SIAM J. SCI. COMPUT.
Vol. 48, No. 2, pp. A784–A803

A NONUNIFORM FAST HANKEL TRANSFORM*

PAUL G. BECKMAN† AND MICHAEL O'NEIL†

Abstract. We describe a fast algorithm for computing discrete Hankel transforms of moderate orders from n nonuniform points to m nonuniform frequencies in $\mathcal{O}((m+n)\log\min(n,m))$ 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 ε. Several numerical examples are provided which demonstrate the speed and accuracy of the algorithm in multiple regimes and applications.

Key words. Hankel transform, fast Fourier transform, asymptotic expansion, Bessel function

MSC codes. 65R10, 33C10

DOI. 10.1137/25M1796758

See reproducibility of computational results at end of the article.

1. Introduction.

The fast Fourier transform (FFT) has revolutionized a wide range of applications across mathematics, statistics, and the physical sciences by enabling signal processing and Fourier analysis tasks to be performed using a computational cost which scales quasi-linearly with the number of data points n. However, the FFT requires that the input signal be sampled at equispaced points in time and that the desired output frequencies are equispaced on the integers. These assumptions are frequently not met in applications such as adaptive numerical partial differential equation (PDE) solvers [3, 21, 4, 32], magnetic resonance imaging [17, 8, 10], and various signal processing tasks [1, 43]. To overcome this setback, nonuniform FFT (NUFFT) algorithms have been developed [14, 16] which achieve near-FFT speeds in one dimension, assuming that the distribution of time samples and frequency outputs is not pathological. In higher dimensions, NUFFTs are less competitive with standard FFTs, but the computational task at hand is also significantly harder.

The FFT and NUFFT grew out of a need to perform Fourier transforms in Cartesian coordinates. However, depending on the particular problem, the relevant continuous Fourier analysis might be better suited to other coordinate systems. One such commonly encountered situation is computing the Fourier transform of radially symmetric functions in dimensions $d \geq 2$. For example, in two dimensions the Fourier transform of a function f is given by

$$g(\omega_1,\omega_2) = \frac{1}{4\pi^2}\iint_{\mathbb{R}^2} f(x_1,x_2)\, e^{-i(\omega_1 x_1+\omega_2 x_2)}\, dx_1\, dx_2. \tag{1.1}$$

*Submitted to the journal's Numerical Algorithms for Scientific Computing section September 27, 2025; accepted for publication (in revised form) November 6, 2025; published electronically April 6, 2026.

https://doi.org/10.1137/25M1796758

Funding: The first author was partially supported by the Office of Naval Research under award N00014-21-1-2383 and by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, Department of Energy Computational Science Graduate Fellowship under Award DE-SC0022158. The second author was partially supported by the Office of Naval Research under award N00014-21-1-2383.

†Courant Institute, New York University, New York, NY 10012 USA (paul.beckman@cims.nyu.edu, oneil@cims.nyu.edu).

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

$$\begin{aligned} g(\omega, \alpha) &= \frac{1}{4\pi^2} \int_0^{2\pi} \int_0^{\infty} f(r, \theta)\, e^{-i\omega r(\cos\alpha\cos\theta + \sin\alpha\sin\theta)}\, r\, dr\, d\theta \\ &= \frac{1}{4\pi^2} \int_0^{2\pi} \int_0^{\infty} f(r, \theta)\, e^{-i\omega r\cos(\alpha-\theta)}\, r\, dr\, d\theta. \end{aligned} \tag{1.2}$$

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

$$\begin{aligned} g(\omega) &= \frac{1}{4\pi^2} \int_0^{\infty} f(r)\, r \int_0^{2\pi} e^{-i\omega r\cos(\alpha-\theta)}\, d\theta\, dr \\ &= \frac{1}{2\pi} \int_0^{\infty} f(r)\, J_0(\omega r)\, r\, dr, \end{aligned} \tag{1.3}$$

where we have used the integral representation of the zeroth-order Bessel function [34]

$$J_0(x) = \frac{1}{\pi} \int_0^{\pi} \cos(x\cos\theta)\, d\theta. \tag{1.4}$$

The final integral involving J_0 in equation (1.3) 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 (1.2) has a particular periodic dependence in θ so that $f(r, \theta) = f(r)e^{i\nu\theta}$ with $\nu \in \mathbb{Z}$, then we have

$$\begin{aligned} g(\omega) &= \frac{1}{4\pi^2} \int_0^{\infty} f(r)\, r \int_0^{2\pi} e^{-i\omega r\cos(\alpha-\theta)}\, e^{i\nu\theta}\, d\theta\, dr \\ &= \frac{i^{\nu}}{2\pi} \int_0^{\infty} f(r)\, r\, J_{\nu}(\omega r)\, dr, \end{aligned} \tag{1.5}$$

where, again, we have invoked an integral representation for J_ν [34].

In order to numerically compute g in (1.3) or (1.5) at a collection of m "frequencies" ω_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

$$\begin{aligned} g(\omega_j) \approx g_j &:= \sum_{k=1}^{n} w_k\, f(r_k)\, r_k\, J_{\nu}(\omega_j r_k) \\ &= \sum_{k=1}^{n} c_k\, J_{\nu}(\omega_j r_k) \qquad \text{for } j = 1, \ldots, m. \end{aligned} \tag{1.6}$$

The above sum will be referred to as the discrete Hankel ransform (DHT) of order ν.

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 [19, 49, 30], statistics [28, 15], and separation of variables methods in PDEs [7, 2, 50]. In many such applications, a fully nonuniform DHT is desired, as the relevant frequencies ω_j

may not be equispaced, and the most efficient quadrature rule for discretizing (1.3) may have nodes r_k which are also not equispaced.

The algorithm of this work allows for arbitrary selection of the frequencies ω_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 [26, 44] 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 jth root of J_ν. In the most restrictive cases [23], 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 (1.3) and (1.6). These include series expansion methods [29, 11, 12], convolutional approaches [41, 22, 31, 27], and projection-slice or Abel transform-based methods [36, 18, 24]. See [13] 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 ω_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 [27], 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 [35, 25, 37] 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 ω_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, [44] 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 [44] to the fully nonuniform setting in a number of ways. First, we employ an adaptive partitioning scheme which, for any choice of ω_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 ω_j and r_k. Finally, we utilize the low-rank expansion of J_ν, given in [48] in the local regime where asymptotic expansions are not applicable. We derive error bounds for this low-rank expansion, allowing us

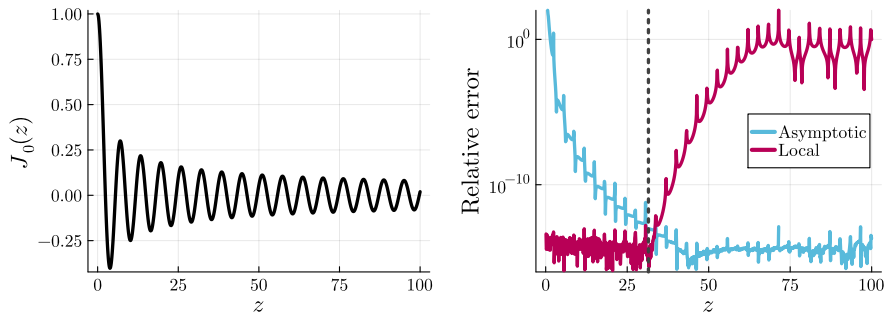

FIG. 1. *Bessel function $J_0(z)$ and pointwise relative error in approximating $J_0(z)$ using 31-term local and 4-term asymptotic expansions. Dotted vertical line shows crossover point where both expansions are accurate to $\varepsilon = 10^{-12}$. Note: color appears only in the online article.*

to choose all approximation parameters automatically by analysis which guarantees that the resulting error is bounded by the user-specified tolerance ε.

