A FAST ANALYSIS-BASED DISCRETE HANKEL TRANSFORM USING ASYMPTOTIC EXPANSIONS

ALEX TOWNSEN*

Abstract. A fast and numerically stable algorithm is described for computing the discrete Hankel transform of order 0 as well as evaluating Schlömilch and Fourier–Bessel expansions in $\mathcal{O}(N(\log N)^2 / \log\log N)$ operations. The algorithm is based on an asymptotic expansion for Bessel functions of large arguments, the fast Fourier transform, and the Neumann addition formula. All the algorithmic parameters are selected from error bounds to achieve a near-optimal computational cost for any accuracy goal. Numerical results demonstrate the efficiency of the resulting algorithm.

Key words. Hankel transform, Bessel functions, asymptotic expansions, fast Fourier transform

AMS subject classifications. 65R10, 33C10

1. Introduction. The Hankel transform of order $\nu \geq 0$ of a function $f : [0,1] \to \mathbb{C}$ is defined as [21, Chap. 9]

$$F_\nu(\omega) = \int_0^1 f(r) J_\nu(r\omega) r \, dr, \qquad \omega > 0, \tag{1.1}$$

where J_ν is the Bessel function of the first kind with parameter ν. When $\nu = 0$, (1.1) is essentially the two-dimensional Fourier transform of a radially symmetric function with support on the unit disc and appears in the Fourier–Hankel–Abel cycle [21, Sec. 9.3]. Furthermore, the Hankel transform of integer order $\nu \geq 1$ is equivalent to the Fourier transform of functions of the form $e^{i\nu\theta} f(r)$, where $\theta \in [0, 2\pi)$. Throughout this paper we take ν to be an integer.

Various semi-discrete and discrete variants of (1.1) are employed in optics [7], electromagnetics [17], medical imaging [13], and the numerical solution of partial differential equations [3]. At some stage these require the computation of the following sums:

$$f_k = \sum_{n=1}^N c_n J_\nu(r_k \omega_n), \qquad 1 \leq k \leq N, \tag{1.2}$$

where ν is an integer, $0 < \omega_1 < \cdots < \omega_n < \infty$, and $0 \leq r_1 < \cdots < r_N \leq 1$. Special cases of (1.2) that are considered in this paper are: (1) The evaluation of Schlömilch expansions at r_k [23] ($\omega_n = n\pi$ or $\omega_n = (n + \nu)\pi$); (2) The evaluation of Fourier–Bessel expansions of order 0 at r_k ($\nu = 0$ and $\omega_n = j_{0,n}$, where $j_{0,n}$ is the nth positive root of J_0); and (3) The discrete Hankel transform of order 0 [15] ($\nu = 0$, $\omega_n = j_{0,n}$, and $r_k = j_{0,k}/j_{0,N+1}$). For algorithms that rely on the fast Fourier transform (FFT), such as the algorithm described in this paper, tasks (1), (2), and (3) are incrementally more difficult. We consider them in turn in sections 4, 5, and 6, respectively.

Our algorithm is based on carefully replacing $J_\nu(z)$ by an asymptotic expansion for large arguments that, up to a certain error, expresses $J_\nu(z)$ as a weighted sum of trigonometric functions. This allows a significant proportion of the computation involved in (1.2) to be achieved by the discrete cosine transform of type I (DCT)

---

*Department of Mathematics, Massachusetts Institute of Technology, 77 Massachusetts Avenue Cambridge, MA 02139-4307. (ajt@mit.edu)

and the discrete sine transform of type I (DST). These are $\mathcal{O}(N \log N)$ FFT-based algorithms for computing matrix-vector products with the matrices C_N^I and S_N^I, where

$$(C_N^I)_{kn} = \cos\left(\frac{(k-1)(n-1)\pi}{N-1}\right), \qquad (S_N^I)_{kn} = \sin\left(\frac{kn\pi}{N+1}\right), \qquad 1 \leq k, n \leq N.$$
(1.3)

Asymptotic expansions of special functions are notoriously tricky to use in practice because it is often difficult to determine the precise regime for which they are accurate approximations. However, for transforms involving orthogonal polynomials, such as the Chebyshev–Legendre transform [8, 16], successful fast algorithms have been derived based on asymptotic expansions. At the point of evaluation, these algorithms are competitive with multipole-like approaches [1], but have the advantage of no precomputation. This allows for more adaptive algorithms such as a convolution algorithm for Chebyshev expansions [9].

In this paper we will use error bounds to determine the precise regime for which an asymptotic expansion of Bessel functions is an accurate approximation and from there derive all the algorithmic parameters to achieve a near-optimal computational cost for any accuracy goal (see Table 4.1). For each task (1)-(3) there are just two user-defined inputs: A vector of coefficients $c_1, \ldots, c_N$ in (1.2) and a working accuracy $\epsilon > 0$. The values of f_k in (1.2) are then calculated to an accuracy of $\mathcal{O}(\epsilon \sum_{n=1}^{N} |c_n|)$. The resulting algorithm has a complexity of $\mathcal{O}(N(\log N)^2 \log(1/\epsilon)^p / \log\log N)$, where $p = 1$ for task (1), $p = 2$ for task (2), and $p = 3$ for task (3). For the majority of this paper we state algorithmic complexities without the dependency on ϵ.

Previous algorithms based on asymptotic expansions for computing (1.2) [4, 24] have either a poor accuracy or a dubious numerical stability [5]. Here, we find that once an asymptotic expansion of Bessel functions is carefully employed the resulting algorithm is numerically stable, can be adapted to any accuracy goal, and requires no precomputation. Moreover, we show how the equally-spaced restriction, which is inherited from the reliance on DCTs and DSTs, can be alleviated.

We use the following notation. A column vector with entries $v_1, \ldots, v_N$ is denoted by $\underline{v}$, a row vector is denoted by $\underline{v}^{\mathsf{T}}$, a diagonal matrix with diagonal entries $v_1, \ldots, v_N$ is written as $D_{\underline{v}}$, and the $N \times N$ matrix with (k, n) entry $J_\nu(a_{kn})$ is denoted by $\mathbf{J}_\nu(A)$ where $A = (a_{kn})_{1 \leq k, n \leq N}$.

The paper is organized as follows. In Section 2 we briefly discuss existing methods for computing (1.2) and in Section 3 we investigate the approximation power of three expansions for Bessel functions. In Section 4 we first describe an $\mathcal{O}(N^{3/2})$ algorithm for evaluating Schlömilch expansions before deriving a faster $\mathcal{O}(N(\log N)^2 / \log\log N)$ algorithm. In Sections 5 and 6 we use the Neumann addition formula to extend the algorithm to the evaluation of Fourier–Bessel expansions and the computation of the discrete Hankel transform.

2. Existing methods. We now give a brief description of some existing methods for evaluating (1.2) that roughly fall into four categories: (1) Direct summation; (2) Evaluation via an integral representation; (3) Evaluation using asymptotic expansions, and; (4) Butterfly schemes. For other approaches see [5].

2.1. Direct summation. One of the simplest ideas is to evaluate $J_\nu(r_k \omega_n)$ for $1 \leq k, n \leq N$ and then naively compute the sums in (1.2). By using the Taylor series expansion in (3.3) for small z, the asymptotic expansion in (3.1) for large z, and Miller's algorithm [18, Sec. 3.6(vi)] (backward recursion with a three-term recurrence) otherwise, each evaluation of $J_\nu(z)$ costs $\mathcal{O}(1)$ operations [2]. Therefore,

this simple algorithm requires a total of $\mathcal{O}(N^2)$ operations and is very efficient for small N ($N \leq 256$). However, the $\mathcal{O}(N^2)$ cost is prohibitive for large N.

Note that in general there is no evaluation scheme for $J_\nu(z)$ that is exact in infinite precision arithmetic. At some level a working accuracy of $\epsilon > 0$ must be introduced. Thus, even in the direct summation approach, the relaxation of the working accuracy that is necessary for deriving a faster algorithm, has already been made.

The sums in (1.2) are equivalent to the matrix-vector product $\mathbf{J}_\nu(\underline{r}\,\underline{\omega}^\mathsf{T})\underline{c}$. One reason that an $\mathcal{O}(N(\log N)^2/\log\log N)$ algorithm is possible is because the argument, $\underline{r}\,\underline{\omega}^\mathsf{T}$, is a rank 1 matrix. A fact that we will exploit several times.

2.2. Evaluation via an integral representation. When ν is an integer, $J_\nu(z)$ can be replaced by its integral representation [18, (10.9.2)],

$$J_\nu(z) = \frac{\mathrm{i}^{-\nu}}{\pi} \int_0^\pi e^{\mathrm{i} z \cos(\theta)} \cos(\nu\theta) d\theta = \frac{\mathrm{i}^{-\nu}}{2\pi} \int_0^{2\pi} e^{\mathrm{i} z \cos(\theta)} \cos(\nu\theta) d\theta, \qquad (2.1)$$

where $\mathrm{i} = \sqrt{-1}$ is the imaginary unit and the last equality holds because $e^{\mathrm{i} z \cos(\theta)} \cos(\nu\theta)$ is even about $t = \pi$. The integrand in (2.1) is 2π-periodic and the integral can be approximated by a K-point trapezium rule. Substituting this approximation into (1.2) gives a double sum for $\underline{f}$ that can be evaluated with one FFT followed by an interpolation step in $\mathcal{O}(KN + N \log N)$ operations [6]. The major issue with this approach is that the integrand in (2.1) is highly oscillatory for large z and often $K \approx N$, resulting in $\mathcal{O}(N^2)$ operations.

2.3. Evaluation using asymptotic expansions. One can replace $J_\nu(z)$ in (1.2) by an asymptotic expansion that involves trigonometric functions. A fast, but perhaps numerically unstable (see Figure 4.1), algorithm then follows by exploiting DCTs and DSTs. Existing work has approximated $J_\nu(z)$ using asymptotic expansions with rational coefficients [20], the approximation $\sqrt{2/(\pi z)}\cos(z - (2\nu + 1)\pi/4)$ [4], and for half-integer ν the asymptotic expansion in (3.1) [24]. However, none of these approaches have rigorously determined the regime in which the employed asymptotic expansion is an accurate approximation and instead involve dubiously chosen algorithmic parameters. In this paper we will use a well-established asymptotic expansion and rigorously determine the regime in which it is an accurate approximation.

2.4. A fast multipole method approach. In 1995, Kapur and Rokhlin described an algorithm for computing the integral in (1.1) when $\nu = 0$ [14]. The main idea is to write (1.1) as a product of two integrals as follows [14, (6)]:

$$\int_0^1 f(r) J_0(r\omega) r\, dr = \frac{1}{\pi} \int_{-\omega}^\omega \frac{1}{\sqrt{\omega^2 - u^2}} \int_0^1 r f(r) \cos(ru)\, dr\, du.$$

The inner integral can be discretized by a (corrected) trapezium rule and computed via a discrete cosine transform, while the outer integral involves a singular kernel and can be computed with the fast multipole method. As described in [14] the algorithm can be used to evaluate Schlömilch expansions in $\mathcal{O}(N \log N)$ operations (see Section 4). With modern advances in fast algorithms for nonuniform transforms, it could now be adapted to the computation of the discrete Hankel transform though we are not aware of such work.

2.5. Butterfly schemes. Recently, a new algorithm was developed for a class of fast transforms involving special functions [19]. The algorithm compresses the ranks

of submatrices of a change of basis matrix by constructing skeleton decompositions in a hierarchical fashion. This hierarchical decomposition of the matrix means that it can then be applied to a vector in $\mathcal{O}(N \log N)$ operations. The compression step of the algorithm is the dominating cost and is observed to require $\mathcal{O}(N^2)$ operations [19, Table. 6]. Fortunately, in many applications the compression step can be regarded as a precomputational cost because it is independent of the values of $c_1, \ldots, c_N$ in (1.2), though it does depend on N.

The methodology behind butterfly schemes is more general than what we present in this paper as it does not require knowledge of asymptotics. Here, we will derive a fast transform that has no precomputational cost and is better suited to applications where N is not known in advance.

Some of the ideas in the spherical harmonic transform literature [22, 26] may be useful for reducing the precomputational cost in [19], though we are not aware of work in this direction.

3. Three expansions of Bessel functions.
In this section we investigate three known expansions of Bessel functions [18, Chap. 10]: (1) An explicit asymptotic expansion for large arguments; (2) A convergent Taylor series expansion for small arguments; and (3) The Neumann addition formula for perturbed arguments. Mathematically, these three expansions are infinite series, but for practical use they must be truncated. Here, we will focus on deriving criteria to ensure that the errors introduced by such truncations are negligible. Related techniques were successfully used to derive a fast algorithm for the discrete Legendre transform [10].

