Radial Basis Function Interpolation and Kansa's Method on Surfaces

Alex Shaffer

April 2024

1 Background

The goal of this independent study was to do two things. First, perform radial basis function interpolation in a non-Euclidean metric space. Second, to use that to perform the numerical partial differential equation (PDE) solution method called Kansa's method to solve some PDE on a non-Euclidean metric space.

Radial basis function (RBF) interpolation is a mesh free approximation method that creates an interpolant of the form in equation 1. The functions $\varphi(r)$ are usually taken to be one of the following: the multiquadric $\varphi(r) = \sqrt{\epsilon^2 + r^2}$, the polyhamonic splines $\varphi(r) = r^n log(r)$ for n = 2, 4, 6 .. and $\varphi(r) = r^n$ for n = 1, 3, 5, ..., and the gaussian $\varphi(r) = e^{-(\epsilon r)^2}$. These are also often refered to as kernels. We may solve for α_i by enforcing the interpolation condition (matching the data exactly) and one is left with the following linear system to solve in equation 2.

$$\hat{\phi}(x) = \sum^{N} \alpha_i \varphi(|x_i - x|) \tag{1}$$

$$\begin{bmatrix} f(x_1) \\ f(x_2) \\ \vdots \\ f(x_i) \end{bmatrix} = \begin{bmatrix} \varphi(||x_1 - x_1||) & \varphi(||x_2 - x_1||) & \cdots & \varphi(||x_i - x_1||) \\ \varphi(||x_1 - x_2||) & \varphi(||x_2 - x_2||) & \cdots & \varphi(||x_i - x_2||) \\ \vdots & \vdots & \ddots & \vdots \\ \varphi(||x_1 - x_i||) & \varphi(||x_2 - x_i||) & \cdots & \varphi(||x_i - x_i||) \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_i \end{bmatrix} \tag{2}$$

Kansa's method is a fairly natural extension of this. For a given boundary value problem one may expect an interpolant of the solution as given in equation 1. This time however we don't have a sampling of the function but rather we have an some discrete differential operator and some specification of a boundary value. Let us call the discrete differential operator $\mathscr{L}$. β corresponds to the boundary condition operator. This can be taken to be Dirichlet, Neumann, or

others. We specify what the differential operator should equal with $f(x_i)$ and what the boundary condition should equal as $g(x_i)$. Now we may solve the following for α_i in equations 3 and 4 [1]. In these equation indices i up to N_i correspond to an internal point, and the rest correspond to boundary points, up to index N.

$$\sum_{1}^{N_i}(\mathscr{L}\varphi)(||x_i - x_j||)\alpha_j = f(x_i), \text{ for } i = 1, 2, 3..., N_i \tag{3}$$

$$\sum_{N_i+1}^{N}(\beta\varphi)(||x_i - x_j||)\alpha_j = g(x_i), \text{ for } i = N_i + 1, N_i + 2, ..., N \tag{4}$$

2 RBFs on Surfaces

We may now consider the non-Euclidean space of the surface. from a computational aspect we will model these discretely as a triangle mesh (simplicial 2-complex). In this case we may perform RBF interpolation in this space, but we replace the Euclidean metric $||x_i - x_j||$ with the geodesic distance between points on the surface which we will just denote $d(x_i, x_j)$. The particular method employed here is the MMP algorithm [2]. As a demostration of the technique let us consider a uv-sphere triangle mesh. Let us assign a function to points on that surface. This will be the one in equation 5. This could also be rewritten in terms of the azimuthal angle which is a parameter of the surface. To perform the interpolation we take every tenth index in the sphere which is an even distribution in the domain it is mapped from, but it is not even in $\mathbb{R}^3$ it is more dense near the poles and less dense in the equator. It is possible to compare the rbf constructed using the geodesic distance metric vs the one in $\mathbb{R}^3$. This is done in figure 3. One can see that the geodesic distance does a better job at minimizing the spread of inaccurate values that are particularly present near the equator.

