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 with a special focus on 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, Mercer's theorem, eigenfunction expansion, 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 $\{\mathrm{e}^{-\varepsilon^2\|\cdot-\boldsymbol{x}_j\|^2}, j = 1, \ldots, N\}$ becomes numerically linearly dependent on $\mathbb{R}^d$. 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, 30]). 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, 22]. This uncertainty principle is tied directly to the use of the standard ("wrong") basis, and we demonstrate here that it can be circumvented by choosing a better — orthonormal — basis.

The idea of using a "better basis" for radial basis function (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 [20] and [26]. 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 [13] for interpolation with zonal kernels on the unit sphere $\mathbb{S}^2$ and in [12] for interpolation with Gaussian kernels using products of Chebyshev polynomials and spherical harmonics. These latter two papers motivated the results presented here.

The main novelties presented in 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 positive scalars λ_n and functions φ_n. Here Ω can be a rather general set. 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 (see Section 3.2 for more details). While this new interpretation opens the door to finding stable evaluation algorithms for many different kernels (by providing

*Department of Applied Mathematics, Illinois Institute of Technology, `fasshauer@iit.edu`, partially supported by the National Science Foundation under Grants No. DMS-0713848 and DMS-1115392

†Center for Applied Mathematics, Cornell University, `mjm458@cornell.edu`

their eigenfunction expansions), we will focus in the following on $\Omega \subseteq \mathbb{R}^d$ and the Gaussian kernel (1.1). 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. This latter least-squares technique can again be transferred to any kernel whose eigenfunction expansion is available. For the Gaussian kernel, both of our algorithms extend in a natural way to anisotropic Gaussian kernels.

In the following we will discuss the expansion 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 RBF 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

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

Clearly, the functions φ_n, $n = 0, 1, 2, \ldots$, 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:

$$\int_{-\infty}^{\infty}\varphi_n^2(x)\mathrm{d}x = \int_{-\infty}^{\infty}x^{2n}\mathrm{e}^{-2\varepsilon^2x^2}\mathrm{d}x = \frac{\Gamma\left(n+\frac{1}{2}\right)}{(2\varepsilon^2)^{n+\frac{1}{2}}}.$$

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 [27]). This, however, does not seem to be practical. In fact, the authors of [12] claimed the series (2.1) was 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 for $\varepsilon \to 0$ the functions $x \mapsto x^n\mathrm{e}^{-\varepsilon^2x^2}$ converge to the standard monomial basis giving rise to the notoriously ill-conditioned Vandermonde matrix. Therefore, the authors of [12] followed up their initial expansion with a transformation to polar, or more generally spherical, coordinates so that their expansions end up being in terms of spherical harmonics. In addition, Chebyshev polynomials (which do represent an orthogonal eigenfunction basis — but only in the radial direction of the transformed coordinate system) were introduced to improve the numerical stability.

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 [20] (or alternatively Hilbert-Schmidt theory [26]) states that every positive definite kernel K can be represented in terms of the (positive) eigenvalues λ_n and (normalized) eigenfunctions φ_n of an associated compact integral operator (see (3.2) for an example), i.e.,

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

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

$$\varphi''(x) + \frac{1}{\lambda}\varphi(x) = 0$$
$$\varphi(0) = \varphi(1) = 0$$

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

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

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

$$(\mathcal{T}_K f)(x) = \int_0^1 K(x,z) f(z)\,\mathrm{d}z. \tag{3.2}$$

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, 10, 11].

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 in Section 3.2 in a straightforward manner using the product form of the Gaussian kernel.

It turns out that one can derive (see [21, 31]) 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

$$\rho(x) = \frac{\alpha}{\sqrt{\pi}}\,\mathrm{e}^{-\alpha^2x^2}, \qquad \alpha > 0. \tag{3.3}$$

This formulation ensures that the weight function has unit integral, and that the parameter α acts on the same scale as the shape parameter ε of the Gaussian kernel. Moreover, both of these parameters act as length scales for the spatial variable x and use the same units. Since the parameter α determines how the global domain $\mathbb{R}$ is "localized" we can interpret it as a *global scale parameter*.

In order to match up our choice of parameters with those used in [21], we replace the original parameters a, b, and $c = \sqrt{a^2+2ab}$ with our own parameters α, β (to be introduced below) and ε in the following way:

$$a = \frac{\alpha^2}{2}, \quad b = \varepsilon^2, \quad c = \frac{\alpha^2\beta^2}{2}.$$

Using this setup along with the following auxiliary parameters β, γ_n and δ defined in terms of α and ε, i.e.,

$$\beta = \left(1+\left(\frac{2\varepsilon}{\alpha}\right)^2\right)^{\frac{1}{4}}, \quad \gamma_n = \sqrt{\frac{\beta}{2^{n-1}\Gamma(n)}}, \quad \delta^2 = \frac{\alpha^2}{2}\left(\beta^2-1\right), \tag{3.4}$$

the eigenfunctions φ_n of the Gaussian turn out to be

$$\varphi_n(x) = \gamma_n \mathrm{e}^{-\delta^2x^2} H_{n-1}(\alpha\beta x), \qquad n = 1,2,\ldots, \tag{3.5a}$$

where H_{n-1} are 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

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

In order to get a perfect match of our formulas with those in [21], the reader needs to take into account the corrected normalization provided in the errata for [21]. It should also be noted that this formulation is related to Mehler's formula and rescaled Hermite *functions* [28, Problems and Exercises, Item 23]. However, in our work we do not directly employ Hermite functions (which are known to be the eigenfunctions of the Schrödinger equation) since our eigenfunctions include an extra exponential factor that can be attributed to the localization effects mentioned above.

The corresponding eigenvalues for the Gaussian are

$$\lambda_n = \sqrt{\frac{\alpha^2}{\alpha^2+\delta^2+\varepsilon^2}}\left(\frac{\varepsilon^2}{\alpha^2+\delta^2+\varepsilon^2}\right)^{n-1}, \quad n=1,2,\ldots. \tag{3.5b}$$

As already mentioned, α is related to the *global scale* of the problem, while the shape parameter ε is related to the *local scale* of the problem. In addition, the parameter δ also reflects the local scale of the problem. However, while ε gives us the scale of the kernel (which in turn defines the underlying reproducing kernel Hilbert space along with a length scale reflected in its norm), the auxiliary parameter δ reflects the length scale of the eigenfunctions.

In principle, the parameters α and ε (or α and δ) can be chosen freely. However, this choice is not totally independent if one wants a convergent and stable algorithm (see the discussion in Section 5.3 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 as this is the range of values of the shape parameter that incurs numerical instability, often with the promise of higher accuracy. It is our goal to circumvent this instability by working with eigenfunctions instead of the usual translates of the Gaussian kernel.

Note that for $\varepsilon \to 0$, i.e., for "flat" Gaussians, and fixed α we always have $\beta \to 1$ and $\delta \to 0$. We see that in this case the eigenfunctions φ_n converge to the normalized Hermite polynomials $\widetilde{H}_{n-1}(x) = \frac{1}{\sqrt{2^{n-1}\Gamma(n)}} H_{n-1}(\alpha x)$, and the eigenvalues behave like $\left(\frac{\varepsilon^2}{\alpha^2}\right)^{n-1}$. 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 [12, 13] can be employed as explained in Section 4.

