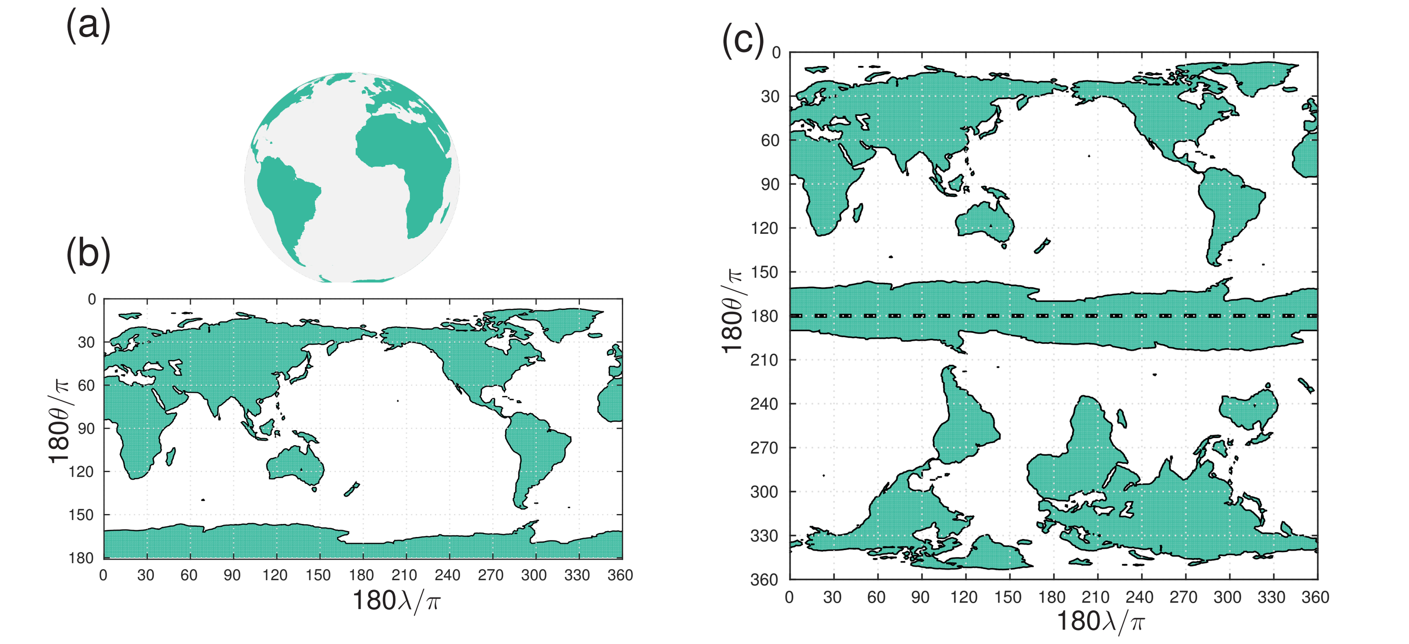
Computing with Functions on the Sphere and Disk

Heather Wilber | Boise State University ○ Dr. Alex Townsend | MIT ○ Dr. Grady Wright | Boise State University

