COMPUTING WITH FUNCTIONS IN SPHERICAL AND POLAR GEOMETRIES II. THE DISK

HEATHER WILBER*, ALEX TOWNSEND†, AND GRADY B. WRIGHT‡

Abstract. A collection of algorithms is described for numerically computing with smooth functions defined on the unit disk. Low rank approximations to functions in polar geometries are formed by synthesizing the disk analogue of the double Fourier sphere method with a structure-preserving variant of iterative Gaussian elimination that is shown to converge geometrically for certain analytic functions. This adaptive procedure is near-optimal in its sampling strategy, producing approximants that are stable for differentiation and facilitate the use of FFT-based algorithms in both variables. The low rank form of the approximants is especially useful for operations such as integration and differentiation, reducing them to essentially 1D procedures, and it is also exploited to formulate a new fast disk Poisson solver that computes low rank approximations to solutions. This work complements a companion paper (Part I) on computing with functions on the surface of the unit sphere.

Key words. low rank approximation, Gaussian elimination, functions, approximation theory

AMS subject classifications. 65D05

1. Introduction. Polar geometries play a central role in scientific computing, with applications in fluid dynamics [22, 36], optics [25], and astrophysics [15, 32]. Advances in these areas require effective representations for functions on the unit disk, and compressed representations of such functions have become increasingly important. We develop a novel variant of iterative Gaussian elimination (GE) that adaptively constructs low rank approximants with near-optimal compression properties; this enables fast and spectrally accurate computations with functions on the disk.

Methods that represent functions on the disk with expansions in the Chebyshev–Fourier basis allow for the use of fast transforms [11, 12, 37], but may not maintain regularity at the origin of the disk when used with GE. Alternatively, representations employing expansions that incorporate regularity in the basis are not readily associated with fast transforms [47]. Unsatisfied with having to choose between either regularity at the origin or fast transforms, we propose an approach that attempts to prioritize both. Combining low rank function approximation with an interpolation method that samples functions over the unit disk in a way that is analogous to the double Fourier sphere (DFS) method [12], we construct approximants with several desirable properties: (1) A structure that permits the use of fast transforms based on the fast Fourier transform (FFT) in both variables, (2) regularity over the origin of the disk, and (3) a near-optimal underlying interpolation grid that does not oversample near the origin.

Using this idea, we have created an integrated computational framework for working with functions in polar geometries. This includes the development of algorithms for integration, function evaluation, vector calculus, and a fast Poisson solver. Our software is publicly available through the open source Chebfun software system written in MATLAB [10]. This development allows investigators to compute

*Center for Applied Mathematics, Cornell University, Ithaca, NY 14853. (hdw27@cornell.edu). This work is supported by a grant from the NASA Idaho Space Grant Consortium.

†Department of Mathematics, Cornell University, Ithaca, NY 14853. (townsend@cornell.edu). This work is supported by National Science Foundation grant No. 1522577.

‡Department of Mathematics, Boise State University, Boise, ID 83725-1555. (gradywright@boisestate.edu). This work is supported by National Science Foundation grant DMS 1160379.

in polar geometries without concern for the underlying discretization or procedural details, providing an intuitive platform for data-driven computations, explorations and visualizations with functions on the unit disk. Various examples are available at www.chebfun.org/examples for the reader to explore.

Part I of this two-part series of papers developed a structure-preserving, iterative variant of Gaussian elimination (GE) for computing with functions on the surface of the unit sphere [43]. Here, we extend the ideas of [43] to functions defined on the unit disk. We also include several new results that were not discussed in Part I. In Section 3.4, we prove that our structure-preserving GE procedure converges geometrically for functions that are analytic in a sufficiently large region in the complex plane. Section 5 describes a new Poisson solver that constructs near-optimal low rank approximations to solutions, and is conceptually quite different from the Poisson solver described in [43]. Additional new results include a weighted singular value decomposition algorithm (Section 4.5), and an extended discussion on the near-optimality of the GE procedure (Section 3.5).

The paper is structured as follows: First, we review existing techniques for computing with functions on the disk (Section 2), including a discussion of the disk analogue to the DFS method. A brief review of low rank function approximation in Section 3 is followed by a detailed description of the structure-preserving GE procedure applied to functions on the disk. A collection of fast algorithms for computing with the resulting low rank approximants is given in Section 4, and a fast disk Poisson solver for computing solutions in low rank form is described in Section 5.

2. Existing techniques for computations on the disk. There is an extensive literature on numerical methods for computing with functions on the disk. An overview in the context of solving Poisson's equation is given in [7]. We briefly review a selection of these strategies.

2.1. Radial basis functions. As a mesh-free method, radial basis functions can be used for applications on many types of geometries [13]. Specific studies of global approximations on the disk include [19, 21], where the interpolation points are arranged so that the computational cost of the method reduces from $\mathcal{O}(N^3)$ to $\mathcal{O}(N \log N)$ operations, where N is the number of function samples taken. Ill-conditioning can cause a loss of 3-5 digits of accuracy in problems of moderate size, but in most applications, this is perfectly acceptable. However, this prevents the construction of approximants that are accurate to machine precision, which is what we require.

2.2. Conformal mapping. Using the inverse of the cosine leminiscate function, a function f on the unit disk can be mapped conformally to the unit square [1, 35]. This mapping avoids introducing a potentially problematic singularity at the origin and allows f to be expressed as a bivariate Chebyshev expansion so that FFT-based transforms are applicable. Unfortunately, the mapping introduces four new artificial singularities corresponding to the corners of the square. Interpolation points unnaturally cluster near these singularities, resulting in excessive oversampling that diminishes the computational efficiency gained from the use of the FFT. In contrast, our approach enables the use of FFT-based transforms, while employing low rank approximation to avoid overresolving functions near the origin.

2.3. Basis expansions. A function $f(x, y)$ defined in Cartesian coordinates on the unit disk can be converted to a function in polar coordinates, $f(\theta, \rho)$, through the

transformation

$$x = \rho\cos\theta, \quad y = \rho\sin\theta, \qquad (\theta,\rho) \in [-\pi,\pi]\times[0,1]. \tag{2.1}$$

This change of variables relates a function on the disk to a function defined on a rectangular domain, where advantageous algorithms can often be employed. Noting that functions on the disk are periodic in the angular variable, θ, a sufficiently smooth function f can be approximated by a Fourier expansion:

$$f(\theta,\rho) \approx \sum_{k=-n/2}^{n/2-1} \phi_k(\rho) e^{ik\theta}, \qquad (\theta,\rho) \in [-\pi,\pi]\times[0,1], \tag{2.2}$$

where n is an even integer. It is not obvious what expansion should be employed for representing the function $\phi_k(\rho)$. Three common choices are:

- **Bessel expansions:** A natural analogue of the trigonometric and spherical harmonic expansions, Bessel expansions are derived from the eigenfunctions of the Laplace operator in polar coordinates [9]. Here, assuming that $f(\theta,1) = 0$ for $\theta \in [-\pi,\pi]$, we write $\phi_k(\rho) = \sum_{\ell=0}^{m-1} a_{\ell k} J_k(\omega_{k\ell}\rho)$, $\rho \in [0,1]$, where $J_k(z)$ is the kth order Bessel function, and $\omega_{k\ell}$ is the ℓth positive root of $J_k(z)$ [29, (10.23)]. The expansion can also be modified to allow for functions that are nonzero at the boundary of the disk. This choice guarantees the expansion is smooth at the origin, but to compute the expansion coefficients, one must approximate integrals involving Bessel functions. While fast algorithms for such computations exist, they are particularly effective only when the parameter k is small [20, 39]. More generalized algorithms typically involve significant precomputational costs [31], and this limits their effectiveness in a regime where functions are resolved on adaptive grids.
- **One-sided Jacobi polynomial expansions:** Writing $\phi_k(\rho)$ as an expansion over the one-sided Jacobi polynomials results in an expansion of $f(\theta,\rho)$ in the Zernike polynomial basis [5, 48]. This set of polynomials is considered theoretically analogous to the Legendre polynomials due to its orthogonality properties [5], and is often the basis of choice for approximation on the disk. More recently, a whole hierarchy of bases related to the one-sided Jacobi polynomials were employed to capture the regularity of vector- and tensor-valued functions on the disk [47]. As before, this choice guarantees the expansion is smooth at the origin, but fast algorithms for computing the expansion coefficients are not efficient in our setting due to precomputational costs [31].
- **Chebyshev expansions:** Expanding $\phi_k(\rho)$ in the Chebyshev basis results in a truncated Chebyshev–Fourier expansion of f, i.e.,

$$f(\theta,\rho) \approx \sum_{k=-n/2}^{n/2-1} \sum_{\ell=0}^{m-1} a_{\ell k} T_\ell(2\rho-1) e^{ik\theta}, \qquad (\theta,\rho) \in [-\pi,\pi]\times[0,1], \tag{2.3}$$

  where T_ℓ is the degree ℓ Chebyshev polynomial defined on $[-1,1]$. Given samples of f on an $m \times n$ Chebyshev–Fourier tensor product grid over $[-\pi,\pi]\times[0,1]$, the coefficients in (2.3) can be computed in $\mathcal{O}(mn\log(mn))$ operations via the FFT. Unfortunately, this grid is artificially clustered near $\rho = 0$ [12], and this choice of basis does not naturally impose any regularity at $\rho = 0$. Our approach alleviates both of these drawbacks by combining the disk analogue to the DFS (see Section 2.4) with a structure-preserving low rank construction procedure (see Section 3).

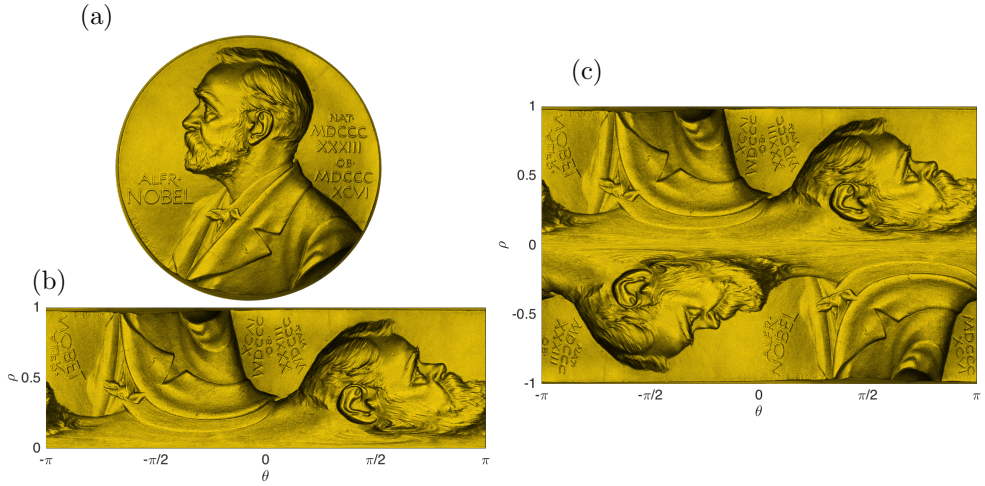

FIG. 1. *The disk analogue of the DFS method applied to the Nobel prize medal. (a) The medal. (b) The projection of the medal using polar coordinates. (c) The medal after applying the disk analogue to the DFS method. This is a BMC-II "function" that is periodic in θ and defined over $\rho \in [-1, 1]$.*

2.4. The disk analogue of the double Fourier sphere method. The disk analogue of the DFS method proceeds by constructing a Chebyshev–Fourier expansion of a function defined on $[-\pi, \pi] \times [-1, 1]$, instead of $[-\pi, \pi] \times [0, 1]$. This strategy "doubles" f over the disk in the sense that f is sampled twice, but $\rho = 0$ is no longer treated as a boundary. Mathematically, this doubled extension of f, which we will call $\tilde{f}$, can be expressed by defining $g(\theta, \rho)$ and $h(\theta, \rho)$ on $[0, \pi] \times [0, 1]$, so that $g(\theta, \rho) = f(\theta - \pi, \rho)$ and $h(\theta, \rho) = f(\theta, \rho)$. Then,

$$\tilde{f}(\theta, \rho) = \begin{cases} g(\theta + \pi, \rho), & (\theta, \rho) \in [-\pi, 0] \times [0, 1], \\ h(\theta, \rho), & (\theta, \rho) \in [0, \pi] \times [0, 1], \\ g(\theta, -\rho), & (\theta, \rho) \in [0, \pi] \times [-1, 0], \\ h(\theta + \pi, -\rho), & (\theta, \rho) \in [-\pi, 0] \times [-1, 0]. \end{cases} \tag{2.4}$$

This idea is conceptually analogous to the DFS method [28], which is used for approximating functions on the surface of the unit sphere [43].