$$f(x, y, z) = \begin{cases} 1 & z \geq 0 \\ -1 & z < 0 \end{cases} \tag{5}$$

3 Kansa's on Surfaces

When computing a solution to a boundary value problem using Kansa's method it is particularly useful to encode the differential operator of interest through a matrix. First recall the form of equation 1. Taking $(\mathscr{L}\hat{\phi})(x)$, we see that we get $(\mathscr{L}\hat{\phi})(x) = \sum^{N_i} \alpha_j(\mathscr{L}\varphi)(d(x, x_j))$. Then we may encode $\mathscr{L}$ as a matrix $\mathbf{L}$ which when multiplied by a vector of function values $\mathbf{u}$ gives $\mathbf{Lu} = (\mathscr{L}u)_i$. Suppose we describe a new matrix $\mathbf{\Phi}$ where $\Phi_{ij} = \varphi(d(x_i, x_j))$. Then we get that $(\mathscr{L}\hat{\phi})(x_i) = (\mathbf{L\Phi a})_i$ where $\mathbf{a}$ is a vector of the weights α_j. This allows us to build a system to solve equations 3 and 4.

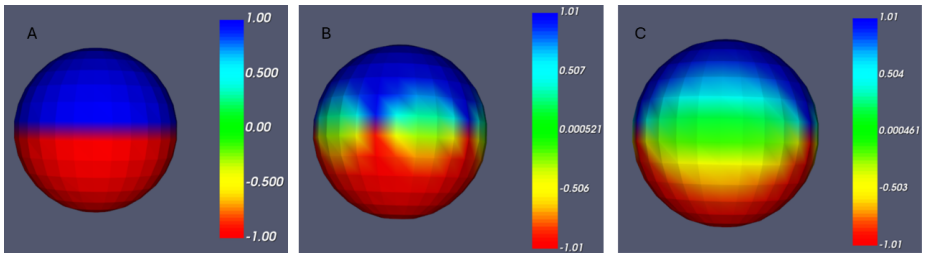

Figure 1: A) The exact coloring based on the function. B) RBF interpolation using geodesic distance. C) RBF interpolation done using the Euclidean $\mathbb{R}^3$ metric

One example that may be computed is a solution to Laplace's equation on a square with specified Dirichlet boundary values. That is to say the domain D = [0, 2] x [0, 2] with the conditions $\Delta u(x, y) = 0$, $u(x, 0) = u(x, 2) = u(0, y) = 0$ and $u(2, y) = sin(\pi y/2)$. These conditions are chosen for simplicity, so that an analytical solution will only have one term and not be a Fourier series to match the boundary condition. The analytical solution then takes the form $u(x, y) = \frac{sinh(\pi x/2)}{sinh(\pi)} sin(\pi y/2)$.

In order to calculate the approximate solution we will take a flat triangle mesh that covers the domain. In this case vertices with an even 0.2 spacing in each direction. Because it is a triangle mesh each vertex will end up having 6 edges in order to cover the domain in the manner. This example was chosen because the Laplacian matrix is well established and easily calculated for a triangle mesh [3]. For this computation the multiquadric kernel is employed due to ubiquity in the literature, and its accurate result here. The L2 error norm of the resulting solution vectors (analytical vs numerical) is 0.0137. In the context of the maximum and minimum values being 1 and 0 respectively this is low.

It is now possible to move to a non-Euclidean case. Each of the z values for the triangle mash is changed so that it is no longer flat. Specifically, the z values are prescribed by $z(x, y) = sin(\pi x/2)sin(\pi y/2)$. The Laplacian matrix is then recomputed and Laplace's equation is solved again. This time the geodesic distance makes a difference as the setting is no longer flat.