The Double Fourier Sphere Method

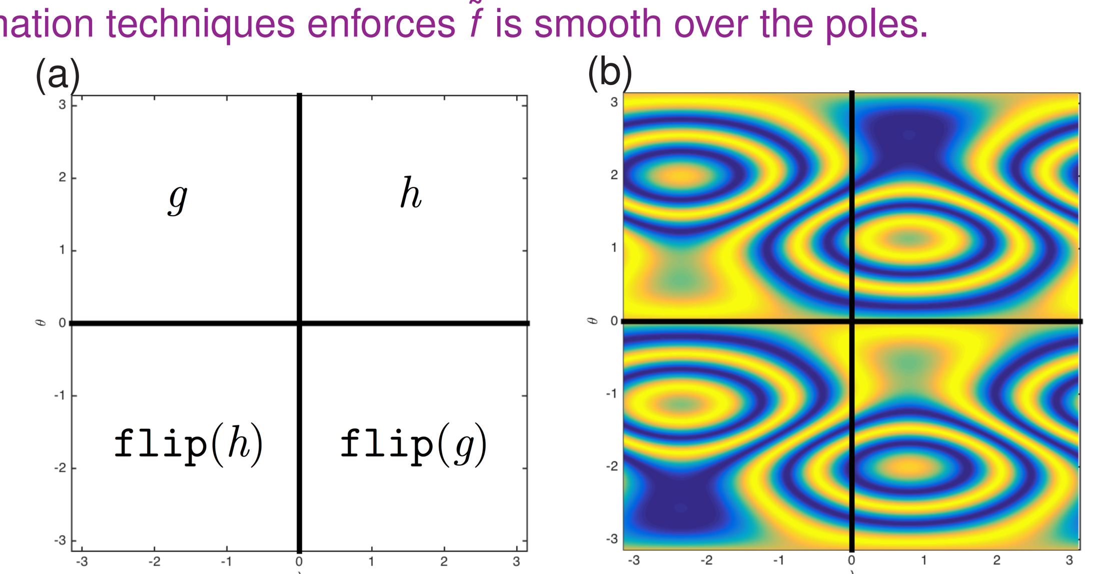
Let $f(x, y, z)$ be defined on the surface of the unit sphere. f is periodic along the great circles passing through its poles, but this feature is lost transforming f to a rectangular grid: $T := f(x, y, z) \longrightarrow f(\lambda, \theta)$, $(\lambda, \theta) \in [-\pi, \pi] \times [0, \pi]$. The function $\tilde{f}$, A doubled-up extension of f, recovers this periodicity.



The DFS method applied to the world atlas. (a) Outline of the land masses plotted on the surface of the sphere. (b) The projection of the land masses using latitude-longitude coordinates. (c) Land masses after applying the DFS method. The result is a "function" that is periodic in both longitude and latitude.

Preservation of symmetry = Smoothness over poles

The transformation T also creates artificial singularities at the poles of f. The extended function $\tilde{f}$ exhibits a particular form of symmetry, which we call block-mirror-centrosymmetric (BMC). Preserving this BMC symmetry in our approximation techniques enforces $\tilde{f}$ is smooth over the poles.



The definition of a BMC function. (a) The function $f(\lambda, \theta)$ written in terms of a quadrant I function $h(\lambda, \theta)$, and a quadrant II function $g(\lambda, \theta)$. The double Fourier extension of f gives a BMC function, $\tilde{f}$, that is defined by extending f to quadrants III and IV as described in the figure. (b) Illustration of a BMC function.

Why not use spherical harmonics?

The spherical harmonic expansion of f is

$$f(\lambda, \theta) = \sum_{\ell=0}^{\infty} \sum_{m=-\ell}^{\ell} c_{\ell,m} Y_\ell^m(\lambda, \theta), \qquad (1)$$

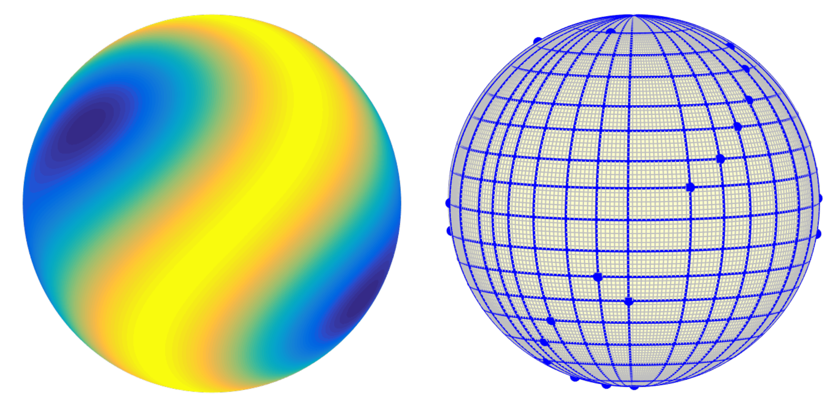
where Y_ℓ^m is the spherical harmonic function with degree ℓ and order m. Analogous to trigonometric expansions, spherical harmonics are the instinctive mathematical choice for representing functions on the surface of the sphere [1]. However, highly adaptive discretizations are computationally unfeasible using such methods. Our alternative setting enables fast algorithms via the FFT.

