $d$-manifold $\mathcal{M}$ for $d \geq 2$.*

Note that the eigenfunctions of the Laplacian on any 1-manifold are $e^{iks}$ where $s$ is arclength. Thus the BA-MHT is exactly a butterfly-based NUFFT [75, 89], which has $O(n + m \log m)$ complexity to store and apply.

## 4.6 Numerical experiments

We now provide a number of numerical experiments to demonstrate the scalability and applicability of our algorithm. We begin by testing the butterfly compression properties of the MHT on various manifolds. First, we consider the sphere. In this case, the Laplace-Beltrami eigenfunctions are the spherical harmonics $\varphi_k(\boldsymbol{x}) = Y_\ell^p(\boldsymbol{x})$ with corresponding eigenvalues $\lambda_k = \ell(\ell + 1)$. Therefore, we can evaluate arbitrary entries of $\boldsymbol{\Phi}$ directly using existing numerical libraries for spherical harmonics, and can avoid discretizing the sphere and numerically computing $\varphi_k$. We choose our spatial locations $\{\boldsymbol{x}_j\}_{j=1}^n$ uniformly at random on the surface of the sphere, and use the BA-MHT to perform a nonuniform spherical harmonic transform [9, 69].

Next, we consider a deformed torus. As this manifold does not have known closed-form eigenfunctions, we must compute $\varphi_k$ numerically. For this purpose we choose our spatial locations $\{\boldsymbol{x}_j\}_{j=1}^n$ to be nodes of a high-order spectral collocation scheme [38] which we use to discretize the Laplace-Beltrami operator. We then employ a Krylov eigensolver [113] to compute blocks of eigenfunctions. On both the sphere and the deformed torus, for various $n$ we compute and compress $m = \lceil n/32 \rceil$ eigenfunctions using the BA-MHT. Figure 4.6 displays three of the highest frequency eigenfunctions on the largest tested mesh of the deformed torus. Figure 4.7 illustrates $O(n^{3/2})$ scaling for the memory required to store the BA-MHT on both manifolds, just as we expect from Corollary 4.7. Relative to the $O(n^2)$ cost of storing the dense matrix, we see that the BA-MHT yields a factor of ~50 reduction in memory for $n \approx 10^6$ on the sphere, and a fac-

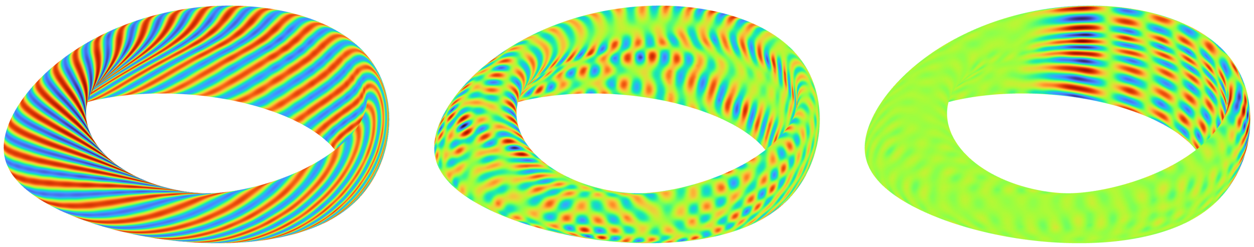


**Figure 4.6:** Three of the highest frequency Laplace-Beltrami eigenfunctions computed on the deformed torus with $n = 51{,}200$ spectral collocation nodes and $m = \lceil n/32 \rceil = 1{,}600$ eigenfunctions.

