Journal of Computational and Applied Mathematics 236 (2011) 575–588

Contents lists available at SciVerse ScienceDirect

Journal of Computational and Applied Mathematics

journal homepage: www.elsevier.com/locate/cam

Bases for kernel-based spaces

Maryam Pazouki, Robert Schaback*

Institut für Numerische und Angewandte Mathematik, Universität Göttingen, Lotzestraße 16-18, D-37073 Göttingen, Germany

ARTICLE INFO

MSC:
41A05
41063
41065
65D05
65D15

Keywords:
Scattered data
Kernels
Matching pursuit
Adaptivity
Stability
Duality

ABSTRACT

Since it is well-known (De Marchi and Schaback (2001) [4]) that standard bases of kernel translates are badly conditioned while the interpolation itself is not unstable in function space, this paper surveys the choices of other bases. All data-dependent bases turn out to be defined via a factorization of the kernel matrix defined by these data, and a discussion of various matrix factorizations (e.g. Cholesky, *QR*, SVD) provides a variety of different bases with different properties. Special attention is given to duality, stability, orthogonality, adaptivity, and computational efficiency. The "Newton" basis arising from a pivoted Cholesky factorization turns out to be stable and computationally cheap while being orthonormal in the "native" Hilbert space of the kernel. Efficient adaptive algorithms for calculating the Newton basis along the lines of orthogonal matching pursuit conclude the paper.

1. Introduction and overview

Let Ω be a nonempty set, and let

$$K : \Omega \times \Omega \to \mathbb{R}$$

be a positive definite symmetric *kernel* on Ω. This means that for all finite sets $X := \{x_1, \dots, x_N\} \subseteq \Omega$ the *kernel matrix*

$$A := (K(x_j, x_k))_{1\le j,\ k\le n} \tag{1}$$

is symmetric and positive definite. It is well-known (see the books [1–3] for a full account of this research area) that these kernels are *reproducing* in a "native" Hilbert space $\mathcal{K}$ of functions on Ω in the sense

$$(f, K(x, \cdot))_{\mathcal{K}} = f(x) \quad \text{for all } x \in \Omega,\ f \in \mathcal{K}. \tag{2}$$

The functions $K(x, \cdot)$ thus are the Riesz representers of the linear point evaluation functionals $\delta_x : f \mapsto f(x)$ in the dual $\mathcal{K}^*$ of $\mathcal{K}$, and the kernel can be expressed as

$$K(x, y) = (K(x, \cdot), K(y, \cdot))_{\mathcal{K}} = (\delta_x, \delta_y)_{\mathcal{K}^*} \quad \text{for all } x, y \in \Omega. \tag{3}$$

This paper will be concerned with suitable bases for subspaces of $\mathcal{K}$. These will come in different forms, and we shall explore their variety and prove results about their connection.

In practice, users have a finite set $X := \{x_1, \dots, x_N\} \subseteq \Omega$ of *centers* or *data locations* and work in a subspace

$$\mathcal{K}_X := \text{span}\,\{K(\cdot, x_j) : x_j \in X\} \tag{4}$$

* Corresponding author.
E-mail addresses: m.pazoki@math.uni-goettingen.de (M. Pazouki), schaback@math.uni-goettingen.de (R. Schaback).

0377-0427/$ – see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.cam.2011.05.021

spanned by the basis of (generalized) *translates* $K(\cdot, x_j)$, $1 \le j \le N$. Bases of $\mathcal{K}_X$ will be called *data-dependent* in this paper, and we shall consider a wide variety of these. It is well-known that the standard basis (4) of translates leads to ill-conditioned kernel matrices (1), but results of [4] show that interpolants to data on X, when viewed as functions, are rather stable. This leads us to consider different data-dependent bases and to compare them with respect to stability, recursive computability, duality, and orthogonality properties. In Section 3 we shall consider general bases, while we shall specialize to $\mathcal{K}$-orthonormal and discretely orthonormal bases in Sections 6 and 7. Our goal is to sort all possible bases into certain classes and to prove as much as we can about their properties and their relations.

But if no data set is specified, we should also consider "data-independent" bases. The most natural is the *eigenfunction basis* coming from the Mercer theorem. We shall describe it in Section 2 and compare it later to data-dependent bases. Of course, data-independent bases are much more "natural" than data-dependent bases, and they provide intrinsic information about the kernel. But when starting with a data-independent basis, problems may arise when working on a set X of centers later, because the Mairhuber–Curtis theorem [5] shows that matrices of values of the basis on points of X have a data-dependent rank, if one is in a truly multivariate and nontrivial situation. This is why we shall use the data-independent eigenfunction basis only for theoretical comparisons.

We ignore bases that are constructed iteratively via Krylov subspace methods and using neighboring points. Such algorithms can be very effective, in particular in cases where local Lagrangian basis functions decay exponentially, e.g. for thin-plate splines. We refer the reader to [6,7] for details. Also, we neglect bases that come from special preconditioning techniques. Instead, we mainly focus here on bases that can be derived by standard linear algebra operations on the usual kernel matrix.

