# A fast analysis-based discrete Hankel transform using asymptotic expansions


Alex Townsend
MIT


IMA Leslie Fox Prize Meeting, 22nd June, 2015

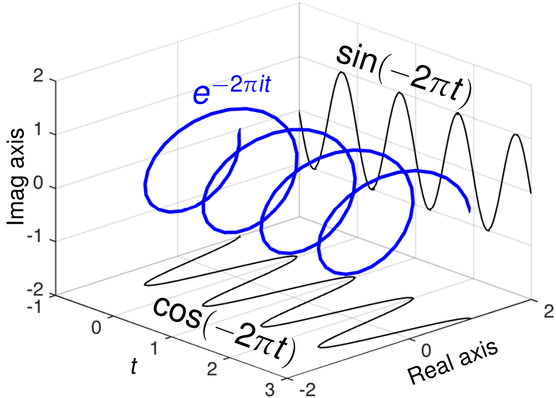
Based on: T., “A fast analysis-based discrete Hankel transform using asymptotic expansions”, to appear in SINUM.

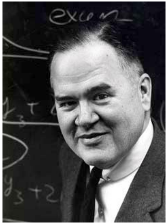
The FFT computes the DFT in $\mathcal{O}(N \log N)$ operations:

$$X_k = \sum_{n=0}^{N-1} x_n \cdot e^{-2\pi i n k/N}, \qquad 0 \leqslant k \leqslant N-1.$$



James Cooley





John Tukey

# Introduction
## A circularly symmetric 2D Fourier transform

Let $f(x, y)$ be a circularly symmetric function. Then, the finite Hankel transform is

$$H_0(\omega) = \frac{1}{2\pi}\iint_{x^2+y^2\leqslant 1} f(x,y)e^{-i(\omega_1 x+\omega_2 y)}dx\,dy = \int_0^1 f(r)J_0(r\omega)r\,dr, \qquad \omega^2 = \omega_1^2+\omega_2^2$$

**Discrete analogue:** $H_0(j_{0,k}) \approx \frac{1}{j_{0,N+1}^2}\sum_{n=1}^{N}\frac{2}{J_1^2(j_{0,n})}f(j_{0,n}/j_{0,N+1})J_0(j_{0,k}j_{0,n}/j_{0,N+1}).$

# Introduction

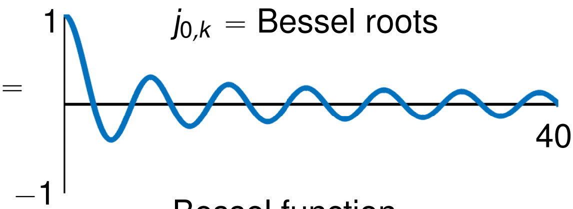
## A circularly symmetric 2D Fourier transform

Let $f(x,y)$ be a circularly symmetric function. Then, the finite Hankel transform is

$$H_0(\omega) = \frac{1}{2\pi}\iint_{x^2+y^2\leqslant 1} f(x,y)e^{-i(\omega_1 x+\omega_2 y)}dx\,dy = \int_0^1 f(r)J_0(r\omega)r\,dr, \qquad \omega^2 = \omega_1^2+\omega_2^2$$

**Discrete analogue:** $H_0(j_{0,k}) \approx \frac{1}{j_{0,N+1}^2}\sum_{n=1}^{N}\frac{2}{J_1^2(j_{0,n})}f(j_{0,n}/j_{0,N+1})J_0(j_{0,k}j_{0,n}/j_{0,N+1}).$

$J_0(z) =$




$j_{0,k}$ = Bessel roots

Bessel function

H. Fisk Johnson

Let $f(x, y)$ be a circularly symmetric function. Then, the finite Hankel transform is

$$H_0(\omega) = \frac{1}{2\pi}\iint_{x^2+y^2\leqslant 1} f(x,y)e^{-i(\omega_1 x+\omega_2 y)}dx\,dy = \int_0^1 f(r)J_0(r\omega)r\,dr, \qquad \omega^2 = \omega_1^2+\omega_2^2$$

**Discrete analogue:** $H_0(j_{0,k}) \approx \frac{1}{j_{0,N+1}^2}\sum_{n=1}^{N}\frac{2}{J_1^2(j_{0,n})}f(j_{0,n}/j_{0,N+1})J_0(j_{0,k}j_{0,n}/j_{0,N+1}).$

The discrete Hankel transform of order 0 (DHT) [Johnson, 86]:

$$X_k = \textstyle\sum_{n=1}^N x_n J_0(j_{0,k}\, j_{0,n}/j_{0,N+1}), \qquad 1 \leqslant k \leqslant N.$$

H. Fisk Johnson

# Introduction

## A circularly symmetric 2D Fourier transform

Let $f(x, y)$ be a circularly symmetric function. Then, the finite Hankel transform is

$$H_0(\omega) = \frac{1}{2\pi}\iint_{x^2+y^2\leqslant 1} f(x,y)e^{-i(\omega_1 x+\omega_2 y)}dx\,dy = \int_0^1 f(r)J_0(r\omega)r\,dr, \qquad \omega^2 = \omega_1^2+\omega_2^2$$

**Discrete analogue:** $$H_0(j_{0,k}) \approx \frac{1}{j_{0,N+1}^2}\sum_{n=1}^{N}\frac{2}{J_1^2(j_{0,n})}f(j_{0,n}/j_{0,N+1})J_0(j_{0,k}j_{0,n}/j_{0,N+1}).$$

The discrete Hankel transform of order 0 (DHT) [Johnson, 86]:

$$\underline{X} = \mathbf{J}_0(\underline{j}_0\,\underline{j}_0^{\mathsf{T}}/j_{0,N+1})\,\underline{x}.$$

Independently rediscovered by [Yu et al., 89] and [Guizar-Sicairos & Gutiérrez-Vega, 04].

H. Fisk Johnson

Let $f(x, y)$ be a circularly symmetric function. Then, the finite Hankel transform is

$$H_0(\omega) = \frac{1}{2\pi}\iint_{x^2+y^2\leqslant 1} f(x,y)e^{-i(\omega_1 x+\omega_2 y)}dx\,dy = \int_0^1 f(r)J_0(r\omega)r\,dr, \qquad \omega^2 = \omega_1^2+\omega_2^2$$

**Discrete analogue:** $$H_0(j_{0,k}) \approx \frac{1}{j_{0,N+1}^2}\sum_{n=1}^{N}\frac{2}{J_1^2(j_{0,n})}f(j_{0,n}/j_{0,N+1})J_0(j_{0,k}j_{0,n}/j_{0,N+1}).$$

The discrete Hankel transform of order 0 (DHT) [Johnson, 86]:

$$\underline{X} = \mathbf{J}_0(\underline{j}_0\,\underline{j}_0^{\mathsf{T}}/j_{0,N+1})\,\underline{x}.$$

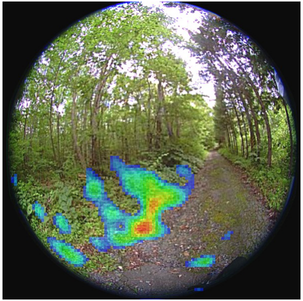
Independently rediscovered by [Yu et al., 89] and [Guizar-Sicairos & Gutiérrez-Vega, 04].