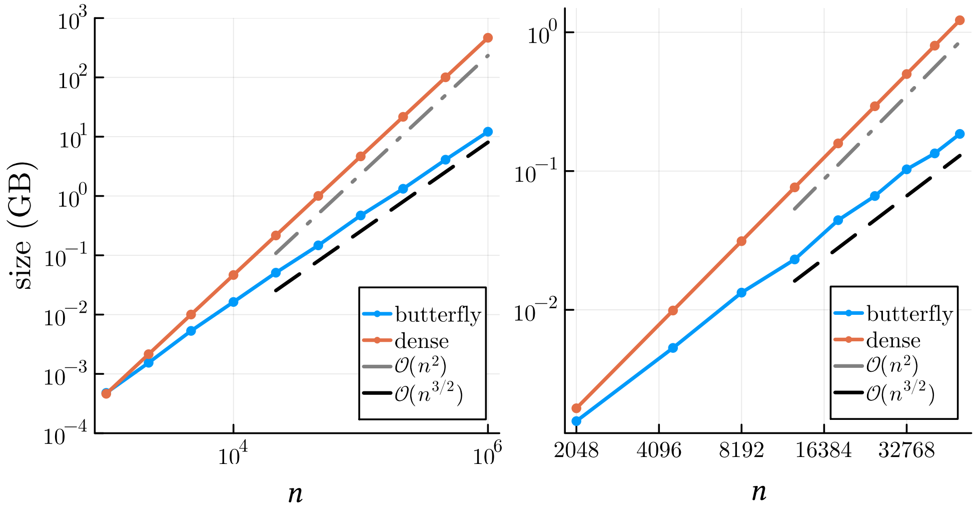
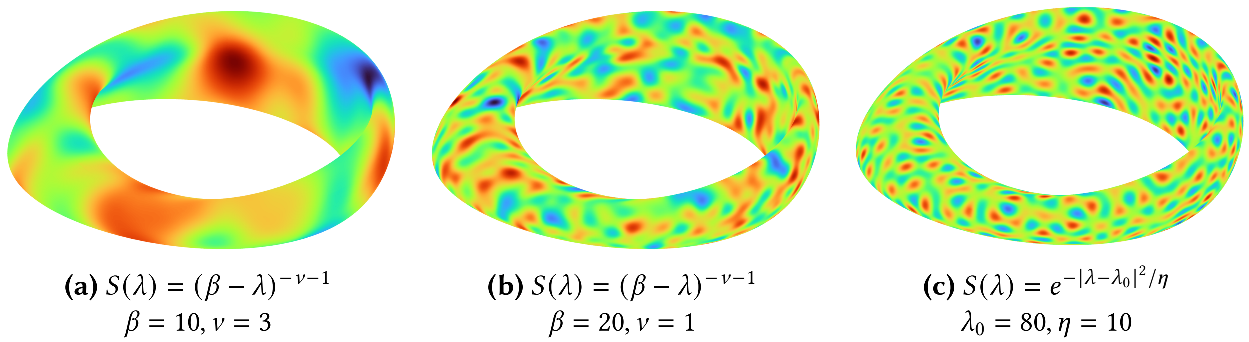


**Figure 4.7:** Scaling of MHT on the sphere (left) and on the deformed torus (right) with tolerance $\varepsilon = 10^{-3}$ and $m = \lceil n/32 \rceil$.

tor of ~10 reduction for $n \approx 50{,}000$ on the deformed torus. Recall that the complexity of storing and applying the MHT are asymptotically equivalent because every stored transfer matrix must be applied to a vector once, and thus this memory scaling is also a proxy for matvec runtime scaling.

### 4.6.1 Gaussian random fields on manifolds

We now return to the motivating context of GRFs on manifolds. Given a spectral density function $S_{\theta}(\lambda)$, one can generate approximate samples from the corresponding process by truncating



**(a)** $S(\lambda) = (\beta - \lambda)^{-\nu-1}$ $\beta = 10, \nu = 3$

**(b)** $S(\lambda) = (\beta - \lambda)^{-\nu-1}$ $\beta = 20, \nu = 1$

**(c)** $S(\lambda) = e^{-|\lambda-\lambda_0|^2/\eta}$ $\lambda_0 = 80, \eta = 10$

**Figure 4.8:** Samples from GRFs for Matérn spectral densities (a)-(b), and for a random surface wave model (c) on deformed torus.

the Karhunen-Loève expansion

$$Z(\boldsymbol{x}_j) = \sum_{k=1}^{\infty} \sqrt{S_\theta(\lambda_k)} Z_k \varphi_k(\boldsymbol{x}_j) \quad \text{for } j = 1, \ldots, n, \quad Z_k \overset{\text{i.i.d.}}{\sim} N(0,1). \tag{4.42}$$

This is precisely a MHT, and thus the BA-MHT can be applied directly to accelerate sampling from these processes. Figure 4.8 shows samples drawn for various GRF models on the deformed torus. As one expects from global spectral methods, this approach is best suited to smooth random fields. For rough random fields with highly localized covariance functions, FEM-based approaches [14, 70] which use approximately local bases will likely be more performant. A thorough numerical comparison of these two approaches remains for future work.

## 4.7 Discussion

In this chapter, we presented a butterfly-accelerated manifold harmonic transform for rapidly computing linear combinations of Laplace-Beltrami eigenfunctions on arbitrary compact manifolds. We present empirical evidence that our algorithm requires $O(n + m^{3/2})$ memory and time to map $m$ coefficients to $n$ point values on any compact 2-manifold in $\mathbb{R}^3$, and conjecture that the complexity is $O(n + m^{\frac{d+1}{2}})$ for general compact $d$-manifolds. We provide bounds on the rank of the Fourier transform between an annulus and a disk in two dimensions, which forms the basis of a proof for the observed asymptotic complexity in the special case of $\mathcal{M} = [-\pi, \pi]^2$.