2. Data-independent bases

The natural basis for Hilbert spaces with reproducing kernels comes from

Theorem 2.1 (*Mercer*). *Continuous positive definite kernels K on bounded domains Ω have an eigenfunction expansion*

$$K(x,y) = \sum_{n=0}^{\infty} \lambda_n \varphi_n(x)\varphi_n(y) \quad \textit{for all } x, y \in \Omega$$

which is absolutely and uniformly convergent. Furthermore,

$$\begin{aligned} &\lambda_n \varphi_n(x) = \int_\Omega K(x,y)\varphi_n(y)\mathrm{d}y \quad \textit{for all } x \in \Omega,\ n \ge 0,\\ &\{\varphi_n\}_n \quad \textit{orthonormal in } \mathcal{K},\\ &\{\varphi_n\}_n \quad \textit{orthogonal in } L_2(\Omega),\\ &\|\varphi_n\|_2^2 = \lambda_n \to 0, \quad n \to \infty. \end{aligned}$$

This basis directly describes the action of the kernel as an integral operator performing a generalized convolution. In many cases, the eigenvalues decrease very rapidly toward zero, and this implies that there is a very good low-rank approximation to the kernel. This observation has serious consequences for kernel-based algorithms, because one has to encounter rank loss in linear systems that are based on values of K. The systems will show very bad condition, but on the other hand one knows that the rank-reduced system will be very close to the exact system. Thus, in many cases, linear systems arising from kernels have bad condition, but they also have a low-rank subsystem that performs like the full system, i.e. allows to approximate the right-hand side very much the same way as the full matrix does. This has been observed in many applications from machine learning to PDE solving, and it cannot be eliminated by other choices of bases, only by proper regularization or by adaptive techniques that find and use good subproblems. We shall come back to this issue later.

The numerical calculation of approximations to the eigenfunction basis can be based on a sufficiently large point set $X = \{x_1, \dots, x_N\}$ which allows numerical integration

$$\int_\Omega f(y)\mathrm{d}y \approx \sum_{j=1}^{N} w_j f(x_j)$$

for functions $f \in \mathcal{K}$ using certain positive weights $w_1, \dots, w_N$. Then the discretization of the eigenfunction equation

$$\lambda_n \varphi_n(x_k) = \int_\Omega K(x_k, y)\varphi_n(y)\mathrm{d}y$$

leads to

$$\lambda_n \underbrace{\sqrt{w_k}\varphi_n(x_k)}_{v_k^{(n)}} \approx \sum_{j=1}^{N} \underbrace{\sqrt{w_k}K(x_k,x_j)\sqrt{w_j}}_{b_{jk}} \underbrace{\sqrt{w_j}\varphi_n(x_j)}_{v_j^{(n)}}$$

and which is an approximation to the discrete eigenvalue problem

$$\lambda_n v_k^{(n)} = \sum_{j=1}^{N} b_{jk} v_j^{(n)}, \quad 1 \le k,\ n \le N \tag{5}$$

involving a scaled version of the kernel matrix (1). The obtained values $v_j^{(n)}$ lead to functions $v^{(n)} \in \mathcal{K}_X$ by solving the interpolation problems

$$v^{(n)}(x_j) = v_j^{(n)}, \quad 1 \le j,\ n \le N,$$

in the space $\mathcal{K}_X$, and we see that we have constructed a data-dependent basis as an approximation to a data-independent basis. Thus this case falls into the next section, and we shall come back to it.

3. General data-dependent bases

From now on, we go back to the notation of the introduction and fix a set $X = \{x_1, \dots, x_N\}$, the kernel matrix $A = (K(x_j, x_k))_{1\le j,\ k\le N}$, and the space $\mathcal{K}_X$ of (4), but dropping X in most of what follows. Any basis $u_1, \dots, u_N$ of $\mathcal{K}_X$ can be arranged into a *row* vector

$$U(x) := (u_1(x), \dots, u_N(x)) \in \mathbb{R}^N,$$

and it can be expressed by the basis T of translates

$$T(x) := (K(x, x_1), \dots, K(x, x_N))$$

by a *coefficient* or *construction* matrix C_U via

$$\begin{aligned} U(x) &= T(X) \cdot C_U, \\ u_k(x) &= \sum_{j=1}^{N} K(x, x_j) c_{jk}, \quad 1 \le k \le N. \end{aligned} \tag{6}$$

The set $\mathcal{B}$ of all possible data-dependent bases is the bijective image of the group $GL(n, \mathbb{R})$ of all nonsingular real $n \times n$ matrices under the map $C \mapsto T(x) \cdot C$. This means that one has a composition $\circ$ of two bases U and V via

$$(U \circ V)(x) := T(x) \cdot C_U \cdot C_V = T(x) \cdot C_{U\circ V}$$

and thus one can define the *inverse* of a basis. This concept, but with a "coordinate space" being fixed instead of a fixed basis T, was introduced and exploited by de Boor in [8]. Here, we just note that the full set of possible bases U can be parametrized by arbitrary matrices $C_U \in GL(n, \mathbb{R})$. Thus we shall express formulas for features of bases U mainly in terms of C_U, but there are other parametrizations as well, as we shall see.

The *evaluation* operator E based on the set X will map functions f into columns

$$E(f) := (f(x_1), \dots, f(x_N))^T \in \mathbb{R}^N$$

of values on X, and rows of functions into matrices, such that

$$E(T) = (K(x_i, x_j))_{1\le i,\ j\le N} = A$$

is the kernel matrix. Similarly, for a general basis U we can form the *value* matrix

$$V_U := E(U) = (u_j(x_i))_{1\le i,\ j\le N}.$$

A very similar way to use columns and rows, and the connection to duality we shall use later are nicely described already in [8].

From the identity

$$V_U = E(U) = E(T) \cdot C_U = A \cdot C_U,$$

we immediately get

Theorem 3.1. *Any data-dependent basis U arises from a factorization*

$$A = V_U \cdot C_U^{-1} \tag{7}$$

of the kernel matrix A into the value matrix $V_U = A \cdot C_U$ and the inverse construction matrix C_U^{-1} of the basis.

