STABLE EVALUATION OF GAUSSIAN RBF INTERPOLANTS[*]

GREGORY E. FASSHAUER[†] AND MICHAEL J. MCCOURT[‡]

Abstract. We provide a new way to compute and evaluate Gaussian radial basis function interpolants in a stable way also for small values of the shape parameter, i.e., for “flat” kernels. This work is motivated by the fundamental ideas proposed earlier by Bengt Fornberg and his co-workers. However, following Mercer's theorem, an $L_2(\mathbb{R}^d, \rho)$-orthonormal expansion of the Gaussian kernel allows us to come up with an algorithm that is simpler than the one proposed by Fornberg, Larsson and Flyer and that is applicable in arbitrary space dimensions d. In addition to obtaining an accurate approximation of the RBF interpolant (using many terms in the series expansion of the kernel) we also propose and investigate a highly accurate least-squares approximation based on early truncation of the kernel expansion.

Key words. Radial basis functions, Gaussian kernel, stable evaluation, QR decomposition.

AMS subject classifications. 65D05, 65D15, 65F35, 41A63, 34L10

1. Introduction.

It is well-known that the standard or direct approach to interpolation at locations $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\} \subset \mathbb{R}^d$ with Gaussian kernels

$$K(\boldsymbol{x}, \boldsymbol{z}) = \mathrm{e}^{-\varepsilon^2 \|\boldsymbol{x}-\boldsymbol{z}\|^2}, \qquad \boldsymbol{x}, \boldsymbol{z} \in \mathbb{R}^d, \tag{1.1}$$

leads to a notoriously ill-conditioned interpolation matrix $\mathsf{K} = [K(\boldsymbol{x}_i, \boldsymbol{x}_j)]_{i,j=1}^N$ whenever ε, the so-called *shape parameter* of the Gaussian, is small, i.e., when the set $\{\mathrm{e}^{-\varepsilon^2\|\boldsymbol{x}-\boldsymbol{x}_j\|^2)}, j = 1, \ldots, N\}$ becomes numerically linearly dependent. This leads to severe numerical instabilities and limits the practical use of Gaussians — even though it is well known that one can approximate a function from the native reproducing kernel Hilbert space associated with the Gaussian kernel with spectral approximation rates (see, e.g., [7, 28]). The fact that most people are content with working in the “wrong” basis therefore has sparked many discussions, including the so-called uncertainty or trade-off principle [5, 21]. This uncertainty principle is tied directly to the use of the standard (“wrong”) basis, and we believe it can be circumvented by choosing a better — orthonormal — basis.

The idea of using a “better basis” for RBF interpolation is not a new one. It was successfully employed in [1] to obtain well-conditioned (and therefore numerically stable) interpolation matrices for polyharmonic splines in the context of a domain decomposition method. The technique used there — reverting to a homogeneous modification of the positive definite reproducing kernel associated with the conditionally positive definite polyharmonic spline kernel — was totally different from the one we pursue here. Our basis comes from a series expansion of the positive definite kernel and is rooted in the pioneering work of [19] and [25]. Combining series expansions of the kernel with a QR decomposition of the interpolation matrix to obtain a so-called RBF-QR algorithm was first proposed in [12] for interpolation with zonal kernels on the unit sphere $\mathbb{S}^2$ and in [11] for interpolation with Gaussian kernels in $\mathbb{R}^2$. These latter two papers motivated the results presented here. The main improvements provided by our work lie in establishing the general connection between the RBF-QR algorithm and Mercer or Hilbert-Schmidt series expansions of a positive definite kernel K defined on $\Omega \times \Omega$ of the form

$$K(\boldsymbol{x}, \boldsymbol{z}) = \sum_{n=1}^{\infty} \lambda_n \varphi_n(\boldsymbol{x}) \varphi_n(\boldsymbol{z}), \qquad \boldsymbol{x}, \boldsymbol{z} \in \Omega,$$

with appropriate scalars λ_n and functions φ_n. Here Ω can be a rather general set; however, in the following we will focus on $\Omega \subseteq \mathbb{R}^d$ (see Section 3 for more details). Having such an expansion allows us to formulate interpolation and approximation algorithms that can be implemented stably without too much trouble in any space dimension. We also consider an alternate highly accurate least-squares approximation algorithm for scattered data fitting with Gaussian kernels that in this form seems to be new to the literature even though general least-squares theory clearly suggests such an approach. In the following we will discuss the expansion

[*] The work of the first author was partially supported by the National Science Foundation under Grant No. DMS-0713848.

[†] Department of Applied Mathematics, Illinois Institute of Technology, fasshauer@iit.edu

[‡] Center for Applied Mathematics, Cornell University, mjm458@cornell.edu

we use for the Gaussian kernel, review the idea of the RBF-QR algorithm and discuss a number of details that are crucial for the implementation of our algorithms. Everything is supported with numerical experiments of Gaussian kernel interpolation and approximation of scattered data in space dimensions ranging from $d = 1$ to $d = 5$.

2. A Simple Series Expansion of the Gaussian. It is almost trivial to get an infinite series expansion for the one-dimensional Gaussian kernel (all this uses is the standard Taylor series expansion of the exponential function):

$$\begin{aligned} \mathrm{e}^{-\varepsilon^2(x-z)^2} &= \mathrm{e}^{2\varepsilon^2 xz}\mathrm{e}^{-\varepsilon^2x^2}\mathrm{e}^{-\varepsilon^2z^2} \\ &= \sum_{n=0}^{\infty}\frac{(2\varepsilon^2)^n}{n!}x^n\mathrm{e}^{-\varepsilon^2x^2}z^n\mathrm{e}^{-\varepsilon^2z^2}. \end{aligned} \tag{2.1}$$

According to this expansion we might be tempted to define

$$\lambda_n = \frac{(2\varepsilon^2)^n}{n!}, \qquad \varphi_n(x) = x^n\mathrm{e}^{-\varepsilon^2x^2}, \qquad n = 0, 1, 2, \dots,$$

Clearly, the functions φ_n, $n = 0, 1, 2, \dots$, are linearly independent on $\mathbb{R}$ and therefore form an alternate basis for the reproducing kernel Hilbert space associated with the one-dimensional Gaussian kernel. However, based on the hypothesis that we want our basis to consist of *orthonormal* functions, we should check whether the φ_n satisfy this condition.

We first look at the normalization of the φ_n. The following is easy to get from tables:

$$\int_{-\infty}^{\infty}\varphi_n^2(x)\mathrm{d}x = \int_{-\infty}^{\infty}x^{2n}\mathrm{e}^{-2\varepsilon^2x^2}\mathrm{d}x = \sqrt{\frac{\pi}{2}}\frac{(2n-1)!!}{2^{2n}\varepsilon^{2n+1}}.$$

A problem with these functions now arises since the φ_n are not orthogonal in the standard L_2 inner product used here. In general we have

$$\int_{-\infty}^{\infty}\varphi_n(x)\varphi_m(x)\mathrm{d}x = \int_{-\infty}^{\infty}x^{n+m}\mathrm{e}^{-\varepsilon^2x^2}\mathrm{d}x \neq 0.$$

For example, the integral for $(n, m) = (1, 3)$ is the same as the $(2, 2)$ integral, and therefore nonzero.

It actually turns out that the functions φ_n defined above *are* orthogonal, but only if interpreted in spaces of complex-valued functions (see [26]). This, however, does not seem to be practical. In fact, the authors of [11] claimed the series (2.1) was not ideal and therefore followed this expansion up with a transformation to polar coordinates and an expansion in terms of Chebyshev polynomials (thus the limitation of their work to $\mathbb{R}^2$). Note that Chebyshev polynomials represent an orthogonal eigenfunction basis — but only in the transformed coordinate system.

In Section 3.1 we consider an alternate series expansion of the Gaussian kernel that provides us with functions that are orthonormal over $\mathbb{R}^d$. The crucial ingredient will be the introduction of an appropriate weight function ρ into the inner product.

3. Eigenfunction Expansions. Mercer's theorem [19] (or alternatively Hilbert-Schmidt theory [25]) states that every positive definite kernel K can be represented in terms of its (positive) eigenvalues λ_n and (normalized) eigenfunctions φ_n as

$$K(\boldsymbol{x}, \boldsymbol{z}) = \sum_{n=1}^{\infty}\lambda_n\varphi_n(\boldsymbol{x})\varphi_n(\boldsymbol{z}). \tag{3.1}$$

To establish a connection between Mercer's theorem and generalized Fourier series as obtained via the much-better known Sturm-Liouville eigenvalue problem we consider the following ODE and boundary conditions

$$\begin{aligned} \varphi''(x) + \frac{1}{\lambda}\varphi(x) &= 0 \\ \varphi(0) = \varphi(1) &= 0 \end{aligned}$$

as an example. For this problem it is well-known that we have eigenvalues and eigenfunctions

$$\lambda_n = \frac{1}{n^2\pi^2}, \qquad \varphi_n(x) = \sin n\pi x.$$

The Green's function G for this ODE boundary value problem can be expressed as a Fourier sine series (with parameter z)

$$G(x,z) = \min(x,z) - xz = \begin{cases} x(1-z), & x < z \\ z(1-x), & x > z \end{cases} = \sum_{n=1}^{\infty} a_n(z)\varphi_n(x).$$

Here the Fourier coefficients are given by

$$a_n(z) = \frac{\sin n\pi z}{(n\pi)^2} = \lambda_n \varphi_n(z),$$

so that we can also identify G with K and write

$$K(x,z) = \sum_{n=1}^{\infty} \lambda_n \varphi_n(x)\varphi_n(z).$$

This kernel is positive definite (since the eigenvalues are positive) and satisfies Mercer's theorem where $1/\lambda_n$ and φ_n are the eigenvalues and eigenfunctions of the (inverse) integral operator defined by

$$(\mathcal{T}_K f)(x) = \int_0^1 K(x,z) f(z)\,\mathrm{d}z.$$

The kernel $K(x,z) = \min(x,z) - xz$ gives rise to the reproducing kernel Hilbert space

$$H_0^1(0,1) = \left\{ f \in H^1(0,1) : f(0) = f(1) = 0 \right\}$$

whose inner product is

$$\langle f, g \rangle = \int_0^1 f'(x) g'(x)\,\mathrm{d}x.$$

This is in fact a well-known Sobolev space similar to those often used in the theory of finite elements. On the other hand, reproducing kernel Hilbert spaces are the kind of function spaces positive definite kernels "live in". The connection between positive definite kernels and Green's kernels is discussed in detail in [6, 9, 10].

As is well-known from Fourier theory, Fourier series expansions are in many ways "optimal" (orthogonality, best approximation in the mean-square sense, fast decay of eigenvalues, etc.). The same is true for the kernel eigenfunction expansions guaranteed by Mercer's theorem and therefore ensures the success of our approach.

3.1. An Eigenfunction Expansion for Gaussians in $L_2(\mathbb{R}, \rho)$. For the remainder of this section we will concentrate on the one-dimensional situation. The generalization to multiple space dimensions d will be established below in a straightforward manner using the product form of the Gaussian kernel.

