Scalar Spherical Harmonics and Slepian Functions

Contents

1 Plot a single spherical harmonic function

We will demonstrate plotting a spherical-harmonic on a sphere, in a standard Matlab plot, on a Mollweide projection, and on random points of a sphere.

First, designate a spherical-harmonic to be plotted:

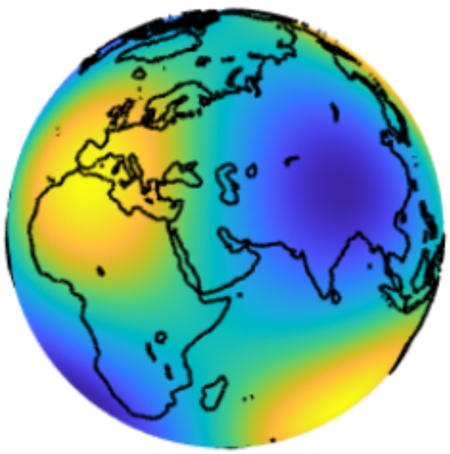
For example,

`l` = 3; `m` = -2;

0 `l` fixes the degree and `m` fixes the order.

1.1 Plot on sphere

```
l = 3; m = -2;
lon = 0:0.5:360;
lat = -90:0.5:90;
Y = ylm(l, m, pi/180*(90-lat), pi/180*lon);
figure;
plotplm(Y, pi/180*lon, pi/180*lat,2)
```



1. Create a grid on the sphere

```
lon = 0:0.5:360; lat = -90:0.5:90;
```

This creates a coordinate point every half-degree.

2. Calculate the values of the function for coordinate points on the sphere

```
Y = ylm(l, m, pi/180*(90-lat), pi/180*lon);
```

The function `slepian_alpha/ylm.m` evaluates the spherical harmonic function of degree `l` and order `m` at every point `pi/180*(90-lat)`, `pi/180*lon` on the grid. We name the vector of the spherical-harmonic values `Y`.

Note that `90-lat` is needed to convert latitude to colatitude and `pi/180` is needed to convert degrees to radians.

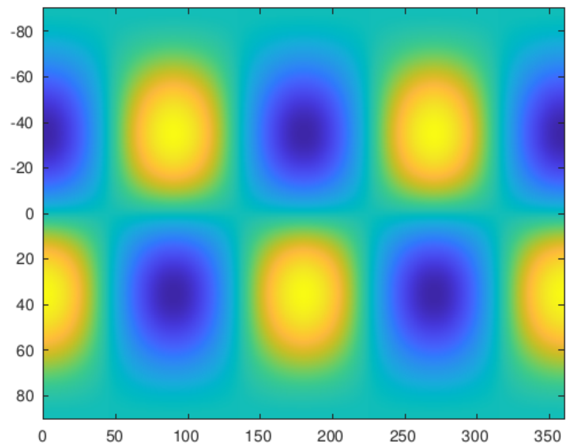
3. Plot

```
figure; plotplm(Y, pi/180*lon, pi/180*lat,2)
```

The function `slepian_alpha/plotplm.m` is here used to plot the vector `Y` using the grid specified by `lon` and `lat` in step 1. The input `2` dictates that the graph be on a sphere.

1.2 Plot in standard Matlab plot

```
l = 3; m = -2;
lon = 0:0.5:360;
lat = -90:0.5:90;
Y = ylm(l, m, pi/180*(90-lat), pi/180*lon);
imagesc(lon, lat, Y)
```

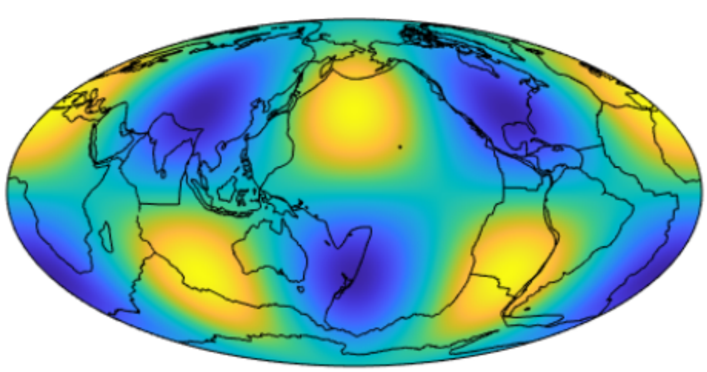


Do steps 1 and 2, and then run

```
imagesc(lon, lat, Y)
```

1.3 Plot on Mollweide projection

```
l = 3; m = -2;
lon = 0:0.5:360;
lat = -90:0.5:90;
Y = ylm(l, m, pi/180*(90-lat), pi/180*lon);
figure;
plotplm(Y, pi/180*lon, pi/180*lat,1)
```