For the basis T of translates, the factorization in (7) is $A = A \cdot I$, where we use I to stand for the $N \times N$ identity matrix. Note that (7) also shows that we could as well parametrize the set of bases U via the value matrices V_U, using $C_U = A^{-1} \cdot V_U$ to come back to the parametrization via C_U.

But for sorting out special classes of bases, we need more. By a short calculation based on (6), the *Gramian* G_U of a general basis U comes out to be

$$G_U := ((u_i, u_j)_{\mathcal{K}})_{1\le i,\, j\le N} = C_U^T \cdot A \cdot C_U. \tag{8}$$

These are the $\mathcal{K}$-inner products, but we also have discrete $\ell_2(X)$ inner products forming a Gramian Γ_U via

$$\begin{aligned}\Gamma_U &:= ((u_i, u_j)_{\ell_2(X)})_{1\le i,\, j\le N}\\ &= \left(\sum_{n=1}^{N} u_i(x_n)u_j(x_n)\right)_{1\le i,\, j\le N}\\ &= V_U^T \cdot V_U\\ &= C_U^T \cdot A^2 \cdot C_U\end{aligned}$$

using (7).

4. Interpolation, functionals, and duality

Interpolants $s_f \in \mathcal{K}_X$ to values $E(f)$ of some function f can be written as

$$s_f(x) = T(x)\alpha$$

with a coefficient vector $\alpha \in \mathbb{R}^N$ satisfying the linear system

$$A\alpha = E(f). \tag{9}$$

This is well-known, but also follows immediately from

$$E(s_f) = E(T)\alpha = A\alpha = E(f)$$

using our notation. For general bases, the interpolant takes the form

$$\begin{aligned}s_f(x) &:= T(x) \cdot A^{-1} \cdot E(f)\\ &= U(x) \cdot \underbrace{C_U^{-1} \cdot A^{-1} \cdot E(f)}_{=:\Lambda_U(f)}\\ &= \sum_{j=1}^{N} u_j(x)\lambda_j(f)\end{aligned} \tag{10}$$

with a column vector

$$\Lambda_U(f) := (\lambda_1(f), \dots, \lambda_N(f))^T = C_U^{-1} \cdot A^{-1} \cdot E(f)$$

of values of linear *functionals*. The corresponding functionals $\lambda_1, \dots, \lambda_N$ are from the span of the point evaluation functionals $\delta_{x_1}, \dots, \delta_{x_N}$, and they can be composed from them via

$$\Lambda_U(f) = \underbrace{C_U^{-1} \cdot A^{-1}}_{=:\Delta_U} \cdot E(f) = \Delta_U \cdot E(f)$$

as a matrix operation with

$$\Delta_U = C_U^{-1} \cdot A^{-1} = V_U^{-1}.$$

We have chosen the notation Δ_U here, because the action of the matrix is like forming divided differences from function values. For the basis T of translates, we have $\Delta_T = V_T^{-1} = A^{-1}$. Note that we could parametrize bases U also via Δ_U.

The evaluation of an interpolant s_f at some point x via (10) can be unstable, if either the $u_j(x)$ or the $\lambda_j(f)$ or both are large, canceling finally when forming the result $s_f(x)$. This regularly occurs for the basis T of translates, because the coefficient vector $A^{-1}E(f)$ tends to have huge absolute values of opposite sign. A measure for the stability of the evaluation of $s_f(x)$ thus is the Hölder–Minkowski bound

$$|s_f(x)| \le \|U(x)\|_p \, \|\Lambda_U(f)\|_q \quad \text{for all } f \in \mathcal{K},\ x \in \Omega \tag{11}$$

with $1/p + 1/q = 1$. We shall have to look at both factors in what follows.

In [4], there is an analysis of the stability of the Lagrange basis along this line, proving that the right-hand side is well-behaved. This implies that interpolation is not unstable in function space, though the calculation of coefficients in the basis of translates is unstable. Consequently, we have to look for other bases which allow good bounds in (11).

The linear functionals in Λ_U are in some sense *dual* to the basis U, but we define duality slightly differently.

Definition 4.1. The *dual* basis to a basis U is the basis U^* of the Riesz representers of the functionals of Λ_U.

Given some basis U, we now have to find the value matrix V_{U^*} and the construction matrix C_{U^*} of the dual basis U^*.

Theorem 4.1. *The dual basis U^* to a data-dependent basis U satisfies*

$$\begin{aligned}
&V_{U^*} = (C_U^T)^{-1},\\
&C_{U^*} = (V_U^T)^{-1},\\
&U^{**} = U,\\
&A = V_U \cdot V_{U^*}^T,\\
&K(x_j, x_k) = \sum_{m=1}^{N} u_m(x_k)u_m^*(x_j), \quad 1 \le j,\ k \le N,\\
&(u_j, u_k^*)_{\mathcal{K}} = \delta_{jk}, \quad 1 \le j,\ k \le N.
\end{aligned}$$

Proof. The dual basis functions u_j^* are defined by

$$\lambda_j(f) = (u_j^*, f)_{\mathcal{K}}, \quad \text{for all}\, f \in \mathcal{K},\ 1 \le j \le N. \tag{12}$$