References: [1] K. Atkinson and W. Han, *Spherical Harmonics and Approximations on the Unit Sphere: An Introduction*, Lecture Notes in Mathematics, Springer Berlin Heidelberg, 2012. [2] M. Bebendorf, *Approximation of boundary element matrices*, Numerische Mathematik, 86 (2000), pp. 565–589. [3] G. J. Boer and L. Steinberg, *Fourier series on spheres*, Atmosphere, 13 (1975), pp. 180–191. [4] B. Fornberg, *A pseudospectral approach for polar and spherical geometries*, SIAM J. Sci. Comput., 16 (1995), pp. 1071–1081. [5] P. E. Merilees, *The pseudospectral approximation applied to the shallow water equations on a sphere*, Atmosphere, 11 (1973), pp. 13–20. [6] S. A. Orszag, *Fourier series on spheres*, Monthly Weather Review, 102 (1974), pp. 56–75. [7] E. Schmidt, *Zur theorie der linearen und nichtlinearen integralgleichungen. iii. teil*, Mathematische Annalen, 65 (1908), pp. 370–399. [8] A. Townsend and L. N. Trefethen, *An extension of Chebfun to two dimensions*, SIAM Journal on Scientific Computing, 35 (2013), pp. C495–C518.

Acknowledgements: This work is generously supported by NASA's Idaho Space Grant Consortium.

Overview

Synthesizing a classic technique known as the double Fourier sphere (DFS) method [4, 5, 6] together with new algorithmic techniques in low rank function approximation [2,8], we develop a new method of approximation for computation with functions on the sphere and disk. This approximation method preserves the bi-periodicity of the sphere, maintains smoothness over the poles in every procedure, and is near-optimal in its underlying interpolation. It resolves many drawbacks encountered by previous methods, and powers a suite of fast, scalable algorithms for computing with functions on the sphere and disk.

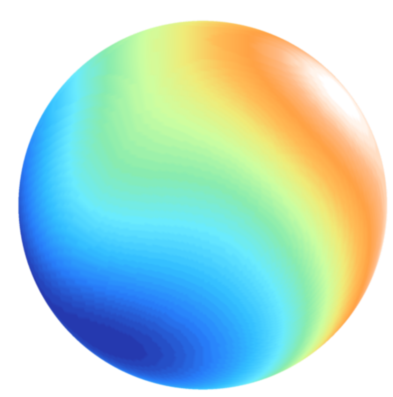


(a) (b)

Constructing an approximation: (a) The function $f(x, y, z) = \cos(xz - \sin y)$ on the sphere, constructed with `spherefun`. (b) The "skeleton" used to approximate f. Samples of f are only taken along the blue lines. The underlying tensor grid (in gray) shows the sample points required without low rank techniques.

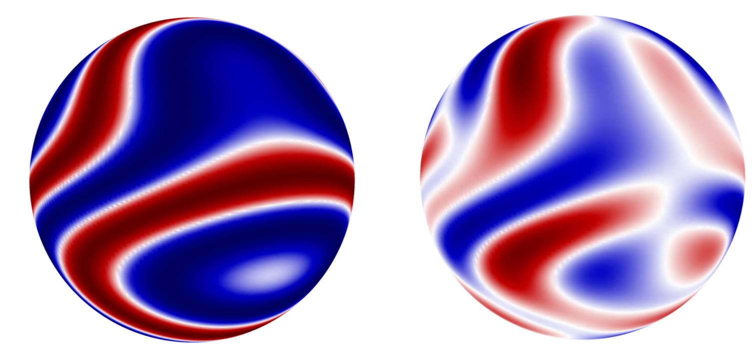
Computing with functions on the sphere: `spherefun` software

We implement our methods in Chebfun, which is an open-source software package in MATLAB. These libraries make function evaluation, integration, differentiation, vector calculus, root-finding, plotting and many other computations easy, accurate, and fast.



