

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 stably and reliably resolve systems with a small shape parameter, alternative methods to direct solvers are used; RBF-RA is one alternative method that tends to work well. 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 undesirable results. Furthermore, we present some ideas surrounding ongoing research involving the problem.

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 ε 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 λ_i are the interpolation weights. This interpolant is incomplete in that we still need to determine each λ_i . A linear system for λ_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 ε will increase the accuracy of the interpolant. Thus, we would like to solve (2) for small ε . However, decreasing ε 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 ε 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 ε , in a well conditioned regime, to

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)	$\phi_\varepsilon(r) = \text{sech}(\varepsilon r)$

stably resolve GA interpolants for small ε . The idea is to pick points to evaluate our interpolant at and then put the interpolants in an analytic vector function of ε 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 Γ is a counterclockwise oriented circle with center at ε . The radius of Γ is ideally picked so that evaluation of the integrand on the contour is at values of ε such that the collocation matrix is well-conditioned.

One of first method to use contour integration to evaluate RBF interpolants is known as RBF Contour-Padé (RBF-CP). RBF-CP utilizes Padé approximates 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 [3]. 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

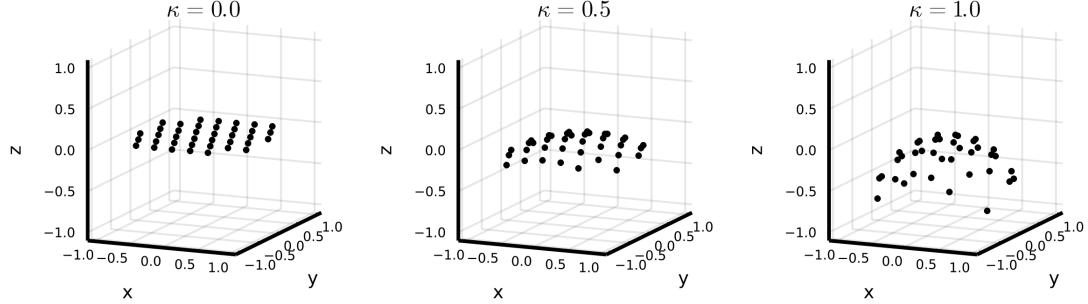
To study the near-flat surface limit, we need to construct some data on a near-flat surface. One of the simplest near-flat surfaces is a sphere with small curvature κ and top of the sphere centered at the origin; the sphere described is given by

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

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, we can define our nodes $\{\mathbf{x}_i\}_i^n$ 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 κ dependent nodes defined, we now seek to understand how the collocation matrix behaves with κ . 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 κ 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 expansion 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 $(r_i^2 - r_j^2) \leq 1$ and so we will never have $(r_i^2 - r_j^2) = \mathcal{O}(\kappa^{-1})$. So in practice, the asymptotic ordering of (6) will not be broken.

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

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

where A_0 is the collection of $\mathcal{O}(1)$ terms in (6), A_1 is the $\mathcal{O}(\kappa^2)$ terms, A_2 is 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 RBF interpolant as $\kappa \rightarrow 0$

The near-flat surface perturbation problem of the interpolation weights can be formed by substituting (7) into (2) to get

$$(A_0 + \kappa^2 A_1 + \kappa^4 A_2 + \mathcal{O}(\kappa^6))\boldsymbol{\lambda} = \mathbf{F}. \quad (8)$$

To find a perturbed solution to (8) we will need some properties of A_0 . Notice that A_0 is just the regular collocation matrix for data on a completely flat surface. Then, for any $\varepsilon \in \mathbb{R}$, $\varepsilon \neq 0$, A_0 is invertible [1]. Furthermore, because A_0 is constructed with flat data, RBF-RA can solve a linear system involving A_0 reliably. With these facts in mind, let's seek a regularly perturbed solution to (8) of the form

$$\boldsymbol{\lambda} = \boldsymbol{\lambda}_0 + \kappa^2 \boldsymbol{\lambda}_1 + \kappa^4 \boldsymbol{\lambda}_2 + \dots \quad (9)$$