A useful connection between the DFS method and its disk analogue is the presence of similar structure in the extended functions. We observe in (2.4) that $\tilde{f}$ possesses *block-mirror centrosymmetric* (BMC) structure [43], and refer to functions that satisfy (2.4) as BMC functions.

The BMC structure of $\tilde{f}$ can be intuitively described as

$$\tilde{f} = \begin{bmatrix} g & h \\ \texttt{flip}(h) & \texttt{flip}(g) \end{bmatrix}, \tag{2.5}$$

where `flip` refers to the MATLAB command that reverses the order of the rows of a matrix. This is also called a glide reflection in group theory [26, §8.1].

In addition to having BMC structure and being periodic in θ, $\tilde{f}$ must be constant along the line representing the origin of the disk, $\rho = 0$. This feature of $\tilde{f}$ is not shared by all BMC functions. For example, the BMC function $\tilde{f}(\theta, \rho) = \sin 2\theta \cos 2\rho$ is not constant along $\tilde{f}(\theta, 0)$ for $\theta \in [-\pi, \pi]$, and therefore does not correspond to a

continuous function on the disk. To capture this important aspect of BMC functions associated with the disk, we define the following variant:

DEFINITION 2.1. *(BMC-II function) A function* $\tilde{f} : [-\pi, \pi] \times [-1, 1] \to \mathbb{C}$ *is a Type-II BMC (BMC-II) function if it is a BMC function and* $f(\cdot, 0) = \alpha$*, where* α *is a constant.*

An analogous variant for computing on the sphere, the BMC-I function, is defined to be constant along two lines corresponding to the north and south poles of the sphere [43].

Figure 1 displays the analogue of the DFS method applied to the Nobel Prize medal and illustrates BMC-II structure. Since every function f on the disk corresponds to a BMC-II function $\tilde{f}$ that is 2π-periodic in θ, we apply our approximation technique and all subsequent algorithms on $\tilde{f}$, with rigid adherence to preserving the BMC-II structure at every step. Calculations performed on $\tilde{f}$ always correspond to a computation on the original function, f, and consistently remain associated with the geometry of the disk. For example, smooth functions with BMC-II structure are always continuously differentiable over $\rho = 0$. In Section 4, we discuss the differentiation of BMC-II functions in more detail.

The strategy of doubling up interpolation grids on the disk to reduce the redundancy of sampling near $\rho = 0$ in spectral collocation methods is well established [12, 44], and several variants have been proposed [11, 18, 37]. These doubling strategies alleviate some, but not all, of the issues associated with oversampling near the origin. Our approach is different in that it combines a doubling strategy with a low rank approximation procedure. Low rank methods provide compressed representations of functions and can therefore further alleviate issues related to the overresolution of functions near the origin of the disk (see Figure 4).

2.5. Software. Our software for computing with functions on the unit disk is called Diskfun.[1] It is implemented within MATLAB as a part of Chebfun [10], and is accessed through the creation of objects called diskfuns. Below, we display the MATLAB code used to represent the function

$$f(\theta, \rho) = \cos\big(3\pi\rho\big) + \sin\big(2\rho\sin\theta - .4\big)$$

as a diskfun object:

```
f = diskfun(@(t,r) cos(3*pi*r)+sin(2*r.*sin(t)-.4),'polar')
f =
  diskfun object:
    domain         rank    vertical scale
   unit disk        13           2
```

The printout provides the numerical *rank* of the function, discussed in Section 3, and it also displays the vertical scale, an approximation of the absolute maximum value of f.

The default setting of Diskfun assumes that functions are supplied in Cartesian coordinates. However, diskfun objects can be constructed from function handles in polar coordinates by adding the flag `'polar'` to the construction command, as shown above. Once a diskfun is created, users have access to a large number of algorithms tailored to functions defined on the disk via overloaded MATLAB commands (see

[1] After our software was developed and posted on GitHub, another software system named "diskfun" was released in the Approxfun software system written in Julia. It is not related to this work.

Section 4). For example, integration of f is performed by the `sum` command, and differentiation is performed by `diff`.

3. Low rank approximation for functions on the disk. In [41], a low rank approximation method for computing with 2D functions on bounded rectangular domains is described. The authors construct compressed representations of bivariate functions that facilitate the use of essentially 1D algorithms in subsequent computations. This makes it especially useful in relation to Chebfun, where efficient 1D procedures are well established and highly optimized. Here, we develop an analogous technique for the polar setting.

A nonzero function $\tilde{f}(\theta,\rho)$ is a rank 1 function if it can be written as a product of two univariate functions, i.e., $\tilde{f}(\theta,\rho) = c(\rho)r(\theta)$. A function $\tilde{f}$ is of rank at most K if it can be written as a sum of K rank 1 functions. While most functions are mathematically of infinite rank, smooth functions can often be approximated to machine precision with a rank K truncation, i.e.,

$$\tilde{f}(\theta,\rho) \approx \sum_{j=1}^{K} c_j(\rho) r_j(\theta), \tag{3.1}$$

for some relatively small K [41]. Below, we develop an efficient procedure for constructing rank K approximants of BMC-II functions that preserve BMC-II structure.

3.1. Iterative Gaussian elimination on functions. Given a matrix A of rank n, $K < n$ steps of Gaussian elimination (GE) with complete or rook pivoting can often be used to construct a near-best rank K approximation to A, provided that the singular values of A decay to zero sufficiently fast [14]. Methods related to GE, such as adaptive cross approximation [2], two-sided interpolative decomposition [17], and Geddes–Newton approximation [8] can be used to find low rank approximations to multivariate functions. In [41], such approximations are constructed using an adaptive, iterative variant of GE with complete pivoting, and we will extend this idea to the approximation of functions in polar geometries.

Given the function $\tilde{f}$, denote the maximum absolute value of $\tilde{f}$ for $(\theta,\rho) \in [-\pi,\pi]\times[-1,1]$ by $\tilde{f}(\theta^*,\rho^*)$. This value serves as a pivot. A GE step with complete pivoting proceeds by forming a rank 1 function from this pivot and subtracting it from $\tilde{f}$:

$$\tilde{f}(\theta,\rho) \quad \longleftarrow \quad \tilde{f}(\theta,\rho) - \underbrace{\frac{\tilde{f}(\theta^*,\rho)\tilde{f}(\theta,\rho^*)}{\tilde{f}(\theta^*,\rho^*)}}_{\text{A rank 1 approx. to } \tilde{f}}. \tag{3.2}$$

In this scheme, functions of the form $\tilde{f}(\theta^*,\rho)$ are referred to as "column slices" of $\tilde{f}$. Similarly, functions of the form $\tilde{f}(\theta,\rho^*)$ are "row slices". The step in (3.2) zeros out the row and column slices containing the pivot. Since $\tilde{f}$ may be of infinite rank, the GE procedure is terminated after the absolute maximum of the residual falls below some specified relative tolerance, such as the product of machine epsilon and the (approximate) maximum value of the function. The number of steps required to achieve this is an upper bound on the *numerical rank* of $\tilde{f}$, which is the minimum rank required to approximate $\tilde{f}$ to machine precision using any bounded function of finite rank [38].

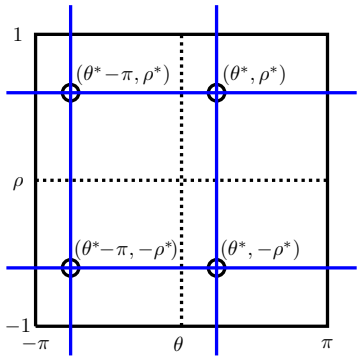

FIG. 2. *A 2×2 pivot (black circles) and corresponding column and row slices (blue lines) used in a GE step on $\tilde{f}$ to preserve the BMC structure of a function.*

Applying the GE procedure to $\tilde{f}$ for K steps, a rank K approximation is constructed:

$$\tilde{f}(\theta, \rho) \approx \sum_{j=1}^{K} d_j c_j(\rho) r_j(\theta). \tag{3.3}$$

Here, d_j is a coefficient related to the GE pivots, and $c_j(\rho)$ and $r_j(\theta)$ are the jth column slice and row slice, respectively, constructed during the GE procedure.

Unfortunately, this GE procedure does not preserve BMC-II symmetry and therefore destroys the association between $\tilde{f}$ and a continuous function on the disk. In [43], a variation of GE that preserves symmetry is described for BMC functions related to the sphere. Crucially, this method only depends on the BMC structure of the function, and not on any additional features related to spherical geometries per se. With some modifications, as we now describe, this procedure also applies to BMC-II functions associated with the disk.

3.2. Structure-preserving Gaussian elimination. The structure-preserving GE algorithm presented in [43] performs a GE step similar to (3.2), but with the scalar pivot replaced with the following 2×2 pivot *matrix*:

$$M = \begin{bmatrix} \tilde{f}(\theta^* - \pi, \rho^*) & \tilde{f}(\theta^*, \rho^*) \\ \tilde{f}(\theta^* - \pi, -\rho^*) & \tilde{f}(\theta^*, -\rho^*) \end{bmatrix}, \tag{3.4}$$

where $(\theta^*, \rho^*) \in [0, \pi] \times [0, 1]$ are fixed values selected by the pivoting strategy described in Figure 3. To understand why this is an appropriate choice, note that BMC symmetry is entirely characterized by the following two equalities: $\tilde{f}(\theta^* - \pi, \rho) = \tilde{f}(\theta^*, -\rho)$, $\rho \in [-1, 1]$, and $\tilde{f}(\theta, \rho^*) = \tilde{f}(\theta - \pi, -\rho^*)$, $\theta \in [-\pi, \pi]$. Figure 2 shows that the location of the entries of M correspond to the intersections of these row and column slices. Letting $\tilde{f}(\theta^* - \pi, \rho^*) = a$ and $\tilde{f}(\theta^*, \rho^*) = b$, (3.4) can be written as the centrosymmetric matrix

$$M = \begin{bmatrix} a & b \\ b & a \end{bmatrix}. \tag{3.5}$$

Assuming M^{-1} exists, a GE step with the pivot matrix M is given by

$$\tilde{f}(\theta,\rho) \quad \longleftarrow \quad \tilde{f}(\theta,\rho) - \underbrace{\begin{bmatrix}\tilde{f}(\theta^*-\pi,\rho) & \tilde{f}(\theta^*,\rho)\end{bmatrix} M^{-1} \begin{bmatrix}\tilde{f}(\theta,\rho^*)\\ \tilde{f}(\theta,-\rho^*)\end{bmatrix}}_{=\tilde{s}(\theta,\rho)}. \tag{3.6}$$

We now show that the GE step in (3.6) preserves BMC symmetry of $\tilde{f}$.