H. Fisk Johnson

**Optics, electromagnetics, imaging, and the numerical solution of PDEs:**
[Siegman, 77], [Oppenheim, 78], [Nachamkin & Maggiore, 80], [Candel, 81], [Gopalan & Chen, 83], [Johnson, 86], [Johnson, 87], [Higgins & Munson, 88], [Yu et al., 89], [Vittorio et al., 92], [Sharafeddin et al., 92], [Agnesi et al., 93], [Zhao & Halling, 93], [Cree & Bones, 94], [Barakat, 96], [Ferrari, 99], [Secada, 99], [Wieder, 99], [Knockaert, 00], [Zhang et al., 02], [Markham & Conchello, 03], [Perciante & Ferrari, 04], [Guizar-Sicairos & Gutiérrez-Vega, 04], [Cerjan, 07], [Poularikas, 10]

**Compton camera:**

- Hankel transform is useful for image reconstruction [Cree & Bones, 94].
- Used in nuclear medical imaging and radioactive waste detection.



1. **"Direct" approach:**
   Naively computes

$$\underline{X} = \mathbf{J}_0(\underline{j}_0 \underline{j}_0^{\mathsf{T}}/j_{0,N+1})\underline{x}$$

   in $\mathcal{O}(N^2)$ operations [Amos, 86].

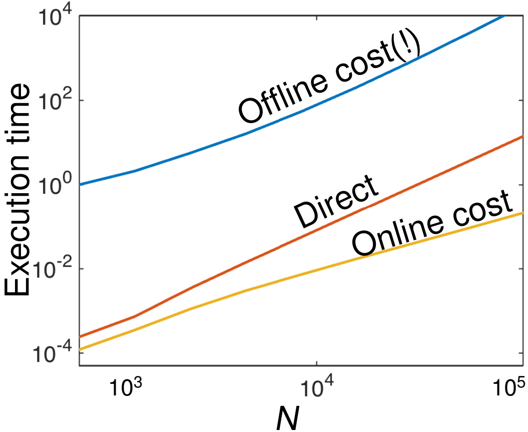
2. **Butterfly scheme:**

$$\text{Online cost} = \mathcal{O}(N \log N)$$

$$\text{Offline cost} = \mathcal{O}(N^2)$$

[O'Neil, Woolfe, & Rokhlin, 10].

Butterfly scheme (state-of-the-art)




Timings from [O'Neil, Woolfe, & Rokhlin, 10].

The discrete Hankel transform of order 0 (DHT) [Johnson, 86]:

$$\underline{X} = \mathbf{J}_0\left( \underline{j}_0 \, \underline{j}_0^{\top} / j_{0,N+1} \right) \underline{x}.$$

1. Kernel. $J_0(z)$ does **not** have constant amplitude.

2. Frequencies. $j_{0,1}, \ldots, j_{0,N}$ are **not** equispaced.

3. Time samples. $j_{0,1}/j_{0,N+1}, \ldots, j_{0,N}/j_{0,N+1}$ are **not** equispaced.

## Three challenging differences between the DFT and DHT

The discrete Hankel transform of order 0 (DHT) [Johnson, 86]:

$$\underline{X} = \mathbf{J}_0(\underline{j}_0 \underline{j}_0^{\intercal}/j_{0,N+1})\,\underline{x}.$$

1. Kernel. $J_0(z)$ does **not** have constant amplitude.
2. Frequencies. $j_{0,1}, \ldots, j_{0,N}$ are **not** equispaced.
3. Time samples. $j_{0,1}/j_{0,N+1}, \ldots, j_{0,N}/j_{0,N+1}$ are **not** equispaced.

# Introduction

## Three challenging differences between the DFT and DHT

The discrete Hankel transform of order 0 (DHT) [Johnson, 86]:

$$\underline{X} = \mathbf{J}_0(\,\underline{j}_0\,\underline{j}_0^{\intercal}/j_{0,N+1}\,)\,\underline{x}.$$

1. Kernel. $J_0(z)$ does **not** have constant amplitude.

2. Frequencies. $j_{0,1},\ldots,j_{0,N}$ are **not** equispaced.

3. Time samples. $j_{0,1}/j_{0,N+1},\ldots,j_{0,N}/j_{0,N+1}$ are **not** equispaced.

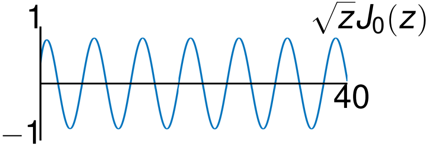
The discrete Hankel transform of order 0 (DHT) [Johnson, 86]:

$$\underline{X} = \mathbf{J}_0(\underline{j}_0 \underline{j}_0^{\mathsf{T}} / j_{0,N+1})\,\underline{x}.$$

1. Kernel. $J_0(z)$ does **not** have constant amplitude.
2. Frequencies. $j_{0,1}, \ldots, j_{0,N}$ are **not** equispaced.
3. Time samples. $j_{0,1}/j_{0,N+1}, \ldots, j_{0,N}/j_{0,N+1}$ are **not** equispaced.

The discrete Hankel transform of order 0 (DHT) [Johnson, 86]:

$$\underline{X} = \mathbf{J}_0\left(\underline{j}_0\,\underline{j}_0^{\mathsf{T}}/j_{0,N+1}\right)\underline{x}.$$




1. Kernel. $J_0(z)$ does **not** have constant amplitude.

2. Frequencies. $j_{0,1},\ldots,j_{0,N}$ are **not** equispaced.

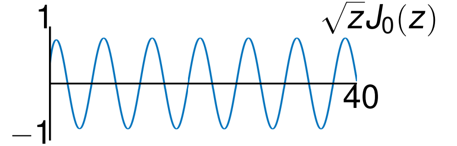
3. Time samples. $j_{0,1}/j_{0,N+1},\ldots,j_{0,N}/j_{0,N+1}$ are **not** equispaced.

# Introduction

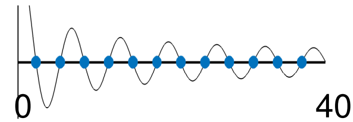
## Three challenging differences between the DFT and DHT

The discrete Hankel transform of order 0 (DHT) [Johnson, 86]:

$$\underline{X} = \mathbf{J}_0(\underline{j}_0 \underline{j}_0^{\mathsf{T}}/j_{0,N+1})\,\underline{x}.$$




1. Kernel. $J_0(z)$ does **not** have constant amplitude.



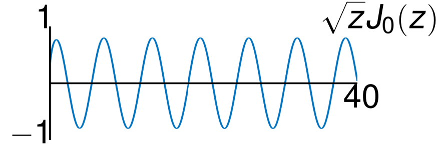

2. Frequencies. $j_{0,1}, \ldots, j_{0,N}$ are **not** equispaced.

3. Time samples. $j_{0,1}/j_{0,N+1}, \ldots, j_{0,N}/j_{0,N+1}$ are **not** equispaced.

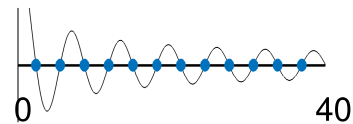
The discrete Hankel transform of order 0 (DHT) [Johnson, 86]:

$$\underline{X} = \mathbf{J}_0(\underline{j}_0 \underline{j}_0^{\mathsf{T}} / j_{0,N+1})\, \underline{x}.$$

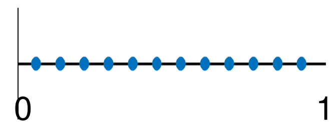
1. Kernel. $J_0(z)$ does **not** have constant amplitude.