3.1. An asymptotic expansion of Bessel functions for large arguments.
The graph of $J_\nu(z)$ on $[0, \infty)$ is often said to look like an oscillating trigonometric function that decays like $1/\sqrt{z}$. This observation can be made precise by an asymptotic expansion of $J_\nu(z)$ for large z.

Let ν and $M \geq 1$ be integers and $z > 0$. The first $2M$ terms of the Hankel asymptotic expansion of $J_\nu(z)$ is given by [18, (10.17.3)] (also see [11]):

$$J_\nu(z) = \left(\frac{2}{\pi z}\right)^{\frac{1}{2}} \left(\cos(\mu)\sum_{m=0}^{M-1}(-1)^m\frac{a_{2m}(\nu)}{z^{2m}} - \sin(\mu)\sum_{m=0}^{M-1}(-1)^m\frac{a_{2m+1}(\nu)}{z^{2m+1}}\right) + R_{\nu,M}(z),$$
(3.1)

where $\mu = z - (2\nu + 1)\pi/4$, $|R_{\nu,M}(z)|$ is the error term, $a_0(\nu) = 1$, and

$$a_m(\nu) = \frac{(4\nu^2 - 1^2)(4\nu^2 - 3^2)\cdots(4\nu^2 - (2m-1)^2)}{m!8^m}, \qquad m \geq 1.$$

The first term in (3.1) shows that the leading asymptotic behavior of $J_\nu(z)$ as $z \to \infty$ is $\sqrt{2/(\pi z)}\cos(\mu)$, which is an oscillating trigonometric function that decays like $1/\sqrt{z}$. The first $2M$ terms show that, up to an error of $|R_{\nu,M}(z)|$, $J_\nu(z)$ can be written as a weighted sum of $\cos(\mu)z^{-2m-1/2}$ and $\sin(\mu)z^{-2m-3/2}$ for $0 \leq m \leq M-1$. Figure 3.1 (left) shows the error term $|R_{0,M}(z)|$ on $z \in (0, 50]$ for $M = 1, 2, 3, 4, 5, 7, 10$. As expected, the asymptotic expansion becomes more accurate as z increases. In particular, for sufficiently large z the error term is negligible, i.e., $|R_{\nu,M}(z)| \leq \epsilon$.

It is important to appreciate that (3.1) is an asymptotic expansion, as opposed to a convergent series, and does not converge pointwise to $J_\nu(z)$ as $M \to \infty$. In practice, increasing M will eventually be detrimental and lead to severe numerical overflow issues. Figure 3.1 (right) shows the error $|R_{\nu,M}(10)|$ as $M \to \infty$ and $\nu = 0, 5, 10, 20$.
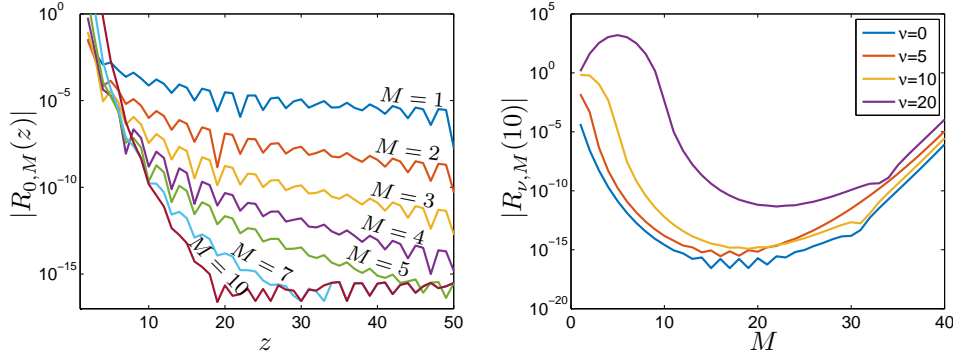
Fig. 3.1: Left: The absolute value of $R_{0,M}(z)$ in (3.1) on $z \in (0, 50]$ for $M = 1, 2, 3, 4, 5, 7, 10$. For any $M \geq 1$ and sufficiently large z, the asymptotic expansion is an accurate approximation. Right: The absolute value of $R_{\nu,M}(10)$ for $1 \leq M \leq 40$ and $\nu = 0, 5, 10, 20$. For a fixed z, increasing M may reduce the accuracy of the asymptotic expansion and will eventually lead to numerical overflow issues.

Thus, the appropriate use of (3.1) is to select an $M \geq 1$ and ask: "For what sufficiently large z is the asymptotic expansion accurate?". For example, from Figure 3.1 (left) we observe that for $M = 10$ we have $|R_{0,10}(z)| \leq 10^{-15}$ for any $z > 17.8$ and hence, we may safely replace $J_0(z)$ by a weighted sum of $\cos(\mu)z^{-2m-1/2}$ and $\sin(\mu)z^{-2m-3/2}$ for $0 \leq m \leq 9$ when $z > 17.8$.

More generally, it is known that the error term $|R_{\nu,M}(z)|$ is bounded by the size of the first neglected terms [18, 10.17(iii)]. Since $|\cos(\omega)| \leq 1$ and $|\sin(\omega)| \leq 1$ we have

$$|R_{\nu,M}(z)| \leq \left(\frac{2}{\pi z}\right)^{\frac{1}{2}} \left(\frac{|a_{2M}(\nu)|}{z^{2M}} + \frac{|a_{2M+1}(\nu)|}{z^{2M+1}}\right). \tag{3.2}$$

This is essentially a sharp bound since for any $c > \sqrt{2}$ there is a $z > 0$ such that $c|R_{\nu,M}(z)|$ is larger than the magnitude of the first neglected terms.[1]

Let $s_{\nu,M}(\epsilon)$ be the smallest real number so that if $z \geq s_{\nu,M}(\epsilon)$ then $|R_{\nu,M}(z)| \leq \epsilon$. From (3.2) we can take $s_{\nu,M}(\epsilon)$ as the number that solves

$$\left(\frac{2}{\pi z}\right)^{\frac{1}{2}} \left(\frac{|a_{2M}(\nu)|}{z^{2M}} + \frac{|a_{2M+1}(\nu)|}{z^{2M+1}}\right) = \epsilon.$$

In general, the equation above has no closed-form solution, but can be solved numerically by a fixed-point iteration. We start with the initial guess of $s_{\nu,M}^{(0)}(\epsilon) = 1$ and iterate as follows:

$$s_{\nu,M}^{(j+1)}(\epsilon) = \left(\frac{\sqrt{2}\left(|a_{2M}(\nu)| + |a_{2M+1}(\nu)|/s_{\nu,M}^{(j)}(\epsilon)\right)}{\sqrt{\pi}\epsilon}\right)^{\frac{1}{2M+\frac{1}{2}}}, \qquad j \geq 0.$$

We terminate this after four iterations and take $s_{\nu,M}(\epsilon) \approx s_{\nu,M}^{(4)}(\epsilon)$. Table 3.1 gives the calculated values of $s_{\nu,M}(10^{-15})$ for $3 \leq M \leq 12$ and $\nu = 0, 1, 2, 10$.

---

[1]To see this pick z of the form $(k/4 + \nu/2 + 1/4)\pi$ for sufficiently large k.

5

M	3	4	5	6	7	8	9	10	11	12
$s_{0,M}$	180.5	70.5	41.5	30.0	24.3	21.1	19.1	17.8	17.0	16.5
$s_{1,M}$	185.2	71.5	41.9	30.2	24.4	21.1	19.2	17.9	17.1	16.5
$s_{2,M}$	200.2	74.8	43.1	30.8	24.8	21.4	19.3	18.0	17.2	16.6
$s_{10,M}$	2330.7	500.0	149.0	64.6	41.4	31.4	26.0	22.9	20.9	19.6

Table 3.1: The asymptotic expansion (3.1) is accurate up to an error of ϵ for any $z \geq s_{\nu,M}(\epsilon)$. Here, the values of $s_{\nu,M}(10^{-15})$ are tabulated for $3 \leq M \leq 12$ and $\nu = 0, 1, 2, 10$.

3.2. The Taylor series expansion of Bessel functions for small arguments. For an integer ν, the Taylor series expansion of $J_\nu(z)$ about $z = 0$ is given by [18, (10.2.2)]

$$J_\nu(z) = \sum_{t=0}^{\infty} \frac{(-1)^t 2^{-2t-\nu}}{t!(t+\nu)!} z^{2t+\nu}. \tag{3.3}$$

In contrast to (3.1), the Taylor series expansion converges pointwise to $J_\nu(z)$ for any z. However, for practical application the infinite series in (3.3) must still be truncated.

Let $T \geq 1$ be an integer and consider the truncated Taylor series expansion that is given by

$$J_\nu(z) = \sum_{t=0}^{T-1} \frac{(-1)^t 2^{-2t-\nu}}{t!(t+\nu)!} z^{2t+\nu} + E_{\nu,T}(z), \tag{3.4}$$

where $|E_{\nu,T}(z)|$ is the error term. As $z \to 0$ the leading asymptotic behavior of $|E_{\nu,T}(z)|$ matches the order of the first neglected term and hence, $\lim_{z\to 0} |E_{\nu,T}(z)| = 0$. In particular, there is a real number $t_{\nu,T}(\epsilon)$ so that if $z \leq t_{\nu,T}(\epsilon)$ then $|E_{\nu,T}(z)| \leq \epsilon$.

To calculate the parameter $t_{\nu,T}(\epsilon)$ we solve the equation $E_{\nu,T}(z) = \epsilon$. Remarkably, an explicit closed-form expression for $E_{\nu,T}(z)$ is known [27] and is given by

$$E_{\nu,T}(z) = \frac{(-1)^T 2^{-2T-\nu} z^{2T+\nu}}{(T+\nu)!T!} {}_1F_2\left(\begin{matrix} 1 \\ T+1 \quad T+\nu+1 \end{matrix}; -\frac{z^2}{4}\right) \approx \frac{(-1)^T 2^{-2T-\nu} z^{2T+\nu}}{(T+\nu)!T!},$$

where ${}_1F_2$ is the generalized hypergeometric function that we approximate by 1 since we are considering z to be small [18, (16.2.1)]. Solving $E_{\nu,T}(z) = \epsilon$ we find that

$$t_{\nu,T}(\epsilon) \approx \left(2^{2T+\nu}(T+\nu)!T!\epsilon\right)^{\frac{1}{2T+\nu}}.$$

Table 3.2 gives the calculated values of $t_{\nu,T}(10^{-15})$ for $1 \leq T \leq 9$ and $\nu = 0, 1, 2, 10$.

3.3. The Neumann addition formula for perturbed arguments. The Neumann addition formula expresses $J_\nu(z + \delta z)$ as an infinite sum of products of Bessel functions of the form $J_{\nu-s}(z)J_s(\delta z)$ for $s \in \mathbb{Z}$. It is given by [18, (10.23.2)]

$$J_\nu(z + \delta z) = \sum_{s=-\infty}^{\infty} J_{\nu-s}(z)J_s(\delta z). \tag{3.5}$$

T	1	2	3	4	5	6	7	8	9
$t_{0,T}$	6×10^{-8}	0.001	0.011	0.059	0.165	0.337	0.573	0.869	1.217
$t_{1,T}$	3×10^{-5}	0.003	0.029	0.104	0.243	0.449	0.716	1.039	1.411
$t_{2,T}$	0.001	0.012	0.061	0.168	0.341	0.579	0.876	1.225	1.618
$t_{10,T}$	0.484	0.743	1.058	1.420	1.823	2.262	2.733	3.230	3.750

Table 3.2: The truncated Taylor series expansion (3.4) is accurate up to an error of ϵ for any $z \le t_{\nu,T}(\epsilon)$. Here, the approximate values of $t_{\nu,T}(10^{-15})$ are tabulated for $1 \le T \le 9$ and $\nu = 0, 1, 2, 10$.

Here, we wish to truncate (3.5) and use it as a numerical approximation to $J_\nu(z+\delta z)$. Fortunately, when $|\delta z|$ is small (so that $z + \delta z$ can be considered as a perturbation of z) the Neumann addition formula is a rapidly converging series for $J_\nu(z + \delta z)$.

LEMMA 3.1. *Let ν be an integer, $z > 0$, and $|\delta z| < e^{-1}$. Then, for $K \ge 1$ we have*

$$\left| J_\nu(z+\delta z) - \sum_{s=-K+1}^{K-1} J_{\nu-s}(z)J_s(\delta z) \right| \le 5.2 \left(\frac{e\,|\delta z|}{2} \right)^K, \tag{3.6}$$

where $e \approx 2.71828$ is Euler's number.