LEMMA 3.1. *Given a BMC function $\tilde{f}$, the update $\tilde{s}$ in* (3.6) *is also a BMC function. That is, the GE step in* (3.6) *preserves BMC-symmetry.*

Proof. To show that $\tilde{s}(\theta,\rho)$ has BMC structure, we employ *quasimatrices*.[2]

Let J denote the 2×2 exchange matrix, so that for a matrix $A\in\mathbb{C}^{2\times n}$, JA reverses the rows of A. Let $\mathcal{J}$ be the reflection operator, $\mathcal{J}:\tilde{s}(\theta,\rho)\to\tilde{s}(\theta,-\rho)$. Now we use blocks of quasimatrices to rewrite $\tilde{s}$. Writing $\tilde{f}$ in terms of the functions g and h given in (2.4), we have $M=\begin{bmatrix}g(\theta^*,\rho^*) & h(\theta^*,\rho^*)\\ h(\theta^*,\rho^*) & g(\theta^*,\rho^*)\end{bmatrix}$. Let Q be the $[0,\pi]\times 2$ quasimatrix defined as $Q=\begin{bmatrix}g(\theta^*,\rho) & | & h(\theta^*,\rho)\end{bmatrix}$, and let P be the $[0,1]\times 2$ quasimatrix defined as $P=\begin{bmatrix}\mathrm{g}(\theta,\rho^*) & | & h(\theta,\rho^*)\end{bmatrix}$. Then, $\tilde{s}$ in (3.6) can be written as

$$\tilde{s}=\begin{bmatrix}Q\\ \mathcal{J}(QJ)\end{bmatrix} M^{-1}\begin{bmatrix}P^T & JP^T\end{bmatrix}. \tag{3.7}$$

Since M^{-1} is centrosymmetric, it commutes with J. Using this fact, (3.7) becomes

$$\tilde{s}=\begin{bmatrix}QM^{-1}P^T & QM^{-1}JP^T\\ \mathcal{J}(QM^{-1}JP^T) & \mathcal{J}(QM^{-1}P^T)\end{bmatrix}, \tag{3.8}$$

which, by the definition of $\mathcal{J}$, is a BMC function. ∎

Lemma 3.1 demonstrates that (3.6) provides a structure-preserving GE procedure for BMC functions that can be used to construct a low rank approximation to $\tilde{f}$ as in (3.3).[3] However, this relies on the fact that M is invertible, which may not always be the case. For example, M is singular for any BMC function that is π–periodic in θ. For this reason, we must replace M^{-1} in (3.6) with $M^{\dagger_\epsilon}$, the ϵ-pseudoinverse of M [16, Sec. 5.5.2]. The matrix $M^{\dagger_\epsilon}$ is associated with the singular values of M and a parameter $\epsilon>0$. We will discuss the choice of ϵ in Section 3.3, and an explicit formula for $M^{\dagger_\epsilon}$ is given in [43]. Using $M^{\dagger_\epsilon}$, the amended GE step is expressed by

$$\tilde{f}(\theta,\rho) \quad \longleftarrow \quad \tilde{f}(\theta,\rho) - \begin{bmatrix}\tilde{f}(\theta^*-\pi,\rho) & \tilde{f}(\theta^*,\rho)\end{bmatrix} M^{\dagger_\epsilon} \begin{bmatrix}\tilde{f}(\theta,\rho^*)\\ \tilde{f}(\theta,-\rho^*)\end{bmatrix}. \tag{3.9}$$

Lemma 3.1 also holds for (3.9) because, like M^{-1}, $M^{\dagger_\epsilon}$ is centrosymmetric.

The strategy used to select each pivot matrix is important, as it relates to the efficiency and convergence of the GE procedure. The 2×2 analogue of complete pivoting proceeds by choosing $(\theta^*,\rho^*)\in[0,\pi]\times[0,1]$ such that $\sigma_1(M)$ is maximized over all M, where $\sigma_1(M)$ is the larger of the two singular values of M. Given the simple form of M in (3.5), it is easy to see that $\sigma_1(M)=\max\{|a+b|,|a-b|\}$. In

[2] A quasimatrix A of size $[a,b]\times n$ is a matrix with n columns, where each column is a function defined on the interval $[a,b]$ [42].

[3] The function $\tilde{s}$ in (3.6) is rank 2 and can be split into two rank 1 BMC functions (see Section 3.3).

Algorithm: Structure-preserving GE on BMC functions

Input: A BMC function $\tilde{f}$ and a coupling parameter $0 \leq \alpha \leq 1$.

Output: A structure-preserving low rank approximation $\tilde{f}_k$ to $\tilde{f}$.

Set $\tilde{f}_0 = 0$ and $\tilde{e}_0 = \tilde{f}$.

for $k = 1, 2, 3, \ldots,$

Find (θ_k, ρ_k) such that $M = \begin{bmatrix} a & b \\ b & a \end{bmatrix}$, where $a = \tilde{e}_{k-1}(\theta_{k-1} - \pi, \rho_{k-1})$ and $b = \tilde{e}_{k-1}(\theta_{k-1}, \rho_{k-1})$ has maximal $\sigma_1(M)$.

Set $\epsilon = \alpha\sigma_1(M)$.

$$\tilde{e}_k = \tilde{e}_{k-1} - \begin{bmatrix} \tilde{e}_{k-1}(\theta_k - \pi, \rho) & \tilde{e}_{k-1}(\theta_k, \rho) \end{bmatrix} M^{\dagger_\epsilon} \begin{bmatrix} \tilde{e}_{k-1}(\theta, \rho_k) \\ \tilde{e}_{k-1}(\theta, -\rho_k) \end{bmatrix}.$$

$$\tilde{f}_k = \tilde{f}_{k-1} - \begin{bmatrix} \tilde{e}_{k-1}(\theta_k - \pi, \rho) & \tilde{e}_{k-1}(\theta_k, \rho) \end{bmatrix} M^{\dagger_\epsilon} \begin{bmatrix} \tilde{e}_{k-1}(\theta, \rho_k) \\ \tilde{e}_{k-1}(\theta, -\rho_k) \end{bmatrix}.$$

end

FIG. 3. *A continuous idealization of our structure-preserving GE procedure on BMC functions. In practice we use a discretization of this procedure and terminate it after a finite number of steps.*

practice, it is much more efficient to choose (θ^*, ρ^*) from a coarse, discrete grid on $[-\pi, \pi] \times [0, 1]$. This results in a large, but not necessarily maximal, value of $\sigma_1(M)$. Fortunately, GE is robust to these kinds of compromises, as a detailed analysis in [40] shows.

The above GE procedure preserves *general* BMC structure, but it does not preserve BMC-II structure: Nothing in (3.9) enforces that each constructed rank 1 function in (3.3) is constant along the line $\tilde{f}(\theta, 0)$. However, in the case where $\tilde{f}(\theta, 0) = 0$, each term in (3.3) constructed through (3.9) will possess BMC-II structure. This suggests a strategy for the case where $\tilde{f}(\theta, 0) \neq 0$. Since $\tilde{f}(\theta, 0)$ is constant by Definition 2.1, we deliberately choose the first GE step to zero out $\tilde{f}(\theta, 0)$ by subtracting off a rank 1 term that is constant in the θ direction:

$$\tilde{f}(\theta, \rho) \quad \longleftarrow \quad \tilde{f}(\theta, \rho) - \tilde{f}(\theta^*, \rho). \tag{3.10}$$

Since the update to $\tilde{f}$ is zero along $\tilde{f}(\theta, 0)$ after this modification, each additional rank 1 term constructed through continued applications of (3.9) possesses BMC-II structure.

A continuous idealization of the BMC-preserving GE process is shown in Figure 3. In practice, the algorithm implemented in Diskfun proceeds in two phases; this process is identical to the method described in [41], except with 2×2 pivots. The result is a low rank approximation to $\tilde{f}$ of the form (3.3). We represent each of the $r_j(\theta)$ and $c_j(\rho)$ functions in (3.3) using Fourier and Chebyshev interpolants, respectively. This process is achieved in $\mathcal{O}(K^3 + K^2(m+n))$ operations [41], where K is the numerical rank of the function, and m and n are the maximum Chebyshev and Fourier coefficients required to resolve the functions $c_j(\rho)$ and $r_j(\theta)$, respectively, to machine precision.

The example in Figure 4 illustrates the form of the final approximant. Each $c_j(\rho)$ defines a radial "slice" of the function, and each $r_j(\theta)$ defines a circular "slice". To form these slices, the GE algorithm adaptively samples $\tilde{f}$ along a sparse collection of lines referred to as the *skeleton*, and constructs a rank K approximant of the form

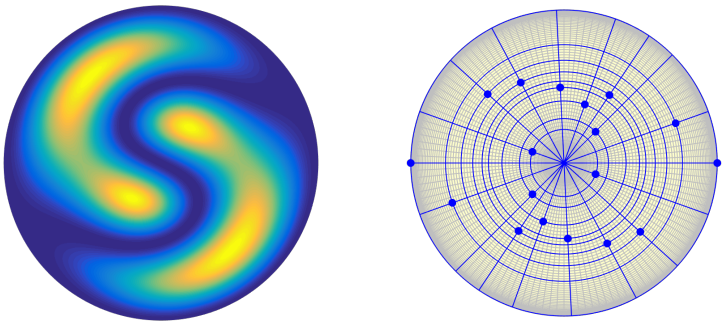

FIG. 4. *Left: The function* $f(\theta,\rho) = -\cos((\sin(\pi\rho)\cos(\theta) + \sin(2\pi\rho)\sin(\theta))/4)$ *on the unit disk, constructed with the* `diskfun` *command* `f = diskfun(@(t,r) -cos((sin(pi*r).*cos(t)+sin(2*pi*r).*sin(t))/4),'polar')` *and plotted with the command* `plot(f)`. *Right: The skeleton used to approximate* f*, plotted with the command* `plot(f,'.-')`. *The blue dots are the pivot locations taken by GE. The GE procedure samples* f *at* m *Chebyshev points along each blue line, and* n *equispaced points along each blue circle, where* m *and* n *correspond to number of Chebyshev coefficients and Fourier modes, respectively, in* (3.3). *The underlying tensor product grid (in gray) shows the sample points required to approximate* f *to machine precision without the GE procedure applied to the DFS method. The overresolution of the tensor grid over the low rank skeleton can be seen.*

of (3.3). In this process, only $K^2 + K(m+n)$ samples are required to approximate $\tilde{f}$ to machine precision, as opposed to the mn samples required for the tensor product. As depicted in Figure 4, the use of low rank methods effectively counters the over-resolution issues associated with applying Chebyshev–Fourier tensor product grids on the disk.

3.3. A parity-based interpretation of structure-preserving GE. For an approximation to a function f on the disk to be continuous and differentiable at $\rho = 0$, the following properties must hold for the Fourier expansion of f given in (2.2):

(i) k is even $\implies \phi_k(\rho)$ is an even function,
(ii) k is odd $\implies \phi_k(\rho)$ is an odd function,
(iii) $k \neq 0 \implies \phi_k(0) = 0$.

In this section we show that these parity properties can be naturally recovered for the BMC-II function $\tilde{f}$, and are preserved by the GE procedure depicted in Figure 3.

Let $\tilde{f}$ be a BMC function defined via functions g and h in (2.4). Let $f^+ = g + h$ and $f^- = g - h$. Then, $\tilde{f}$ can be written as a sum of two BMC functions [43, Section 3.2]:

$$\tilde{f} = \frac{1}{2}\underbrace{\begin{bmatrix} f^+ & f^+ \\ \texttt{flip}(f^+) & \texttt{flip}(f^+) \end{bmatrix}}_{=\tilde{f}^+} + \frac{1}{2}\underbrace{\begin{bmatrix} f^- & -f^- \\ -\texttt{flip}(f^-) & \texttt{flip}(f^-) \end{bmatrix}}_{=\tilde{f}^-}, \tag{3.11}$$

i.e., $\tilde{f} = \frac{1}{2}(\tilde{f}^+ + \tilde{f}^-)$. From (3.11), we can deduce that $\tilde{f}^+$ is an even function in ρ and π-periodic in θ, whereas $\tilde{f}^-$ is an odd function in ρ and π-antiperiodic in θ. This is equivalent to the statement of parity properties (i) and (ii), as π-periodic functions have only even Fourier modes and π-antiperiodic functions have only odd Fourier modes. While many techniques enforce these parity-based restrictions on the Fourier and Chebyshev coefficients of functions on the disk, relating these properties more generally to BMC-II functions allows one to apply these restrictions directly through

the *values* of a function, without ever using the coefficients. This is the premise our GE procedure operates on.