Thus

$$\begin{aligned}
u_j^*(x) &= (u_j^*, K(x, \cdot))_{\mathcal{K}} = \lambda_j(K(x, \cdot)), \quad 1 \le j \le N,\ \text{for all}\, x \in \Omega,\\
U^*(x) &= \Lambda_U^T(K(x, \cdot))\\
&= C_U^{-1} \cdot A^{-1} \cdot E(K(x, \cdot))^T\\
&= V_U^{-1} \cdot E(K(x, \cdot))^T\\
&= T(x)(V_U^{-1})^T
\end{aligned}$$

for all $x \in \Omega$ proves $C_{U^*} = (V_U^T)^{-1}$ via (6). Then (7) yields

$$\begin{aligned}
V_{U^*} &= A \cdot C_{U^*}\\
&= A \cdot (V_U^T)^{-1}\\
&= A \cdot ((A \cdot C_U)^T)^{-1}\\
&= (C_U^T)^{-1},
\end{aligned}$$

and the next three relations are easy consequences. Finally,

$$\begin{aligned}
(u_j, u_k^*)_{\mathcal{K}} &= \lambda_k(u_j)\\
&= e_k^T \Lambda_U(u_j)\\
&= e_k^T \Delta_U \cdot E(u_j)\\
&= e_k^T \Delta_U \cdot E(Ue_j)\\
&= e_k^T \Delta_U \cdot E(U)e_j\\
&= e_k^T C_U^{-1} A^{-1} V_U e_j\\
&= e_k^T e_j, \quad 1 \le j,\ k \le N
\end{aligned}$$

proves the last assertion and shows that the functionals of Λ_U always are a biorthogonal basis with respect to U. □

The transition from a basis U to its dual follows a simple rule-of-thumb. Starting with (7), we take the transpose and re-interpret this in the sense of Theorem 3.1 as the product of a dual value matrix times the inverse of a dual construction matrix:

$$\begin{aligned}
A &= V_U \cdot C_U^{-1}\\
&= A^T = (C_U^{-1})^T \cdot V_U^T\\
&= V_{U^*} \cdot C_{U^*}^{-1}.
\end{aligned}$$

For later use in stability considerations along the lines of (11), we use (10) to get

$$\|s_f\|_{\mathcal{K}}^2 = \Lambda^T(f) G_U \Lambda(f)$$

with the $\mathcal{K}$-Gramian G_U. By standard eigenvalue manipulations and the inequality $\|s_f\|_{\mathcal{K}} \le \|f\|_{\mathcal{K}}$, this implies

Theorem 4.2. *For all $f \in \mathcal{K}$ and all data-dependent bases U,*

$$\|\Lambda(f)\|_2^2 \le \|f\|_{\mathcal{K}}^2 \rho(G_U^{-1})$$

with ρ being the spectral radius.

Note that this bound is particularly bad for the basis T of translates with $G_T = A$.

5. Lagrange basis and power function

We now look at the standard *Lagrange* basis L whose elements are data-dependent and satisfy

$$L_j(x_k) = \delta_{jk}, \quad 1 \le j \le k.$$

Clearly, $V_L = I = \Gamma_L$ and by (7) we get $C_L = A^{-1}$. The $\mathcal{K}$-Gramian is $G_L = A^{-1}$ by (8). By Theorem 4.1 or by the above rule-of-thumb in the form

$$A = A \cdot I = V_T \cdot C_T^{-1} = A^T = I \cdot A^T = V_{T^*} \cdot C_{T^*}^{-1}$$

we get

Theorem 5.1. *The Lagrange basis L and the basis T of translates are a dual pair.* □

Another way to see this duality is by noting that the functionals $\lambda_j(f) = \delta_{x_j}(f)$ of the Lagrange basis are the Riesz representers of the kernel translates $K(\cdot, x_j)$.

Before we turn to other cases, we still have to introduce the *power function* and to express it for general bases. The pointwise error functional

$$f \mapsto \delta_x(f) - s_f(x) = \delta_x(f) - \sum_{j=1}^{N} L_j(x)\delta_{x_j}(f)$$

has the norm

$$P(x) = \left\| \delta_x - \sum_{j=1}^{N} L_j(x)\delta_{x_j} \right\|$$

which is called the *power function* of the interpolation process. By definition, it allows pointwise error bounds

$$|f(x) - s_f(x)| \le P(x)\|f\|_{\mathcal{K}} \quad \text{for all } x \in \Omega,\ f \in \mathcal{K}.$$

By well-known optimality arguments [2], it satisfies

$$P^2(x) = \min_{a \in \mathbb{R}^N} \left\| \delta_x - \sum_{j=1}^{N} a_j\delta_{x_j} \right\|^2$$

and thus is independent of the basis, the optimal coefficients $a_j^*(x)$ being given by the Lagrange basis functions $L_j(x)$. Using this optimality, the formula (3) and some standard algebraic manipulations within our formalism, we can express it in terms of a general basis as

$$\begin{aligned} P^2(x) &= K(x,x) - 2\sum_{j=1}^{N} K(x,x_j)L_j(x) + \sum_{j,k=1}^{N} K(x_k,x_j)L_j(x)L_k(x) \\ &= K(x,x) - \sum_{j=1}^{N} K(x,x_j)L_j(x) \\ &= K(x,x) - T(x) \cdot L^T(x) \\ &= K(x,x) - T(x) \cdot A^{-1} \cdot T^T(x) \\ &= K(x,x) - U(x) \cdot C_U^{-1} \cdot A^{-1} \cdot (C_U^{-1})^T \cdot U^T(x) \\ &= K(x,x) - U(x) \cdot G_U^{-1} \cdot U^T(x). \end{aligned} \tag{13}$$

Due to positive definiteness of the $\mathcal{K}$-Gramian G_U, this yields bounds

$$0 \le U(x) \cdot G_U^{-1} \cdot U^T(x) = K(x,x) - P^2(x) \le K(x,x)$$

for the pointwise behavior of the general basis U. By standard eigenvalue bounds, this implies a bound that is useful for (11).

