FAST EXPANSION INTO HARMONICS ON THE DISK: A STEERABLE BASIS WITH FAST RADIAL CONVOLUTIONS

NICHOLAS F. MARSHALL, OSCAR MICKELIN, AND AMIT SINGER

ABSTRACT. We present a fast and numerically accurate method for expanding digitized $L \times L$ images representing functions on $[-1,1]^2$ supported on the disk $\{x \in \mathbb{R}^2 : |x| < 1\}$ in the harmonics (Dirichlet Laplacian eigenfunctions) on the disk. Our method runs in $\mathcal{O}(L^2 \log L)$ operations. This basis is also known as the Fourier-Bessel basis and it has several computational advantages: it is orthogonal, ordered by frequency, and steerable in the sense that images expanded in the basis can be rotated by applying a diagonal transform to the coefficients. Moreover, we show that convolution with radial functions can also be efficiently computed by applying a diagonal transform to the coefficients.

1. INTRODUCTION

1.1. **Motivation.** Decomposing a function into its Fourier series can be viewed as representing a function in the eigenfunctions of the Laplacian on the torus $\mathbb{T} := [0, 2\pi]$ where 0 and 2π are identified. Indeed,

$$-\Delta e^{\imath k x} = k^2 e^{\imath k x}.$$

The eigenfunctions of the Laplacian (harmonics) on the disk $\{x \in \mathbb{R}^2 : |x| < 1\}$ that satisfy the Dirichlet boundary conditions can be written in polar coordinates $(r, \theta) \in [0, 1) \times [0, 2\pi)$ as

$$\psi_{nk}(r, \theta) = c_{nk} J_n(\lambda_{nk} r) e^{\imath n \theta}, \tag{1}$$

where c_{nk} is a normalization constant, J_n is the n-th order Bessel function of the first kind, and λ_{nk} is the k-th smallest positive root of J_n. The indices run over $(n, k) \in \mathbb{Z} \times \mathbb{Z}_{>0}$. The functions ψ_{nk} satisfy

$$-\Delta \psi_{nk} = \lambda_{nk}^2 \psi_{nk}. \tag{2}$$

In this paper, we present a fast and accurate transform of digitized $L \times L$ images into this eigenfunction basis, often referred to as the Fourier-Bessel basis. For computational purposes, this basis is convenient for a number of reasons:

(i) *Orthonormal:* these eigenfunctions are an orthonormal basis for square integrable functions on the disk.
(ii) *Ordered by frequency:* the basis functions are ordered by eigenvalues, which can be interpreted as frequencies due to the connection with the Laplacian and Fourier series described above; therefore, low-pass filtering can be performed by retaining basis coefficients up to a given threshold.

N.F.M. was supported in part by NSF DMS-1903015. A.S. was supported in part by AFOSR FA9550-20-1-0266, the Simons Foundation Math+X Investigator Award, NSF BIGDATA Award IIS-1837992, NSF DMS-2009753, and NIH/NIGMS 1R01GM136780-01.

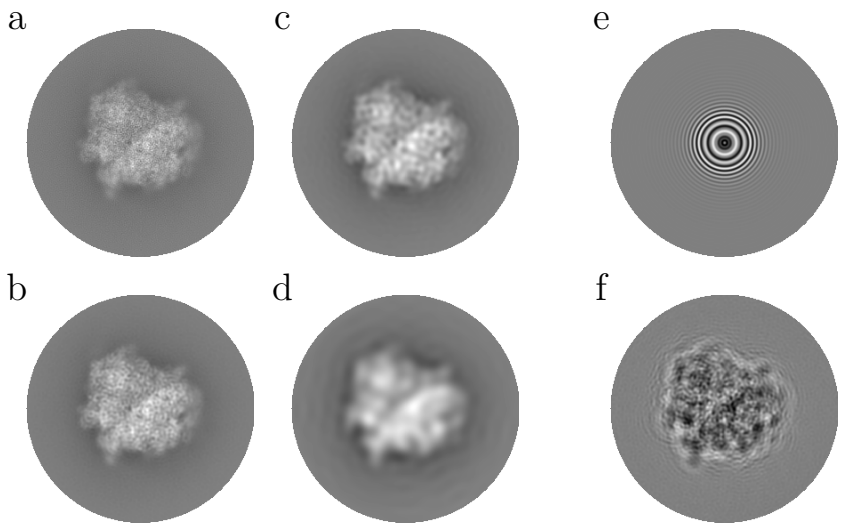

FIGURE 1. Illustration of method for $L \times L$ images with $L = 256$. Original image (a), low-pass filter of the original image using a decreasing number of basis functions (b–d), radial function (e), convolution of original image with radial function (f).

(iii) *Steerable:* functions expanded in the basis can be rotated by applying a diagonal transform corresponding to phase modulation of the coefficients.
(iv) *Fast radial convolutions:* we show that the convolution with radial functions can be computed by applying a diagonal transform to the coefficients.

Our method involves $\mathcal{O}(L^2 \log L)$ operations and has precise accuracy guarantees. Python code that implements our method is publicly available online[1]. To the best of our knowledge, existing methods for computing the expansion coefficients in a steerable basis either require $\mathcal{O}(L^3)$ operations [19, 20] or lack accuracy guarantees [34, 35].

Steerable bases have been utilized in numerous image processing problems including image alignment [25], image classification [35] and image denoising [34], including applications to machine learning [5, 7, 33] and data-driven science, such as applications to cryo-electron microscopy (cryo-EM) [6, 23], and computer vision [24], among other areas.

There are many possible choices of steerable bases, for instance prolate spheroidal wave functions which are bandlimited functions optimally supported on the disk [19, 20, 28], or Zernike polynomials which are widely used in optics [31]. The harmonics on the disk (which satisfy Dirichlet boundary conditions) [34, 35] are one natural choice due to their orthogonality, ordering by frequency, and fast radial convolution.

We illustrate the frequency ordering property of the Laplacian eigenbasis by performing a low-pass filter by projecting onto the span of eigenfunctions whose eigenvalues are below a sequence of bandlimits that decrease the number of basis functions successively by factors of four, starting from 39593 coefficients. Since the

[1]An implementation is available at `https://github.com/nmarshallf/fle_2d`.

basis is orthonormal this is equivalent to setting coefficients above the bandlimit equal to zero; see Fig. 1 (a–d). Further, we demonstrate the radial convolution property by illustrating the convolution with a point spread function, which is a function used in computational microscopy [32]; see Fig. 1 (e–f). The image used for this example is a tomographic projection of a 3-D density map representing a bio-molecule (E. coli 70S ribosome) [26].

1.2. **Notation.** Suppose that f is an $L \times L$ image representing a function $\tilde{f} : [-1,1]^2 \to \mathbb{R}$ that is supported on the unit disk $\{x \in \mathbb{R}^2 : |x| < 1\}$. More precisely, we assume that

$$f(j_1, j_2) = \tilde{f}(x_{j_1,j_2}), \quad \text{where} \quad x_{j_1,j_2} := \frac{1}{\lfloor L/2 \rfloor} \left(j_1 - \lfloor L/2 \rfloor, j_2 - \lfloor L/2 \rfloor\right),$$

for $j_1, j_2 = 1, \ldots, L$. Let $p := L^2$ be the number of pixels in the image f, and let

$$x_1, \ldots, x_p \quad \text{be an enumeration of} \quad \{x_{j_1,j_2} : j_1, j_2 = 1, \ldots, L\}.$$

We can consider the $L \times L$ image f as a vector $f \in \mathbb{R}^p$ whose j-th entry is defined by

$$f_j = \tilde{f}(x_j).$$

Similarly, for a fixed bandlimit parameter λ, let

$$\psi_1, \ldots, \psi_m \quad \text{be an enumeration of} \quad \{\psi_{nk} : \lambda_{nk} \leq \lambda\},$$

where ψ_{nk} is defined in (1) above. With this notation, coefficients of a function in the basis of eigenfunctions up to bandlimit λ is a vector in $\mathbb{R}^m$. By Weyl's law, see for example [17], we have $m \sim \lambda^2/4$.

In the following, we consider a linear transformation B which maps coefficients in $\mathbb{R}^m$ to images in $\mathbb{R}^p$, and its adjoint B^* which maps images in $\mathbb{R}^p$ to coefficients in $\mathbb{R}^m$.

