

# Choosing Basis Functions and Shape Parameters for Radial Basis Function Methods

Michael Mongillo

October 25, 2011

## Abstract

Radial basis function (RBF) methods have broad applications in numerical analysis and statistics. They have found uses in the numerical solution of PDEs, data mining, machine learning, and kriging methods in statistics. This work examines the use of radial basis functions in scattered data approximation. In particular, the experiments in this paper test the properties of shape parameters in RBF methods, as well as methods for finding an optimal shape parameter. Locating an optimal shape parameter is a difficult problem and a topic of current research. Some experiments also consider whether the same methods can be applied to the more general problem of selecting basis functions.

## 1 Introduction

Radial basis function (RBF) methods are an active area of mathematical research. The initial motivation for RBF methods came from geodesy, mapping, and meteorology. They have since found applications in other areas, such as the numerical solution of PDEs, machine learning, and statistics. In the 1970s, Rolland Hardy suggested what he called the multiquadric method for applications in cartography because he was not satisfied with the results of polynomial interpolation. His new method for fitting data could handle unevenly distributed data sites with greater consistency than previous methods [Har90]. The method of RBF interpolation used in this paper is a generalization of Hardy's multiquadric and inverse multiquadric methods.

Many RBFs are defined by a constant called the shape parameter. The choice of basis function and shape parameter have a significant impact on the accuracy of an RBF method. This paper investigates methods for selecting basis functions and their shape parameters. Leave one out cross validation, generalized cross validation, and a maximum likelihood estimator are used to select shape parameters in Section 2. In Section 3, those methods are applied to the problem of selecting basis functions.

### 1.1 Interpolation of Scattered Data

A radial function is a function that is radially symmetric around some point  $\mathbf{x}_c$  called the function's center. For a kernel  $K : \mathbb{R}^s \times \mathbb{R}^s \rightarrow \mathbb{R}$  with input vectors  $\mathbf{x} = [x_1, x_2, \dots, x_s]^T$  and  $\mathbf{x}_c = [(x_c)_1, (x_c)_2, \dots, (x_c)_s]^T$ ,  $K$  is a radial function if it can be defined as  $K(\mathbf{x}, \mathbf{x}_c) = \kappa(r)$ , where

$r = \|\mathbf{x} - \mathbf{x}_c\|_2$  is the Euclidean distance between the points  $\mathbf{x}$  and  $\mathbf{x}_c$ . To use  $\kappa(r)$  as a basis function in an RBF method, the center  $\mathbf{x}_c$  is set to a constant point and  $\mathbf{x}$  is taken to be the input variable. Note that the univariate function  $\kappa$  is independent from the number of input dimensions  $s$ . As a result, methods that use radial functions for their basis functions are, in principle, easily adapted to solve problems in higher dimensions.

In a scattered data approximation problem, we are given a set of  $n$  distinct data points  $\mathbb{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$  in  $\mathbb{R}^s$  and a corresponding set of  $n$  values  $y_1, y_2, \dots, y_n$  sampled from an unknown function  $f$  such that  $y_i = f(\mathbf{x}_i)$ . We can then choose a radial function  $\kappa$  and a set of centers,  $\{\mathbf{x}_{c_1}, \mathbf{x}_{c_2}, \dots, \mathbf{x}_{c_m}\}$  for some  $m \in \mathbb{N}$ , to obtain a basis  $\{\kappa(\|\cdot - \mathbf{x}_{c_1}\|), \kappa(\|\cdot - \mathbf{x}_{c_2}\|), \dots, \kappa(\|\cdot - \mathbf{x}_{c_m}\|)\}$ . This basis can then be used to construct an approximation  $\tilde{f}$  of the function  $f$ .

For example, the left side of Figure 1 shows the graph of a function  $f : \mathbb{R}^2 \rightarrow \mathbb{R}$ . The right side of Figure 1 shows a set of 8 data points sampled from that function at scattered locations in  $[1, 2] \times [1, 2]$ .

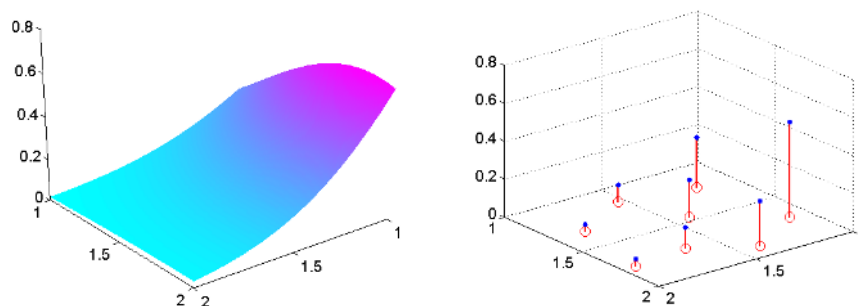


Figure 1: Sample data from test function  $f(\mathbf{x}) = x_1 \cdot \sin^2(x_1) \cdot e^{-x_2^2}$  on the domain  $[1, 2] \times [1, 2]$ .

One option is to center an RBF on each data site. In that case, the approximation will be constructed from  $n$  radial basis functions, and there will be one basis function with  $\mathbf{x}_c = \mathbf{x}_i$  for each  $i = 1, 2, \dots, n$ . The approximation  $\tilde{f}$  is then constructed from a linear combination of those  $n$  RBFs, so that

$$\tilde{f}(\mathbf{x}) = \sum_{j=1}^n c_j \kappa(\|\mathbf{x} - \mathbf{x}_j\|_2) \quad (1)$$

with constant coefficients  $c_j$ . In an interpolation, the constants  $c_j$  are determined by ensuring that the approximation will exactly match the given data at the data points. This is accomplished by enforcing  $\tilde{f}(\mathbf{x}_i) = y_i$ , which produces the system of linear equations

$$\mathbf{K}\mathbf{c} = \mathbf{y}, \quad (2)$$

$$\text{where } \mathbf{K} = \begin{bmatrix} \kappa(\|\mathbf{x}_1 - \mathbf{x}_1\|_2) & \kappa(\|\mathbf{x}_1 - \mathbf{x}_2\|_2) & \cdots & \kappa(\|\mathbf{x}_1 - \mathbf{x}_n\|_2) \\ \kappa(\|\mathbf{x}_2 - \mathbf{x}_1\|_2) & \kappa(\|\mathbf{x}_2 - \mathbf{x}_2\|_2) & \cdots & \kappa(\|\mathbf{x}_2 - \mathbf{x}_n\|_2) \\ \vdots & \vdots & \ddots & \vdots \\ \kappa(\|\mathbf{x}_n - \mathbf{x}_1\|_2) & \kappa(\|\mathbf{x}_n - \mathbf{x}_2\|_2) & \cdots & \kappa(\|\mathbf{x}_n - \mathbf{x}_n\|_2) \end{bmatrix},$$

$$\mathbf{c} = [c_1, c_2, \dots, c_n]^T, \text{ and } \mathbf{y} = [y_1, y_2, \dots, y_n]^T.$$

The coefficients  $c_j$  are then found by solving the linear system  $\mathbf{K}\mathbf{c} = \mathbf{y}$ . With the coefficients defined, the only choice remaining is how to define  $\kappa$ . Choosing an appropriate RBF is an important decision, and there are numerous choices available.

## 1.2 Basis Functions Used

The choice of basis functions will determine which methods are available for solving (2), and whether such a solution even exists. If the interpolation matrix  $\mathbf{K}$  is symmetric positive definite, then the linear system has a unique solution. One property of a positive definite matrix is that all of its eigenvalues are positive. Therefore, a positive definite matrix is invertible. Additionally, numerical methods such as Cholesky factorization can be used to solve a symmetric positive definite linear system more efficiently than methods designed for a general linear system [TB97, p. 172].