There remain several interesting directions which are the subject of ongoing and future work. First, there exist a number of applications to which the BA-MHT can be directly applied, including spectral methods for surface PDEs [35], geometry processing [120], and kernel learning [8]. For these applications, generalizations of this work to higher dimensions and to manifolds with boundary must be considered. We believe these cases can be treated in a straightforward way using our algorithm. In addition, methods for reducing the eigenfunction computation and butterfly factorization costs remain of significant practical interest, as both these precomputations require $\tilde{O}(nm)$ time in general. Of immediate interest are multiresolution schemes in which low-frequency eigenfunctions are computed on coarser discretizations, and multiscale approaches [10] in which the global eigenproblem is subdivided into local ones.

# 5 | Conclusion

In this dissertation, we have introduced three numerical methods which facilitate efficient modeling of Gaussian random fields directly from their spectral densities in three different geometric settings: the real line, higher-dimensional Euclidean space, and compact manifolds. Given a spectral density $S_{\boldsymbol{\theta}}$, Chapters 2 and 3 each present one of the necessary components for rapid evaluation of the corresponding stationary covariance function $K_{\boldsymbol{\theta}}(\boldsymbol{r}_j)$ at a large number of distances $j = 1, \ldots, n$ in $\mathbb{R}^d$. Efforts are underway to combine these ideas, utilizing the nonuniform fast Hankel transform of Chapter 3 to evaluate panel integrals within the adaptive integration framework of Chapter 2 for applications in $d \geq 2$. Integrating these algorithms with a convolutional method for specifying nonstationary GRFs [93] and the Vecchia approximation [112] for rapid maximum likelihood estimation would facilitate scalable inference for the types of complex, large scale, and highly nonstationary data found in spatiotemporal statistics applications. We look forward to pursuing these highly flexible spectral modeling possibilities.

We pivot away from entry evaluation schemes and towards “Fourier feature”-type methods in Chapter 4, presenting an accelerated manifold harmonic transform for rapidly computing linear combinations of Laplace-Beltrami eigenfunctions on compact manifolds $\mathcal{M}$. These eigenfunctions must be numerically precomputed in general, which is the computational bottleneck in practice. Therefore, various multiscale methods of accelerating this precomputation are being explored. In addition, we remain particularly interested in applications to smooth Gaussian random fields, to graphics [120], and to kernel learning and forecasting [8, 46]. In these cases, only relatively few eigenfunctions (e.g. hundreds to tens of thousands) may need be computed in order to accurately capture the relevant phenomena, and thus our algorithm can be efficiently applied without being limited by its precomputation cost.

# Bibliography

[1] Ismail Ali and S Kalla. "A generalized Hankel transform and its use for solving certain partial differential equations". In: *The ANZIAM Journal* 41.1 (1999), pp. 105–117.

[2] Alex Barnett, Philip Greengard, and Manas Rachh. "Uniform approximation of common Gaussian process kernels using equispaced Fourier grids". In: *arXiv preprint arXiv:2305.11065* (2023).

[3] Alexander H. Barnett, Jeremy Magland, and Ludvig af Klinteberg. "A parallel nonuniform fast Fourier transform library based on an "exponential of semicircle" kernel". In: *SIAM Journal on Scientific Computing* 41.5 (2019), pp. C479–C504.

[4] Paul G Beckman and Christopher J Geoga. "Fast adaptive Fourier integration for spectral densities of Gaussian processes". In: *Statistics and Computing* 34.6 (2024), p. 217.

[5] Paul G Beckman and Michael O'Neil. "A Nonuniform Fast Hankel Transform". In: *arXiv preprint arXiv:2411.09583* (2024).

[6] Paul G Beckman et al. "Scalable computations for nonstationary Gaussian processes". In: *Statistics and Computing* 33.4 (2023), p. 84.

[7] Mikhail Belkin and Partha Niyogi. "Laplacian eigenmaps and spectral techniques for embedding and clustering". In: *Advances in neural information processing systems* 14 (2001).

[8] Mikhail Belkin and Partha Niyogi. "Laplacian eigenmaps for dimensionality reduction and data representation". In: *Neural computation* 15.6 (2003), pp. 1373–1396.

[9] Sebastian Belkner et al. "cunuSHT: GPU accelerated spherical harmonic transforms on arbitrary pixelizations". In: *RAS Techniques and Instruments* 3.1 (2024), pp. 711–721.

[10] Jeffrey K Bennighof and Richard B Lehoucq. "An automated multilevel substructuring method for eigenspace computation in linear elastodynamics". In: *SIAM Journal on Scientific Computing* 25.6 (2004), pp. 2084–2106.

[11] Matt Berger et al. "Fiedler trees for multiscale surface analysis". In: *Computers & Graphics* 34.3 (2010), pp. 272–281.