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, 14, 17, 18, 19, 24]).

Looking at (3.5b), one can observe that the eigenvalues $\lambda_n \to 0$ exponentially fast as $n \to \infty$ since the inequality $\varepsilon^2 < \alpha^2 + \delta^2 + \varepsilon^2$ is always true. This idea was used in [7, 8] 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

$$\lambda_{\boldsymbol{n}} = \prod_{j=1}^{d} \lambda_{n_j} = \prod_{j=1}^{d} \sqrt{\frac{\alpha_j^2}{\alpha_j^2 + \delta_j^2 + \varepsilon_j^2}} \left(\frac{\varepsilon_j^2}{\alpha_j^2 + \delta_j^2 + \varepsilon_j^2}\right)^{n_j - 1}, \tag{3.6a}$$

$$\varphi_{\boldsymbol{n}}(\boldsymbol{x}) = \prod_{j=1}^{d} \varphi_{n_j}(x_j) = \prod_{j=1}^{d} \gamma_{n_j} \exp\left(-\delta_j^2 x_j^2\right) H_{n_j - 1}(\alpha_j \beta_j x_j), \tag{3.6b}$$

and $\boldsymbol{x} = (x_1, \ldots, x_d) \in \mathbb{R}^d$. Note that this formulation allows us to take different shape parameters ε_j and different integral weights α_j for different space dimensions (i.e., K may be an anisotropic kernel), or we can take them all equal, i.e., $\alpha_j = \alpha$ and $\varepsilon_j = \varepsilon$, $j = 1, \ldots, d$ (and then K is isotropic or radial). For the purposes of this work, we restrict ourselves to using the same α_j and ε_j in all dimensions, but in future work we plan to investigate the use of individual α_j and ε_j for each dimension.

4. A Stable Evaluation Algorithm. As already mentioned earlier, the starting point in [12] was an expansion of the form (2.1), which was deemed ineffective in overcoming the ill-conditioning associated with the use of the standard Gaussian kernel basis. If we instead use an eigenfunction expansion as discussed in the previous section, then the source of ill-conditioning can be separated from the eigenfunctions (which are orthogonal by definition) and transferred to 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 follow the QR-based strategy suggested for the spherical setting in [13].

4.1. The RBF-QR Algorithm. In particular, we now use the Gaussian kernel (1.1) along with its eigenvalues 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 (3.6). This matter is not a trivial one and needs to be dealt with carefully in the implementation (see some comments in Section 4.2.2). The QR-based algorithm of [12] 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 4.2.2, but assuming that an M has been chosen, 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 our specific choice of M is postponed until later, it is important to note that since it is our immediate goal to avoid the ill-conditioning associated with radial basis interpolation as $\varepsilon \to 0$, we require $M \geq N$. This is in accordance 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 computations — our continued use of an equality sign for the matrix factorization of K.

We are interested in obtaining a new basis in which 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 its n^{th} column contains only values of the n^{th} eigenfunction at all the data sites $\boldsymbol{x}_1, \dots, \boldsymbol{x}_N$. 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 that

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

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

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

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 $\alpha^2 + \delta^2$). 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. The orthogonality of the eigenfunctions φ_n, $n = 1, \dots, M$, ensures the nonsingularity of R_1 and thus a good choice for the matrix X is given by

$$
\mathsf{X} = \Lambda_1 \mathsf{R}_1^T \mathsf{Q}^T. \qquad (4.3)
$$

We are now interested in the new system defined by

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

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:

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

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

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

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

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

only now the ill-conditioning related to $\mathsf{\Lambda}_1$ 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 [13] 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., [9]).

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

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

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 $\boldsymbol{x}_1, \dots, \boldsymbol{x}_N$, and $\boldsymbol{c}$ 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

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

where the matrix $\mathsf{\Psi}$ was defined in (4.4), $\boldsymbol{y}$ is as above, and $\boldsymbol{b}$ 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

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

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

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

where, as before

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

and

$$\mathsf{\Lambda} = \begin{pmatrix} \mathsf{\Lambda}_1 & \\ & \mathsf{\Lambda}_2 \end{pmatrix}$$

with the first block $\mathsf{\Lambda}_1$ of size $N \times N$.

4.2.1. Some linear algebra details. 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:

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

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

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

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 stored during earlier work and the cost of $\mathcal{O}(N^2(M-N))$ can be saved.

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

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

$$\left[\mathsf{I}_N + \mathsf{R}_1^{-1}\mathsf{R}_2\Lambda_2\mathsf{R}_2^T\mathsf{R}_1^{-T}\Lambda_1^{-1}\right]\boldsymbol{b} = \hat{\boldsymbol{b}}. \tag{4.9b}$$

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, i.e.,

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

As mentioned earlier, here the diagonal scaling by Λ_1^{-1} and Λ_2 is performed analytically.

4.2.2. Choosing the length of the eigenfunction expansion. Up until now we have treated the truncation length M as a fixed but unknown quantity. For the RBF interpolation algorithm discussed thus far, our choice of M coincides with that of [13], where M is chosen as the smallest value that satisfies $\lambda_M < \epsilon_{\text{mach}}\lambda_N$. Here ϵ_{mach} is machine precision (assumed to be 10^{-16}) and λ_n is defined in (3.5b). Solving this inequality for M produces

$$\begin{aligned}\sqrt{\frac{\alpha^2}{\alpha^2+\delta^2+\varepsilon^2}}\left(\frac{\varepsilon^2}{\alpha^2+\delta^2+\varepsilon^2}\right)^{M-1} &< \epsilon_{\text{mach}}\sqrt{\frac{\alpha^2}{\alpha^2+\delta^2+\varepsilon^2}}\left(\frac{\varepsilon^2}{\alpha^2+\delta^2+\varepsilon^2}\right)^{N-1} \\ \left(\frac{\varepsilon^2}{\alpha^2+\delta^2+\varepsilon^2}\right)^{M-N} &< \epsilon_{\text{mach}} \\ (M-N)\log\left(\frac{\varepsilon^2}{\alpha^2+\delta^2+\varepsilon^2}\right) &< \log(\epsilon_{\text{mach}}) \\ M > N + \log(\epsilon_{\text{mach}})&\left(\log\left(\frac{\varepsilon^2}{\alpha^2+\delta^2+\varepsilon^2}\right)\right)^{-1}.\end{aligned} \tag{4.10}$$