It turns out that one can derive (see [20]) a Mercer expansion

$$\mathrm{e}^{-\varepsilon^2(x-z)^2} = \sum_{n=1}^{\infty} \lambda_n \varphi_n(x)\varphi_n(z)$$

for the Gaussian kernel (1.1), with the functions φ_n being orthonormal in $L_2(\mathbb{R}, \rho)$. Here the inner product that determines how we measure orthogonality of functions in $L_2(\mathbb{R})$ is weighted by

$$\rho(x) = \sqrt{\frac{2a}{\pi}}\,\mathrm{e}^{-2ax^2}, \qquad a > 0. \tag{3.2}$$

The parameter a determines how the global domain $\mathbb{R}$ is “localized” and therefore can be interpreted as a *global scale parameter*.

Using this setup, the eigenfunctions φ_n of the Gaussian turn out to be

$$\varphi_n(x) = \frac{1}{\sqrt{2^{n-1}(n-1)!\sqrt{\frac{a}{c}}}}\, \mathrm{e}^{-(c-a)x^2} H_{n-1}\left(\sqrt{2c}x\right), \qquad n = 1, 2, \ldots, \tag{3.3}$$

with H_{n-1} the *classical Hermite polynomials* of degree $n-1$ defined by their Rodrigues’ formula

$$H_{n-1}(x) = (-1)^{n-1}\mathrm{e}^{x^2}\frac{\mathrm{d}^{n-1}}{\mathrm{d}x^{n-1}}\mathrm{e}^{-x^2} \quad \text{for all } x \in \mathbb{R},\ n = 1, 2, \ldots,$$

so that

$$\int_{\mathbb{R}} H_{n-1}^2(x)\,\mathrm{e}^{-x^2}\,\mathrm{d}x = \sqrt{\pi}\,2^{n-1}(n-1)! \qquad \text{for } \ n = 1, 2, \ldots .$$

The positive parameter c that appears in (3.3) is coupled to the global scale parameter a and the shape parameter ε via $c = \sqrt{a^2 + 2a\varepsilon^2}$.

The corresponding eigenvalues are

$$\lambda_n = \sqrt{\frac{2a}{a + \varepsilon^2 + c}}\left(\frac{\varepsilon^2}{a + \varepsilon^2 + c}\right)^{n-1}, \quad n = 1, 2, \ldots . \tag{3.4}$$

As already mentioned, the shape parameter ε is related to the *local scale of the problem* while a is related to the *global scale of the problem*. In principle, these two parameters can be chosen freely. Unfortunately, this choice is not totally independent if one wants a convergent and stable algorithm (see the discussion in Sections 5.2 and 6.3 for more details). As mentioned in the introduction, the shape parameter ε has important consequences for the stability and accuracy of Gaussian kernel interpolants. In this paper we will generally be interested in small values of ε and in most numerical experiments we will focus our attention on an interval of ε-values ranging from one on down to values very near zero.

When dealing with the global scale parameter a we have the option of either fixing a and taking $c = \sqrt{a^2 + 2a\varepsilon^2}$, or focusing our attention on the scaling of the Hermite polynomials and therefore fixing c, which implies that $a = -\varepsilon^2 + \sqrt{\varepsilon^4 + c^2}$. Alternatively, one might consider choosing a as some function of ε. We will discuss our parameter choices in more detail in the implementation section.

Note that for $\varepsilon \to 0$, i.e., for “flat” Gaussians, we always have $c \to a$ (or $a \to c$) and we see that the eigenfunctions φ_n converge to the normalized Hermite polynomials $\widetilde{H}_{n-1}(x) = \frac{1}{\sqrt{2^{n-1}n-1!}}H_{n-1}(\sqrt{2a}x)$, and the eigenvalues behave like $\left(\frac{\varepsilon^2}{2a}\right)^{n-1}$. This shows that the main source of ill-conditioning of the Gaussian basis is associated with the eigenvalues, and the RBF-QR strategy suggested in [11, 12] can be employed as explained in the next section.

These observations also provide another simple explanation as to why the “flat” limit of a Gaussian interpolant is a polynomial (see, e.g., [2, 4, 13, 16, 17, 18, 23]).

Looking at (3.4), one can observe that the eigenvalues $\lambda_n \to 0$ exponentially fast as $n \to \infty$ since the inequality $\varepsilon^2 < a + \varepsilon^2 + c$ holds for any $a, c, \varepsilon > 0$. This was used in [7] to establish dimension-independent convergence rates for approximation with Gaussian kernels.

3.2. Multivariate Eigenfunction Expansion. As mentioned above, the multivariate case is easily obtained using the tensor product form of the Gaussian kernel, i.e., for d-variate functions we have

$$\begin{aligned} K(\boldsymbol{x}, \boldsymbol{z}) &= \exp\left(-\varepsilon_1^2(x_1 - z_1)^2 - \ldots - \varepsilon_d^2(x_d - z_d)^2\right) \\ &= \sum_{\boldsymbol{n} \in \mathbb{N}^d} \lambda_{\boldsymbol{n}}\varphi_{\boldsymbol{n}}(x)\varphi_{\boldsymbol{n}}(z), \end{aligned}$$

where

$$\lambda_{\boldsymbol{n}} = \prod_{j=1}^{d} \lambda_{n_j} = \prod_{j=1}^{d} \sqrt{\frac{2a_j}{a_j+\varepsilon_j^2+c_j}} \left(\frac{\varepsilon_j^2}{a_j+\varepsilon_j^2+c_j}\right)^{n_j-1}, \tag{3.5}$$

$$\varphi_{\boldsymbol{n}}(\boldsymbol{x}) = \prod_{j=1}^{d} \varphi_{n_j}(x_j) = \prod_{j=1}^{d} \frac{1}{\sqrt{2^{n_j-1}(n_j-1)!\sqrt{\frac{a_j}{c_j}}}} \mathrm{e}^{-(c_j-a_j)x_j^2} H_{n_j-1}\left(\sqrt{2c_j}x_j\right), \tag{3.6}$$

and $\boldsymbol{x} = (x_1,\dots,x_d) \in \mathbb{R}^d$. Note that here we are allowed to take different shape parameters ε_j for different space dimensions (i.e., K will be an anisotropic kernel), or we can take them all equal, i.e., $\varepsilon_j = \varepsilon$, $j = 1,\dots,d$ (and then K is isotropic or radial). Because ε_j is allowed to vary by dimension, the values (a_j, c_j) may also vary by dimension, and in fact the same values for a_j and c_j are not required across dimensions. For the purposes of this work, we restrict ourselves to using the same ε_j in all dimensions (i.e., $\varepsilon_j = \varepsilon$ for $j = 1,\dots,d$), but future work will investigate the use of individual ε_j for each dimension.

4. A Stable Evaluation Algorithm. As already mentioned earlier, the starting point in [11] was an expansion of the form

$$\mathrm{e}^{-\varepsilon^2(x-z)^2} = \sum_{n=0}^{\infty} \frac{(2\varepsilon^2)^n}{n!} x^n \mathrm{e}^{-\varepsilon^2 x^2} z^n \mathrm{e}^{-\varepsilon^2 z^2}. \tag{2.1}$$

However, the authors claimed that this series is not ideal for stable "flat" limit calculations since it does not provide an effective separation of the terms that cause the ill-conditioning associated with small ε-values. Most likely, the poor conditioning of this new basis is due to the fact that the functions $x \mapsto x^n \mathrm{e}^{-\varepsilon^2 x^2}$ are not orthogonal in $L_2(\mathbb{R})$. Indeed, for $\varepsilon \to 0$ these functions converge to the standard monomial basis giving rise to the notoriously ill-conditioned Vandermonde matrix. Therefore, the authors followed up their initial expansion with a transformation to polar coordinates and an expansion in terms of Chebyshev polynomials (and thus the limitation of their algorithm to $\mathbb{R}^2$).

If one instead uses an eigenfunction expansion as discussed in the previous section, then the source of ill-conditioning of the Gaussian basis can be separated from the eigenfunctions and moved into the eigenvalues. Moreover, for a smooth kernel such as the Gaussian the eigenvalues decay very quickly so that we should now be able to directly (i.e., without having to deal with an additional transformation to Chebyshev polynomials) follow the QR-based strategy suggested in [11].

4.1. The RBF-QR Algorithm. In particular, we now use the Gaussian kernel (1.1) along with its eigenvalues (3.5) and eigenfunctions (3.6) as discussed above. To keep the notation simple, we assume that the eigenvalues and their associated eigenfunctions have been sorted linearly so that we can enumerate them with integer subscripts instead of the multi-index notation used in (3.5-3.6). This matter is not a trivial one and needs to be dealt with carefully in the implementation. The QR-based algorithm of [11] corresponds to the following: using the eigen-decomposition of the kernel function K, we can rewrite the kernel matrix K appearing in the linear system for the interpolation problem as

$$\begin{aligned}\mathsf{K} &= \begin{pmatrix} K(\boldsymbol{x}_1,\boldsymbol{x}_1) & \dots & K(\boldsymbol{x}_1,\boldsymbol{x}_N) \\ \vdots & & \vdots \\ K(\boldsymbol{x}_N,\boldsymbol{x}_1) & \dots & K(\boldsymbol{x}_N,\boldsymbol{x}_N) \end{pmatrix} \\ &= \begin{pmatrix} \varphi_1(\boldsymbol{x}_1) & \dots & \varphi_M(\boldsymbol{x}_1) & \dots \\ \vdots & & \vdots & \\ \varphi_1(\boldsymbol{x}_N) & \dots & \varphi_M(\boldsymbol{x}_N) & \dots \end{pmatrix} \begin{pmatrix} \lambda_1 & & & \\ & \ddots & & \\ & & \lambda_M & \\ & & & \ddots \end{pmatrix} \begin{pmatrix} \varphi_1(\boldsymbol{x}_1) & \dots & \varphi_1(\boldsymbol{x}_N) \\ \vdots & & \vdots \\ \varphi_M(\boldsymbol{x}_1) & \dots & \varphi_M(\boldsymbol{x}_N) \\ \vdots & & \vdots \end{pmatrix}.\end{aligned}$$

Of course we can not conduct computation on an infinite matrix, so we need to choose a truncation value M after which we neglect the remaining terms in the series. Since the eigenvalues $\lambda_n \to 0$ as $n \to \infty$ we have a necessary condition to encourage such a truncation. A particular choice of M will be discussed in Section 6, but given that we have chosen one the system changes to the much more manageable

$$\mathsf{K} = \underbrace{\begin{pmatrix} \varphi_1(\boldsymbol{x}_1) & \dots & \varphi_M(\boldsymbol{x}_1) \\ \vdots & & \vdots \\ \varphi_1(\boldsymbol{x}_N) & \dots & \varphi_M(\boldsymbol{x}_N) \end{pmatrix}}_{=\Phi} \underbrace{\begin{pmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_M \end{pmatrix}}_{=\Lambda} \underbrace{\begin{pmatrix} \varphi_1(\boldsymbol{x}_1) & \dots & \varphi_1(\boldsymbol{x}_N) \\ \vdots & & \vdots \\ \varphi_M(\boldsymbol{x}_1) & \dots & \varphi_M(\boldsymbol{x}_N) \end{pmatrix}}_{=\Phi^T},$$

or simply

$$\mathsf{K} = \Phi\Lambda\Phi^T. \tag{4.1}$$

Although a specific choice of M can be postponed until later, it is important to note that since our immediate goal is to avoid the ill-conditioning associated with radial basis interpolation as $\varepsilon \to 0$, we require $M \geq N$. This is in following with the work of Fornberg, and seeks to ensure that all of the eigenfunctions φ_n, $n = 1, \dots, M$, used above are obtained to machine precision. This also justifies — for all practical purposes — our continued use of an equality sign for the matrix factorization of K.