[12] Nicholas H Bingham, Charles M Goldie, and Jef L Teugels. *Regular variation.* 27. Cambridge university press, 1989.

[13] Rob Bisseling and Ronnie Kosloff. "The fast Hankel transform as a tool in the solution of the time dependent Schrödinger equation". In: *Journal of Computational Physics* 59.1 (1985), pp. 136–151.

[14] David Bolin and Kristin Kirchner. "The rational SPDE approach for Gaussian random fields with general smoothness". In: *Journal of Computational and Graphical Statistics* 29.2 (2020), pp. 274–285.

[15] David Bolin, Vaibhav Mehandiratta, and Alexandre B Simas. "Linear cost and exponentially convergent approximation of Gaussian Matérn processes on intervals". In: *Journal of Machine Learning Research* 26.96 (2025), pp. 1–34.

[16] Andrea Bonito et al. "AFEM for geometric PDE: the Laplace-Beltrami operator". In: *Analysis and numerics of partial differential equations* (2013), pp. 257–306.

[17] Viacheslav Borovitskiy, Alexander Terenin, Peter Mostowsky, et al. "Matérn Gaussian processes on Riemannian manifolds". In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 12426–12437.

[18] John P Boyd and Fu Yu. "Comparing seven spectral methods for interpolation and for solving the Poisson equation in a disk: Zernike polynomials, Logan–Shepp ridge polynomials, Chebyshev–Fourier series, cylindrical Robert functions, Bessel–Fourier expansions, square-to-disk conformal mapping and radial basis functions". In: *Journal of Computational Physics* 230.4 (2011), pp. 1408–1438.

[19] J Brunol and P Chavel. "Fourier transformation of rotationally invariant two-variable functions: Computer implementation of Hankel transform". In: *Proceedings of the IEEE* 65.7 (1977), pp. 1089–1090.

[20] Richard H. Byrd, Jorge Nocedal, and Richard A. Waltz. "Knitro: an integrated package for nonlinear optimization". en. In: *Large-Scale Nonlinear Optimization.* Ed. by G. Di Pillo and M. Roma. Nonconvex Optimization and Its Applications. Boston, MA: Springer US, 2006, pp. 35–59. ISBN: 9780387300658. DOI: 10.1007/0-387-30065-1_4.

[21] Emmanuel Candes, Laurent Demanet, and Lexing Ying. "A fast butterfly algorithm for the computation of Fourier integral operators". In: *Multiscale Modeling & Simulation* 7.4 (2009), pp. 1727–1750.

[22] Eduardo Cavanagh and B Cook. "Numerical evaluation of Hankel transforms via Gaussian-Laguerre polynomial expansions". In: *IEEE transactions on acoustics, speech, and signal processing* 27.4 (1979), pp. 361–366.

[23] Isaac Chavel. *Eigenvalues in Riemannian geometry.* Vol. 115. Academic press, 1984.

[24] Jeff Cheeger. "A lower bound for the smallest eigenvalue of the Laplacian". In: *Problems in analysis* 625.195-199 (1970), p. 110.

[25] Jie Chen and Michael L Stein. "Linear-cost covariance functions for Gaussian random fields". In: *Journal of the American Statistical Association* 118.541 (2023), pp. 147–164.

[26] Jiawei Chiu and Laurent Demanet. “Sublinear randomized algorithms for skeleton decompositions”. In: *SIAM Journal on Matrix Analysis and Applications* 34.3 (2013), pp. 1361–1383.

[27] Ronald R Coifman and Mauro Maggioni. “Diffusion wavelets”. In: *Applied and computational harmonic analysis* 21.1 (2006), pp. 53–94.

[28] James W. Cooley and John W. Tukey. “An algorithm for the machine calculation of complex Fourier series”. In: *Mathematics of computation* 19.90 (1965), pp. 297–301.

[29] M. J. Cree and P. J. Bones. “Algorithms to numerically evaluate the Hankel transform”. In: *Computers & Mathematics with Applications* 26.1 (1993), pp. 1–12.

[30] Noel Cressie and Hsin-Cheng Huang. “Classes of nonseparable, spatio-temporal stationary covariance functions”. In: *Journal of the American Statistical association* 94.448 (1999), pp. 1330–1339.

[31] Laurent Demanet et al. “A butterfly algorithm for synthetic aperture radar imaging”. In: *SIAM Journal on Imaging Sciences* 5.1 (2012), pp. 203–243.

[32] Claude R Dietrich and Garry N Newsam. “Fast and exact simulation of stationary Gaussian processes through circulant embedding of the covariance matrix”. In: *SIAM Journal on Scientific Computing* 18.4 (1997), pp. 1088–1107.