If one basis function is centered on each data site (as shown above), then one immediate result of using radial basis functions is that the interpolation matrix  $\mathbf{K}$  is symmetric. This is clear from (2) because  $\|\mathbf{x}_i - \mathbf{x}_j\|_2 = \|\mathbf{x}_j - \mathbf{x}_i\|_2$  means that  $K_{ij} = K_{ji}$ .

The interpolation matrix  $\mathbf{K}$  is positive definite if

$$\mathbf{t}^T \mathbf{K} \mathbf{t} > 0 \tag{3}$$

for every non-zero vector  $\mathbf{t} = [t_1, t_2, \dots, t_n]^T \in \mathbb{R}^n$ . Similarly, a symmetric kernel  $K : \mathbb{R}^s \times \mathbb{R}^s \rightarrow \mathbb{R}$  is a positive definite kernel if

$$\sum_{i=1}^m \sum_{j=1}^m K(\mathbf{x}_i, \mathbf{x}_j) t_i \bar{t}_j > 0 \tag{4}$$

for any distinct points  $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_m$  in  $\mathbb{R}^s$  and every non-zero vector  $\mathbf{t} \in \mathbb{C}^m, \forall m \in \mathbb{N}$ . In an RBF interpolation with one basis function centered on each data site, (3) is a result of (4). Therefore, if  $K$  is a positive definite kernel, then the matrix  $\mathbf{K}$  defined in (2) is a positive definite matrix [Fas07, p. 37].

If the RBF chosen is a reproducing kernel, then the interpolant produced by an RBF interpolation also exhibits other useful properties. These include a series of optimality results which can be found in [Fas07, p. 159]. Also, reproducing kernels are always at least positive semi-definite functions [Aro50], meaning that

$$\sum_{i=1}^m \sum_{j=1}^m K(\mathbf{x}_i, \mathbf{x}_j) t_i \bar{t}_j \geq 0$$

for any distinct points  $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_m$  and any constants  $\mathbf{t} = [t_1, t_2, \dots, t_m]^T \in \mathbb{C}^m, \forall m \in \mathbb{N}$ . If it can be shown that the inequality above is an equality only when  $\mathbf{t} = \mathbf{0}$ , then the reproducing kernel in question is also a positive definite function.

Table 1 lists some common RBFs which are used in the following experiments. As before, the variable  $r$  stands for  $\|\mathbf{x} - \mathbf{x}_c\|_2$ . With the exception of the multiquadric, all of the RBFs listed are positive definite reproducing kernels. The multiquadric is not positive definite or a reproducing kernel, but it is conditionally negative definite. In an RBF interpolation, it gives rise to

Name	Definition
Gaussian	$\kappa(r) = e^{-(\varepsilon r)^2}$
Multiquadric	$\kappa(r) = \sqrt{1 + (\varepsilon r)^2}$
Inverse Multiquadric	$\kappa(r) = \frac{1}{\sqrt{1 + (\varepsilon r)^2}}$
$C^0$ Matérn	$\kappa(r) = e^{-\varepsilon r}$
$C^2$ Matérn	$\kappa(r) = e^{-\varepsilon r} \cdot (1 + \varepsilon r)$
$C^4$ Matérn	$\kappa(r) = e^{-\varepsilon r} \cdot (3 + 3\varepsilon r + (\varepsilon r)^2)$

Table 1: Some common radial basis functions.

a matrix of size  $n$  with one positive eigenvalue, and  $n - 1$  negative eigenvalues. Therefore, an RBF interpolation which uses multiquadric basis functions has a unique solution, but that solution cannot necessarily be computed using methods specialized for solving positive definite systems. The multiquadric and inverse multiquadric are both part of the generalized multiquadric family of RBFs defined by  $\kappa(r) = (1 + (\varepsilon r)^2)^\beta$ . A detailed investigation of generalized multiquadrics can be found in [Che09]. While this paper focuses primarily on the shape parameter  $\varepsilon$ , [Che09] investigates the parameter  $\beta$  of the generalized multiquadric.

### 1.3 Shape Parameters

Many RBFs, including all of the ones studied here, have a variable  $\varepsilon$  in their definitions. This variable  $\varepsilon$  is called the shape parameter. For the RBF definitions listed in Table 1, a smaller shape parameter corresponds to a “flatter” or “wider” basis function. The limit as  $\varepsilon \rightarrow 0$  is often referred to as the “flat” limit, because  $\varepsilon = 0$  corresponds to a constant basis function.

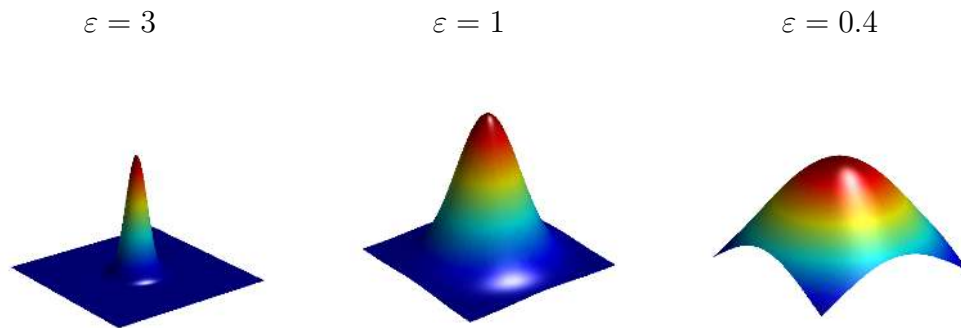


Figure 2: Gaussian RBFs with different shape parameters plotted on the same domain.

Figure 2 shows three Gaussian RBFs with different shape parameters. The RBFs are graphed with two-dimensional inputs on the same domain. As we will see below, changing the shape parameter of an RBF alters the interpolant, and can have a significant impact on the accuracy of the approximation.

## 2 Choosing a Shape Parameter

Different shape parameters correspond to different approximations resulting from RBF interpolation. Finding the shape parameter that will produce the most accurate approximation is a topic of current research. The experiments in this section investigate methods for optimizing shape parameters with respect to error in an RBF interpolation.

For an example, Figure 1 shows a graph of the function  $f(\mathbf{x}) = x_1 \cdot \sin^2(x_1) \cdot e^{-x_2^2}$  and 8 data sites sampled from that function. Figure 2 shows three Gaussian RBFs with different shape parameters. Figure 3 contains graphs of three different interpolants resulting from an RBF interpolation using the data from Figure 1 and translates of the basis functions from Figure 2. Each interpolant is defined by (1) with the coefficients  $c_j$  computed using (2). The interpolants are different because the RBFs used to construct them have a different shape parameter, as noted above each graph of Figure 3. The function  $f$  is graphed in green alongside each interpolant for reference.

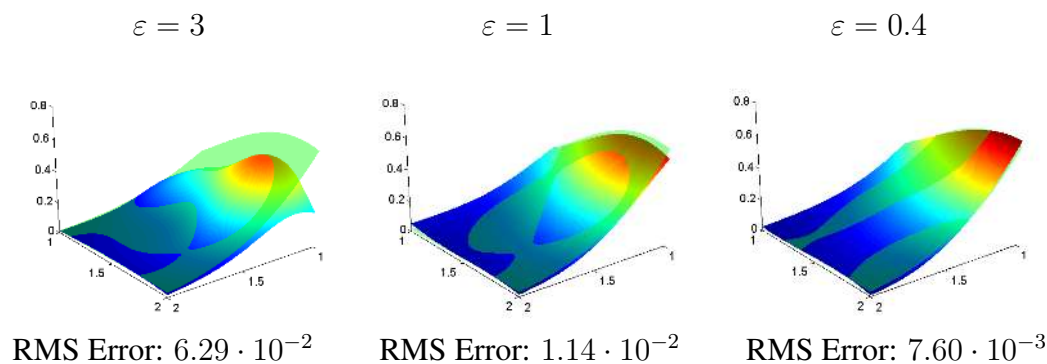


Figure 3: RBF interpolants using different shape parameters (approximated function shown in green).