Outline of the paper. The paper is organized as follows. In section 2 we give a high level view of our algorithm, omitting technical details. Then in section 3 we study the local and asymptotic expansions of Bessel functions which serve as the key building blocks of the algorithm. Afterward, in section 4, we provide a detailed description of the algorithm and its associated complexity. Various numerical examples are provided in section 5, and we conclude with some additional discussion in section 6.

2. Overview of the algorithm. To more concisely describe our approach, we write the DHT (1.6) as the equivalent matrix-vector product with $\mathbf{A} \in \mathbb{R}^{m \times n}$:

$$\mathbf{g} = \mathbf{A}\mathbf{f}, \qquad \mathbf{A}(j,k) = J_\nu(\omega_j r_k). \tag{2.1}$$

The matrix $\mathbf{A}$ is in general full rank and possesses complex oscillatory structure. As a result, no straightforward fast algorithm exists to apply the full matrix $\mathbf{A}$ 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_ν.

When the argument $\omega_j r_k$ is small, J_ν is smooth and essentially nonoscillatory, and we use a closed-form local expansion which approximates J_ν 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_ν as a sum of a small number of decaying sinusoids, and can therefore be applied to a vector in quasi-linear time using the NUFFT. Figure 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 ε in the regions $\omega_j r_k \le z$ and $\omega_j r_k > z$, respectively. Next, we adaptively subdivide $\mathbf{A}$ into disjoint blocks so that either $\omega_j r_k \le z$ or $\omega_j r_k > z$ for all ω_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, and which can be directly applied. Figure 2 shows a Hankel transform matrix $\mathbf{A}$ divided into local

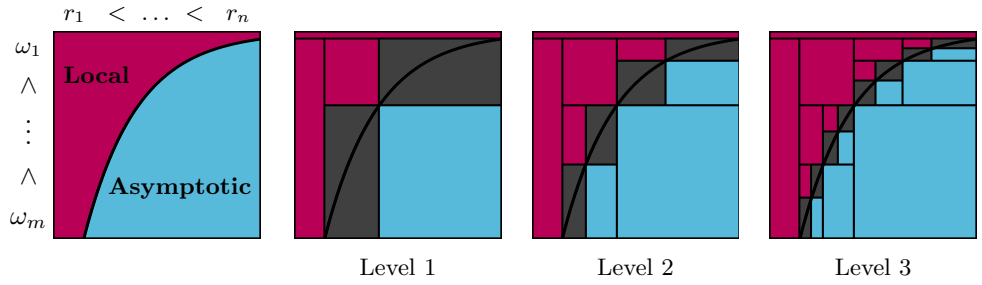

FIG. 2. *Splitting of Hankel transform matrix* **A** *along the curve* $\omega r = z$ *into local and asymptotic regions. Adaptive subdivision of* **A** *into corresponding local (red), asymptotic (blue), and mixed (gray) subblocks at various levels. Note: color appears only in the online article.*

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. 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 ε accuracy in both regimes.

3.1. The Wimp expansion. Near the origin, J_ν is a smooth and essentially nonoscillatory function. As a result, $J_\nu(\omega r)$ is a numerically low-rank function of all sufficiently small inputs ω and r. Fortuitously, one such low-rank expansion—which we refer to as the *Wimp expansion* — is available in closed form for integer ν [48]. In the case that ν is even, we have

$$
\begin{aligned}
J_\nu(\omega r) &= \sum_{\ell=0}^{\infty} C_\ell(\omega)\, T_{2\ell}(r), \\
C_\ell(\omega) &= \delta_\ell\, J_{\frac{\nu}{2}+\ell}(\omega)\, J_{\frac{\nu}{2}-\ell}(\omega), \\
\delta_\ell &= \begin{cases} 1, & \ell = 0, \\ 2, & \text{otherwise} \end{cases}
\end{aligned}
\tag{3.1}
$$

for all $|r| \le 1$, where $T_{2\ell}$ is the Chebyshev polynomial of the first kind of order 2ℓ. A similar expansion exists for ν odd [48, 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 ε-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 ν, the space-frequency product ΩR, and the number of retained terms L.

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

$$
\left| J_\nu(\omega r) - \sum_{\ell=0}^{L-1} C_\ell(\omega R) T_{2\ell}\left(\frac{r}{R}\right) \right| \le \frac{2\exp\left\{\frac{\nu}{2}(\beta-\gamma) + L(\beta+\gamma)\right\}}{1 - e^{\beta+\gamma}} =: B^{\mathrm{LOC}}_{\nu,L}(\Omega R)
\tag{3.2}
$$

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

$$\psi(p) := \log p + \sqrt{1-p^2} - \log\left(1+\sqrt{1-p^2}\right), \tag{3.3}$$

$$\beta := \psi\left(\frac{\Omega R}{2L+\nu}\right), \tag{3.4}$$

$$\gamma := \begin{cases} \psi\left(\frac{\Omega R}{2L-\nu}\right), & L > \dfrac{\nu}{2}, \\ 0, & \textit{otherwise.} \end{cases} \tag{3.5}$$

Proof. For ν even, the truncation error after L terms is bounded by

$$\left|\sum_{\ell=L}^{\infty} C_\ell(\omega R) T_{2\ell}\left(\frac{r}{R}\right)\right| \le 2\sum_{\ell=L}^{\infty} \left|J_{\frac{\nu}{2}+\ell}\left(\frac{\omega R}{2}\right)\right| \left|J_{\frac{\nu}{2}-\ell}\left(\frac{\omega R}{2}\right)\right|. \tag{3.6}$$

Define $p_\ell(\omega) := \omega R/(\nu+2\ell)$. Then by Siegel's bound [34, 10.14.5] we have

$$\left|J_{\frac{\nu}{2}+\ell}\left(\frac{\omega R}{2}\right)\right| = \left|J_{\frac{\nu}{2}+\ell}\left(\left(\frac{\nu}{2}+\ell\right) p_\ell(\omega)\right)\right| \tag{3.7}$$

$$\le \exp\left\{\left(\frac{\nu}{2}+\ell\right)\psi\left(p_\ell(\omega)\right)\right\} \tag{3.8}$$

$$\le \exp\left\{\left(\frac{\nu}{2}+\ell\right)\beta\right\}, \tag{3.9}$$

where the last inequality follows from the fact that ψ is an increasing function on $(0,1)$, and thus $\psi(p_\ell(\omega)) \le \beta < 0$ 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

$$\left|J_{\frac{\nu}{2}-\ell}\left(\frac{\omega R}{2}\right)\right| = \left|J_{\ell-\frac{\nu}{2}}\left(\left(\ell-\frac{\nu}{2}\right) q_\ell(\omega)\right)\right| \le \exp\left\{\left(\ell-\frac{\nu}{2}\right)\gamma\right\}. \tag{3.10}$$