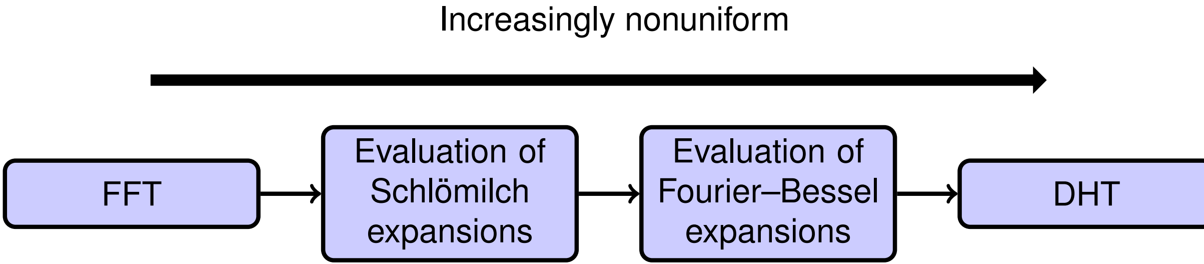



2. Frequencies. $j_{0,1}, \ldots, j_{0,N}$ are **not** equispaced.




3. Time samples. $j_{0,1}/j_{0,N+1}, \ldots, j_{0,N}/j_{0,N+1}$ are **not** equispaced.

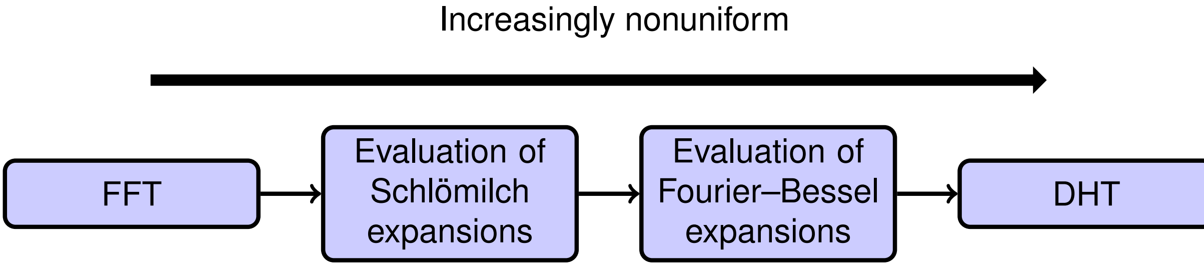





DFT $\qquad \mathbf{J_0}(\underline{r}\,\underline{\omega}^\intercal) \qquad \mathbf{J_0}(\underline{r}\,\underline{j}_0^\intercal) \qquad \mathbf{J_0}(\underline{j}_0\,\underline{j}_0^\intercal/j_{0,N+1})$

$$r_k = k/N, \qquad \omega_n = n\pi, \qquad j_{0,n} = \text{Bessel roots.}$$

**Goal:** FFT-based and analysis-based fast transforms with zero offline cost.




DFT ✓ $\qquad \mathbf{J_0}(\underline{r}\,\underline{\omega}^{\intercal}) \qquad \mathbf{J_0}(\underline{r}\,\underline{j}_0^{\intercal}) \qquad \mathbf{J_0}(\underline{j}_0\,\underline{j}_0^{\intercal}/j_{0,N+1})$

$r_k = k/N, \qquad \omega_n = n\pi, \qquad j_{0,n} =$ Bessel roots

**Goal:** FFT-based and analysis-based fast transforms with zero offline cost.


Oscar Schlömilch

Schlömilch expansion of $f : [0,1] \to \mathbb{C}$ is [Schlömilch, 1846]:

$$f(r) = \sum_{n=1}^{N} x_n J_0(n\pi r).$$

**Aim:** A fast algorithm for evaluating $f(r)$ at $r_k = k/N$ for $1 \leqslant k \leqslant N$.

Equivalent to the following matrix-vector product:

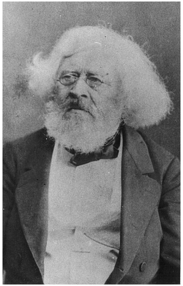
$$\mathbf{J_0}(\underline{r}\,\underline{\omega}^T)\underline{x} = \begin{pmatrix} J_0(r_1\omega_1) & \dots & J_0(r_1\omega_N) \\ \vdots & \ddots & \vdots \\ J_0(r_N\omega_1) & \dots & J_0(r_N\omega_N) \end{pmatrix} \begin{pmatrix} x_1 \\ \vdots \\ x_N \end{pmatrix}, \quad r_k = k/N, \quad \omega_n = n\pi.$$

# Fast evaluation of Schlömilch expansions

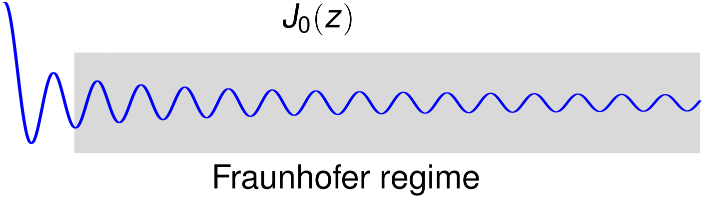
## Bessel functions are trigonometric-like for large argument

For fixed $M \geqslant 1$ and real $z \to \infty$:

$$J_0(z) \sim \sqrt{\frac{2}{\pi z}}\left(\cos(z - \tfrac{\pi}{4})\sum_{m=0}^{M-1}\frac{(-1)^m a_{2m}}{z^{2m}} - \sin(z - \tfrac{\pi}{4})\sum_{m=0}^{M-1}\frac{(-1)^m a_{2m+1}}{z^{2m+1}}\right).$$



Peter A. Hansen



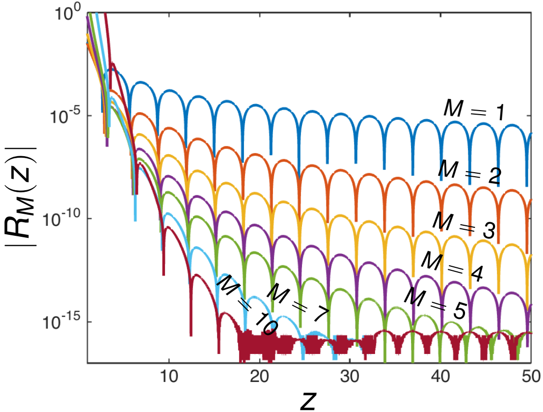

Hermann Hankel

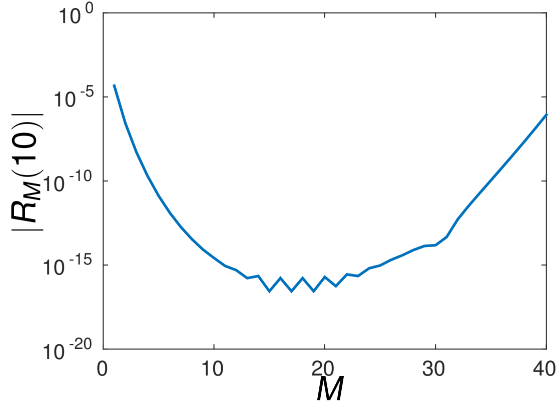
A 19th century hobby: [Poisson, 1823], [Kummer, 1837], [Jacobi, 1843], [Lipschitz, 1859], [Hamilton, 1907]

# Fast evaluation of Schlömilch expansions

## Numerical pitfalls of an asymptotic expansionist

$$J_0(z) = \sqrt{\frac{2}{\pi z}}\left(\cos(z - \tfrac{\pi}{4})\sum_{m=0}^{M-1}\frac{(-1)^m a_{2m}}{z^{2m}} - \sin(z - \tfrac{\pi}{4})\sum_{m=0}^{M-1}\frac{(-1)^m a_{2m+1}}{z^{2m+1}}\right) + R_M(z)$$




Fix $M$. Where is the asymptotic expansion accurate?