This bound is derived in 1D, although it extends naturally to multiple dimensions. In doing so, the uniqueness of the eigenfunction expansion is lost because there may be several multi-indices $\boldsymbol{M}$ which satisfy the inequality. Rederiving this for a d-dimensional problem using $|\boldsymbol{M}| = \sum_j M_j$ and (3.6a) produces

$$\begin{aligned}\left(\frac{\alpha^2}{\alpha^2+\delta^2+\varepsilon^2}\right)^{d/2}\left(\frac{\varepsilon^2}{\alpha^2+\delta^2+\varepsilon^2}\right)^{|\boldsymbol{M}|-d} &< \epsilon_{\text{mach}}\left(\frac{\alpha^2}{\alpha^2+\delta^2+\varepsilon^2}\right)^{d/2}\left(\frac{\varepsilon^2}{\alpha^2+\delta^2+\varepsilon^2}\right)^{|\boldsymbol{N}|-d} \\ \left(\frac{\varepsilon^2}{\alpha^2+\delta^2+\varepsilon^2}\right)^{|\boldsymbol{M}|-|\boldsymbol{N}|} &< \epsilon_{\text{mach}} \\ |\boldsymbol{M}| > |\boldsymbol{N}| + \log(\epsilon_{\text{mach}})&\left(\log\left(\frac{\varepsilon^2}{\alpha^2+\delta^2+\varepsilon^2}\right)\right)^{-1}.\end{aligned} \tag{4.11}$$

We provide a simple example to illustrate the multivariate case. Suppose $d = 2$, $\varepsilon^2/(\alpha^2+\delta^2+\varepsilon^2) = \mathrm{e}^{-1}$, $N = 5$, and $\epsilon_{\text{mach}} = 10^{-16}$. This yields the inequality

$$|\boldsymbol{M}| > |\boldsymbol{N}| + 16.$$

When trying to determine $\boldsymbol{N}$ to separate the eigenfunctions from the correction we need to consider the indices of the dominating eigenvalue, i.e.,

$$\begin{array}{lcccccccccc} N_1 & = & 1 & 2 & 1 & \mathbf{3} & \mathbf{2}^* & \mathbf{1} & 4 & \ldots \\ N_2 & = & 1 & 1 & 2 & \mathbf{1} & \mathbf{2}^* & \mathbf{3} & 1 & \ldots \end{array}$$

Here we have marked the entries that correspond to the fifth index with a star. The eigenfunctions corresponding to these entries are included in the $\boldsymbol{\Phi}_1$ matrix and any later terms are pushed to the correction. Note, however, that there are several entries (marked in bold) that correspond to eigenvalues of the same magnitude as the fifth index, and we could just as easily have chosen these for the $\boldsymbol{\Phi}_1$ matrix instead. This exposes a non-uniqueness, but we do not believe this to be a problem as similar non-uniqueness results were described in [12].

Since we chose $N = 5$ for this example we see that $|\boldsymbol{N}| = 4$, and thus we need to select $|\boldsymbol{M}| > 20$ to meet our truncation criterion. Of course, thanks to Pascal's triangle, we know that there are 19 different values which satisfy $|\boldsymbol{M}| = 20$ meaning that the number of eigenfunctions needed in the correction $\boldsymbol{\Phi}_2$ may be quite significant.

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. All of our numerical experiments were performed in Matlab, using the built-in QR factorization and triangular solvers for the RBF-QR results. A Matlab implementation of a direct solver via the "backslash" operator \ (`mldivide`) was used to produce the RBF-Direct results for comparison. The polynomial fits were generated with the Matlab function `polyfit`. The software used for these numerical experiments is available online at

`http://math.iit.edu/~mccomic/gaussqr`.

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 $[x_a, x_b]$, i.e.,

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

The interpolation is conducted using the eigenfunction-QR algorithm (abbreviated RBF-QR) over shape parameter values logarithmically spaced within $\varepsilon \in [10^{-2}, 10^{0.4}]$. 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 y_i are obtained by sampling a function f at the points x_i and the error in the interpolant s is computed by

$$\text{error} = \frac{1}{\bar{N}}\sqrt{\sum_{k=1}^{\bar{N}}\left[\frac{f(\bar{x}_k) - s(\bar{x}_k)}{f(\bar{x}_k)}\right]^2}, \tag{5.2}$$

where the $\bar{x}_k$ 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

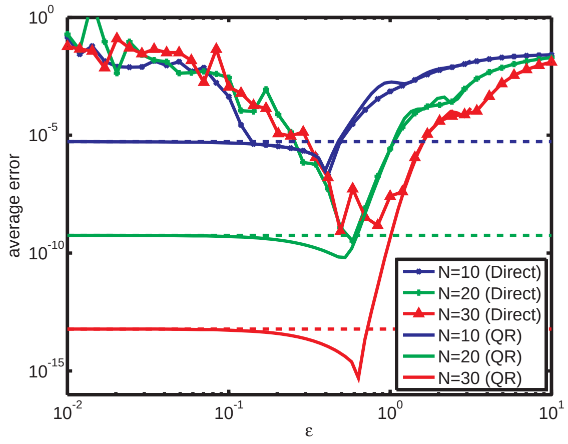
$$f_1(x) = \frac{\sinh x}{1 + \cosh x}, \qquad x \in [-3, 3],$$

using $N = \{10, 20, 30\}$ and $\alpha = 1$, which can be seen in Figure 5.1a. The second function considered was

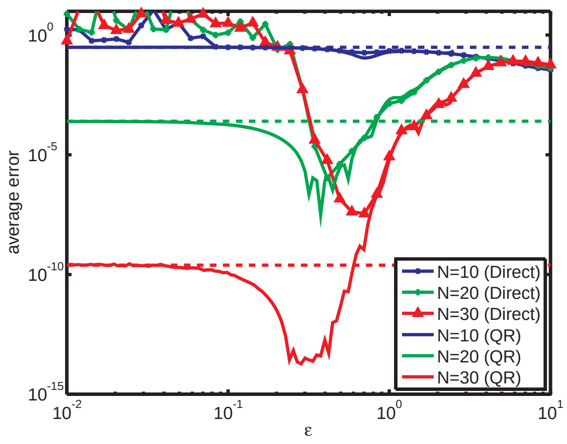
$$f_2(x) = \sin\left(\frac{x}{2}\right) - 2\cos x + 4\sin(\pi x), \qquad x \in [-4, 4],$$

using $N = \{10, 20, 30\}$ and $\alpha = .65$, which can be seen in Figure 5.1b.

These initial results confirm that for $\varepsilon \to 0$ the RBF-QR algorithm evaluates the Gaussian interpolant without the ill-conditioning associated with the use of RBF-Direct. Note that two different choices of α were used for these two problems. We will elaborate on this further in later sections.



(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.

The two examples presented here also illustrate that interpolation with Gaussian kernels may be 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. We point out that in some cases the Gaussian kernel (for $\varepsilon \gg 0$) beats the polynomial interpolant by several orders of magnitude.