Theorem 5.2. *For arbitrary data-dependent bases U, we have*

$$\|U(x)\|_2^2 \le K(x,x)\rho(G_U) \quad \textit{for all } x \in \Omega$$

with the spectral radius $\rho(G_U)$ of the $\mathcal{K}$-Gramian. Furthermore, the stability bound (11) for $p = q = 2$ is

$$|s_f(x)|^2 \le \|U(x)\|_2^2 \|\Lambda_U(f)\|_2^2 \le \|f\|_K^2 K(x,x)\mathrm{cond}_2(G_U) \tag{14}$$

for all $f \in \mathcal{K}$, $x \in \Omega$, where $\mathrm{cond}_2(G_U)$ is the condition number of the $\mathcal{K}$-Gramian with respect to the Euclidean norm. □

Note how the first part of (14) shows the nice factorization into a basis-dependent term and an f-dependent term.

6. $\mathcal{K}$-orthonormal bases

From (14) we see that we should look for bases U with $G_U = I$, i.e. for $\mathcal{K}$-orthonormal bases.

Theorem 6.1. *Each data-dependent $\mathcal{K}$-orthonormal basis U arises from a decomposition*

$$A = B^T \cdot B \quad \textit{with } B = C_U^{-1}, \qquad V_U = B^T = (C_U^{-1})^T. \tag{15}$$

Among all data-dependent bases, the $\mathcal{K}$-orthonormal bases are exactly those which are self-dual.

Proof. Clearly, $G_U = C_U^T \cdot A \cdot C_U = I$ is equivalent to $A = (C_U^{-1})^T C_U^{-1}$, proving the first assertion. By Theorem 4.1, all $\mathcal{K}$-orthonormal bases are self-dual. Conversely, if U is a self-dual basis, then

$$V_U^{-1} = C_U^T,$$
$$A = V_U \cdot (V_U^{-1})^T,$$

and the second assertion follows from the first. □

There are two important special cases.

The Cholesky decomposition $A = L \cdot L^T$ with a nonsingular lower triangular matrix L leads to the *Newton* basis N treated in [9] with a different normalization. It can be recursively calculated and has the property $N_j(x_k) = 0,\ 1 \le k < j \le N$ like the basis of functions

$$N_j(x) = \prod_{1 \le k < j} (x - x_k), \quad 1 \le j \le N$$

in Newton's formula for polynomial interpolation.

The other case is induced by *singular value decomposition* (SVD) in the form $A = Q^T \cdot \Sigma^2 \cdot Q$ with an orthogonal matrix Q and a diagonal matrix Σ having the eigenvalues of A on its diagonal. This *SVD basis* S satisfies

$$B = \Sigma \cdot Q, \qquad C_S = Q^T \cdot \Sigma^{-1}, \qquad V_S = Q^T \cdot \Sigma.$$

Before we analyze these special cases further, we prove slightly more than we had for Theorem 5.2.

Theorem 6.2. *For all $\mathcal{K}$-orthonormal bases U, the value of $\|U(x)\|_2$ for fixed $x \in \Omega$ is the same and bounded above by $K(x,x)$ independent of the placement and number of data points. Dually, the value of $\|\Lambda(f)\|_2$ for fixed $f \in \mathcal{K}$ is the same for all $\mathcal{K}$-orthonormal bases and bounded above by $\|f\|_{\mathcal{K}}$ independent of the placement and number of data points.*

Proof. Eq. (13) yields

$$\sum_{j=1}^N u_j^2(x) = \|U(x)\|_2^2 = K(x,x) - P^2(x) \le K(x,x) \quad \text{for all } x \in \Omega,$$

and this proves the first assertion, because the power function is basis-independent. Writing an interpolant s_f in the form (10) is an orthonormal representation. Thus, being basis-independent as well,

$$\|s_f\|_{\mathcal{K}}^2 = \sum_{j=1}^N \lambda_j^2(f) \le \|f\|_{\mathcal{K}}^2 \tag{16}$$

proves the second. □

This shows that $\mathcal{K}$-orthonormal bases will lead to stable results in function space even for nearly-coalescing data points, provided that the data come from a function in the native space. The functionals act like divided differences and have norm 1 in the dual of the native space, irrespective of the placement of data points.

From Theorem 5.2 we get the bound

$$|s_f(x)|^2 \le K(x,x)\|f\|^2_{\mathcal{K}} \quad \text{for all } f \in \mathcal{K},\ x \in \Omega \tag{17}$$

for all $\mathcal{K}$-orthonormal bases U, implying that the evaluation of the interpolant is stable provided that the $u_j(x)$ and $\lambda_j(f)$ can be evaluated stably. The basic equations for these are

$$U(x) = T(x)B^{-1} \quad \text{for all } x \in \Omega, \qquad \Lambda_U(f) = (B^{-1})^T E(f) \quad \text{for all } f \in \mathcal{K},$$

and we see that in both cases the matrix B^{-1} is involved, or a system with coefficient matrix B has to be solved. For the condition of B we have.

Theorem 6.3. *The general solution B of (15) is always of the form $B = Q_1 \Sigma Q$ with an orthogonal matrix Q_1, when $A = Q^T \Sigma^2 Q$ is an SVD of A. Thus the spectrum of A is factored by (15), and the spectral condition of B is the square root of the spectral condition of A.*

Proof. From $A = Q^T\Sigma^2 Q = B^T B$ we get $I = \Sigma^{-1} Q B^T B Q^T \Sigma^{-1}$. Thus $Q_1 := BQ^T\Sigma^{-1}$ is orthogonal and $B = Q_1 \Sigma Q$. The matrix B has an SVD with singular values being the square roots of those of A. □

Thus all $\mathcal{K}$-orthonormal bases divide the ill-conditioning of A fairly between the function and the functional part. Later, we shall consider special adaptive algorithms for the Newton case.