[33] Pierre Duhamel and Martin Vetterli. “Fast Fourier transforms: a tutorial review and a state of the art”. In: *Signal processing* 19.4 (1990), pp. 259–299.

[34] Alok Dutt and Vladimir Rokhlin. “Fast Fourier transforms for nonequispaced data”. In: *SIAM Journal on Scientific computing* 14.6 (1993), pp. 1368–1393.

[35] Gerhard Dziuk and Charles M Elliott. “Finite element methods for surface PDEs”. In: *Acta Numerica* 22 (2013), pp. 289–396.

[36] Björn Engquist and Lexing Ying. "A fast directional algorithm for high frequency acoustic scattering in two dimensions". In: (2009).

[37] Miroslav Fiedler. "A property of eigenvectors of nonnegative symmetric matrices and its application to graph theory". In: *Czechoslovak mathematical journal* 25.4 (1975), pp. 619–633.

[38] Daniel Fortunato. "A high-order fast direct solver for surface PDEs". In: *SIAM Journal on Scientific Computing* 46.4 (2024), A2582–A2606.

[39] Jochen Fritz, Insa Neuweiler, and Wolfgang Nowak. "Application of FFT-based algorithms for large-scale universal kriging problems". In: *Mathematical Geosciences* 41.5 (2009), pp. 509–533.

[40] Lehman H Garrison et al. "nifty-ls: Fast and Accurate Lomb–Scargle Periodograms Using a Non-uniform FFT". In: *Research Notes of the AAS* 8.10 (2024), p. 250.

[41] Marc G Genton and David J Gorsich. "Nonparametric variogram and covariogram estimation with Fourier–Bessel matrices". In: *Computational Statistics & Data Analysis* 41.1 (2002), pp. 47–57.

[42] Christopher J Geoga, Mihai Anitescu, and Michael L Stein. "Flexible nonstationary spatiotemporal modeling of high-frequency monitoring data". In: *Environmetrics* 32.5 (2021), e2670.

[43] Christopher J Geoga, Mihai Anitescu, and Michael L Stein. "Scalable Gaussian process computations using hierarchical matrices". In: *Journal of Computational and Graphical Statistics* 29.2 (2020), pp. 227–237.

[44] Christopher J Geoga and Michael L Stein. "A Scalable Method to Exploit Screening in Gaussian Process Models with Noise". In: *Journal of Computational and Graphical Statistics* just-accepted (2023), pp. 1–19.

[45] Christopher J Geoga et al. "Fitting Matérn smoothness parameters using automatic differentiation". In: *Statistics and Computing* 33.2 (2023), p. 48.

[46] Dimitrios Giannakis. "Data-driven spectral decomposition and forecasting of ergodic dynamical systems". In: *Applied and Computational Harmonic Analysis* 47.2 (2019), pp. 338–396.

[47] Andreas Glaser, Xiangtao Liu, and Vladimir Rokhlin. "A fast algorithm for the calculation of the roots of special functions". In: *SIAM Journal on Scientific Computing* 29.4 (2007), pp. 1420–1438.

[48] Gene H Golub and Charles F Van Loan. *Matrix computations*. John Hopkins University Press, 1983.

[49] Pedro Gonnet. "A review of error estimation in adaptive quadrature". In: *ACM Computing Surveys (CSUR)* 44.4 (2012), pp. 1–36.

[50] Jake P Grainger et al. "Estimating the parameters of ocean wave spectra". In: *Ocean Engineering* 229 (2021), p. 108934.

[51] Leslie Greengard and June-Yub Lee. "Accelerating the nonuniform fast Fourier transform". In: *SIAM review* 46.3 (2004), pp. 443–454.

[52] Leslie Greengard, June-Yub Lee, and Souheil Inati. "The fast sinc transform and image reconstruction from nonuniform samples in k-space". In: *Communications in Applied Mathematics and Computational Science* 1.1 (2007), pp. 121–131.

[53] Philip Greengard. "Efficient Fourier Representations of Families of Gaussian Processes". In: *SIAM Journal on Scientific Computing* 47.2 (2025), A971–A990.

[54] Philip Greengard, Manas Rachh, and Alex Barnett. "Equispaced Fourier representations for efficient Gaussian process regression from a billion data points". In: *arXiv preprint arXiv:2210.10210* (2022).

[55] Andreas Griewank and Andrea Walther. *Evaluating derivatives: principles and techniques of algorithmic differentiation*. SIAM, 2008.

[56] Joseph Guinness. "Gaussian process learning via Fisher scoring of Vecchia's approximation". In: *Statistics and Computing* 31.3 (2021), p. 25.