Evaluation of a Schlömilch expansion is equivalent to the following matrix-vector product:

$$\mathbf{J_0}(\underline{r}\,\underline{\omega}^T)\underline{x} = \begin{pmatrix} J_0(r_1\omega_1) & \dots & J_0(r_1\omega_N) \\ \vdots & \ddots & \vdots \\ J_0(r_N\omega_1) & \dots & J_0(r_N\omega_N) \end{pmatrix} \begin{pmatrix} x_1 \\ \vdots \\ x_N \end{pmatrix}, \quad r_k = k/N, \quad \omega_n = n\pi.$$

# Fast evaluation of Schlömilch expansions
## Asymptotic expansions as a matrix decomposition

The asymptotic expansion gives a matrix decomposition:

$$J_0(r_k\omega_n) = \sqrt{\frac{2}{\pi}}\left(\cos(r_k\omega_n - \tfrac{\pi}{4})\sum_{m=0}^{M-1}\frac{(-1)^m a_{2m}}{(r_k\omega_n)^{2m+\frac{1}{2}}} - \sin(r_k\omega_n - \tfrac{\pi}{4})\sum_{m=0}^{M-1}\frac{(-1)^m a_{2m+1}}{(r_k\omega_n)^{2m+\frac{3}{2}}}\right) + R_M(r_k\omega_n)$$

which is a sum of diagonally scaled DCTs and DSTs when $r_k = \frac{k}{N}$ and $\omega_n = n\pi$:

$$\mathbf{J_0}(\underline{r}\,\underline{\omega}^T) = \sum_{m=0}^{M-1}\left(D_{u_m^1} P^T \mathbf{C}_{N+1} P D_{u_m^2} + D_{v_m^1}\begin{bmatrix}\mathbf{S}_{N-1} & 0\\ 0 & 0\end{bmatrix} D_{v_m^2}\right) + \mathbf{R}_M.$$

$$\mathbf{J_0} = \mathbf{J_0}^{\text{ASY}} + \mathbf{R}_M$$

# Fast evaluation of Schlömilch expansions

## Asymptotic expansions as a matrix decomposition

The asymptotic expansion gives a matrix decomposition:

$$J_0(r_k\omega_n)=\sqrt{\frac{2}{\pi}}\left(\cos(r_k\omega_n-\tfrac{\pi}{4})\sum_{m=0}^{M-1}\frac{(-1)^m a_{2m}}{(r_k\omega_n)^{2m+\frac{1}{2}}}-\sin(r_k\omega_n-\tfrac{\pi}{4})\sum_{m=0}^{M-1}\frac{(-1)^m a_{2m+1}}{(r_k\omega_n)^{2m+\frac{3}{2}}}\right)+R_M(r_k\omega_n)$$

which is a sum of diagonally scaled DCTs and DSTs when $r_k=\frac{k}{N}$ and $\omega_n=n\pi$:

$$\mathbf{J_0}(\underline{r}\,\underline{\omega}^T)=\sum_{m=0}^{M-1}\left(D_{u_m^1}P^T\mathbf{C}_{N+1}PD_{u_m^2}+D_{v_m^1}\begin{bmatrix}\mathbf{S}_{N-1} & 0\\ 0 & 0\end{bmatrix}D_{v_m^2}\right)+\mathbf{R}_M.$$

# Fast evaluation of Schlömilch expansions

## Asymptotic expansions as a matrix decomposition

The asymptotic expansion gives a matrix decomposition:

$$J_0(r_k\omega_n) = \sqrt{\frac{2}{\pi}}\left(\cos(r_k\omega_n - \tfrac{\pi}{4})\sum_{m=0}^{M-1}\frac{(-1)^m a_{2m}}{(r_k\omega_n)^{2m+\frac{1}{2}}} - \sin(r_k\omega_n - \tfrac{\pi}{4})\sum_{m=0}^{M-1}\frac{(-1)^m a_{2m+1}}{(r_k\omega_n)^{2m+\frac{3}{2}}}\right) + R_M(r_k\omega_n)$$

which is a sum of diagonally scaled DCTs and DSTs when $r_k = \frac{k}{N}$ and $\omega_n = n\pi$:

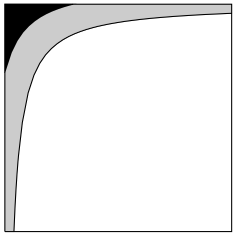
$$\mathbf{J_0}(\underline{r}\,\underline{\omega}^T) = \sum_{m=0}^{M-1}\left(D_{u_m^1} P^T \mathbf{C}_{N+1} P D_{u_m^2} + D_{v_m^1}\begin{bmatrix}\mathbf{S}_{N-1} & 0\\ 0 & 0\end{bmatrix} D_{v_m^2}\right) + \mathbf{R}_M.$$

$$\mathbf{J_0} = \mathbf{J_0}^{\mathrm{ASY}} + \mathbf{R}_M$$

# Fast evaluation of Schlömilch expansions

## Be careful and stay safe

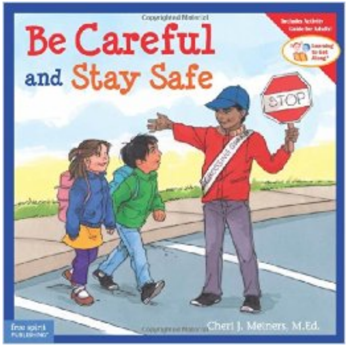
Error curve: $|R_M(r_k \omega_n)| = \epsilon$

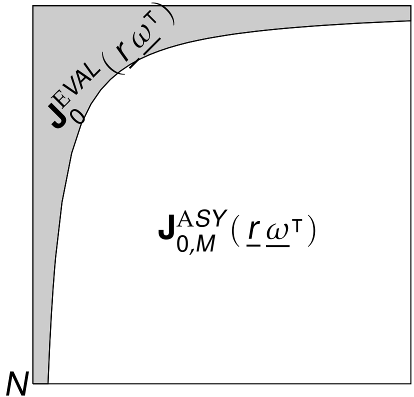
$\mathbf{R}_M(\underline{r}\,\underline{\omega}^\intercal) =$ 

$$n\,k \leqslant \frac{N}{\pi} s_{0,M}(\epsilon) \quad \Longrightarrow \quad |R_M(r_k \omega_n)| < \epsilon$$

Computed by a fixed-point iteration

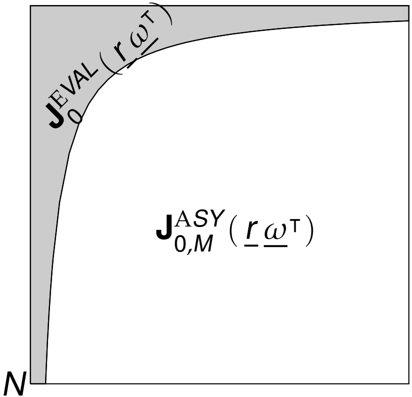
Fix $M$. Where is the asymptotic expansion accurate?