Proof. Let ν be an integer, $z > 0$, $|\delta z| < e^{-1}$, and $K \ge 1$. Denote the left-hand side of the inequality in (3.6) by $|A_{\nu,K}(z,\delta z)|$ so that

$$A_{\nu,K}(z,\delta z) = \sum_{s=-\infty}^{-K} J_{\nu-s}(z)J_s(\delta z) + \sum_{s=K}^{\infty} J_{\nu-s}(z)J_s(\delta z).$$

Since $|J_\nu(z)| \le 1$ [18, (10.14.1)] and $J_{-\nu}(z) = (-1)^\nu J_\nu(z)$ [18, (10.4.1)], we have

$$\left| A_{\nu,K}(z,\delta z) \right| \le \sum_{s=-\infty}^{-K} |J_s(\delta z)| + \sum_{s=K}^{\infty} |J_s(\delta z)| \le 2 \sum_{s=K}^{\infty} |J_s(\delta z)|. \tag{3.7}$$

By Kapteyn's inequality [18, (10.14.8)] we can bound $|J_s(\delta z)|$ from above as follows:

$$|J_s(\delta z)| = |J_s(s(\delta z/s))| \le \frac{|\delta z/s|^s \, e^{s(1-|\delta z/s|^2)^{\frac{1}{2}}}}{\left(1 + (1 - |\delta z/s|^2)^{\frac{1}{2}} \right)^s} \le \left(\frac{e\,|\delta z|}{2s} \right)^s, \tag{3.8}$$

where the last inequality comes from $e^x \le e(1+x)/2$ for $0 \le x \le 1$. Therefore, by substituting (3.8) into (3.7) we find that

$$\left| A_{\nu,K}(z,\delta z) \right| \le 2 \sum_{s=K}^{\infty} \left(\frac{e\,|\delta z|}{2s} \right)^s \le 2 \sum_{s=K}^{\infty} s^{-s} \sum_{s=K}^{\infty} \left(\frac{e\,|\delta z|}{2} \right)^s \le 5.2 \left(\frac{e\,|\delta z|}{2} \right)^K,$$

where we used $\sum_{s=1}^{\infty} s^{-s} \le 1.3$ and $\sum_{s=0}^{\infty} (e|\delta z|/2)^s = 1/(1 - e|\delta z|/2) \le 2$. $\square$

Lemma 3.1 shows that $\sum_{s=-K+1}^{K-1} J_{\nu-s}(z)J_s(\delta z)$ approximates $J_\nu(z+\delta z)$, up to an error of ϵ, provided that $5.2(e|\delta z|/2)^K \le \epsilon$. Equivalently, for any

$$|\delta z| \le \frac{2}{e} \left(\frac{\epsilon}{5.2} \right)^{\frac{1}{K}}. \tag{3.9}$$

K	3	4	5	6	7	8	9	10
(3.9)	4×10^{-6}	9×10^{-5}	0.001	0.002	0.004	0.008	0.013	0.020

Table 3.3: Lemma 3.1 shows that $\sum_{s=-K+1}^{K-1} J_{\nu-s}(z)J_s(\delta z)$ is an approximation to $J_\nu(z+\delta z)$, up to an error of ϵ, for any $|\delta z| \le 2(\epsilon/5.2)^{1/K}/e$. Here, the values of $2(\epsilon/5.2)^{1/K}/e$ are tabulated for $\epsilon = 10^{-15}$ and $3 \le K \le 10$.

Table 3.3 gives the values of $2(\epsilon/5.2)^{1/K}/e$ for $\epsilon = 10^{-15}$.

At first it may not be clear why Lemma 3.1 is useful for practical computations since a single evaluation of $J_\nu(z+\delta z)$ is replaced by a seemingly trickier sum of products of Bessel functions. However, in sections 5 and 6 the truncated Neumann addition formula will be the approximation that will allow us to evaluate Fourier–Bessel expansions and compute the discrete Hankel transform. The crucial observation is that the positive roots of $J_0(z)$ can be regarded as a perturbed equally-spaced grid.

3.3.1. The roots of $\mathbf{J_0(z)}$ as a perturbed equally-spaced grid. Let $j_{0,n}$ denote the nth positive root of $J_0(z)$. We know from (3.1) that the leading asymptotic behavior of $J_0(z)$ is $\sqrt{2/(\pi z)}\cos(z - \pi/4)$ for large z. Therefore, since $\cos((n - 1/2)\pi) = 0$ for $n \ge 1$ the leading asymptotic behavior of $j_{0,n}$ is $(n - 1/4)\pi$ for large n. Similarly, the leading asymptotic behavior of $j_{0,k}/j_{0,N+1}$ is $(k - 1/4)/(N + 3/4)$ for large k and N. The next lemma shows that $j_{0,1}, \ldots, j_{0,N}$ and the ratios $j_{0,1}/j_{0,N+1}, \ldots, j_{0,N}/j_{0,N+1}$ can be regarded as perturbed equally-spaced grids.

LEMMA 3.2. *Let $j_{0,n}$ denote the nth positive root of $J_0(z)$. Then,*

$$j_{0,n} = \left(n - \frac{1}{4}\right)\pi + b_n, \qquad 0 \le b_n \le \frac{1}{8(n - \frac{1}{4})\pi},$$

and

$$\frac{j_{0,k}}{j_{0,N+1}} = \frac{k - \frac{1}{4}}{N + \frac{3}{4}} + e_{k,N}, \qquad |e_{k,N}| \le \frac{1}{8(N + \frac{3}{4})(k - \frac{1}{4})\pi^2}.$$

Proof. The bounds on b_n are given in [12, Thm. 3]. For the bound on $|e_{k,N}|$ we have (since $j_{0,k} = (k - 1/4)\pi + b_k$ and $j_{0,N+1} = (N + 3/4)\pi + b_{N+1}$)

$$|e_{k,N}| = \left| \frac{j_{0,k}}{j_{0,N+1}} - \frac{(k - \frac{1}{4})}{(N + \frac{3}{4})} \right| = \left| \frac{b_k(N + \frac{3}{4}) - b_{N+1}(k - \frac{1}{4})}{((N + \frac{3}{4})\pi + b_{N+1})(N + \frac{3}{4})} \right|.$$

The result follows from $b_{N+1} \ge 0$ and $|b_k(N+3/4) - b_{N+1}(k-1/4)| \le (N+3/4)/(8(k-1/4)\pi)$. ∎

Lemma 3.2 shows that we can consider $j_{0,1}, \ldots, j_{0,N}$ as a small perturbation of $3\pi/4, 7\pi/4, \ldots, (N - 1/4)\pi$ and that we can consider $j_{0,1}/j_{0,N+1}, \ldots, j_{0,N}/j_{0,N+1}$ as a small perturbation of $3/(4N + 3), 7/(4N + 3), \ldots, (4N - 1)/(4N + 3)$. This is an important observation for sections 5 and 6.

4. Fast evaluation schemes for Schlömilch expansions. A Schlömilch expansion of a function $f : [0, 1] \to \mathbb{C}$ takes the form [23]:

$$f(r) = \sum_{n=1}^{N} c_n J_\nu(n\pi r). \tag{4.1}$$

8

Here, we take $\nu \geq 0$ to be an integer and develop a fast $\mathcal{O}(N(\log N)^2/\log\log N)$ algorithm for evaluating Schlömilch expansions at $r_1, \ldots, r_N$, where $r_k = k/N$. The evaluation of (4.1) at $r_1, \ldots, r_N$ is equivalent to computing the matrix-vector product $\underline{f} = \mathbf{J}_\nu(\underline{r}\,\underline{\omega}^{\mathsf{T}})\underline{c}$, i.e.,

$$
\begin{pmatrix} f_1 \\ \vdots \\ f_N \end{pmatrix} = \underbrace{\begin{pmatrix} J_\nu(r_1\omega_1) & \cdots & J_\nu(r_1\omega_N) \\ \vdots & \ddots & \vdots \\ J_\nu(r_N\omega_1) & \cdots & J_\nu(r_N\omega_N) \end{pmatrix}}_{\mathbf{J}_\nu(\underline{r}\,\underline{\omega}^{\mathsf{T}})} \begin{pmatrix} c_1 \\ \vdots \\ c_N \end{pmatrix}, \tag{4.2}
$$

where $f_k = f(r_k)$ and $\omega_n = n\pi$. The (k,n) entry of $\mathbf{J}_\nu(\underline{r}\,\underline{\omega}^{\mathsf{T}})$ is $J_\nu(kn\pi/N)$, which takes a similar form to the $(k+1, n+1)$ entry of C_{N+1}^I, $\cos(kn\pi/N)$, and the (k,n) entry of S_{N-1}^I, $\sin(kn\pi/N)$, see (1.3). We will carefully use the asymptotic expansion in (3.1) to approximate $\mathbf{J}_\nu(\underline{r}\,\underline{\omega}^{\mathsf{T}})$ by a weighted sum of C_{N+1}^I and S_{N-1}^I.

4.1. A simple and fast evaluation scheme for Schlömilch expansions. To derive a fast matrix-vector product for $\mathbf{J}_\nu(\underline{r}\,\underline{\omega}^{\mathsf{T}})$, we first substitute $z = r_k\omega_n$ into the asymptotic expansion (3.1). By the trigonometric addition formula we have

$$
\cos\left(r_k\omega_n - (2\nu+1)\frac{\pi}{4}\right) = \cos\left(\frac{kn\pi}{N}\right)\cos\left((2\nu+1)\frac{\pi}{4}\right) + \sin\left(\frac{kn\pi}{N}\right)\sin\left((2\nu+1)\frac{\pi}{4}\right).
$$

As a matrix decomposition we can write this as the following sum of C_{N+1}^I and S_{N-1}^I:

$$
\cos\left(\underline{r}\,\underline{\omega}^{\mathsf{T}} - (2\nu+1)\frac{\pi}{4}\right) = d_1 Q^{\mathsf{T}} C_{N+1}^I Q + d_2 \begin{bmatrix} S_{N-1}^I & \underline{0} \\ \underline{0}^{\mathsf{T}} & 0 \end{bmatrix}, \tag{4.3}
$$

where $\underline{0}$ is a zero column vector, $Q = [\underline{0}\ I_N]^{\mathsf{T}}$, $d_1 = \cos((2\nu+1)\pi/4)$, and $d_2 = \sin((2\nu+1)\pi/4)$. Similarly, for sine we have

$$
\sin\left(\underline{r}\,\underline{\omega}^{\mathsf{T}} - (2\nu+1)\frac{\pi}{4}\right) = -d_2 Q^{\mathsf{T}} C_{N+1}^I Q + d_1 \begin{bmatrix} S_{N-1}^I & \underline{0} \\ \underline{0}^{\mathsf{T}} & 0 \end{bmatrix}. \tag{4.4}
$$

Thus, by substituting (4.3) and (4.4) into (3.1) with $z = \underline{r}\,\underline{\omega}^{\mathsf{T}}$ we obtain a matrix decomposition for $\mathbf{J}_\nu(\underline{r}\,\underline{\omega}^{\mathsf{T}})$,

$$
\begin{aligned}
\mathbf{J}_\nu(\underline{r}\,\underline{\omega}^{\mathsf{T}}) = {}& \sum_{m=0}^{M-1} \frac{(-1)^m a_{2m}(\nu)}{\sqrt{\pi/2}} D_{\underline{r}}^{-2m-\frac{1}{2}} \left(d_1 Q^{\mathsf{T}} C_{N+1}^I Q + d_2 \begin{bmatrix} S_{N-1}^I & \underline{0} \\ \underline{0}^{\mathsf{T}} & 0 \end{bmatrix} \right) D_{\underline{\omega}}^{-2m-\frac{1}{2}} \\
& - \sum_{m=0}^{M-1} \frac{(-1)^m a_{2m+1}(\nu)}{\sqrt{\pi/2}} D_{\underline{r}}^{-2m-\frac{3}{2}} \left(-d_2 Q^{\mathsf{T}} C_{N+1}^I Q + d_1 \begin{bmatrix} S_{N-1}^I & \underline{0} \\ \underline{0}^{\mathsf{T}} & 0 \end{bmatrix} \right) D_{\underline{\omega}}^{-2m-\frac{3}{2}} \\
& + \mathbf{R}_{\nu,M}(\underline{r}\,\underline{\omega}^{\mathsf{T}}),
\end{aligned} \tag{4.5}
$$

where we used the fact that $\underline{r}\,\underline{\omega}^{\mathsf{T}}$ is a rank 1 matrix. We can write (4.5) more conveniently as

$$
\mathbf{J}_\nu(\underline{r}\,\underline{\omega}^{\mathsf{T}}) = \mathbf{J}_{\nu,M}^{\mathrm{ASY}}(\underline{r}\,\underline{\omega}^{\mathsf{T}}) + \mathbf{R}_{\nu,M}(\underline{r}\,\underline{\omega}^{\mathsf{T}}). \tag{4.6}
$$