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

All that remains is to apply a geometric series argument

$$\left|J_\nu(\omega r) - \sum_{\ell=0}^{L-1} C_\ell(\omega R) T_{2\ell}\left(\frac{r}{R}\right)\right| \le 2\sum_{\ell=L}^{\infty} \exp\left\{\left(\frac{\nu}{2}+\ell\right)\beta + \left(\ell-\frac{\nu}{2}\right)\gamma\right\} \tag{3.11}$$

$$= 2\exp\left\{\frac{\nu}{2}(\beta-\gamma)\right\} \sum_{\ell=L}^{\infty} \left(e^{\beta+\gamma}\right)^\ell \tag{3.12}$$

$$= \frac{2\exp\left\{\frac{\nu}{2}(\beta-\gamma) + L(\beta+\gamma)\right\}}{1-e^{\beta+\gamma}}. \tag{3.13}$$

A similar calculation can be carried out for ν 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 ψ, as done in [40, Lemma 1] for large ν. 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 ν. It is therefore well suited to our purposes because, given $z, L > 0$, it provides a bound $B^{\mathrm{LOC}}_{\nu,L}(z)$ on the pointwise error in approximating any block of the matrix $J_\nu(\omega_j r_k)$ for which $\omega r \le z$ 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,

$$\mathbf{A}(j_0 : j_1, k_0 : k_1) \approx \mathbf{C}\mathbf{T}^\top, \tag{3.14}$$

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

$$\mathbf{C}(j,\ell) = C_{\ell-1}(\omega_j r_{k_1}) \qquad \text{and} \qquad \mathbf{T}(k,\ell) = T_{2\ell-2}\left(\frac{r_k}{r_{k_1}}\right), \tag{3.15}$$

where we have scaled ω_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 $\mathcal{O}(L(m_b + n_b))$ time by first applying $\boldsymbol{T}^\top$ then applying $\boldsymbol{C}$.

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

$$J_\nu(\omega r) \sim \sqrt{\frac{2}{\pi\omega r}}\left(\cos(\omega r + \phi)\sum_{\ell=0}^{\infty}\frac{(-1)^\ell a_{2\ell}(\nu)}{(\omega r)^{2\ell}} - \sin(\omega r + \phi)\sum_{\ell=0}^{\infty}\frac{(-1)^\ell a_{2\ell+1}(\nu)}{(\omega r)^{2\ell+1}}\right), \tag{3.16}$$

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

$$a_\ell(\nu) := \frac{(4\nu^2-1)(4\nu^2-3)\cdots(4\nu^2-(2\ell-1)^2)}{\ell!\, 8^\ell}. \tag{3.17}$$

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 [45, section 7.3],

$$\begin{aligned}\left|J_\nu(\omega r) - \sqrt{\frac{2}{\pi}}\sum_{\ell=0}^{M-1}\left[\frac{(-1)^\ell a_{2\ell}(\nu)}{\omega^{2\ell+\frac{1}{2}}}\mathrm{Re}\left(\frac{e^{i(\omega r+\phi)}}{r^{2\ell+\frac{1}{2}}}\right) - \frac{(-1)^\ell a_{2\ell+1}(\nu)}{\omega^{2\ell+\frac{3}{2}}}\mathrm{Im}\left(\frac{e^{i(\omega r+\phi)}}{r^{2\ell+\frac{3}{2}}}\right)\right]\right| \\ \le \sqrt{\frac{2}{\pi}}\left(\frac{|a_{2M}(\nu)|}{z^{2M+\frac{1}{2}}} + \frac{|a_{2M+1}(\nu)|}{z^{2M+\frac{3}{2}}}\right) =: B^{\mathrm{ASY}}_{\nu,M}(z)\end{aligned} \tag{3.18}$$

for all $\omega r \ge z$.

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 $2M$ NUFFTs,

$$\begin{aligned}\mathbf{A}(j_0 : j_1, k_0 : k_1)\,\mathbf{c} \approx \sqrt{\frac{2}{\pi}}\sum_{\ell=0}^{M-1}(-1)^\ell\Bigg[a_{2\ell}(\nu)\mathbf{D}_\omega^{-2\ell-\frac{1}{2}}\mathrm{Re}\left(e^{i\phi}\mathbf{F}\mathbf{D}_r^{-2\ell-\frac{1}{2}}\mathbf{c}\right) \\ - a_{2\ell+1}(\nu)\mathbf{D}_\omega^{-2\ell-\frac{3}{2}}\mathrm{Im}\left(e^{i\phi}\mathbf{F}\mathbf{D}_r^{-2\ell-\frac{3}{2}}\mathbf{c}\right)\Bigg],\end{aligned} \tag{3.19}$$

where $\mathbf{F} \in \mathbb{C}^{(j_1-j_0+1)\times(k_1-k_0+1)}$ is the nonuniform discrete Fourier transform 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 $\mathbf{D}_\omega := \mathrm{diag}(\omega_{j_0}, \ldots, \omega_{j_1})$, and $\mathbf{D}_r := \mathrm{diag}(r_{k_0}, \ldots, r_{k_1})$. For general nonuniform ω_j

and r_k, the NUFFTs required are of type-III, i.e., "nonuniform to nonuniform." If either of $\{\omega_j\}_{j=1}^m$ or $\{r_k\}_{k=1}^n$ are equispaced, then a type-I "nonuniform to uniform" or type-II "uniform to nonuniform" transform can be used, respectively, resulting in an improved prefactor in runtime cost. See [14, 5] for a description of NUFFT transform types.

3.3. Determining order of expansions and crossover point. With these error bounds in hand, we precompute the parameters $z_{\nu,\varepsilon}^M$ and $L_{\nu,\varepsilon}^M$ for tolerances $\varepsilon = 10^{-4}, \ldots, 10^{-15}$, orders $\nu = 1, \ldots, 100$, and number of asymptotic expansion terms $M = 1, \ldots, 20$:

- $z_{\nu,\varepsilon}^M$ such that M-term Hankel expansion of $J_\nu(\omega r)$ is ε-accurate $\forall\ \omega r > z_{\nu,\varepsilon}^M$,
- $L_{\nu,\varepsilon}^M$ such that $L_{\nu,\varepsilon}^M$-term Wimp expansion of $J_\nu(\omega r)$ is ε-accurate $\forall\ \omega r \le z_{\nu,\varepsilon}^M$.

First, the crossover points $z_{\nu,\varepsilon}^M$ are computed using Newton's method on the function $\zeta(z) := B_{\nu,M}^{\mathrm{ASY}}(z) - \varepsilon$. Then the number of local expansion terms $L_{\nu,\varepsilon}^M$ are taken to be the smallest integer such that $B_{\nu,L}^{\mathrm{LOC}}(z_{\nu,\varepsilon}^M) < \varepsilon$. 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 ν we can look up a pair of complementary local and asymptotic expansions with error everywhere bounded by the requested tolerance ε. 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

$$M_{\nu,\varepsilon} = \min\left(\left\lfloor 1 + \frac{\nu}{5} - \frac{\log_{10}(\varepsilon)}{4} \right\rfloor, 20\right). \tag{3.20}$$

4. The NUFHT. 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.

4.1. Subdividing the matrix into blocks by expansion. Having established error bounds which allow us to automatically precompute the number of asymptotic terms M, local terms L, and crossover point z given a tolerance ε and order ν, we now turn to the problem of computing the Hankel transform for a new set of frequencies $\{\omega_j\}_{j=1}^m$ and points $\{r_k\}_{k=1}^n$. We begin by subdividing the corresponding transform matrix $\mathbf{A}$ into three sets of blocks, each of which can be efficiently applied to a vector as described above:

- Local blocks $\mathscr{L} = \{\mathbf{A}(j_0 : j_1, k_0 : k_1) \mid \omega_j r_k \le z\ \forall\ j_0 \le j \le j_1,\ k_0 \le k \le k_1\}$.
- Asymptotic blocks $\mathscr{A} = \{\mathbf{A}(j_0 : j_1, k_0 : k_1) \mid \omega_j r_k > z\ \forall\ j_0 \le j \le j_1,\ k_0 \le k \le k_1\}$.
- Direct blocks $\mathscr{D}$ which are small enough that no fast expansion is needed.