1.3. **Main result.** We consider the linear transform $B : \mathbb{R}^m \to \mathbb{R}^p$ which maps coefficients to images by

$$(Ba)_j = \sum_{i=1}^{m} a_i \psi_i(x_j) h, \tag{3}$$

and its adjoint transform $B^* : \mathbb{R}^p \to \mathbb{R}^m$ which maps images to coefficients by

$$(B^* f)_i = \sum_{j=1}^{p} f_j \overline{\psi_i(x_j)} h, \tag{4}$$

where $h = 1/\lfloor (\sqrt{p} - 1)/2 \rfloor$ is a normalization constant, chosen such that B^*B is close to the identity. In this paper, we present a fast and accurate method to apply B and B^*. In particular, the presented method can apply the operators B and B^* to vectors in $\mathcal{O}(p \log p)$ operations for any fixed relative error ε. We again emphasize that this is in contrast to previous results, since to the best of our knowledge, existing methods for computing the expansion coefficients in a steerable basis either require $\mathcal{O}(p^{3/2})$ operations [19, 20] or lack accuracy guarantees [34, 35]. The application of the operators B and B^* can be used in an iterative method to determine least squares optimal expansion coefficients for a given image. Alternatively, applying B^* to f can be viewed as estimating the continuous inner products that define the coefficients by using quadrature points on a grid (and potentially quadrature weights to provide an endpoint correction).

We report the relative errors of our fast method for applying B and B^* compared to matrix multiplication for images with up to $p = 160^2$ pixels with varying values of ε, and report timings for images with up to $p = 512^2$ pixels. Moreover, we present a numerical example involving rotations, radial convolutions, and deconvolutions.

1.4. **Organization.** The remainder of the paper is organized as follows. In §2 we describe the analytical apparatus underlying the method. In §3 we describe the computational method. In §4 we justify why the discretization of the analytical apparatus performed by the computational method achieves the specified precision. In §5 we present numerical results. In §6 we discuss implications of the method and potential extensions.

2. Analytical apparatus

2.1. **Notation.** The eigenfunctions of the Laplacian on the unit disk (that satisfy Dirichlet boundary conditions) defined in (1) can be extended to $\mathbb{R}^2$ as functions supported on the unit disk by

$$\psi_{nk}(r,\theta) = c_{nk} J_n(\lambda_{nk} r) e^{\imath n\theta} \chi_{[0,1)}(r), \tag{5}$$

for $(n,k) \in \mathbb{Z} \times \mathbb{Z}_{>0}$, where $\chi_{[0,1)}$ denotes an indicator function for $[0,1)$. For the sake of completeness, we note that the normalization constants c_{nk} which ensure that $\|\psi_{nk}\|_{L^2} = 1$ are defined by

$$c_{0k} = \frac{1}{\pi^{1/2} J_1(\lambda_{0k})}, \quad \text{and} \quad c_{nk} = \frac{2^{1/2}}{\pi^{1/2} J_{n+1}(\lambda_{nk})}, \quad \text{for } n \neq 0. \tag{6}$$

We use the convention that the Fourier transform $\widehat{f} : \mathbb{R}^2 \to \mathbb{C}$ of an integrable function $f : \mathbb{R}^2 \to \mathbb{C}$ is defined by

$$\widehat{f}(\xi) = \frac{1}{2\pi} \int_{\mathbb{R}^2} f(x) e^{-\imath x \cdot \xi} dx, \tag{7}$$

where $x \cdot \xi$ denotes the Euclidean inner product. We define the convolution of two functions $f, g : \mathbb{R}^2 \to \mathbb{C}$ by

$$(f * g)(x) = \int_{\mathbb{R}^2} f(x-y) g(y) dy.$$

Furthermore, we will make use of the identity

$$J_n(r) = \frac{1}{2\pi} \int_0^{2\pi} e^{\imath r \sin\theta} e^{-\imath n\theta} d\theta, \tag{8}$$

see for example [29, Eq. 9.19]. To verify this identity, recall that $J_n(r)$ is the solution to Bessel's equation

$$r^2 \partial_{rr} y + r \partial_r y + (r^2 - n^2) y = 0, \tag{9}$$

see [9, Eq. 10.2.1]. Observe that

$$(r^2 \partial_{rr} + r\partial r + r^2) e^{\imath r \sin\theta} = -\partial_{\theta\theta} e^{\imath r \sin\theta}.$$

Since taking the second derivative of a function multiplies its n-th Fourier coefficient by $-n^2$, it follows that

$$(r\partial_{rr} + r\partial_r + (r^2 - n^2)) \frac{1}{2\pi} \int_0^{2\pi} e^{\imath r \sin\theta} e^{-\imath n\theta} d\theta = 0, \tag{10}$$

which establishes that (8) is a solution to (9). It is straightforward to verify that (8) also satisfies the same scaling relations, see [9, Eq. 10.2.2], as $J_n(r)$ when $r \to 0$.

2.2. **Fourier transform of eigenfunctions.** The analytic foundation for the presented fast method is the following expression for the Fourier transform of the functions ψ_{nk} defined in (5).

Lemma 2.1. *The Fourier transform $\widehat{\psi}_{nk}$ can be expressed by*

$$\widehat{\psi}_{nk}(\xi) = (-\imath)^n e^{\imath n\phi} \int_0^1 c_{nk} J_n(\lambda_{nk} r) J_n(\rho r) r dr, \tag{11}$$

where (ρ, ϕ) are polar coordinates for ξ.

We prove this lemma for completeness.

Proof of Lemma 2.1. By the definition of the Fourier transform (7) we have

$$\widehat{\psi}_{nk}(\xi) = \frac{1}{2\pi} \int_{\mathbb{R}^2} \psi_{nk}(x) e^{-\imath x\cdot\xi} dx.$$

Changing to polar coordinates $\xi = (\rho\cos\phi, \rho\sin\phi)$ and $x = (r\cos\theta, r\sin\theta)$ gives

$$\widehat{\psi}_{nk}(\xi) = \frac{1}{2\pi} \int_0^{2\pi} \int_0^1 c_{nk} J_n(\lambda_{nk} r) e^{\imath n\theta} e^{-\imath r\rho\cos(\theta-\phi)} r dr d\theta,$$

where we used the fact that $x\cdot\xi = r\rho\cos(\theta-\phi)$. Changing variables $\theta \mapsto -\theta+\phi-\pi/2$ and taking the integral over θ gives

$$\widehat{\psi}_{nk}(\xi) = (-\imath)^n e^{\imath n\phi} \int_0^1 c_{nk} J_n(\lambda_{nk} r) J_n(\rho r) r dr,$$

as desired. □

2.3. **Coefficients from eigenfunction Fourier transform.** Next, we observe how the coefficients of a function in the eigenfunction basis can be computed by an application of Lemma 2.1. In the following, we will write the arguments of Fourier transforms of functions in polar coordinates (ρ, ϕ). We have the following result:

Lemma 2.2. *Suppose that $\mathcal{I} \subset \mathbb{Z} \times \mathbb{Z}_{>0}$ is a finite index set, and set*

$$f = \sum_{(n,k)\in\mathcal{I}} \alpha_{nk} \psi_{nk}, \tag{12}$$

where $\alpha_{nk} \in \mathbb{C}$ are coefficients. Define $\beta_n : [0,\infty) \to \mathbb{R}$ by

$$\beta_n(\rho) := \imath^n \int_0^{2\pi} \widehat{f}(\rho,\phi) e^{-\imath n\phi} d\phi. \tag{13}$$

It then holds that

$$\alpha_{nk} = c_{nk} \beta_n(\lambda_{nk}). \tag{14}$$

The proof is a direct consequence of Lemma 2.1.