One could now compute $\mathbf{J}_{\nu,M}(\underline{r}\,\underline{\omega}^{\mathsf{T}})\underline{c}$ by first computing the vector $\mathbf{J}_{\nu,M}^{\mathrm{ASY}}(\underline{r}\,\underline{\omega}^{\mathsf{T}})\underline{c}$ and then correcting it by $\mathbf{R}_{\nu,M}(\underline{r}\,\underline{\omega}^{\mathsf{T}})\underline{c}$. The matrix-vector product $\mathbf{J}_{\nu,M}^{\mathrm{ASY}}(\underline{r}\,\underline{\omega}^{\mathsf{T}})\underline{c}$
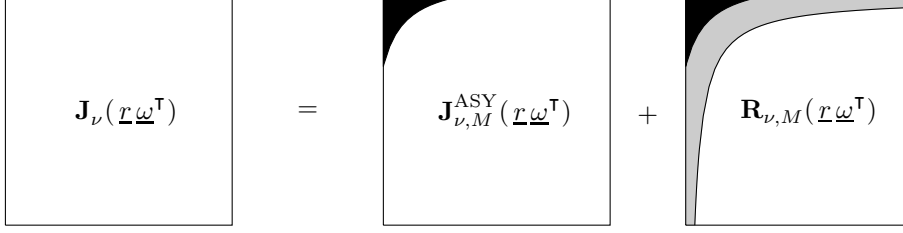
Fig. 4.1: The cause of numerical instability in the matrix decomposition (4.6). The entries in the black region of $\mathbf{J}_{\nu,M}^{\mathrm{ASY}}(\underline{r}\,\underline{\omega}^{\mathsf{T}})$ and $\mathbf{R}_{\nu,M}(\underline{r}\,\underline{\omega}^{\mathsf{T}})$ are large, of similar magnitude, and of opposite sign, resulting in numerical overflow issues and severe cancellation errors when added together. The nonwhite region of $\mathbf{R}_{\nu,M}(\underline{r}\,\underline{\omega}^{\mathsf{T}})$ contains all the entries of $\mathbf{R}_{\nu,M}(\underline{r}\,\underline{\omega}^{\mathsf{T}})$ that are larger in magnitude than ϵ. The asymptotic expansion should only be employed when $|R_{\nu,M}(r_k\omega_n)| \leq \epsilon$.

can be computed in $\mathcal{O}(N \log N)$ operations since the expression in (4.5) decomposes $\mathbf{J}_{\nu,M}^{\mathrm{ASY}}(\underline{r}\,\underline{\omega}^{\mathsf{T}})$ as a weighted sum of diagonal matrices and $2M$ DCTs and DSTs.

While this does lead to an $\mathcal{O}(N \log N)$ evaluation scheme for Schlömilch expansions, it is numerically unstable because the asymptotic expansion is employed for entries $J_\nu(r_k\omega_n)$ for which $r_k\omega_n$ is small (see Section 3.1). Numerical overflow issues and severe cancellation errors plague this approach. Figure 4.1 illustrates the entries that cause the most numerical problems. We must be more careful and only employ the asymptotic expansion for the (k,n) entry of $\mathbf{J}_{\nu,M}(\underline{r}\,\underline{\omega}^{\mathsf{T}})$ if $|R_{\nu,M}(r_k\omega_n)| \leq \epsilon$.

By Section 3.1 we know that if $z \geq s_{\nu,M}(\epsilon)$ then $|R_{\nu,M}(z)| \leq \epsilon$. Therefore, we can use (4.6) for the (k,n) entry of $\mathbf{J}_\nu(\underline{r}\,\underline{\omega}^{\mathsf{T}})$ provided that $r_k\omega_n = nk\pi/N \geq s_{\nu,M}(\epsilon)$. This is guaranteed, for instance, when

$$\min(k,n) \geq \lceil \alpha N^{\frac{1}{2}} \rceil, \tag{4.7}$$

where $\alpha = (s_{\nu,M}(\epsilon)/\pi)^{1/2}$. Therefore, we can be more careful by taking the $N \times N$ diagonal matrix Q_1 with $(Q_1)_{ii} = 1$ for $i \geq \lceil \alpha N^{\frac{1}{2}} \rceil$ and 0 otherwise, and compute $\mathbf{J}_\nu(\underline{r}\,\underline{\omega}^{\mathsf{T}})\underline{c}$ using the following matrix decomposition:

$$\mathbf{J}_\nu(\underline{r}\,\underline{\omega}^{\mathsf{T}}) = Q_1\mathbf{J}_{\nu,M}^{\mathrm{ASY}}(\underline{r}\,\underline{\omega}^{\mathsf{T}})Q_1 + Q_1\mathbf{R}_{\nu,M}(\underline{r}\,\underline{\omega}^{\mathsf{T}})Q_1 + \mathbf{J}_\nu^{\mathrm{EVAL}}(\underline{r}\,\underline{\omega}^{\mathsf{T}}), \tag{4.8}$$

where $\mathbf{J}_\nu^{\mathrm{EVAL}}(\underline{r}\,\underline{\omega}^{\mathsf{T}})$ is the matrix

$$(\mathbf{J}_\nu^{\mathrm{EVAL}}(\underline{r}\,\underline{\omega}^{\mathsf{T}}))_{kn} = \begin{cases} J_\nu(r_k\omega_n), & \min(k,n) < \lceil \alpha N^{\frac{1}{2}} \rceil, \\ 0, & \text{otherwise.} \end{cases}$$

If the (k,n) entry of $Q_1\mathbf{J}_{\nu,M}^{\mathrm{ASY}}(\underline{r}\,\underline{\omega}^{\mathsf{T}})Q_1$ is nonzero, then $nk\pi/N \geq s_{\nu,M}(\epsilon)$ and hence, (4.8) only approximates $J_\nu(r_k\omega_n)$ by a weighted sum of trigonometric functions when it is safe to do so. Figure 4.2 shows how the criterion (4.7) partitions the entries of $\mathbf{J}_\nu(\underline{r}\,\underline{\omega}^{\mathsf{T}})$. A stable $\mathcal{O}(N^{3/2})$ algorithm for evaluating Schlömilch expansions (4.1) follows, as we will now explain.

By construction all the entries of $Q_1\mathbf{R}_{\nu,M}(\underline{r}\,\underline{\omega}^{\mathsf{T}})Q_1$ are less than ϵ in magnitude so its contribution to the matrix-vector product $\mathbf{J}_\nu(\underline{r}\,\underline{\omega}^{\mathsf{T}})\underline{c}$ can be ignored, i.e., we make the approximation

$$\mathbf{J}_\nu(\underline{r}\,\underline{\omega}^{\mathsf{T}})\underline{c} \approx Q_1\mathbf{J}_{\nu,M}^{\mathrm{ASY}}(\underline{r}\,\underline{\omega}^{\mathsf{T}})Q_1\underline{c} + \mathbf{J}_\nu^{\mathrm{EVAL}}(\underline{r}\,\underline{\omega}^{\mathsf{T}})\underline{c}.$$
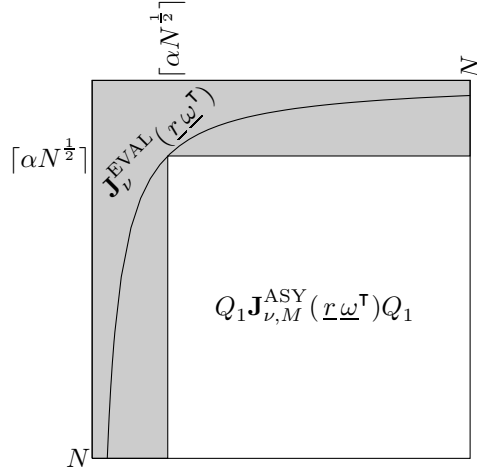
Fig. 4.2: A partition of the entries of $\mathbf{J}_\nu(\underline{r}\,\underline{\omega}^\mathsf{T})$ that motivates the matrix decomposition (4.8). The asymptotic expansion (3.1) is used for the (k,n) entry if it satisfies $\min(k,n) \geq \lceil \alpha N^{\frac{1}{2}} \rceil$. The gray region shows the nonzero entries of $\mathbf{J}_\nu^{\mathrm{EVAL}}(\underline{r}\,\underline{\omega}^\mathsf{T})$ that are not dealt with by the asymptotic expansion. The black curve indicates the entries satisfying $|R_{\nu,M}(r_k w_n)| = \epsilon$. A numerically stable $\mathcal{O}(N^{3/2})$ algorithm for evaluating Schlömilch expansions follows.

The matrix-vector product $Q_1 \mathbf{J}_{\nu,M}^{\mathrm{ASY}}(\underline{r}\,\underline{\omega}^\mathsf{T}) Q_1 \underline{c}$ can be computed in $\mathcal{O}(N \log N)$ operations using DCTs and DSTs by (4.5). Moreover, the number of nonzero entries in $\mathbf{J}_\nu^{\mathrm{EVAL}}(\underline{r}\,\underline{\omega}^\mathsf{T})$ is $\mathcal{O}(\alpha N^{1/2} N) = \mathcal{O}(N^{3/2})$ and hence, the matrix-vector product $\mathbf{J}_\nu^{\mathrm{EVAL}}(\underline{r}\,\underline{\omega}^\mathsf{T}) \underline{c}$ can be computed in $\mathcal{O}(N^{3/2})$ operations using direct summation.

This algorithm successfully stabilizes the $\mathcal{O}(N \log N)$ matrix-vector product that results from (4.6) as it only employs the asymptotic expansion for entries of $J_\nu(r_k \omega_n)$ for which $r_k \omega_n$ is sufficiently large. In practice, this algorithm is faster than direct summation for $N \geq 100$ (see Figure 4.4). However, the algorithm has an $\mathcal{O}(N^{3/2})$ complexity because the criterion (4.7) is in hindsight too cautious. We can do better.

4.2. A faster evaluation scheme for Schlömilch expansions. Note that the asymptotic expansion (3.1) is accurate for all $\mathcal{O}(N^2)$ entries of $\mathbf{J}_\nu(\underline{r}\,\underline{\omega}^\mathsf{T})$ except for

$$\mathcal{O}\left(\sum_{n=1}^{N} \sum_{k=1}^{N} \mathbf{1}_{\{r_k \omega_n < s_{\nu,M}(\epsilon)\}} \right) = \mathcal{O}\left(\sum_{n=1}^{N} \alpha^2 N/n \right) = \mathcal{O}\left(N \log N \right)$$

of them, where $\mathbf{1}$ is the indicator function and the last equality is from the leading asymptotic behavior of the Nth harmonic number as $N \to \infty$. Therefore, a lower complexity than $\mathcal{O}(N^{3/2})$ seems possible if we use the asymptotic expansion for more entries of $\mathbf{J}_\nu(\underline{r}\,\underline{\omega}^\mathsf{T})$. Roughly speaking, we need to refine the partitioning of $\mathbf{J}_\nu(\underline{r}\,\underline{\omega}^\mathsf{T})$ so that the black curve, $|R_{\nu,M}(r_k \omega_n)| = \epsilon$, in Figure 4.2 is better approximated.

First, we are only allowed to partition $\mathbf{J}_\nu(\underline{r}\,\underline{\omega}^\mathsf{T})$ with rectangular matrices since for those entries that we do employ the asymptotic expansion we need to be able to use DCTs and DSTs to compute matrix-vector products. Second, there is a balance to be

11

found since each new rectangular partition requires $2M$ more DCTs and DSTs. Too many partitions and the cost of computing DCTs and DSTs will be overwhelming, but too few and the cost of computing $\mathbf{J}_\nu^{\mathrm{EVAL}}(\underline{r}\,\underline{\omega}^{\mathsf{T}})\underline{c}$ will dominate. These two competing costs must be balanced.

To find the balance we introduce a parameter $0 < \beta < 1$. Roughly speaking, if $\beta \approx 1$ then we partition as much as possible and the asymptotic expansion is used for every entry satisfying $r_k\omega_n \geq s_{\nu,M}(\epsilon)$. If $\beta \approx 0$ then we do not refine and we keep the $\mathcal{O}(N^{3/2})$ algorithm from Section 4.1. An intermediate value of β will balance the two competing costs.