The term “RMS error” listed underneath each graph is the “root mean squared error” of the approximation. The RMS error is the square root of the mean squared error, or

$$\sqrt{\frac{1}{P} \sum_{j=1}^P [\tilde{f}(\mathbf{p}_j) - f(\mathbf{p}_j)]^2},$$

where  $P$  is the number of points being used to compute error and the  $\mathbf{p}_j$ 's are those points. In this case, the RMS error was computed using 64 points distributed in a uniform grid on  $[1, 2] \times [1, 2]$ . The RMS error will also be used to evaluate error in the following experiments.

Even from visual inspection, it can be seen that changing the shape parameter used in the basis functions has an impact on the accuracy of the resulting approximation. Of the three interpolants shown in Figure 3, the one with the smallest shape parameter appears to be the most accurate. To get a more complete picture of the interpolant's behavior with respect to the shape parameter, Figure 4 shows the RMS error plotted against the shape parameter of the Gaussian RBFs used to construct the interpolant.

The graph in Figure 4 indicates that the lowest RMS error was computed with a shape parameter between 0.4 and 0.5. It has been observed that accuracy improves with most decreases in  $\varepsilon$ .

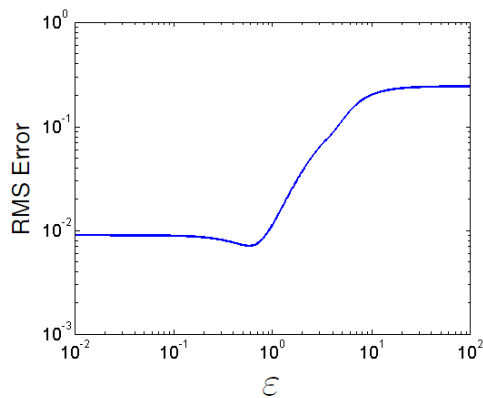


Figure 4: Error for example problem.

However, the smallest error is not found at  $\varepsilon = 0$ . A minimum error is often found at some “small” value of  $\varepsilon$ , as is the case in this example. While this is typical, results change based on  $f$ , the basis functions chosen, and the data set used to construct the approximation.

In addition to accuracy, a prominent concern in numerical computations is stability. As a follow-up to the first example, the following example illustrates the relation between the shape parameter and numerical instability. This example attempts to approximate the same function (from Figure 1) using Gaussian RBFs, but in this case we use  $n = 120$  randomly distributed data sites to construct the approximation. This means that the interpolant will contain 120 basis functions, with one centered on each of the 120 data sites. As a result, the interpolation matrix  $K$  will be a  $120 \times 120$  matrix. Figure 5 shows the RMS error from a follow-up to the example above. The graph in Figure 5 appears unstable for most values of  $\varepsilon$  shown. Specifically, it appears that the problem is unstable for approximately  $\varepsilon < 3$ .

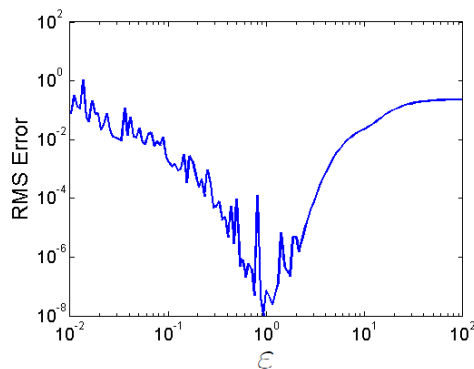


Figure 5: Error for example problem with 120 randomly distributed data sites.

The problem of increased instability as  $\varepsilon$  decreases is well known. This is particularly significant because highest accuracy is often found at some “small” shape parameter, which may be in the unstable region. This conflict between accuracy and stability is sometimes referred to as the “trade-off principle” [Fas07, p. 138]. This “trade-off principle” is linked to the choice of ba-

sis. Recently, methods have been developed to stably compute interpolants with small parameters by using an alternate basis, such as Contour-Padé and RBF-QR, e.g. [FW04, Pir07]. However, those methods are not investigated in this paper. The following experiments use the “RBF-Direct” method, which is the method described in Section 1.

## 2.1 Predictors

As seen above, varying the shape parameter of the basis functions can have a significant impact on the accuracy of an RBF interpolation. A method for finding a good shape parameter is needed. One strategy is to minimize a “predictor function” which mimics the interpolation error. Several functions have been derived for that purpose. Three such functions will be used in the following experiments.

### Leave One Out Cross Validation

Cross validation attempts to test the accuracy of a method by separating the available data into two or more parts. One part of the data set is used to construct an approximation, and error is measured using a different part of the data set. Leave one out cross validation (LOOCV) uses  $n - 1$  points from the data set to construct an approximation, then checks that approximation’s error at the remaining data site. The procedure is repeated leaving out each data site once, and the resulting set of errors are used to estimate the method’s relative accuracy [Wah90].

For this paper, the sum of the squares of the errors is used for the LOOCV so that

$$LOOCV(\varepsilon) = \sum_{i=1}^n |\tilde{f}_i(\mathbf{x}_i) - y_i|^2,$$

where  $\tilde{f}_i$  is the interpolant formed when excluding the point  $\mathbf{x}_i$  and using basis functions with shape parameter  $\varepsilon$ . However, computing the LOOCV by solving  $n$  linear systems, each with a dense matrix, would require  $O(n^4)$  operations. The computational complexity of solving a linear system with a dense matrix is only  $O(n^3)$ , therefore incorporating LOOCV as stated above into an RBF interpolation algorithm would increase the computational complexity of the algorithm using it.

In [Rip99], Shmuel Rippa showed that

$$\tilde{f}_i(\mathbf{x}_i) - y_i = \frac{c_i}{K_{ii}^{-1}},$$

where  $c_i$  is the  $i$ ’th coefficient of the interpolant  $\tilde{f}$  constructed using the full data set and  $K_{ii}^{-1}$  is the  $i$ ’th diagonal element of  $K^{-1}$ . The vector  $\mathbf{c}$  can be found by solving  $K\mathbf{c} = \mathbf{y}$ , and the diagonal elements of  $K^{-1}$  can be computed using a matrix factorization. As a result, computing the LOOCV using Rippa’s formula has computational complexity  $O(n^3)$ . For the following experiments, LOOCV will be computed using the formula

$$LOOCV(\varepsilon) = \sum_{i=1}^n \left[ \frac{c_i}{K_{ii}^{-1}} \right]^2.$$

## Generalized Cross Validation

Generalized cross validation (GCV) is a variation of leave one out cross validation which replaces the diagonal elements of  $\mathbf{K}^{-1}$  with their average. GCV is similar to LOOCV, but has some invariance properties which LOOCV lacks. As with LOOCV, the computational complexity of GCV is  $O(n^3)$ . Further information on generalized cross validation can be found in [Wah90, p. 52]. In the following experiments, GCV is calculated using the equation

$$GCV(\varepsilon) = \frac{\sum_{i=1}^n c_i^2}{\left[\frac{1}{n} \sum_{i=1}^n \mathbf{K}_{ii}^{-1}\right]^2}.$$

## Maximum Likelihood Estimator

Another method for predicting which shape parameter  $\varepsilon$  will minimize the error of an RBF interpolation is to use a maximum likelihood estimator (MLE). Assuming that  $f$  is a Gaussian process, maximizing the likelihood function is equivalent to minimizing

$$\mathbf{y}^T \mathbf{K}^{-1} \mathbf{y} \cdot [\det(\mathbf{K})]^{\frac{1}{n}}$$

[Ste99, p. 169]. This formula rapidly approaches 0 as  $n$  increases, causing numerical error. The resulting numerical error can be prevented by taking the logarithm of the function in its computation and applying the identity

$$\det(\mathbf{K}) = \prod_{i=1}^n \lambda_i(\mathbf{K}),$$

where  $\lambda_i(\mathbf{K})$  is the  $i$ 'th eigenvalue of  $\mathbf{K}$ . Additionally, computing  $\mathbf{K}^{-1}$  can be avoided because  $\mathbf{K}^{-1} \mathbf{y} = \mathbf{c}$ .