In order to determine a subdivision of $\mathbf{A}$ 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 subblocks 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 $m_b n_b <$

Algorithm 1. Block subdivision of Hankel transform matrix.

```
SUBDIVIDE({r_k}_{k=1}^n, {ω_j}_{j=1}^m, z, min_size):
    L = A = D = ∅
    M = {(1 : m, 1 : n)}
    while M ≠ ∅ do
        Pop a submatrix A(j_0 : j_1, k_0 : k_1) from M
        (j, k) = SPLITINDICES({r_k}_{k=k_0}^{k_1}, {ω_j}_{j=j_0}^{j_1}, z)
        Append A(j_0 : j, k_0 : k) to L
        Append A(j+1 : j_1, k+1 : k_1) to A
        Append A(j_0 : j, k+1 : k_1) to M if (j−j_0+1)(k_1−k) > min_size else D
        Append A(j+1 : j_1, k_0 : k) to M if (j_1−j)(k_1−k+1) > min_size else D
    end
    return (L, A, D)
```

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. See Algorithm 1.

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:

$$(j,k) = \text{SPLITINDICES}(r_1,\dots,r_n,\omega_1,\dots,\omega_m,z) \tag{4.1}$$

$$:= \begin{cases} \underset{j,k\in\mathbb{Z}}{\arg\max} & (j-j_0)(k_1-k)+(j_1-j)(k-k_0) \\ \text{subject to} & j_0 \le j \le j_1, \\ & k_0 \le k \le k_1, \\ & \omega_j r_k \le z. \end{cases} \tag{4.2}$$

This problem can be solved exactly in $\mathcal{O}(j_1-j_0+k_1-k_0)$ 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 of equispaced indices $j \in \{j_0,\dots,j_1\}$, compute the corresponding $k = \arg\max\{k \mid r_k \le \frac{z}{\omega_j}\}$ for each j, and choose (j,k) as the pair which minimizes the objective function of (4.2) among this small collection.

This block subdivision process must be carried out for each new set of inputs $(\{\omega_j\}_{j=1}^m, \{r_k\}_{k=1}^n)$. For repeated transforms on the same points and frequencies, the block partition can be reused. In addition, even when the partition is recomputed for every transform, the empirical and asymptotic costs are significantly lower than those associated with applying blocks of the transform and, therefore, partitioning is never a bottleneck in practice.

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 [5] which we use in our NUFHT implementation—consist of three steps. First, delta masses centered at each nonuniform point are convolved with a *spreading function* which smears them onto a fine

Algorithm 2. Nonuniform fast Hankel transform.

```
NUFHT(ν, ε, {r_k}_{k=1}^n, {c_k}_{k=1}^n, {ω_j}_{j=1}^m):
    g = 0 ∈ C^m
    Choose M_{ν,ε} using (3.20)
    Look up L^M_{ν,ε} and z^M_{ν,ε} from precomputed tables
    Set min_size
    (L, A, D) = SUBDIVIDE({r_k}_{k=1}^n, {ω_j}_{j=1}^m, z^M_{ν,ε}, min_size)
    for B ∈ (L, A, D) do
        for A(j_0 : j_1, k_0 : k_1) ∈ B do
            g(j_0 : j_1) += A(j_0 : j_1, k_0 : k_1) c(k_0 : k_1) using expansion (3.14) for
              L, (3.19) for A, or direct evaluation for D
        end
    end
    return g
```

N-point uniform grid. Then, a standard equispaced FFT is computed on the fine grid. Finally, a diagonal deconvolution 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 [14, 16, 5]. For n points r_k and m frequencies ω_j, spreading the input points to a finer grid is $\mathcal{O}(n)$, the FFT on the finer grid is $\mathcal{O}(N \log N)$, and the global deconvolution at the output frequencies is $\mathcal{O}(m)$. For the type-III NUFFT, the size N of the fine grid typically scales linearly with the space-frequency product $p := (\omega_m - \omega_1)(r_n - r_1)$ [5, 16]. Therefore the total cost of the NUFFT is $\mathcal{O}(n + m + p \log p)$. 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 4.1. *Take* $0 \le \omega_1 < \ldots < \omega_m \le \Omega$ *and* $0 \le r_1 < \ldots < r_n \le R$ *and define the space-frequency product* $p := \Omega R$. *Then the complexity of computing the NUFHT of order* ν *to tolerance* ε *using Algorithm* 2 *is*

$$\mathcal{O}\left((L + M)(m + n) \log \min(n, m) + Mp \log p\right),$$

where L *and* M *are the number of local and asymptotic terms, respectively, chosen according to* ν *and* ε.

Proof. For notational clarity we suppress the dependence of $z^M_{\nu,\varepsilon}$ 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 $\mathcal{O}(L(m + n))$ 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 $\mathcal{O}(M(m + n + p \log p))$ complexity.

Otherwise consider the case where $\mathbf{A}$ contains both local and asymptotic entries. First, note that the number of levels N_{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 ℓ of subdividing every mixed block, we obtain 2^ℓ new mixed blocks, $2^{\ell-1}$ new local blocks, and $2^{\ell-1}$ new asymptotic blocks. Let the local blocks be of size $m_{\ell,b}^{(\mathrm{loc})} \times n_{\ell,b}^{(\mathrm{loc})}$ for $b = 1, \ldots, 2^{\ell-1}$. Then,

$$\sum_{b=1}^{2^{\ell-1}} m_{\ell,b}^{(\mathrm{loc})} \leq m \qquad \text{and} \qquad \sum_{b=1}^{2^{\ell-1}} n_{\ell,b}^{(\mathrm{loc})} \leq n. \tag{4.3}$$

An analogous fact holds for the asymptotic blocks.