Then, plugging this ansatz into (8) and combining ordered terms yields

$$\mathcal{O}(1) : A_0 \boldsymbol{\lambda}_0 = \mathbf{F}$$

which, from the properties of A_0 , implies

$$\boldsymbol{\lambda}_0 = A_0^{-1} \mathbf{F}.$$

Going further,

$$\mathcal{O}(\kappa^2) : A_0 \boldsymbol{\lambda}_1 + A_1 \boldsymbol{\lambda}_0 = 0$$

which implies,

$$\boldsymbol{\lambda}_1 = -A_0^{-1} A_1 \boldsymbol{\lambda}_0.$$

One more time,

$$\mathcal{O}(\kappa^4) : A_0 \boldsymbol{\lambda}_2 + A_1 \boldsymbol{\lambda}_1 + A_2 \boldsymbol{\lambda}_0 = 0$$

which yields

$$\boldsymbol{\lambda}_2 = -A_0^{-1} (A_1 \boldsymbol{\lambda}_1 + A_2 \boldsymbol{\lambda}_0).$$

From here, we can see a pattern forming for $\boldsymbol{\lambda}_n$ as

$$\boldsymbol{\lambda}_n = -A_0^{-1} (A_1 \boldsymbol{\lambda}_{n-1} + A_2 \boldsymbol{\lambda}_{n-2} + \dots + A_n \boldsymbol{\lambda}_0). \quad (10)$$

So, using (10), we have a nice iterative process for solving (9) to arbitrary orders.

The first remark about (10) is that we only need to invert A_0 to get the next term. This is a relief because A_0 is a matrix that we can invert or solve well using RBF-RA. The next remark is that (9) and (10) show that $\boldsymbol{\lambda}$ behaves well in the small κ regime. From here, this confirms our suspicion earlier that RBF-RA is giving erroneous results in the near flat data regime when it ought to give well behaved results.

Luckily, we can use (9) and (10) to create a “patch” for RBF-RA in the near flat surface regime. The “patch” is described as

- (a) Compute a κ or curvature estimate of the given data set using an algorithm of choice,
- (b) Use RBF-RA to compute each λ_n up to desired order using (10),
- (c) Compute λ using (9).

This is a nice and easily stated algorithm that could fix the instability of RBF-RA in the near-flat surface regime, but it is not ideal to perform. Not only does the algorithm increase the cost of solving for interpolation weights in the near-flat limit, but it also doesn't fix the root cause of instability. To better understand why the instability in RBF-RA arises in near-flat data, we will look at the eigenvalues of A .

4 Perturbed eigenvalue problem

Of the many appealing properties of the collocation matrix, a property that will motivate the eigenvalue problem is that A is normal [1]. One consequence of A being normal is that

$$\text{cond}(A) = \frac{|\lambda_{\max}(A)|}{|\lambda_{\min}(A)|}$$

where $\lambda_{\max}(A)$ and $\lambda_{\min}(A)$ are the largest and smallest eigenvalues of A in absolute value respectively. Therefore, because the conditioning of A depends directly on the eigenvalues of A , understanding how the eigenvalues of A change under κ perturbations may help our understanding of the instability of RBF-RA.

Using the perturbed collocation matrix in (7), the perturbed eigenvalue problem is simply

$$(A_0 + \kappa^2 A_1 + \mathcal{O}(\kappa^4))\mathbf{x} = \lambda\mathbf{x}$$

where λ and \mathbf{x} are an eigenvalue/eigenvector pair for the perturbed system. When it comes to finding perturbed solutions to these eigenvalue problems, having some idea of the eigenvalue structure before hand can help immensely. In this case, due to the collocation matrix taking arbitrary data, it is reasonable to assume all of the eigenvalues of A are distinct. In the context of perturbation theory, this means the eigenvalues and eigenvectors will have regularly perturbed solutions [2]. So, let's assume regularly perturbed

$$\begin{aligned}\mathbf{x} &= \mathbf{x}_0 + \kappa^2 \mathbf{x}_1 + \dots \\ \lambda &= \lambda_0 + \kappa^2 \lambda_1 + \dots\end{aligned}\tag{11}$$

