

# RBF Interpolants Over Near-Flat Surfaces

Caleb Jacobs

27 April 2022

## Abstract

In the context of smooth radial basis function (RBF) based methods, utilizing the collocation matrix with a small shape parameter is desirable. A small shape parameter tends to increase accuracy of RBF interpolants or finite difference weights. To properly resolve systems with a small shape parameter, methods such as RBF-RA are used. Methods such as RBF-RA work great for data that is on a flat surface or a curved surface. However, applying RBF-RA to data that is on a near-flat surface leads to erroneous results. We present results based on asymptotology and perturbation theory to better understand these erroneous results. Furthermore, we present current, in-progress research to again further this understanding.

## 1 Background

In the context of data interpolation, finite differences, and an abundance of many other data driven problems, radial basis function (RBF) based methods are becoming more and more popular due to there flexibility and high orders of accuracy. An RBF is any radially symmetric function  $\phi_\varepsilon(r)$  where  $\varepsilon$  is called the shape parameter. For this article, we will focus on smooth RBFs and even more specifically, we will only consider the GA RBF in [Table 1](#). As mentioned before, RBFs can be used as a general interpolation basis which we will consider next.

Suppose we have data  $\{\mathbf{x}_i, f_i\}_{i=1}^n$  that we would like to interpolate using RBFs. Then, an RBF interpolant of the data can be written as

$$s(\mathbf{x}) = \sum_{i=1}^n \lambda_i \phi_\varepsilon(\|\mathbf{x} - \mathbf{x}_i\|_2) \quad (1)$$

where  $\lambda_i$  are the interpolation weights. This interpolant is incomplete in that we still need to determine each  $\lambda_i$ . A linear system for  $\lambda_i$  is given by

$$\underbrace{\begin{pmatrix} \phi_\varepsilon(\|\mathbf{x}_1 - \mathbf{x}_1\|) & \phi_\varepsilon(\|\mathbf{x}_1 - \mathbf{x}_2\|) & \cdots & \phi_\varepsilon(\|\mathbf{x}_1 - \mathbf{x}_n\|) \\ \phi_\varepsilon(\|\mathbf{x}_2 - \mathbf{x}_1\|) & \phi_\varepsilon(\|\mathbf{x}_2 - \mathbf{x}_2\|) & \cdots & \phi_\varepsilon(\|\mathbf{x}_2 - \mathbf{x}_n\|) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_\varepsilon(\|\mathbf{x}_n - \mathbf{x}_1\|) & \phi_\varepsilon(\|\mathbf{x}_n - \mathbf{x}_2\|) & \cdots & \phi_\varepsilon(\|\mathbf{x}_n - \mathbf{x}_n\|) \end{pmatrix}}_A \underbrace{\begin{pmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_n \end{pmatrix}}_\lambda = \underbrace{\begin{pmatrix} f_1 \\ f_2 \\ \vdots \\ f_n \end{pmatrix}}_F \quad (2)$$

where  $A$  is known as the collocation matrix. A known property of smooth RBF interpolants is that decreasing the shape parameter  $\varepsilon$  will increase the accuracy of the interpolant. Thus, we would like to solve (2) for small  $\varepsilon$ . However, decreasing  $\varepsilon$  will quickly lead to a very ill-conditioned collocation matrix [1]. Solving (2) using a standard linear solve function in your preferred language is known as RBF-Direct; RBF-Direct greatly suffers from the small  $\varepsilon$  ill-conditioning of  $A$ . Instead, we will solve (2) using an indirect method that leverages properties of the GA RBF and some other smooth RBFs.

One of the most important properties of the GA RBF is that it is complex analytic. This analyticity of GA means we can use contour integration in  $\varepsilon$  in a well conditioned regime to stably resolve GA interpolants for small  $\varepsilon$ . The idea is to pick points to evaluate our interpolant

Table 1: Commonly used, smooth radial basis functions.