4 Concluding Remarks

This work has demonstrated some of the utility of Radial Basis Function interpolation to non-Euclidean spaces and to numerical solutions to partial differential solutions in those spaces. It is however important to note that the well-posedness of Kansa's method is still unknown. That is to say that the existence of a solution to the system described by equations 3 and 4 is not well established, as the differential operator may violate the symmetry that is typically present in the standard RBF system which guarantees the existence of its solution for positive

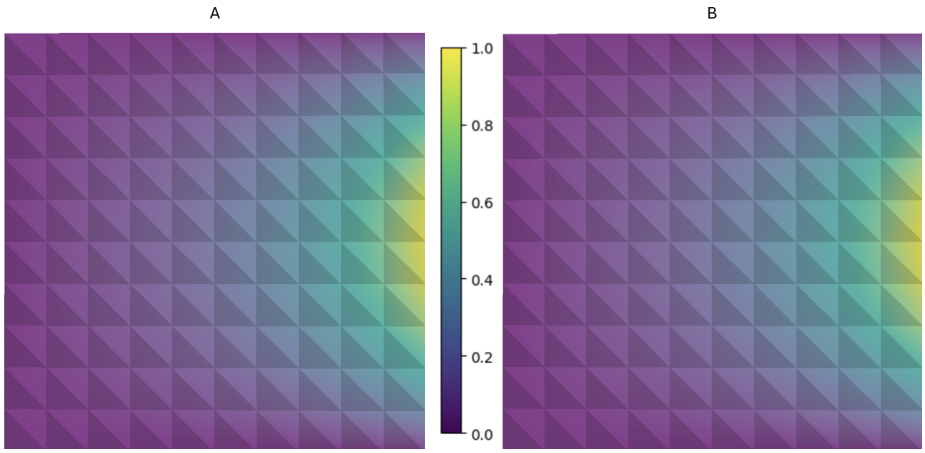

Figure 2: A) The exact solution to Laplace's equation with the given boundary values. B) The numerical solution to Laplace's equation using the discrete Laplacian and Kansa's method.

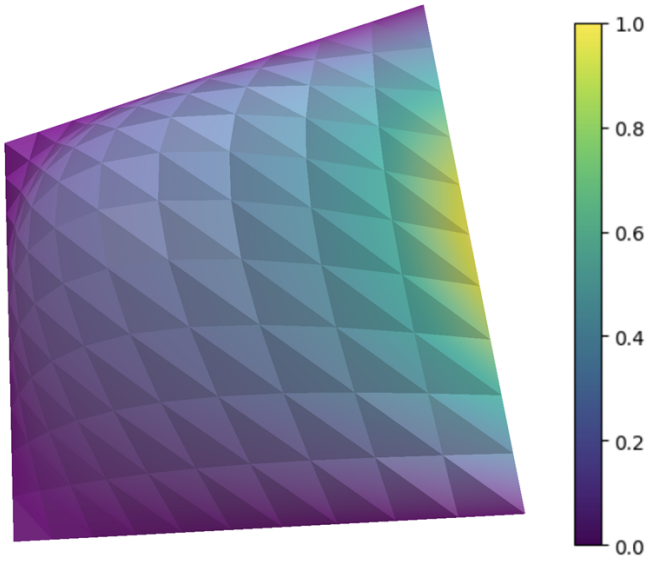

Figure 3: The solution to Laplace's Equation on a surface using Kansas's method and geodesic distance

definite kernels.

The systems of linear equations that arise in the computation of RBF interpolants and in Kansa's method is typically limited through the size of data set. This can be alleviated by the use of compact support radial basis functions. These compact support functions cause the systems to be sparse, allowing for more efficient methods to be used to solve them. Additionally, discrete differential operators typically are described in sparse matrices, as they are local operators. This means that Kansa's method may even be viable for large data sets as long as a compact support kernel is used. For future work it would be interesting to explore this.

References

[1] J. Li, A. H.-D. Cheng, and C.-S. Chen, "A comparison of efficiency and error convergence of multiquadric collocation method and finite element method," *Engineering Analysis with Boundary Elements*, vol. 27, no. 3, pp. 251–257, 2003. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0955799702000814

[2] V. Surazhsky, T. Surazhsky, D. Kirsanov, S. J. Gortler, and H. Hoppe, "Fast exact and approximate geodesics on meshes," *ACM Trans. Graph.*, vol. 24, no. 3, p. 553–560, jul 2005. [Online]. Available: https://doi.org/10.1145/1073204.1073228

[3] N. Sharp and K. Crane, "A laplacian for nonmanifold triangle meshes," *Computer Graphics Forum*, vol. 39, no. 5, pp. 69–80, 2020. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.14069