Then, at

$$\mathcal{O}(1) : A_0 \mathbf{x}_0 = \lambda_0 \mathbf{x}_0\tag{12}$$

which is the standard flat eigenvalue problem which is solvable in the usual sense. Moving on, at

$$\mathcal{O}(\kappa^2) : A_0 \mathbf{x}_1 + A_1 \mathbf{x}_0 = \lambda_0 \mathbf{x}_1 + \lambda_1 \mathbf{x}_0$$

which implies

$$(A_0 - \lambda_0 I)\mathbf{x}_1 = (\lambda_1 I - A_1)\mathbf{x}_0.$$

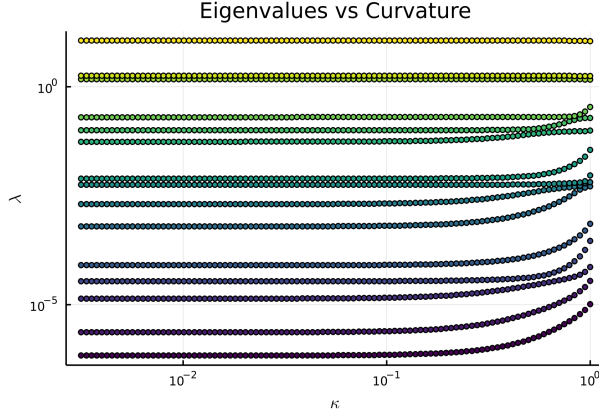
To continue, we invoke solvability (i.e. the RHS needs to be perpendicular to kernel of the adjoint of the LHS). In other words, for

$$\mathbf{w} \in \mathcal{N}(A_0^* - \lambda_0^* I)$$

we require

$$\langle (\lambda_1 I - A_1)\mathbf{x}_0, \mathbf{w} \rangle = 0$$

Figure 2: Numerically found eigenvalues of the collocation matrix with $\varepsilon = 0.5$ and $n = 15$. Note, each color corresponds to a distinct eigenvalue.



which, with the standard vector dot product, becomes

$$\lambda_1 \mathbf{x}_0 \cdot \mathbf{w} - (A_1 \mathbf{x}_0) \cdot \mathbf{w} = 0$$

or

$$\lambda_1 = \frac{(A_1 \mathbf{x}_0) \cdot \mathbf{w}}{\mathbf{x}_0 \cdot \mathbf{w}}. \quad (13)$$

Using this perturbed eigenvalue, we can solve for \mathbf{x}_1 as

$$\mathbf{x}_1 = (A_0 - \lambda_0 I)^{-1}(\lambda_1 I - A_1)x_0 + \alpha x_0$$

where $(A_0 - \lambda_0 I)$ is “invertible” due to solvability and α is any scalar. Going through a similar process with solvability, we can obtain the next order eigenvalue perturbation as

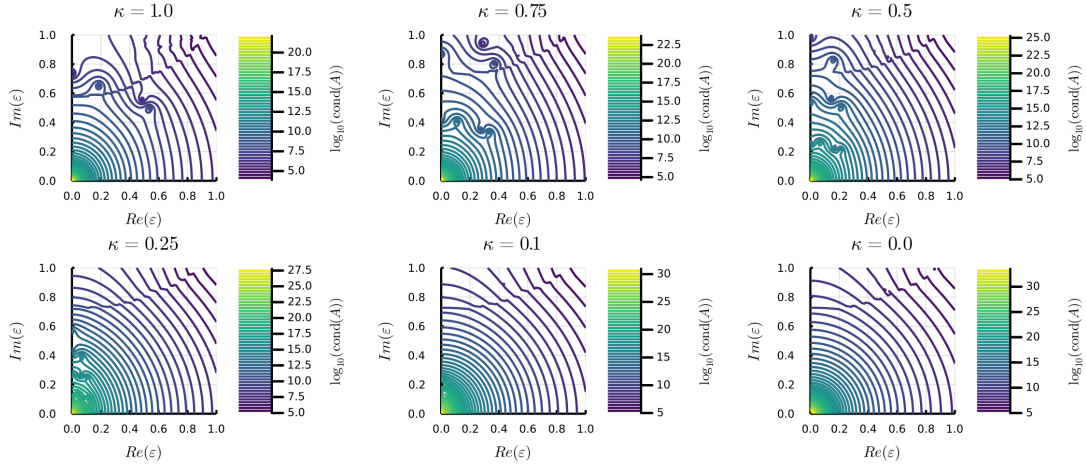
$$\lambda_2 = \frac{(A_2 \mathbf{x}_0 + (A_1 - \lambda_1 I)\mathbf{x}_1) \cdot \mathbf{w}}{x_0 \cdot \mathbf{w}}. \quad (14)$$