RBF Name	Radial function $\phi(r)$
Gaussian (GA)	$\phi_\varepsilon(r) = e^{-(\varepsilon r)^2}$
Multiquadric (MQ)	$\phi_\varepsilon(r) = \sqrt{1 + (\varepsilon r)^2}$
Inverse Multiquadric (IMQ)	$\phi_\varepsilon(r) = 1/\sqrt{1 + (\varepsilon r)^2}$
Inverse Quadratic (IQ)	$\phi_\varepsilon(r) = 1/(1 + (\varepsilon r)^2)$
Sech (SH)	$\phi_\varepsilon(r) = \text{sech}(\varepsilon r)$

at and then put the interpolants in an analytic vector function of  $\varepsilon$  denoted by  $\mathbf{s}(\varepsilon)$ . Then, using Cauchy's integral formula, we have

$$\mathbf{s}(\varepsilon) = \frac{1}{2\pi i} \oint_{\Gamma} \frac{\mathbf{s}(z)}{z - \varepsilon} dz$$

where  $\Gamma$  is a counterclockwise oriented circle with center at  $\varepsilon$ . The radius of  $\Gamma$  is ideally picked so that evaluation of the integrand on the contour is at values of  $\varepsilon$  such that the collocation matrix is well-conditioned.

The first method to use contour integration to evaluate RBF interpolants is known as RBF Contour-Padé (RBF-CP). RBF-CP utilizes Padé approximations to construct a rational approximation that we then apply contour integration to. Padé approximations work alright but there are a few pitfalls with stability and finding poles of the approximation [2]. A recent successor to RBF-CP is known as RBF-RA where the RA stands for rational approximation. RBF-RA differs from RBF-CP in that the rational vector approximation is now constructed to be a more general rational function with the poles being easily accessible. RBF-RA appears to work great most of the time but fails in some unexpected cases.

Applying RBF-RA to data that lies on a completely flat surface (i.e. a plane) gives stable and desired results. Similarly, RBF-RA applied to data that lies on a relatively curvy surface also works well. Yet, interestingly, applying RBF-RA to data that lies on a near-flat surface yields erroneous and unusable results! The instability of RBF-RA for near-flat data is unexpected because it works well in both limiting cases (i.e. completely flat and non-flat data). We wish to explore the so called near-flat surface limit of the RBF interpolation problem.

## 2 Flat surface perturbations of the collocation matrix

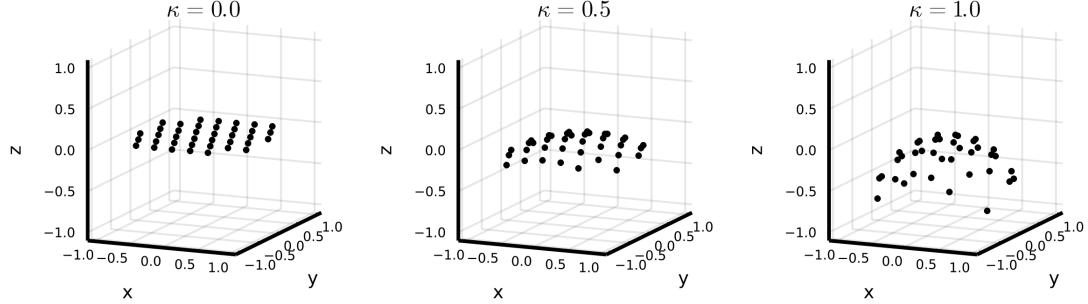
To study the near-flat surface limit, we need to construct some data on a near flat surface. One of the simplest near-flat surfaces is a sphere with small curvature  $\kappa$  and top of the sphere centered at the origin; the sphere described is given by

$$x^2 + y^2 + \left(z + \frac{1}{\kappa}\right)^2 = \frac{1}{\kappa^2}, \quad 0 < \kappa \ll 1. \quad (3)$$

We can then construct data satisfying (3). Suppose we have  $n$  distinct  $x$ - $y$  pairs of data  $\{x_i, y_i\}_i^n$  scattered over the unit disk. Then, define our nodes or data set  $\{\mathbf{x}_i\}_i^n$  satisfying (3) as

$$\mathbf{x}_i = \left\langle x_i, y_i, \sqrt{\frac{1}{\kappa^2} - r_i^2} - \frac{1}{\kappa} \right\rangle \quad (4)$$

Figure 1: Sample of 64 nodes generated using (4) for various curvatures. Note, RBF-RA works well for data in the left and right plots but struggles with data in the middle plot.



where the radii  $r_i$  from the origin are given as

$$r_i = \sqrt{x_i^2 + y_i^2}.$$

It is worth mentioning that the nodes defined in (4) are just points in the  $x$ - $y$  plane projected down onto the sphere described in (3). Example node sets can be seen in Figure 1.