[57] Nicholas Hale and Alex Townsend. "Fast and accurate computation of Gauss–Legendre and Gauss–Jacobi quadrature nodes and weights". In: *SIAM Journal on Scientific Computing* 35.2 (2013), A652–A674.

[58] E Hansen. "Fast Hankel transform algorithm". In: *IEEE transactions on acoustics, speech, and signal processing* 33.3 (1985), pp. 666–671.

[59] James Hensman, Nicolas Durrande, and Arno Solin. "Variational Fourier features for Gaussian processes". In: *Journal of Machine Learning Research* 18.151 (2018), pp. 1–52.

[60] William E Higgins and David C Munson. "A Hankel transform approach to tomographic image reconstruction". In: *IEEE transactions on medical imaging* 7.1 (1988), pp. 59–72.

[61] Michael T Horrell and Michael L Stein. "Half-spectral space–time covariance models". In: *Spatial Statistics* 19 (2017), pp. 90–100.

[62] I. A. Ibragimov and Y. A. Rozanov. *Gaussian random processes*. en. Stochastic Modelling and Applied Probability. New York: Springer-Verlag, 1978. ISBN: 9780387903026. DOI: 10.1007/978-1-4612-6275-6.

[63] Hae Kyung Im, Michael L Stein, and Zhengyuan Zhu. "Semiparametric estimation of spectral density with irregular observations". In: *Journal of the American Statistical Association* 102.478 (2007), pp. 726–735.

[64] H. K. Johansen and K. Sørensen. "Fast Hankel transforms". In: *Geophysical Prospecting* 27.4 (1979), pp. 876–901.

[65] F. Johansson. “Arb: efficient arbitrary-precision midpoint-radius interval arithmetic”. In: *IEEE T Comput* 66 (8 2017), pp. 1281–1292.

[66] H Fisk Johnson. “An improved method for computing a discrete Hankel transform”. In: *Computer physics communications* 43.2 (1987), pp. 181–202.

[67] Sharad Kapur and Vladimir Rokhlin. *An algorithm for the fast Hankel transform.* Tech. rep. Computer Science Department, Yale University, 1995.

[68] Matthias Katzfuss and Joseph Guinness. “A general framework for Vecchia approximations of Gaussian processes”. In: (2021).

[69] Stefan Kunis and Daniel Potts. “Fast spherical Fourier algorithms”. In: *Journal of Computational and Applied Mathematics* 161.1 (2003), pp. 75–98.

[70] Annika Lang and Mike Pereira. “Galerkin–Chebyshev approximation of Gaussian random fields on compact Riemannian manifolds”. In: *BIT Numerical Mathematics* 63.4 (2023), p. 51.

[71] Annika Lang and Christoph Schwab. “Isotropic Gaussian random fields on the sphere: regularity, fast simulation and stochastic partial differential equations”. In: *The Annals of Applied Probability* (2015), pp. 3047–3094.

[72] Bruno Lévy. “Laplace-Beltrami eigenfunctions towards an algorithm that “understands” geometry”. In: *IEEE International Conference on Shape Modeling and Applications 2006 (SMI’06).* IEEE. 2006, pp. 13–13.

[73] Bo Li, Yan-Rui Li, and Xiao-Long Zhang. “A survey on Laplacian eigenmaps based manifold learning methods”. In: *Neurocomputing* 335 (2019), pp. 336–351.

[74] Yingzhou Li and Haizhao Yang. “Interpolative butterfly factorization”. In: *SIAM Journal on Scientific Computing* 39.2 (2017), A503–A531.

[75] Yingzhou Li et al. “Butterfly factorization”. In: *Multiscale Modeling & Simulation* 13.2 (2015), pp. 714–732.

[76] Finn Lindgren, Håvard Rue, and Johan Lindström. “An explicit link between Gaussian fields and Gaussian Markov random fields: the stochastic partial differential equation approach”. In: *Journal of the Royal Statistical Society Series B: Statistical Methodology* 73.4 (2011), pp. 423–498.

[77] CM Linton. “Schlömilch series that arise in diffraction theory and their efficient computation”. In: *Journal of Physics A: Mathematical and General* 39.13 (2006), p. 3325.

[78] Qing Huo Liu and Zhong Qing Zhang. “Nonuniform fast Hankel transform (NUFHT) algorithm”. In: *Applied optics* 38.32 (1999), pp. 6705–6708.

[79] Reginald Douglas Lord. “The use of the Hankel transform in statistics I. General theory and examples”. In: *Biometrika* 41.1/2 (1954), pp. 44–55.

[80] Reginald Douglas Lord. “The use of the Hankel transform in statistics II. Methods of computation”. In: *Biometrika* 41.3/4 (1954), pp. 344–350.