Proof of Lemma 2.2. Observe that (11) implies

$$\imath^n \int_0^{2\pi} \widehat{\psi}_{n'k'}(\rho,\phi) e^{-\imath n\phi} d\phi = 2\pi\delta(n-n') \int_0^1 c_{n'k'} J_n(\lambda_{n'k'} r) J_n(\rho r) r dr, \tag{15}$$

where $\delta(x) = 1$ if $x = 0$ and $\delta(x) = 0$ otherwise. Evaluating (15) at radius $\rho = \lambda_{nk}$ gives

$$\begin{aligned}\imath^n \int_0^{2\pi} \widehat{\psi}_{n'k'}(\lambda_{nk}, \phi) e^{-\imath n\phi} d\phi &= 2\pi\delta(n-n') \int_0^1 c_{n'k'} J_n(\lambda_{n'k'} r) J_n(\lambda_{nk} r) r dr \\ = 2\pi\delta(n-n') \int_0^1 c_{nk'} J_n(\lambda_{nk'} r) J_n(\lambda_{nk} r) r dr &= \frac{1}{c_{nk}} \delta(n-n')\delta(k-k'),\end{aligned}$$

where the final equality follows from the orthogonality of the eigenfunctions $\psi_{nk'}$, (which is a consequence of the fact that the Laplacian is self-adjoint). By the definition of β_n in (13), this implies that

$$\begin{aligned}\beta_n(\lambda_{nk}) &= \sum_{(n',k')\in\mathcal{I}} \alpha_{n'k'} \imath^n \int_0^{2\pi} \widehat{\psi}_{n'k'}(\rho, \phi) e^{-\imath n\phi} d\phi \\ &= \sum_{(n',k')\in\mathcal{I}} \frac{\alpha_{n'k'}}{c_{n'k'}} \delta(n-n')\delta(k-k') = \frac{\alpha_{nk}}{c_{nk}},\end{aligned}$$

which concludes the proof. □

Remark 2.1 (Special property of Bessel functions). We emphasize that the integral expression (8) of the Bessel function is crucial for the fast method of this paper. The possibility of extending the approach to create other fast transforms defined on domains in $\mathbb{R}^2$ therefore hinges on identifying equally useful integral expressions for the corresponding transforms.

2.4. **Convolution with radial functions.** Let $g(x) = g(|x|)$ be a radial function. In this section, we observe how the convolution with g can be computed via a diagonal transform of the coefficients. More precisely, we compute the projection of the convolution with g onto the span of any finite basis of the eigenfunctions ψ_{nk}.

Lemma 2.3. *Let f be a function with coefficients α_{nk} as in* (12)*, and $g(x) = g(|x|)$ be a radial function. We have*

$$P_{\mathcal{I}}(f * g) = \sum_{(n,k)\in\mathcal{I}} \alpha_{nk} \widehat{g}(\lambda_{nk}) \psi_{nk},$$

where $P_{\mathcal{I}}$ denotes the orthogonal projection onto the span of $\{\psi_{nk}\}_{(n,k)\in\mathcal{I}}$.

The proof is a direct application of Lemma 2.2.

Proof of Lemma 2.3. We use the notation $g(x) = g(|x|)$ and $\widehat{g}(\xi) = \widehat{g}(|\xi|)$. Since the functions ψ_{nk} are an orthonormal basis, in order to compute the orthogonal projection $P_{\mathcal{I}}$, it suffices to determine the coefficients of $f * g$ with respect to ψ_{nk} for $(n,k) \in \mathcal{I}$. Since $\widehat{(f*g)}(\rho,\phi) = \widehat{f}(\rho,\phi)\widehat{g}(\rho)$, and $\widehat{g}$ is radial, we have

$$\begin{aligned}\imath^n \int_0^{2\pi} \widehat{(f*g)}(\lambda_{nk}, \phi) e^{-\imath n\phi} d\phi &= \imath^n \int_0^{2\pi} \widehat{f}(\lambda_{nk}, \phi) \widehat{g}(\lambda_{nk}) e^{-\imath n\phi} d\phi \\ &= \frac{\alpha_{nk}}{c_{nk}} \widehat{g}(\lambda_{nk}),\end{aligned}$$

where the final equality follows from (14). An application of Lemma 2.2 then completes the proof. □

3. Computational method

3.1. **Notation.** Recall that p is the number of pixels in the given image, and m is the number of functions in our expansion. Throughout, we write all computational complexities in terms of p with the understanding that $m \leq p$. This assumption is not restrictive since it does not make sense to expand an image using more functions than the number of pixels.

Remark 3.1 (Precision of calculations). The computational complexities also have a (logarithmic) dependence on the reciprocal of the desired precision ε; for simplicity, we treat ε as a fixed constant, say, $\varepsilon = 10^{-7}$, and do not include it in complexity statements.

Remark 3.2 (Continuous to discrete). Throughout this section, for clarity of exposition, we use the same notation used in the continuous setting described in §2 to represent the corresponding discrete objects. We justify why the discrete sums of the algorithm will approximate the continuous integrals to the desired precision in §4.

3.2. **Overview.** In the following, we describe how to apply the operators B and B^* defined above in §1.3 in $\mathcal{O}(p \log p)$ operations. For the purpose of exposition, we start by describing a simplified method before presenting the full method. The section is organized as follows:

- In §3.3 we describe a simplified method to apply B and B^* in $\mathcal{O}(p^{3/2} \log p)$ operations. The simplified method is a direct application of the lemmas from the previous section.
- In §3.4 we provide an informal description of how to modify the simplified method to create a fast method to apply B and B^* in $\mathcal{O}(p \log p)$ operations. The main additional ingredient is fast interpolation from Chebyshev nodes.
- In §3.5 we give a detailed description of the fast method to apply B and B^* in $\mathcal{O}(p \log p)$ operations.

3.3. **Simplified method.** In this section, for the purpose of exposition, we present a simplified method that applies B and B^* in $\mathcal{O}(p^{3/2} \log p)$ operations. Recall that $f_1, \ldots, f_p \in \mathbb{R}$ are the pixel values of the image, $x_1, \ldots, x_p \in [-1, 1]^2$ are the pixel coordinates, and $\psi_1, \ldots, \psi_m$ are the eigenfunctions below the given bandlimit, see §1.2. Let $\lambda_1, \ldots, \lambda_m$ denote the Bessel function roots (square root of the eigenvalues) corresponding to $\psi_1, \ldots, \psi_m$. Further, suppose that (n_i, k_i) are the indices such that $\psi_i = \psi_{n_i k_i}$ and $\lambda_i = \lambda_{n_i k_i}$.

Fix a positive integer $s = \lfloor \max\{7.08\sqrt{p}, \log_2 \varepsilon^{-1} + \log_2 p\} \rfloor$ (see §4.3 for a justification of this parameter choice). We first describe how to apply $B^* : \mathbb{R}^p \to \mathbb{R}^m$ which is defined by

$$(B^* f)_i = \sum_{j=1}^{p} f_j \overline{\psi_i(x_j)} h,$$

where h is a normalization parameter. The simplified method has three steps, which make use of the non-uniform fast Fourier transform (NUFFT) [11, 14, 22].

Step 1. Using the type-2 2-D NUFFT, in $\mathcal{O}(p^{3/2} \log p)$ operations, compute:

$$\hat{f}_{i\ell} := \sum_{j=1}^{p} f_j e^{-\imath x_j \cdot \xi_{i\ell}} \quad \text{where} \quad \xi_{i\ell} := \lambda_i(\cos 2\pi\ell/s, \sin 2\pi\ell/s),$$

for $(i, \ell) \in \{1, \dots, m\} \times \{0, \dots, s-1\}$ to precision ε.

Step 2. Using the FFT, in $\mathcal{O}(p^{3/2} \log p)$ operations, compute:

$$\beta_n(\lambda_i) := \sum_{\ell=1}^{s} \hat{f}_{i\ell} e^{-\imath n 2\pi \ell / s},$$

for $(i, n) \in \{1, \dots, m\} \times \{0, \dots, s-1\}$.

Step 3. By Lemma 2.2 it follows that

$$(B^* f)_i = \beta_{n_i}(\lambda_i) c_i h,$$

where n_i is the integer such that $\psi_i = \psi_{n_i k_i}$ and $c_i = c_{n_i k_i}$.

In the following remarks, we discuss how reversing these steps provides an immediate way to apply B and discuss limitations of the simplified method.

Remark 3.3 (Applying B). Note that each step of the algorithm consists of applying a linear transform whose adjoint can be applied in a similar number of operations.