We are interested in determining a new basis where the interpolation can be conducted without the condition issues associated with the matrix K, while still remaining in the same space spanned by the Gaussian kernel function K. Thus an invertible matrix X^{-1} is needed such that $\mathsf{K}\mathsf{X}^{-1}$ is better conditioned than K. Of course, the simple choice would be $\mathsf{X}^{-1} = \mathsf{K}^{-1}$, but if that were available to machine precision this problem would be trivial.

The structure of the matrix Φ provides one possible avenue since each column contains eigenfunctions of the same order. This provides the opportunity to conduct a QR decomposition of Φ without mixing eigenfunctions of different orders. For $M > N$, the matrix Φ is "short and fat", meaning that the QR decomposition takes the form

$$\left(\begin{array}{ccc|ccc} \varphi_1(\boldsymbol{x}_1) & \dots & \varphi_N(\boldsymbol{x}_1) & \varphi_{N+1}(\boldsymbol{x}_1) & \dots & \varphi_M(\boldsymbol{x}_1) \\ \vdots & & \vdots & \vdots & & \vdots \\ \varphi_1(\boldsymbol{x}_N) & \dots & \varphi_N(\boldsymbol{x}_N) & \varphi_{N+1}(\boldsymbol{x}_N) & \dots & \varphi_M(\boldsymbol{x}_N) \end{array}\right) = \mathsf{QR} = \mathsf{Q}\left(\begin{array}{c|c} \mathsf{R}_1 & \mathsf{R}_2 \end{array}\right),$$

where the R_1 block is a square matrix of size N and R_2 is $N \times (M-N)$.

Substituting this decomposition for Φ^T in (4.1) we see

$$\mathsf{K} = \Phi\Lambda\mathsf{R}^T\mathsf{Q}^T.$$

By imposing the same block structure on Λ that was imposed on R we can rewrite this full system in blocks as

$$\begin{aligned} \mathsf{K} &= \Phi\begin{pmatrix} \Lambda_1 & \\ & \Lambda_2 \end{pmatrix}\begin{pmatrix} \mathsf{R}_1^T \\ \mathsf{R}_2^T \end{pmatrix}\mathsf{Q}^T \\ &= \Phi\begin{pmatrix} \Lambda_1\mathsf{R}_1^T \\ \Lambda_2\mathsf{R}_2^T \end{pmatrix}\mathsf{Q}^T \\ &= \Phi\begin{pmatrix} \mathsf{I}_N \\ \Lambda_2\mathsf{R}_2^T\mathsf{R}_1^{-T}\Lambda_1^{-1} \end{pmatrix}\Lambda_1\mathsf{R}_1^T\mathsf{Q}^T\,. \end{aligned} \tag{4.2}$$

The final form of this representation is significant because of the structure of the $\Lambda_2\mathsf{R}_2^T\mathsf{R}_1^{-T}\Lambda_1^{-1}$ term. In Section 3.1 we noticed that the eigenvalues $\lambda_n \to 0$ as $n \to \infty$ (and especially quickly if ε^2 is small relative to $a + c$). This means that the eigenvalues in Λ_2 are smaller than those in Λ_1 and thus none of the entries in $\mathsf{R}_2^T\mathsf{R}_1^{-T}$ are magnified when we form $\Lambda_2\mathsf{R}_2^T\mathsf{R}_1^{-T}\Lambda_1^{-1}$.

Since we can perform the multiplications by the diagonal matrices Λ_2 and Λ_1^{-1} *analytically* we avoid the ill-conditioning that would otherwise be associated with underflow (the values in Λ_2 are as small as ε^{2M-2}) or overflow (the values in Λ_1^{-1} are as large as ε^{-2N-2}).

Let us now return to the original goal of determining a new basis that allows us to conduct the interpolation in a safe and stable manner. Since we have now concluded that as $\varepsilon \to 0$ the $\Lambda_2\mathsf{R}_2^T\mathsf{R}_1^{-T}\Lambda_1^{-1}$ term poses no problems, we are left to consider the $\Lambda_1 \mathsf{R}_1^T \mathsf{Q}^T$ term. This matrix will be nonsingular if Φ^T has full row rank because Λ_1 is diagonal with nonzero (in exact arithmetic) values and R_1 is upper triangular and will have the same rank as Φ^T. Because of the orthogonality of the eigenfunctions φ_n, $n = 1, \ldots, M$, we will have nonsingularity of R_1 and thus a good choice for the matrix X is given by

$$\mathsf{X} = \Lambda_1 \mathsf{R}_1^T \mathsf{Q}^T. \tag{4.3}$$

We are now interested in the new system defined by

$$\begin{aligned} \Psi &= \mathsf{K}\mathsf{X}^{-1} \\ &= \left[\Phi \begin{pmatrix} \mathsf{I}_N \\ \Lambda_2 \mathsf{R}_2^T \mathsf{R}_1^{-T} \Lambda_1^{-1} \end{pmatrix} \Lambda_1 \mathsf{R}_1^T \mathsf{Q}^T\right] \left[\Lambda_1 \mathsf{R}_1^T \mathsf{Q}^T\right]^{-1} \\ &= \Phi \begin{pmatrix} \mathsf{I}_N \\ \Lambda_2 \mathsf{R}_2^T \mathsf{R}_1^{-T} \Lambda_1^{-1} \end{pmatrix}. \end{aligned} \tag{4.4}$$

Here we used (4.3) and the decomposition (4.2) of K. We can interpret the columns of Ψ as being created by new basis functions which can be thought of as the first N eigenfunctions plus a correction involving a linear combination of the next $M - N$ eigenfunctions:

$$\begin{aligned} \Psi &= \left(\begin{array}{ccc|ccc} \varphi_1(\boldsymbol{x}_1) & \ldots & \varphi_N(\boldsymbol{x}_1) & \varphi_{N+1}(\boldsymbol{x}_1) & \ldots & \varphi_M(\boldsymbol{x}_1) \\ \vdots & & \vdots & \vdots & & \vdots \\ \varphi_1(\boldsymbol{x}_N) & \ldots & \varphi_N(\boldsymbol{x}_1) & \varphi_{N+1}(\boldsymbol{x}_1) & \ldots & \varphi_M(\boldsymbol{x}_N) \end{array}\right) \begin{pmatrix} \mathsf{I}_N \\ \Lambda_2 \mathsf{R}_2^T \mathsf{R}_1^{-T} \Lambda_1^{-1} \end{pmatrix} \\ &= \left(\begin{array}{c|c} \Phi_1 & \Phi_2 \end{array}\right) \begin{pmatrix} \mathsf{I}_N \\ \Lambda_2 \mathsf{R}_2^T \mathsf{R}_1^{-T} \Lambda_1^{-1} \end{pmatrix} \\ &= \Phi_1 + \Phi_2 \left[\Lambda_2 \mathsf{R}_2^T \mathsf{R}_1^{-T} \Lambda_1^{-1}\right]. \end{aligned} \tag{4.5}$$

In order to see the actual basis functions we consider the vector $\Psi(\boldsymbol{x})$ defined as

$$\begin{aligned} \Psi(\boldsymbol{x})^T &= \begin{pmatrix} \psi_1(\boldsymbol{x}) & \ldots & \psi_N(\boldsymbol{x}) \end{pmatrix} \\ &= \begin{pmatrix} \varphi_1(\boldsymbol{x}) & \ldots & \varphi_M(\boldsymbol{x}) \end{pmatrix} \begin{pmatrix} \mathsf{I}_N \\ \Lambda_2 \mathsf{R}_2^T \mathsf{R}_1^{-T} \Lambda_1^{-1} \end{pmatrix}. \end{aligned}$$

This representation is in the same fashion that the standard kernel basis could be written as

$$\begin{aligned} \boldsymbol{k}(\boldsymbol{x})^T &= \begin{pmatrix} K(\boldsymbol{x}, \boldsymbol{x}_1) & \ldots & K(\boldsymbol{x}, \boldsymbol{x}_N) \end{pmatrix} \\ &= \begin{pmatrix} \varphi_1(\boldsymbol{x}) & \ldots & \varphi_M(\boldsymbol{x}) \end{pmatrix} \Lambda \Phi^T \\ &= \begin{pmatrix} \varphi_1(\boldsymbol{x}) & \ldots & \varphi_M(\boldsymbol{x}) \end{pmatrix} \begin{pmatrix} \mathsf{I}_N \\ \Lambda_2 \mathsf{R}_2^T \mathsf{R}_1^{-T} \Lambda_1^{-1} \end{pmatrix} \Lambda_1 \mathsf{R}_1^T \mathsf{Q}^T, \end{aligned} \tag{4.6}$$

only now the ill-conditioning related to Λ_1 has been removed from the basis.

The approach described in this section should be applicable whenever one knows the eigenfunction (or other orthonormal basis) expansion of a positive definite kernel. One such example is provided by the approach taken in [12] for stable radial basis function approximation on the sphere, where the connection between the (zonal) kernels being employed on the sphere and spherical harmonics, which are the eigenfunctions of the Laplace-Beltrami operator on the sphere, has traditionally been a much closer one (see, e.g., [8]).

4.2. Implementation Details. The interpolation problem described in Section 1 can be written in matrix notation as

$$\mathsf{K}\boldsymbol{\gamma} = \boldsymbol{y}, \tag{4.7}$$

where K is the $N \times N$ kernel matrix, $\boldsymbol{y} = (y_1, \dots, y_N)^T$ is input data denoting the function values to be fitted at the points $\boldsymbol{x}_i$, $i = 1, \dots, N$, and $\boldsymbol{\gamma}$ is the unknown vector of coefficients. In the new basis $\Psi = (\psi_1, \dots, \psi_N)^T$ the system is still of size $N \times N$ and can be written in the form

$$\mathsf{\Psi}\boldsymbol{\beta} = \boldsymbol{y},$$

where the matrix $\mathsf{\Psi}$ was defined in (4.4), $\boldsymbol{y}$ is as above, and $\boldsymbol{\beta}$ is a new vector of coefficients. Once we have solved for these coefficients, the Gaussian kernel interpolant s can be evaluated at an arbitrary point $\boldsymbol{x} \in \mathbb{R}^d$ via

$$s(\boldsymbol{x}) = \Psi(\boldsymbol{x})^T \boldsymbol{\beta}.$$

Using (4.4), the linear solve itself takes the form

$$\mathsf{\Phi} \begin{pmatrix} \mathsf{I}_N \\ \mathsf{\Lambda}_2 \mathsf{R}_2^T \mathsf{R}_1^{-T} \mathsf{\Lambda}_1^{-1} \end{pmatrix} \boldsymbol{\beta} = \boldsymbol{y}, \tag{4.8}$$

where as before

$$\mathsf{\Phi}^T = \begin{pmatrix} \mathsf{R}_1^T \\ \\ \mathsf{R}_2^T \end{pmatrix} \mathsf{Q}^T,$$

and the block structure is defined with first blocks of size $N \times N$,

$$\mathsf{\Lambda} = \begin{pmatrix} \mathsf{\Lambda}_1 & \\ & \mathsf{\Lambda}_2 \end{pmatrix}, \qquad \mathsf{\Phi} = \begin{pmatrix} \mathsf{\Phi}_1 & \mathsf{\Phi}_2 \end{pmatrix}.$$