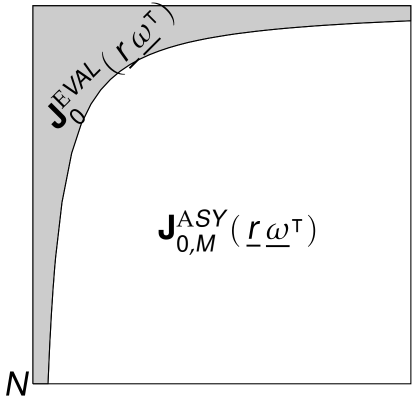



**Theorem**

*The matrix-vector product $\underline{X} = \mathbf{J_0}(\underline{r}\,\underline{\omega}^T)\underline{x}$ can be computed in $\mathcal{O}(N(\log N)^2/\log\log N)$ operations.*



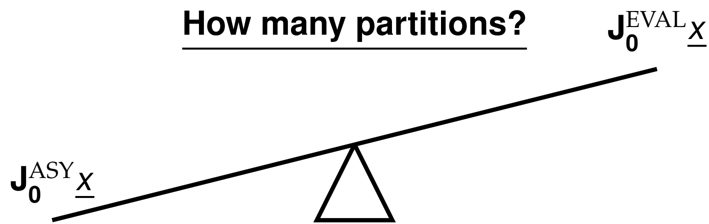

### Theorem
*The matrix-vector product* $\underline{X} = \mathbf{J_0}(\underline{r}\,\underline{\omega}^T)\underline{x}$ *can be computed in* $\mathcal{O}(N(\log N)^2/\log\log N)$ *operations.*




### Theorem

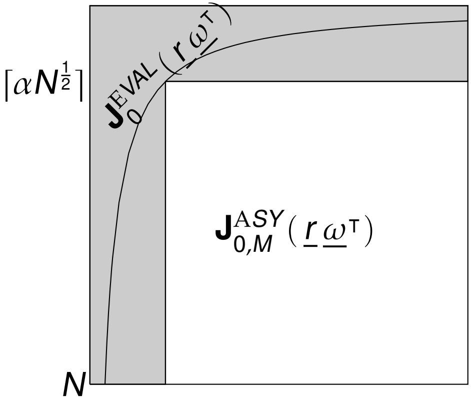
*The matrix-vector product* $\underline{X} = \mathbf{J_0}(\underline{r}\,\underline{\omega}^T)\underline{x}$ *can be computed in* $\mathcal{O}(N(\log N)^2/\operatorname{loglog} N)$ *operations.*

**How many partitions?**



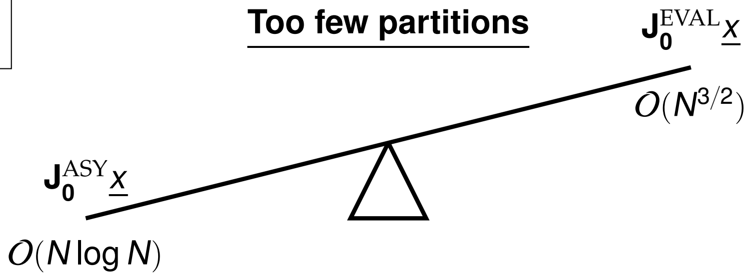
# Fast evaluation of Schlömilch expansions

## Partitioning and balancing competing costs



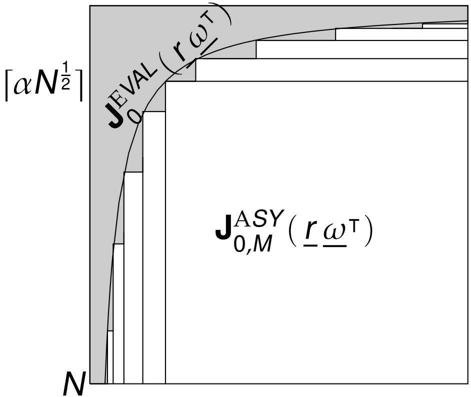

**Theorem**

*The matrix-vector product $\underline{X} = \mathbf{J_0}(\underline{r}\,\underline{\omega}^T)\underline{x}$ can be computed in $\mathcal{O}(N(\log N)^2/\operatorname{loglog} N)$ operations.*



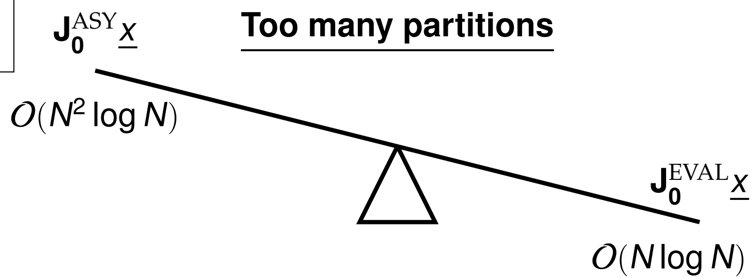
# Fast evaluation of Schlömilch expansions

## Partitioning and balancing competing costs



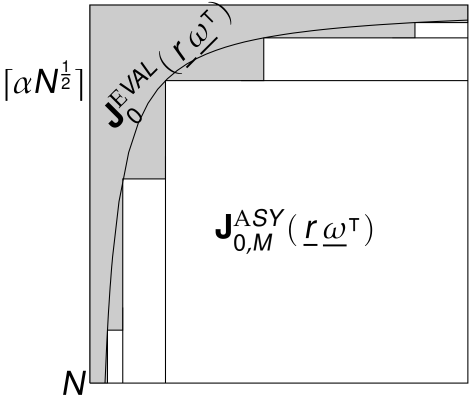

**Theorem**

*The matrix-vector product $\underline{X} = \mathbf{J_0}(\underline{r}\,\underline{\omega}^T)\underline{x}$ can be computed in $\mathcal{O}(N(\log N)^2/\log\log N)$ operations.*



# Fast evaluation of Schlömilch expansions

## Partitioning and balancing competing costs




### Theorem

*The matrix-vector product $\underline{X} = \mathbf{J_0}(\underline{r}\,\underline{\omega}^T)\underline{x}$ can be computed in $\mathcal{O}(N(\log N)^2/\operatorname{loglog} N)$ operations.*

$\mathcal{O}(\log N/\log\log N)$ **partitions**

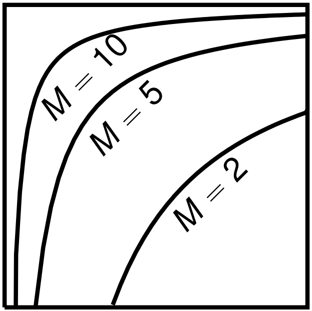
$\mathbf{J}_\mathbf{0}^{\mathrm{ASY}}\underline{x}$ $\qquad$ $\mathbf{J}_\mathbf{0}^{\mathrm{EVAL}}\underline{x}$

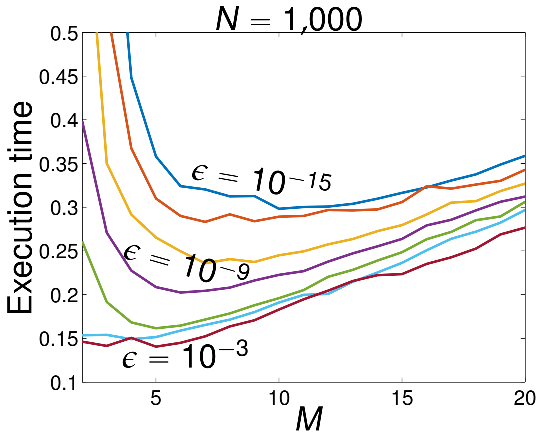
$\mathcal{O}(N(\log N)^2/\operatorname{loglog} N)$ $\qquad$ $\mathcal{O}(N(\log N)^2/\operatorname{loglog} N)$

Fix $M$. But, what $M$ should we pick?

Pick the working tolerance: $\epsilon > 0$:

$\mathbf{R}_M(\underline{r}\,\underline{\omega}^\intercal) =$



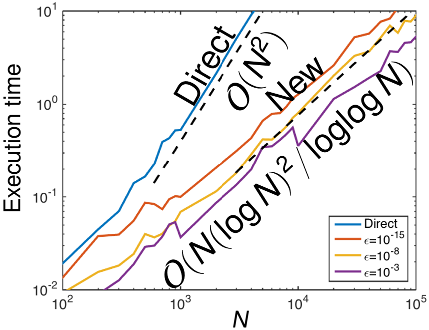

For computational efficiency:

$$M = \max(\lfloor 0.3 \log(1/\epsilon) \rfloor, 3)$$

- Other algorithmic parameters are decided upon by precise error bounds.