Figure 4.3 shows the partition of $\mathbf{J}_\nu(\underline{r}\,\underline{\omega}^{\mathsf{T}})$ that we consider and it is worth carefully examining that diagram. The partition corresponds to the following matrix decomposition:

$$\mathbf{J}_\nu(\underline{r}\,\underline{\omega}^{\mathsf{T}}) \approx Q_1 \mathbf{J}_{\nu,M}^{\mathrm{ASY}}(\underline{r}\,\underline{\omega}^{\mathsf{T}})Q_1 + \sum_{p=1}^{P}\left(Q_{2p+1}\mathbf{J}_{\nu,M}^{\mathrm{ASY}}(\underline{r}\,\underline{\omega}^{\mathsf{T}})Q_{2p} + Q_{2p}\mathbf{J}_{\nu,M}^{\mathrm{ASY}}(\underline{r}\,\underline{\omega}^{\mathsf{T}})Q_{2p+1}\right)$$
$$+ \mathbf{J}_\nu^{\mathrm{EVAL}}(\underline{r}\,\underline{\omega}^{\mathsf{T}}),$$
$$(4.9)$$

where P is the number of partitions, Q_1 is the diagonal matrix in Section 4.1, and the $N \times N$ matrices Q_{2p} and Q_{2p+1} are diagonal matrices with

$$(Q_{2p})_{ii} = \begin{cases} 1, & \lceil \alpha\beta^{-p}N^{\frac{1}{2}} \rceil \leq i \leq N, \\ 0, & \text{otherwise,} \end{cases} \quad (Q_{2p+1})_{ii} = \begin{cases} 1, & \lceil \alpha\beta^{p}N^{\frac{1}{2}} \rceil \leq i \leq \lfloor \alpha\beta^{p+1}N^{\frac{1}{2}} \rfloor, \\ 0, & \text{otherwise.} \end{cases}$$

Note that if the (k, n) entry of the matrix $Q_{2p+1}\mathbf{J}_{\nu,M}^{\mathrm{ASY}}(\underline{r}\,\underline{\omega}^{\mathsf{T}})Q_{2p}$ is nonzero, then $k \geq \lceil \alpha\beta^p N^{\frac{1}{2}} \rceil$ and $n \geq \lceil \alpha\beta^{-p}N^{\frac{1}{2}} \rceil$. Hence, $nk \geq \alpha^2 N$ and $r_k\omega_n \geq s_{\nu,M}(\epsilon)$. Similarly, if the (k, n) entry of $Q_{2p}\mathbf{J}_{\nu,M}^{\mathrm{ASY}}(\underline{r}\,\underline{\omega}^{\mathsf{T}})Q_{2p+1}$ is nonzero, then $r_k\omega_n \geq s_{\nu,M}(\epsilon)$. Therefore, we are only employing the asymptotic expansion (3.1) on entries of $\mathbf{J}_\nu(\underline{r}\,\underline{\omega}^{\mathsf{T}})$ for which it is accurate and numerically stable to do so.

Each matrix-vector product of the form $Q_{2p+1}\mathbf{J}_\nu^{\mathrm{ASY}}(\underline{r}\,\underline{\omega}^{\mathsf{T}})Q_{2p}\underline{c}$ requires $2M$ DCTs and DSTs from (4.5) and hence, costs $\mathcal{O}(N\log N)$ operations. There are a total of $2P + 1$ matrices of this form in (4.9) so exploiting the asymptotic expansion requires $\mathcal{O}(PN\log N)$ operations.

The cost of the matrix-vector product $\mathbf{J}_\nu^{\mathrm{EVAL}}(\underline{r}\,\underline{\omega}^{\mathsf{T}})\underline{c}$ is proportional to the number of nonzero entries in $\mathbf{J}_\nu^{\mathrm{EVAL}}(\underline{r}\,\underline{\omega}^{\mathsf{T}})$. Ignoring the constant α, this is approximately (by counting the entries in rectangular submatrices of $\mathbf{J}_\nu^{\mathrm{EVAL}}(\underline{r}\,\underline{\omega}^{\mathsf{T}})$, see Figure 4.3)

$$2\beta^P N^{\frac{3}{2}} + 2\sum_{p=1}^{P}\beta^{-p}N^{\frac{1}{2}}(\beta^{p-1}N^{\frac{1}{2}} - \beta^p N^{\frac{1}{2}}) = 2\beta^P N^{\frac{3}{2}} + 2PN\left(\frac{1}{\beta} - 1\right) = \mathcal{O}\left(PN/\beta\right),$$

where the last equality follows from the assumption that $\beta^P N^{1/2} = \mathcal{O}(1)$.

To balance the competing $\mathcal{O}(PN\log N)$ cost of exploiting the asymptotic expansion with the $\mathcal{O}(PN/\beta)$ cost of computing $\mathbf{J}_\nu^{\mathrm{EVAL}}(\underline{r}\,\underline{\omega}^{\mathsf{T}})\underline{c}$, we set $PN\log N = PN/\beta$ and find that $\beta = \mathcal{O}(1/\log N)$. Moreover, to ensure that the assumption $\beta^P N^{\frac{1}{2}} = \mathcal{O}(1)$ holds we take $P = \mathcal{O}(\log N/\log\beta) = \mathcal{O}(\log N/\log\log N)$. Thus, the number of partitions should slowly grow with N to balance the competing costs. With these asymptotically optimal choices of β and P the algorithm for computing the matrix-vector product $\mathbf{J}_\nu(\underline{r}\,\underline{\omega}^{\mathsf{T}})\underline{c}$ via (4.9) requires $\mathcal{O}(PN\log N + PN/\beta) =$
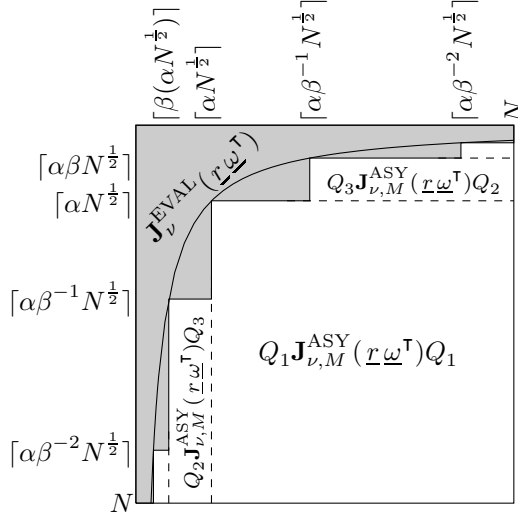
Fig. 4.3: A partition of the entries of $\mathbf{J}_\nu(\underline{r}\,\underline{\omega}^\mathsf{T})$ that motivates the matrix decomposition (4.9). The gray region shows the nonzero entries of $\mathbf{J}_\nu^{\mathrm{EVAL}}(\underline{r}\,\underline{\omega}^\mathsf{T})$ that are not dealt with by the asymptotic expansion. The black curve indicates the entries satisfying $|R_{\nu,M}(r_k w_n)| = \epsilon$. The asymptotically near-optimal choice of the parameter β is $\mathcal{O}(1/\log N)$ and the number of partitions should grow like $\mathcal{O}(\log N/\log\log N)$ with N. A numerically stable $\mathcal{O}(N(\log N)^2/\log\log N)$ algorithm for evaluating Schlömilch expansions follows.

$\mathcal{O}(N(\log N)^2/\log\log N)$ operations. More specifically, the complexity of the algorithm is $\mathcal{O}(N(\log N)^2\log(1/\epsilon)/\log\log N)$ since $M = \mathcal{O}(\log(1/\epsilon))$, see Figure 4.4.

Though, the implicit constants in $\beta = \mathcal{O}(1/\log N)$ and $P = \mathcal{O}(\log N/\log\log N)$ do not change the complexity of the algorithm, they must still be decided on. After numerical experiments we set $\beta = \min(3/\log N, 1)$ and $P = \lceil \log(30\alpha^{-1}N^{-1/2})/\log\beta \rceil$ for computational efficiency reasons.

Table 4.1 summarizes the algorithmic parameters that we have carefully selected. The user is required to specify the problem with an integer ν, a vector of expansion coefficients $\underline{c}$ in (4.1), and then provide a working accuracy $\epsilon > 0$. All other algorithmic parameters are selected in a near-optimal manner based on analysis.

REMARK 4.1. *A fast evaluation scheme for $f(r) = \sum_{n=1}^N c_n J_\nu((n+\gamma)\pi r)$, where γ is a constant, immediately follows from this section. The constants d_1 and d_2 in (4.3) and (4.4) should be replaced by diagonal matrices $D_{\underline{v}}$ and $D_{\underline{w}}$, where $v_k = \cos(-\gamma k\pi/N + (2\nu+1)\pi/4)$ and $w_k = \sin(-\gamma k\pi/N + (2\nu+1)\pi/4)$, respectively.*

4.3. Numerical results for evaluating Schlömilch expansions. We now compare three algorithms for evaluating Schlömilch expansions: (1) Direct summation (see Section 2); (2) Our $\mathcal{O}(N^{3/2})$ algorithm (see Section 4.1), and; (3) Our $\mathcal{O}(N(\log N)^2/\log\log N)$ algorithm (see Section 4.2). The three algorithms have been implemented in MATLAB and are publicly available from [25].

Figure 4.4 (left) shows the execution time for the three algorithms for evaluating Schlömilch expansions for $\nu = 0$. Here, we select the working accuracy as $\epsilon = 10^{-15}$ and use Table 4.1 to determine the other algorithmic parameters. We find that our

Parameter	Short description	Formula
M	Number of terms in (3.1)	$\max(\lfloor 0.3 \log(1/\epsilon)\rfloor, 3)$
$s_{\nu,M}(\epsilon)$	$z \geq s_{\nu,M}(\epsilon) \Rightarrow |R_{\nu,M}(z)| \leq \epsilon$	see Table 3.1
α	Partitioning parameter	$(s_{\nu,M}(\epsilon)/\pi)^{1/2}$
β	Refining parameter	$\min(3/\log N, 1)$
P	Number of partitions	$\lceil \log(30\alpha^{-1}N^{-1/2})/\log\beta \rceil$

Table 4.1: Summary of the algorithmic parameters for computing (4.2). The user is required to specify the problem with an integer ν, a vector of expansion coefficients $\underline{c}$ in (4.1), and provide a working accuracy $\epsilon > 0$. All other algorithmic parameters are carefully selected based on analysis.
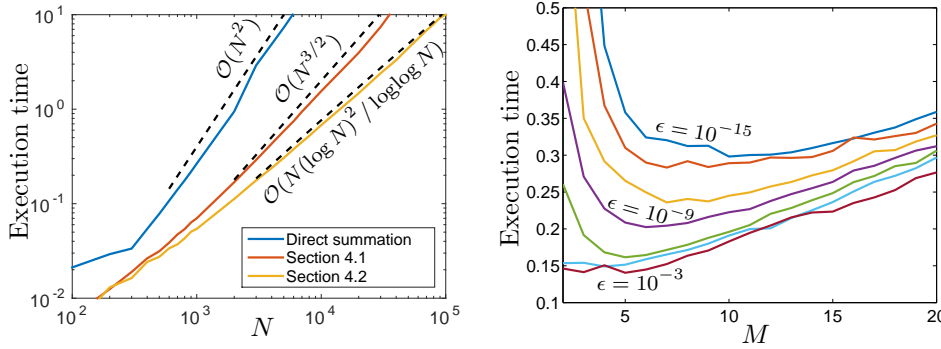

Fig. 4.4: Left: Execution time for computing (4.2) with $\mathcal{O}(N^2)$ direct summation, the $\mathcal{O}(N^{3/2})$ algorithm (see Section 4.1), and the $\mathcal{O}(N(\log N)^2/\log\log N)$ algorithm (see Section 4.2). Right: Computational timings for $N = 5{,}000$ with $2 \leq M \leq 20$ and $\epsilon = 10^{-15}, 10^{-11}, \ldots, 10^{-3}$. Numerical experiments like these motivate the choice $M = \max(\lfloor 0.3 \log(1/\epsilon)\rfloor, 3)$ in (4.5).

$\mathcal{O}(N(\log N)^2/\log\log N)$ algorithm in Section 4.2 is faster than direct summation for $N \geq 100$ and faster than our $\mathcal{O}(N^{3/2})$ algorithm for $N \geq 160$. In fact, for $N \leq 158$ the number of partitions, P, is selected to be 0 and the $\mathcal{O}(N(\log N)^2/\log\log N)$ algorithm in Section 4.2 is identical to the $\mathcal{O}(N^{3/2})$ algorithm in Section 4.1. For a relaxed working accuracy of $\epsilon = 10^{-3}$ or $\epsilon = 10^{-8}$ our $\mathcal{O}(N(\log N)^2/\log\log N)$ algorithm becomes even more computationally advantageous over direct summation.