Indeed, the adjoint of Step 1 (which uses type-2 NUFFT) is a type-1 2-D NUFFT [1], and the adjoint of Step 2 (which uses a standard FFT) is an inverse FFT. Step 3 (interpolation) can be computed in a variety of ways (including by using the NUFFT or FFT and sparse matrices, see Remark 3.5) which each have similarly fast adjoints [11, 14, 22]. Therefore, in order to apply $B : \mathbb{R}^m \to \mathbb{R}^p$, which is defined by

$$(Ba)_j = \sum_{i=1}^{m} a_i \psi_i(x_j) h,$$

we can just apply the adjoint of each step in the reverse order.

Remark 3.4 (Limitations of simplified method). The problem with the simplified method is that the first step is prohibitively expensive since we want a method that works in $\mathcal{O}(p \log p)$ operations. The key to overcome this limitation is to observe that $\beta_{n_i}(\lambda_i)$ are samples from analytic functions, and to add the additional ingredient of fast interpolation from Chebyshev nodes.

3.4. **Summary of fast method.** In this section, we describe how the method of the previous section can be improved from $\mathcal{O}(p^{3/2} \log p)$ to $\mathcal{O}(p \log p)$ by using fast interpolation from Chebyshev nodes. Recall that Lemma 2.2 defines

$$\beta_n(\rho) := \imath^n \int_0^{2\pi} \widehat{f}(\rho, \phi) e^{-\imath n \phi} d\phi,$$

and states that the coefficients can be computed by evaluating the appropriate β_n at λ_i. The problem with the simplified method is the first step: computing $\widehat{f}(\rho, \theta)$ for $\rho = \lambda_1, \dots, \lambda_m$ and a number of points θ so that we are sampling at the Nyquist rate is a set of $\mathcal{O}(p^{3/2})$ points, which is already prohibitive. Fortunately, there is a simple solution to this issue: for each n, the functions $\beta_n(\rho)$ are analytic functions of ρ which can be interpolated to determine the coefficients, if they are tabulated at appropriate points; see §4.2 for a discussion of why $\mathcal{O}(\sqrt{p})$ points are sufficient. Here is an informal summary of the fast method:

- Compute the Fourier transform of f at

  $$\xi_{i\ell} := t_i(\cos 2\pi\phi_\ell, \sin 2\pi\phi_\ell),$$

  for $\mathcal{O}(\sqrt{p})$ Chebyshev nodes t_i and $\mathcal{O}(\sqrt{p})$ angles ϕ_ℓ.

- Compute $\beta_n(t_i)$ for the $\mathcal{O}(\sqrt{p})$ Chebyshev nodes t_i and $\mathcal{O}(\sqrt{p})$ frequencies n.
- For each of the $\mathcal{O}(\sqrt{p})$ frequencies, use fast interpolation from the $\mathcal{O}(\sqrt{p})$ Chebyshev nodes t_i to the $\mathcal{O}(\sqrt{p})$ Bessel function roots associated with each frequency n. We illustrate the interpolation step in Fig. 2.

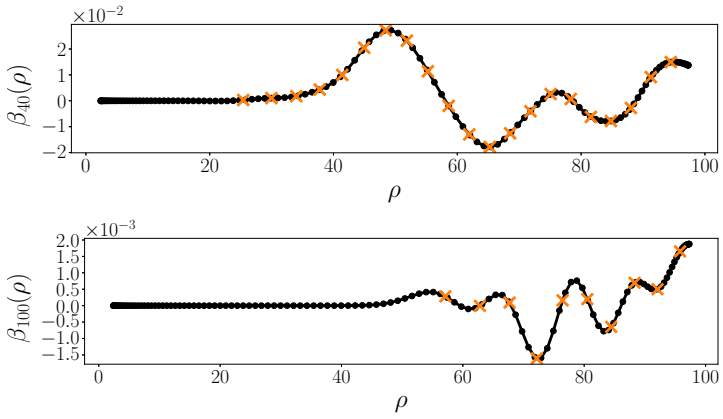

FIGURE 2. We visualize the interpolation step for a 64×64 input image. For $n = 40, 100$ we plot $\beta_n(\rho)$ (black line), interpolation source nodes (black dots), and target points (orange crosses).

3.5. **Detailed description of fast method.** Recall that $\psi_1, \ldots, \psi_m$ are the eigenfunctions below our bandlimit, and $\lambda_1, \ldots, \lambda_m$ are the associated Bessel function roots (which are the square root of the associated eigenvalues). Assume that $\lambda_1, \ldots, \lambda_m$ are listed in ascending order so that they are contained in the interval $[\lambda_1, \lambda_m]$. Fix $q = \lceil \max\left\{2.4\sqrt{p}, \log_2 \varepsilon^{-1}\right\} \rceil$, where ε is the desired relative error, see §4.2 for a justification of this parameter choice. Recall that Chebyshev nodes of the first kind for the interval $[\lambda_1, \lambda_m]$ are defined by

$$t_i = (\lambda_m - \lambda_1)\frac{1 - \cos(\pi(i-1)/q)}{2} + \lambda_1,$$

for $i = 1, \ldots, q$. The fast method follows the same strategy as the simplified method. The key difference is that the fast method replaces the final step with fast interpolation, and therefore evaluates the Fourier coefficients on a smaller number of nodes (the Chebyshev nodes).

Step 1. Using the NUFFT, in $\mathcal{O}(p \log p)$ operations, compute:

$$\hat{f}_{i\ell} := \sum_{j=1}^{p} f_j e^{-\imath x_j \cdot \xi_{i\ell}} \quad \text{where} \quad \xi_{i\ell} := t_i(\cos 2\pi\ell/s, \sin 2\pi\ell/s),$$

for $(i, \ell) \in \{1, \ldots, q\} \times \{0, \ldots, s-1\}$.

Step 2. Using the FFT, in $\mathcal{O}(p \log p)$ operations, compute:

$$\beta_n(t_i) := \sum_{\ell=1}^{s} \hat{f}_{i\ell} e^{-\imath n 2\pi\ell/s},$$

for $(i,n) \in \{1,\dots,m\} \times \{0,\dots,s-1\}$.

Step 3. Using fast interpolation from Chebyshev nodes, in $\mathcal{O}(p \log p)$ operations, compute:

$$(B^* f)_i = \beta_{n_i}(\lambda_i),$$

for $i \in \{1,\dots,m\}$ where n_i is the integer such that $\psi_i = \psi_{n_i k_i}$.

As detailed in Remark 3.3, the fact that each step is a linear operator whose adjoint can be applied in a similar number of operations provides a similar fast method to apply B.

Remark 3.5 (Methods for fast interpolation from Chebyshev nodes). Theoretically, the most straightforward way to perform fast interpolation from Chebyshev nodes is to consider the function values at the Chebyshev nodes as function values associated with equally spaced points on a half circle (by a change of variables). Then the fast interpolation (called spectral interpolation) can be computed by using the NUFFT in $\mathcal{O}(p \log p)$ operations or from arbitrary nodes using the Fast Multipole Method, see [10]. Practically speaking, since the NUFFT run-time constant is high, choosing a fixed number of source points centered around the target points (say 20 source points) and then applying a precomputed sparse (barycentric interpolation [3]) matrix may be more practical; sparse interpolation can be used in combination with standard spectral interpolation (using the discrete cosine transform) to double or quadruple the number of Chebyshev nodes before the sparse interpolation step.

4. Continuous to discrete

We conclude the description of the fast algorithm by discussing additional analytic details that justify transitioning from the continuous identities in §2 to the discrete implementations of §3.

4.1. **Maximum bandlimit.**

In this section, we derive a bound for the bandlimit parameter, which controls the number of basis functions used in the expansion. The number of basis functions should not exceed the number of pixels of the image corresponding to points within the unit disk. In combination with Weyl's law, this leads to a bound on the maximum bandlimit.