Therefore, the total cost of local evaluation is

$$\begin{aligned} \sum_{\ell=1}^{N_{\mathrm{level}}} \sum_{b=1}^{2^{\ell-1}} \mathcal{O}\left(L\left(m_{\ell,b}^{(\mathrm{loc})} + n_{\ell,b}^{(\mathrm{loc})}\right)\right) &= \sum_{\ell=1}^{N_{\mathrm{level}}} \mathcal{O}(L(m+n)) \\ &= \mathcal{O}\left(L(m+n)\log\min(n,m)\right). \end{aligned} \tag{4.4}$$

Let $p_{\ell,b}$ be the space-frequency product of box b at level ℓ. 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 subrectangles of R. Therefore the sum of their areas is bounded by the area of R, so that

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

Then by Hölder's inequality we obtain

$$\sum_{\ell=1}^{N_{\mathrm{level}}} \sum_{b=1}^{2^{\ell-1}} p_{\ell,b} \log p_{\ell,b} \leq \left(\sum_{\ell=1}^{N_{\mathrm{level}}} \sum_{b=1}^{2^{\ell-1}} p_{\ell,b}\right)\left(\max_{\ell,b} \log p_{\ell,b}\right) \leq p \log p. \tag{4.5}$$

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

$$\begin{aligned} &\sum_{\ell=1}^{N_{\mathrm{level}}} \sum_{b=1}^{2^{\ell-1}} \mathcal{O}\left(M\left(m_{\ell,b}^{(\mathrm{asy})} + n_{\ell,b}^{(\mathrm{asy})} + p_{\ell,b}^{(\mathrm{asy})} \log p_{\ell,b}^{(\mathrm{asy})}\right)\right) \\ &\quad = \sum_{\ell=1}^{N_{\mathrm{level}}} \mathcal{O}(M(m+n)) + \sum_{\ell=1}^{N_{\mathrm{level}}} \sum_{b=1}^{2^{\ell-1}} \mathcal{O}\left(M\left(p_{\ell,b}^{(\mathrm{asy})} \log p_{\ell,b}^{(\mathrm{asy})}\right)\right) \\ &\quad = \mathcal{O}\left(M(m+n)\log\min(n,m) + Mp\log p\right). \end{aligned} \tag{4.6}$$

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 ω_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 4.2. *Take* $0 \le \omega_1 < \ldots < \omega_n$ *and* $0 \le r_1 < \ldots < r_n$ *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 4.3. 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 [34, 10.23.2]

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

so that the Hankel transform can be approximated by

$$\sum_{k=1}^{m} c_k J_\nu(\omega_j r_k) \approx \sum_{\ell=-N}^{N} J_{\nu-\ell}(\omega_j r_1) \sum_{k=1}^{m} c_k J_\ell\left(\omega_j (r_k - r_1)\right). \tag{4.7}$$

Each inner sum above can be computed with an NUFHT of order ℓ with space-frequency product $\tilde{p} \ll p$. The outer sum converges rapidly in N for $N > \tilde{p}$, and thus the $\mathcal{O}(\tilde{p}^2 \log \tilde{p})$ cost of evaluating (4.7) may be smaller than the $\mathcal{O}(p \log p)$ cost of using Algorithm 2 directly. The same procedure can be used in ω if $\omega_1 \gg 0$.

Remark 4.4. There exist butterfly factorization-based NUFFT methods that could be used to remove the dependence on the space-frequency product p in Theorem 4.1 using linear algebraic approximations [37]. 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.

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.

5.1. Comparison to direct evaluation. We start by empirically verifying the error analysis in sections 3.1 and 3.2, and the asymptotic scaling analysis in section 4.2 by comparing to direct evaluation of the Hankel transform.

5.1.1. Asymptotic scaling. In order to study the impact of each of the relevant parameters in the scaling analysis of Theorem 4.1 independently, we take n equispaced points r_k in the interval $[0, \sqrt{10^5}]$ and m equispaced frequencies ω_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 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 4.1 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 4.2. 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 ν.

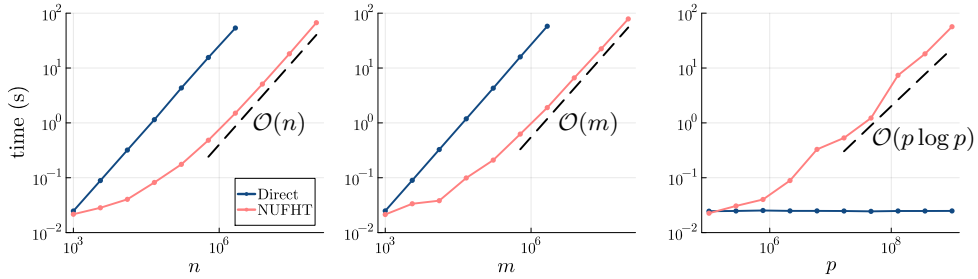

FIG. 3. *Scaling with n, m, and p, respectively, with the other variables held constant. Note: color appears only in the online article.*

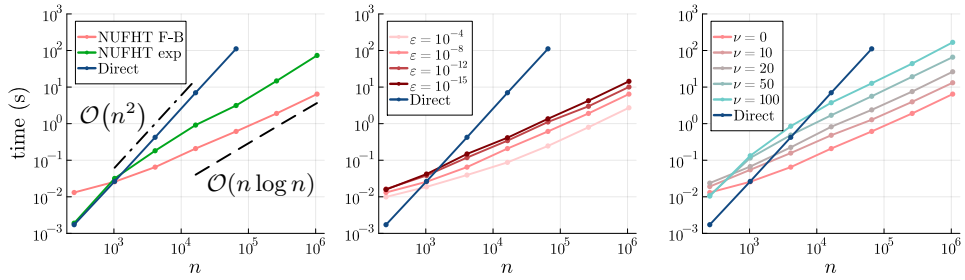

FIG. 4. *Scaling with n for $p = \mathcal{O}(n)$ test cases. In the first plot, we fix $\nu = 0, \varepsilon = 10^{-8}$ and time the NUFHT for both the Fourier–Bessel and exponentially distributed cases. In the second and third plots, we consider the Fourier–Bessel series only, and fix one of the parameters $\nu = 0$ 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. Note: color appears only in the online article.*

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

$$\sqrt{\frac{\Omega z}{R}} = \underset{\frac{z}{R} \leq \omega \leq \Omega}{\arg\max} \, (\Omega - \omega) \left(R - \frac{z}{\omega} \right) \tag{5.1}$$

shows that if we subdivide a block with space frequency product ΩR at a point (ω, 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 ω to be the midpoint 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 4.1, 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 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.

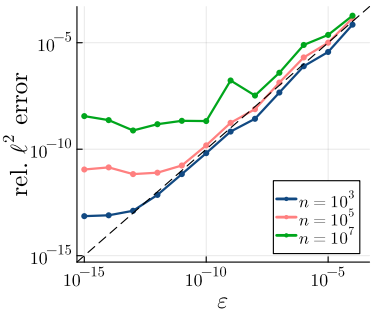

FIG. 5. *Relative 2-norm error* $\|\mathbf{g}-\tilde{\mathbf{g}}\|_2/\|\mathbf{g}\|_2$ *as a function of tolerance* ε *for an NUFHT of order* $\nu = 0$ *for various* n. *Note: color appears only in the online article.*

5.1.2. Impact of the order and tolerance on runtime. As the order ν increases or the tolerance ε decreases, the number of necessary terms L and M in the local and asymptotic expansions, respectively, both grow. From Theorem 4.1, we expect the runtime to grow linearly with $L + M$. Figure 4 shows the runtime of our method for various ε with $\nu = 0$ held constant, as well as for multiple ν's with $\varepsilon = 10^{-8}$ fixed. The $\mathcal{O}(n \log n)$ 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.

5.1.3. Approximation error. Finally, we study the relative error in the output $\mathbf{g}$ as a function of the desired tolerance ε. 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 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 ε down to $\varepsilon = 10^{-14}$ or so. This suggests that the analysis used in section 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 ε, our method gives at most 9 digits 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 [5, Remark 9].