[81] Nicholas F Marshall, Oscar Mickelin, and Amit Singer. “Fast expansion into harmonics on the disk: A steerable basis with fast radial convolutions”. In: *SIAM Journal on Scientific Computing* 45.5 (2023), A2431–A2457.

[82] Martin J Mohlenkamp. “A fast transform for spherical harmonics”. In: *Journal of Fourier analysis and applications* 5 (1999), pp. 159–184.

[83] D Mook. “An algorithm for the numerical evaluation of the Hankel and Abel transforms”. In: *IEEE transactions on acoustics, speech, and signal processing* 31.4 (1983), pp. 979–985.

[84] Paytsar Muradyan and Richard L Coulter. *Radar Wind Profiler (RWP) and Radio Acoustic Sounding System (RASS) Instrument Handbook (DOE/SC-ARM-TR-044).* United States: DOE Office of Science Atmospheric Radiation Measurement (ARM) Program, 2020.

[85] Ahmad Nasikun, Christopher Brandt, and Klaus Hildebrandt. "Fast Approximation of Laplace-Beltrami Eigenproblems". In: *Computer Graphics Forum*. Vol. 37. 5. Wiley Online Library. 2018, pp. 121–134.

[86] R. K. Newsom. *Doppler lidar (Dl) handbook*. English. Tech. rep. DOE/SC-ARM/TR-101. DOE Office of Science Atmospheric Radiation Measurement (ARM) Program (United States), 2012. DOI: 10.2172/1034640.

[87] John P. Nolan. "Multivariate elliptically contoured stable distributions: theory and estimation". In: *Computational statistics* 28 (2013), pp. 2067–2089.

[88] Michael O'Neil. *A new class of analysis-based fast transforms*. Tech. rep. Yale University, 2007.

[89] Michael O'Neil, Franco Woolfe, and Vladimir Rokhlin. "An algorithm for the rapid evaluation of special function transforms". In: *Applied and Computational Harmonic Analysis* 28.2 (2010), pp. 203–226.

[90] Frank W. Olver et al. *NIST handbook of mathematical functions*. Cambridge university press, 2010.

[91] Alan V Oppenheim, George V Frisk, and David R Martinez. "Computation of the Hankel transform using projections". In: *The Journal of the Acoustical Society of America* 68.2 (1980), pp. 523–529.

[92] Christopher J Paciorek. "Bayesian smoothing with Gaussian processes using Fourier basis functions in the spectralGP package". In: *Journal of statistical software* 19 (2007), pp. 1–38.

[93] Christopher J Paciorek and Mark J Schervish. "Spatial modelling using a new class of nonstationary covariance functions". In: *Environmetrics: The official journal of the International Environmetrics Society* 17.5 (2006), pp. 483–506.

[94] Qiyuan Pang, Kenneth L Ho, and Haizhao Yang. “Interpolative decomposition butterfly factorization”. In: *SIAM Journal on Scientific Computing* 42.2 (2020), A1097–A1115.

[95] Giuseppe Patanè. “Laplacian spectral basis functions”. In: *Computer aided geometric design* 65 (2018), pp. 31–47.

[96] Emilio Porcu and Michael L Stein. “On some local, global and regularity behaviour of some classes of covariance functions”. In: *Advances and challenges in space-time modelling of natural events*. Springer. 2012, pp. 221–238.

[97] Ali Rahimi and Benjamin Recht. “Random features for large-scale kernel machines”. In: *Advances in neural information processing systems* 20 (2007).

[98] Aaditya Rangan et al. “Factorization of the translation kernel for fast rigid image alignment”. In: *Inverse Problems* 36.2 (2020), p. 024001.

[99] Martin Reuter et al. “Discrete Laplace–Beltrami operators for shape analysis and segmentation”. In: *Computers & Graphics* 33.3 (2009), pp. 381–390.

[100] Walter Rudin. *Fourier analysis on groups*. Wiley, 1962.

[101] Paul D Sampson and Peter Guttorp. “Nonparametric estimation of nonstationary spatial covariance structure”. In: *Journal of the American Statistical Association* 87.417 (1992), pp. 108–119.

[102] Daniel Sanz-Alonso and Ruiyi Yang. “Finite element representations of Gaussian processes: Balancing numerical and statistical accuracy”. In: *SIAM/ASA Journal on Uncertainty Quantification* 10.4 (2022), pp. 1323–1349.

[103] DS Seljebotn. “Wavemoth–fast spherical harmonic transforms by butterfly matrix compression”. In: *The Astrophysical Journal Supplement Series* 199.1 (2012), p. 5.

[104] Seongho Seo, Moo K Chung, and Houri K Vorperian. “Heat kernel smoothing using Laplace-Beltrami eigenfunctions”. In: *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2010: 13th International Conference, Beijing, China, September 20-24, 2010, Proceedings, Part III 13*. Springer. 2010, pp. 505–512.