Figure 4.4 (right) shows the execution time for our $\mathcal{O}(N(\log N)^2/\log\log N)$ algorithm with $N = 5{,}000$, $1 \leq M \leq 20$, and working accuracies $\epsilon = 10^{-15}, 10^{-11}, \ldots, 10^{-3}$. For each $\epsilon > 0$ there is a choice of M that minimizes the execution time. Numerical experiments like these motivate the choice $M = \max(\lfloor 0.3 \log(1/\epsilon)\rfloor, 3)$ in (4.5).

5. A fast evaluation scheme for Fourier–Bessel expansions. In this section we adapt the $\mathcal{O}(N(\log N)^2/\log\log N)$ algorithm in Section 4.2 to an evaluation

scheme for expansions of the form:

$$f(r) = \sum_{n=1}^{N} c_n J_\nu(rj_{0,n}), \qquad r \in [0,1], \tag{5.1}$$

where ν is an integer and $j_{0,n}$ is the nth positive root of J_0. For $\nu = 0$, (5.1) is the Fourier–Bessel expansion of order 0 of the function $f : [0,1] \to \mathbb{C}$. The algorithm that we describe evaluates (5.1) at $r_k = k/N$ for $1 \leq k \leq N$, which is equivalent to computing the matrix-vector product $\mathbf{J}_\nu(\underline{r}\,\underline{j}_0^\mathsf{T})\underline{c}$, where $\underline{j}_0$ is a column vector with entries $j_{0,1}, \ldots, j_{0,N}$.

Developing a fast algorithm for computing the matrix-vector product $\mathbf{J}_\nu(\underline{r}\,\underline{j}_0^\mathsf{T})\underline{c}$ is a challenge for FFT-based algorithms because the positive roots of J_0 are not equally-spaced. Here, we are able to derive a fast evaluation scheme by considering $j_{0,1}, \ldots, j_{0,N}$ as a perturbed equally-spaced grid (see Section 3.3.1).

First, we derive a matrix decomposition for $\mathbf{J}_\nu(\underline{u}\,(\underline{v}+\underline{w})^\mathsf{T})$, where $\underline{u}$, $\underline{v}$, and $\underline{w}$ are column vectors. Afterwards, we will consider $\underline{v}+\underline{w}$ as a perturbation of $\underline{v}$.

PROPOSITION 5.1. *For integers ν and $N \geq 1$ and for column vectors $\underline{u}$, $\underline{v}$, and $\underline{w}$ of length N, we have the following matrix decomposition:*

$$\mathbf{J}_\nu(\underline{u}\,(\underline{v}+\underline{w})^\mathsf{T}) = \sum_{s=-\infty}^{\infty} \sum_{t=0}^{\infty} \frac{(-1)^t 2^{-2t-s}}{t!(t+s)!} D_{\underline{u}}^{2t+s} \mathbf{J}_{\nu-s}(\underline{u}\,\underline{v}^\mathsf{T}) D_{\underline{w}}^{2t+s}. \tag{5.2}$$

Proof. We prove (5.2) by showing that the (k,n) entry of the matrix on the left-hand side and right-hand side are equal for $1 \leq k, n \leq N$.

Pick any $1 \leq k, n \leq N$. By the Neumann addition formula (3.5) we have

$$(\mathbf{J}_\nu(\underline{u}\,(\underline{v}+\underline{w})^\mathsf{T}))_{kn} = J_\nu(u_k v_n + u_k w_n) = \sum_{s=-\infty}^{\infty} J_{\nu-s}(u_k v_n)J_s(u_k w_n). \tag{5.3}$$

Moreover, by applying the Taylor series expansion of Bessel functions (3.3) we have

$$J_s(u_k w_n) = \sum_{t=0}^{\infty} \frac{(-1)^t 2^{-2t-s}}{t!(t+s)!}(u_k w_n)^{2t+s}. \tag{5.4}$$

By substituting (5.4) into (5.3) we obtain

$$(\mathbf{J}_\nu(\underline{u}\,(\underline{v}+\underline{w})^\mathsf{T}))_{kn} = \sum_{s=-\infty}^{\infty} \sum_{t=0}^{\infty} \frac{(-1)^t 2^{-2t-s}}{t!(t+s)!} J_{\nu-s}(u_k v_n)(u_k w_n)^{2t+s},$$

and the result follows. $\square$

We now wish to write down a particular instance of (5.2) that is useful for computing $\mathbf{J}_\nu(\underline{r}\,\underline{j}_0^\mathsf{T})\underline{c}$. By Lemma 3.2 we can write $j_{0,n} = \tilde{\omega}_n + b_n$, where $\tilde{\omega}_n = (n-1/4)\pi$ and b_n is a number such that $0 \leq b_n \leq 1/(8(n-1/4)\pi)$. In vector notation we have $\underline{j}_0 = \underline{\tilde{\omega}} + \underline{b}$. Hence, by Proposition 5.1 we have

$$\mathbf{J}_\nu(\underline{r}\,\underline{j}_0^\mathsf{T}) = \mathbf{J}_\nu(\underline{r}\,(\underline{\tilde{\omega}}+\underline{b})^\mathsf{T}) = \sum_{s=-\infty}^{\infty} \sum_{t=0}^{\infty} \frac{(-1)^t 2^{-2t-s}}{t!(t+s)!} D_{\underline{r}}^{2t+s} \mathbf{J}_{\nu-s}(\underline{r}\,\underline{\tilde{\omega}}^\mathsf{T}) D_{\underline{b}}^{2t+s}. \tag{5.5}$$

This is a useful matrix decomposition since each term in the double sum can be applied to a vector in $\mathcal{O}(N(\log N)^2/\log\log N)$ operations. The diagonal matrices $D_{\underline{b}}^{2t+s}$ and

15

$D_{\underline{r}}^{2t+s}$ have $\mathcal{O}(N)$ matrix-vector products and $\mathbf{J}_{\nu-s}(\underline{r}\,\tilde{\underline{\omega}}^{\mathsf{T}})$ has $\mathcal{O}(N(\log N)^2/\log\log N)$ matrix-vector products (see Section 4.2 and Remark 4.1). However, for (5.5) to be practical we must truncate the inner and outer sums.

Let $K \geq 1$ be an integer and truncate the outer sum in (5.5) to $\sum_{s=-K+1}^{K-1}$. By Lemma 3.1, with $z = r_k\tilde{\omega}_n$ and $\delta z = r_k b_n$, the truncated Neumann addition formula is still accurate, up to an error of ϵ, for the (k,n) entry of $\mathbf{J}_\nu(\underline{r}\,\underline{j}_0^{\mathsf{T}})$ provided that

$$
|r_k b_n| \leq |b_n| \leq \frac{1}{8(n-\frac{1}{4})\pi} \leq \frac{2}{e}\left(\frac{\epsilon}{5.2}\right)^{\frac{1}{K}}.
$$

Solving for n we find that

$$
n \geq \frac{e}{16\pi}\left(\frac{5.2}{\epsilon}\right)^{\frac{1}{K}} + \frac{1}{4} =: p_K(\epsilon). \tag{5.6}
$$

That is, once we truncate the outer sum in (5.5) to $\sum_{s=-K+1}^{K-1}$ the matrix decomposition is still accurate for all the columns of $\mathbf{J}_\nu(\underline{r}\,\underline{j}_0^{\mathsf{T}})$ except the first $\lfloor p_K(\epsilon) \rfloor$. For example when $K = 6$, $p_6(10^{-15}) \approx 22.7$ and hence, once the outer sum is truncated we must not use (5.5) on the first 22 columns of $\mathbf{J}_\nu(\underline{r}\,\underline{j}_0^{\mathsf{T}})$. Based on numerical experiments we pick $K \geq 1$ to be the smallest integer so that $p_K(\epsilon) \leq 30$ and hence, $K = \mathcal{O}(\log(1/\epsilon))$.

Next, we let $T \geq 1$ be an integer and truncate the inner sum to $\sum_{t=0}^{T-1}$. By Section 3.2 the truncated Taylor series expansion is accurate, up to an error of ϵ, for the (k,n) entry of $\mathbf{J}_\nu(\underline{r}\,\underline{j}_0^{\mathsf{T}})$ provided that

$$
|r_k b_n| \leq |b_n| \leq \frac{1}{8(n-\frac{1}{4})\pi} \leq \min_{0\leq s \leq K-1} t_{s,T}(\epsilon) \approx \left(2^{2T}(T!)^2\epsilon\right)^{\frac{1}{2T}},
$$

where the minimization is taken over $0 \leq s \leq K - 1$, instead of $-K+1 \leq s \leq K-1$, because of the relation $J_{-s}(z) = (-1)^s J_s(z)$ [18, (10.4.1)]. Solving for n we find that

$$
n \geq \frac{\epsilon^{-1/(2T)}}{16\pi(T!)^{1/T}} + \frac{1}{4} =: q_T(\epsilon). \tag{5.7}
$$

Thus, once the inner sum is truncated in (5.5) to $\sum_{t=0}^{T-1}$ the decomposition is accurate for all but the first $\lfloor q_T(\epsilon) \rfloor$ columns of $\mathbf{J}_\nu(\underline{r}\,\underline{j}_0^{\mathsf{T}})$. For example when $T = 3$, $q_3(10^{-15}) \approx 3.7$ and we must not use (5.5) on the first 3 columns of $\mathbf{J}_\nu(\underline{r}\,\underline{j}_0^{\mathsf{T}})$. In practice, we select $T \geq 1$ to be the smallest integer so that $q_T(\epsilon) \leq 30$.

To avoid employing the decomposition (5.5) with truncated sums $\sum_{-K+1}^{K-1}\sum_{t=0}^{T-1}$ on the first $\lfloor \max(p_K(\epsilon), q_T(\epsilon)) \rfloor$ columns of $\mathbf{J}_\nu(\underline{r}\,\underline{j}_0^{\mathsf{T}})$, we partition the vector $\underline{c}$. That is, we write $\mathbf{J}_\nu(\underline{r}\,\underline{j}_0^{\mathsf{T}})\underline{c} = \mathbf{J}_\nu(\underline{r}\,\underline{j}_0^{\mathsf{T}})\underline{c}_1 + \mathbf{J}_\nu(\underline{r}\,\underline{j}_0^{\mathsf{T}})\underline{c}_2$, where

$$
(\underline{c}_1)_n = \begin{cases} c_n, & n > \lfloor \max(p_K(\epsilon), q_T(\epsilon)) \rfloor, \\ 0, & \text{otherwise}, \end{cases} \qquad (\underline{c}_2)_n = \begin{cases} c_n, & n \leq \lfloor \max(p_K(\epsilon), q_T(\epsilon)) \rfloor, \\ 0, & \text{otherwise}. \end{cases}
$$

First, we compute $\underline{f}_1 = \mathbf{J}_\nu(\underline{r}\,\underline{j}_0^{\mathsf{T}})\underline{c}_1$ using (5.5) with truncated sums $\sum_{-K+1}^{K-1}\sum_{t=0}^{T-1}$ in $\mathcal{O}(N(\log N)^2/\log\log N)$ operations and then compute $\underline{f}_2 = \mathbf{J}_\nu(\underline{r}\,\underline{j}_0^{\mathsf{T}})\underline{c}_2$ using direct summation in $\mathcal{O}(N)$ operations.

5.1. Rearranging the computation for efficiency. At first it seems that computing $\mathbf{J}_\nu(\underline{r}\,\underline{j}_0^\mathsf{T})\underline{c}$ is $(2K-1)T$ times the cost of evaluating a Schlömilch expansion, since there are $(2K-1)T$ terms in $\sum_{-K+1}^{K-1}\sum_{t=0}^{T-1}$. However, the computation can be rearranged so the evaluation of (5.1) is only $2T+K-2$ times the cost. For $K=6$ and $T=3$, the values that we take when $\epsilon=10^{-15}$, we have $(2K-1)T=33$ and $2T+K-2=10$ so there is a significant computational saving here.