At this point, the system (4.8) could be solved by conducting the matrix-matrix multiplication and working with the resulting $N \times N$ matrix:

$$\left[\mathsf{\Phi}_1 + \mathsf{\Phi}_2 \left[\mathsf{\Lambda}_2 \mathsf{R}_2^T \mathsf{R}_1^{-T} \mathsf{\Lambda}_1^{-1}\right]\right] \boldsymbol{\beta} = \boldsymbol{y}.$$

Doing so, however, would disregard the QR decomposition that was already computed. Instead, we can use the fact that $\mathsf{\Phi} = \mathsf{QR}$ in (4.8) to rewrite the system as

$$\begin{aligned} & \mathsf{Q} \begin{pmatrix} \mathsf{R}_1 & \mathsf{R}_2 \end{pmatrix} \begin{pmatrix} \mathsf{I}_N \\ \mathsf{\Lambda}_2 \mathsf{R}_2^T \mathsf{R}_1^{-T} \mathsf{\Lambda}_1^{-1} \end{pmatrix} \boldsymbol{\beta} = \boldsymbol{y} \\ \iff \quad & \mathsf{Q}\mathsf{R}_1 \begin{pmatrix} \mathsf{I}_N & \mathsf{R}_1^{-1}\mathsf{R}_2 \end{pmatrix} \begin{pmatrix} \mathsf{I}_N \\ \mathsf{\Lambda}_2 \mathsf{R}_2^T \mathsf{R}_1^{-T} \mathsf{\Lambda}_1^{-1} \end{pmatrix} \boldsymbol{\beta} = \boldsymbol{y} \\ \iff \quad & \mathsf{Q}\mathsf{R}_1 \left[\mathsf{I}_N + \mathsf{R}_1^{-1}\mathsf{R}_2 \mathsf{\Lambda}_2 \mathsf{R}_2^T \mathsf{R}_1^{-T} \mathsf{\Lambda}_1^{-1}\right] \boldsymbol{\beta} = \boldsymbol{y}\,. \end{aligned}$$

This is especially nice because the term $\mathsf{R}_1^{-1}\mathsf{R}_2$ is just the transpose of $\mathsf{R}_2^T \mathsf{R}_1^{-T}$ and thus its value can be saved from earlier and the cost of $\mathcal{O}(N^2(M-N))$ can be saved.

Now the linear solve can be broken into two parts, where

$$\mathsf{Q}\mathsf{R}_1 \hat{\boldsymbol{\beta}} = \boldsymbol{y}, \tag{4.9a}$$

$$\left[\mathsf{I}_N + \mathsf{R}_1^{-1}\mathsf{R}_2 \mathsf{\Lambda}_2 \mathsf{R}_2^T \mathsf{R}_1^{-T} \mathsf{\Lambda}_1^{-1}\right] \boldsymbol{\beta} = \hat{\boldsymbol{\beta}}. \tag{4.9b}$$

Solving (4.9a) is almost trivial, since Q is orthonormal and R_1 is upper triangular. Solving (4.9b) can be done cleverly depending on the value of M:

- If M is chosen such that $M < 2N$, then the linear system can be treated as a low rank update to the identity and the inverse can be applied with the Sherman-Morrison formula. Total cost would be $\mathcal{O}\left((N^2(M-N))\right)$.
- If $M \geq 2N$, the cost of the interior inverse in the Sherman-Morrison formula would be greater than simply solving the original system, so a direct approach is preferred. Total cost would be $\mathcal{O}(N^3)$.

Because this search for a new basis is conducted with the goal of working in the "flat" kernel regime, it is logical to assume that we are dealing with small ε. Therefore, it is reasonable to assume that the value of M can be chosen relatively close to N because additional terms would be truncated. As a result, (4.9b) will in general be solved using the Sherman-Morrison formula:

$$\begin{aligned}\boldsymbol{\beta} &= \left[\mathsf{I}_N + \mathsf{R}_1^{-1}\mathsf{R}_2\Lambda_2\mathsf{R}_2^T\mathsf{R}_1^{-T}\Lambda_1^{-1}\right]^{-1}\hat{\boldsymbol{\beta}} \\ &= \left[\mathsf{I}_N - \mathsf{R}_1^{-1}\mathsf{R}_2\left[\mathsf{I}_{M-N} + \Lambda_2\mathsf{R}_2^T\mathsf{R}_1^{-T}\Lambda_1^{-1}\mathsf{R}_1^{-1}\mathsf{R}_2\right]^{-1}\Lambda_2\mathsf{R}_2^T\mathsf{R}_1^{-T}\Lambda_1^{-1}\right]\hat{\boldsymbol{\beta}}\,.\end{aligned}$$

Thus far the value of M has been fixed but left unknown. Our choice of M coincides with that of [12], where M is the smallest value that satisfies $\lambda_M < 10^{-16}\lambda_N$.

5. Numerical Experiments I.

To determine the eigenfunction expansion's ability to accurately carry out radial basis interpolation with Gaussian kernels in the $\varepsilon \to 0$ limit, some experiments need to be conducted.

5.1. 1D Interpolation.

The first set of experiments is limited to 1D and studies the effect of increasing the number of data points N. All the data points are located at the Chebyshev nodes within an interval $[x_a, x_b]$

$$x_i = \frac{1}{2}(x_b + x_a) - \frac{1}{2}(x_b - x_a)\cos\left(\pi\frac{i-1}{N-1}\right), \qquad i = 1, \ldots, N. \tag{5.1}$$

The interpolation is conducted using the eigenfunction-QR algorithm (abbreviated RBF-QR) over shape parameter values logarithmically spaced within $\varepsilon \in [10^{-2}, 10^{0.1}]$. For comparison, the solution to (4.7) is computed using the traditional [5] RBF solution (abbreviated RBF-Direct) over $\varepsilon \in [10^{-2}, 10^{1}]$.

Input values are produced by a function f and the error in the interpolant s is computed by

$$\text{error} = \frac{1}{\bar{N}}\sqrt{\sum_{k=1}^{\bar{N}}\left[\frac{f(\bar{x}_k) - s(\bar{x}_k)}{f(\bar{x}_k)}\right]^2},$$

where $\bar{x}_k$ are $\bar{N}$ uniformly spaced points at which s is compared to f. For the 1D experiments, $\bar{N} = 1000$, although this choice was made arbitrarily.

The experiments in 1D can be seen in Figure 5.1. Two functions were considered, first

$$f_1(x) = \frac{\sinh x}{1 + \cosh x}, \qquad x \in [-3, 3],$$

using $N = \{10, 20, 30\}$, which can be seen in Figure 5.1a. The second function considered was

$$f_2(x) = \sin\left(\frac{x}{2}\right) - 2\cos x + 4\sin(\pi x), \qquad x \in [-4, 4],$$

using $N = \{10, 20, 30\}$, which can be seen in Figure 5.1b.

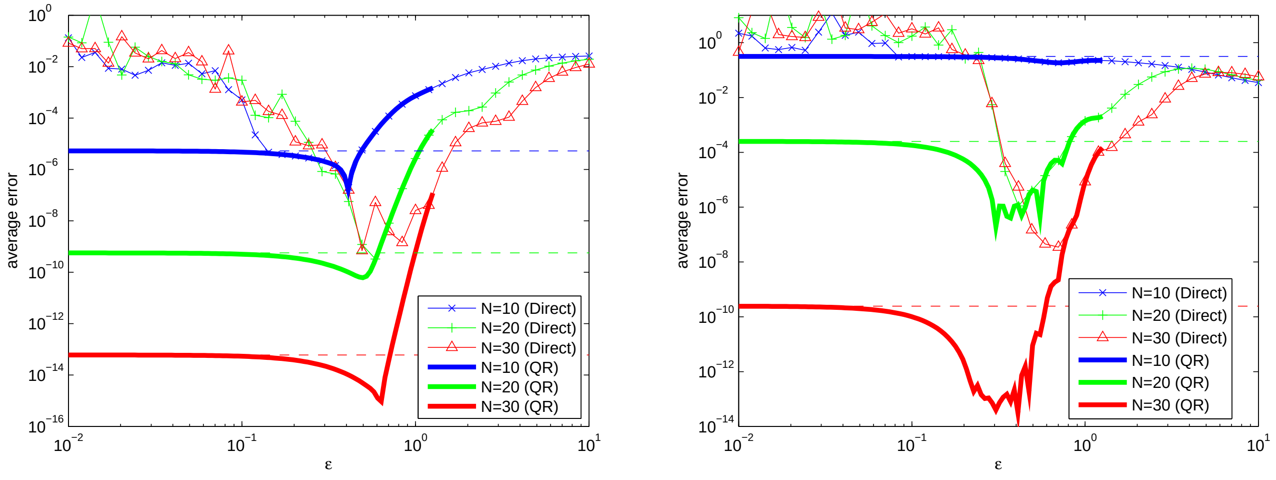
These initial results confirm that for $\varepsilon < 1$ the RBF-QR algorithm evaluates the Gaussian interpolant without the ill-conditioning associated with the $\varepsilon \to 0$ limit. The choice of series parameters here were $(a, c) = (1, \sqrt{1 + 2\varepsilon^2})$; using $a = 1$ sets the scale of the weight function to an appropriate scale for these interpolation problems.

The examples presented here also illustrate that interpolation with Gaussian kernels is more accurate than polynomial interpolation (which corresponds to the $\varepsilon \to 0$ limit) — even though both methods are known to be spectrally accurate. The errors for the corresponding polynomial interpolants are included as dashed horizontal lines in Figure 5.1.

5.2. Limitations to the Interpolation Algorithm.

Although the previous experiments successfully illustrate the usefulness of the eigenfunction expansion for the solution of the interpolation problem, performing a similar test with the Runge function

$$f_3(x) = \frac{1}{1 + x^2}, \qquad x \in [-3, 3],$$



(a) $f_1(x) = \sinh x(1 + \cosh x)^{-1}$

(b) $f_2(x) = \sin\left(\frac{x}{2}\right) - 2\cos x + 4\sin(\pi x)$

Fig. 5.1: Comparison of RBF-QR and RBF-Direct; dashed horizontal lines represent errors of limiting polynomial interpolants.