As shown in Section 3.2 of [43], we can write the GE step (3.9) as

$$\tilde{f}(\theta,\rho) \longleftarrow \frac{1}{2}(\tilde{f}^+(\theta,\rho) - m^+\tilde{f}^+(\theta^*,\rho)\tilde{f}^+(\theta,\rho^*)) + \frac{1}{2}(\tilde{f}^-(\theta,\rho) - m^-\tilde{f}^-(\theta^*,\rho)\tilde{f}^-(\theta,\rho^*)), \tag{3.12}$$

where m^+ and m^- are values[4] derived from the spectral decomposition of $M^{\dagger_\epsilon}$, and are given by

$$(m^+,m^-) = \begin{cases} (1/(a+b),0), & \text{if } |a-b| < \alpha|a+b|, \\ (0,1/(a-b)), & \text{if } |a+b| < \alpha|a-b|, \\ (1/(a+b),1/(a-b)), & \text{otherwise.} \end{cases} \tag{3.13}$$

Here, $0 < \alpha < 1$ is referred as the *coupling parameter* for the GE procedure, and α determines ϵ in $M^{\dagger_\epsilon}$: $\alpha = \epsilon/\sigma_1(M) = \epsilon/\max\{|a+b|,|a-b|\}$. The decomposition in (3.12) reveals an alternative interpretation of structure-preserving GE as a coupled process involving two standard GE procedures. If either of the first two cases of (3.13) is chosen, GE with complete pivoting is performed on only one term in (3.12), resulting in a rank 1 update. In the third case of (3.13), $M^{\dagger_\epsilon} = M^{-1}$, and a rank 2 update is achieved. It is desirable to perform as many rank 2 updates as possible, as this reduces the overall number of pivot searches required by the GE procedure. Too small a value of α may allow the use of M^{-1} when it is ill–conditioned, but choosing α too close to 1 hampers the efficiency of the procedure. We have experimented with several values for α and find that $\alpha = 1/100$ works well in practice. The role of α in the convergence rate of the GE procedure is discussed further in Section 3.4.

Following [43], we can exploit (3.12) to write the low rank approximation to $\tilde{f}$ as

$$\tilde{f}(\theta,\rho) \approx \sum_{j=1}^{K} d_j c_j(\rho) r_j(\theta) = \sum_{j=1}^{K^+} d_j^+ c_j^+(\rho) r_j^+(\theta) + \sum_{j=1}^{K^-} d_j^- c_j^-(\rho) r_j^-(\theta), \tag{3.14}$$

where $K^+ + K^- = K$. Here, the functions $c_j^+(\rho)$ and $r_j^+(\theta)$ for $1 \le j \le K^+$ are even and π-periodic, respectively, while $c_j^-(\rho)$ and $r_j^-(\theta)$ for $1 \le j \le K^-$ are odd and π-antiperiodic, respectively. The pivots, d^+ and d^-, are related to the 2×2 pivot matrix given in (3.4) [43]. If f is non-zero at the origin, the first step of the GE procedure is given by (3.10). This chooses $c_1^+(\rho) = \tilde{f}(\theta^*,\rho)$, $r_1^+(\theta) = 1$, and $d_1^+ = 1$, so that for $j > 1$, $c_j(0) = 0$. Crucially, this ensures that parity property (iii) is preserved in the decomposition.

Using (3.14), the parity properties of $\tilde{f}$ are given explicitly, and this can be used to simplify algorithmic procedures. An example is given in Section 4.3 on integration. This expression also clarifies why our approximants are stable for differentiation (see Section 4.4).

3.4. Convergence. In [43], it is shown that BMC structure-preserving GE exactly recovers BMC functions of finite rank. In this section, we prove that for certain analytic functions of infinite rank, structure-preserving GE converges at a geometric rate. Specifically, we will consider a function $\tilde{f}$ that is analytically continuable in at least one variable to a sufficiently large region of the complex plane.We characterize this region formally using the concept of a *stadium*.

[4]Note that m^+ and m^- are not related to m in (2.3).

DEFINITION 3.2 (Stadium). *The stadium S_β with radius $\beta > 0$ is the region in the complex plane consisting of all numbers lying at a distance $\leq \beta$ from an interval $[c, d]$, i.e.,*

$$S_\beta = \left\{ z \in \mathbb{C} : \inf_{x \in [c,d]} |x - z| \leq \beta \right\}.$$

To understand convergence, we will view structure-preserving GE as a coupled procedure involving the functions $\tilde{f}^+$ and $\tilde{f}^-$ defined in Section 3.3. The proof requires an examination of the error produced after applying the GE step (3.12), and we see in (3.13) that there are three cases to consider. Bounds on the error are intimately tied to the *growth factors* of the GE procedures that are applied to $\tilde{f}^+$ and $\tilde{f}^-$. The growth factors quantify the worst possible increase in the absolute maximum of the function after a rank one update. Geometric convergence can be proven if the size of the stadium in which $\tilde{f}$ is analytic is large enough to counteract the potential growth induced by GE.

The connection between the region of analyticity and the GE growth factor is made clear in the proof of Theorem 8.1 in [42], which shows that iterative GE with complete pivoting as in (3.2) converges geometrically for functions that are analytic within a sufficiently large stadium. In the first or second case of (3.13), standard GE with complete pivoting is applied to either $\tilde{f}^+$ or $\tilde{f}^-$, and we may use Theorem 8.1 directly. In the third case of (3.13), two GE procedures are performed: a GE step with complete pivoting is applied to whichever of the two functions $\tilde{f}^+$ or $\tilde{f}^-$ has a larger absolute maximum value, and a GE step with a nonstandard pivoting strategy is applied to the other function. If a bound on the growth factor of this nonstandard GE step is known, then as long as $\tilde{f}$ is assumed to be analytic in an appropriately-sized region of the complex plane, we can apply a mild generalization of Theorem 8.1. For this reason, we precede the main convergence result with the following lemma:

LEMMA 3.3. *The growth factor for the nonstandard GE procedure applied within BMC structure-preserving GE is bounded above by $1 + \alpha^{-1}$, where α is the coupling parameter in* (3.13).

Proof. Consider performing one step of BMC structure-preserving GE on $\tilde{f}$ by operating on $\tilde{f}^+$ and $\tilde{f}^-$ from (3.11), and suppose we are in the third case of (3.13). Without loss of generality, suppose that $|m^-| > |m^+|$. Then, $|m^+| = 1/\sigma_1(M)$, $|m^-| = 1/\sigma_2(M)$, and a nonstandard GE step is performed on $\tilde{f}^-$ using the pivot m^-. Here, $\sigma_k(M)$ denotes the kth singular value of M.

After the nonstandard GE step is applied, the supremum norm of the residual is

$$\|\tilde{e}_1^-\|_\infty = \left\| \tilde{f}^- - \operatorname{sgn}(m^-) \frac{\tilde{f}^-(\theta_*, \cdot)\tilde{f}^-(\cdot, \rho_*)}{\sigma_2(M)} \right\|_\infty \leq \|\tilde{f}^-\|_\infty + \frac{\|\tilde{f}^-\|_\infty^2}{\sigma_2(M)}, \tag{3.15}$$

where $(\theta_*, \rho_*) \in [-\pi, \pi] \times [0, 1]$ is the location of the pivot in the first quadrant.

Since we are in the third case of (3.13), we have $\sigma_2(M) \geq \alpha\sigma_1(M)$, and therefore $\sigma_2(M) \geq \alpha\|\tilde{f}^-\|_\infty$. Applying these results to (3.15) gives that

$$\|\tilde{e}_1^-\|_\infty \leq (1 + \alpha^{-1})\|\tilde{f}^-\|_\infty, \tag{3.16}$$

i.e., the growth factor for the nonstandard GE step cannot exceed $1 + \alpha^{-1}$. ∎

Since bounds on the growth factors are known for each GE procedure applied on $\tilde{f}^+$ and $\tilde{f}^-$, geometric convergence of BMC structure-preserving GE can be proven. An analogous theorem holds with the roles of θ and ρ exchanged.

THEOREM 3.4. *Let $\tilde{f} : [-\pi, \pi] \times [-1, 1] \to \mathbb{R}$ be a BMC function such that $\tilde{f}(\theta, \cdot)$ is continuous for any $\theta \in [-\pi, \pi]$ and $\tilde{f}(\cdot, \rho)$ is analytic and uniformly bounded in a stadium S_β of radius $\beta = \max(2, 1 + \alpha^{-1})2\pi\kappa$, $\kappa > 1$, for any $\rho \in [-1, 1]$. Then, there exists a constant $C > 0$ such that*

$$\|\tilde{f} - \tilde{f}_k\|_\infty = \|\tilde{e}_k\|_\infty \leq C\mu^{-k},$$

where $\mu = \min\{\kappa, \alpha^{-1}\}$, α is the coupling parameter described in (3.13), and $\tilde{f}_k$ is the approximant constructed after k steps of the BMC structure-preserving GE procedure.

Proof. For $k \geq 0$, $\tilde{e}_k$ is a BMC function and can be written as the sum of an even π-periodic and odd π-antiperiodic function, i.e., $\tilde{e}_k = \tilde{e}_k^+ + \tilde{e}_k^-$ (see Section 3.3). Let $\mu = \min\{\kappa, \alpha^{-1}\}$, and choose a constant $C > 0$ so that $\|\tilde{e}_0^+\|_\infty \leq C/2$ and $\|\tilde{e}_0^-\|_\infty \leq C/2$. We will show by induction that $\|e_k\|_\infty \leq C\mu^{-k}$ for all $k > 0$.

When $k = 0$, $\max\{\|\tilde{e}_0^+\|_\infty, \|\tilde{e}_0^-\|_\infty\} \leq (C/2)$. Suppose that for $k > 0$, the following induction hypothesis holds:

$$\max\{\|\tilde{e}_k^+\|_\infty, \|\tilde{e}_k^-\|_\infty\} \leq (C/2)\mu^{-k}. \tag{3.17}$$

Consider the next structure-preserving GE step. Using (3.13), there are three cases to consider.

Case 1: Here, $\|\tilde{e}_k^-\|_\infty < \alpha\|\tilde{e}_k^+\|_\infty$, and only $\tilde{e}_k^+$ is updated (see Section 3.3). This step is equivalent to performing a standard GE step with complete pivoting as in (3.2) on $\tilde{e}_k^+$. By Theorem 8.1 in [42], we have

$$\|\tilde{e}_{k+1}^+\|_\infty \leq \kappa^{-1}\|\tilde{e}_k^+\|_\infty.$$

Since $\tilde{e}_{k+1}^- = \tilde{e}_k^-$, we find that

$$\|\tilde{e}_{k+1}^-\|_\infty = \|\tilde{e}_k^-\|_\infty < \alpha\|\tilde{e}_k^+\|_\infty,$$

and using the definition of μ and (3.17), we conclude that

$$\max\{\|\tilde{e}_{k+1}^+\|_\infty, \|\tilde{e}_{k+1}^-\|_\infty\} \leq \mu^{-1}\max\{\|\tilde{e}_k^+\|_\infty, \|\tilde{e}_k^-\|_\infty\} \leq (C/2)\mu^{-(k+1)}. \tag{3.18}$$

Case 2: Here, $\|\tilde{e}_k^+\|_\infty < \alpha\|\tilde{e}_k^-\|_\infty$, and only $\tilde{e}_k^-$ is updated. This is equivalent to Case 1 with the roles of $\tilde{e}_k^+$ and $\tilde{e}_k^-$ interchanged.