In more detail, from Weyl's law, the number of Dirichlet eigenvalues of the Laplacian in the disk is bounded by λ is

$$\#\Big\{(n,k) \in \mathbb{Z} \times \mathbb{Z}_{>0} : \lambda_{nk} \leq \lambda\Big\} = \frac{\lambda^2}{4} - \frac{\lambda}{2} + \mathcal{O}(\lambda^{2/3}), \tag{16}$$

see [8]. Similarly, the inscribed circle in an $L \times L$ image has radius $\left\lfloor \frac{L-1}{2} \right\rfloor$. The number of pixels contained inside this circle is therefore

$$\#\left\{(i,j) \in \mathbb{Z} \times \mathbb{Z} : i^2 + j^2 \leq \left\lfloor \frac{L-1}{2} \right\rfloor^2 \right\} = \pi \left\lfloor \frac{L-1}{2} \right\rfloor^2 + \mathcal{O}(L^{2/3}), \tag{17}$$

see for example [13, 16, 30]. Equating (16) with (17) results in

$$\lambda = 2\sqrt{\pi} \left\lfloor \frac{L-1}{2} \right\rfloor + 1 + \mathcal{O}\left(L^{-1/3}\right). \tag{18}$$

Practically speaking, it can be advantageous to expand the image in fewer basis functions than described by (18), but, in any case, $\lambda = \mathcal{O}(L)$ is a natural choice and different motivations exist in the literature [34].

Remark 4.1 (FFT Bandlimit heuristic). One heuristic for setting the bandlimit is based on the fast Fourier transform (FFT). For a centered FFT on a signal of length L, the maximum frequency is $\pi^2(L/2)^2$ which corresponds to a bandlimit of $\lambda = \pi L/2$. Note that

$$\pi L/2 \approx 1.57L < 1.77L \approx \sqrt{\pi}\lfloor (L-1)/2 \rfloor /2, \tag{19}$$

so this FFT bandlimit heuristic does indeed produce a reasonable bandlimit below the bound (18) derived from Weyl's law. We use this bandlimit for our numerical experiments. The computational complexity and accuracy guarantees of the method presented in this paper hold for any bandlimit $\lambda = \mathcal{O}(L)$. However, the fact that the fast method performs interpolation in Fourier space inside a disk bounded by the maximum bandlimit provides additional motivation for this FFT-based heuristic since it will ensure that the disk will be contained within the square in frequency space used by the two-dimensional FFT.

4.2. **Number of radial nodes.** The following lemma shows that $\mathcal{O}(\sqrt{p})$ Chebyshev nodes are sufficient for accurate interpolation.

Lemma 4.1. *When running the algorithm on an image with p pixels at the maximum bandlimit, choosing the number of Chebyshev nodes q to satisfy*

$$q = \lfloor \max\left\{2.4\sqrt{p}, \log_2 \varepsilon^{-1}\right\} \rfloor, \tag{20}$$

achieves relative error less than ε in the interpolation step of §3.5.

It will be clear from the proof that the constant 2.4 in the statement of the lemma is an overestimate, see Remark 4.2 for a discussion of how this constant can be improved.

Proof of Lemma 4.1. When interpolating a smooth differentiable function h defined on the interval $[a, b]$ using an interpolating polynomial p_q at q Chebyshev nodes, the residual term $R(x) = h(x) - p_q(x)$ can be written as

$$|R(x)| \leq \frac{2C_q}{q!}\left(\frac{b-a}{4}\right)^q,$$

where $C_q := \max_{x\in[a,b]} |h^{(q)}(x)|$; indeed, this can be deduced from [9, § 3.3]. If we apply this result with $[a, b] = [\lambda_1, \lambda_m]$, the residual satisfies

$$|R(x)| \leq \frac{2C_q}{q!}\left(\frac{\lambda_m - \lambda_1}{4}\right)^q \leq \frac{2C_q}{q!}\left(\frac{\sqrt{\pi p}}{4}\right)^q$$

where the final inequality follows from the bound $\lambda_m \leq \sqrt{\pi p}$; see § 4.1. In order to apply this bound to $h(\rho) = \beta_n(\rho)$, we estimate

$$C_q := \max_{\rho\in[\lambda_1,\lambda_m]} |\beta_n^{(q)}(\rho)|,$$

where $\beta_n^{(q)}$ denotes the qth derivative of β_n defined in (13). Since f is compactly supported, its Fourier transform $\hat{f}$ is analytic and the C_q are therefore all finite. More precisely, the term C_q can be bounded uniformly by using

$$|\beta_n^{(q)}(\rho)| = \left|\frac{\mathrm{d}^q}{\mathrm{d}\rho^q}\left(\frac{1}{2\pi}\int_0^{2\pi} \widehat{f}(\rho,\phi)e^{-in\phi}\mathrm{d}\phi\right)\right| \leq \left(\frac{1}{2\pi}\int_0^{2\pi}\left|\frac{\mathrm{d}^q}{\mathrm{d}\rho^q}\widehat{f}(\rho,\phi)\right|\mathrm{d}\phi\right).$$

We write out

$$\begin{aligned}\left|\frac{\mathrm{d}^q}{\mathrm{d}\rho^q}\widehat{f}(\rho,\phi)\right| &= \left|\frac{1}{2\pi}\frac{\mathrm{d}^q}{\mathrm{d}\rho^q}\int_0^{2\pi}\int_0^{\infty} f(r,\theta)e^{-\imath r\rho\cos(\theta-\phi)}r\mathrm{d}r\mathrm{d}\theta\right| \\ &= \left|\frac{1}{2\pi}\int_0^{2\pi}\int_0^{\infty} (-\imath r\cos(\theta-\phi))^q\, f(r,\theta)e^{-\imath r\rho\cos(\theta-\phi)}r\mathrm{d}r\mathrm{d}\theta\right| \\ &= \left|\frac{1}{2\pi}\int_0^{2\pi}\int_0^{1} (-\imath r\cos(\theta-\phi))^q\, f(r,\theta)e^{-\imath r\rho\cos(\theta-\phi)}r\mathrm{d}r\mathrm{d}\theta\right| \\ &\leq \frac{1}{2\pi}\int_0^{2\pi}\int_0^{1} |f(r,\theta)|\, r\mathrm{d}r\mathrm{d}\theta = \frac{1}{2\pi}\|f\|_{L^1} \leq \frac{1}{2\sqrt{\pi}}\|f\|_{L^2},\end{aligned}$$

where the third equality uses the fact that f is supported on the disk of radius 1 and the last inequality uses Cauchy-Schwarz applied to f multiplied by the indicator function of the unit disk. It follows that

$$|R(x)| \leq \frac{\|f\|_{L^2}}{\sqrt{\pi}q!}\left(\frac{\sqrt{\pi p}}{4}\right)^q.$$

Therefore, in order to achieve relative error $|R(x)|/\|f\|_{L^2} = \varepsilon$, it suffices to set q such that

$$\varepsilon \leq \frac{1}{\sqrt{\pi}q!}\left(\frac{\sqrt{\pi p}}{4}\right)^q. \tag{21}$$

It follows from Stirling's approximation [9, 5.11.3] that

$$\frac{|R(x)|}{\|f\|_{L^2}} \leq \left(\frac{\sqrt{\pi p}e}{4q}\right)^q \tag{22}$$

Setting $\sqrt{\pi p}e/4q = 1/2$ and solving for q gives

$$q = \frac{\sqrt{\pi}e\sqrt{p}}{2} \approx 2.4\sqrt{p} \quad \Longrightarrow \quad q \geq \max\{2.4\sqrt{p}, \log_2 \varepsilon^{-1}),$$

is sufficient to achieve relative error less than ε. □

4.3. **Number of angular nodes.** The following lemma shows that $s = \mathcal{O}(\sqrt{p})$ angular nodes is sufficient to achieve relative error ε.

Lemma 4.2. *When running the algorithm on an image with p pixels at the maximum bandwidth, choosing the number of equispaced angular nodes s to satisfy*

$$s = \lfloor \max\{7.08\sqrt{p}, \log_2 \varepsilon^{-1} + \log_2 p\}\rfloor. \tag{23}$$

achieves relative error less than ε in the FFT step of §3.5.

As above, we emphasize that the constant 7.08 in the statement of this result is an overestimate. See 4.2 for discussion about how this constant can be improved.

Proof of Lemma 4.2. We will require a classical result, see for example [18]: suppose that $f : [0, 2\pi] \to \mathbb{R}$ is a smooth periodic function on the torus $[0, 2\pi]$ where 0 and 2π are identified. Then

$$\left|\int_0^{2\pi} f(\theta)d\theta - \frac{1}{n}\sum_{j=0}^{n-1} f(\theta_j)\right| < 2\frac{\|f^{(n)}\|_2}{n^n},$$

for all $n \geq 2$, where $\theta_j = 2\pi j/n$.