For the following experiments, the function

$$MLE(\varepsilon) = \log[\mathbf{y}^T \mathbf{c}] + \frac{1}{n} \sum_{i=1}^n \log[\lambda_i(\mathbf{K})]$$

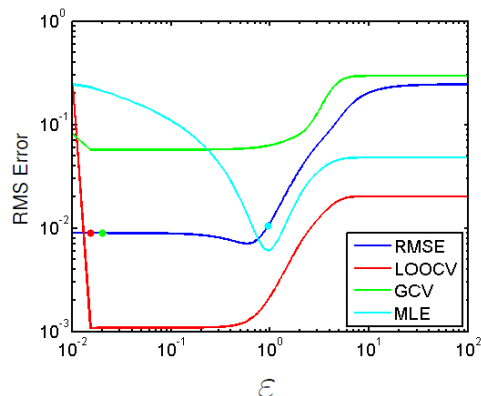
will be minimized to find the maximum likelihood estimate for the value of  $\varepsilon$  which minimizes the error of the interpolant. Computing the coefficients  $\mathbf{c}$  and computing the eigenvalues of a matrix each require  $O(n^3)$  operations. Therefore, computing the value of this function for a single shape parameter  $\varepsilon$  has computational complexity  $O(n^3)$ .

## 2.2 Experiment 1: Selecting $\varepsilon$ Using Predictors

This set of experiments investigates the use of LOOCV, GCV, and MLE as methods for locating a shape parameter  $\varepsilon$  for use in an RBF interpolation. In addition to the value of  $\varepsilon$  located by each method, it is also important that the value can be found efficiently. Since the predictors must be minimized to locate the appropriate shape parameter, issues such as multiple local minima are a potential concern. These experiments analyze the LOOCV, GCV, and MLE by plotting them alongside the RMS error for a given problem.



In each experiment, a scattered data approximation problem is repeatedly solved using RBF interpolation with differing shape parameters. For each interpolant, an RMS error is computed using  $n^s$  evaluation points, where  $n$  is the number of data sites used to construct the approximation and  $s$  is the number of input variables for  $f$ . These evaluation points are uniformly distributed on the domain given for the problem. Values for LOOCV, GCV, and MLE are computed using the formulas defined in Section 2.1. The results are then plotted with the predictor functions scaled and translated vertically to fit alongside the graph of RMS error. To make the minima easier to identify, a point of the same color as each predictor's curve is placed on the RMS error curve at the value of  $\varepsilon$  which minimizes the associated predictor function. Each graph is also accompanied by a table listing the  $\varepsilon$  at which each curve is minimized and the RMS error of an interpolant using that shape parameter.

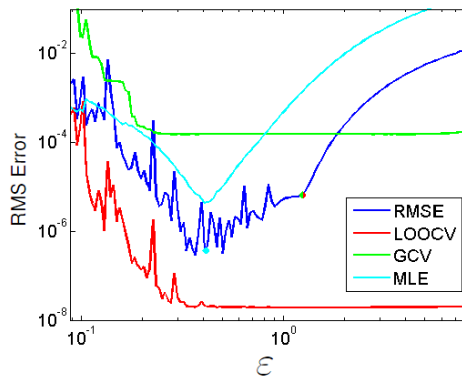


Curve	$\varepsilon$ When Minimized	RMS Error
RMS Error	$5.74 \cdot 10^{-1}$	$9.402 \cdot 10^{-3}$
LOOCV	$1.36 \cdot 10^{-2}$	$1.243 \cdot 10^{-2}$
GCV	$2.06 \cdot 10^{-2}$	$1.242 \cdot 10^{-2}$
MLE	$9.33 \cdot 10^{-1}$	$1.391 \cdot 10^{-2}$

Figure 6: Error and predictors for RBF interpolation of  $f(\mathbf{x}) = x_1 \cdot \sin^2(x_1) \cdot e^{-x_2^2}$  on  $[1, 2] \times [1, 2]$  using Gaussian RBFs with  $n = 8$  data sites.

Figure 6 graphs the RMS error and relative values of predictors when attempting to approximate the function  $f(\mathbf{x}) = x_1 \cdot \sin^2(x_1) \cdot e^{-x_2^2}$  on the domain  $[1, 2] \times [1, 2]$  using  $n = 8$  data sites distributed as depicted in Figure 1. The RMS error was computed using 64 uniformly distributed points in  $[1, 2] \times [1, 2]$ . In this experiment, the MLE locates the value of  $\varepsilon$  closest to the value of  $\varepsilon$  that minimizes the error. However, the LOOCV and GCV both locate a value of  $\varepsilon$  which produces smaller RMS error than the value located by the MLE. Each predictor appears to have a single local minimum in this problem.

Figure 7 graphs the RMS error and relative values of predictors for a variation on the previous problem using inverse multiquadric RBFs with  $n = 120$  randomly distributed data sites on the domain  $[1, 2] \times [1, 2]$ . Unlike the previous experiment, the results in Figure 7 show instability. The MLE notably appears to remain smooth through much of the instability. The LOOCV and GCV begin to increase as soon as the instability begins, with their minima being found near the edge of



Curve	$\varepsilon$ When Minimized	RMS Error
RMS Error	$3.79 \cdot 10^{-1}$	$2.887 \cdot 10^{-7}$
LOOCV	1.23	$6.473 \cdot 10^{-6}$
GCV	1.23	$6.473 \cdot 10^{-6}$
MLE	$4.06 \cdot 10^{-1}$	$3.958 \cdot 10^{-7}$

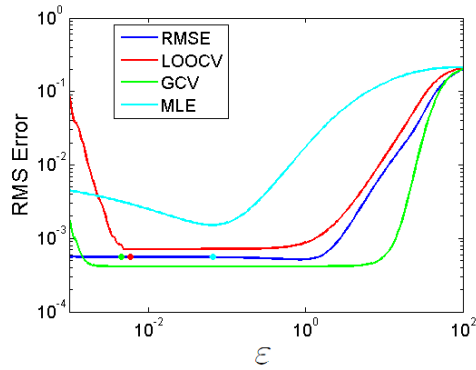
Figure 7: Error and predictors from RBF interpolation of  $f(\mathbf{x}) = x_1 \cdot \sin^2(x_1) \cdot e^{-x_2^2}$  on  $[1, 2] \times [1, 2]$  using inverse multiquadric RBFs with 120 data sites.

the unstable region. They also become unstable as  $\varepsilon$  decreases, causing them to have numerous local minima. While the MLE remains stable through much of the instability, it has an additional minimum near  $\varepsilon = 0.01$  at the left side of the graph. This local minimum corresponds to an RMS error larger than  $10^{-3}$ , so even using the MLE to select a shape parameter could encounter difficulties.

Changing basis functions can have a significant impact on both the accuracy and stability of the interpolation. Figure 8 shows the RMS error and relative values of predictors for the same problem as in Figure 7, but in this case  $C^2$  Matérn basis functions are used. The result is a much less accurate, but more stable interpolation. Using  $C^2$  Matérn basis functions the errors using shape parameters found by the predictors are on the order  $10^{-4}$ , compared to  $10^{-6}$  and  $10^{-8}$  when using inverse multiquadric basis functions. However, Figure 7 also shows that minimizing the predictors may be difficult when solving this problem using inverse multiquadric basis functions, especially if LOOCV or GCV is used to choose a shape parameter. This is particularly important because many of the local minima for the predictors in Figure 7 correspond to RMS errors above  $10^{-4}$ . Although the smallest RMS errors found when using  $C^2$  Matérn basis functions are not as small, each of the predictors has a single minimum with error of order  $10^{-4}$ . As a result, error on the order of  $10^{-4}$  will be easily found when using  $C^2$  Matérn basis functions in this approximation.