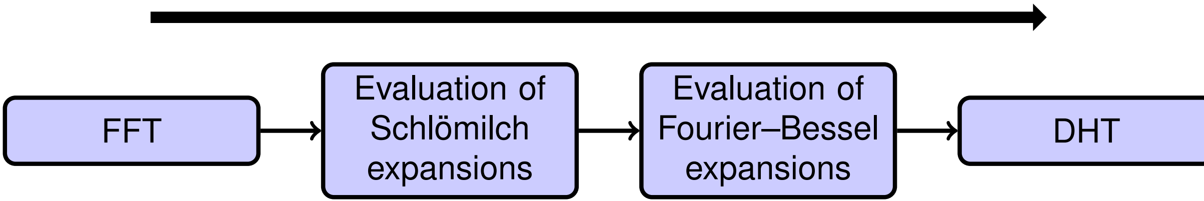
## Analysis-based algorithmic parameters

- A handful of FFTs
- Zero offline cost
- Analysis-based → guaranteed accuracy




| Parameter | Short description | Formula |
|---|---|---|
| $M$ | Number of terms in asy | $\max(\lfloor 0.3\log(1/\epsilon)\rfloor, 3)$ |
| $s_{\nu,M}(\epsilon)$ | $z \geqslant s_{\nu,M}(\epsilon) \Rightarrow \lvert R_{\nu,M}(z)\rvert \leqslant \epsilon$ | [Table 4.1, T., 15] |
| $\alpha$ | Partitioning parameter | $(s_{\nu,M}(\epsilon)/\pi)^{1/2}$ |
| $\beta$ | Refining parameter | $\min(3/\log N, 1)$ |
| $P$ | Number of partitions | $\lceil \log(30\alpha^{-1}N^{-1/2})/\log\beta \rceil$ |

Increasingly nonuniform




DFT ✓ $\qquad \mathbf{J_0}(\,\underline{r}\,\underline{\omega}^\intercal)$ ✓ $\qquad \mathbf{J_0}(\,\underline{r}\,\underline{j}_0^\intercal) \qquad \mathbf{J_0}(\,\underline{j}_0\,\underline{j}_0^\intercal/j_{0,N+1})$

$r_k = k/N, \qquad \omega_n = n\pi, \qquad j_{0,n} =$ Bessel roots

**Goal:** FFT-based and analysis-based fast transforms with zero offline cost.

# Fast evaluation of Fourier–Bessel expansions
## Setup

Fourier–Bessel expansion of $f : [0, 1] \to \mathbb{C}$ is:

$$f(r) = \sum_{n=1}^{N} x_n J_0(j_{0,n} r).$$

**Aim:** A fast algorithm for evaluating $f(r)$ at $r_k = k/N$ for $1 \leqslant k \leqslant N$.

Equivalent to the following matrix-vector product:

$$\mathbf{J_0}(\underline{r}\,\underline{j}_0^{\mathsf{T}})\underline{x} = \begin{pmatrix} J_0(r_1 j_{0,1}) & \dots & J_0(r_1 j_{0,N}) \\ \vdots & \ddots & \vdots \\ J_0(r_N j_{0,1}) & \dots & J_0(r_N j_{0,N}) \end{pmatrix} \begin{pmatrix} x_1 \\ \vdots \\ x_N \end{pmatrix}, \quad r_k = k/N, \quad J_0(j_{0,n}) = 0.$$

**Idea:** Bessel roots are nearly equispaced.

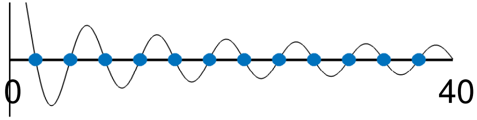
## Lemma (Theorem 3 in [Hethcote, 70].)

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

$$\left| j_{0,n} - \left(n - \frac{1}{4}\right)\pi \right| \leqslant \frac{1}{8(n - \frac{1}{4})\pi}.$$

We can write:

$$j_{0,n} = \underbrace{(n - 1/4)\pi}_{\tilde{\omega}_n} + b_n, \quad |b_n| \leqslant \frac{1}{8(n - \frac{1}{4})\pi}$$



### Theorem

*For integers $K \geqslant 1$ and $T \geqslant 1$ we have*

$$\mathbf{J_0}(\underline{r}\,\underline{j}_0^\intercal) = \mathbf{J_0}(\underline{r}\,\underline{\tilde{\omega}}^\intercal + \underline{r}\,\underline{b}^\intercal) = \sum_{s=-K+1}^{K-1}\sum_{t=0}^{T-1} \frac{(-1)^t 2^{-2t+s}}{t!(t-s)!} \overbrace{D_{\underline{r}}^{2t-s}\mathbf{J_s}(\underline{r}\,\underline{\tilde{\omega}}^\intercal)D_{\underline{b}}^{2t-s}}^{\textit{diag-weighted Schlömilch}} + \mathbf{E}_{K,T}.$$

### Proof.

Neumann addition formula and Taylor expansion of $J_0(z)$ about $z = 0$. □

**Rearrange computation:**

For $\epsilon = 10^{-15}$, $T$ and $K$ are:

$$K = 6, \qquad T = 3.$$

Looks like

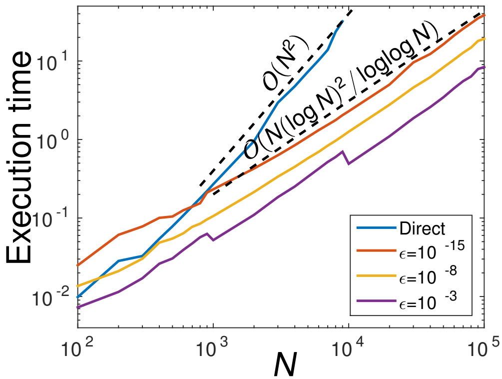
$$\text{Fourier-Bessel} = \underbrace{(2K-1)T}_{=33} \times \text{Schlömilch}$$

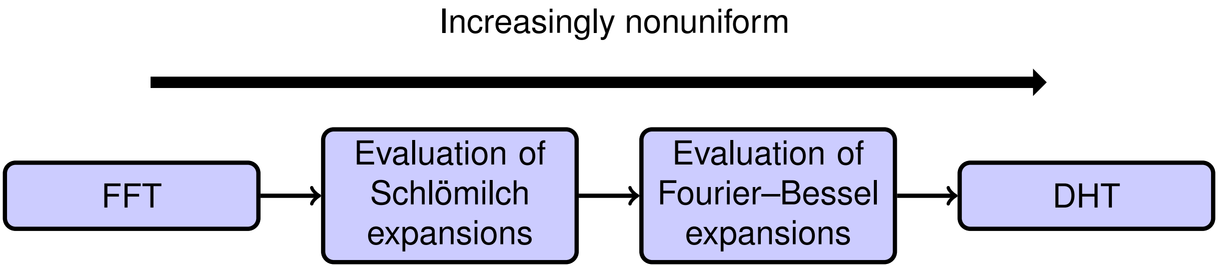
But, after rearranging sum:

$$\text{Fourier-Bessel} = \underbrace{(2T+K-2)}_{=10} \times \text{Schlömilch}$$

**Fourier–Bessel expansions:**

$$X_k = \sum_{n=1}^{N} x_n J_0(j_{0,n}k/N), \qquad 1 \leqslant k \leqslant N.$$






DFT ✓  $\mathbf{J_0}(\,\underline{r}\,\underline{\omega}^{\intercal})$ ✓  $\mathbf{J_0}(\,\underline{r}\,\underline{j}_0^{\intercal})$ ✓  $\mathbf{J_0}(\,\underline{j}_0\,\underline{j}_0^{\intercal}/j_{0,N+1})$