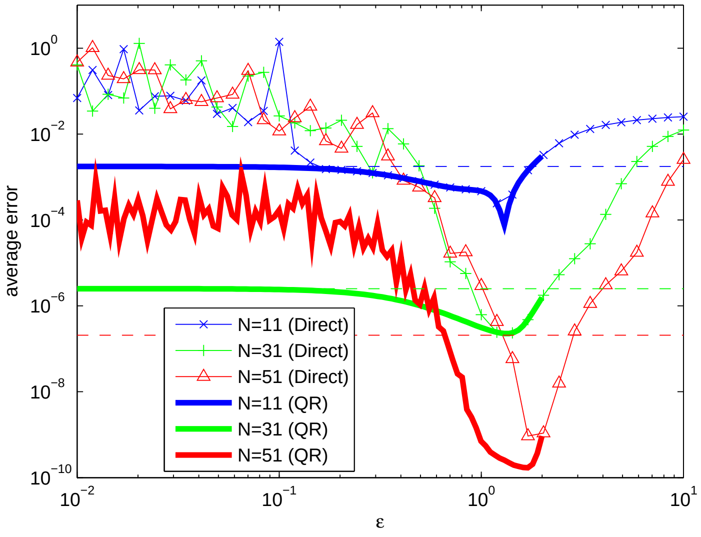


Fig. 5.2: For larger values of N, and therefore M, the eigenfunctions begin to show instabilities in the RBF-QR interpolation algorithm; dashed horizontal lines represent errors of limiting polynomial interpolants.

on an interval that is not affected by the famous Runge phenomenon will expose some limitations of our RBF-QR interpolation algorithm, as seen in Figure 5.2. Specific attention should be paid to the effect of increasing N on the emerging oscillations in the error of the interpolant.

This new source of error is different than the instability encountered in the $\varepsilon \to 0$ limit. To understand this, it is valuable to review Section 3.1 and evaluate the construction of the eigenvalues and eigenfunctions. For any pair of values (a, c), where $c = \sqrt{a^2 + 2a\varepsilon^2}$, the eigenvalues satisfying the orthonormality construction are

$$\lambda_n = \sqrt{\frac{2a}{a+\varepsilon^2+c}} \left(\frac{\varepsilon^2}{a+\varepsilon^2+c}\right)^{n-1}, \tag{3.4}$$

and the eigenfunctions are

$$\varphi_n(x) = \left(2^{n-1}(n-1)!\sqrt{\frac{a}{c}}\right)^{-1/2} \exp\left(-(c-a)x^2\right) H_{n-1}(\sqrt{2c}x). \tag{3.3}$$

The choice of $a = 1$, $c = \sqrt{1+2\varepsilon^2}$ produces an eigenpair

$$\begin{aligned}\lambda_n &= \sqrt{2}\left(1+\varepsilon^2+\sqrt{1+2\varepsilon^2}\right)^{-n+1/2} \varepsilon^{2n-2}, \\ \varphi_n(x) &= \left(2^{n-1}(n-1)!\right)^{-1/2} \left(1+2\varepsilon^2\right)^{1/8} \exp\left((1-\sqrt{1+2\varepsilon^2})x^2\right) H_{n-1}\left((4+8\varepsilon^2)^{1/4}x\right),\end{aligned}$$

which in the $\varepsilon \to 0$ limit reduces to

$$\begin{aligned}\lim_{\varepsilon\to 0} \lambda_n &= 2^{-n+1}\varepsilon^{2n-2}, \\ \lim_{\varepsilon\to 0} \varphi_n &= (2^{n-1}(n-1)!)^{-1/2} \exp(-\varepsilon^2 x^2) H_{n-1}(\sqrt{2}x)\end{aligned}$$

using the series expansion

$$1 - \sqrt{1+2\varepsilon^2} = -\varepsilon^2 + O(\varepsilon^4).$$

This analysis shows that for the pair $(a, c) = (1, \sqrt{1+2\varepsilon^2})$ chosen here the problem moves from traditional RBF interpolation to polynomial interpolation as $\varepsilon \to 0$: specifically, H_{n-1} takes inputs on the scale x, whereas the Gaussian still exists on the scale εx. Although the near-polynomial interpolation problem is better conditioned than the RBF problem, as evidenced by Figure 5.2, there is still a maximum accuracy with which this near-polynomial interpolation can be conducted.

The next logical thought is that a different choice of (a, c) might provide a better conditioned system. One possibility would be to choose c on the same order as ε^2: $c = \frac{1}{2}\varepsilon^2$, $a = \frac{1}{2}\varepsilon^2(\sqrt{5}-2)$. This leads to the eigenpair

$$\begin{aligned}\lambda_n &= \sqrt{\sqrt{5}-2}\left(\frac{\sqrt{5}+1}{2}\right)^{-n+1/2}, \\ \varphi_n(x) &= \left(2^{n-1}(n-1)!\sqrt{\sqrt{5}-2}\right)^{-1/2} \exp\left(-\varepsilon^2 x^2 \frac{3-\sqrt{5}}{2}\right) H_{n-1}(\varepsilon x),\end{aligned}$$

which is already at its $\varepsilon \to 0$ limit. Unfortunately, this eigenpair does not move the system far enough from the original RBF interpolation, as both the Hermite polynomials and Gaussian terms in the eigenfunctions take arguments on the same scale εx. This has the effect of trivializing the H_{n-1} term and the resulting eigenexpansion system has roughly the same condition as the original RBF system.

Because it is necessary to choose a so that the support of the weight function (3.2) matches the domain of the data points, $a = 1$ is an appropriate choice for this example. What may be causing the loss of accuracy above is instead the condition of the eigenfunctions, and specifically the increasingly high order of polynomials contributing to the eigenfunctions. The next section discusses this issue and a solution.

6. Early Truncation. As shown above, the ideal situation for the transformation of the RBF interpolation problem to the eigenexpansion problem seems to be when the Hermite polynomials exist on the scale of x. This particular choice moves the problem as far as possible from the ill-conditioning associated with the standard Gaussian kernel basis, but in turn is limited by a residual accuracy bound associated with near-polynomial interpolation.

6.1. Low-Rank Approximation. A possible strategy to remedy this issue essentially due to the use of high-degree polynomials would be to instead use regression with fewer eigenfunctions than are needed to construct the decomposition of the RBF interpolation matrix to within machine precision. In this way we will obtain a low-rank RBF approximation s of order M to the N pieces of data originally given in terms of the M leading eigenfunctions of the kernel. Rather than choosing M larger than N so as to evaluate the RBF interpolant to machine precision, M may be chosen smaller than N to minimize the error contributed to the polynomials via the linear combinations that constitute the high-order "correction terms" (see (4.5)) while still allowing for increasing N.

To introduce this problem in the same context as Section 4, $M \leq N$ is fixed and all the eigenvalues λ_n with $M < n \leq N$ will be set to zero. This results in an approximate kernel matrix decomposition

$$\begin{aligned}\mathsf{K} &\approx \Phi\tilde{\Lambda}\Phi^T \\ &= \begin{pmatrix} \Phi_1 & \Phi_2 \end{pmatrix} \begin{pmatrix} \Lambda_1 & \\ & 0 \end{pmatrix} \begin{pmatrix} \Phi_1 & \Phi_2 \end{pmatrix}^T,\end{aligned}$$

where Φ_1 contains the first M eigenfunctions, Λ_1 contains the first M (and only nonzero) eigenvalues, and Φ_2 contains the remaining $N - M$ eigenfunctions. Note that all these matrices are $N \times N$, and thus the QR decomposition from before is no longer necessary because Φ^T is invertible.

Defining the basis transformation matrix X analogously to (4.3) we now have

$$\mathsf{X} = \tilde{\Lambda}\Phi^T,$$

and because $\tilde{\Lambda}$ is not invertible we must instead consider the pseudoinverse [15]

$$\begin{aligned}\mathsf{X}^\dagger &= \Phi^{-T}\tilde{\Lambda}^\dagger \\ &= \Phi^{-T}\begin{pmatrix} \Lambda_1^{-1} & \\ & 0 \end{pmatrix}.\end{aligned}$$

This means that our new basis functions will be

$$\Psi(\boldsymbol{x})^T = \boldsymbol{k}(\boldsymbol{x})^T\mathsf{X}^\dagger,$$

which when expressed in terms of the eigenfunctions by expanding the kernel as in (4.6) yields

$$\begin{aligned}\Psi(\boldsymbol{x}) &= (\varphi_1(\boldsymbol{x}) \quad \dots \quad \varphi_N(\boldsymbol{x}))\Lambda\Phi^T\mathsf{X}^\dagger \\ &= (\varphi_1(\boldsymbol{x}) \quad \dots \quad \varphi_N(\boldsymbol{x}))\Lambda\Phi^T\Phi^{-T}\tilde{\Lambda}^\dagger \\ &= (\varphi_1(\boldsymbol{x}) \quad \dots \quad \varphi_N(\boldsymbol{x}))\Lambda\tilde{\Lambda}^\dagger \\ &= (\varphi_1(\boldsymbol{x}) \quad \dots \quad \varphi_N(\boldsymbol{x}))\begin{pmatrix} \mathsf{I}_M & \\ & 0 \end{pmatrix} \\ &= (\varphi_1(\boldsymbol{x}) \quad \dots \quad \varphi_M(\boldsymbol{x}) \quad 0 \quad \dots \quad 0)\end{aligned}$$

analytically setting the last $N - M$ eigenfunctions equal to 0. Recasting the original linear system in this new basis then gives

$$\begin{aligned}&\Psi\boldsymbol{\beta} = \boldsymbol{y} \\ \iff \quad &\Phi\Lambda\Phi^T\mathsf{X}^\dagger\boldsymbol{\beta} = \boldsymbol{y} \\ \iff \quad &\begin{pmatrix} \Phi_1 & \Phi_2 \end{pmatrix}\begin{pmatrix} \mathsf{I}_M & \\ & 0 \end{pmatrix}\boldsymbol{\beta} = \boldsymbol{y}.\end{aligned}$$

As this is written, it is clearly a low rank system, which is appropriate since M nonzero functions are being fit to $N > M$ data points. There are two ways, identical in exact arithmetic, to solve this low-rank linear system in a least squares sense: the first uses the theoretically guaranteed invertibility of Φ in applying the pseudoinverse, i.e.,

$$\boldsymbol{\beta} = \begin{pmatrix} \mathsf{I}_M & \\ & 0 \end{pmatrix} \Phi^{-1}\boldsymbol{y}.$$

This of course requires forming $\Phi^{-1}\boldsymbol{y}$ which would subject this problem to the same accuracy issues as before that arise from the near-polynomial eigenfunctions (for $\varepsilon \to 0$) given increasing N. Instead it seems the safest method of solving this system is to perform the matrix-matrix multiplication $\Phi\Lambda\Lambda^{\dagger}$ analytically, leaving the system

$$\begin{pmatrix} \Phi_1 & 0 \end{pmatrix} \boldsymbol{\beta} = \boldsymbol{y}. \tag{6.1}$$

Zeroing out the eigenvalues analytically has the effect of ignoring the final $N-M$ components of the coefficient vector $\boldsymbol{\beta}$ during the solve, as should be expected. Solving this in a least-squares sense requires solving

$$\min_{\boldsymbol{\beta}} \left\| \begin{pmatrix} \Phi_1 & 0 \end{pmatrix} \begin{pmatrix} \boldsymbol{\beta}_1 \\ \boldsymbol{\beta}_2 \end{pmatrix} - \boldsymbol{y} \right\|_2^2 \quad \Longleftrightarrow \quad \min_{\boldsymbol{\beta}} \left\| \Phi_1 \boldsymbol{\beta}_1 - \boldsymbol{y} \right\|_2^2,$$

where the components $\boldsymbol{\beta}_1$ and $\boldsymbol{\beta}_2$ of the coefficient vector $\boldsymbol{\beta}$ are of size M and $N-M$, respectively. Following this logic, the solution is

$$\boldsymbol{\beta}_1 = \Phi_1^{\dagger}\boldsymbol{y},$$

and $\boldsymbol{\beta}_2$ is unconstrained because the eigenfunctions associated with $\boldsymbol{\beta}_2$ are all identically zero.