Integration: $\int_{\mathbb{S}^2} f \, d\Omega \longrightarrow$ **`sum2(f)`**

```
f = spherefun(@(x,y,z)
1+x+y.^2+x.^2.*y+x.^4+y.^5+
(x.*y.*z).^2);
intf = sum2(f);
exact = 216*pi/35;
error = intf-exact
 3.552713678800501e-15
```

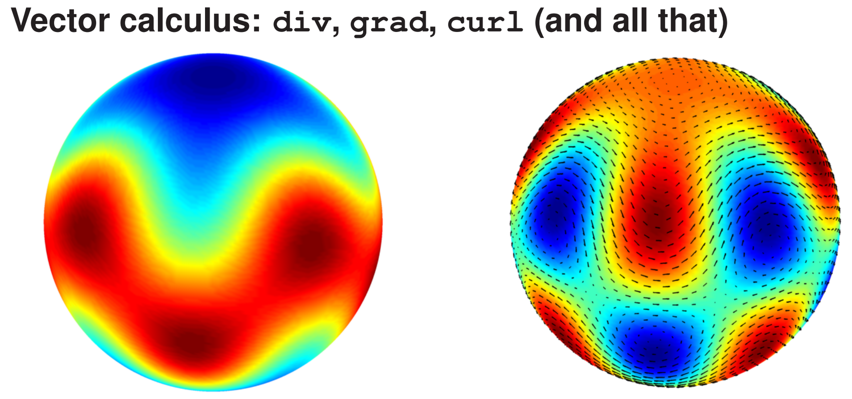


f $\qquad \frac{\partial f}{\partial y} = \frac{\cos\lambda}{\sin\theta}\frac{\partial f}{\partial \lambda} + \sin\lambda\cos\theta\frac{\partial f}{\partial \theta},$

Differentiation: $\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z} \longrightarrow$ **`diff`**

```
f = spherefun(@(lam,th)
cos(1+2*pi*(cos(lam-0.2).*sin(th))
+pi*sin(pi*cos(th))));
dfdy = diff(f, 2);
```

Vector calculus: **`div`**, **`grad`**, **`curl`** **(and all that)**

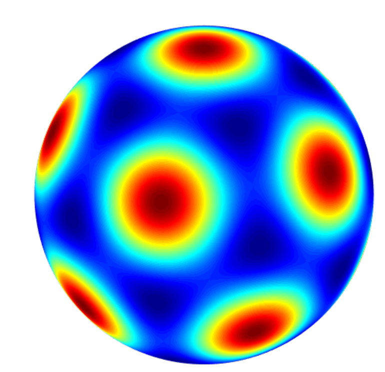


```
f = spherefun( @(l,t)
cos(4*l).*cos(t).*sin(t).^4
-cos(t).^2);
surf(f)
```

```
u = curl(f);
v = vort(u);
quiver(u)
surf(v)
```

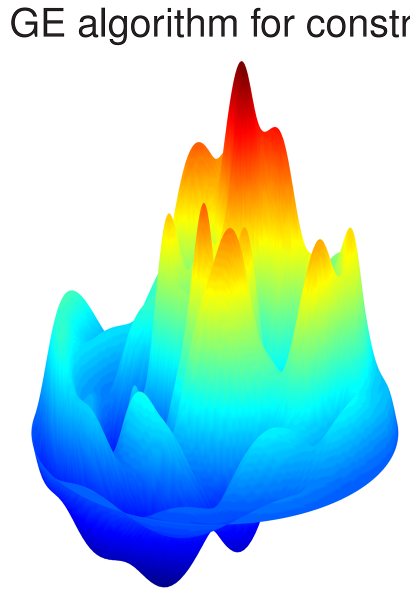
Poisson's Equation: $\Delta_{\mathbb{S}^2} u = f \longrightarrow$ **`Poisson(f)`**

```
exact = spherefun.sphharm(6,0)
+ sqrt(14/11)*spherefun.sphharm(6,5);
f = -42*exact;
u = spherefun.Poisson(f,0,16); surf(v)
err = norm(u-exact,2)
 3.794084381686475e-14
```