Since $J_{-s}(z)=(-1)^s J_s(z)$ [18, (10.4.1)] we have

$$\mathbf{J}_\nu(\underline{r}\,\underline{j}_0^\mathsf{T})\underline{c} = \sum_{s=-K+1}^{K-1}\sum_{t=0}^{T-1}\frac{(-1)^t 2^{-2t-s}}{t!(t+s)!}D_{\underline{r}}^{2t+s}\mathbf{J}_{\nu-s}(\underline{r}\,\underline{\tilde{\omega}}^\mathsf{T})D_{\underline{b}}^{2t+s}\underline{c}$$

$$= \sideset{}{'}\sum_{s=0}^{K-1}\sum_{t=0}^{T-1}\frac{(-1)^t 2^{-2t-s}}{t!(t+s)!}D_{\underline{r}}^{2t+s}\left[\mathbf{J}_{\nu-s}(\underline{r}\,\underline{\tilde{\omega}}^\mathsf{T})+(-1)^s\mathbf{J}_{\nu+s}(\underline{r}\,\underline{\tilde{\omega}}^\mathsf{T})\right]D_{\underline{b}}^{2t+s}\underline{c}$$

$$= \sum_{u=0}^{2T+K-3}D_{\underline{r}}^u\left[\sideset{}{''}\sum_{t=\max(\lceil\frac{u-K+1}{2}\rceil,0)}^{\min(\lfloor\frac{u}{2}\rfloor,T-1)}\frac{(-1)^t 2^{-u}}{t!(u-t)!}\left[\mathbf{J}_{\nu-u+2t}(\underline{r}\,\underline{\tilde{\omega}}^\mathsf{T})+(-1)^{u-2t}\mathbf{J}_{\nu+u-2t}(\underline{r}\,\underline{\tilde{\omega}}^\mathsf{T})\right]\right]D_{\underline{b}}^u\underline{c},$$

where the single prime on the sum indicates that the first term is halved and the double prime indicates that the last term is halved. Here, the last equality follows by letting $u=2t+s$ and performing a change of variables. Now, for each $0\leq u\leq 2T+K-3$, the inner sum can be computed at the cost of one application of the $\mathcal{O}(N(\log N)^2/\log\log N)$ algorithm in Section 4.2. Since each evaluation of a Schlömilch expansion costs $\mathcal{O}(N(\log N)^2\log(1/\epsilon)/\log\log N)$ operations and $K=\mathcal{O}(\log(1/\epsilon))$, a Fourier–Bessel evaluation costs $\mathcal{O}(N(\log N)^2\log(1/\epsilon)^2/\log\log N)$ operations.

5.2. Numerical results for evaluating Fourier–Bessel expansions. Figure 5.1 (left) shows the execution time for evaluating Fourier–Bessel expansions of order 0 with direct summation and our $\mathcal{O}(N(\log N)^2/\log\log N)$ algorithm with working accuracies of $\epsilon=10^{-15},10^{-8},10^{-3}$. Our algorithm is faster than direct summation for $N\geq 700$ when $\epsilon=10^{-15}$. Relaxing the working accuracy not only decreases K and T, but also speeds up the evaluation of each Schlömilch expansion. When $\epsilon=10^{-3}$, our algorithm is faster than direct summation for $N\geq 100$.

The algorithmic parameters are selected based on careful analysis so that the evaluation scheme is numerically stable and the computed vector $\underline{f}$ has a componentwise accuracy of $\mathcal{O}(\epsilon\|\underline{c}\|_1)$, where $\|\cdot\|_1$ is the vector 1-norm. In Figure 5.1 (right) we verify this for $\epsilon=10^{-15},10^{-8},10^{-3}$ by considering the error $\|\underline{f}-\underline{f}^{\mathrm{quad}}\|_\infty$, where $\|\cdot\|_\infty$ is the absolute maximum vector norm and $\underline{f}^{\mathrm{quad}}$ is the vector of values computed in quadruple precision using direct summation. For these experiments we take the entries of $\underline{c}$ as realizations of independent Gaussians so that $\|\underline{c}\|_1=\mathcal{O}(N^{1/2})$.

6. The discrete Hankel transform. We now adapt the algorithm in Section 5 to the computation of the discrete Hankel transform (DHT) of order 0, which is defined as [15]

$$f_k = \sum_{n=1}^{N} c_n J_0(j_{0,k}j_{0,n}/j_{0,N+1}), \qquad 1\leq k\leq N, \tag{6.1}$$

where $j_{0,n}$ is the nth positive root of $J_0(z)$. It is equivalent to the matrix-vector product $\mathbf{J}_0(\underline{j}_0\,\underline{j}_0^\mathsf{T}/j_{0,N+1})\underline{c}$ and is more difficult for FFT-based algorithms than the
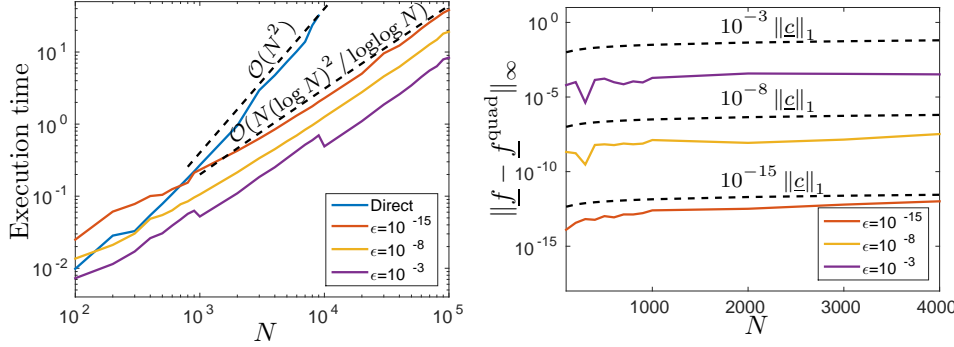
Fig. 5.1: Left: Execution time for evaluating (5.1) at $1/N, \ldots, N/N$ with $\mathcal{O}(N^2)$ direct summation and the $\mathcal{O}(N(\log N)^2/\log\log N)$ algorithm described in this section. For a relaxed working accuracy our algorithm becomes even more computationally advantageous over direct summation. Right: Maximum absolute error in the computed vector $\underline{f}$. Here, we compare the observed error (solid lines) with the expected bound of $\mathcal{O}(\epsilon\|\underline{c}\|_1)$ (dotted lines).

task in Section 5 because it may be regarded as evaluating a Fourier–Bessel expansion at points that are not equally-spaced.

To extend our $\mathcal{O}(N(\log N)^2/\log\log N)$ algorithm to this setting we consider the ratios $j_{0,1}/j_{0,N+1}, \ldots, j_{0,N}/j_{0,N+1}$ as a perturbed equally-spaced grid. By Lemma 3.2 we have

$$\frac{j_{0,k}}{j_{0,N+1}} = \frac{k-\frac{1}{4}}{N+\frac{3}{4}} + e_{k,N}, \quad |e_{k,N}| \leq \frac{1}{8(N+\frac{3}{4})(k-\frac{1}{4})\pi^2}, \qquad 1 \leq k \leq N.$$

We can write this in vector notation as $\underline{j}_0/j_{0,N+1} = \underline{\tilde{r}} + \underline{e}_N$, where $\tilde{r}_k = (k-1/4)/(N+3/4)$. Since $\mathbf{J}_0((\underline{v}+\underline{w})\underline{u}^\mathsf{T}) = \mathbf{J}_0(\underline{u}(\underline{v}+\underline{w})^\mathsf{T})^\mathsf{T}$ and $\mathbf{J}_0(\underline{j}_0\,\underline{j}_0^\mathsf{T}/j_{0,N+1}) = \mathbf{J}_0((\underline{\tilde{r}}+\underline{e}_N)\,\underline{j}_0^\mathsf{T})$, we have by Theorem 5.1 the following matrix decomposition:

$$\mathbf{J}_0(\underline{j}_0\,\underline{j}_0^\mathsf{T}/j_{0,N+1}) = \sum_{s=-\infty}^{\infty}\sum_{t=0}^{\infty} \frac{(-1)^t 2^{-2t-s}}{t!(t+s)!} D_{\underline{e}_N}^{2t+s}\mathbf{J}_{-s}(\underline{\tilde{r}}\,\underline{j}_0^\mathsf{T})D_{\underline{j}_0}^{2t+s}. \qquad (6.2)$$

Each term in (6.2) can be applied to a vector in $\mathcal{O}(N(\log N)^2/\log\log N)$ operations. The diagonal matrices $D_{\underline{j}_0}^{2t+s}$ and $D_{\underline{e}_N}^{2t+s}$ have $\mathcal{O}(N)$ matrix-vector products and by Section 5 the matrix-vector product $\mathbf{J}_{-s}(\underline{\tilde{r}}\,\underline{j}_0^\mathsf{T})\underline{c}$ can be computed in $\mathcal{O}(N(\log N)^2/\log\log N)$ operations.[2] However, for (6.2) to be practical we must truncate the double sum.

Let $K \geq 1$ and $T \geq 1$ be integers and consider truncating the sum in (6.2) to $\sum_{s=-K+1}^{K-1}\sum_{t=0}^{T-1}$. Using similar reasoning to that in Section 5, if we truncate (6.2) then the matrix decomposition is accurate, up to an error of ϵ, provided that

$$|e_{k,N}j_{0,n}| \leq \frac{j_{0,n}}{8(N+\frac{3}{4})(k-\frac{1}{4})\pi^2} \leq \frac{1+\frac{1}{8(n-\frac{1}{4})(N+\frac{3}{4})\pi^2}}{8(k-\frac{1}{4})\pi} \leq \frac{1.01}{8(k-\frac{1}{4})\pi} \leq \frac{2}{e}\left(\frac{\epsilon}{5.2}\right)^{\frac{1}{K}}$$

---

[2] Here, $\tilde{r}_k = (k-1/4)/(N+3/4) = (4k-1)/(4N+3)$ so to compute $\mathbf{J}_{-s}(\underline{\tilde{r}}\,\underline{j}_0^\mathsf{T})\underline{c}$ pad the vector $\underline{c}$ with $3N+3$ zeros, use the algorithm in Section 5 with N replaced by $4N+3$, and then only keep f_{4i-1} for $1 \leq i \leq N$.

18

and

$$|e_{k,N} j_{0,n}| \leq \frac{1.01}{8(k-\frac{1}{4})\pi} \leq \min_{0 \leq s \leq K-1} t_{s,T}(\epsilon) \approx \left(2^{2T}(T!)^2 \epsilon\right)^{\frac{1}{2T}}.$$

where we used the bound $j_{0,n}/((N+3/4)\pi) \leq (n-1/4)/(N+3/4)+1/(8(n-1/4)(N+3/4)\pi^2) \leq 1.01$ for $n, N \geq 1$. Equivalently, provided that

$$k \geq 1.01 \times \max(p_K(\epsilon), q_T(\epsilon)),$$

where $p_K(\epsilon)$ and $q_T(\epsilon)$ are defined in (5.6) and (5.7), respectively. That is, after truncating the sums in (6.2) we can use the decomposition on all the entries of $\mathbf{J}_0(\underline{j}_0 \underline{j}_0^\mathsf{T}/j_{0,N+1})$ except for those in the first $\lfloor 1.01 \max(p_K(\epsilon), q_T(\epsilon))\rfloor$ rows. In practice, we take $K \geq 1$ and $T \geq 1$ to be the smallest integers so that $1.01 \max(p_K(\epsilon), q_T(\epsilon)) \leq 30$.

To avoid employing a truncated version of (6.2) on the first few rows we compute $\underline{f} = \mathbf{J}_0(\underline{j}_0 \underline{j}_0^\mathsf{T}/j_{0,N+1})\underline{c}$ using (6.2) with sums $\sum_{s=-K+1}^{K-1} \sum_{t=0}^{T-1}$ in $\mathcal{O}(N(\log N)^2/\log\log N)$ operations, discard the first $\lfloor 1.01 \max(p_K(\epsilon), q_T(\epsilon))\rfloor$ entries of $\underline{f}$, and then compute the discarded entries using direct summation in $\mathcal{O}(N)$ operations.

In the same way as Section 5.1, we can rearrange the computation so that only $2T + K - 2$, instead of $(2K-1)T$, Fourier–Bessel evaluations are required. Each Fourier–Bessel evaluation requires $2T+K-2$ Schlömilch evaluations. Hence, a discrete Hankel transform requires $(2T+K-2)^2$ Schlömilch evaluations. Since $K = \log(1/\epsilon)$, our algorithm for the DHT requires $\mathcal{O}(N(\log N)^2 \log(1/\epsilon)^3/\log\log N)$ operations.

The exposition in this section may suggest that the same algorithmic ideas continue to work for $\nu > 0$. While this is correct, the computational performance rapidly deteriorates as ν increases. This is because the vector of Bessel roots $\underline{j}_\nu$ and the vector of ratios $\underline{j}_\nu^\mathsf{T}/j_{\nu,N+1}$ are distributed less like an equally-spaced grid for $\nu > 0$. This significantly increases the number of terms required in the Taylor expansion and Neumann addition formula. For large $\nu > 0$, different algorithmic ideas should be employed, for example, one should take into account that $|J_\nu(z)|$ can be very small when $0 \leq z < \nu$. It is expected that the matched asymptotic (transition) region of $J_\nu(z)$, when $z \approx \nu$, will be the most difficult to exploit when deriving a fast algorithm.

6.1. Numerical results for computing the discrete Hankel transform. We have implemented our $\mathcal{O}(N(\log N)^2/\log\log N)$ algorithm for computing the DHT in MATLAB. It is publicly available from [25].