7. Discretely orthonormal bases

Theorem 7.1. *Each data-dependent discretely orthonormal basis arises from a decomposition*

$$A = Q \cdot B$$

with $Q = V_U$ orthogonal and $B = C_U^{-1} = Q^T \cdot A$.

Proof. By the formula $\Gamma_U = C_U^T \cdot A^T \cdot A \cdot C_U$ for the discrete Gramian Γ_U, and setting $Q := A \cdot C_U$, we see that $\Gamma_U = I$ is equivalent to orthogonality of Q. □

Again, we have two special cases. A standard QR decomposition $A = QR$ into an orthogonal matrix Q and an upper triangular matrix R will lead to a basis we shall denote by O with $C_O = R^{-1},\ V_O = Q$. This is nothing else than Gram–Schmidt orthonormalization of the values of the translate basis T on X. The second case comes from rescaling an SVD basis. In fact, any SVD $A = Q^T\Sigma^2 Q$ can be split into $A = Q \cdot B$ with $B = \Sigma^2 Q$. This makes the value matrix orthonormal, while the ill-conditioning is completely shifted into the construction matrix. Thus, if scaling is ignored, all SVD bases are both discretely and $\mathcal{K}$-orthogonal. The converse is also true:

Theorem 7.2. *All data-dependent bases which are discretely and $\mathcal{K}$-orthogonal are scaled SVD bases.*

Proof. Any such basis U can be rescaled to be $\mathcal{K}$-orthonormal. We then have $A = B^T \cdot B$ with $B = C_U^{-1}$ and $\Gamma_U = C_U^T \cdot A^2 C_U = D^2$ with a nonsingular diagonal matrix D. This implies $I = D^{-1} C_U^T A^T A C_U D^{-1}$ and that

$$Q := A \cdot C_U \cdot D^{-1} = B^T \cdot B \cdot C_U \cdot D^{-1} = (C_U^{-1})^T \cdot D^{-1}$$

is orthogonal. But then

$$\begin{aligned} A \cdot Q &= A \cdot A \cdot C_U \cdot D^{-1} \\ &= (C_U^{-1})^T D^{-1} \\ &= Q \cdot D^2 \end{aligned}$$

leads to the SVD of $A = Q \cdot D^2 \cdot Q^T$ with $B = C_U^{-1} = D \cdot Q^T$. □

For any discretely orthonormal basis U, the $\mathcal{K}$-Gramian is

$$G_U = C_U^T \cdot A \cdot C_U = Q^T \cdot A^{-1} \cdot A \cdot A^{-1} \cdot Q = Q^T \cdot A^{-1} \cdot Q$$

and thus spectrally equivalent to A^{-1}. In view of Theorem 5.2, this is comparable to the Lagrange and the translates basis.

Theorem 7.3. *The duals of the discretely orthonormal bases arise from decompositions*

$$A = B \cdot Q$$

with Q orthogonal.

Proof. Following our rule-of-thumb for a discretely orthonormal basis U, we get

$$A = Q \cdot B = A^T = B^T \cdot Q^T = V_{U^*} \cdot C_{U^*}^{-1}, \qquad V_{U^*} = B^T, \qquad C_{U^*} = Q. \quad \square$$

These bases have orthogonal construction matrices instead of orthogonal value matrices. Again, a scaled SVD basis is a special case, and also the transpose $A = R^T \cdot Q^T$ of a QR decomposition $A = Q \cdot R$. The $\mathcal{K}$-Gramians of these bases are of the form $Q \cdot A \cdot Q^T$, and thus again spectrally equivalent to the translate and Lagrange bases, as far as the spectral condition is concerned.

8. SVD bases

Though their computation is rather involved, the SVD bases have some nice properties, as we have seen in the previous sections, in particular in Theorem 7.2. Going back to Section 2 and using an integration formula with well-distributed points and equal weights $w_k = w > 0$, we see that the discretized solution $v_j^{(n)}, \ 1 \leq j, \ n \leq N$ of the eigenvalue problem (5) is related to the eigenvalue problem of A itself, and thus is a scaled SVD basis. Thus we can expect that SVD bases for large and well-placed data point sets are approximations of the data-independent eigenfunction basis. For kernels with rapidly decaying eigenvalues, one has to expect numerical rank loss in the kernel matrix A, but the SVD is the best known way to control this.

In theory, the SVD basis S based on an SVD $A = Q^T \cdot \Sigma^2 \cdot Q$ is given by $S(x) = T(x) \cdot Q^T \cdot \Sigma^{-1}$, while the value matrix is $V_S = Q^T \cdot \Sigma$. If singular values σ_j^2 are sorted to decrease with increasing j, the columns of the value matrix have decreasing norms $\|V_U e_j\|_2 = \sigma_j$ for increasing j. The usual Tychonov regularization will replace small $\sigma_j < \epsilon$ by zero, thus making the basis shorter. In that case, the numerical result of the reconstruction of f from given data $E(f)$ will be a non-interpolatory projection of f into the span of the selected SVD basis functions, and if these are corresponding to the eigenfunctions with large eigenvalues, the result will be an accurate and stable reproduction of the projection of f into the span of these eigenfunctions. If, however, f has a large projection onto higher eigenfunctions, this will lead to unavoidable errors, but these often look like noise, while the numerical solution looks smooth. This makes the SVD useful for a lot of applications where deterministic worst-case error bounds make no sense, and in particular where the data are noisy anyway and exact interpolation is not desirable.

9. Newton basis

This basis was already treated in [9], but with a different normalization that concealed its $\mathcal{K}$-orthonormality somewhat. Theorem 6.2 and the stability bound (17) are not in [9], but proved here for general $\mathcal{K}$-orthonormal bases.