### 2.3 Experiment 2: Locating Instability using Error Bounds

Given the importance of stability to the performance of shape parameter optimization methods, it would be useful to have an efficient method for predicting instability. Some attention has been given to developing bounds on the condition number of the matrix  $K$ . The following bound for a



Curve	$\varepsilon$ When Minimized	RMS Error
RMS Error	$8.12 \cdot 10^{-1}$	$5.151 \cdot 10^{-4}$
LOOCV	$5.92 \cdot 10^{-3}$	$5.598 \cdot 10^{-4}$
GCV	$4.49 \cdot 10^{-3}$	$5.599 \cdot 10^{-4}$
MLE	$6.25 \cdot 10^{-2}$	$5.543 \cdot 10^{-4}$

Figure 8: Error and predictors from RBF interpolation of  $f(\mathbf{x}) = x_1 \cdot \sin^2(x_1) \cdot e^{-x_2^2}$  on  $[1, 2] \times [1, 2]$  using  $C^2$  Matérn RBFs with 120 data sites.

Gaussian kernel is recounted from [Fas07, p. 135].

For a positive definite matrix,

$$\text{Cond}(\mathbf{K}) = \frac{\lambda_{max}}{\lambda_{min}},$$

where  $\lambda_{max}$  and  $\lambda_{min}$  are the largest and smallest eigenvalues of  $\mathbf{K}$ . If the RBF  $\kappa$  is positive definite, then  $\lambda_{max} \leq n \cdot \kappa(0)$ . For a Gaussian kernel,

$$\lambda_{min} \geq C_s (\sqrt{2\varepsilon})^{-s} e^{-40.71s^2/(q_x\varepsilon)^2} q_x^{-s},$$

where  $q_x = \frac{1}{2} \min_{i \neq j} (\|\mathbf{x}_i - \mathbf{x}_j\|_2)$  and  $s$  is the dimension of  $\mathbf{x}$ . The constant  $C_s$  is given by

$$C_s = \frac{1}{2\Gamma(\frac{s+2}{2})} \left( \frac{M_s}{\sqrt{8}} \right)^s, \text{ where } M_s = 12 \left( \frac{\pi\Gamma^2(\frac{s+2}{2})}{9} \right)^{\frac{1}{s+1}}$$

and  $\Gamma(\cdot)$  is the Gamma function.

For a Gaussian kernel in one dimension, this simplifies to

$$\text{Cond}(\mathbf{K}) \leq \text{CondBound}(\mathbf{K}) = \frac{2\varepsilon n q_x \cdot e^{40.71/(q_x\varepsilon)^2}}{\sqrt{\pi}}.$$

To investigate the potential usefulness of this bound in an algorithm, the following graphs plot the ratio  $\frac{\text{CondBound}(\mathbf{K})}{\text{Cond}(\mathbf{K})}$  for varying values of  $\varepsilon$  with  $n = 10$  and varying values of  $n$  with  $\varepsilon = 10$ . For this test, we use a uniform point distribution on the domain  $[0, 10]$ .

These graphs show that in this case

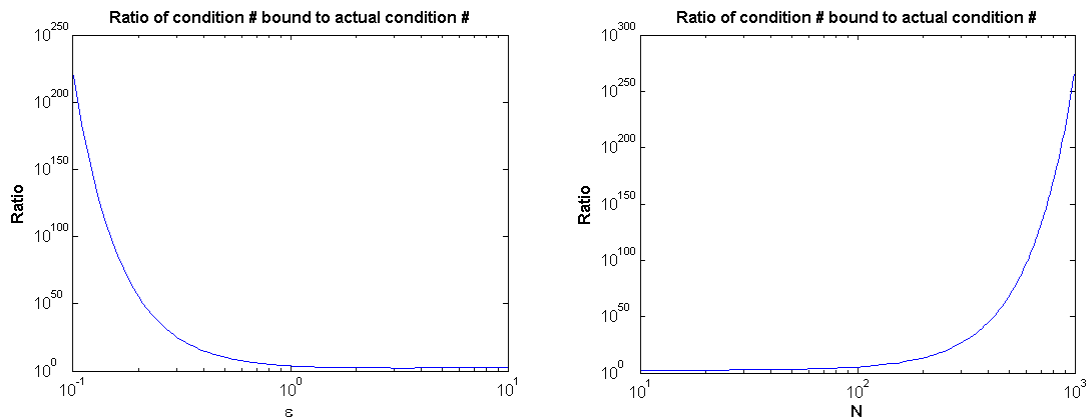


Figure 9: Ratio  $\frac{\text{CondBound}(K)}{\text{Cond}(K)}$  for varying values of  $\varepsilon$  with  $n = 10$  (left) and varying values of  $n$  with  $\varepsilon = 10$  (right).

- The bound becomes significantly less tight as  $\varepsilon$  decreases.
- The bound becomes significantly less tight as  $n$  increases.

This bound is loose for small  $\varepsilon$  and large  $n$ , the places where the highest accuracy is generally found. Therefore, this bound will not be very useful.

## 2.4 Experiment 3: Locating Instability Using Positive Definiteness of the Interpolation Matrix

Even when reproducing kernels are known to be positive definite functions, numerical error may cause the interpolation matrix to not be numerically positive definite. Many numerical algorithms for solving positive definite systems integrate a method for verifying numerical positive definiteness into the solver. Some examples include MATLAB's backslash operator and the integrated MATLAB function `chol()`. Therefore, if numerical positive definiteness is an accurate predictor of instability, that factor could be used efficiently in optimization algorithms.

To investigate this possibility, we consider graphs of RMS error plotted against the shape parameter  $\varepsilon$  for a given approximation problem, such as Figure 4 and Figure 5. However, the following graphs will also indicate whether the interpolation matrix,  $K$  from (2), is numerically positive definite. To identify the numerical positive definiteness of  $K$ , the error curve is colored blue when  $K$  is numerically positive definite, and the error curve is colored red when  $K$  is not numerically positive definite as calculated by MATLAB.

Figure 10 shows the RMS error and numerical positive definiteness of the interpolation matrix for RBF interpolations approximating  $f(\mathbf{x}) = x_1 \cdot \sin^2(x_1) \cdot e^{-x_2^2}$  on  $[1, 2] \times [1, 2]$  for many values of  $\varepsilon$ . The  $n = 120$  randomly distributed data sites used to construct the interpolants are the same data sites used in producing Figure 5. From Figure 10, it appears that the interpolation matrix  $K$  is numerically positive definite for all shape parameters greater than  $\varepsilon \approx 2.14$ , and that  $K$  is not numerically positive definite for shape parameters smaller than  $\varepsilon \approx 2.14$ . In this case, it appears that a stable linear system coincides with a numerically positive definite interpolation matrix.

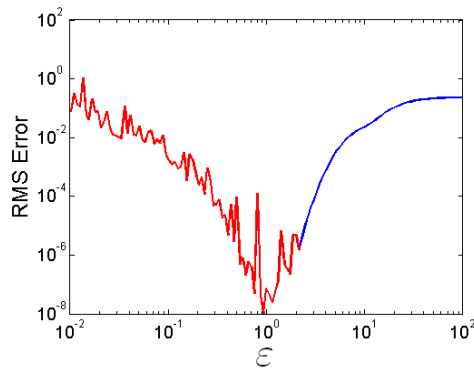


Figure 10: Error for RBF interpolation of  $f(\mathbf{x}) = x_1 \cdot \sin^2(x_1) \cdot e^{-x_2^2}$  on  $[1, 2] \times [1, 2]$  using Gaussian RBFs with shape parameter  $\varepsilon$  and 120 randomly distributed data sites.