5.2. 2D Interpolation. In Figure 5.2 we show the errors for a series of interpolation experiments with different values of ε for a small 2D problem involving the function

$$f_3(x,y) = \cos(x^2+y^2), \qquad (x,y) \in [-3,3]^2,$$

sampled on a tensor product grid of Chebyshev nodes. The RBF-QR scheme works as it should, but the

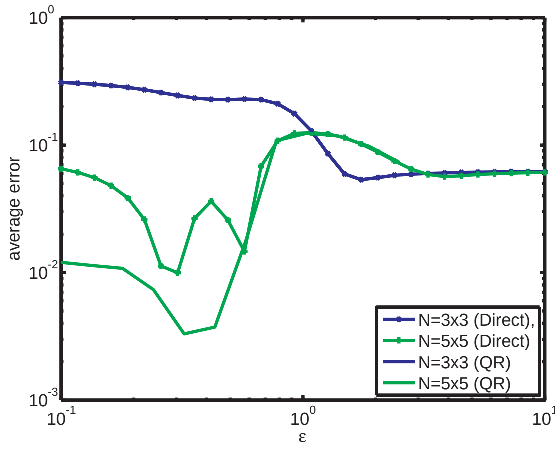


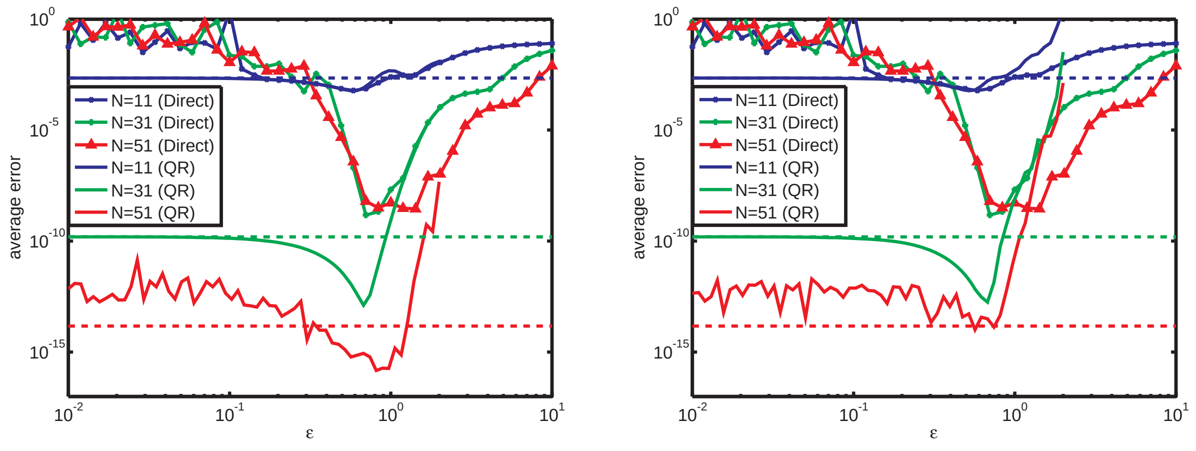
Fig. 5.2: RBF-QR is able to resolve the interpolant to f_3 accurately as $\varepsilon \to 0$.

computational cost is quite significant, which is why fewer ε values were considered for RBF-QR than in the previous 1D graphs. This cost issue will be discussed further in Section 6.

5.3. Complications with the Interpolation Algorithm. Although the previous experiments successfully illustrate the usefulness of the eigenfunction expansion for the solution of the RBF interpolation problem, Figure 5.3 exposes some subtle complexities of our RBF-QR interpolation algorithm when we perform a similar test with the function

$$f_4(x) = 10e^{-x^2} + x^2, \qquad x \in [-3, 3].$$

Specific attention should be paid to the effect of increasing N on the emerging oscillations in the error of the interpolant.



(a) $\alpha = 1$ produces bad results for small ε and larger N.

(b) $\alpha = 3$ produces bad results for large ε.

Fig. 5.3: Different values of α have a significant effect on the stability of the interpolation to f_4 at N Chebyshev points in $[-3, 3]$.

This new source of error is separate from the instability encountered in the $\varepsilon \to 0$ limit, although it has similar roots. Recall the structure of the eigenfunctions from earlier:

$$\varphi_n(x) = \gamma_n \mathrm{e}^{-\delta^2 x^2} H_{n-1}(\alpha\beta x). \tag{3.5a}$$

In much the same way as ε is a significant source of ill-conditioning for the original Gaussian basis, the value of δ (and therefore α, since we defined δ in terms of α and ε, see (3.4)) may now be a source of ill-conditioning for the eigenfunction basis.

The reader may recall that the interpolation problems under consideration all exist on a compact domain, but the orthogonality properties of the eigenfunctions demand integration to infinity. The choice of α is a balancing act between interacting eigenfunctions to achieve accuracy (small α) and quickly decaying eigenfunctions to numerically maintain orthogonality on the compact domain (large α).

To better understand what is going on here, we fix α and analyze two limits:

$$\text{as } \varepsilon \to 0, \qquad \beta \to 1, \qquad \delta^2 \to \varepsilon^2,$$
$$\text{as } \varepsilon \to \infty, \qquad \beta \to \sqrt{\frac{2\varepsilon}{\alpha}}, \qquad \delta^2 \to \alpha\varepsilon.$$

We then consider the eigenfunctions in these two limiting situations, i.e.,

$$\lim_{\varepsilon \to 0} \varphi_n(x) = \gamma_n \mathrm{e}^{-\varepsilon^2 x^2} H_{n-1}(\alpha x),$$
$$\lim_{\varepsilon \to \infty} \varphi_n(x) = \gamma_n \mathrm{e}^{-\alpha\varepsilon x^2} H_{n-1}(\sqrt{2\alpha\varepsilon}x).$$

In the $\varepsilon \to 0$ case, the effects of the two parameters ε and α are decoupled and each can be handled according to its needs (ε for accuracy and α for orthogonality). When $\varepsilon \to \infty$ this is no longer the case, and both the exponential and polynomial portions of the eigenfunctions exist on the same scale $\sqrt{\alpha\varepsilon}x$. While this analysis does provide an argument against using RBF-QR for larger values of ε, we remind the reader that this case is really of no practical importance since in this case the RBF-Direct method can be applied to great satisfaction.

To gain some more insight into the choice of α, we study the eigenfunctions graphically on the domain $[-4, 4]$ in Figure 5.4. The eigenfunctions corresponding to $\alpha = .1$ exhibit virtually no exponential decay on

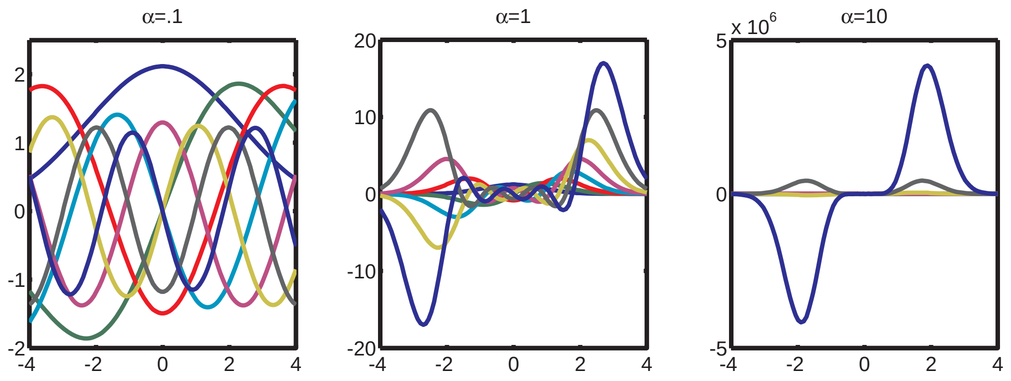


Fig. 5.4: The first 8 eigenfunctions evaluated when $\varepsilon = 1$ behave very differently for different values of α.

this domain, and therefore their orthogonality is not well preserved on $[-4, 4]$. For $\alpha = 10$, on the other hand, the eigenfunctions exist on drastically different scales, meaning that effectively only the very largest eigenfunctions contribute to the solution since all the smaller functions are indistinguishable from each other. When $\alpha = 1$ there is a "good" balance of locality and distinction for the eigenfunctions.