A Cholesky decomposition $A = L \cdot L^T$ with a nonsingular lower triangular matrix L leads to the Newton basis N with $N(x) = T(x) \cdot C_N = T(x) \cdot (L^T)^{-1}$ and $V_N = L$. In practice, the Cholesky decomposition would be pivoted, but we shall describe another adaptive algorithm below.

The construction of the Newton basis is recursive like the Cholesky algorithm, and this means that the first n basis functions $N_1, \dots, N_n$ need not be recalculated when going over from n to $n + 1$. This has some remarkable consequences which were not noted in [9]. In particular, we can interpret (13) in the form

$$\sum_{j=1}^{n} N_j^2(x) = K(x, x) - P_n^2(x), \tag{18}$$

recursively, if we denote the power function on the n points of $X_n := \{x_1, \dots, x_n\}$ by P_n. Thus

Theorem 9.1.

$$N_n^2(x) = P_{n-1}^2(x) - P_n^2(x) \leq P_{n-1}^2(x). \tag{19}$$

If x_n is chosen recursively as

$$x_n := arg\, max\, P_{n-1}^2(\cdot),$$

then

$$N_n^2(x) \leq N_n^2(x_n) \quad \textit{for all } x \in \Omega,$$

i.e. the basis has no parasitic maxima.

Proof. The first statement follows from (18), and the second from $N_n^2(x) \leq P_{n-1}^2(x) \leq P_{n-1}^2(x_n) = N_n^2(x_n)$ because of $P_n(x_n) = 0$. $\square$

This argument was already used in [4] for the Lagrange basis. If $x_1, \dots, x_n$ are fixed, the functions L_n and N_n differ only by a normalization factor, but L_n will change when we go over to $n+1$.

In (18), one can take the limit $n \to \infty$ without problems, and it was proven in [4] that

$$\sum_{j=1}^{\infty} N_j^2(x) = K(x,x)$$

if the points $x_1, x_2, \dots$ get dense in a bounded domain $\Omega \subset \mathbb{R}^d$. On such a domain, and for a continuous kernel K, we also get

$$\sum_{j=1}^{n} \|N_j\|^2_{L_2(\Omega)} \le \int_\Omega K(x,x)\mathrm{d}x$$

by integration of (18), for $n \to \infty$, and for the craziest possible point distributions. Together with (19), this shows that the Newton basis does not seriously degenerate when points get close. This is in line with the nondegeneracy of the data provided by (16), if the data come from a function f in the native space.

By construction, the functions $N_1, \dots, N_n$ are an orthonormal basis for the span of the translates $K(\cdot, x_1), \dots, K(\cdot, x_n)$. Thus the action of the reproducing kernel K on that space is given by

$$K_n(x,y) = \sum_{j=1}^{n} N_j(x)N_j(y)$$

and the action of the kernel K on the orthogonal complement

$$\{f \in \mathcal{K} : f(x_j) = 0,\ 1 \le j \le n\}$$

is given by the kernel $K(x,y) - K_n(x,y)$. In [10], the latter was called the *power kernel* and represented differently, without using the Newton basis. If the points $x_1, x_2, \dots$ are dense in a bounded domain Ω, this argument proves the series representation

$$K(x,y) = \sum_{j=1}^{\infty} N_j(x)N_j(y) \tag{20}$$

of the kernel. From Section 2 we know that there may be a good low-rank approximation to the kernel, and thus we have to anticipate that, for a special ordering of the points, convergence of the series may be rapid and connected to the decay of the eigenvalues of the kernel. This means that one should consider adaptive point selections that make the series converge fast. This will be the topic of the next section.

10. Adaptive calculation of Newton bases

We first consider the case where we want to find a good basis for *all* possible interpolation problems, i.e. we do not care for single data and focus on point selection instead. Let $\Omega \subset \mathbb{R}^d$ be a bounded domain with a continuous positive definite kernel K on it. For applications, users will take a large finite subset $X = \{x_1, \dots, x_N\} \subset \Omega$ but avoid to form the huge $N \times N$ kernel matrix A for all points of X. Instead, one can use a column-based version of a pivoted Cholesky decomposition which performs m steps of computational complexity $\mathcal{O}(Nm)$ to stop at rank m. Its overall complexity thus is $\mathcal{O}(Nm^2)$, and it requires a total storage of $\mathcal{O}(mN)$. It builds the first m columns of the value matrix V_N and thus the first m Newton basis functions on the full set X.

Though it is well-known how to perform a pivoted column-based Cholesky decomposition, we describe the algorithm here because we want to make use of the kernel background and to end up with functions, not just with matrices. Having (18) in mind, we start the algorithm by choosing x_1 by permutation of points as

$$x_1 = \arg\max\{K(x,x) : x \in X\}.$$

Since we always assume the kernel to be easily evaluated, this will use $\mathcal{O}(N)$ operations, and we store the vector

$$z := (K(x_1,x_1), \dots, K(x_N,x_N))^T$$

for later use. The first Newton basis function then is

$$N_1(x) := \frac{K(x,x_1)}{\sqrt{K(x_1,x_1)}} \tag{21}$$

due to (3), and we evaluate it on X, giving us the first column v_1 of the value matrix V_N. Another N-vector w should store the values $N_1^2(x_j),\ 1 \le j \le N$. In the following, we denote the Newton basis *functions* by $N_1, N_2, \dots$, while their *values* on X are columns $v_1, v_2, \dots$ of the value matrix V_N.