[105] Jianbo Shi and Jitendra Malik. “Normalized cuts and image segmentation”. In: *IEEE Transactions on pattern analysis and machine intelligence* 22.8 (2000), pp. 888–905.

[106] Paz Fink Shustin and Haim Avron. “Gauss-Legendre features for Gaussian process regression”. In: *Journal of Machine Learning Research* 23.92 (2022), pp. 1–47.

[107] AE Siegman. “Quasi fast Hankel transform”. In: *Optics letters* 1.1 (1977), pp. 13–15.

[108] Arno Solin and Manon Kok. “Know your boundaries: Constraining Gaussian processes by variational harmonic features”. In: *The 22nd International Conference on Artificial Intelligence and Statistics*. PMLR. 2019, pp. 2193–2202.

[109] Arno Solin and Simo Särkkä. “Hilbert space methods for reduced-rank Gaussian process regression”. In: *Statistics and Computing* 30.2 (2020), pp. 419–446.

[110] Michael L Stein. *Interpolation of spatial data: some theory for kriging*. Springer Science & Business Media, 1999.

[111] Michael L Stein. “Statistical methods for regular monitoring data”. In: *Journal of the Royal Statistical Society Series B: Statistical Methodology* 67.5 (2005), pp. 667–687.

[112] Michael L Stein, Zhiyi Chi, and Leah J Welty. “Approximating likelihoods for large spatial data sets”. In: *Journal of the Royal Statistical Society Series B: Statistical Methodology* 66.2 (2004), pp. 275–296.

[113] Gilbert W Stewart. “A Krylov–Schur algorithm for large eigenproblems”. In: *SIAM Journal on Matrix Analysis and Applications* 23.3 (2002), pp. 601–614.

[114] Adam M Sykulski et al. “The debiased Whittle likelihood”. In: *Biometrika* 106.2 (2019), pp. 251–266.

[115] Arthur D Szlam et al. “Diffusion-driven multiscale analysis on manifolds and graphs: top-down and bottom-up constructions”. In: *Wavelets XI*. Vol. 5914. SPIE. 2005, pp. 445–455.

[116] David J Thomson. “Spectrum estimation and harmonic analysis”. In: *Proceedings of the IEEE* 70.9 (2005), pp. 1055–1096.

[117] Alex Townsend. “A fast analysis-based discrete Hankel transform using asymptotic expansions”. In: *SIAM Journal on Numerical Analysis* 53.4 (2015), pp. 1897–1917.

[118] Lloyd N Trefethen. *Approximation theory and approximation practice, extended edition*. SIAM, 2019.

[119] Mark Tygert. “Fast algorithms for spherical harmonic expansions, III”. In: *Journal of Computational Physics* 229.18 (2010), pp. 6181–6192.

[120] Bruno Vallet and Bruno Lévy. “Spectral geometry processing with manifold harmonics”. In: *Computer Graphics Forum*. Vol. 27. Wiley Online Library. 2008, pp. 251–260.

[121] Aldo V Vecchia. “Estimation and model identification for continuous spatial processes”. In: *Journal of the Royal Statistical Society Series B: Statistical Methodology* 50.2 (1988), pp. 297–312.

[122] George Neville Watson. *A treatise on the theory of Bessel functions*. Vol. 2. The University Press, 1922.

[123] Holger Wendland. “Error estimates for interpolation by compactly supported radial basis functions of minimal degree”. In: *Journal of approximation theory* 93.2 (1998), pp. 258–272.

[124] Peter Whittle. “Hypothesis testing in time series analysis”. In: *(No Title)* (1951).

[125] Peter Whittle. “Stochastic-processes in several dimensions”. In: *Bulletin of the International Statistical Institute* 40.2 (1963), pp. 974–994.

[126] Andrew Wilson and Ryan Adams. “Gaussian process kernels for pattern discovery and extrapolation”. In: *International conference on machine learning*. PMLR. 2013, pp. 1067–1075.

[127] Jet Wimp. “Polynomial expansions of Bessel functions and some associated functions”. In: *Mathematics of Computation* 16.80 (1962), pp. 446–458.

[128] Hao Zhang. “Inconsistent estimation and asymptotically equal interpolations in model-based geostatistics”. In: *Journal of the American Statistical Association* 99.465 (2004), pp. 250–261.

[129] Zhizhen Zhao and Amit Singer. “Fourier–Bessel rotational invariant eigenimages”. In: *JOSA A* 30.5 (2013), pp. 871–877.

[130] Rundong Zhou and Nicolas Grisouard. “Spectral solver for Cauchy problems in polar co-ordinates using discrete Hankel transforms”. In: *arXiv preprint arXiv:2210.09736* (2022).