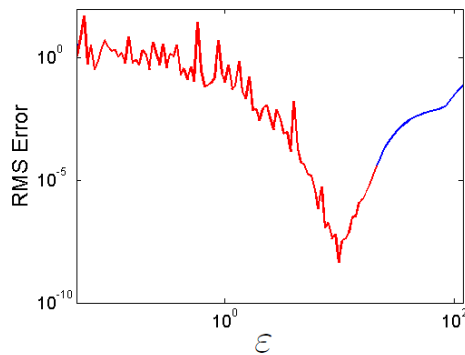


Figure 11: Error for RBF interpolation of  $f(x) = \frac{1}{25x^2+1}$  on  $[-\frac{1}{2}, \frac{1}{2}]$  using Gaussian RBFs with shape parameter  $\varepsilon$  and 80 randomly distributed data sites.

Figure 11 shows the RMS error and numerical positive definiteness of the interpolation matrix for RBF interpolations approximating  $f(x) = \frac{1}{25x^2+1}$  on  $[-\frac{1}{2}, \frac{1}{2}]$  for many values of  $\varepsilon$ . Gaussian RBFs with  $n = 80$  randomly distributed data sites are used to construct the interpolants. In this case, it appears that the interpolation matrix  $K$  is numerically positive definite for all shape parameters greater than  $\varepsilon \approx 20$ , and that  $K$  is not numerically positive definite for shape parameters smaller than  $\varepsilon \approx 20$ . However, the problem appears to remain stable for shape parameters greater than  $\varepsilon \approx 16$ . The discrepancy is particularly significant in this example, because as  $\varepsilon$  decreases from 20 to 16, the RMSE error drops from  $10^{-5}$  to  $10^{-6}$ .

Figure 12 shows the RMS error and numerical positive definiteness of the interpolation matrix for RBF interpolations approximating  $f(\mathbf{x}) = \frac{x_1}{x_2} \cdot \sin(x_1) + x_3$  on  $[1, 2] \times [1, 2] \times [1, 2]$  for many values of  $\varepsilon$ . The interpolants are constructed using Gaussian RBFs with  $n = 80$  randomly distributed data sites. Due to memory constraints, the RMS error for this problem is computed using  $(\frac{n}{4})^s$  evaluation points, uniformly distributed on  $[1, 2] \times [1, 2] \times [1, 2]$ . This problem appears to be unstable for shape parameters smaller than  $\varepsilon \approx 0.3$ . However, the interpolation matrix  $K$  remains numerically positive definite for shape parameters greater than  $\varepsilon \approx 0.25$ .

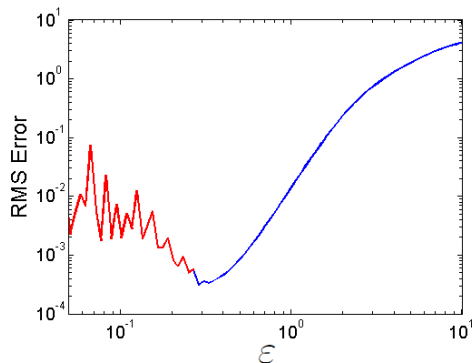


Figure 12: Error for RBF interpolation of  $f(\mathbf{x}) = \frac{x_1}{x_2} \cdot \sin(x_1) + x_3$  on  $[1, 2] \times [1, 2] \times [1, 2]$  using Gaussian RBFs with shape parameter  $\varepsilon$  and 80 randomly distributed data sites.

One notable observation is that, in each graph, the interpolation matrix  $\mathbf{K}$  appears to make only a single transition between being numerically positive definite and not numerically positive definite. Additionally, this transition occurs near the observed transition from a stable interpolation problem to an unstable interpolation problem. Numerical positive definiteness of the interpolation matrix may be useful for efficiently predicting instability.

## 2.5 Experiment 4: Continuity when Adding New Data Sites

Previous experiments have investigated how changing the shape parameter of RBFs affects the accuracy of an RBF approximation of a function  $f$  using a fixed data set. This section investigates how adding and removing data sites from an RBF interpolation changes the relation between RMS error of the interpolant and the shape parameter of the basis functions used. Adding or removing data sites will add or remove terms from the interpolant and also result in a new set of coefficients  $\mathbf{c}$ . To investigate the effects of changing the data set, the following experiments will compute an RBF interpolation for a given problem using many different shape parameters, then new points are added to the data set and the process is repeated.

For the following experiments, we are given a function  $f : \mathbb{R}^s \rightarrow \mathbb{R}$  and a set of data  $N$  sites  $\mathbb{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$  along with values of  $f$  at those data sites. At each step, the first  $n$  data sites from  $\mathbb{X}$  are used to construct interpolants using many values of  $\varepsilon$ , and an RMS error is computed for each interpolant. In this case, the RMS error for the interpolants is computed using  $N^s$  evaluation points, uniformly distributed on the given domain.

Figure 13 displays the RMS error for interpolants attempting to approximate  $f(\mathbf{x}) = x_1 \cdot \sin^2(x_1) \cdot e^{-x_2^2}$  on  $[1, 2] \times [1, 2]$  using Gaussian RBFs. The first 8 data sites used are the data sites shown in Figure 1, and at each step, 4 additional data sites are added at random points in  $[1, 2] \times [1, 2]$ . The colors in Figure 13 represent RMS error. Although this problem becomes unstable as  $n$  increases, the RMS error appears to be continuous with respect to both changes in  $\varepsilon$  and changes in the data set as data sites are added or removed.

The predictor methods mentioned in Section 2.1 each have computational cost  $O(n^3)$ . Therefore, changing  $n$  has a significant impact on the efficiency of computing the values of predictors. Given the apparent continuity of the RMS error with respect to changes in the data set, algorithms

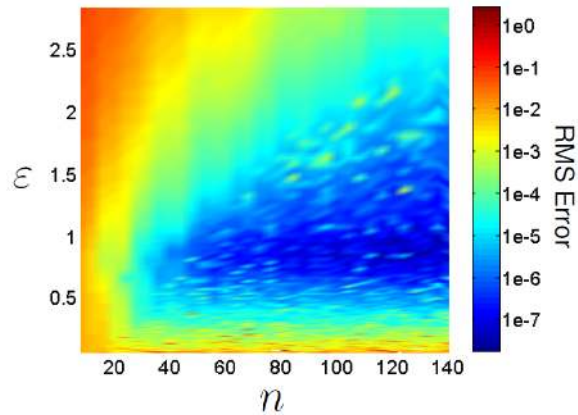


Figure 13: Error for RBF interpolation of  $f(\mathbf{x}) = x_1 \cdot \sin^2(x_1) \cdot e^{-x_2^2}$  on  $[1, 2] \times [1, 2]$  using Gaussian RBFs with shape parameter  $\varepsilon$  and  $n$  data sites.

attempting to optimize the shape parameter  $\varepsilon$  may benefit from including steps which use a subset of the problem's full data set.

### 3 Choosing a Kernel

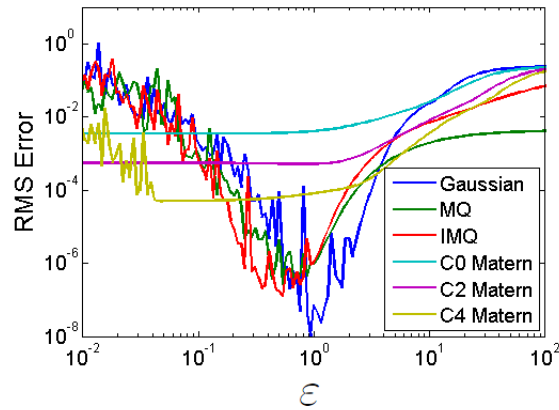
In a typical RBF interpolation, the radial basis functions used are translations of a single radial kernel  $K(\mathbf{x}, \mathbf{x}_c) = \kappa(\|\mathbf{x} - \mathbf{x}_c\|_2)$ . Section 2 explored methods for choosing a shape parameter to produce minimal error when using a given kernel. As seen in Section 2.2, the choice of basis functions can have a significant impact on the accuracy and stability of an interpolation. This section considers the problem of choosing a kernel for use in an RBF interpolation.