This discussion has been entirely qualitative, but it is meant to explain holistically why there should be an optimal value for α which produces the true RBF interpolant for any given ε. In Figure 5.5 we provide some additional computational evidence that there is an α (indicated by a small circle) for each ε to alleviate the instability shown in Figure 5.3.

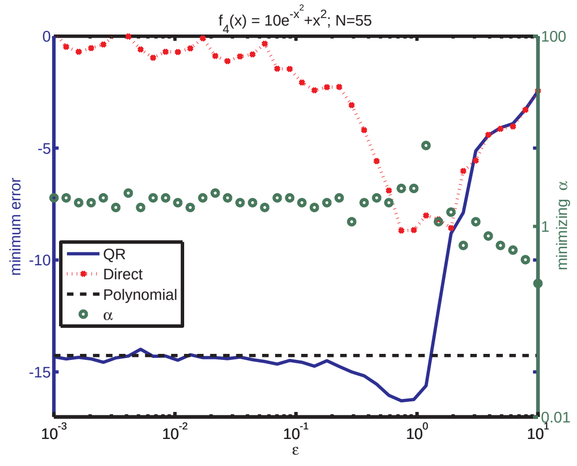


Fig. 5.5: A "good" value of α can be found for each ε to produce an accurate interpolant to f_4 at $N = 55$ Chebyshev points in $[-3, 3]$.

These results show that, given a set of data points $x_1, \ldots, x_N$, a function f and an ε for which you want to produce a Gaussian interpolant, there should exist an α that lets you accomplish this without suffering from the ill-conditioning associated with the $\varepsilon \to 0$ "flat limit". Furthermore, we see that the "optimal" choice of α in the small ε regime is rather stable, while — as ε exits the asymptotically small regime — the choice of an "optimal" α becomes a function of ε, as we predicted earlier when we discussed the $\varepsilon \to \infty$ scenario.

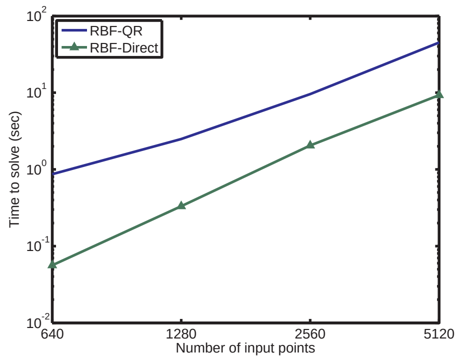
Unfortunately, these results do not describe an approach to choosing the appropriate α. They only suggest the existence of an "optimal" α, and that there may be some relationship to ε. Since in this work we are only interested in exploration of the $\varepsilon \to 0$ regime in which the choice of α is less sensitive, we will not pursue this issue further. Determining an appropriate α a priori, and doing so efficiently, will certainly be of significance in future work.

6. Early Truncation.

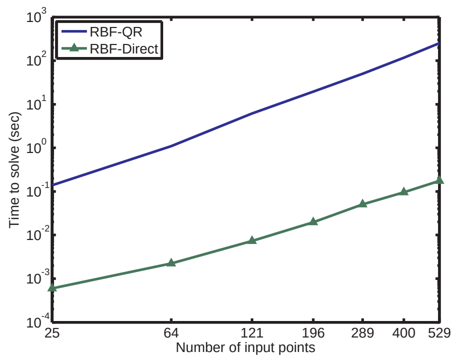
RBF-QR allows us to stably compute with shrinking ε values for which RBF-Direct is too ill-conditioned. Unfortunately, the cost associated with RBF-QR grows substantially with increasing ε and in multiple dimensions. For larger values of ε, RBF-Direct is a viable option, but the increased cost of RBF-QR in multiple dimensions is unavoidable — even for small values of ε. This is a direct result of our definition of the truncation length $\boldsymbol{M}$ in Section 4.2.2. Assuming that all the combinations of eigenfunctions which satisfy (4.11) are needed, the total number of columns in the matrix Φ, for a 2D problem, is

$$\sum_{k=1}^{|\boldsymbol{M}|} k = \frac{|\boldsymbol{M}|(|\boldsymbol{M}|+1)}{2}.$$

For comparison, this means that a 2D interpolation problem based on $N = 25$ points with $\varepsilon = .1$ and $\alpha = 1$ will result in $|\boldsymbol{N}| = 7$ and $|\boldsymbol{M}| = 15$, and thus require 120 eigenfunctions. If we take $\varepsilon = 1$, then $|\boldsymbol{M}| = 45$ and we need 1035 eigenfunctions to approximate an interpolation problem based on only $N = 25$ data points.



(a) In 1D, the cost of RBF-QR is about half an order of magnitude more than that of RBF-Direct.



(b) In 2D, the cost of RBF-QR is multiple orders of magnitude more than that of RBF-Direct.

N	M
640	645
1280	1285
2560	2565
5120	5125

(c) In 1D, the truncation length M for the eigenfunction corrections remains reasonable.

N	M
25	435
64	2346
121	7875
196	20100
289	43071
400	81810
529	142311

(d) In 2D, the cost of storing the eigenfunction corrections becomes unreasonable.

Fig. 6.1: For ε only as large as .01, the cost of RBF-QR becomes unsustainable in higher dimensions.

In Figure 6.1 we illustrate how the cost of performing RBF-QR using the truncation strategy outlined in Section 4.2.2 is unreasonable for dimension $d = 2$ (and even more so in higher dimensions). Here cost is defined as the time required to find the coefficients associated with the interpolation; associated with this cost is also the memory required to perform the interpolation as shown in Figures 6.1c and 6.1d. We use a value of $\varepsilon = .01$ together with several other values of N. While the cost of doing RBF-QR in 1D is reasonable, and the payoff in terms of accuracy gain is high, this is no longer true in 2D. Clearly, we will benefit from using some other approach in higher dimensions.

The dominant cost in performing the RBF-QR algorithm is associated with the QR factorization of the matrix Φ, which is more expensive than the LU factorization used to solve the RBF-Direct system, although on the same order when $M \approx N$. Due to the aforementioned explosion in M as the dimension increases, it is not viable to use RBF-QR for high-dimensional problems. To combat this, we now explore the feasibility of choosing a truncation length M which is smaller than N.

6.1. Low-Rank Approximation. Our goal for this section is to produce a low-rank approximation to the N-term RBF interpolant using $M < N$ eigenfunctions. The motivation behind this is to eliminate high-order eigenfunctions which contribute minimally to the solution, but greatly to the computational cost. Additionally, this may reduce the sensitivity of the solution to α as shown in Figure 5.3. The discussion from Section 5.3 suggests that the choice of an "optimal" α depends on ε and is also more sensitive with increasing M. We therefore hope that reducing M will help mitigate this sensitivity issue.