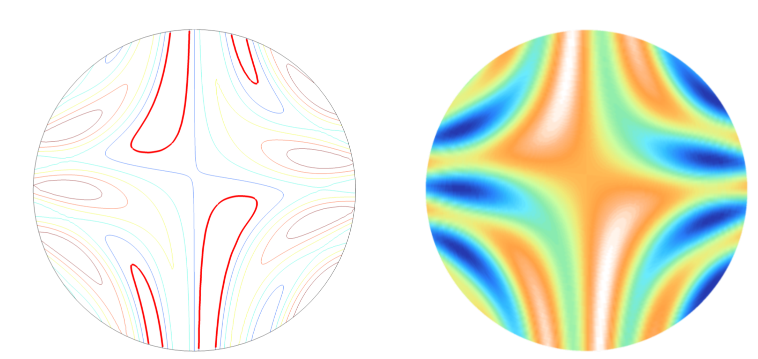
Computing with functions on the disk: `diskfun` software

A variant of the double Fourier method extends functions on the disk to BMC functions. We apply a similar GE algorithm for constructing approximants.



Algebra with disk harmonics: **`diskharm`**
The sum of six eigenfunctions of the Laplacian produce a rank 6 function.

```
nu = [1, 1, 2, 3, 5, 8];
w = (-1).^(1:6).*(0:5);
G = diskfun();
for j=1:6
  G = G +
 diskfun.diskharm(w(j),nu(j));
end
surf(G)
```



`contour(h)` `surf(h)`

Rootfinding: **`roots`**
The zero contour of $h = 2^{-\cos 4x} + \sin(7\pi x \sin(y - .1))$ is given by `roots(h)` and displayed in red.

Low rank approximation of BMC functions

The function $\tilde{f}(\lambda, \theta)$ is said to be

- **rank 1** if it is nonzero and can be written as $\tilde{f}(\lambda, \theta) = c(\theta)r(\lambda)$.
- **rank at most** K if it can be expressed as a sum of K rank 1 functions.

Most functions are of infinite rank, but smooth functions can often be approximated well by a finite, low rank function:

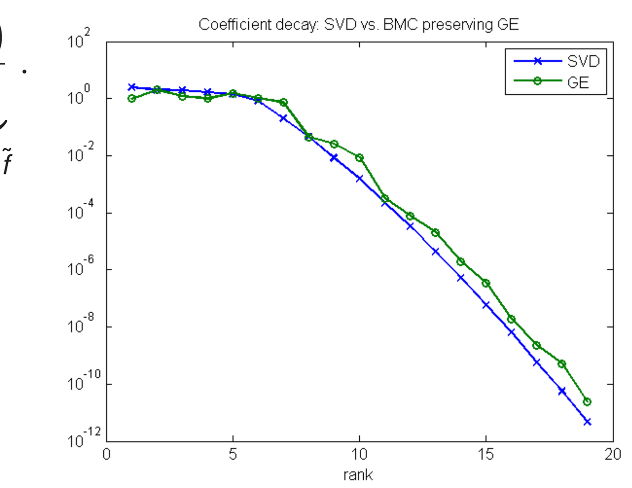
$$\tilde{f}(\lambda, \theta) \approx \tilde{f}_K(\lambda, \theta) = \sum_{j=1}^{K} d_j c_j(\theta) r_j(\lambda). \qquad (2)$$

The continuous SVD (Karhunen-Loève expansion) is an optimal way to derive such an approximation [7], but it is computationally expensive. Our method gives a near-optimal approximation with near-optimal computational complexity.

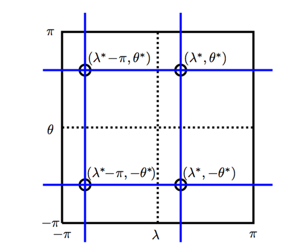
Iterative Gaussian elimination (GE)

Iterative GE is a near-optimal method for deriving a low-rank approximation [8]. It proceeds by choosing a maximal value $\tilde{f}(\lambda_*, \theta_*)$, which serves as a pivot. Performing a GE-step with the pivot results in a reduced-rank update of $\tilde{f}$:

$$\tilde{f}(\lambda, \theta) \longleftarrow \tilde{f}(\lambda, \theta) - \underbrace{\frac{\tilde{f}(\lambda_*, \theta)\tilde{f}(\lambda, \theta_*)}{\tilde{f}(\lambda_*, \theta_*)}}_{\text{A rank 1 approx. to } \tilde{f}}. \qquad (3)$$



Running the algorithm forward K steps subtracts a series of K rank 1 terms from $\tilde{f}$. This series of terms, when added together, reproduces a rank K approximation to $\tilde{f}$. Problematically, this method does not preserve the BMC structure of $\tilde{f}$.



