Montestigliano Workshop	Problem 3 Geometric modeling with RBFs	Level 1 Grady Wright

Here we consider how RBF interpolants on the sphere can be used for geometric modeling of surfaces that can be parameterized using spherical coordinates. The idea is to use RBF interpolation on the sphere in a parametric fashion, where each coordinate of the surface is interpolated with an RBF. The idea was first proposed in [1] for geometric modeling of platelets. Here is an overview of the procedure.

Suppose $\mathcal{S}$ is a two-dimensional surface in $\mathbb{R}^3$ that can be represented parametrically in spherical coordinates as follows:

$$\mathcal{S} = \left\{\mathbf{x} \in \mathbb{R}^3 \middle| \mathbf{x}(\lambda,\theta) = (x(\lambda,\theta), y(\lambda,\theta), z(\lambda,\theta))\right\} \tag{1}$$

where $-\pi \le \lambda \le \pi$ and $-\pi/2 \le \theta \le \pi/2$. The functions $x(\lambda,\theta)$, $y(\lambda,\theta)$, and $z(\lambda,\theta)$ are assumed to be smooth and $\mathbf{x}(\lambda,\theta)$ is assumed to satisfy the end conditions:

$$\begin{aligned}
\mathbf{x}(-\pi,\theta) &= \mathbf{x}(\pi,\theta), \text{ for } -\pi/2 \le \theta \le \pi/2,\\
\mathbf{x}(\lambda,\pi/2) &= \mathbf{x}(\lambda+\pi,\pi/2), \text{ for } -\pi \le \lambda \le 0,\\
\mathbf{x}(\lambda,-\pi/2) &= \mathbf{x}(\lambda+\pi,-\pi/2), \text{ for } -\pi \le \lambda \le 0,\\
\mathbf{x}(\lambda,\pi/2) &= \mathbf{x}(\lambda-\pi,\pi/2), \text{ for } 0 < \lambda \le \pi,\\
\mathbf{x}(\lambda,-\pi/2) &= \mathbf{x}(\lambda-\pi,-\pi/2), \text{ for } 0 < \lambda \le \pi.
\end{aligned}$$

These end enforce periodicity of $\mathbf{x}$ in λ and continuity of $\mathbf{x}$ at the poles of the spherical coordinate system. In the case the object is a sphere of radius r, $\mathbf{x}(\lambda,\theta) = (r\cos\lambda\cos\theta, r\sin\lambda\cos\theta, r\sin\theta)$.

The parametric interpolation idea is to reconstruct the surface $\mathbf{x}(\lambda,\theta)$ from smooth interpolations of each of its components which are given at some finite collection of locations

$$\{\mathbf{x}(\lambda_k,\theta_k)\}_{k=1}^N = \{(x(\lambda_k,\theta_k), y(\lambda_k,\theta_k), z(\lambda_k,\theta_k)\}_{k=1}^N.$$

These are the values of the data we are going to interpolate at the nodes $\{(\lambda_k,\theta_k)\}_{k=1}^N$, which happen to be nodes on the unit sphere. Figure 1 illustrates this reconstruction problem, of which the main ingredient is the interpolation of a function defined on the unit sphere.

For this application it may make sense to express an RBF interpolant on the unit sphere in terms of spherical coordinates. This is rather easy to do by simply noting that if $\mathbf{x}, \mathbf{x}_j \in \mathbb{S}^2$, then $\mathbf{x} = (\cos\lambda\cos\theta, \sin\lambda\cos\theta, \sin\theta)$ and $\mathbf{x}_j = (\cos\lambda_j\cos\theta_j, \sin\lambda_j\cos\theta_j, \sin\theta_j)$. Then the Euclidean distance between $\mathbf{x}$ and $\mathbf{x}_j$ is

$$\|\mathbf{x}-\mathbf{x}_j\| = \sqrt{2(1-\cos\theta\cos\theta_k\cos(\lambda-\lambda_k) - \sin\theta\sin\theta_k)}$$

So, the *basic* RBF interpolant to the values $\{f_j\}_{j=1}^N$ sampled at $X = \{\mathbf{x}_j\}_{j=1}^N$ takes the form

$$I_X f(\mathbf{x}) := s(\lambda,\theta) = \sum_{j=1}^N c_k \phi\left(\sqrt{2(1-\cos\theta\cos\theta_j\cos(\lambda-\lambda_j) - \sin\theta\sin\theta_j)}\right), \tag{2}$$

and the interpolation coefficients are determined by requiring $s(\lambda_j,\theta_j) = f_j$, $j = 1,\ldots,N$. The *general* RBF interpolant can be similarly written in terms of spherical coordinates by expressing the additional spherical harmonics in terms of (λ,θ).

The goal is to reconstruct $\{x(\lambda_k,\theta_k)\}_{k=1}^N$ with an interpolant $s^x(\lambda,\theta)$, $\{y(\lambda_k,\theta_k)\}_{k=1}^N$ with an interpolant $s^y(\lambda,\theta)$, and $\{z(\lambda_k,\theta_k)\}_{k=1}^N$ with an interpolant $s^z(\lambda,\theta)$, all of the form (2) (or using the general RBF interpolant with appended spherical harmonics).