In order to introduce this problem in the same context as Section 4 we assume that $M \leq N$ is fixed and set all the eigenvalues λ_n with $M < n \leq N$ 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 [16]

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

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

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

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.,

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

This, of course, requires forming $\Phi^{-1}\boldsymbol{y}$, which would subject this problem to the same sensitivity issues as before that stem from the unreasonably large M values given increasing N and/or ε. Moreover, inverting the matrix Φ would be more costly than is necessary since the final solution will be a least-squares solution of rank at most M, whereas Φ has rank N. Instead, it seems that the more efficient and logical method of solving this system is to perform the matrix-matrix multiplication $\Phi\Lambda\Lambda^\dagger$ analytically, leaving the system

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

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

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

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

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

and $\boldsymbol{b}_2$ is unconstrained because the eigenfunctions associated with $\boldsymbol{b}_2$ 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 significant in reducing the computational complexity of the approximation, but the most important factor in choosing M is producing a quality approximation.

To give an initial idea of the effect of M on the quality of the approximation, let us consider a simple approximation problem. Given $f(x) = \cos x + \mathrm{e}^{-(x-1)^2} + \mathrm{e}^{-(x+1)^2}$ and fixing $\alpha = \varepsilon = 1$, we consider N evenly spaced values of f in the interval $[-3, 3]$. Different values of N ranging from 10 to 500 are chosen to conduct the regression with five different sets of M-values corresponding to $\{.1N, .2N, .3N, .4N, .5N\}$. The approximation error curves are displayed in Figure 6.2.

Regardless of the size of N, Figure 6.2 shows that the optimal accuracy of the approximation consistently occurs for the same value of $M \approx 40$. This is encouraging because it suggests that for a fixed ε, given any problem size N, the underlying test function f has a certain complexity in the native reproducing kernel Hilbert space of the Gaussian which is captured to within machine precision by using an approximation (sub-)space $\mathcal{R}$ of dimension $M \approx 40$. As was shown in [7], and has been known since well before (see, e.g., [29]), if one is allowed to optimally sample the function f, then the best M-term L_2 approximation from

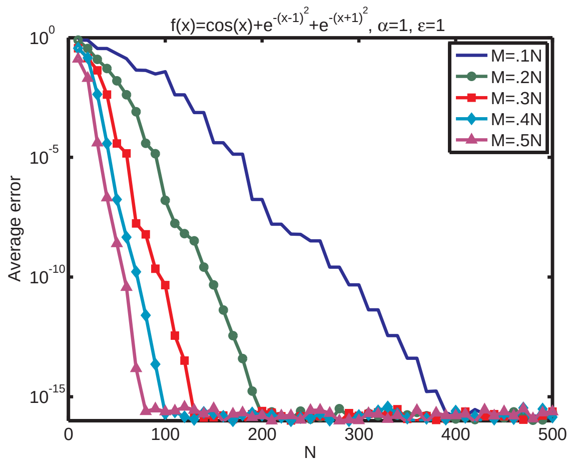


Fig. 6.2: For any number N of data sites (and fixed $\alpha = \varepsilon = 1$), $M \approx 40$ eigenfunctions are adequate for optimal accuracy of the QR regression algorithm.

the reproducing kernel Hilbert space associated with the Gaussian kernel is given by a linear combination of its first M eigenfunctions. However, for this optimal approximation, the data needs to be given in the form of the "Fourier coefficients" of the eigenfunctions, i.e., the L_2 projections of the test function onto the eigenfunctions. The problem we consider here is of a slightly different nature, since the data is given in terms of function values of f at certain data locations $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$, and we are determining a discrete least squares approximation to the given values $f(\boldsymbol{x}_1), \ldots, f(\boldsymbol{x}_N)$ from the space $\mathcal{R}$. While we do not in general know what the "correct" dimension of the approximation space $\mathcal{R}$ is, we will assume that we are using a "good" value in our experiments (often based on a trial and error approach starting from a small value of M and increasing as deemed necessary to achieve a desired accuracy).

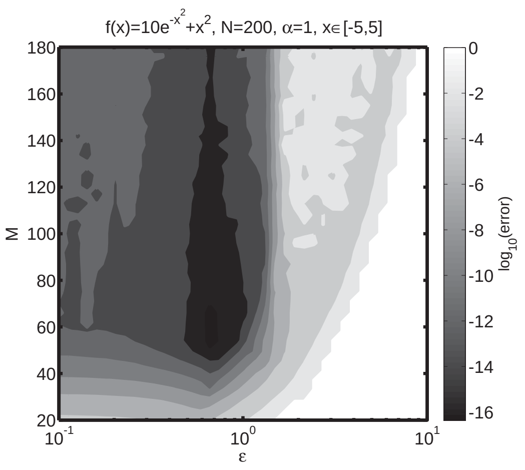
The fact that the effective dimension of the space needed for an accurate kernel approximation or interpolation is often rather small seems to be *mathematical folklore* and appears in different guises in various communities dealing with kernels. In the RBF literature we have, e.g., [15, 23] or the unpublished lecture notes [25]. In [25], 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 [31] 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 6.3a. The contour plot there shows the approximation error, for fixed $N = 200$ and $\alpha = 1$, obtained with values of M and ε that vary independently. The function used to generate the data for this approximation problem is

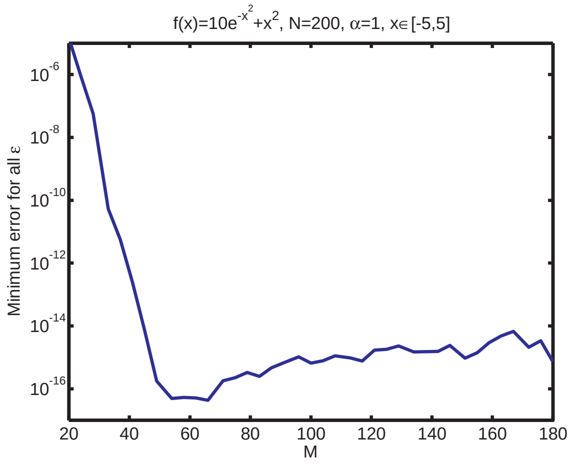
$$f_4(x) = 10e^{-x^2} + x^2, \qquad x \in [-5, 5].$$

The function f_4 is useful because in the absence of the exponential function term, the polynomial alone would be best approximated with M on the same order as the polynomial and $\varepsilon \to 0$, i.e., the "flat" polynomial limit of the RBF interpolant [5]. The additional exponential term gives rise to a region of optimal M and ε centered around $(M, \varepsilon) \approx (66, 0.7)$ far from the $\varepsilon = 0$ axis. Also note that the M with least error is far away from the RBF-QR realm of $M > N$, although the difference in accuracy between the optimal error at $M = 66$ overall and the optimal error for $M = 180$ is small at $10^{-16.4}$ and $10^{-15.1}$, respectively. Here the error is computed with (5.2) by evaluating the interpolant at 1000 points in $[-5, 5]$.

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



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



(b) Beyond $M \approx 70$ the optimal error increases.

Fig. 6.3: Over a range of ε values (with fixed N and $\alpha = 1$), experiments show an optimal M range.

of α for (3.5b) 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 has already been chosen.

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 length M used in the regression is specified.