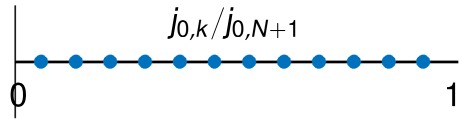
$r_k = k/N, \qquad \omega_n = n\pi, \qquad j_{0,n} =$ Bessel roots

**Goal:** FFT-based and analysis-based fast transforms with zero offline cost.

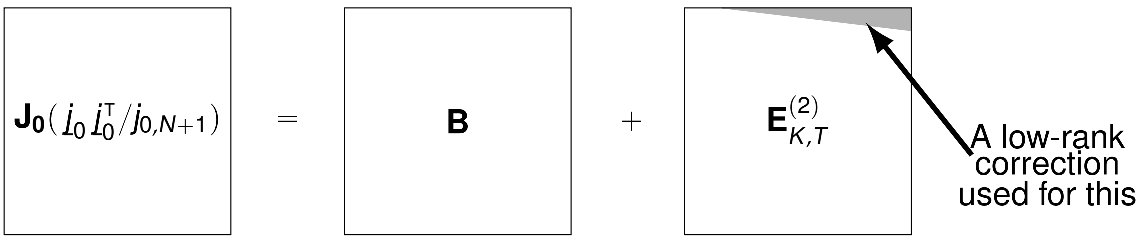
# The discrete Hankel transform
## Discrete Hankel transform

Let $j_{0,k}$ denote the $n$th positive root of $J_0(z)$. The ratios are nearly equispaced:

$$\left|\frac{j_{0,k}}{j_{0,N+1}} - \frac{k-\frac{1}{4}}{N+\frac{3}{4}}\right| \leqslant \frac{1}{8(N+\frac{3}{4})(k-\frac{1}{4})\pi^2}.$$




In a similar manner to the Fourier–Bessel case, we have



# The discrete Hankel transform
## Numerical results

For $\epsilon = 10^{-15}$, we select:

$$K = 6, \qquad T = 3.$$

We have

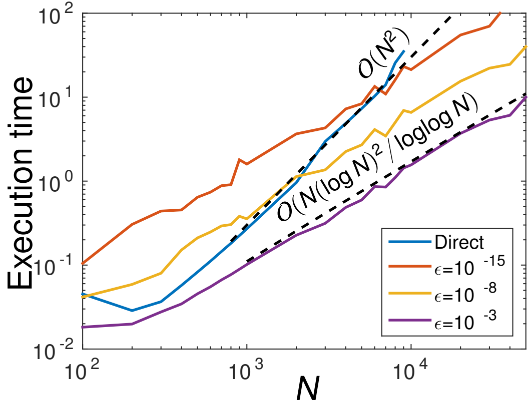
$$\text{DHT} = \underbrace{(2T + K - 2)}_{=10} \times \text{Fourier–Bessel}$$

and

$$\text{DHT} = \underbrace{(2T + K - 2)^2}_{=100} \times \text{Schlömilch}$$

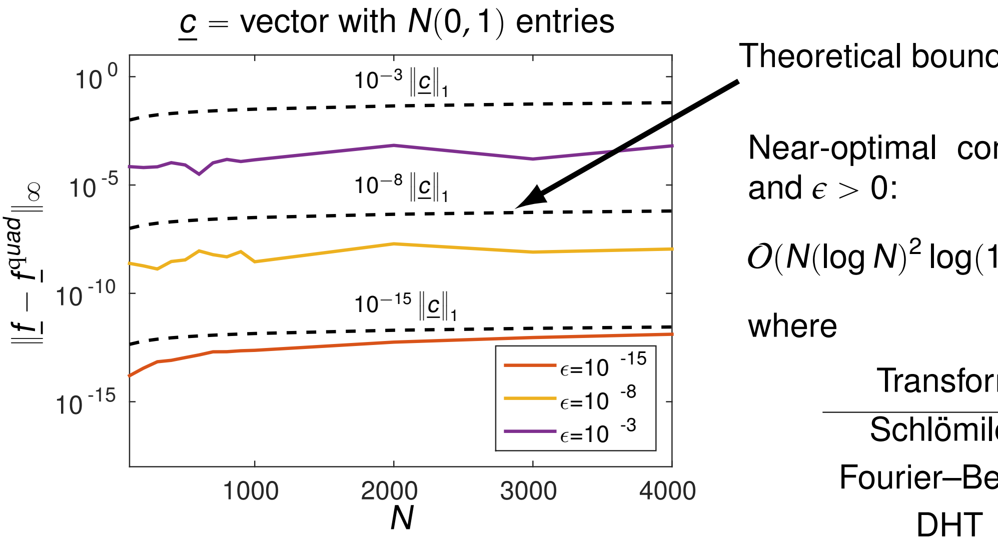
**Discrete Hankel transform**

$$X_k = \sum_{n=1}^{N} x_n J_0(j_{0,n} j_{0,k} / j_{0,N+1}), \qquad 1 \leqslant k \leqslant N.$$



# The discrete Hankel transform

## Errors in practice versus theory



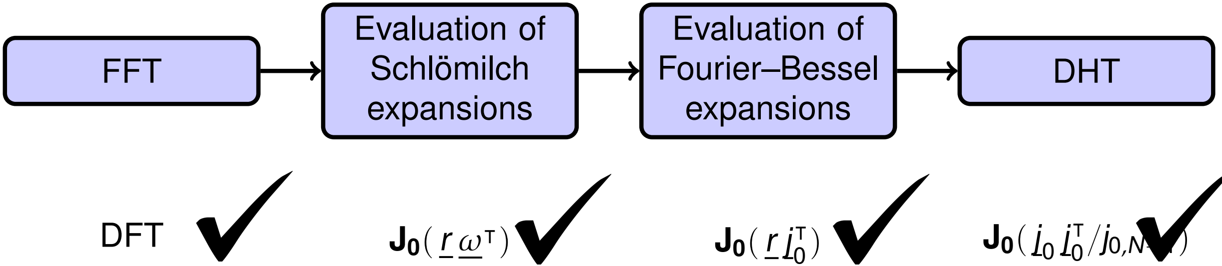

Near-optimal complexity in $N$ and $\epsilon > 0$:

$$\mathcal{O}(N(\log N)^2 \log(1/\epsilon)^p / \log\log N),$$

where

| Transform | $p$ |
|---|---|
| Schlömilch | 1 |
| Fourier–Bessel | 2 |
| DHT | 3 |

# Conclusion




**Novel features of our algorithm for the DHT:**

1. Zero offline cost.
2. Implementation adapts to individual computer architecture.
3. Guaranteed error control (no heuristics).
4. No hierarchical data structures.

# Thank you

Thank you

More information in: T., “A fast analysis-based discrete Hankel transform using asymptotic expansions”, to appear in SINUM.