With  $\kappa$  dependent nodes defined, we now seek to understand how the collocation matrix behaves with  $\kappa$ . To construct the collocation matrix, we need to compute all pairwise differences in our node set. By direct computation with (4), we have

$$\|\mathbf{x}_i - \mathbf{x}_j\|_2^2 = (x_i - x_j)^2 + (y_i - y_j)^2 + \left( \sqrt{\frac{1}{\kappa^2} - r_i^2} - \sqrt{\frac{1}{\kappa^2} - r_j^2} \right)^2. \quad (5)$$

Then, each entry of the collocation matrix  $A$  is given by

$$\begin{aligned} A_{ij} &= e^{-(\varepsilon \|\mathbf{x}_i - \mathbf{x}_j\|_2)^2} = e^{-(\varepsilon \|\mathbf{x}_i - \mathbf{x}_j\|_2^2)} \\ &= e^{-(\varepsilon \|\mathbf{x}_i - \mathbf{x}_j\|_2^2)} = e^{-\varepsilon^2 \|\mathbf{x}_i - \mathbf{x}_j\|_2^2} \\ &= e^{-\varepsilon^2 \left( (x_i - x_j)^2 + (y_i - y_j)^2 + \left( \sqrt{\frac{1}{\kappa^2} - r_i^2} - \sqrt{\frac{1}{\kappa^2} - r_j^2} \right)^2 \right)} \\ &= e^{-\varepsilon^2 ((x_i - x_j)^2 + (y_i - y_j)^2)} e^{-\varepsilon^2 \left( \sqrt{\frac{1}{\kappa^2} - r_i^2} - \sqrt{\frac{1}{\kappa^2} - r_j^2} \right)^2}. \end{aligned}$$

Now, Taylor expanding the  $\kappa$  dependent part of  $A_{ij}$  yields

$$\begin{aligned} A_{ij} &= e^{-\varepsilon^2 ((x_i - x_j)^2 + (y_i - y_j)^2)} e^{-\varepsilon^2 \left( \sqrt{\frac{1}{\kappa^2} - r_i^2} - \sqrt{\frac{1}{\kappa^2} - r_j^2} \right)^2} \\ &= e^{-\varepsilon^2 ((x_i - x_j)^2 + (y_i - y_j)^2)} \left( 1 - \frac{1}{4} \varepsilon^2 \kappa^2 (r_i^2 - r_j^2)^2 + \dots \right) \\ &= e^{-\varepsilon^2 ((x_i - x_j)^2 + (y_i - y_j)^2)} - \kappa^2 \frac{1}{4} \varepsilon^2 (r_i^2 - r_j^2)^2 e^{-\varepsilon^2 ((x_i - x_j)^2 + (y_i - y_j)^2)} + \mathcal{O}(\kappa^4). \end{aligned} \quad (6)$$

Before moving on, it is worth discussing the asymptotic ordering of this series for  $A_{ij}$ . It appears that if  $\varepsilon = \mathcal{O}(\kappa^{-1})$  or if  $(r_i^2 - r_j^2) = \mathcal{O}(\kappa^{-1})$ , then asymptotic ordering of our series will be broken. This is true, but recall that  $0 < \kappa \ll 1$  which implies we would need  $\varepsilon \gg 1$  to break asymptotic ordering. However, we are interested in  $\varepsilon < 1$  and so  $\varepsilon = \mathcal{O}(\kappa^{-1})$  will not happen. Similarly,  $0 \leq r_i \leq 1$  which implies  $(r_i^2 - r_j^2)^{-1} \leq 1$  and so we will never have  $(r_i^2 - r_j^2) = \mathcal{O}(\kappa^{-1})$ . So the asymptotic ordering of (6) will not be broken in practice.

Onward, we can use (6) to construct an expansion of the collocation matrix as a whole. This expansion can be written as

$$A = A_0 + \kappa^2 A_1 + \kappa^4 A_2 + \cdots \quad (7)$$

where  $A_0$  is the collection of  $\mathcal{O}(1)$  terms in (6),  $A_1$  are the  $\mathcal{O}(\kappa^2)$  terms,  $A_2$  are the  $\mathcal{O}(\kappa^4)$  terms, and so on. Now, we seek to use the perturbed collocation matrix in (7) to understand the effects of near-flat surface data on the interpolation weights given in (2).

### 3 Asymptotic behavior of interpolant as $\kappa \rightarrow 0$

## References

- [1] B. FORNBERG AND N. FLYER, *A Primer on Radial Basis Functions with Applications to the Geosciences*, vol. 87 of CBMS-NSF regional conference series in applied mathematics, SIAM, 2015.
- [2] G. B. WRIGHT AND B. FORNBERG, *Stable computations with flat radial basis functions using vector-valued rational approximations*, Journal of Computational Physics, 331 (2017), pp. 137–156.