For the following experiments, multiple M values were sampled and one was chosen to represent RBF-QRr. When experiments include RBF-QR results, the required M values were chosen via (4.10). For each ε, α was chosen as small as possible such that orthogonality is guaranteed for the first 25 eigenfunctions. The reader should keep in mind that these are just choices made to display the usefulness of RBF-QRr when exploring the $\varepsilon \to 0$ region, and more work needs to be done to determine optimal M and α values in general.

7.1. 1D Approximation.

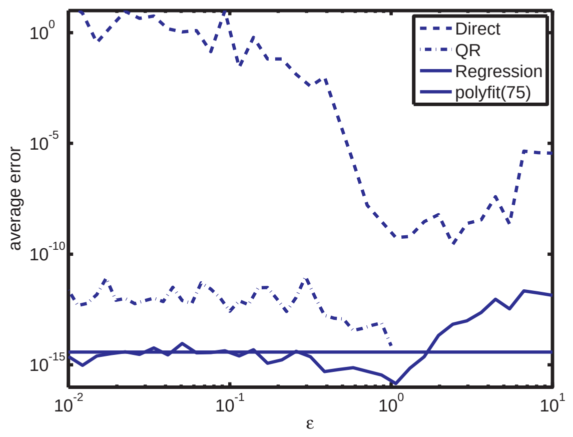
In this series of experiments the data is generated by two different univariate functions f evaluated at $N = 150$ Chebyshev nodes (5.1) on their respective domains.

For Figure 7.1a we reprise the function from Figure 6.2 on the interval $[-3, 3]$ and see that better accuracy can be achieved with RBF-QRr *approximation* instead of RBF-QR *interpolation*, even at less cost. In Figure 7.1b RBF-QRr maintains higher accuracy than both RBF-QR and RBF-Direct. Note that the function used in Figure 7.1b is the notorious Runge function on the interval $[-4, 4]$ and is much harder to approximate by polynomials than the function used in Figure 7.1a. In fact, we can see that the MATLAB algorithm `polyfit` for degree 90 polynomials is no longer stable and the "flat limit" Gaussian approximation, which uses orthogonal polynomials instead of a Vandermonde matrix, is considerably more accurate than the polynomial interpolant. Moreover, the Gaussian approximation for $\varepsilon \approx 1.5$ is many orders of magnitude more accurate than the polynomial interpolant.

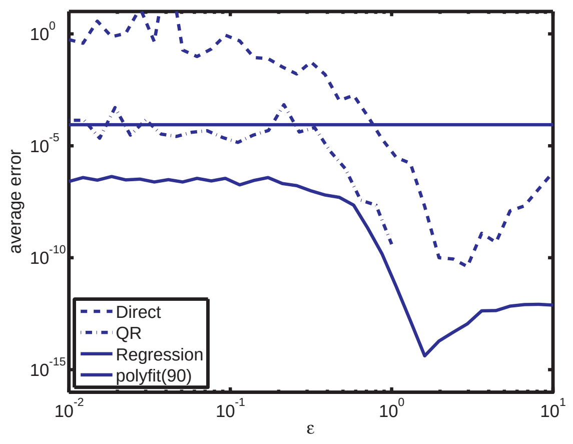
Looking at these two graphs, the reader will notice that RBF-QRr fails to reproduce RBF-Direct as ε grows and RBF-Direct is sufficiently well-conditioned. This is because the eigenvalues decay more slowly for larger ε and thus the choice of a small value of M may no longer be appropriate. One should require $M > N$ to conduct the approximation as $\varepsilon \to \infty$. However, there would be no reason to use RBF-QR or RBF-QRr in this regime since RBF-Direct is stable and more efficient.

7.2. Higher-dimensional Approximation.

One of the great benefits of considering radial basis functions for scattered data fitting 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



(a) $f(x) = \cos x + \mathrm{e}^{-(x-1)^2} - \mathrm{e}^{-(x+1)^2}$, $N = 150$, $M = 75$



(b) $f(x) = (1 + x^2)^{-1}$, $N = 150$, $M = 90$

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

Section 3.2. For the experiments in this section only regression is considered because of the computational cost associated with the use of RBF-QR in higher dimensions. The scale parameter α was chosen to satisfy orthogonality for up to the fourth eigenfunction. This choice is somewhat arbitrary, but seems to produce good results in the $\varepsilon \to 0$ region of interest.

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 evenly spaced points in the region $[-1, 1]^2$. As before, RBF-QRr drifts further from the true RBF interpolant for large values of the shape parameter ε because the number of eigenfunctions required to conduct a more accurate approximation is too great to complete the regression efficiently. As mentioned several times before, for this ε-regime it is not necessary to use RBF-QRr since RBF-Direct has acceptable condition, but it is worth noting that also the RBF-QRr method has its limitations. In the small-ε regime, on the other hand, RBF-QRr performs stably and accurately as promised.

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 total 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 accuracy of the approximation is weak for larger ε, as was noted previously. These functions were sampled at N Halton points (see e.g., [5]) and the error was evaluated at 4000 Halton points; the first N error evaluations were ignored because those points were input points due to the nesting property of Halton point sets.

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 the traditional approach to RBF interpolation. This "flat-limit" regime is of particular practical interest since it 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 [13] replacing the use of spherical harmonics with an eigenfunction decomposition parametrized by a value α 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

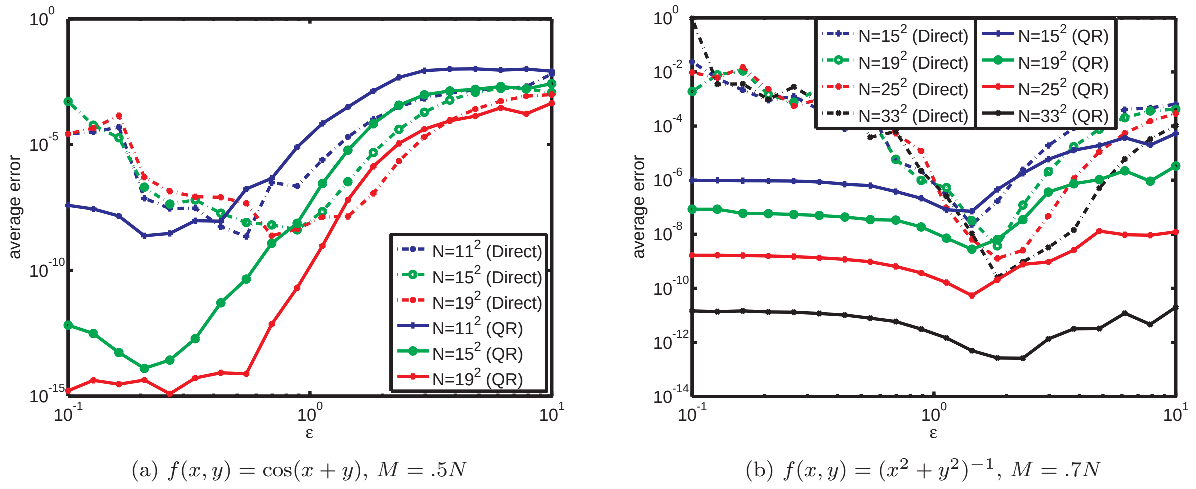


Fig. 7.2: Comparison of RBF-Direct and RBF-QRr regression in 2D using various evenly spaced data points in $[-1, 1]^2$.