The graph in Figure 14 shows the RMS error for RBF interpolations of a single problem using different kernels for many values of each kernel's shape parameter  $\varepsilon$ . The kernels used are the six kernels defined in Table 1 (in Section 1.2). The RMS error shown results from performing an RBF interpolation using the labeled kernel attempting to approximate the function  $f(\mathbf{x}) = x_1 \cdot \sin^2(x_1) \cdot e^{-x_2^2}$  on the domain  $[1, 2] \times [1, 2]$  using shape parameter  $\varepsilon$  and  $n = 120$  randomly distributed data sites. The accompanying table lists the smallest RMS error found for each kernel and the corresponding shape parameter.

As can be seen in Figure 14, choosing a good kernel is dependent on the shape parameter(s) used. Depending on the shape parameters considered, any of these six kernels could produce the smallest error. Therefore, in order to choose a kernel, a shape parameter for each kernel must also be chosen.

#### 3.1 Experiment 5: Using Predictors to Select a Kernel

The experiments in this section investigate the potential use of predictor functions for choosing a kernel. An explanation of the predictor functions used can be found in Section 2.1. The following experiment compares both RMS errors and the values of predictors for an interpolation problem using each of the six kernels defined in Table 1 (in Section 1.2). Rather than comparing



Kernel	$\varepsilon$ When Minimized	Smallest RMS Error
Gaussian	$9.33 \cdot 10^{-1}$	$1.065 \cdot 10^{-8}$
Multiquadric	$5.36 \cdot 10^{-1}$	$2.589 \cdot 10^{-7}$
Inverse Multiquadric	$5.36 \cdot 10^{-1}$	$1.214 \cdot 10^{-7}$
$C^0$ Matérn	$1.09 \cdot 10^{-1}$	$3.485 \cdot 10^{-3}$
$C^2$ Matérn	$8.12 \cdot 10^{-1}$	$5.151 \cdot 10^{-4}$
$C^4$ Matérn	$4.74 \cdot 10^{-2}$	$4.890 \cdot 10^{-5}$

Figure 14: Error for RBF interpolation of  $f(\mathbf{x}) = x_1 \cdot \sin^2(x_1) \cdot e^{-x_2^2}$  on the domain  $[1, 2] \times [1, 2]$  using the specified basis functions with shape parameter  $\varepsilon$  and 120 data sites.

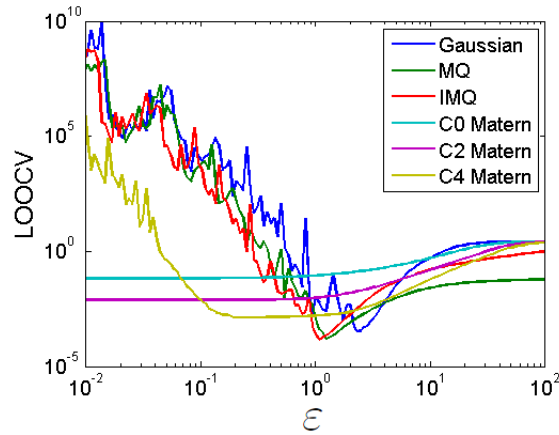
the predictors for each kernel at arbitrary shape parameters, this experiment considers the problem of choosing a shape parameter for each kernel independently using predictors. Predictors are then used to choose between the optimized interpolants.

First, consider the errors in Figure 14. The smallest RMS error was found using a Gaussian kernel, but if we restrict our focus to the stable region, then the smallest RMS error for interpolants using a Gaussian, multiquadric, or inverse multiquadric kernel is on the order of  $10^{-6}$ . The experiments in Section 2.4 indicate that restricting an interpolation to the stable region might be accomplished by checking for positive definiteness of the interpolation matrix  $K$ . However, the multiquadric kernel is not a positive definite function, and so that method cannot be used with a multiquadric kernel.

Figures 15 - 17 show the LOOCV, GCV, and MLE values associated with the same RBF interpolations as Figure 14. Each graph is accompanied by a table noting the shape parameters which minimize the predictor in question for each kernel, as well as the RMS error for an interpolation using that shape parameter with the associated kernel. The values of the predictors for each kernel with its chosen shape parameter are then used to rank the optimized interpolants. The final column lists the order in which the predictor function ranks the accuracy of the optimized interpolants, with 1st being the most accurate and 6th being the least accurate. As before, key concerns are the RMS error for an interpolant using the shape parameter which minimizes the predictor and how easily that minimum can be found.

Figure 15 shows the LOOCV values for each of the interpolants used to construct Figure 14.





Kernel	$\varepsilon$ When Minimized	RMS Error	Ranked by LOOCV
Gaussian	2.30	$3.926 \cdot 10^{-6}$	3rd
Multiquadric	1.23	$3.332 \cdot 10^{-6}$	2nd
Inverse Multiquadric	1.07	$1.682 \cdot 10^{-6}$	1st
$C^0$ Matérn	$1.03 \cdot 10^{-2}$	$3.495 \cdot 10^{-3}$	6th
$C^2$ Matérn	$1.03 \cdot 10^{-2}$	$5.594 \cdot 10^{-4}$	5th
$C^4$ Matérn	$1.89 \cdot 10^{-1}$	$5.270 \cdot 10^{-5}$	4th

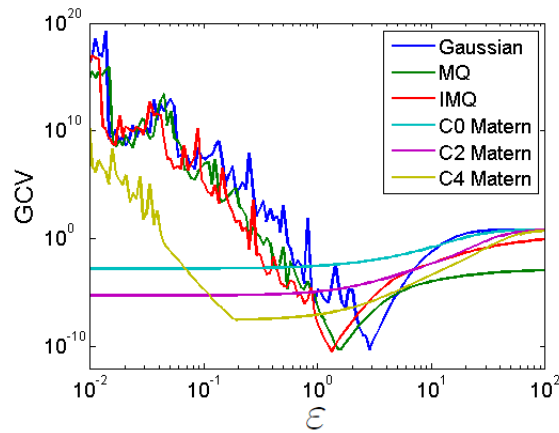
Figure 15: LOOCV values for RBF interpolation of  $f(\mathbf{x}) = x_1 \cdot \sin^2(x_1) \cdot e^{-x_2^2}$  on the domain  $[1, 2] \times [1, 2]$  using the specified basis functions with shape parameter  $\varepsilon$  and 120 data sites.

For the optimized interpolants, LOOCV correctly mirrored their relative accuracy. The smallest RMS errors found using LOOCV were on the order of  $10^{-6}$ , greater than the best errors possible. However, each curve reaches its minimum while the problem is stable. In this case, restricting our focus to the stable region would allow minima to be found easily without excluding the minima listed in Figure 15.

If any two pairings of basis function and shape parameter are chosen, using LOOCV results in a correct decision in 63.4% of cases. If we restrict our search to the stable region, the more accurate pairing is chosen in 93.4% of cases, and the incorrect decisions occur when RMS errors are close. Within the stable region, the largest impact from an incorrect decision was a 57% increase in RMS error.