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

$$\hat{f}(\boldsymbol{\omega}) = \int_{\mathbb{R}^d} f(\|\boldsymbol{r}\|)\, e^{i\boldsymbol{\omega}^\top \boldsymbol{r}}\, \mathrm{d}\boldsymbol{r} = \frac{(2\pi)^{\frac{d}{2}}}{\omega^{\frac{d}{2}-1}} \int_0^\infty f(r)\, J_{\frac{d}{2}-1}(\omega r)\, r^{\frac{d}{2}}\, \mathrm{d}r. \tag{5.2}$$

This situation is of particular relevance in spatial statistics and kernel learning problems (with the roles of r and ω reversed) [20, 39, 47]. In this setting, one often constructs a positive definite function $\hat{f}$ to model correlations between data by specifying a nonnegative integrable function f, and computing the integral (5.2) analytically or numerically. See [42, 46] for a detailed exposition of Gaussian process models and their spectral properties.

We consider the isotropic "singular Matérn" model proposed in [38], which we parameterize in two dimensions as

$$f(\boldsymbol{r}) := \|\boldsymbol{r}\|^{-\alpha}(\rho^2 + \|\boldsymbol{r}\|^2)^{-\nu-1} =: |r|^{-\alpha} f_0(r). \tag{5.3}$$

The parameters ρ, ν, and α control the length scale, 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 [6]. In order to evaluate $\hat{f}$ for a large set of $\{\omega_j\}_{j=1}^m$—which in the statistical setting corresponds to kernel evaluation for modern-scale datasets with many observations—a fast transform is imperative.

We take $\alpha = 1.4, \rho = 0.01, \nu = 2$ in (5.3) 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,

$$\begin{aligned} \hat{f}(\omega) &= 2\pi \int_0^1 r^{-\alpha} f_0(r)\, J_0(\omega r)\, \mathrm{d}r \\ &\approx 2\pi \sum_{k=1}^{n} w_k\, f_0(r_k)\, J_0(\omega r_k), \end{aligned} \tag{5.4}$$

doubling the number of nodes m until the error in the computed integral is less than ε. 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 θ 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 ε. We then utilize the two-dimensional (2D) NUFFT to compute the resulting double sum:

$$\hat{f}(\omega) = \frac{1}{4\pi^2} \int_0^{2\pi} \int_0^1 r^{-\alpha} f_0(r)\, e^{-i\omega r \cos\theta}\, \mathrm{d}r\, \mathrm{d}\theta \tag{5.5}$$

$$\approx \frac{1}{4\pi^2} \sum_{k=1}^{n} w_k\, f_0(r_k)\, \frac{2\pi}{t_k} \sum_{s=1}^{t_k} \exp\left\{ -i\omega r_k \cos\left(\frac{2\pi s}{t_k} \right) \right\}. \tag{5.6}$$

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 ω 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 2D transform is often faster than the NUFHT. However, for larger $\omega_{\max}$ the integrands become more oscillatory, and in two dimensions $n = \mathcal{O}(\omega_{\max}^2)$ nodes are needed to resolve these oscillations. Therefore the $\mathcal{O}(n)$ spreading step in the NUFFT becomes prohibitively expensive. However, by using radial symmetry to reduce to a one-dimensional integral, the NUFHT requires only $\mathcal{O}(\omega_{\max})$ quadrature nodes, avoiding the curse of dimensionality. Figure 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.

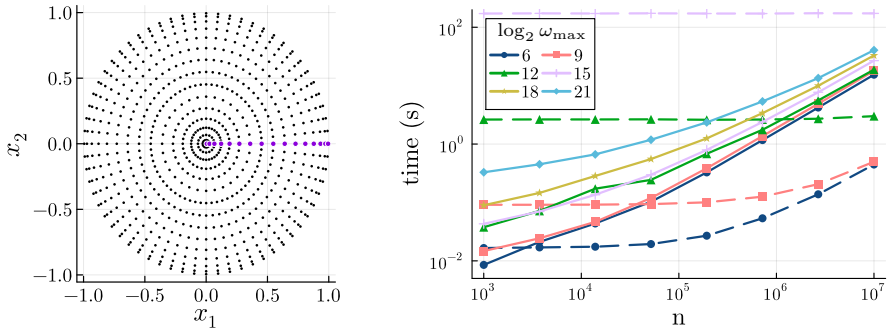

FIG. 6. *Example 2D quadrature nodes for the NUFFT, with one-dimensional radial Gauss–Legendre quadrature on* $[0,1]$ *for the NUFHT emphasized. Runtime comparison between NUFHT and 2D NUFFT for various choices of the maximum frequency* ω_{max} *and the number of evaluation points n. Solid lines indicate the NUFHT, and the corresponding dashed lines indicate the 2D NUFFT. Note: color appears only in the online article.*

5.3. A Helmholtz solver using Fourier–Bessel expansions. Finally, we demonstrate the application of the nonuniform Hankel transform to solving PDEs on the disk using Fourier–Bessel expansions. Consider the following inhomogeneous Helmholtz problem on the unit disk D:

$$\begin{aligned} (\Delta + \kappa^2)u(r,\theta) &= f(r,\theta) && \text{for } r \in [0,1), \quad \theta \in [0,2\pi), \\ u(1,\theta) &= 0 && \text{for } \theta \in [0,2\pi). \end{aligned} \tag{5.7}$$

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

$$\Delta\psi_{j\ell}(r,\theta) = \lambda_{j\ell}\psi_{j\ell}(r,\theta), \tag{5.8}$$

where $\lambda_{j\ell} = -\xi_{\ell,j}^2$ [9, 45]. Therefore, writing the forcing function f and solution u in terms of their respective Fourier–Bessel expansions

$$f(r,\theta) = \sum_{\ell=-\infty}^{\infty}\sum_{j=1}^{\infty} \alpha_{j\ell}\, J_\ell(\xi_{\ell,j} r)\, e^{i\ell\theta}, \qquad u(r,\theta) = \sum_{\ell=-\infty}^{\infty}\sum_{j=1}^{\infty} \beta_{j\ell}\, J_\ell(\xi_{\ell,j} r)\, e^{i\ell\theta}, \tag{5.9}$$

decouples (5.7) into a system of diagonal equations resulting in an explicit formula for the coefficients $\beta_{j\ell}$:

$$\beta_{j\ell} = \frac{\alpha_{j\ell}}{\lambda_{j\ell} + \kappa^2}. \tag{5.10}$$

Due to the orthogonality of the Bessel functions J_ℓ, the Fourier–Bessel coefficients of the forcing f can be computed as

$$\alpha_{j\ell} = \frac{2}{J_{\ell+1}(\xi_{\ell,j})^2} \int_0^{2\pi}\int_0^1 f(r,\theta)\, J_\ell(\xi_{\ell,j} r)\, e^{-i\ell\theta}\, r\, \mathrm{d}r\, \mathrm{d}\theta, \tag{5.11}$$

and the Fourier–Bessel expansion of the solution u can then be written explicitly as

$$u(r,\theta) = \sum_{\ell=-\infty}^{\infty}\sum_{j=1}^{\infty} \frac{\alpha_{j\ell}}{\lambda_{j\ell} + \kappa^2}\, J_\ell(\xi_{\ell,j} r)\, e^{i\ell\theta}. \tag{5.12}$$