6.2. Implementing Truncation. The implementation of this regression approach is more straightforward than that of the interpolation problem because this system can be rephrased as an over-determined least squares problem. One aspect that has thus far been omitted from our discussion is the selection of an M-value appropriate for early truncation. This choice is likely to play an important role given the instability in interpolation for increasing M observed in Section 5.2.

To give an initial idea of the effect of M on the quality of the approximation, let us consider a simple scattered data fitting problem. Given $f(x) = \cos x + \mathrm{e}^{-(x-1)^2} + \mathrm{e}^{-(x+1)^2}$ and fixing $\varepsilon = 10^{-5}$, we are given N evenly spaced values of f on the interval $[-3, 3]$. Different values of N ranging from 12 to 236 are chosen to conduct the regression with four different sets of M-values corresponding to 0.25, 0.5, 0.75, and 1 times N. The approximation error curves are displayed in Figure 6.1.

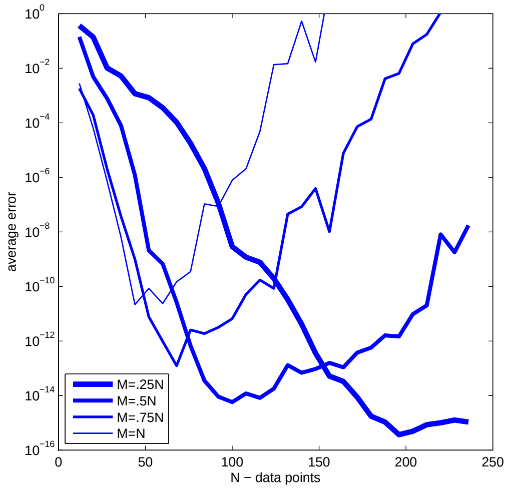


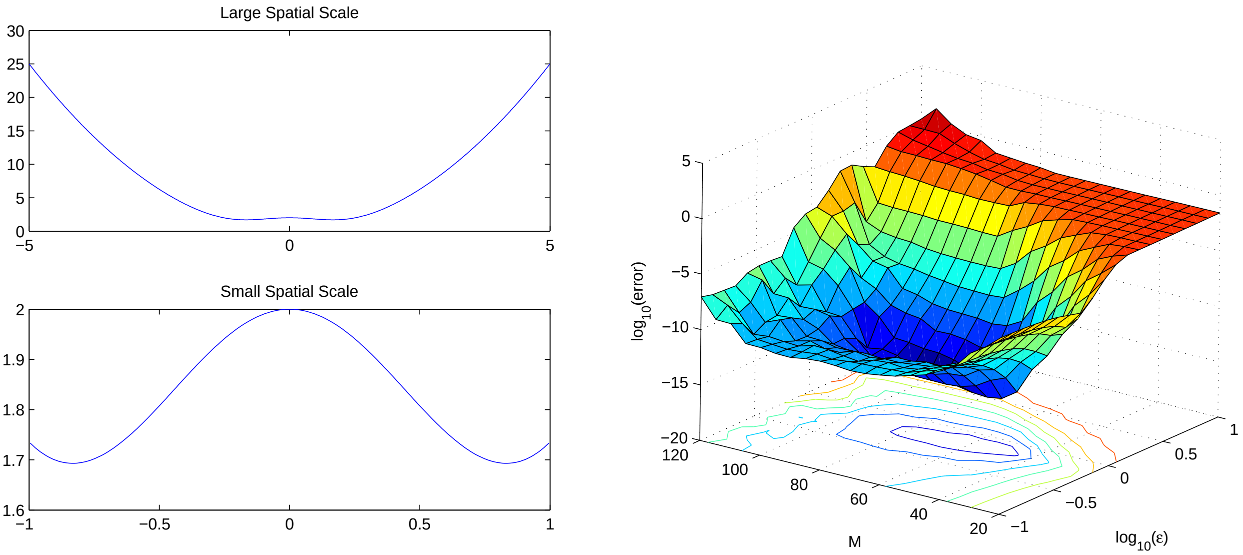
Fig. 6.1: For any number N of data sites (and fixed $\varepsilon = 10^{-5}$), $M \approx 48$ eigenfunctions are adequate for optimal accuracy of the QR regression algorithm.

Regardless of the size of N, Figure 6.1 shows that the optimal accuracy of the approximation consistently occurs for the same value of $M \approx 48$. This is encouraging because it indicates that for fixed ε, given any problem size N, there is a maximum space that the eigenfunctions can effectively span. The fact that the effective dimension of the space needed for an accurate kernel approximation is often rather small seems to be *mathematical folklore* and appears in different guises in various communities dealing with kernels. In the RBF literature we have, e.g., [14, 22] or the unpublished lecture notes [24]. In [24], for example, one can read that "there are low-rank subsystems which already provide good approximate solutions". In the statistical learning literature, the authors of [29] state that their main goal is "to find a fixed optimal m-dimensional linear model independent of the input data $\boldsymbol{x}$ [of sample size n], where the dimension m is also independent of n". Matrices that are of the same type as the kernel interpolation matrix K also arise in the method of fundamental solutions (MFS) for which the authors of [3] make similar observations regarding the non-trivial singular values, i.e., numerical rank, of K. In particular, in one of their examples [3, Figure 5] they see "no significant difference in accuracy using more than 40 [of 100] singular values".

To consider cases with varying ε, refer to Figure 6.2b. The 2D plot there shows the approximation error, for fixed $N = 200$, obtained with values of M and ε that vary independently. The function used to generate the data for the approximation problem is

$$f(x) = 2e^{-x^2} + x^2.$$

A plot of this function is provided in Figure 6.2a, both on the small spatial scale $x \in [-1, 1]$, where the exponential function dominates, and on the large spatial scale $x \in [-5, 5]$, where the polynomial term dominates. f is a useful function because in the absence of the exponential function term, the polynomial alone would be best interpolated with M on the same order as the polynomial and $\varepsilon \to 0$ [5]. The additional term gives rise to a region of optimal M and ε centered around $(M, \varepsilon) \approx (65, 0.7)$ far from the $\varepsilon = 0$ axis. Here the error is computed by taking the norm of the relative error computed at 1000 points in $[-5, 5]$.



(a) The function f exists on multiple scales.

(b) M and ε both play key roles for accuracy.

Fig. 6.2: Over a range of ε values (with fixed N), experiments show an optimal M range.

Finding an optimal value of M is still an open problem, as it probably depends not only on the choice of ε, but also on such factors as the location of the data points, anisotropy in higher dimensions, the choice of the pair (a, c) for (3.4) and more. It is possible that future work can examine optimal values of both M and ε simultaneously to determine an optimal approximation. For the purposes of this work, we are interested primarily in exploring the $\varepsilon \to 0$ limit and thus we will assume for future experiments that a good value of M is already chosen.

6.3. The Effects of M and a on Conditioning. Thus far we have been concerned exclusively with exploring the $\varepsilon \to 0$ limit for radial basis interpolation, which cannot be explored via the RBF-Direct approach because of ill-conditioning. Transitioning the traditional RBF problem into the RBF-QR formulation shifts the ill-conditioning from the radial basis functions to the eigenvalues which are inverted analytically. In Figure 5.2 it was shown that the eigenfunctions could themselves become ill-conditioned as the number of data points included in the problem increased. This necessitated RBF-QRr (QR regression) as discussed in Section 6.1, but in truncating the problem new issues arise.

Those issues can be isolated in the choices of series truncation value M and a, the term appearing first in (3.2), the definition of the weight function ρ. For RBF-QR, a was chosen to represent the global size of the problem so that orthogonality via the weight function was preserved, and M was chosen large enough so that $\lambda_M < \epsilon_{\text{mach}}\lambda_N$, where $\epsilon_{\text{mach}} = 10^{-16}$ is machine precision. In RBF-QRr we will now want to choose M at some value smaller than N, preferably much smaller for both conditioning and computational efficiency purposes. It is known that for any fixed value of a, the truncated eigenfunction expansion provides the best M-term approximation (see, e.g., [27]). However, the precise relationship between a and M (and also ε) and condition is as yet unknown because we are no longer considering the entire space spanned by the Gaussians, but rather an optimal M-term approximation to it parametrized by a.

Examine Figure 6.3 as a snapshot of the typical condition of a regression system for various values of M, a, N and ε. Here the condition number is defined as σ_1/σ_M where σ_k is the k^{th} largest singular value of the matrix Φ_1 as used in (6.1).

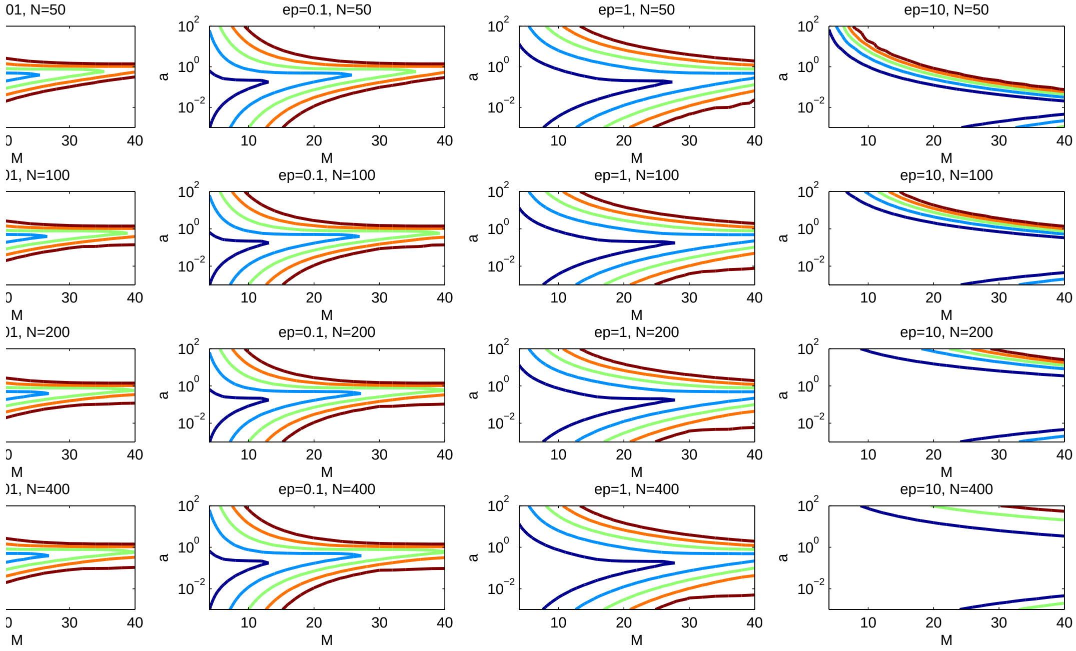


Fig. 6.3: Contour lines at condition values of $\{10^2, 10^5, 10^8, 10^{11}, 10^{14}\}$. Data points are evenly spaced in $[-5, 5]$.