Case 3: Without loss of generality, suppose that $|m^-| > |m^+|$. Then, a standard GE step with complete pivoting is applied to $\tilde{e}_k^+$, and a GE step with nonstandard pivoting is performed on $\tilde{e}_k^-$. As in Case 1, we find that

$$\|\tilde{e}_{k+1}^+\|_\infty \leq \kappa^{-1}\|\tilde{e}_k^+\|_\infty.$$

For $\tilde{e}_{k+1}^-$ we use the bound on the growth factor from Lemma 3.3 to apply a slight generalization of Theorem 8.1 in [42], finding that

$$\|\tilde{e}_{k+1}^-\|_\infty \leq \kappa^{-1}\|\tilde{e}_k^-\|_\infty.$$

It follows from the definition of μ and (3.17) that

$$\max\{\|\tilde{e}_{k+1}^+\|_\infty, \|\tilde{e}_{k+1}^-\|_\infty\} \leq (C/2)\mu^{-(k+1)}.$$

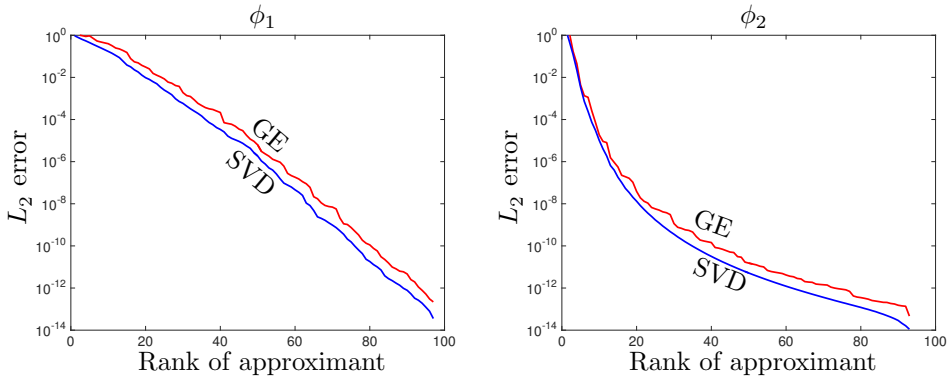

FIG. 5. *A comparison of low rank approximations to the functions in* (3.20) *computed using the SVD and the iterative GE procedure. The L_2 error is plotted against the rank of the approximants to ϕ_1 and ϕ_2. The L_2 error given by the SVD approximants are optimal and we observe that that the low rank approximants constructed by the GE procedure are near-optimal.*

By induction, we have that

$$\max\{\|\tilde{e}_k^+\|_\infty, \|\tilde{e}_k^-\|_\infty\} \leq (C/2)\mu^{-k}, \qquad k \geq 0,$$

and the result follows from the fact that $\|\tilde{e}_k\|_\infty \leq \|\tilde{e}_k^+\|_\infty + \|\tilde{e}_k^-\|_\infty$. ∎

The assumptions required on $\tilde{f}$ in Theorem 3.4 are rather restrictive, as the proof of convergence requires us to consider GE growth rates that account for the worst-case scenario. Empirically, we observe convergence for a much broader class of functions, and at rates that are asymptotically optimal. This is described in the next section.

3.5. Near-optimality. While Section 3.4 proves that convergence of the GE procedure in Figure 3 is geometric when f is analytic in a sufficiently large region of the complex plane, we observe in practice that the procedure converges at near-optimal rates for functions that are only a few times differentiable.

If $\tilde{f}$ is Lipschitz continuous with respect to both variables for $(\theta, \rho) \in [-\pi, \pi] \times [-1, 1]$, then the best rank K approximation to $\tilde{f}$ is given by the Karhunen-Loève expansion, also called the the singular value decomposition (SVD), of $\tilde{f}$:

$$\tilde{f}(\theta, \rho) = \sum_{j=1}^{\infty} \sigma_j u_j(\rho) v_j(\theta), \qquad (\theta, \rho) \in [-\pi, \pi] \times [-1, 1]. \tag{3.19}$$

The non-increasing sequence $\sigma_1 \geq \sigma_2 \geq \cdots$ of real, nonnegative numbers are the *singular values* of $\tilde{f}$. The continuous *singular functions* $\{u_j(\rho)\}$ and $\{v_j(\theta)\}$ each form an orthonormal set of functions with respect to the standard L_2 inner product. A best rank K approximation to $\tilde{f}$, in the sense of the L_2 norm, is constructed by truncating (3.19) after K terms [34].

For reasons closely related to those discussed in Section 3.3, the SVD preserves the BMC structure of $\tilde{f}$ [49]. Unfortunately, the high cost of computing the SVD makes this an untenable approach for constructing low rank approximants to $\tilde{f}$ in practice. Nonetheless, approximants constructed via the SVD are optimal with respect to $\|\cdot\|_2$, and this provides a way to check the quality of the low rank approximants constructed by our GE procedure. Figure 5 displays the L_2 error over $[-\pi, \pi] \times [-1, 1]$ for rank K approximations constructed via the SVD and the GE procedure for the following

two BMC-II functions:

$$\begin{aligned}\phi_1(\theta,\rho) &= \exp\left[-(\cos(11\rho\sin\theta)+\sin(\rho\cos\theta))^2\right],\\ \phi_2(\theta,\rho) &= (1-\omega)_+^6\left(35(\omega)^2+18\omega+3\right),\end{aligned} \tag{3.20}$$

where $\omega(\theta,\rho) = \left((\rho\cos\theta-.2)^2+(\rho\sin\theta-.2)^2\right)^{1/2}$ and $\zeta_+ = \max\{\zeta,0\}$. The error given by the SVD behaves in accordance with known theoretical results, decaying geometrically for the function ϕ_1 and at an algebraic rate for ϕ_2 [38]. In experiments, it is observed that our GE procedure constructs near-best low rank approximants to smooth BMC functions.

4. Algorithms for numerical computation with functions on the disk. In this section, we describe several of the algorithms used in the Diskfun software. These methods rely on the fact that every smooth function f on the disk is associated with a BMC-II function $\tilde{f}$ that is periodic in θ. We compute with a low rank approximation to $\tilde{f}$ as in (3.3), which is constructed by the GE procedure in Figure 3. We rely on the fact that in (3.3), each $c_j(\rho)$ and $r_j(\theta)$ can be approximated by a Chebyshev and Fourier series, respectively, so that for $1 \le j \le K$,

$$c_j(\rho) \approx \sum_{\ell=0}^{m-1} a_\ell^j\, T_\ell(\rho), \qquad r_j(\theta) \approx \sum_{k=-n/2}^{n/2-1} b_k^j\, e^{ik\theta}, \tag{4.1}$$

where $T_\ell(\rho)$ is the Chebyshev polynomial of degree ℓ, and n is an even integer.

The algorithms for computing with functions represented in Chebyshev and Fourier bases differ considerably from one another. However, implementation in the Chebfun environment is significantly simplified due to its underlying object-oriented class structure. For example, Chebfun overloads commands such as `sum(g)` (integration) or `diff(g)` (differentiation), so that the same syntax executes different underlying algorithms based on whether the object `g` is represented by a Chebyshev series or a Fourier series [50].

4.1. Pointwise evaluation. To efficiently evaluate $\tilde{f}$ at a fixed point (θ_*,ρ_*), we use (3.3), observing that

$$\tilde{f}(\theta_*,\rho_*) \approx \sum_{j=1}^{K} d_j c_j(\rho_*) r_j(\theta_*). \tag{4.2}$$

Evaluation of $\tilde{f}$ proceeds as $2K$ 1D function evaluations. Functions $c_j(\rho)$, $1 \le j \le K$, are evaluated using Clenshaw's algorithm [46, Ch. 19], and functions $r_j(\theta)$, $1 \le j \le K$, are evaluated using Horner's scheme [50]. Altogether, this requires $\mathcal{O}(K(m+n))$ operations. The algorithm is implemented in the `feval` command.

4.2. Computation of Chebyshev–Fourier coefficients. The low rank form of $\tilde{f}$ facilitates the use of fast transform methods based on the FFT. We can write the truncated tensor product Chebyshev–Fourier expansion of $\tilde{f}$ as follows:

$$\tilde{f}(\theta,\rho) \approx \sum_{k=-n/2}^{n/2-1} \sum_{\ell=0}^{m-1} X_{\ell k} T_\ell(\rho) e^{ik\theta}, \tag{4.3}$$

where X is a matrix whose entries are the $2D$ Chebyshev–Fourier coefficients of $\tilde{f}$. Using the low rank form of $\tilde{f}$ given by (3.3), the matrix X can also be expressed in low rank form as $X = ADB^T$. Here, A is an $m \times K$ matrix whose jth column contains the coefficients $\{a_\ell^j\}$ from (4.1), D is a K-by-K diagonal matrix consisting of the pivot values $\{d_j\}$, and B is an $n \times K$ matrix whose jth column contains the coefficients $\{b_k^j\}$ from (4.1). Given a sample of $\tilde{f}$ on an $m \times n$ Chebyshev–Fourier grid, the direct computation of the Chebyshev–Fourier coefficients of $\tilde{f}$ costs $\mathcal{O}(mn\log(mn))$ operations. However, using the GE procedure in Section 3.2, the low rank form of X can be found in only $\mathcal{O}(K^3 + K^2(m+n) + K(m\log m + n\log n))$ operations. This is because once the GE process adaptively selects the skeleton representing $\tilde{f}$ at a cost of $\mathcal{O}(K^3 + K^2(m+n))$, the coefficients in (4.1) for every $c_j(\rho)$ and $r_j(\theta)$ in (3.3) can be found in only $\mathcal{O}(K(m\log m + n\log n))$ operations.

Several procedures, such as integration and differentiation, can be executed using the low rank factorization of X. Using the command `coeffs2` in Diskfun, X can be explicitly computed with an additional $\mathcal{O}(Kmn)$ operations.

The above operation retrieves coefficients when supplied with a sample of $\tilde{f}$, and the inverse of this operation provides an efficient way to sample $\tilde{f}$ on a $m \times n$ Chebyshev–Fourier grid. Given X in low rank form, this proceeds in $\mathcal{O}(K(m\log m + n\log n))$ operations; the algorithm is implemented in the `sample` command.

4.3. Integration. To integrate $\tilde{f}(\theta,\rho)$ over the unit disk, we again take advantage of the low rank form of (3.3), transforming the double integral into sums of 1D integrals:

$$\int_{-\pi}^{\pi}\int_0^1 \tilde{f}(\theta,\rho)\rho\,d\rho\,d\theta \approx \sum_{j=1}^{K} d_j \int_{-\pi}^{\pi} r_j(\theta)\,d\theta \int_0^1 c_j(\rho)\rho\,d\rho. \tag{4.4}$$

For integration of the periodic $r_j(\theta)$ functions, the trapezoidal rule is used. To evaluate $\int_0^1 c_j(\rho)\rho\,d\rho$, the coefficients for $\rho c_j(\rho)$ are computed, and then Clenshaw-Curtis quadrature is applied [46, Ch. 19]. These $2K$ 1D integrals can be computed in a total of $\mathcal{O}(Km)$ operations. This can be further reduced using (3.11) since only the even, π-periodic terms will contribute to the value of the integral.

Integration is implemented in the `sum2` command. For example, the integral of $f(x,y) = -x^2 - 3xy - (y-1)^2$ over the unit disk is $-3\pi/2$, and can be computed in Diskfun as

```
f = diskfun(@(x,y) -x.^2-3*x.*y -(y-1).^2);
sum2(f)
ans =
   -4.712388980384692
```

The error is determined with `abs(sum2(f)+3*pi/2)`, which gives 1.7764×10^{-15}.