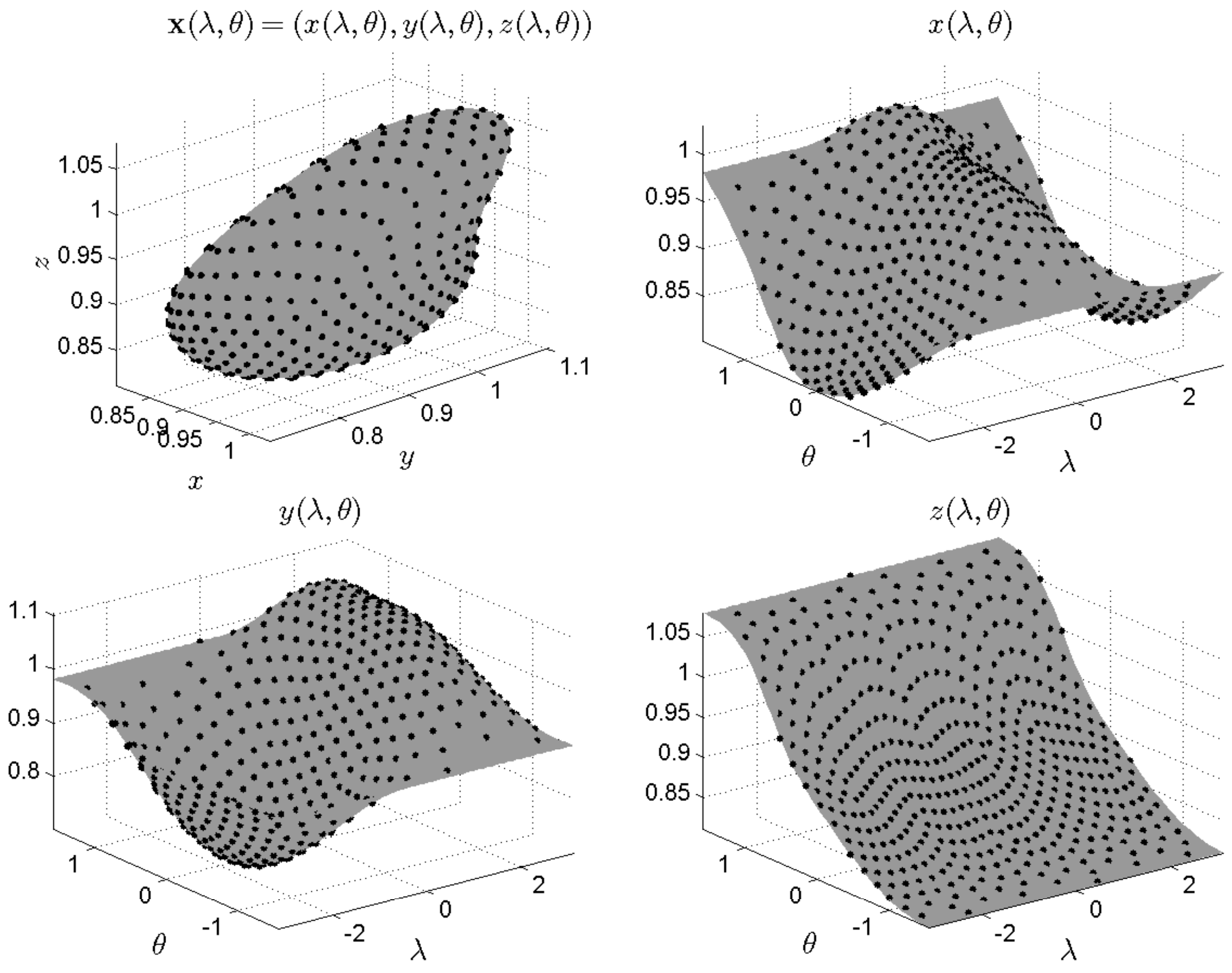

Figure 1: Illustration of the parametric representation of a 3D object $\mathbf{x}(\lambda, \theta)$ and the reconstruction from a finite number of data sites. Top left figure shows the 3D object together with discrete data sites $\{(x(\lambda_k, \theta_k), y(\lambda_k, \theta_k), z(\lambda_k, \theta_k)\}_{k=1}^{N}$ represented as black solid spheres. Top right figure shows the x component of the object in spherical parametric space and its values at the node set $\{(\lambda_k, \theta_k)\}_{k=1}^{N}$ (marked by black solid spheres), while the bottom left and right figures show the respective y and z components and their corresponding values. The goal is to reconstruct $x(\lambda, \theta)$, $y(\lambda, \theta)$, and $z(\lambda, \theta)$ from interpolations of the values at the node sets shown and then use these to reconstruct $\mathbf{x}(\lambda, \theta)$.

3.a) Load the data file BumpySphere, from the "data" directory of the `rbfsphere` package using the command

```
load BumpySphere
```

This file contains an N-by-3 array with the (x, y, z) coordinates of points on the so called Bumpy Sphere surface from the AIM@shape repository `http://shapes.aimatshape.net`. You can visualize these points using the command

```
plot3(x(:,1),x(:,2),x(:,3),'b.');
```

Construct a parametric model of this surface using RBF interpolation as discussed above. Make a plot of the resulting surface similar to the plot in the upper left of Figure 1, but make it more colorful than this one (don't forget to add the original sample points).

Hint: You can get the spherical coordinates for the data in the `x` array using the command:

```
[lambda,theta] = cart2sphm(x);
```

Now you can use these values as the nodes in your RBF interpolant of each column of the `x` array. You can plot the surface by sampling the interpolant of each component of `x` on a regular longitude-latitude grid:

```
[L,T] = meshgrid(linspace(-pi,pi,81),linspace(-pi/2,pi/2,41));
```

If sx, sy, and sz denote the interpolants of each column of `x`, respectively, then the parametric surface can be plotted as

```
surf(sx,sy,sz,'FaceColor','g','EdgeColor','none');
```

You can beautify it using, the `camlight` and `lighting phong` functions (as well as many others).

References

[1] V. Shankar, G. B. Wright, A. L. Fogelson, and R. M. Kirby, *A study of different modeling choices for simulating platlets within the immersed boundary method*, Appl. Num. Math., 63 (2013), pp. 58–77.