There are many implications to consider from Figure 6.3, keeping in mind that the purpose of RBF-QRr is to allow us to explore the $\varepsilon \to 0$ regime for RBF interpolation.

- As $\varepsilon \to 0$ (i.e., looking at the columns of plots from right to left) there seems to be a limiting condition distribution. This is to be expected as it has already been shown in, e.g., [18] and mentioned in Sections 3.1 and 5.2 that RBF interpolation reaches a polynomial limit as $\varepsilon \to 0$.
- As $\varepsilon \to \infty$ (corresponding to the right-most column of plots) greater values of M can be chosen without incurring a condition penalty. This corresponds to the RBF-Direct case where allowing

$\varepsilon \to \infty$ produces more peaked/localized functions and a well-conditioned interpolation matrix.

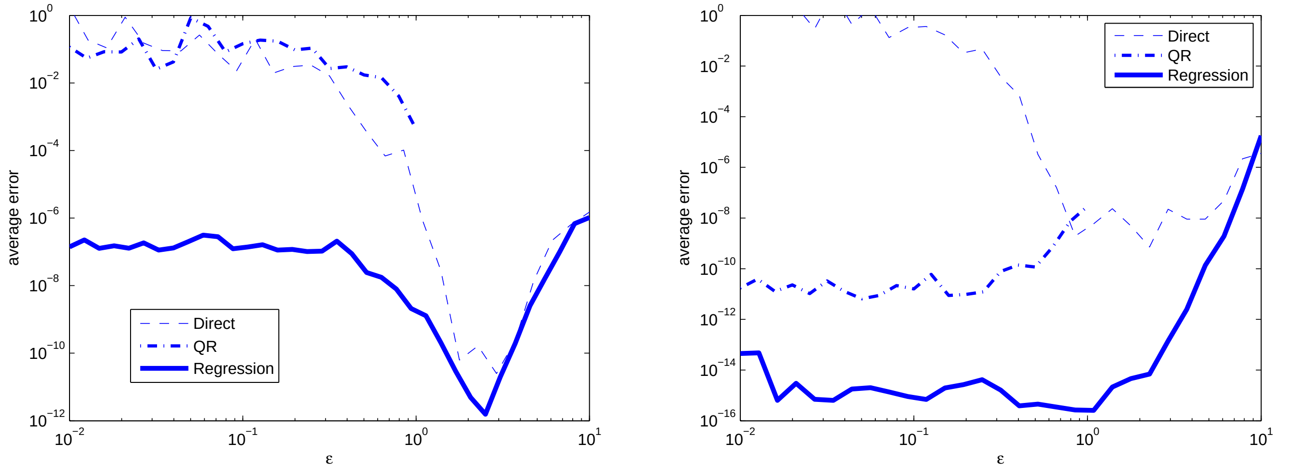
- For a given M, there is a single value of a which produces the optimally conditioned system. When interpreted as the scale of the weight function ρ in (3.2) defining the inner product in which we measure orthogonality, it seems logical that there is only one a-value which best represents the scale.
- Increasing the number N of data points (i.e., looking at the plots from top to bottom) has no negative effect on the condition. The only visible effect is with larger ε, which generates more localized Gaussians and in turn a better conditioned system. The $\varepsilon = 10$ pictures show the region of low condition significantly larger as N grows which we attribute to there being more "space" to fill with more eigenfunctions, and thus higher M is permitted without compromising condition.
- There is a region $M \in (M_{\text{max}}, \infty)$ for which no a exists such that the condition of the system is less than $1/\epsilon_{\text{mach}}$. In some sense, this means that the dimension of the space we are interested in approximating can only be stably represented using $M < M_{\text{max}}$ eigenfunctions. This reinforces our findings of Section 5.2.

Using these insights we have set the following guidelines governing our choices of M and a for the RBF-QRr regression experiments reported in Section 7:

1. Given a value of ε, M should be chosen as large as possible under the constraint that the condition κ of the system (6.1) be less than $\kappa_{\text{max}} = 10^8$.
2. a should be chosen to minimize the condition of the regression system.

7. Numerical Experiments II. In this section we provide comparisons of the RBF-QRr regression algorithm to the RBF-direct method and — in Section 7.1 to RBF-QR as described in Section 4 — for various data sets in various space dimensions. In each of these experiments, the truncation range for the value of M used in the regression is specified.

7.1. 1D Approximation. In this series of experiments the data is generated by two different univariate functions f evaluated at N evenly spaced points.



(a) $f(x) = (1 + x^2)^{-1}$, $N = 100$, $45 \le M \le 66$

(b) $f(x) = \cos x + \mathrm{e}^{-(x-1)^2} - \mathrm{e}^{-(x+1)^2}$, $N = 150$, $40 \le M \le 64$

Fig. 7.1: Regression avoids both the ill-conditioning in RBF-QR associated with large N, and the $\varepsilon \to 0$ ill-conditioning in RBF-Direct.

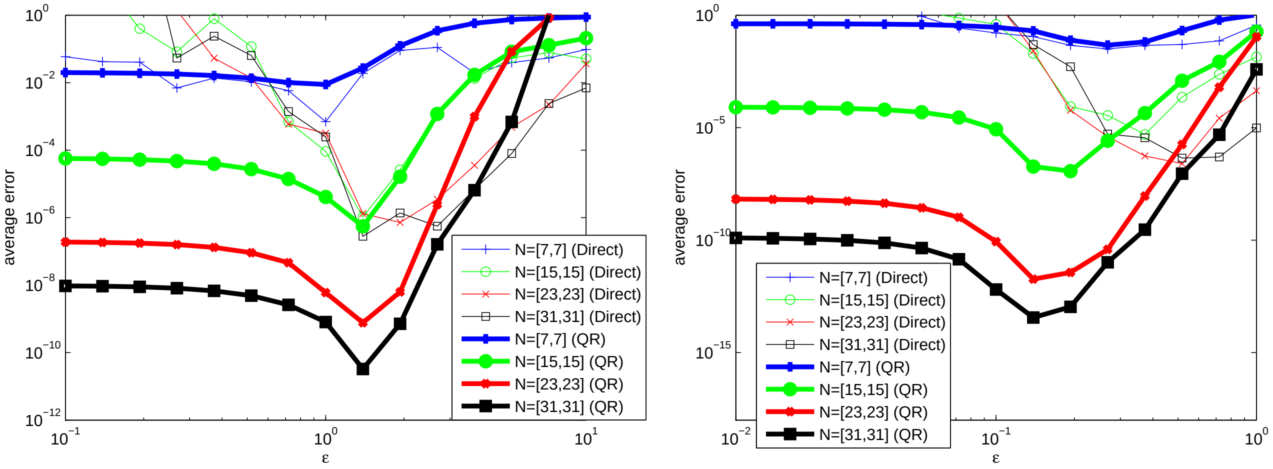
In Figure 7.1a we recall the earlier interpolation problem described in Figure 5.2, where RBF-QR was successfully used to approximate $f(x) = (1+x^2)^{-1}$ on $[-3, 3]$ for a small number of points but failed for more points. Here we can see that for $N = 100$ points RBF-QR now suffers from an even worse ill-conditioning. Regression with an appropriate M-value avoids this ill-conditioning and produces an accurate approximation to the given function f as $\varepsilon \to 0$. As a reference, for $\varepsilon = 10^{-2}$ our strategy for choosing the truncation length M for the regression algorithm leads to the use of $M = 45$ eigenfunctions, which, for such a small ε, are essentially polynomials. This matches the error for polynomial interpolation shown as a horizontal line in Figure 5.2 for the case $N = 51$ quite well.

In Figure 7.1b a linear combination of trigonometric and exponential functions is used to generate data at $N = 150$ points. RBF-QR again outperforms RBF-Direct, but for large N we still see that regression is preferred.

7.2. Higher-dimensional Approximation. One of the great benefits of considering radial basis functions for interpolation is their natural adaptation to use in higher dimensions. That flexibility is not lost when using an eigenfunction expansion to approximate the Gaussian, as was initially described in Section 3.2. The only new consideration to make in the multidimensional setting is that the truncation value M needs to be chosen for each dimension.

Because we have decided to make no preference between x and y dimensions in the choice of the shape parameter ε we fix the same truncation value M in all dimensions as well. Just as for the experiments in Section 7.1, for each value of ε represented in Figure 7.2, M is chosen as large as possible so that the condition of the system is less than 10^8, and then a is chosen to produce the lowest possible condition.

In Figure 7.2 we present examples of RBF-QRr compared to RBF-Direct for two different test functions. The data is generated by sampling these functions at N Halton points (see, e.g., [5]) in the unit square. As before, RBF-QRr loses accuracy for large values of the shape parameter ε because the necessary number of eigenfunctions to conduct a quality approximation is too great to complete a regression. For this region, it is not necessary to use RBF-QRr because RBF-Direct has acceptable condition, but it is worth noting that also the RBF-QRr method has its limitations.



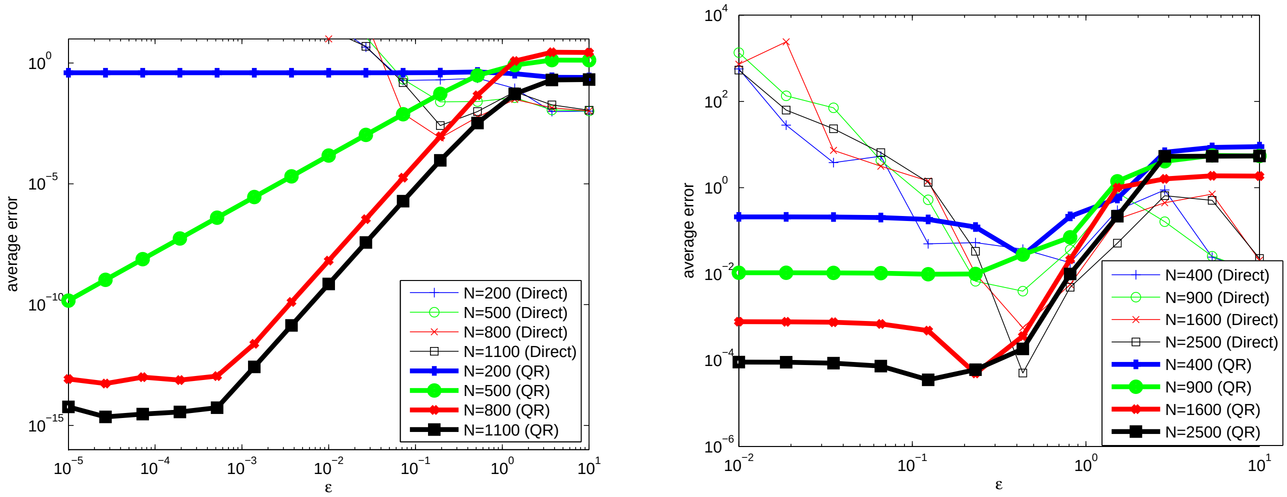
(a) $f(x, y) = \tanh(x + y)$

(b) $f(x, y) = \cos(x^2 + y^2)$

Fig. 7.2: Comparison of RBF-Direct and RBF-QRr regression in 2D using a different number, N, of Halton data points in $[-1, 1]^2$.