4.4. Differentiation. When considering derivatives on the disk, note that partial differentiation with respect to ρ can lead to artificial singularities at $\rho = 0$. For example, if $f(\theta,\rho) = \rho^2$, then $\partial f/\partial\rho = 2\rho$, which is not smooth on the disk. In contrast, for a smooth function $\tilde{f}$, partial derivatives with respect to x and y will always be well-defined. For this reason, and because of the usefulness of these operators in vector calculus (see Section 4.6), we consider efficient and stable ways to calculate $\partial\tilde{f}/\partial x$ and $\partial\tilde{f}/\partial y$.

By (2.1), $\rho = \sqrt{x^2+y^2}$, and $\theta = \tan^{-1}(y/x)$, so the chain rule can be applied to

obtain

$$\frac{\partial \tilde{f}}{\partial x} = \cos\theta \frac{\partial \tilde{f}}{\partial \rho} - \frac{1}{\rho}\sin\theta \frac{\partial \tilde{f}}{\partial \theta}, \tag{4.5}$$

$$\frac{\partial \tilde{f}}{\partial y} = \sin\theta \frac{\partial \tilde{f}}{\partial \rho} + \frac{1}{\rho}\cos\theta \frac{\partial \tilde{f}}{\partial \theta}. \tag{4.6}$$

Exploiting the low rank form given in (3.3), (4.5) can be written as

$$\frac{\partial \tilde{f}}{\partial x} \approx \sum_{j=1}^{K} d_j \left(\frac{\partial c_j(\rho)}{\partial \rho} \right) \Big(\cos\theta \, r_j(\theta) \Big) - \sum_{j=1}^{K} d_j \left(\frac{c_j(\rho)}{\rho} \right) \Big(\sin\theta \, \frac{\partial r_j(\theta)}{\partial \theta} \Big). \tag{4.7}$$

A similar expression can be used for (4.6).

Here we make an important observation. The above result establishes that approximants on the disk are continuously differentiable at $\rho = 0$ only if $\sum_{j=1}^{K} c_j(\rho)$ is divisible by ρ. Suppose $\tilde{f}$ is nonzero at $\rho = 0$ and write the approximant in the form given by (3.14). Then, because of (3.10), for $2 \le j \le K^+$, each term $d_j^+ c_j^+(\rho) r_j^+(\theta)$ is zero at $\rho = 0$. Since $c_j^+(\rho)$ is an even Chebyshev polynomial, it must be of the form $\alpha_1\rho^2 + \alpha_2\rho^4 + \cdots + \alpha_q\rho^{2q}$, where $q \le \lfloor (m-1)/2 \rfloor$. This implies that these functions are all divisible by ρ. For $j = 1$, $r_1^+(\theta)$ is constant by (3.10), and so all terms in (4.7) involving derivatives of $r_1^+(\theta)$ with respect to θ vanish. Since every $c_j^-(\rho)$ function for $1 \le j \le K^-$ is an odd function, these are also always divisible by ρ. This means that the approximants constructed by the BMC-II structure preserving GE procedure have inherited properties ensuring that they are continuously differentiable at $\rho = 0$.

There are $2K$ 1D derivatives to compute in (4.7). Using (4.1),

$$\sin\theta \, \frac{\partial r_j(\theta)}{\partial \theta} = \sum_{k=-n/2}^{n/2-1} \frac{-(k+1)b_{k+1}^j + (k-1)b_{k-1}^j}{2} e^{ik\theta}, \tag{4.8}$$

$$\cos\theta \, r_j(\theta) = \sum_{k=-n/2}^{n/2-1} \frac{b_{k+1}^j + b_{k-1}^j}{2} e^{ik\theta}, \tag{4.9}$$

where $b_{-n/2-1}$ and $b_{n/2}$ are set to zero. Expanding each $c_j(\rho)$ as in (4.1), the recursion formula in [27, p. 34] gives the coefficients for $\partial c_j(\rho)/\partial\rho$ in $\mathcal{O}(m)$ operations. To determine $c_j(\rho)/\rho$, we construct the operator B_ρ, which represents multiplication by the function $g(\rho) = \rho$ in the Chebyshev basis. Then,

$$\frac{c_j(\rho)}{\rho} = \sum_{\ell=0}^{m-1} (B_\rho^{-1} \underline{a}^j)_\ell T_\ell(\rho), \qquad B_\rho = \begin{pmatrix} 0 & \frac{1}{2} & & & & \\ 1 & 0 & \frac{1}{2} & & & \\ & \frac{1}{2} & \ddots & \ddots & & \\ & & \ddots & \ddots & \frac{1}{2} & \\ & & & \frac{1}{2} & 0 & \frac{1}{2} \\ & & & & \frac{1}{2} & 0 \end{pmatrix}, \tag{4.10}$$

where $\underline{a}^j = (a_0^j, \ldots, a_{m-1}^j)^T$. Here, B_ρ^{-1} exists because we choose B_ρ to be of size $m \times m$, where m is an even integer. Working directly with the coefficients via (4.10) is an efficient way to bypass the artificial singularity introduced in (4.7), without

explicitly avoiding computation at $\rho = 0$. In contrast, the standard procedure when working on function values with the DFS method uses a "shifted grid" strategy [12,18].

Differentiation is accessed through the `diff` command in Diskfun, and requires $\mathcal{O}(K(m+n))$ operations.

4.5. The L_2 norm and the weighted singular value decomposition. In Diskfun, `norm(f)` is overloaded to compute the L_2 norm on the disk, which is the continuous analogue of the matrix Frobenius norm [42]. This is one of the very few instances in Diskfun where it makes more sense to work with f directly, rather than $\tilde{f}$. The L_2 norm of a function f on the disk is given in polar coordinates as

$$\|f\|_2^2 = \int_{-\pi}^{\pi} \int_0^1 |f(\theta,\rho)|^2 \rho \, d\rho \, d\theta. \tag{4.11}$$

Computing $\|f\|_2$ using (4.11) directly is numerically unstable, especially when f is near zero. A more stable formulation is given in [34]: If f is L_2 integrable, then

$$\|f\|_2^2 = \sum_{j=1}^{\infty} \sigma_j^2, \tag{4.12}$$

where $\sigma_1 \geq \sigma_2 \geq \cdots \geq 0$ are real and nonnegative numbers referred to as the *(weighted) singular values* of f. For this reason, we are interested in the weighted SVD of f, which is given by

$$f(\theta,\rho) = \sum_{j=1}^{\infty} \sigma_j u_j(\rho) v_j(\theta), \qquad (\theta,\rho) \in [-\pi,\pi] \times [0,1]. \tag{4.13}$$

The singular functions $\{u_j(\rho)\}$, $\rho \in [0,1]$, and $\{v_j(\theta)\}$, $\theta \in [-\pi,\pi]$, are orthonormal under the following inner products, respectively:

$$< u, s >_\rho = \int_0^1 u(\rho)\overline{s(\rho)}\rho \, d\rho, \qquad < v, w >= \int_{-\pi}^{\pi} v(\theta)\overline{w(\theta)} \, d\theta, \tag{4.14}$$

where the bars on s and w denote complex conjugation.

The weighted SVD for a function on the disk is determined by applying a generalization of QR factorization to quasimatrices. Restricting the low rank approximation to $\tilde{f}$ given by (3.3) to $(\theta,\rho) \in [-\pi,\pi] \times [0,1]$, we form a $[0,1] \times K$ quasimatrix C such that the jth column of C is $c_j(\rho)$ in (3.3) restricted to the domain $[0,1]$. Similarly, we form the $[-\pi,\pi] \times K$ quasimatrix R such that the jth column of R is $r_j(\theta)$. A QR quasimatrix factorization with respect to the standard L_2 inner product on $[-\pi,\pi] \times [0,1]$ is given in [45] and selects the Legendre polynomials to orthogonalize against, and this procedure is applied to R. In consideration of (4.14), C is orthogonalized against the functions

$$\frac{\sqrt{2}}{J_1(\omega_k)} J_0(\omega_k \rho), \qquad k = 1, 2, \ldots,$$

where J_ν is the Bessel function of order ν, and ω_k is the kth positive root of $J_0(\rho)$. This finds $\{u_j(\rho)\}$, which are orthonormal with respect to (4.14). Once the QR factorizations for C and R are known, the SVD is determined through standard techniques, as discussed in [42].

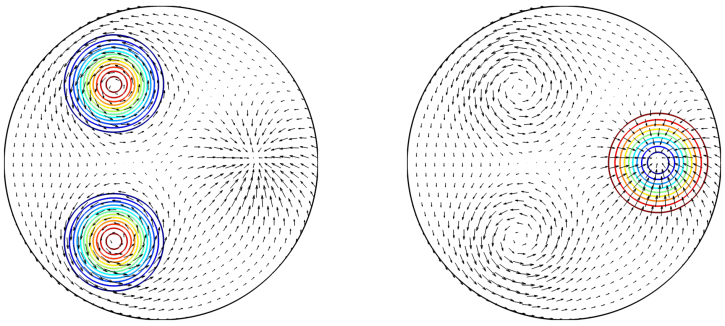

FIG. 6. *The vector function* $\mathbf{u} = \nabla \times \psi + \nabla\phi$*, with* ψ *and* ϕ *defined in* (4.15)*, together with its curl,* $\nabla \times \mathbf{u}$ *(left), and divergence,* $\nabla \cdot \mathbf{u}$ *(right). The field was plotted using* `quiver(u)`*, while the curl and divergence were computed using* `curl(u)` *and* `div(u)`*, respectively, and plotted using the* `contour` *command.*

In addition to providing a mathematically stable way to compute (4.11), the weighted SVD gives the best rank K approximation to f with respect to the L_2 inner product on the disk. Unfortunately, the use of the weighted SVD as a low-rank approximation method is limited because the rank 1 terms in (4.13) may be discontinuous at the origin of the disk [49], and consequently, the truncation of (4.13) may not be smooth. The SVD is accessed in Diskfun through the `svd` command, and is used internally in the `norm` command.

4.6. Vector-valued functions and vector calculus on the disk. Vector-valued functions can also be constructed in Diskfun. These functions are represented with respect to the Cartesian coordinate basis vectors $\hat{\mathbf{i}}$ and $\hat{\mathbf{j}}$, since not all smooth vector fields defined over the disk have smooth components when represented with respect to the polar coordinate basis vectors, $\hat{\mathbf{r}}$ and $\hat{\boldsymbol{\theta}}$. For example, the vector field given by $\mathbf{f} = 0\hat{\mathbf{i}} + \hat{\mathbf{j}}$ is expressed as $\mathbf{f} = \sin\theta\hat{\mathbf{r}} + \cos\theta\hat{\boldsymbol{\theta}}$ in polar coordinates, and both of these components are discontinuous at the origin of the disk.

Vector-valued functions are accessed in Diskfun through the creation of diskfunv objects. A diskfunv consists of two diskfun objects, one for each component of the vector-valued function. Algorithms involving diskfunv objects are implemented for algebraic actions, such as addition, as well as vector-based operations, such as the dot/cross products, and divergence. Commands that map scalar-valued functions to vector-valued functions and vice-versa, such as `grad(f)` and `curl(f)`, are also included. In the latter case, the standard interpretations are used, i.e., $\nabla \times f = [f_y, -f_x]$ for a scalar function f, and $\nabla \times \mathbf{u} = v_x - u_y$ when $\mathbf{u} = [v, u]$ is a vector-valued function. As an example, consider the potential functions given by

$$\begin{aligned} \psi(x,y) &= 10e^{-10(x+.3)^2-10(y+.5)^2} + 10e^{-10(x+.3)^2-10(y-.5)^2} + 15(1-x^2-y^2), \\ \phi(x,y) &= 10e^{-10(x-.6)^2-40y^2}, \end{aligned} \tag{4.15}$$

and the vector field $\mathbf{u} = \nabla\times\psi+\nabla\phi$. This field consists of the sum of a divergence-free term, $\nabla \times \psi$, and a curl-free term, $\nabla\phi$. Once ψ and ϕ are constructed as diskfun objects, $\mathbf{u}$ can be constructed with a single line of code: `u = curl(psi)+grad(phi)`. Figure 6 displays a plot of $\mathbf{u}$ together with its curl and divergence.