Indeed, we start by writing $f(\theta) = \sum_{k=-\infty}^{\infty} a_k e^{\imath k\theta}$ and use the fact that $\frac{1}{n}\sum_{j=0}^{n-1} e^{\imath k\theta_j}$ is equal to 1 if k is a multiple of n and is 0 otherwise to obtain:

$$\left|\int_0^{2\pi} f(\theta)d\theta - \frac{1}{n}\sum_{j=0}^{n-1} f(\theta_j)\right| = \left|a_0 - \frac{1}{n}\sum_{j=0}^{n-1}\sum_{k=-\infty}^{\infty} a_k e^{\imath k\theta_j}\right| = \left|\sum_{t\neq 0} a_{nt}\right|.$$

Multiplying and dividing terms by $(\imath nt)^n$, and using Cauchy-Schwarz:

$$\begin{aligned}\left|\sum_{t\neq 0} a_{nt}\frac{(\imath nt)^n}{(\imath nt)^n}\right| &\leq \left(\sum_{t\neq 0}|(\imath nt)^n a_{nt}|^2\right)^{1/2}\left(\sum_{t>0}\frac{2}{(nt)^{2n}}\right)^{1/2} \\ &\leq \left(\sum_{\ell=-\infty}^{\infty}|(\imath\ell)^n a_\ell|^2\right)^{1/2}\left(\sum_{t>0}\frac{2}{(nt)^{2n}}\right)^{1/2} < 2\frac{\|f^{(n)}\|_{L^2}}{n^n},\end{aligned}$$

where the last inequality follows from Parseval's identity and the identity $\sum_{t>0}\frac{1}{t^4} = \frac{\pi^4}{90}$, together with $\sqrt{\pi^4/45} < 2$.

In order to determine the required number of angular nodes we write the image f and its Fourier transform $\widehat{f}$ as

$$f(x) = \sum_{j=1}^{p} f_j \delta_{x_j}(x), \qquad \widehat{f}(\xi) = \sum_{j=1}^{p} f_j \frac{1}{2\pi} e^{i\xi\cdot x_j}.$$

By the triangle inequality, it suffices to estimate integrals of functions of the form

$$g(\phi) = e^{ir\rho\cos(\theta-\phi)-in\phi}$$

to precision $\frac{\varepsilon}{p}$ for all $r \in [0,1)$, $\theta \in [0,2\pi)$, $\rho \in [\lambda_1, \lambda_m]$, and $|n| \leq n_m$. We have the estimate

$$|g^{(s)}| \leq (\lambda_m + n_m)^s.$$

Thus, in order to integrate these functions, we need to choose s such that

$$\left(\frac{\lambda_m + n_m}{s}\right)^s \leq \frac{\varepsilon}{p}. \tag{24}$$

It follows that choosing

$$s = \max\{2(\lambda_m + n_m), \log_2 \varepsilon^{-1} + \log_2 p\},$$

achieves relative error ε. To complete the proof we note that $\lambda_m \leq \lambda$, where λ is the maximum bandlimit from § 4.1. Also by [9, 10.21.40] we have

$$\lambda_{n1} = n + 1.8575n^{1/3} + \mathcal{O}(n^{-1/3}),$$

which implies that the maximum angular frequency $n_m \leq \lambda$. We conclude that

$$s = \max\{4\lambda, \log\varepsilon^{-1} + \log_2 p\},$$

is sufficient to achieve error ε; using the bound $\lambda \leq 1.77\sqrt{p}$ from (19) completes the proof. □

Remark 4.2 (Improving estimates for number of radial and angular nodes). While Lemmas 4.1 and 4.2 show that the number of radial nodes q and angular nodes s are $\mathcal{O}(\sqrt{p})$, the constants in the lemmas are not optimal. For practical purposes, choosing the minimal number of nodes possible to achieve the desired error is advantageous to improve the run time constant of the algorithm, and it is clear from

the proofs how the estimates can be refined. For Lemma 4.1 we set $Q = \lceil 2.4\sqrt{p} \rceil$, and motivated by (21) compute

$$\varepsilon_{\text{radial}}(q) = \frac{1}{\sqrt{\pi} q!} \left(\frac{\sqrt{\pi p}}{4} \right)^q,$$

for $q = 1, \ldots, Q$ and choose the smallest value q^* of q such that $\varepsilon_{\text{radial}}(q^*) \leq \varepsilon$. Similarly, for Lemma 4.2, we set $S = \lceil 7.08\sqrt{p} \rceil$, and motivated by (24) compute

$$\varepsilon_{\text{angular}}(s) = p \left(\frac{\lambda_m + n_m}{s} \right)^s,$$

for $s = 1, \ldots, S$ and choose the smallest value s^* of s such that $\varepsilon_{\text{angular}}(s) \leq \varepsilon$. Then, it follows that $2.4\sqrt{p}$ and $7.08\sqrt{p}$ can be replaced by q^* and s^*, in the statements of Lemmas 4.1 and 4.2, respectively. This procedure improves the estimate of the required number of angular and radial nodes by a constant factor.

5. Numerical results

5.1. **Accuracy results.** In this section, we report numerical results for the accuracy of the fast algorithm compared to matrix multiplication. Recall that $B : \mathbb{R}^m \to \mathbb{R}^p$ maps coefficients to images by

$$(Ba)_j = \sum_{i=1}^{m} a_i \psi_i(x_j) h,$$

and its adjoint transform $B^* : \mathbb{R}^p \to \mathbb{R}^m$ maps images to coefficients by

$$(B^* f)_i = \sum_{j=1}^{p} f_j \overline{\psi_i(x_j)} h,$$

see § 1.3. By defining the $m \times p$ matrix B by

$$B(i,j) = \psi_i(x_j) h,$$

we can apply B and B^* by dense matrix multiplication to test the accuracy of our fast method. Since the size of the matrix scales like L^4 for $L \times L$ images, constructing these matrices quickly becomes prohibitive so the comparison is only given up to $L = 160$, see Table 1, where

$$\text{err}_a = \frac{\|a_{\text{fast}} - a_{\text{dense}}\|_2}{\|a_{\text{dense}}\|_2} \quad \text{and} \quad \text{err}_f = \frac{\|f_{\text{fast}} - f_{\text{dense}}\|_2}{\|f_{\text{dense}}\|_2},$$

denote the relative errors of the coefficients and the image, respectively, where $a_{\text{dense}} = B^* f$ and $f_{\text{dense}} = Ba$ are computed by dense matrix multiplication and a_{fast} and f_{fast} are the corresponding quantities computed using the fast algorithm of this paper.

The image used for the accuracy comparison is a tomographic projection of a 3-D density map representing a bio-molecule (E. coli 70S ribosome) [26], retrieved from the online EM data bank [21].

TABLE 1. Relative error of fast method compared to dense matrix multiplication.

l	ϵ	err_a	err_f
64	1.00e-04	1.92422e-05	2.10862e-05
96	1.00e-04	1.82062e-05	2.52219e-05
128	1.00e-04	1.90648e-05	2.41142e-05
160	1.00e-04	2.00748e-05	2.49488e-05
64	1.00e-07	2.03272e-08	2.98083e-08
96	1.00e-07	2.28480e-08	2.58272e-08
128	1.00e-07	2.69215e-08	2.27676e-08
160	1.00e-07	2.47053e-08	2.51146e-08
64	1.00e-10	3.55320e-11	2.36873e-11
96	1.00e-10	2.99849e-11	2.48166e-11
128	1.00e-10	3.25650e-11	2.61890e-11
160	1.00e-10	3.13903e-11	3.50455e-11
64	1.00e-14	7.41374e-15	6.82660e-15
96	1.00e-14	9.82890e-15	8.80843e-15
128	1.00e-14	1.21146e-14	1.11909e-14
160	1.00e-14	1.36735e-14	1.51430e-14

5.2. **Timing results.** In this section, we plot the timing of the method for $L \times L$ images with $p = L^2$ pixels. We demonstrate that the method does indeed have complexity $\mathcal{O}(p \log p)$ and that the timings are practical. We plot the time of pre-computation and the time of applying B using the fast method; for comparison, we include timings for forming and applying the dense matrix B, see Fig. 3. The timing for applying B^* are similar to the timings for applying B (since the algorithm consists of applying similar transforms in the reverse order) so a separate plot was not included.