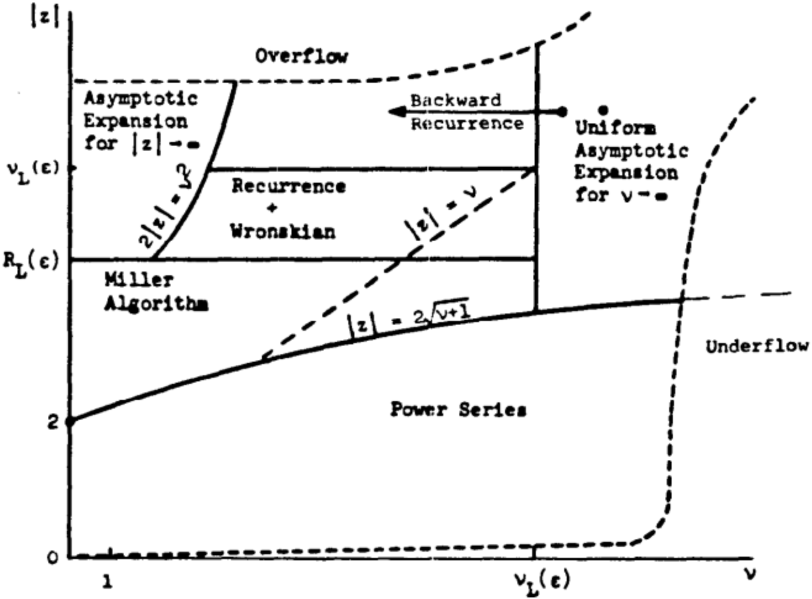
Code publicly available from:
`https://github.com/ajt60gaibb/FastAsyTransforms`

# References


- M. J. Cree and P. J. Bones, Algorithms to numerically evaluate the Hankel transform, J. Franklin Institute, 1983.
- H. F. Johnson, An improved method for computing a discrete Hankel transform, Computer Physics Communications, 1987.
- N. Hale and A. Townsend, A fast, simple, and stable Chebyshev–Legendre transform using an asymptotic formula, SISC, 2014.
- M. O'Neil, F. Woolfe, and V. Rokhlin, An algorithm for the rapid evaluation of special function transforms, Appl. Comput. Harm. Anal., 2010.
- A. Townsend, The race to compute high-order Gauss quadrature, SIAM News, 2015.
- A. Townsend, A fast analysis-based discrete Hankel transform using asymptotics expansions, to appear in SINUM, 2015.
- M. Tygert, Fast algorithms for spherical harmonic expansions, II, J. Comput. Phys., 2008.

## Direct approach [Amos, 86]

**Compute $\mathbf{J}_\nu(\mathbf{z})$ :**



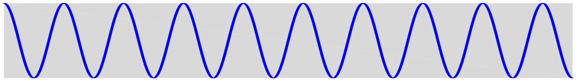

Lots going on!

# Extra slides

## Many special functions are trigonometric-like

Trigonometric functions

$\cos(\omega x)$, $\sin(\omega x)$



Chebyshev polynomials

$T_n(x)$

Legendre polynomials

$P_n(x)$

Bessel functions

$J_\nu(z)$

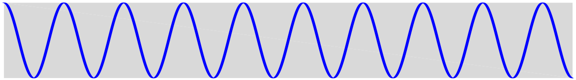
Airy functions

$Ai(x)$

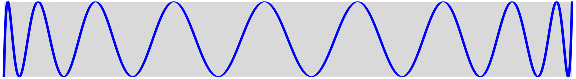
Also, Jacobi polynomials, Hermite polynomials, cylinder functions, etc.

## Many special functions are trigonometric-like

Trigonometric functions
$\cos(\omega x)$, $\sin(\omega x)$



Chebyshev polynomials
$T_n(x)$



Legendre polynomials
$P_n(x)$

Bessel functions
$J_\nu(z)$

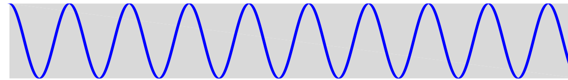
Airy functions
$Ai(x)$

Also, Jacobi polynomials, Hermite polynomials, cylinder functions, etc.

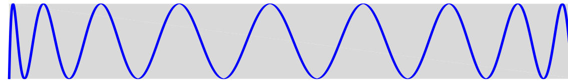
# Extra slides

## Many special functions are trigonometric-like

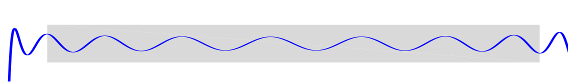
Trigonometric functions
$\cos(\omega x)$, $\sin(\omega x)$



Chebyshev polynomials
$T_n(x)$



Legendre polynomials
$P_n(x)$



Bessel functions
$J_\nu(z)$

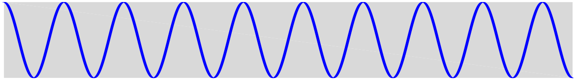
Airy functions
$Ai(x)$

Also, Jacobi polynomials, Hermite polynomials, cylinder functions, etc.

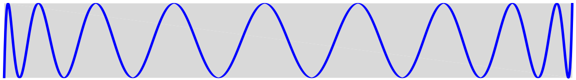
# Extra slides

## Many special functions are trigonometric-like

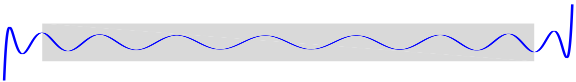
Trigonometric functions
$\cos(\omega x)$, $\sin(\omega x)$



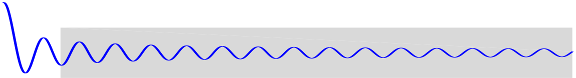
Chebyshev polynomials
$T_n(x)$



Legendre polynomials
$P_n(x)$



Bessel functions
$J_\nu(z)$



Airy functions
$Ai(x)$

Also, Jacobi polynomials, Hermite polynomials, cylinder functions, etc.

# Extra slides
## Many special functions are trigonometric-like

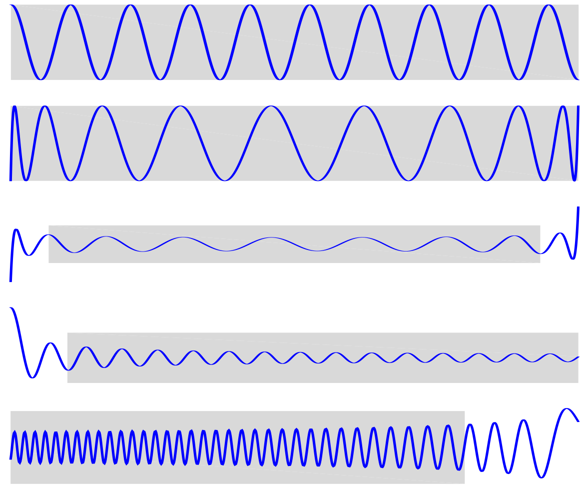
Trigonometric functions
$\cos(\omega x)$, $\sin(\omega x)$

Chebyshev polynomials
$T_n(x)$

Legendre polynomials
$P_n(x)$

Bessel functions
$J_\nu(z)$

Airy functions
$Ai(x)$

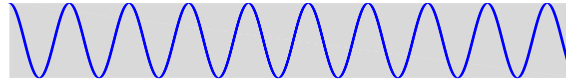


Also, Jacobi polynomials, Hermite polynomials, cylinder functions, etc.

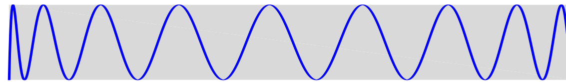
# Extra slides

## Many special functions are trigonometric-like

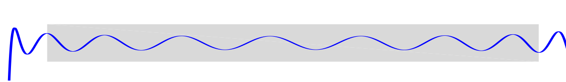
Trigonometric functions
$\cos(\omega x)$, $\sin(\omega x)$



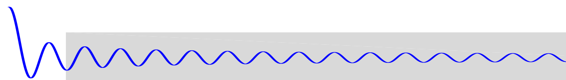
Chebyshev polynomials
$T_n(x)$



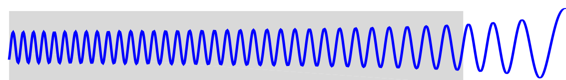
Legendre polynomials
$P_n(x)$



Bessel functions
$J_\nu(z)$



Airy functions
$Ai(x)$



Also, Jacobi polynomials, Hermite polynomials, cylinder functions, etc.