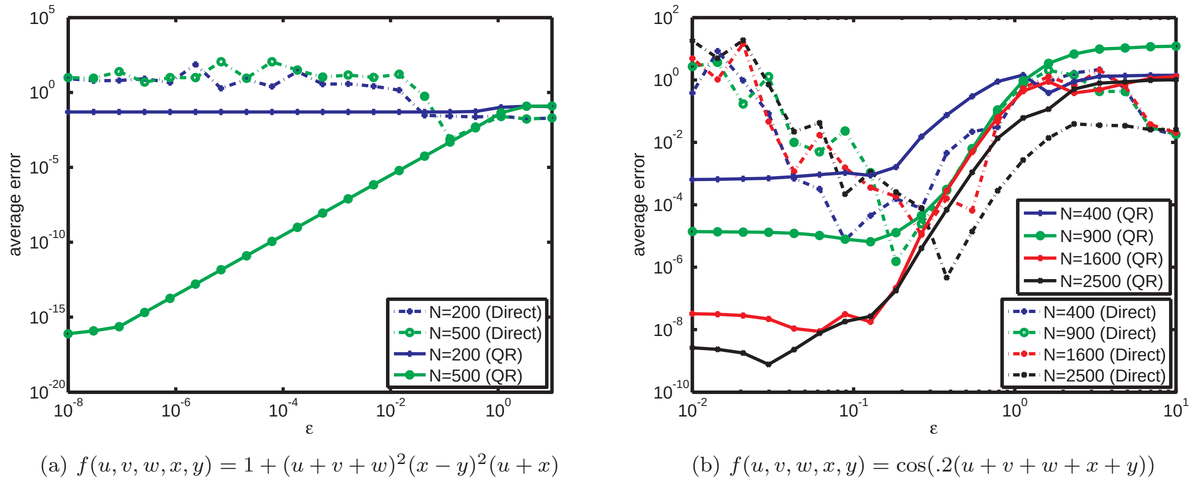


Fig. 7.3: Comparison of RBF-Direct and RBF-QRr regression in 5D using a different number, N, of Halton data points in $[-1, 1]^5$.

the N-dimensional Gaussian approximation space which leads to some of the conditioning problems observed in Figure 5.3.

For certain values of N and ε this approach worked well, but for larger values of N we encountered limitations incurred by the computational expense of the algorithm. To compensate for this, 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, but more or less ad hoc, choices of α and M to balance the cost and sensitivity of the solution against producing the best approximation from 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. While the point distribution is important for RBF-QR solutions, we have noticed very little effect of the data point distribution on the condition and accuracy of the RBF-QRr solution provided the domain is "covered well" by the data points. Is this true in general, i.e., will the distribution of data points have little or no significance on the effectiveness of RBF-QRr? The references [3, 15, 23, 25, 31] mentioned earlier in our disussion of the "correct" approximation subspace $\mathcal{R}$ may suggest that this should indeed be true, but rigorous proof of this fact is still missing.

8.2. Analytic Relationship of the Parameters ε, M and α. Rigorous analysis needs to be done on the relationship between α and ε. Every Gaussian kernel with shape parameter ε has a family of equivalent eigenfunction expansions parametrized by α. 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 α-value is appropriate for each M, or if there even always exists such a (unique) value? Or, should we first determine a "best" value of α and then find the suitable M?

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 (especially in higher space dimensions). Likewise, determining the optimal values of M and α 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 α 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 α 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 [12] some ideas for the oscillatory Bessel or Poisson kernels are presented. What about inverse multiquadrics, Matérn kernels, and many others?

Acknowledgment. We would like to thank our colleague, Fred Hickernell, for fruitful discussions on this topic and the anonymous reviewers for their valuable comments which helped us improve the final version of the paper.

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. Comput. Math., 24 (2006), pp. 143–153.

[3] C. S. Chen, H. A. Cho, and M. A. Golberg, *Some comments on the ill-conditioning of the method of fundamental solutions*, Eng. Anal. Bound. Elem., 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 with* Matlab, World Scientific Publishing Co., Inc., River Edge, NJ, USA, 2007.

[6] ———, *Green's functions: Taking another look at kernel approximation, radial basis functions, and splines*, in Approximation Theory XIII: San Antonio 2010, M. Neamtu and L. L. Schumaker, eds., Springer Proceedings in Mathematics Series, Springer, 2012, pp. 37–63.

[7] G. E. Fasshauer, F. J. Hickernell, and H. Woźniakowski, *On dimension-independent rates of convergence for function approximation with Gaussian kernels*, under revision, (2011).

[8] ———, *Average case approximation: convergence and tractability of Gaussian kernels*, in Monte Carlo and Quasi-Monte Carlo Methods 2010, H. Woźniakowski and L. Plaskota, eds., Springer, to appear.

[9] 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.

[10] G. E. Fasshauer and Q. Ye, *Reproducing kernels of generalized Sobolev spaces via a Green function approach with distributional operators*, Numer. Math., 119 (2011), pp. 585–611.

[11] ———, *Reproducing kernels of Sobolev spaces via a Green function approach with differential operators and boundary operators*, submitted, (2011).

[12] B. Fornberg, E. Larsson, and N. Flyer, *Stable computations with Gaussian radial basis functions*, SIAM J. Sci. Comput., 33 (2011), pp. 869–892.

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

[14] B. Fornberg and G. Wright, *Stable computation of multiquadric interpolants for all values of the shape parameter*, Comput. Math. Appl., 48 (2004), pp. 853–867.

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

[16] G. H. Golub and C. F. Van Loan, *Matrix computations (3rd ed.)*, Johns Hopkins University Press, Baltimore, MD, USA, 1996.

[17] 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.

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

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

[20] J. Mercer, *Functions of positive and negative type, and their connection with the theory of integral equations*, Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character, 209 (1909), pp. 415–446.

[21] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*, The MIT Press, 2005.

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

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

[24] ———, *Limit problems for interpolation by analytic radial basis functions*, J. Comput. Appl. Math., 212 (2008), pp. 127–149.

[25] ———, *Approximationsverfahren II.* Institut für Numerische und Angewandte Mathematik, Universität Göttingen, 2011. `http://num.math.uni-goettingen.de/schaback/teaching/Appverf_II.pdf`.

[26] E. Schmidt, *Über die Auflösung linearer Gleichungen mit unendlich vielen Unbekannten*, Rendiconti del Circolo Matematico di Palermo, 25 (1908), pp. 53–77.

[27] I. Steinwart and A. Christmann, *Support Vector Machines*, Information Science and Statistics, Springer, 2008.

[28] G. Szegő, *Orthogonal Polynomials*, Colloquium Publications - American Mathematical Society, American Mathematical Society, 1939.

[29] J. F. Traub, G. W. Wasilkowski, and H. Woźniakowski, *Information-Based Complexity*, Academic Press Professional, Inc., San Diego, CA, USA, 1988.

[30] H. Wendland, *Scattered Data Approximation*, Cambridge Monographs on Applied and Computational Mathematics, Cambridge University Press, 2005.

[31] H. Zhu, C. K. I. 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.