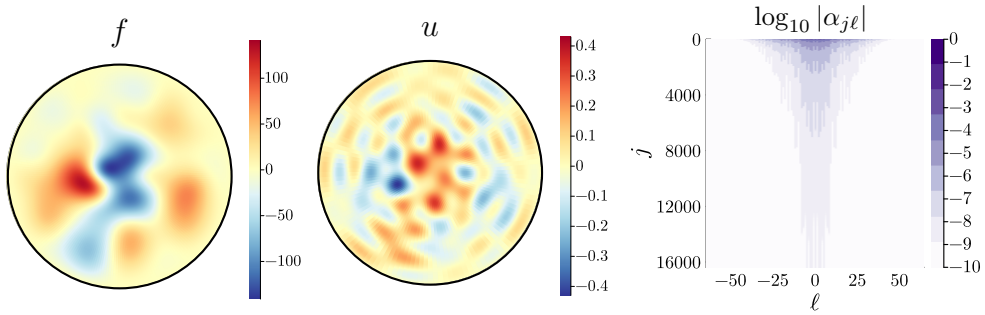

FIG. 7. *Forcing f, solution u, and log magnitude of Fourier–Bessel expansion coefficients $\alpha_{j\ell}$ for* (5.7) *with $\kappa = 25$. Note: color appears only in the online article.*

By diagonalizing the Laplacian, this Fourier–Bessel solver thus provides in the Dirichlet disk setting a direct analogue of commonly used spectral methods on a periodic rectangle which leverage bivariate Fourier expansions, and inherits many of the merits of these 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 $|\alpha_{j\ell}| \to 0$ exponentially fast in both j and ℓ [9]. In addition, solutions for arbitrary κ can be evaluated without additional computations involving f, assuming that κ^2 is not itself a Dirichlet eigenvalue of the Laplacian on D. Therefore, solvers which use Fourier–Bessel expansions are well suited to multifrequency problems.

To compute the Fourier–Bessel coefficients $\alpha_{j\ell}$ of f using (5.11), we use an m-point Gauss–Legendre rule in r and a t-point trapezoidal rule in θ. We iteratively double the number of nodes in each rule until the relative norm difference in computed coefficients between iterations is less than ε (controlling the discretization error) and the relative norm of the coefficients appended in the last iteration is less than ε (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 $\mathcal{O}(tm \log m + mt \log t)$ total complexity. Figure 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 (5.7) computed to relative precision $\varepsilon = 10^{-9}$.

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 $0 \le q \le p-1$, then $|\alpha_{j\ell}| \sim j^{-2p-\frac{1}{2}}$, with exponential convergence only possible if $\Delta^q f(r)|_{r=1} = 0$ for all integers q [9]. 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 ℓ, as demonstrated in Figure 4. As $\alpha_{j\ell}$ decrease spectrally in ℓ for smooth functions f, very large ℓ are not often needed. However, as in any spectral method, functions with sharp features or discontinuous derivatives will yield only algebraic decay in ℓ, requiring more Fourier bases. In such cases the corresponding high order NUFHTs become intractable using the method described here.

6. Discussion. In this manuscript we have presented a fast algorithm for computing DHTs of moderate orders from n nonuniform points to m nonuniform frequencies in $\mathcal{O}((m+n)\log\min(n,m))$ 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 requires minimal 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 paper 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, e.g., high-dimensional statistical settings [28, 33], alternative expansions and asymptotics of J_ν need to be used or derived. This is the focus of ongoing research.

Competing interests. The authors report no competing interests.

Reproducibility of computational results. This paper has been awarded the "SIAM Reproducibility Badge: Code and data available" as a recognition that the authors have followed reproducibility principles valued by SISC and the scientific computing community. Code and data that allow readers to reproduce the results in this paper are available at https://github.com/pbeckman/FastHankelTransform.jl and in the supplementary materials (FastHankelTransform-jl-main.zip [local/web 21.4KB]).

Acknowledgment. The authors would like to thank Alex Barnett for suggesting the use of the Wimp expansion.

REFERENCES

[1] T. S. Alexander, *Adaptive Signal Processing: Theory and Applications*, Springer, New York, 2012.

[2] I. Ali and S. Kalla, *A generalized Hankel transform and its use for solving certain partial differential equations*, ANZIAM J., 41 (1999), pp. 105–117.

[3] B. Alpert, G. Beylkin, D. Gines, and L. Vozovoi, *Adaptive solution of partial differential equations in multiwavelet bases*, J. Comput. Phys., 182 (2002), pp. 149–190, https://doi.org/10.1006/jcph.2002.7160.

[4] T. Askham and A. J. Cerfon, *An adaptive fast multipole accelerated Poisson solver for complex geometries*, J. Comput. Phys., 344 (2017), pp. 1–22, https://doi.org/10.1016/j.jcp.2017.04.063.

[5] A. H. Barnett, J. Magland, and L. af Klinteberg, *A parallel nonuniform fast Fourier transform library based on an "exponential of semicircle" kernel*, SIAM J. Sci. Comput., 41 (2019), pp. C479–C504, https://doi.org/10.1137/18M120885X.

[6] P. G. Beckman and C. J. Geoga, *Fast adaptive Fourier integration for spectral densities of Gaussian processes*, Stat. Comput., 34 (2024), 217, https://doi.org/10.1007/s11222-024-10519-0.

[7] R. Bisseling and R. Kosloff, *The fast Hankel transform as a tool in the solution of the time dependent Schrödinger equation*, J. Comput. Phys., 59 (1985), pp. 136–151.

[8] D. Bondesson, M. J. Schneider, T. Gaass, B. Kühn, G. Bauman, O. Dietrich, and J. Dinkel, *Nonuniform Fourier-decomposition MRI for ventilation-and perfusion-weighted imaging of the lung*, Magn. Reson. Med., 82 (2019), pp. 1312–1321, 10.1002/mrm.27803.

[9] J. P. Boyd and F. 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*, J. Comput. Phys., 230 (2011), pp. 1408–1438.

[10] M. M. Bronstein, A. M. Bronstein, M. Zibulevsky, and H. Azhari, *Reconstruction in diffraction ultrasound tomography using nonuniform FFT*, IEEE Trans. Med. Imaging, 21 (2002), pp. 1395–1401.