## Background and Problem

Given a small parameter  $\varepsilon$  that we will call the shape parameter and data  $\{\mathbf{x}_i, f_i\}_{i=1}^n$ , we can write out a radial basis function (RBF) based interpolant of the data as

$$s(\mathbf{x}) = \sum_{i=1}^n \lambda_i \phi_\varepsilon(\|\mathbf{x} - \mathbf{x}_i\|_2) \quad (8)$$

where  $\lambda_i$  are interpolant weights and  $\phi_\varepsilon$  is a radial function (in this case we will take  $\phi_\varepsilon(r) = e^{-(\varepsilon r)^2}$ ). In the most direct form, we can find  $\lambda_i$  by solving the system

$$\underbrace{\begin{pmatrix} \phi_\varepsilon(\|\mathbf{x}_1 - \mathbf{x}_1\|) & \phi_\varepsilon(\|\mathbf{x}_1 - \mathbf{x}_2\|) & \cdots & \phi_\varepsilon(\|\mathbf{x}_1 - \mathbf{x}_n\|) \\ \phi_\varepsilon(\|\mathbf{x}_2 - \mathbf{x}_1\|) & \phi_\varepsilon(\|\mathbf{x}_2 - \mathbf{x}_2\|) & \cdots & \phi_\varepsilon(\|\mathbf{x}_2 - \mathbf{x}_n\|) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_\varepsilon(\|\mathbf{x}_n - \mathbf{x}_1\|) & \phi_\varepsilon(\|\mathbf{x}_n - \mathbf{x}_2\|) & \cdots & \phi_\varepsilon(\|\mathbf{x}_n - \mathbf{x}_n\|) \end{pmatrix}}_A \underbrace{\begin{pmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_n \end{pmatrix}}_\lambda = \underbrace{\begin{pmatrix} f_1 \\ f_2 \\ \vdots \\ f_n \end{pmatrix}}_F. \quad (9)$$

Where  $A$  is known as the collocation matrix. In solving this system numerically, it is known that taking  $\varepsilon$  smaller and smaller tends to increase the accuracy of our interpolant. However, if  $\varepsilon$  becomes too small, the matrix above becomes severely ill-conditioned and the interpolant becomes unusable numerically even when it should be well behaved. So, to find  $\lambda_i$  even when  $\varepsilon$  is small, Bengt and some of his previous students have developed multiple methods that stably recover the small  $\varepsilon$  interpolant. The most recent and robust method, so called RBF-RA, is based on rational approximations of our underlying function.

Now with small  $\varepsilon$ , we can apply RBF-RA to data that is on a surface with a large curvature  $\kappa$  or a completely flat surface (i.e.  $\kappa = 0$ ) to get stable, desired results. However, applying RBF-RA to data on a surface that is nearly flat (i.e.  $0 < \kappa \ll 1$ ) can lead to erroneous results.

Herein lies my project, I am seeking to understand how our interpolation weights to (??) behave when we perturb  $\kappa$  near zero. To explore this behavior, I will take nodes over a circle that just touches the origin with curvature  $\kappa$ . Explicitly, we can define our  $n$  nodes as

$$\begin{cases} x_i = -1 + \frac{2}{n-1}i, & i = 1, 2, \dots, n \\ y_i = \sqrt{\frac{1}{\kappa^2} - x_i^2} - \frac{1}{\kappa}, & 0 < \kappa \ll 1 \end{cases}$$

where our nodes are defined as the pairs  $\mathbf{x}_i = (x_i, y_i)$  for  $i = 1, 2, \dots, n$ . Then, if we plug these nodes into the collocation matrix above, we can expand  $A$  in  $\kappa$  about  $\kappa = 0$  as

$$A = A_0 + \kappa A_1 + \kappa^2 A_2 + \cdots. \quad (10)$$

Inserting (10) into (2) yields the perturbed system

$$(A_0 + \kappa A_1 + \cdots)\lambda = F. \quad (11)$$

which is a perfect candidate to be solved with asymptotic methods. Solving and understanding (??) is my main objective for this project but if time permits I would also like to explore the  $\kappa$ -perturbed eigenvalue problem of  $A$ ,

$$(A_0 + \kappa A_1 + \cdots)\mathbf{x} = \lambda \mathbf{x}.$$

It will also be interesting to see if other small curvature node sets behave differently than the proposed node set above (this could also include higher dimensional node sets). In any case, I will need to take a closer look at our collocation matrix to see if asymptotic order is broken for  $\varepsilon$  small enough which might lead to singularly perturbed solutions.