Figure 6.1 (left) shows the execution time for computing the DHT of order 0 with direct summation and our $\mathcal{O}(N(\log N)^2/\log\log N)$ algorithm with working tolerances of $\epsilon = 10^{-15}, 10^{-8}, 10^{-3}$. Our algorithm is faster than direct summation for $N \geq 6{,}000$ when $\epsilon = 10^{-15}$, for $N \geq 2{,}000$ when $\epsilon = 10^{-8}$, and for $N \geq 100$ when $\epsilon = 10^{-3}$. Figure 6.1 (right) verifies that the selected algorithmic parameters do achieve the desired error bound of $\|\underline{f} - \underline{f}^{\mathrm{quad}}\|_\infty = \mathcal{O}(\epsilon\|\underline{c}\|_1)$, where $\underline{f}$ is the vector of the computed values of (6.1) and $\underline{f}^{\mathrm{quad}}$ is the vector of values computed in quadruple precision using direct summation. In this experiment we take the entries of $\underline{c}$ to be realizations of independent Gaussians (with mean 0 and variance 1) so that $\|\underline{c}\|_1 = \mathcal{O}(N^{1/2})$.

The DHT is, up to a diagonal scaling, its own inverse as $N \to \infty$. In particular, by [15] we have, as $N \to \infty$,

$$\frac{1}{j_{0,N+1}^2}\mathbf{J}_0(\underline{j}_0 \underline{j}_0^\mathsf{T}/j_{0,N+1})D_{\underline{v}}\mathbf{J}_0(\underline{j}_0 \underline{j}_0^\mathsf{T}/j_{0,N+1})D_{\underline{v}} \sim I_N, \qquad (6.3)$$
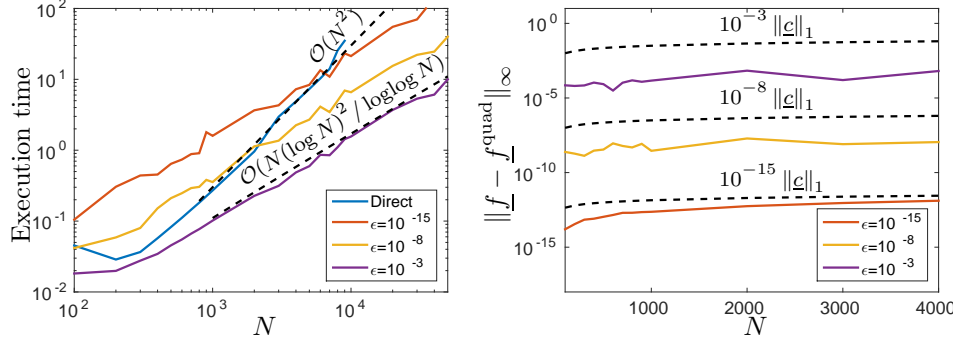
Fig. 6.1: Left: Execution time for computing the DHT in (6.1) using direct summation and our $\mathcal{O}(N(\log N)^2/\log\log N)$ algorithm with $\epsilon = 10^{-15}, 10^{-8}, 10^{-3}$. Right: The maximum absolute error in the computed vector $\underline{f}$. Here, we compare the computed error (solid lines) with the expected error bound $\overline{\mathcal{O}}(\epsilon\|\underline{c}\|_1)$.

where $D_{\underline{v}}$ is a diagonal matrix with entries $v_n = 2/J_1(j_{0,n})^2$, and I_N is the $N \times N$ identity matrix. We can use the relation (6.3) to verify our algorithm for the DHT. Figure 6.2 shows the error $\|\tilde{\underline{c}} - \underline{c}\|_\infty$, where $\underline{c}$ is a column vector with entries drawn from independent Gaussians and

$$\tilde{\underline{c}} = \frac{1}{j_{0,N+1}^2}\mathbf{J}_0(\underline{j}_0\,\underline{j}_0^\mathsf{T}/j_{0,N+1})D_{\underline{v}}\mathbf{J}_0(\underline{j}_0\,\underline{j}_0^\mathsf{T}/j_{0,N+1})D_{\underline{v}}\underline{c}. \tag{6.4}$$

For small N, $\|\tilde{\underline{c}}-\underline{c}\|_\infty$ will be large because (6.3) only holds as $N \to \infty$. However, even for large N, we observe that $\|\tilde{\underline{c}} - \underline{c}\|_\infty$ grows like $\mathcal{O}(N^{3/2})$. This can be explained by tracking the accumulation of numerical errors. Since the entries of $\underline{c}$ are drawn from independent Gaussians and $v_n = \mathcal{O}(n)$ we have $\|D_{\underline{v}}\underline{c}\|_1 = \mathcal{O}(N^{3/2})$ and hence, we expect $\|\mathbf{J}_0(\underline{j}_0\,\underline{j}_0^\mathsf{T}/j_{0,N+1})D_{\underline{v}}\underline{c}\|_\infty = \mathcal{O}(\epsilon\|D_{\underline{v}}\underline{c}\|_1) = \mathcal{O}(\epsilon N^{3/2})$. By the same reasoning we expect $\|\mathbf{J}_0(\underline{j}_0\,\underline{j}_0^\mathsf{T}/j_{0,N+1})D_{\underline{v}}\mathbf{J}_0(\underline{j}_0\,\underline{j}_0^\mathsf{T}/j_{0,N+1})D_{\underline{v}}\underline{c}\|_\infty = \mathcal{O}(\epsilon N^{7/2})$. Finally, this gets multiplied by $j_{0,N+1}^{-2} = \mathcal{O}(N^{-2})$ in (6.4) so we expect $\|\tilde{\underline{c}} - \underline{c}\|_\infty = \mathcal{O}(N^{3/2})$. If in practice we desire $\|\tilde{\underline{c}} - \underline{c}\|_\infty = \mathcal{O}(1)$, then it is sufficient to have $|c_n| = o(n^{-2})$.

Conclusion. An asymptotic expansion of Bessel functions for large arguments is carefully employed to derive a numerically stable $\mathcal{O}(N(\log N)^2/\log\log N)$ algorithm for evaluating Schlömilch and Fourier–Bessel expansions as well as computing the discrete Hankel transform. All algorithmic parameters are selected based on error bounds to achieve a near-optimal computational cost for any accuracy goal. For a working accuracy of 10^{-15}, the algorithm is faster than direct summation for evaluating Schlömilch expansions when $N \geq 100$, Fourier–Bessel expansions when $N \geq 700$, and the discrete Hankel transform when $N \geq 6,000$.

REFERENCES
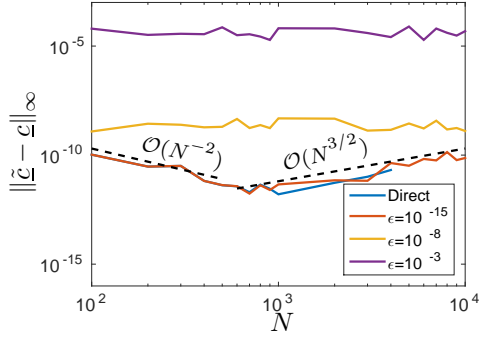
Fig. 6.2: The maximum absolute error $\|\underline{\tilde{c}} - \underline{c}\|$, where $\underline{c}$ and $\underline{\tilde{c}}$ satisfy (6.4). Here, the DHT was computed with direct summation and our algorithm with $\epsilon = 10^{-3}, 10^{-8}, 10^{-15}$. As $N \to \infty$ the DHT is its own inverse, up to a diagonal scaling. For $N < 700$ the error $\|\underline{\tilde{c}} - \underline{c}\|$ decays like $\mathcal{O}(N^{-2})$ because (6.3) only holds asymptotically as $N \to \infty$. For $N > 700$ the error $\|\underline{\tilde{c}} - \underline{c}\|$ grows like $\mathcal{O}(N^{3/2})$ from numerical error accumulation.

[1] B. K. ALPERT AND V. ROKHLIN, *A fast algorithm for the evaluation of Legendre expansions*, SIAM J. Sci. Stat. Comput., 12 (1991), pp. 158–179.

[2] D. E. AMOS, *Algorithm 644: A portable package for Bessel functions of a complex argument and nonnegative order*, ACM Trans. Math. Soft., 12 (1986), pp. 265–273.

[3] R. BISSELING AND R. KOSLOFF, *The fast Hankel transform as a tool in the solution of the time dependent Schrdinger equation*, J. Comput. Phys., 59 (1985), pp. 136–151.

[4] S. M. CANDEL, *Dual algorithms for fast calculation of the Fourier-Bessel transform*, IEEE Trans. Acoust. Sp. Sig. Proc., 29 (1981), pp. 963–972.

[5] M. J. CREE AND P. J. BONES, *Algorithms to numerically evaluate the Hankel transform*, Computers and Mathematics with Applications, 26 (1993), pp. 1–12.

[6] K. GOPALAN AND C. S. CHEN, *Fast computation of zero order Hankel transform*, J. Franklin Institute, 316 (1983), pp. 317–326.

[7] M. GUIZAR-SICAIROS AND J. C. GUTIÉRREZ-VEGA, *Computation of quasi-discrete Hankel transforms of integer order for propagating optical wave fields*, J. Opt. Soc. Amer. A, 21 (2004), pp. 53–58.

[8] N. HALE AND A. TOWNSEND, *A fast, simple, and stable Chebyshev–Legendre transform using an asymptotic formula*, SIAM J. Sci. Comput., 36 (2014), A148–A167.

[9] N. HALE AND A. TOWNSEND, *An algorithm for the convolution of Legendre series*, SIAM J. Sci. Comput., 36 (2014), A1207–A1220.

[10] N. HALE AND A. TOWNSEND, *A fast FFT-based discrete Legendre transform*, submitted.

[11] H. HANKEL, *Die Cylinderfunctionen erster und zweiter Art*, Math. Ann., 1 (1869), pp. 467–501.

[12] H. W. HETHCOTE, *Bounds for zeros of some special functions*, Proc. Amer. Math. Soc., 25. 1970, pp. 72–74.

[13] W. E. HIGGINS AND D. C. MUNSON JR., *A Hankel transform approach to tomographic image reconstruction*, IEEE Transactions on Medical Imaging, 7 (1988), pp. 59–72.

[14] K. KAPUR AND V. ROKHLIN, *An algorithm for the fast Hankel transform*, Yale technical report, August 1995.

[15] H. F. JOHNSON, *An improved method for computing a discrete Hankel transform*, Computer Physics Communications, 43 (1987), pp. 181–202.

[16] A. MORI, R. SUDA, AND M. SUGIHARA, *An improvement on Orszag's fast algorithm for Legendre polynomial transform*, Trans. Info. Processing Soc. Japan, 40 (1999), pp. 3612–3615.

[17] J. NACHAMKIN AND C. J. MAGGIORE, *A Fourier Bessel transform method for efficiently calculating the magnetic field of solenoids*, J. Comput. Phys., 37 (1980), pp. 41–55.

[18] F. W. J. OLVER, D. W. LOZIER, R. F. BOISVERT, AND C. W. CLARK, *NIST Handbook of Mathematical Functions*, Cambridge University Press, 2010.

[19] M. O'NEIL, F. WOOLFE, AND V. ROKHLIN, *An algorithm for the rapid evaluation of special*

       function transforms, Appl. Comput. Harm. Anal., 28 (2010), pp. 203–226.

[20] R. PIESSENS AND M. BRANDERS, *Approximation for Bessel functions and their application in the computation of Hankel transforms*, Comput. Math. Appl., 8 (1982), pp. 305–311.

[21] A. D. POULARIKAS, *Transforms and Applications Handbook, Third Edition*, CRC Press, 2010.

[22] V. ROKHLIN AND M. TYGERT, *Fast algorithms for spherical harmonic expansions*, SIAM J. Sci. Comput., 27 (2006), pp. 1903–1928.

[23] O. SCHLÖMILCH, *Note sur la variation des constantes arbitraires d'une intégrale définie*, Journal für die reine und angewandte Mathematik, 33 (1846), pp. 268–280.

[24] O. A. SHARAFEDDIN, H. F. BOWEN, D. J. KOURI, AND D. K. HOFFMAN, *Numerical evaluation of spherical Bessel transforms via fast Fourier transforms*, J. Comput. Phys., 100 (1992), pp. 294–296.

[25] A. TOWNSEND, FastAsyTransforms, `http://github.com/ajt60gaibb/FastAsyTransforms`.

[26] M. TYGERT, *Fast algorithms for spherical harmonic expansions, II*, J. Comput. Phys., 227 (2008), pp. 4260–4279.

[27] Wolfram Research, `http://functions.wolfram.com/03.01.06.0037.01`.