Now we can continue this game to any order we desire because we assumed the eigenvalues of A were all distinct. From our perturbed eigenvalue solutions, we can confirm that the eigenvalues behave well in the $\kappa \rightarrow 0$ limit as numerically seen in Figure 2. Furthermore, our regular perturbation solution to the eigenvalues does reflect the shape of the curves seen in Figure 2. However, it is still hard to see how the conditioning of the problem changes in the near-flat regime. We will come back to these eigenvalue results shortly but first we need to pose a more insightful problem.

5 A perturbed, nonlinear root finding problem

Let’s begin by numerically exploring the conditioning of A over the complex plane in ε where RBF-RA performs its contour integration. In Figure 3, we can see singularities, in the condition number of A , move towards the origin as κ decreases into the near-flat regime until κ becomes so small that the singularities collapse into the origin. Due to A being normal, Figure 3 heuristically implies the eigenvalues of A have roots in ε that migrate towards the origin in the $\kappa \rightarrow 0$ limit. So, a natural perturbation problem arises; how do the roots of the eigenvalues of A change under small perturbations in κ .

Figure 3: Conditioning of the collocation matrix with $n = 9$ over the complex plane for various curvatures. A 2D node set over a circle of curvature κ was used.



This perturbation problem is still an active part of my research and is not yet complete but I will highlight my current approach in the remainder of the paper.

To attempt to use asymptotic methods to find perturbed roots of the eigenvalues, we will try to use the heuristic insight from Figure 3 as a basis of solution. That is, we will assume the unperturbed roots of the eigenvalues will be at $\varepsilon = 0$. Now, we know at least one of the eigenvalues will be nonzero at the origin because plugging $\varepsilon = 0$ into A yields a matrix of all ones which has all zero eigenvalues except one which is equal to n ; we will just ignore this nonzero eigenvalue as it shouldn't effect the position of the singularities seen in Figure 3. With these ideas in mind, we can pose the perturbed eigenvalue problem using (11) as

$$\lambda(\varepsilon; \kappa) = \lambda_0(\varepsilon) + \kappa^2 \lambda_1(\varepsilon) + \kappa^4 \lambda_2(\varepsilon) + \dots = 0, \quad \varepsilon = \varepsilon(\kappa)$$

where λ_0, λ_1 , and λ_2 are given by (12), (13), and (14) respectively. This problem looks simple enough as is, but unfortunately each $\lambda_i(\varepsilon)$ is very nonlinear in ε making this system considerably harder to solve. To try to overcome the non-linearity, I will first assume