4.7. Miscellaneous operations. Diskfun is included as an object class in Chebfun, and so has access to many of the operations in Chebfun. Operations that do not

strictly require symmetry properties related to the geometry of the disk are computed using Chebfun2 with functions defined in polar coordinates [41]. This includes optimization routines, such as `min2`, `max2`, and `roots`, as well as procedures inspired by matrices such as `trace` and `lu`. Operations that use Chebfun2 are performed automatically, without requiring adjustments or intervention by the user.

5. A fast Poisson solver for computing solutions in low rank form. In [49] and [37], optimal complexity solvers for Poisson's equation on the disk are formulated through the use of parity properties associated with the Chebyshev–Fourier coefficients of BMC-II functions. Unfortunately, these solvers cannot capitalize on the low rank structure of the approximants in (3.3), and they do not guarantee that the computed solution has good compression properties. Finding a low rank representation of the solution requires additional work, and such representations are essential in Diskfun. This has motivated the development of a fast Poisson solver that directly computes low rank approximations to solutions.

Our method uses the factored alternating direction implicit (ADI) method [4, 23] to work independently on the Chebyshev and Fourier coefficients in (4.1). We combine ADI with the Fourier and ultraspherical spectral methods, so that every linear system we solve is sparse and spectral accuracy is guaranteed [30]. We find that the ADI-based method efficiently constructs low rank solutions whenever the numerical rank of the forcing function is sufficiently low.

Given a function $f(\theta,\rho)$ on the unit disk, we seek the solution $u(\theta,\rho)$ to Poisson's equation, $\nabla^2 u = f$, where $(\theta,\rho) \in [-\pi,\pi]\times[0,1]$. To ensure a unique solution, Dirichlet conditions are prescribed as $u(\theta,1) = g(\theta)$, where g is a 2π-periodic function. In this section, we will assume that $g(\theta) = 0$.[5]

To enforce that the numerical solution u is continuous over $u(\theta,0)$, we apply the disk analogue to the DFS method and consider solving the related equation $\nabla^2\tilde{u} = \tilde{f}$, where $\tilde{f}$ is the BMC-II extension of f given by (2.4). The equation $\nabla^2\tilde{u} = \tilde{f}$ is expressed in polar coordinates as

$$\rho^2\frac{\partial^2\tilde{u}}{\partial\rho^2} + \rho\frac{\partial\tilde{u}}{\partial\rho} + \frac{\partial^2\tilde{u}}{\partial\theta^2} = \rho^2\tilde{f}, \qquad (\theta,\rho)\in[-\pi,\pi]\times[-1,1], \tag{5.1}$$

where the standard formulation is multiplied by ρ^2 so that the variable coefficients are low degree polynomials in ρ. It is straightforward to show that $\tilde{u}$ must also possess BMC-II symmetry and therefore corresponds to a differentiable function on the disk. Restricting $\tilde{u}$ to $[-\pi,\pi]\times[0,1]$ gives u.

To ensure that $\tilde{u}$ satisfies homogeneous boundary conditions, we will express it as a product of $1-\rho^2$ and an unknown function $\hat{u}$. Expanding $\hat{u}$ in the Chebyshev–Fourier basis, we find that

$$\tilde{u}(\theta,\rho) \approx (1-\rho^2)\hat{u}(\theta,\rho) = (1-\rho^2)\sum_{k=-n/2}^{n/2-1}\sum_{\ell=0}^{m-1} Y_{\ell k}T_\ell(\rho)e^{ik\theta}, \tag{5.2}$$

where n is an even integer.

We seek a low rank approximation to the Chebyshev–Fourier coefficient matrix $Y\in\mathbb{C}^{m\times n}$. Since $1-\rho^2 = (T_0(\rho)-T_2(\rho))/2$, we can represent multiplication by $1-\rho^2$

[5] Whenever $g(\theta)$ is nonzero, the system can be solved by relating it to a system with homogeneous boundary conditions (see [6, Ch. 6]).

in the Chebyshev basis with a sparse operator M. Then, MY is the Chebyshev–Fourier coefficient matrix of $\tilde{u}$, i.e., $MY = X$ in (4.3).

To use ADI, the discretization of (5.1) must be expressed as a Sylvester matrix equation of the form $AY - YB = C$, with the matrices $A \in \mathbb{C}^{m\times m}$ and $B \in \mathbb{C}^{n\times n}$ represented in a data-sparse way. Plugging (5.2) into (5.1) and applying the chain rule, we rewrite (5.1) with respect to $\hat{u}$:

$$\underbrace{\rho^2(1-\rho^2)\frac{\partial^2\hat{u}}{\partial\rho^2} + (-5\rho^3+\rho)\frac{\partial\hat{u}}{\partial\rho} - 4\rho^2\hat{u}}_{=\mathcal{L}} + (1-\rho^2)\frac{\partial^2\hat{u}}{\partial\theta^2} = \rho^2\tilde{f}. \tag{5.3}$$

We now seek a discrete counterpart to the operator $\mathcal{L}$ that acts on the Chebyshev coefficients of $\hat{u}$. To formulate such an operator, we apply a variant of the ultraspherical spectral method [30]. This method uses recurrence relations between the Chebyshev and other ultraspherical polynomials to define sparse differential operators. Applying the ultraspherical spectral method directly results in a discretization of $\mathcal{L}$ that is sparse and banded. However, the bandwidth of this operator can be further reduced if we use a recurrence relation between the Chebyshev polynomials of the first and second kind that involves the term $1-\rho^2$. Using [29, (18.9.10)], we have that

$$(1-\rho^2)\frac{d^2}{d\rho^2}T_\ell(\rho) = -\ell(\ell+1)\left(\frac{1}{2}U_\ell(\rho) - \frac{1}{2}U_{\ell-2}(\rho)\right) + \ell U_\ell(\rho), \qquad \ell \geq 2, \tag{5.4}$$

where $\{U_\ell\}$ are the Chebyshev polynomials of the first kind. We use (5.4) to define a discrete operator $D_{2(1)}$ that represents $(1-\rho^2)\partial^2/\partial\rho^2$. Like all differentiation operators in the the ultraspherical spectral method, $D_{2(1)}$ acts on coefficients in one basis and converts them to another. Specifically, it acts on Chebyshev coefficients and returns coefficients in the $\{U_\ell\}$ basis. The remaining terms in $\mathcal{L}$ are expressed using standard techniques in the ultraspherical spectral method, and the resulting discretization of $\mathcal{L}$, denoted as L, is a banded matrix of bandwidth 4.

We will use L and the differentiation matrix

$$D_F^2 = \operatorname{diag}\left(-(\tfrac{n}{2})^2, -(\tfrac{n-1}{2})^2, \cdots, 0, -1, -4, \cdots, -(\tfrac{n-1}{2})^2\right),$$

which discretizes $\partial^2/\partial\theta^2$ and acts on Fourier coefficients, to write the discretization of (5.3) as a generalized Sylvester equation:

$$LY + S_1 M Y D_F^2 = S_1 M_{\rho^2} F. \tag{5.5}$$

Recall that the matrix M is an operator representing multiplication by $1-\rho^2$.[6] The tridiagonal matrix S_1 converts coefficients in the Chebyshev basis to the $\{U_\ell\}$ basis; this is required due to the action of L (see [30]). On the right-hand side, F is the Chebyshev–Fourier matrix of coefficients for $\tilde{f}$, and M_{ρ^2} is a tridiagonal matrix representing multiplication by ρ^2.

To apply ADI, we must write (5.5) in the following form:

$$\underbrace{(S_1 M)^{-1} L}_{=A} Y - Y \underbrace{(-D_{F^2}^2)}_{=B} = \underbrace{M^{-1} M_{\rho^2} F}_{=C} \tag{5.6}$$

[6] Note that in the first term, multiplication by $1-\rho^2$ occurs implicitly via (5.3).

The matrices L and S_1M are each banded with a bandwidth of 4, and B is diagonal. We solve (5.5) by applying the factored ADI method in [4]. This method never requires F to be formed explicitly. Rather, it operates directly on the low rank factorization of F described in Section 4.2. The solution is returned as a low rank factorization, $Y = ZDG^*$, where Z is a collection of Chebyshev coefficients, D is diagonal, and G is a collection of Fourier coefficients.

ADI is an iterative method, and the convergence of the method is sensitive to the selection of a set of shift parameters [24, 33]. The spectrum of A in (5.6), denoted as $\sigma(A)$, can be contained in an interval on the real line that is well-separated from the interval containing $\sigma(B)$. In such a scenario, near-optimal shift parameters are known and can efficiently be computed [33].

The computational cost of ADI is dependent on the numerical rank of the matrix C and properties of the matrices A and B. If A and B were normal, one could directly apply bounds given in [3, 24, 33] to find the maximum number of ADI iterations required for approximating Y to within the tolerance ε.[7] However, A is not a normal matrix. Fortunately, the matrix V in the eigendecomposition $A = V\Lambda V^{-1}$ is well-conditioned, with $\kappa_2(V) = \|V\|_2\|V^{-1}\|_2$ growing approximately quadratically with m. We apply the bound for normal matrices given in [3] to the eigendecomposition of A and find that we require at most N steps of ADI, where $N = \left\lceil \pi^{-2} \log(4\kappa_2(V)/\varepsilon) \log(4\gamma) \right\rceil$. Here, γ, described in Corollary 4.2 of [3], is a function of $\sigma(A)$ and $\sigma(B)$. Empirically, we observe that γ grows approximately quadratically as $(m+n)$ increases. Each iteration of ADI requires $2K$ sparse, linear solves, so the total cost for performing factorized ADI on (5.6) is $\mathcal{O}(NK(m+n))$.

The ADI method results in an overestimation of the numerical rank of Y. This is remedied by applying a compression step on the factorization $Y = ZDG^*$ via the SVD, at a computational cost of $\mathcal{O}((NK)^2(m+n) + (NK)^3)$. Accounting for the logarithmic growth of N, the overall cost of our procedure is $\mathcal{O}(K^2(n+m)(\log(n)\log(m))^2 + K^3(\log(n)\log(m))^3)$.

In contrast, optimal complexity methods that ignore the numerical rank of $\tilde{f}$ find a low rank approximation to Y in $\mathcal{O}(mn \log mn + \tilde{K}^3 + \tilde{K}^2(m+n))$, where $\tilde{K}$ is the numerical rank of Y. This is because one can decouple (5.6) and find the coefficient matrix Y in $\mathcal{O}(mn)$ operations. A low rank approximation to Y can then be constructed by retrieving the function values associated with Y via the FFT, and then performing BMC structure-preserving GE.

The ADI-based method is beneficial when the numerical rank of Y is sufficiently small, and in practice, we use the alternative solver described in [49] whenever ADI is not advantageous. Figure 7 (left) compares the rate at which these two methods construct a low rank approximation, represented as a diskfun object, to the solution of (5.1). For choices of $\tilde{f}$ with various numerical ranks, we plot the wall clock time in seconds against increasingly large values of n, with $m = 2n + 1$. The alternative solver, which is insensitive to the rank of $\tilde{f}$, is represented in black. The ADI-based method proves effective for moderate-sized problems ($n = 1048$) when the rank of $\tilde{f}$ is below 10, performing up to 5 times faster than the alternative method. With $n = 10{,}000$ and $\tilde{f}$ of numerical rank 5, the ADI solver constructs a low rank solution

[7] Bounds are also supplied in [3] and [33] for the case of non-normal A and B through the use of pseudospectra and fields of values, respectively. In our case, the matrix V in the eigendecomposition $A = V\Lambda V^{-1}$ is well-conditioned, and we therefore only require a slight generalization on the bounds supplied for normal operators.