Do steps 1 and 2, and then run

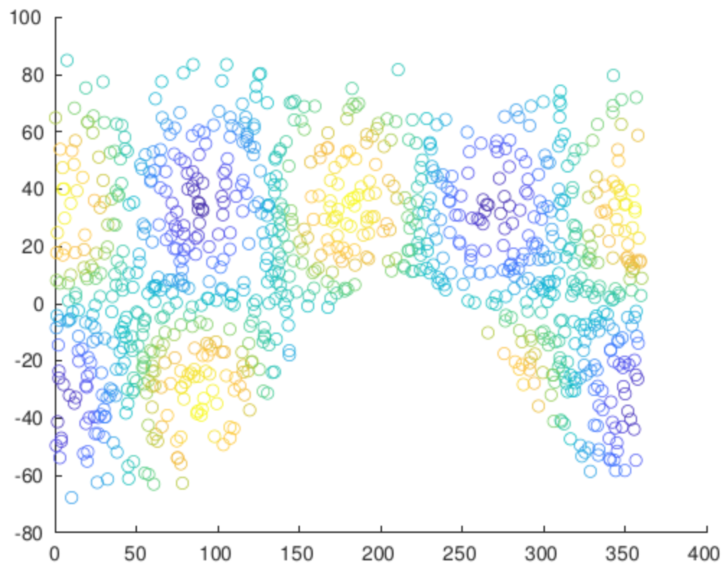
```
figure;

plotplm(Y, pi/180*lon, pi/180*lat,1)
```

The input `1` dictates that the graph be on the Mollweide projection.

1.4 Plot for random points on a sphere

```
l = 3; m = -2;
TH = 120; lon0 = 30; cola0 = 40; N=1000;
[lon, lat] = randpatch(N,TH,lon0,cola0);
Y = ylm(l, m, pi/180*(90-lat), pi/180*lon,[],[],[],1);
scatter(lon, lat, [], Y)
```



1. Generate a subset of the sphere consisting of random points

In particular, we will create `N` randomly-generated coordinate points within a spherical cap of opening angle `TH` and centered at longitude `lon0` and colatitude `cola0`

For example,

```
TH = 120; lon0 = 30; cola0 = 40; N=1000;

[lon, lat] = randpatch(N,TH,lon0,cola0);
```

The function `slepian_alpha/randpatch.m` creates the set of random points within the spherical cap of the specified values. We name those coordinate points `[lon,lat]`.

2. Calculate the values of the spherical harmonic at those points

```
Y = ylm(l, m, pi/180*(90-lat), pi/180*lon,[],[],[],1);
```

`ylm.m` takes the arguments `l`, `m`, `pi/180*(90-lat)`, `pi/180*lon` as before. Run `help ylm` for information on all eight arguments.

3. Plot

If necessary, use the Matlab command

```
clf;
```

To clear existing figures, and then run the Matlab command

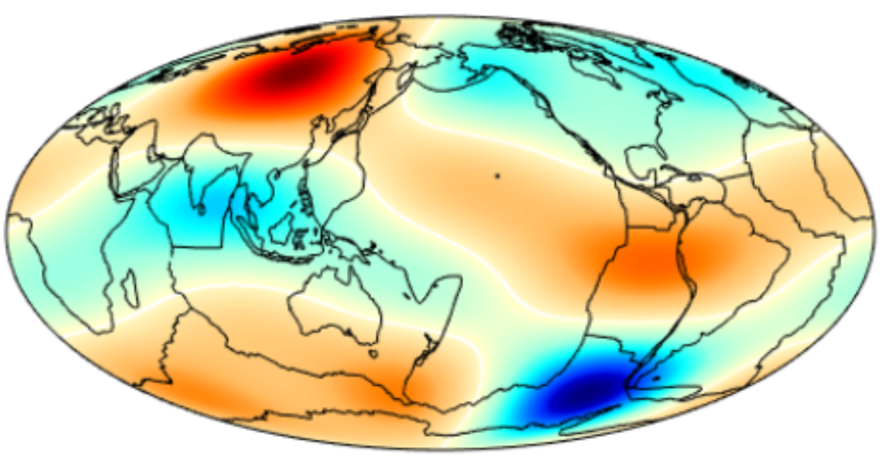
```
scatter(lon, lat, [], Y)
```

To create a scatter plot of circles having locations `[lon, lat]`. Here, `[]` indicates the default value for circle size and the vector of spherical-harmonic values `Y` is used to determine circle color.

Please see `Ch_01` in the `.edu` folder for more detailed information.

2 Plot a linear combination of spherical harmonics

```
lon = 0:0.5:360;
lat = -90:0.5:90;
Y1=ylm(3,1,pi/180*(90-lat),pi/180*lon);
Y2=ylm(1,-1,pi/180*(90-lat),pi/180*lon);
Y3=ylm(5,-2,pi/180*(90-lat),pi/180*lon);
Y4=4*Y1-0.2*Y2+2*Y3;
plotplm(Y4, pi/180*lon, pi/180*lat,1);
kelicol(1)
```



This task is a simple variation on the first.

Let us define three spherical harmonics:

```
lon = 0:0.5:360;
```

```
lat = -90:0.5:90;
```

```
Y1=ylm(3,1,pi/180*(90-lat),pi/180*lon);
```

```
Y2=ylm(1,-1,pi/180*(90-lat),pi/180*lon);
```

```
Y3=ylm(5,-2,pi/180*(90-lat),pi/180*lon);
```

Next, create a vector which is a linear combination of these three. For example,

```
Y4=4*Y1-0.2*Y2+2*Y3;
```

To plot the function, use `plotplm.m`. For example,

```
plotplm(Y4, pi/180*lon, pi/180*lat,1)
```

If you're interested in another color scheme, try out

```
kelicol(1)
```

Please see `Ch_01` in the `.edu` folder for more detailed information.