Structure-preserving GE

To preserve BMC structure, we must simultaneously eliminate symmetric pairs of rows and columns, as shown in the figure (left). A block-pivoting strategy with a 2×2 matrix accomplishes this.

Define a matrix M based on the BMC-symmetric values:

$$M = \begin{bmatrix} \tilde{f}(\lambda^* - \pi, \theta^*) & \tilde{f}(\lambda^*, \theta^*) \\ \tilde{f}(\lambda^* - \pi, -\theta^*) & \tilde{f}(\lambda^*, -\theta^*) \end{bmatrix} = \begin{bmatrix} \tilde{f}(\lambda^* - \pi, \theta^*) & \tilde{f}(\lambda^*, \theta^*) \\ \tilde{f}(\lambda^*, \theta^*) & \tilde{f}(\lambda^* - \pi, \theta^*) \end{bmatrix} = \begin{bmatrix} a & b \\ b & a \end{bmatrix}.$$

Our structure-preserving GE step, analogous to (3), is given by

$$\tilde{f}(\lambda, \theta) \longleftarrow \tilde{f}(\lambda, \theta) - \begin{bmatrix} \tilde{f}(\lambda^* - \pi, \theta) & \tilde{f}(\lambda^*, \theta) \end{bmatrix} M^{+_\epsilon} \begin{bmatrix} \tilde{f}(\lambda, \theta^*) \\ \tilde{f}(\lambda, -\theta^*) \end{bmatrix},$$

where M^{+_ϵ} is the ϵ-pseudoinverse of M.

Algorithm: Structure-preserving GE on BMC functions
Set $\tilde{f}_0 = 0$ and $\tilde{e}_0 = \tilde{f}$

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

Find (λ_k, θ_k) such that $M = \begin{bmatrix} a & b \\ b & a \end{bmatrix}$, where $a = \tilde{e}_{k-1}(\lambda_{k-1} - \pi, \theta_{k-1})$ and

$b = \tilde{e}_{k-1}(\lambda_{k-1}, \theta_{k-1})$ maximizes $\sigma_1(M)$, the largest singular value of M

$\tilde{e}_k = \tilde{e}_{k-1} - \begin{bmatrix} \tilde{e}_{k-1}(\lambda_k - \pi, \theta) & \tilde{e}_{k-1}(\lambda_k, \theta) \end{bmatrix} M^{+_\epsilon} \begin{bmatrix} \tilde{e}_{k-1}(\lambda, \theta_k) \\ \tilde{e}_{k-1}(\lambda, -\theta_k) \end{bmatrix}$

$\tilde{f}_k = \tilde{f}_{k-1} - \begin{bmatrix} \tilde{e}_{k-1}(\lambda_k - \pi, \theta) & \tilde{e}_{k-1}(\lambda_k, \theta) \end{bmatrix} M^{+_\epsilon} \begin{bmatrix} \tilde{e}_{k-1}(\lambda, \theta_k) \\ \tilde{e}_{k-1}(\lambda, -\theta_k) \end{bmatrix}$

end

Pseudocode for our structure-preserving GE procedure on BMC functions. This is a continuous idealization and in practice we use a discretization of this procedure and terminate it after a finite number of steps.

Geometric convergence for sufficiently analytic functions

Theorem: Let $\tilde{f} : [-\pi, \pi] \times [-\pi, \pi] \to \mathbb{R}$ be a BMC function such that $f(\lambda, \cdot)$ is continuous for any $\lambda \in [-\pi, \pi]$ and $f(\cdot, \theta)$ is analytic and uniformly bounded in a stadium S of radius $(1 + \alpha)\rho\pi$, $\rho > 1$, for $\theta \in [-\pi, \pi]$.

Then, as $k \to \infty$, $||\tilde{e}_k||_\infty \to 0$. That is, the BMC preserving GE procedure converges. Furthermore, this convergences is geometric.

email: heatherwilber@boisestate.edu