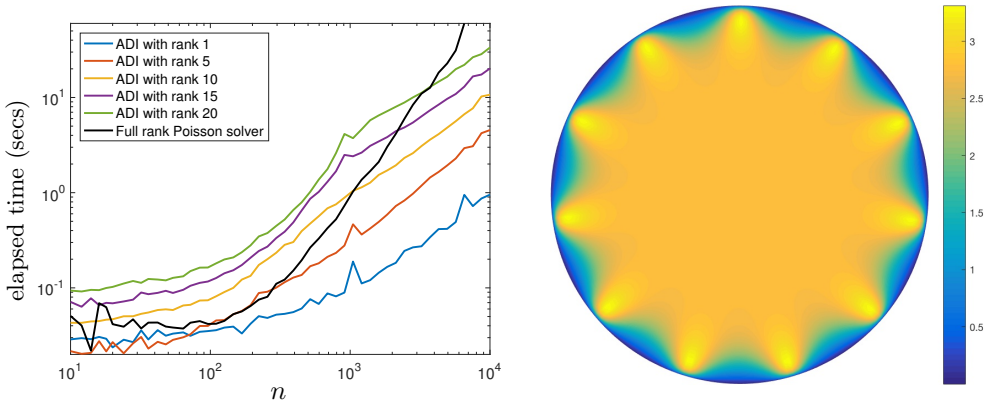

FIG. 7. *Left: Comparison of the execution (wall clock) time for the ADI-based Poisson solver and an optimal complexity solver that does not account for low rank structures (black), as a function of n, where the problem size is $(2n+1) \times n$. Timings include the construction of a diskfun object. Right: Solution to $\nabla^2 u = f$ with boundary condition $u(\theta, 1) = 0$, where f is given in* (5.7).

in under 5 seconds.[8]

Our solver is implemented in Diskfun in an integrated way: The output returned is automatically represented as a diskfun object, and can therefore immediately be visualized or operated on using Diskfun commands. For example, Figure 7 (right) displays the solution to $\nabla^2 u = f$ computed with the `poisson` command in Diskfun. Here, f is numerically a rank 16 function, given by

$$f(\theta, \rho) = e^{-40(\rho^2-1)^4} \sinh\left(5 - 5\rho^{11}\cos(11\theta - 11/\sqrt{2})\right), \tag{5.7}$$

and the boundary condition is $u(\theta, 1) = 0$.

6. Conclusions. The analogue of the double Fourier sphere (DFS) method for functions on the unit disk provides a useful structure that is retained through a new iterative Gaussian elimination procedure on functions. We use this concept to construct low rank approximations to functions on the disk that facilitate fast and stable computations based on the FFT. Fast and spectrally accurate algorithms exploiting low rank structures are described for several operations, including differentiation, integration, vector calculus, and the solving of Poisson's equation. We have implemented these ideas in Diskfun, which is part of the publicly available, open-source software Chebfun. This allows investigators to compute with functions in polar geometries in an intuitive, accurate, and highly efficient way, without concern for the underlying discretization procedure.

Acknowledgments. We are grateful to Nick Trefethen for his detailed comments on a draft of the paper. We thank Nick Hale and Stefan Güttel for observations concerning the computation of the weighted SVD in Section 4.5, and Jared Aurentz for valuable feedback. We thank Behnam Hashemi and the Chebfun team for reviewing the Diskfun software. We thank the editor and referees for their valuable comments, and are particularly appreciative of an anonymous reviewer of [43], whose comments motivated the development of our fast disk Poisson solver.

[8] Timings were performed in MATLAB R2016a on a 2015 Macbook Pro with no explicit parallelization. The degrees of freedom used in this experiment were increased artificially to demonstrate asymptotic complexity.

REFERENCES

[1] P. Amore, *Solving the Helmholtz equation for membranes of arbitrary shape: numerical results*, J. of Phys. A: Mathematical and Theoretical, 41 (2008), pp. 265–206.

[2] M. Bebendorf, *Approximation of boundary element matrices*, Numer. Math., 86 (2000), pp. 565–589.

[3] B. Beckermann and A. Townsend, *On the singular values of matrices with displacement structure*, arXiv preprint arXiv:1609.09494, (2016).

[4] P. Benner, R.-C. Li, and N. Truhar, *On the ADI method for Sylvester equations*, J. Comput. Appl. Math., 233 (2009), pp. 1035–1045.

[5] A. Bhatia and E. Wolf, *On the circle polynomials of Zernike and related orthogonal sets*, in Mathematical Proceedings of the Cambridge Philosophical Society, vol. 50, Cambridge Univ Press, 1954, pp. 40–48.

[6] J. P. Boyd, *Chebyshev and Fourier Spectral Methods*, Courier Corporation, 2001.

[7] J. P. Boyd and F. Yu, *Comparing seven spectral methods for interpolation and for solving the Poisson equation in a disk: Zernike polynomials, Logan–Shepp ridge polynomials, Chebyshev–Fourier series, cylindrical Robert functions, Bessel–Fourier expansions, square-to-disk conformal mapping and radial basis functions*, J. Comput. Phys., 230 (2011), pp. 1408–1438.

[8] O. A. Carvajal, F. W. Chapman, and K. O. Geddes, *Hybrid symbolic-numeric integration in multiple dimensions via tensor-product series*, in Proceedings of the 2005 international symposium on symbolic and algebraic computation, ACM, 2005, pp. 84–91.

[9] R. Churchill, *Fourier Series and Boundary Value Problems*, McGraw-Hill book Company, Incorporated, 1941.

[10] T. A. Driscoll, N. Hale, and L. N. Trefethen, eds., *Chebfun Guide*, Pafnuty Publications, Oxford, 2014.

[11] H. Eisen, W. Heinrichs, and K. Witsch, *Spectral collocation methods and polar coordinate singularities*, J. Comput. Phys., 96 (1991), pp. 241–257.

[12] B. Fornberg, *A pseudospectral approach for polar and spherical geometries*, SIAM J. Sci. Comp., 16 (1995), pp. 1071–1081.

[13] B. Fornberg and N. Flyer, *A Primer on Radial Basis Functions with Applications to the Geosciences*, SIAM, Philadelphia, 2015.

[14] L. V. Foster and X. Liu, *Comparison of rank revealing algorithms applied to matrices with well defined numerical ranks*, 2006.

[15] P. Godon, *Numerical modeling of tidal effects in polytropic accretion disks*, The Astrophysical Journal, 480 (1997), p. 329.

[16] G. H. Golub and C. F. Van Loan, *Matrix Computations*, Johns Hopkins University Press, 2012. 4th edition.

[17] N. Halko, P.-G. Martinsson, and J. A. Tropp, *Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions*, SIAM review, 53 (2011), pp. 217–288.

[18] W. Heinrichs, *Spectral collocation schemes on the unit disc*, J. Comput. Phys., 199 (2004), pp. 66–86.

[19] A. R. H. Heryudono and T. A. Driscoll, *Radial basis function interpolation on irregular domains through conformal transplantation*, J. Sci. Comput., 44 (2010), pp. 286–300.

[20] S. Kapurl, *An algorithm for the fast Hankel transform*, 1995. Yale technical report.

[21] A. Karageorghis, C. Chen, and Y.-S. Smyrlis, *A matrix decomposition RBF algorithm: Approximation of functions and their derivatives*, Appl. Numer. Math., 57 (2007), pp. 304–319.

[22] R. Kerswell, *Recent progress in understanding the transition to turbulence in a pipe*, Nonlinearity, 18 (2005), p. R17.

[23] J.-R. Li and J. White, *Low rank solution of Lyapunov equations*, SIAM J. Matrix Anal. Appl., 24 (2002), pp. 260–280.

[24] A. Lu and E. L. Wachspress, *Solution of Lyapunov equations by alternating direction implicit iteration*, Comp. & Math. with Appl., 21 (1991), pp. 43–58.

[25] V. N. Mahajan and G.-m. Dai, *Orthonormal polynomials in wavefront analysis: analytical solution*, JOSA A, 24 (2007), pp. 2994–3016.

[26] G. Martin, *Transformation Geometry: An Introduction to Symmetry*, Springer, New York, 2012.

[27] J. C. Mason and D. C. Handscomb, *Chebyshev Polynomials*, CRC Press, 2002.

[28] P. E. Merilees, *The pseudospectral approximation applied to the shallow water equations on a sphere*, Atmosphere, 11 (1973), pp. 13–20.

[29] F. W. Olver, D. W. Lozier, R. F. Boisver, and C. W. Clark, *NIST Handbook of Mathematical Functions*, Cambridge University Press, 2010.
[30] S. Olver and A. Townsend, *A fast and well-conditioned spectral method*, SIAM Review, 55 (2013), pp. 462–489.
[31] M. O'Neil, F. Woolfe, and V. Rokhlin, *An algorithm for the rapid evaluation of special function transforms*, App. Comp. Harm. Analy., 28 (2010), pp. 203–226.
[32] J. Pringle, *Accretion discs in astrophysics*, Annual Review of Astronomy and Astrophysics, 19 (1981), pp. 137–162.
[33] J. Sabino, *Solution of large-scale Lyapunov equations via the block modified Smith method*, PhD thesis, Rice University, 2006.
[34] E. Schmidt, *Zur Theorie der linearen und nichtlinearen Integralgleichungen. iii. Teil*, Mathematische Annalen, 65 (1908), pp. 370–399.
[35] H. A. Schwarz, *Ueber einige Abbildungsaufgaben*, Journal für die reine und angewandte Mathematik, 70 (1869), pp. 105–120.
[36] E. Serre and J. Pulicani, *A three-dimensional pseudospectral method for rotating flows in a cylinder*, Computers and Fluids, 30 (2001), pp. 491–519.
[37] J. Shen, *A new fast Chebyshev–Fourier algorithm for Poisson-type equations in polar geometries*, Appl. Numer. Math., 33 (2000), pp. 183–190.
[38] A. Townsend, *Computing with functions in two dimensions*, PhD thesis, University of Oxford, 2014.
[39] ———, *A fast analysis-based discrete hankel transform using asymptotic expansions*, SIAM J. Numer. Anal., 53 (2015), pp. 1897–1917.
[40] ———, *Gaussian elimination corrects pivoting mistakes*, arXiv preprint arXiv:1602.06602, (2016).
[41] A. Townsend and L. N. Trefethen, *An extension of Chebfun to two dimensions*, SIAM J. Sci. Comp., 35 (2013), pp. C495–C518.
[42] ———, *Continuous analogues of matrix factorizations*, in Proc. Royal Soc. A, vol. 471, 2015, pp. 1–21.
[43] A. Townsend, H. Wilber, and G. B. Wright, *Computing with functions in spherical and polar geometries I. The sphere*, SIAM J. Sci. Comp., 38-4 (2016), pp. C403–C425.
[44] L. N. Trefethen, *Spectral Methods in MATLAB*, SIAM, 2000.
[45] ———, *Householder triangularization of a quasimatrix*, IMA J. Numer. Anal., (2009), p. drp018.
[46] ———, *Approximation Theory and Approximation Practice*, SIAM, 2013.
[47] G. M. Vasil, K. J. Burns, D. Lecoanet, S. Olver, B. P. Brown, and J. S. Oishi, *Tensor calculus in polar coordinates using Jacobi polynomials*, J. Comput. Phys., (2016). to appear.
[48] Z. von F, *Beugungstheorie des schneidenver fahrens und seiner verbesserten form, der phasenkontrastmethode*, Physica, 1 (1934), pp. 689–704.
[49] H. Wilber, *Numerical computing with functions on the sphere and disk*, Master's thesis, Boise State University, 2016.
[50] G. B. Wright, M. Javed, H. Montanelli, and L. N. Trefethen, *Extension of Chebfun to periodic functions*, SIAM J. Sci. Comp., 37 (2015), pp. C554–C573.