[11] J. Brunol and P. Chavel, *Fourier transformation of rotationally invariant two-variable functions: Computer implementation of Hankel transform*, Proc. IEEE, 65 (1977), pp. 1089–1090.
[12] E. Cavanagh and B. Cook, *Numerical evaluation of Hankel transforms via Gaussian-Laguerre polynomial expansions*, IEEE Trans. Acoust. Speech Signal Process., 27 (1979), pp. 361–366.
[13] M. Cree and P. Bones, *Algorithms to numerically evaluate the Hankel transform*, Comput. Math. Appl., 26 (1993), pp. 1–12, https://doi.org/10.1016/0898-1221(93)90081-6.
[14] A. Dutt and V. Rokhlin, *Fast Fourier transforms for nonequispaced data*, SIAM J. Sci. Comput., 14 (1993), pp. 1368–1393, https://doi.org/10.1137/0914081.
[15] M. G. Genton and D. J. Gorsich, *Nonparametric variogram and covariogram estimation with Fourier–Bessel matrices*, Comput. Statist. Data Anal., 41 (2002), pp. 47–57, https://doi.org/10.1016/S0167-9473(02)00062-2.
[16] L. Greengard and J.-Y. Lee, *Accelerating the nonuniform fast Fourier transform*, SIAM Rev., 46 (2004), pp. 443–454, https://doi.org/10.1137/S003614450343200X.
[17] L. Greengard, J.-Y. Lee, and S. Inati, *The fast sinc transform and image reconstruction from nonuniform samples in k-space*, Commun. Appl. Math. Comput. Sci., 1 (2007), pp. 121–131, https://doi.org/10.2140/camcos.2006.1.121.
[18] E. Hansen, *Fast Hankel transform algorithm*, IEEE Trans. Acoust. Speech Signal Process., 33 (1985), pp. 666–671, https://doi.org/10.1109/TASSP.1985.1164579.
[19] W. E. Higgins and D. C. Munson, *A Hankel transform approach to tomographic image reconstruction*, IEEE Trans. Med. Imaging, 7 (1988), pp. 59–72, https://doi.org/10.1109/42.3929.
[20] H. K. Im, M. L. Stein, and Z. Zhu, *Semiparametric estimation of spectral density with irregular observations*, J. Amer. Statist. Assoc., 102 (2007), pp. 726–735, https://doi.org/10.1198/016214507000000220.
[21] S. Jiang and L. Greengard, *A dual-space multilevel kernel-splitting framework for discrete and continuous convolution*, Comm. Pure Appl. Math., 78 (2025), pp. 1086–1143.
[22] H. Johansen and K. Sørensen, *Fast Hankel transforms*, Geophys. Prospect., 27 (1979), pp. 876–901, https://doi.org/10.1111/j.1365-2478.1979.tb01005.x.
[23] H. F. Johnson, *An improved method for computing a discrete Hankel transform*, Comput. Phys. Commun., 43 (1987), pp. 181–202, https://doi.org/10.1016/0010-4655(87)90204-9.
[24] S. Kapur and V. Rokhlin, *An Algorithm for the Fast Hankel Transform*, Technical Report 1045, Computer Science Department, Yale University, New Haven, CT, 1995.
[25] Y. Li, H. Yang, E. R. Martin, K. L. Ho, and L. Ying, *Butterfly factorization*, Multiscale Model. Simul., 13 (2015), pp. 714–732, https://doi.org/10.1137/15M1007173.
[26] C. M. Linton, *Schlömilch series that arise in diffraction theory and their efficient computation*, J. Phys. A, 39 (2006), pp. 3325–3339, https://doi.org/10.1088/0305-4470/39/13/012.
[27] Q. H. Liu and Z. Q. Zhang, *Nonuniform fast Hankel transform (NUFHT) algorithm*, Appl. Optics, 38 (1999), pp. 6705–6708, https://doi.org/10.1364/AO.38.006705.
[28] R. D. Lord, *The use of the Hankel transform in statistics* I. *General theory and examples*, Biometrika, 41 (1954), pp. 44–55.
[29] R. D. Lord, *The use of the Hankel transform in statistics* II. *Methods of computation*, Biometrika, 41 (1954), pp. 344–350.
[30] N. F. Marshall, O. Mickelin, and A. Singer, *Fast expansion into harmonics on the disk: A steerable basis with fast radial convolutions*, SIAM J. Sci. Comput., 45 (2023), pp. A2431–A2457, https://doi.org/10.1137/22M1542775.
[31] D. Mook, *An algorithm for the numerical evaluation of the Hankel and Abel transforms*, IEEE Trans. Acoust. Speech Signal Process., 31 (1983), pp. 979–985, https://doi.org/10.1109/TASSP.1983.1164161.
[32] R. H. Nochetto, K. G. Siebert, and A. Veeser, *Theory of adaptive finite element methods: An introduction*, in Multiscale, Nonlinear and Adaptive Approximation: Dedicated to Wolfgang Dahmen on the Occasion of his 60th Birthday, Springer, Berlin, 2009, pp. 409–542.
[33] J. P. Nolan, *Multivariate elliptically contoured stable distributions: Theory and estimation*, Comput. Statist., 28 (2013), pp. 2067–2089, https://doi.org/10.1007/s00180-013-0396-7.
[34] F. W. Olver, *NIST Handbook of Mathematical Functions*, Cambridge University Press, Cambridge, 2010.
[35] M. O'Neil, F. Woolfe, and V. Rokhlin, *An algorithm for the rapid evaluation of special function transforms*, Appl. Comput. Harmon. Anal., 28 (2010), pp. 203–226, https://doi.org/10.1016/j.acha.2009.08.005.

[36] A. V. Oppenheim, G. V. Frisk, and D. R. Martinez, *Computation of the Hankel transform using projections*, J. Acoust. Soc. Amer., 68 (1980), pp. 523–529, https://doi.org/10.1121/1.384765.

[37] Q. Pang, K. L. Ho, and H. Yang, *Interpolative decomposition butterfly factorization*, SIAM J. Sci. Comput., 42 (2020), pp. A1097–A1115, https://doi.org/10.1137/19M1294873.

[38] E. Porcu and M. 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, Berlin, 2012, pp. 221–238.

[39] A. Rahimi and B. Recht, *Random features for large-scale kernel machines*, Adv. Neural Inform. Process. Syst., 20 (2007), pp. 1177–1184.

[40] A. Rangan, M. Spivak, J. Andén, and A. Barnett, *Factorization of the translation kernel for fast rigid image alignment*, Inverse Problems, 36 (2020), 024001, https://doi.org/10.1088/1361-6420/ab4e66.

[41] A. Siegman, *Quasi fast Hankel transform*, Opt. Lett., 1 (1977), pp. 13–15, https://doi.org/10.1364/OL.1.000013.

[42] M. L. Stein, *Interpolation of Spatial Data: Some Theory for Kriging*, Springer, New York, 1999.

[43] G. Thakur and H.-T. Wu, *Synchrosqueezing-based recovery of instantaneous frequency from nonuniform samples*, SIAM J. Math. Anal., 43 (2011), pp. 2078–2095, https://doi.org/10.1137/100798818.

[44] A. Townsend, *A fast analysis-based discrete Hankel transform using asymptotic expansions*, SIAM J. Numer. Anal., 53 (2015), pp. 1897–1917, https://doi.org/10.1137/151003106.

[45] G. N. Watson, *A Treatise on the Theory of Bessel Functions*, Vol. 2, Cambridge University Press, Cambridge, 1922.

[46] C. K. Williams and C. E. Rasmussen, *Gaussian Processes for Machine Learning*, Vol. 2, MIT Press, Cambridge, MA, 2006.

[47] A. Wilson and R. Adams, *Gaussian process kernels for pattern discovery and extrapolation*, Proc. Mach. Learn. Res. (PMLR), 28 (2013), pp. 1067–1075.

[48] J. Wimp, *Polynomial expansions of Bessel functions and some associated functions*, Math. Comp. 16 (1962), pp. 446–458, https://doi.org/10.1090/S0025-5718-1962-0148956-3.

[49] Z. Zhao and A. Singer, *Fourier–Bessel rotational invariant eigenimages*, J. Opt. Soc. Amer. A, 30 (2013), pp. 871–877, https://doi.org/10.1364/JOSAA.30.000871.

[50] R. Zhou and N. Grisouard, *Spectral Solver for Cauchy Problems in Polar Coordinates Using Discrete Hankel Transforms*, preprint, arXiv:2210.09736, 2022.