Just as in the 2D setting, eigenfunction expansions work for higher-dimensional settings as well. Figure 7.3 shows examples for two functions of five variables with very different ε profiles. As one would expect, the polynomial in Figure 7.3a is reproduced to within machine precision as soon as enough eigenfunctions are used and ε is chosen small enough (note that the dimension of the space of polynomials of degree five in five variables is 252). For the trigonometric test function illustrated in Figure 7.3b the RBF-QRr method is again more accurate and more stable than RBF-Direct. However, the overall accuracy of the approximation is more modest.

8. Conclusions. The stated purpose of this work was to provide a technique to allow for stable evaluation of RBF interpolants when the shape parameter values are so small that ill-conditioning overwhelms



(a) $f(u,v,w,x,y) = 1 + (u+v+w)^2(x-y)^2(u+x)$

(b) $f(u,v,w,x,y) = \cos(u+v) - \sin(v+w-x) - \cos(u-2y)$

Fig. 7.3: Comparison of RBF-Direct and RBF-QRr regression in 5D using a different number, N, of Halton data points in $[-1,1]^5$.

the traditional approach to RBF interpolation. This “flat-limit” regime is of particular practical interest since this often corresponds to the range of the shape parameter that will provide the most accurate RBF interpolant (provided it can be stably computed and evaluated). Our initial approach closely followed [12] replacing the use of spherical harmonics with an eigenfunction decomposition parametrized by a value a related to the global scale of the problem. This technique consists of replacing the Gaussian basis functions centered at the N data sites with $M > N$ new basis functions that reproduce the Gaussian kernel within the limits of machine precision. The choice of this new basis was driven by the desire to have condition properties superior to the Gaussian for sufficiently small ε, but it introduces some redundancy in the representation of the N-dimensional Gaussian approximation space which leads to some of the conditioning problems observed in Figure 5.2.

For certain values of N and ε this approach worked well, but for larger values of N we encountered limitations incurred by the condition of the eigenfunctions which were absent from the work involving spherical harmonics. To compensate for this new source of ill-conditioning (independent of the shape parameter) a new approach was devised involving $M < N$ basis functions and a least squares solution to the approximation problem. This technique overcame the ill-conditioning of the interpolation problem using careful choices of a and M to balance the condition of the problem against producing the best approximation to the space spanned by the Gaussians.

Given that we have seen the potential for success with this eigenfunction approximation of the Gaussian kernels, there is still much to be investigated to fully understand the work started here. We end by briefly discussing some of these topics.

8.1. Location of Data Points. In our work leading to this paper we have studied input data on an evenly spaced grid, on the Chebyshev points and at the Halton points. Overall we have noticed very little effect of the data point distribution on the condition and accuracy of the RBF-QR and RBF-QRr solutions provided the domain is “covered well” by the data points. Is this true in general; will the distribution of points have little or no significance on the effectiveness of RBF-QR and RBF-QRr?

8.2. Analytic Relationship of the Parameters ε, M and a. In Figure 6.3 we illustrated how the truncation value M and global scale parameter a affect the condition of the RBF-QRr regression algorithm for given values of ε and varying problem size N. Rigorous analysis needs to be done on the relationship between a and ε. Every Gaussian kernel with shape parameter ε has a family of equivalent eigenfunction expansions parametrized by a. In exact arithmetic with $M \to \infty$ all of these series are equal to the Gaussian kernel, but for a finite M there are significant differences which may lead to ill-conditioned systems. Can we determine analytically what a-value is appropriate for each M, or if there even always exists such a (unique) value?

8.3. Computational Cost. There is a significant computational cost involved in performing RBF interpolation with the RBF-QR algorithm when M is much greater than N. Likewise, determining the optimal values of M and a for the RBF-QRr regression method is costly and overwhelms the savings of solving an overdetermined $N \times M$ problem via least squares instead of an $M \times M$ problem (with $M \gg N$) for RBF-QR or an (ill-conditioned) $N \times N$ system for RBF-Direct. How can this cost be reduced? Having analytical guidance for "good" choices of M and a would certainly help.

8.4. Choice of Algorithm. The RBF-QRr regression method is built on the assumption that we use $M < N$ eigenfunctions. This limits its accuracy for large ε-values. As ε grows it is generally assumed that RBF-Direct will be well conditioned, which alleviates much of the demand for RBF-QRr in that range. Even so, is there a type of problem for which RBF-Direct is unreliable for an ε-value which is unacceptably inaccurate for RBF-QRr? Would that then force RBF-QR into action, requiring $M \gg N$? What guidelines might one use to build a general-purpose hybrid algorithm?

8.5. Anisotropic Approximation. In this paper we have worked under the assumption that the same shape parameter ε should be used in each space dimension. As mentioned in Section 3.2, the theoretical possibilities of our eigenfunction-based QR algorithms are much more general. We could choose to write the kernel as $K(\boldsymbol{x}, \boldsymbol{z}) = \exp((\boldsymbol{x} - \boldsymbol{z})^T \mathsf{E}(\boldsymbol{x} - \boldsymbol{z}))$, where the standard isotropic Gaussian would correspond to $\mathsf{E} = \varepsilon^2 \mathsf{I}_N$. The derivation in Section 3.2 provides a natural route to using eigenfunction expansions for anisotropic Gaussians (i.e., with E diagonal, but not a scalar multiple of the identity). However, in so doing there is also the opportunity to use a strategy to employ different choices of M and a in different dimensions. What flexibility and accuracy does this added freedom offer? How does this affect the complexity of the implementation and execution of the method? A different choice of E (still positive definite) would result in a different kernel and more flexibility when conducting interpolation in higher dimensions. Can the theoretical foundation be extended to cover eigenfunction decompositions for a nondiagonal E?

8.6. Other Kernels. The work in this paper focused on the Gaussian kernel. There are many other positive definite kernels (see, e.g., [5]) that involve a shape parameter for which the RBF-Direct method is associated with the trade-off principle, i.e., increased accuracy comes at the price of a loss in numerical stability. In [11] some ideas for the oscillatory Bessel or Poisson kernels are presented. What about inverse multiquadrics, Matérn kernels, and many others?

REFERENCES

[1] R. K. Beatson, W. A. Light and S. Billings, *Fast solution of the radial basis function interpolation equations: domain decomposition methods*, SIAM J. Sci. Comput. 22 (2000), pp. 1717–1740.

[2] C. de Boor, *On interpolation by radial polynomials*, Adv. Comp. Math., 24 (2006), pp. 143–153.

[3] H. A. Cho, C. S. Chen, and M.A. Golberg, *Some comments on mitigating the ill-conditioning of the method of fundamental solutions*, Engineering Analysis with Boundary Elements, 30 (2006), pp. 405–410.

[4] T. A. Driscoll and B. Fornberg, *Interpolation in the limit of increasingly flat radial basis functions*, Comput. Math. Appl., 43 (2002), pp. 413–422.

[5] G. E. Fasshauer, *Meshfree Approximation Methods in* Matlab, World Scientific Press, Singapore, 2007.

[6] G. E. Fasshauer, *Green's functions: taking another look at kernel approximation, radial basis functions and splines*, to appear in Approximation Theory XIII, San Antonio 2010.

[7] G. E. Fasshauer, F. J. Hickernell and H. Woźniakowski, *Rate of convergence and tractability of the radial function approximation problem*, submitted.

[8] G. E. Fasshauer and L. L. Schumaker, *Scattered data fitting on the sphere*, in Mathematical Methods for Curves and Surfaces II, M. Dæhlen, T. Lyche, and L. L. Schumaker, eds., Vanderbilt University Press, 1998, pp. 117–166.

[9] G. E. Fasshauer and Q. Ye, *Reproducing kernels of generalized Sobolev spaces via a Green function approach with distributional operators*, submitted.

[10] G. E. Fasshauer and Q. Ye, *Reproducing kernels of Sobolev spaces via a Green function approach with differential operators and boundary operators*, in preparation.

[11] B. Fornberg, E. Larsson and N. Flyer, *Stable computations with Gaussian radial basis functions in 2-D*, Technical Report 2009-020, Uppsala University, Department of Information Technology.

[12] B. Fornberg and C. Piret, *A stable algorithm for flat radial basis functions on a sphere*, SIAM J. Sci. Comp., 30 (2007), pp. 60–80.

[13] B. Fornberg and G. Wright, *Stable computation of multiquadric interpolants for all values of the shape parameter*, Comput. Math. Appl., 47 (2004), pp. 497–523.

[14] B. Fornberg and J. Zuev, *The Runge phenomenon and spatially variable shape parameters in RBF interpolation*, Comput. Math. Appl., 54 (2007), pp. 379–398.

[15] G. Golub and C. Van Loan, *Matrix Computations*, 3rd ed., Johns Hopkins Press, Baltimore, 1996.

[16] E. Larsson and B. Fornberg, *A numerical study of some radial basis function based solution methods for elliptic PDEs*, Comput. Math. Appl., 46 (2003), pp. 891–902.

[17] E. Larsson and B. Fornberg, *Theoretical and computational aspects of multivariate interpolation with increasingly flat radial basis functions*, Comput. Math. Appl., 49 (2005), pp. 103–130.

[18] Y. J. Lee, G. J. Yoon and J. Yoon, *Convergence of increasingly flat radia basis interpolants to polynomial interpolants*, SIAM J. Math. Anal., 39 (2007), pp. 537–553.

[19] J. Mercer, *Functions of positive and negative type, and their connection with the theory of integral equations*, Phil. Trans. Royal Soc. London Series A, 209 (1909), pp. 415–446.

[20] C. E. Rasmussen and C. Williams, *Gaussian Processes for Machine Learning*, MIT Press, Cambridge, MA, 2006 (online version at `http://www.gaussianprocess.org/gpml/`).

[21] R. Schaback, *Error estimates and condition numbers for radial basis function interpolation*, Adv. Comp. Math., 3 (1995), pp. 251–264.

[22] R. Schaback, *Convergence of unsymmetric kernel-based meshless collocation methods*, SIAM J. Numer. Anal., 45 (2007), pp. 333–351.

[23] R. Schaback, *Limit problems for interpolation by analytic radial basis functions*, J. Comp. Appl. Math., 212 (2008), pp. 127–149.

[24] R. Schaback, *Approximationsverfahren II (Lecture notes on kernel methods)*, Universität Göttingen, 2011, `http://num.math.uni-goettingen.de/schaback/teaching/AV_2.pdf`.

[25] E. Schmidt, *Über die Auflösung linearer Gleichungen mit unendlich vielen Unbekannten*, Rend. Circ. Mat. Palermo, 25 (1908), pp. 53–77.

[26] I. Steinwart and A. Christmann, *Support Vector Machines*, Springer, Berlin, 2008.

[27] J. F. Traub, G. W. Wasilkowski and H. Woźniakowski, *Information-Based Complexity*, Academic Press, 1988.

[28] H. Wendland, *Scattered Data Approximation*, Cambridge University Press, Cambridge, 2005.

[29] H. Zhu, C. K. Williams, R. J. Rohwer, and M. Morciniec, *Gaussian regression and optimal finite dimensional linear models*, in Neural Networks and Machine Learning, C. M. Bishop (ed.), Springer-Verlag, Berlin, 1998.