Figure 16 shows the GCV values for each of the interpolants used to construct Figure 14. GCV correctly identified the inverse multiquadric kernel as the most accurate using the shape parameters it located. However, it incorrectly indicates that the Gaussian kernel produces a more accurate interpolant than the multiquadric using the shape parameters identified by GCV, when in fact the Gaussian kernel produced an error more than twice as large. As with LOOCV, the GCV for each kernel has a single well defined minimum in the stable region, which also appears to be its global minimum on  $\varepsilon \in [10^{-2}, 10^2]$ .

If any two pairings of basis function and shape parameter are chosen, using GCV results in a correct decision in 63.3% of cases. If we restrict our search to the stable region, the more accurate



Kernel	$\varepsilon$ When Minimized	RMS Error	Ranked by GCV
Gaussian	2.83	$3.970 \cdot 10^{-5}$	2nd
Multiquadric	1.52	$1.193 \cdot 10^{-5}$	3rd
Inverse Multiquadric	1.32	$8.687 \cdot 10^{-6}$	1st
$C^0$ Matérn	$1.03 \cdot 10^{-2}$	$3.495 \cdot 10^{-3}$	6th
$C^2$ Matérn	$1.03 \cdot 10^{-2}$	$5.594 \cdot 10^{-4}$	5th
$C^4$ Matérn	$1.89 \cdot 10^{-1}$	$5.270 \cdot 10^{-5}$	4th

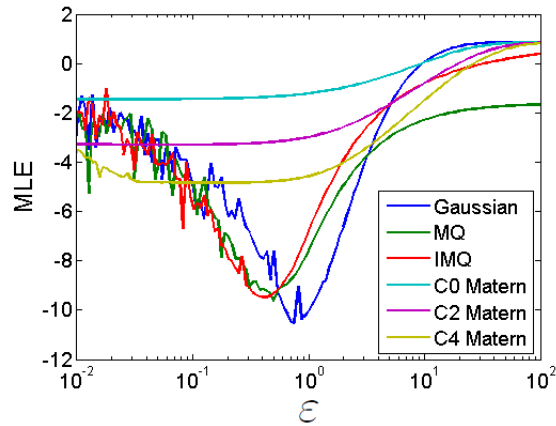
Figure 16: GCV values for RBF interpolation of  $f(\mathbf{x}) = x_1 \cdot \sin^2(x_1) \cdot e^{-x_2^2}$  on the domain  $[1, 2] \times [1, 2]$  using the specified basis functions with shape parameter  $\varepsilon$  and 120 data sites.

pairing is chosen in 93% of cases. Within the stable region, the largest impact from an incorrect decision was a 61% increase in RMS error.

Figure 17 shows the MLE values for each of the interpolants used to construct Figure 14. The MLE correctly identified the Gaussian kernel as producing the most accurate result. However, the MLE also incorrectly concluded that the multiquadric produced a more accurate interpolation than the inverse multiquadric. The errors found using the MLE were smaller than those found using LOOCV and GCV, and were near the smallest errors possible. However, the minima found for the Gaussian, multiquadric, and inverse multiquadric kernels are in the unstable region. Therefore, restricting the search to the stable region would result in larger RMS errors for the kernels which produced the most accurate interpolants. Not restricting the search would include numerous local minima, making it difficult to minimize the predictor function.

If any two pairings of basis function and shape parameter are chosen, using MLE results in a correct decision in 63.7% of cases. If we restrict our search to the stable region, the more accurate pairing is chosen in 92.9% of cases. Within the stable region, the worst result of an incorrect decision was a 60% increase in RMS error.

In this experiment, each of the predictor functions correctly identified which kernel produced the smallest RMS error when using the shape parameters identified by the same predictor function. Overall, the errors found by using GCV to choose a shape parameter were not as small as those found using LOOCV or MLE. GCV and MLE were less accurate than LOOCV in selecting a



Kernel	$\varepsilon$ When Minimized	RMS Error	Ranked by MLE
Gaussian	$7.58 \cdot 10^{-1}$	$4.964 \cdot 10^{-8}$	1st
Multiquadric	$5.00 \cdot 10^{-1}$	$4.339 \cdot 10^{-6}$	2nd
Inverse Multiquadric	$4.35 \cdot 10^{-1}$	$5.283 \cdot 10^{-7}$	3rd
$C^0$ Matérn	$1.56 \cdot 10^{-2}$	$3.494 \cdot 10^{-3}$	6th
$C^2$ Matérn	$6.25 \cdot 10^{-2}$	$5.543 \cdot 10^{-4}$	5th
$C^4$ Matérn	$1.77 \cdot 10^{-1}$	$5.238 \cdot 10^{-5}$	4th

Figure 17: MLE values for RBF interpolation of  $f(\mathbf{x}) = x_1 \cdot \sin^2(x_1) \cdot e^{-x_1^2}$  on the domain  $[1, 2] \times [1, 2]$  using the specified basis functions with shape parameter  $\varepsilon$  and 120 data sites.

kernel using those optimized shape parameters. Within the problem's stable region, each method was accurate when comparing shape parameters and kernels. Although incorrect decisions were made, in each case the impact of an incorrect decision was less than an order of magnitude.

## 4 Conclusions

Implementing radial basis function (RBF) methods requires choosing which basis functions to use, and many basis functions are defined by a shape parameter which must also be chosen. The choices made have a tremendous impact on the accuracy of the results and the numerical stability of the method used. The experimental results in this work indicate that leave-one-out cross validation (LOOCV), generalized cross validation (GCV), and maximum likelihood estimation (MLE) may be reasonable methods for selecting both basis functions and shape parameters. However, LOOCV and GCV are not accurate when the problem is unstable. Section 2.4 suggests an efficient method for avoiding such instability for interpolations using a numerically positive definite interpolation matrix in the case of LOOCV and GCV. Further, Figure 13 suggests that it may be possible to use a small data set to choose the shape parameter for a superset of that data.

Although MLE often produced the most accurate results, its behavior in unstable regions does not mirror the stability of the underlying interpolant. Without a method to avoid unstable behavior in the MLE, its improved accuracy cannot be achieved without risking inaccurate results caused

by potential instability.

## Acknowledgements

Portions of this work were supported by NSF grant DMS-0713848 with REU supplement. I would like to thank Dr Greg Fasshauer, Dr Fred Hickernell, and PhD student Qi Ye for many helpful conversations on these topics. I would also like to thank the referees for this paper who provided thorough and insightful comments.

## References

- [Aro50] N. Aronszajn. Theory of reproducing kernels. *Transactions of the American Mathematical Society*, 68:337–404, 1950.
- [Che09] Maggie E. Chenoweth. A numerical study of generalized multiquadric radial basis function interpolation. *SIAM Undergraduate Research Online*, 2:58–70, 2009.
- [Fas07] Gregory E. Fasshauer. *Meshfree Approximation Methods with MATLAB*. World Scientific Publishers, 2007.
- [FW04] Bengt Fornberg and Grady Wright. Stable computation of multiquadric interpolants for all values of the shape parameter. *Computers and Mathematics with Applications*, 48:853–867, 2004.
- [Har90] Rolland L. Hardy. Theory and applications of the multiquadric-biharmonic method: 20 years of discovery. *Computers and Mathematics with Applications*, 19:163–208, 1990.
- [Pir07] Cecile Piret. *Analytical and Numerical Advances in Radial Basis Functions*. PhD thesis, University of Colorado at Boulder, 2007.
- [Rip99] Shmuel Rippa. An algorithm for selecting a good value for the parameter  $c$  in radial basis function interpolation. *Advances in Computational Mathematics*, 11:193–210, 1999.
- [Ste99] Michael L. Stein. *Interpolation of Spatial Data: Some Theory for Kriging*. Springer, 1999.
- [TB97] Lloyd N. Trefethen and David Bau, III. *Numerical Linear Algebra*. Society for Industrial and Applied Mathematics, 1997.
- [Wah90] Grace Wahba. *Spline Models for Observational Data*. Society for Industrial and Applied Mathematics, 1990.