$$\varepsilon = \kappa^p \varepsilon_1 + \kappa^{2p} \varepsilon_2 + \dots$$

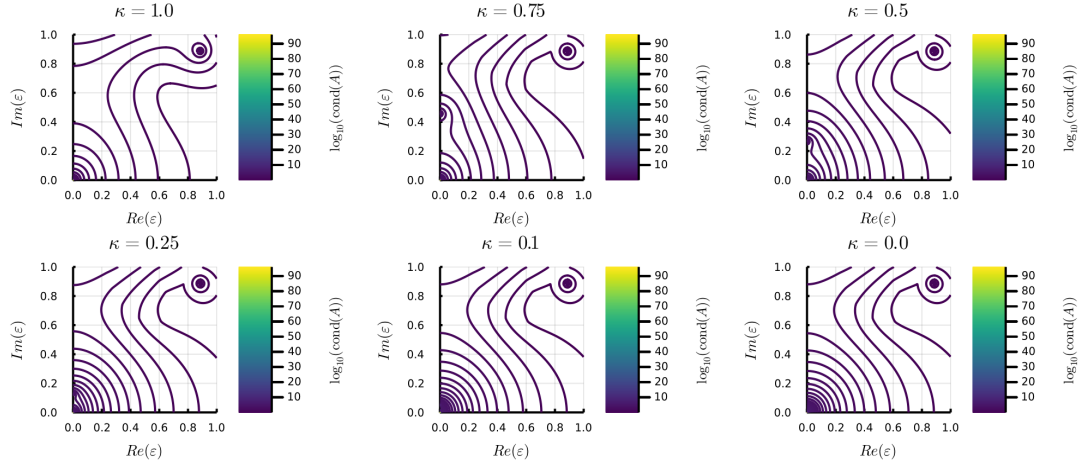
Notice I am not including the $\mathcal{O}(1)$ term in my expansion of ε because by assumption, the unperturbed root is at $\varepsilon = 0$. The next step is to expand each of our λ_i in κ about $\kappa = 0$. And so another hurdle arises, how can we expand λ_i when getting a closed and complete expression for λ_i is very tricky. At this point, it is worth trying our hand at a simplified toy problem.

The toy problem is given by taking the node set

$$\left\{ \left(-1, \sqrt{\frac{1}{\kappa^2 - 1}} - \frac{1}{\kappa} \right), (0, 0), \left(1, \sqrt{\frac{1}{\kappa^2 - 1}} - \frac{1}{\kappa} \right) \right\}.$$

This is only a system of three, equispaced nodes but it still proves to have some complex behavior. With a little help from Mathematica, the perturbed eigenvalues are given by

$$\begin{aligned} \lambda_a &= e^{-4\varepsilon^2} (e^{4\varepsilon^2} - 1) \\ \lambda_b &= \frac{1}{2} e^{-5\varepsilon^2} \left(-\sqrt{e^{2\varepsilon^2} (8e^{6\varepsilon^2} + 1)} + e^{\varepsilon^2} + 2e^{5\varepsilon^2} \right) + \kappa^2 \frac{e^{\varepsilon^2} \varepsilon^2 \sqrt{e^{2\varepsilon^2} (8e^{6\varepsilon^2} + 1)}}{8e^{6\varepsilon^2} + 1} \\ \lambda_c &= \frac{1}{2} e^{-5\varepsilon^2} \left(\sqrt{e^{2\varepsilon^2} (8e^{6\varepsilon^2} + 1)} + e^{\varepsilon^2} + 2e^{5\varepsilon^2} \right) - \kappa^2 \frac{e^{\varepsilon^2} \varepsilon^2 \sqrt{e^{2\varepsilon^2} (8e^{6\varepsilon^2} + 1)}}{8e^{6\varepsilon^2} + 1}. \end{aligned}$$

Figure 4: Conditioning of A over the complex plane for the $n = 3$ toy problem.

For the toy problem, λ_c is the eigenvector that does not have a zero at the origin; so we will ignore λ_c . Furthermore, λ_a does not depend on κ and so the the root at the origin will remain unperturbed. That leaves us with analyzing $\lambda_b(\varepsilon_b)$. With the help of Mathematica, we are able to compute the nontrivial perturbed root as

$$\varepsilon_b(\kappa) = \frac{i}{2}\kappa + \frac{i}{8}\kappa^3 + \mathcal{O}(\kappa^5).$$

Now, we can directly compare the locations of the singularities in [Figure 4](#) with the perturbed λ_b root. The roots of λ_b corresponding to the curvatures given in [Figure 4](#) can be seen below.

κ	$\varepsilon_b(\kappa)$
1.0	$0.6250i$
0.75	$0.4277i$
0.5	$0.2656i$
0.25	$0.1270i$
0.1	$0.0501i$
0.0	0.0000

In the eyeball norm, the roots from the table (aside from $\kappa = 1$) line up perfectly with the singularities seen in [Figure 4](#)! So, our root problem as posed appears to be working well. The next step is in completing the non-toy root perturbation problem.

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] E. J. HINCH, *Perturbation Methods*, Cambridge Texts in Applied Mathematics, Cambridge University Press, 1991.
- [3] 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.