The timings were carried out on a computer with an AMD 5600X processor and 24 GB of memory. We set $\varepsilon = 10^{-7}$ for the reported timings, and compare to the dense method up to $L = 160$. For $L > 160$ comparison to the dense method was prohibitively expensive. For reference, storing the dense transform matrix in double precision complex numbers for $L = 512$ would require about 640 GB of memory. The NUFFT uses the FINUFFT implementation [1, 2]. The image used for the timing results is a tomographic projection of a 3-D density map representing the SARS-CoV-2 Omicron spike glycoprotein complex [15], retrieved from the online EM data bank [21].

Remark 5.1 (Pre-computation time negligible when transforming many images). The pre-computation involves organizing Bessel function roots, and creating data structures for the NUFFT and interpolation steps of the algorithm. The pre-computation only needs to be preformed once for a given size of image L, and becomes negligible when the method is used to expand a large enough set of images (around 100 images), which is a typical use-case in, for example, applications in cryo-EM [4].

Remark 5.2 (Breakdown of timing of fast algorithm). Each step of the algorithm has roughly the same magnitude. For example, for $L = 512$ and $\varepsilon = 10^{-7}$ the timings of the NUFFT, FFT, and Interpolation steps of the algorithm for applying B are 0.035, 0.046, and 0.026 seconds, respectively. We note that the timing of each

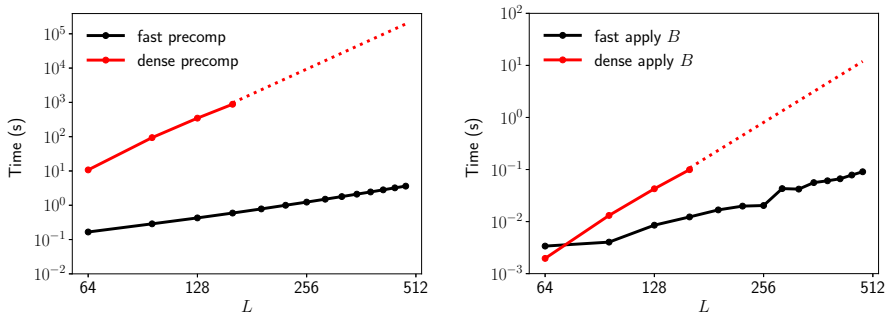

FIGURE 3. Timings of fast method versus dense method for precomputation (left) and applying B (right). The timings for the dense method for $L > 160$ are extrapolated since the memory requirements for the dense method were prohibitive.

step is dependent on the choice of parameters. For example, sampling more points will increase the cost of the NUFFT step, but decrease the cost of the interpolation step, since sparser interpolation matrices can be used; decreasing ε will increase the cost of the NUFFT step.

Remark 5.3 (Parallelization). The timings reported in Fig. 3 are for a single-threaded CPU code. However, each step of the code is amenable to parallelization through GPU implementations. Indeed, the NUFFT step has a GPU implementation [27], and the 2-D FFT and interpolation steps can also benefit from straightforward parallelization schemes.

5.3. **Numerical example: convolution and rotation.** We lastly present an example illustrating the use of the steerable and fast radial convolution properties of the eigenbasis. The example is motivated by cryo-EM, wherein tomographic projection images of biological molecules in a sample are registered by electron beams; see for example, [12] for more information. Because of aberrations within the electron-microscope and random in-plane rotations of the molecular samples, the registered image I_r does not precisely coincide with the true projection image I_p and the following model is used:

$$I_r(x) = c(|x|) * R_\theta \left(I_p(x)\right) + \varepsilon, \tag{25}$$

where R_θ describes rotation around the origin by an angle of θ, c is a radial function termed the point-spread function and ε is additive white noise. The function $\widehat{c}$ is in turn known as the contrast transfer function (CTF). Examples of point spread functions are shown in Fig. 4.

Notably, the regions of the frequency space where $\widehat{c}$ equals zero destroy information of $I_p(x)$. However, the fact that convolution is a diagonal transformation of the coefficients in the basis of eigenfunctions enables reconstruction of a fixed projection image I_p from a small number of registered images $I_r^{(i)}$ with different point spread functions $c_i(|x|)$, rotations R_{θ_i}, and noise $\varepsilon^{(i)}$, for $i = 1, \ldots, t$. From Lemma 2.2, it follows that the basis coefficients $\alpha_{nk}^{(i)}$ of the registered images satisfy

$$\alpha_{nk}^{(i)} = \widehat{c}_i(\lambda_{nk}) e^{\imath n \theta_i} \alpha_{nk}^{(0)} + \varepsilon_{nk}^{(i)}, \qquad \text{for } i = 1, \ldots, t, \tag{26}$$

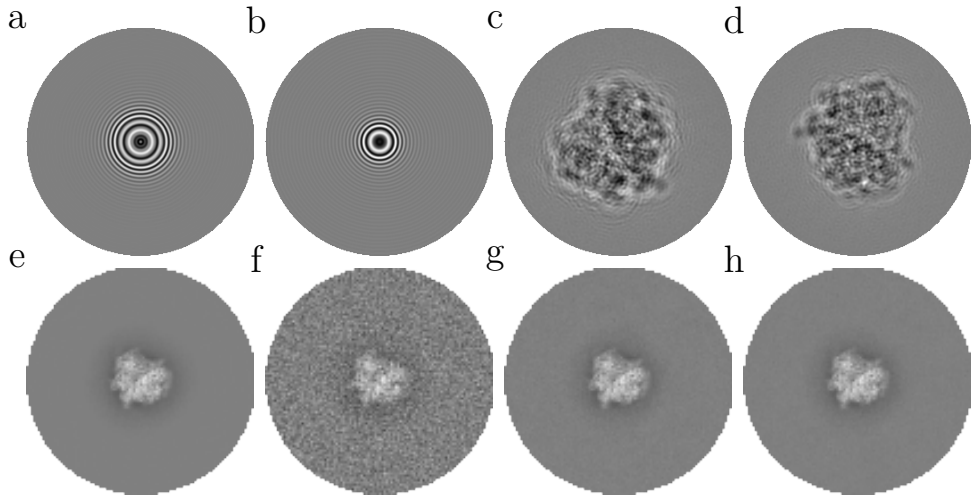

FIGURE 4. Two different point spread functions (a–b), result of their convolution with a fixed image and subsequent rotation (c–d), (e) Projection of reference image into the eigenbasis using the fast algorithm. (f-g) result of deconvolution algorithm using $t = 1, 3, 5$, respectively.

where $\alpha_{nk}^{(0)}$ denote the basis coefficients of I_p. We assume that the parameters θ_i and c_i are known or estimated to a desired precision. We remark that the standard FFT can be used to solve this problem when there are no rotations.

To recover the $\alpha_{nk}^{(0)}$, we find the least-squares optimizers of (26). To improve the conditioning of the problem, (26) is thresholded to exclude the values of i for which $\widehat{c}_i(\lambda_{nk})$ has sufficiently low magnitude. We therefore estimate $\alpha_{nk}^{(0)}$ by $\alpha_{nk}^{(0)} \approx \alpha_{nk}$, with α_{nk} defined by

$$\alpha_{nk} = \min_{\alpha_{nk}} \sum_{(n,k)\in\mathcal{I}} \sum_{i=1}^{t} \gamma_{nk}^{(i)} \cdot \left| \alpha_{nk}^{(i)} - \widehat{c}_i(\lambda_{nk}) e^{\imath n\theta_i} \alpha_{nk} \right|^2, \tag{27}$$

where $\gamma_{nk}^{(i)} = 0$ if $|\widehat{c}_i(\lambda_{nk})| < \tau$, for a given threshold τ, and $\gamma_{nk}^{(i)} = 1$ otherwise. This describes a decoupled least-squares problem for each coefficient α_{nk}, which can be solved efficiently. We remark that (27) is a basic version of Wiener filtering [4], which we use for simplicity of exposition. The result of this procedure for different values of t and a non-zero value of the noise ε is shown in Fig. 4.

6. Discussion

This paper presents a fast method for expanding a set of $L \times L$-images into the basis of eigenfunctions of the Laplacian on the disk. The approach calculates the expansion coefficients from interpolation of the Fourier-transform of the image on distinguished subsets of the frequency space, and relies on an integral identity of the Fourier-transform of the eigenfunctions. Unlike previous approaches [34], we demonstrate that our fast method is guaranteed to coincide with a dense, equivalent method up to a user-specified precision. Moreover, our method provides a natural

way to compute the convolution with radial functions. Potential extensions of the presented method include extending the method to three dimensions or other domains in two dimensions.

Acknowledgements. The authors would like to thank Joakim Andén and Yunpeng Shi for their helpful comments on a draft of this paper.

References

[1] Alex H Barnett. Aliasing error of the $\exp(\beta\sqrt{1-z^2})$ kernel in the nonuniform fast Fourier transform. *Applied and Computational Harmonic Analysis*, 51:1–16, 2021.

[2] Alexander H Barnett, Jeremy Magland, and Ludvig af Klinteberg. A parallel nonuniform fast Fourier transform library based on an "Exponential of semicircle" kernel. *SIAM Journal on Scientific Computing*, 41(5):C479–C504, 2019.

[3] Jean-Paul Berrut and Lloyd N Trefethen. Barycentric lagrange interpolation. *SIAM review*, 46(3):501–517, 2004.

[4] Tejal Bhamre, Teng Zhang, and Amit Singer. Denoising and covariance estimation of single particle cryo-EM images. *Journal of structural biology*, 195(1):72–81, 2016.

[5] Xiuyuan Cheng, Qiang Qiu, Robert Calderbank, and Guillermo Sapiro. Rotdcf: Decomposition of convolutional filters for rotation-equivariant deep networks. *arXiv preprint arXiv:1805.06846*, 2018.

[6] Yifan Cheng, Nikolaus Grigorieff, Pawel A Penczek, and Thomas Walz. A primer to single-particle cryo-electron microscopy. *Cell*, 161(3):438–449, 2015.

[7] Taco S Cohen and Max Welling. Steerable CNNs. *arXiv preprint arXiv:1612.08498*, 2016.

[8] Yves Colin de Verdière. On the remainder in the Weyl formula for the Euclidean disk. *Séminaire de théorie spectrale et géométrie*, 29:1–13, 2010.

[9] *NIST Digital Library of Mathematical Functions.* http://dlmf.nist.gov/, Release 1.1.5 of 2022-03-15. F. W. J. Olver, A. B. Olde Daalhuis, D. W. Lozier, B. I. Schneider, R. F. Boisvert, C. W. Clark, B. R. Miller, B. V. Saunders, H. S. Cohl, and M. A. McClain, eds.

[10] A. Dutt, M. Gu, and V. Rokhlin. Fast Algorithms for Polynomial Interpolation, Integration, and Differentiation. *SIAM Journal on Numerical Analysis*, 33(5):1689–1711, 1996.

[11] Alok Dutt and Vladimir Rokhlin. Fast fourier transforms for nonequispaced data. *SIAM Journal on Scientific computing*, 14(6):1368–1393, 1993.

[12] Joachim Frank. *Three-Dimensional Electron Microscopy of Macromolecular Assemblies: Visualization of Biological Molecules in Their Native State*. Oxford University Press, 2006.

[13] Carl Friedrich Gauss. *De nexu inter multitudinem classium, in quas formae binariae secundi gradus distribuuntur, earumque determinantem*, volume 2 of *Cambridge Library Collection - Mathematics*, page 269–291. Cambridge University Press, 2011.

[14] Leslie Greengard and June-Yub Lee. Accelerating the nonuniform fast Fourier transform. *SIAM review*, 46(3):443–454, 2004.

[15] Hangtian Guo, Yan Gao, Tinghan Li, Tingting Li, Yuchi Lu, Le Zheng, Yue Liu, Tingting Yang, Feiyang Luo, Shuyi Song, et al. Structures of Omicron spike complexes and implications for neutralizing antibody development. *Cell reports*, 39(5):110770, 2022.

[16] Godfrey Harold Hardy and Edmund Landau. The lattice points of a circle. *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character*, 105(731):244–258, 1924.

[17] Victor Ivrii. 100 years of Weyl's law. *Bulletin of Mathematical Sciences*, 6(3):379–452, 2016.

[18] Alexander Kurganov and Jeffrey Rauch. The order of accuracy of quadrature formulae for periodic functions. In *Advances in phase space analysis of partial differential equations*, pages 155–159. Springer, 2009.

[19] Boris Landa and Yoel Shkolnisky. Approximation scheme for essentially bandlimited and space-concentrated functions on a disk. *Applied and Computational Harmonic Analysis*, 43(3):381–403, 2017.

[20] Boris Landa and Yoel Shkolnisky. Steerable principal components for space-frequency localized images. *SIAM journal on imaging sciences*, 10(2):508–534, 2017.

[21] Catherine L Lawson, Ardan Patwardhan, Matthew L Baker, Corey Hryc, Eduardo Sanz Garcia, Brian P Hudson, Ingvar Lagerstedt, Steven J Ludtke, Grigore Pintilie, Raul Sala,

et al. EMDataBank unified data resource for 3DEM. *Nucleic Acids Research*, 44(D1):D396–D403, 2016.

[22] June-Yub Lee and Leslie Greengard. The type 3 nonuniform FFT and its applications. *Journal of Computational Physics*, 206(1):1–5, 2005.

[23] Eva Nogales and Sjors HW Scheres. Cryo-EM: a unique tool for the visualization of macromolecular complexity. *Molecular cell*, 58(4):677–689, 2015.

[24] George A Papakostas, Yiannis S Boutalis, Dimitris A Karras, and Basil G Mertzios. A new class of Zernike moments for computer vision applications. *Information Sciences*, 177(13):2802–2819, 2007.

[25] Aaditya Rangan, Marina Spivak, Joakim Andén, and Alex Barnett. Factorization of the translation kernel for fast rigid image alignment. *Inverse Problems*, 36(2):024001, 2020.

[26] Tanvir R Shaikh, Haixiao Gao, William T Baxter, Francisco J Asturias, Nicolas Boisset, Ardean Leith, and Joachim Frank. SPIDER image processing for single-particle reconstruction of biological macromolecules from electron micrographs. *Nature protocols*, 3(12):1941–1974, 2008.

[27] Yu-hsuan Shih, Garrett Wright, Joakim Andén, Johannes Blaschke, and Alex H Barnett. cuFINUFFT: a load-balanced GPU library for general-purpose nonuniform FFTs. In *2021 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*, pages 688–697. IEEE, 2021.

[28] David Slepian and Henry O Pollak. Prolate spheroidal wave functions, Fourier analysis and uncertainty—I. *Bell System Technical Journal*, 40(1):43–63, 1961.

[29] Nico M Temme. *Special functions: An introduction to the classical functions of mathematical physics*. John Wiley & Sons, 1996.

[30] JG Van der Corput. Neue zahlentheoretische Abschätzungen. *Mathematische Annalen*, 89(3):215–254, 1923.

[31] Zernike von F. Beugungstheorie des schneidenver-fahrens und seiner verbesserten form, der phasenkontrastmethode. *Physica*, 1(7-12):689–704, 1934.

[32] RH Wade. A brief look at imaging and contrast transfer. *Ultramicroscopy*, 46(1-4):145–156, 1992.

[33] Maurice Weiler, Fred A Hamprecht, and Martin Storath. Learning steerable filters for rotation equivariant CNNs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 849–858, 2018.

[34] Zhizhen Zhao, Yoel Shkolnisky, and Amit Singer. Fast steerable principal component analysis. *IEEE transactions on computational imaging*, 2(1):1–12, 2016.

[35] Zhizhen Zhao and Amit Singer. Rotationally invariant image representation for viewing direction classification in cryo-EM. *Journal of structural biology*, 186(1):153–166, 2014.

Email address: marsnich@oregonstate.edu

Email address: hm6655@princeton.edu

Email address: amits@math.princeton.edu