Assume now that we have fixed the first m points and the first m columns of V_N, forming an $N \times m$ matrix V_m. The vector w should contain the values

$$w_j = \sum_{k=1}^{m} N_k^2(x_j), \quad 1 \le j \le N.$$

For points $x_j \in X$, the power function P_m is

$$P_m^2(x_j) = K(x_j, x_j) - \sum_{k=1}^{m} N_k^2(x_j) = z_j - w_j, \quad 1 \le j \le N,$$

and we find its maximum and assume that it is attained at x_{m+1}. Note that the first m components of $z - w$ should be zero, giving us some check for roundoff.

The algorithm stops if this maximum is smaller than a tolerance ϵ^2. In that case, $P_m(x) \le \epsilon$ holds on all of X, and all functions f in $\mathcal{K}_X$ can be replaced by their interpolants s_f in $x_1, \dots, x_m$ with errors

$$|f(x) - s_f(x)| \le \epsilon \|f\|_{\mathcal{K}} \quad \text{for all } x \in X.$$

Thus, using more than our m well-selected points of X is useless if we can tolerate the above error.

If we decide to continue, we now generate the column Ae_{m+1} of A consisting of values $K(x_{m+1}, x_j)$, $1 \le j \le N$ and form the vector $u := Ae_{m+1} - V_m \cdot (V_m^T e_{m+1})$ at cost $\mathcal{O}(Nm)$. This contains the values on X of the function

$$u(x) := K(x, x_{m+1}) - \sum_{j=1}^{m} N_j(x) N_j(x_{m+1}), \tag{22}$$

and this function satisfies

$$(u, N_k)_{\mathcal{K}} = N_k(x_{m+1}) - \sum_{j=1}^{m} N_j(x_{m+1})(N_j, N_k)_{\mathcal{K}} = 0, \quad 1 \le k \le N.$$

Since the span of $N_1, \dots, N_m$ coincides with the span of $K(\cdot, x_1), \dots, K(\cdot, x_m)$, we also have $u_j = u(x_j) = 0$, $1 \le j \le m$, giving us another check on roundoff.

We then define

$$N_{m+1}(x) := \frac{u(x)}{\|u\|_{\mathcal{K}}}$$

and use $\|u\|_{\mathcal{K}}^2 = z_{m+1} - w_{m+1} = P_m^2(x_{m+1})$ for this. To prove this identity, we employ orthonormality of the N_j in (22) to get

$$\begin{aligned} \sum_{j=1}^{m} N_j^2(x_{m+1}) &= \|u - K(\cdot, x_{m+1})\|_{\mathcal{K}}^2 \\ &= \|u\|_{\mathcal{K}}^2 - 2(u, K(\cdot, x_{m+1}))_{\mathcal{K}} + K(x_{m+1}, x_{m+1}) \\ &= \|u\|_{\mathcal{K}}^2 - 2u(x_{m+1}) + K(x_{m+1}, x_{m+1}) \\ &= w_{m+1} = \|u\|_{\mathcal{K}}^2 - 2u_{m+1} + z_{m+1} \end{aligned}$$

and insert $u_{m+1} = z_{m+1} - w_{m+1}$. We update w and add the vector $u/\|u\|_{\mathcal{K}}$ as a new column to N_m to finish step $m + 1$.

This algorithm provides the first m Newton basis functions on N points at cost $\mathcal{O}(Nm^2)$. It is particularly useful if the kernel has a good low-rank approximation and if the user wants results on a large but fixed point set X.

If data $E(f)$ of a function f are given on X, one might simply set up the overdetermined system

$$V_m c = E(f)$$

and solve it in the least-squares sense for a coefficient vector $c \in \mathbb{R}^m$. Another possibility is to take only the first m rows of this system, thus getting away with an interpolant on $x_1, \dots, x_m$. This system is triangular, can be solved at cost $\mathcal{O}(m^2)$ and usually is quite sufficient, because the main algorithm usually is stopped when the power function is very small on all of X.

If we denote the top $m \times m$ part of V_m by L_m, we get the values of the Lagrange basis for nodes $x_1, \dots, x_m$ on all of X as the matrix $V_m \cdot L_m^{-1}$. The "divided differences" for the Newton basis $N_1, \dots, N_m$ are obtainable as $L_m^{-1} E_m(f)$, if we take the first m components of $E(f)$ as $E_m(f)$.

For use with meshless methods, whose bases should be expressed "entirely in terms of nodes" [11], we suggest not to use the Lagrange basis based on function values in $x_1, \dots, x_m$, but rather the Newton basis, the "divided differences" being the parametrization instead of the function values at those nodes. If a result is expressed as a coefficient vector $c \in \mathbb{R}^m$ in this parametrization, the resulting values at all nodes of X are given by $V_m \cdot c$.

If the values of the Newton basis are needed at other points, or if derivatives are to be calculated, we can use the standard equation (6) and insert $V_U = C_U^{-1}$ for a $\mathcal{K}$-orthonormal basis U. Then we get the linear system

$$V_N \cdot N^T(x) = T(x)^T$$

for the Newton basis. If the basis is shortened to m functions, this system is shortened to be $m \times m$, and then it has the lower triangular coefficient matrix L_m we had before. If linear maps $\mathcal{L}$ like derivatives have to be evaluated, we use the system

$$V_N \cdot \mathcal{L}(N^T(\cdot)) = \mathcal{L}(T(\cdot)^T)$$

in shortened form. These evaluations are of complexity $\mathcal{O}(m^2)$ each, which is compatible with the complexity $\mathcal{O}(Nm^2)$ we already had for getting values on N points.

If we are given a specific vector $E(f)$ of data on a point set X, there are fully data-dependent adaptive techniques for approximation of f. A particularly simple one is based on a pivoted QR decomposition and can be viewed as an instance of orthogonal matching pursuit [12] in the column space of the kernel matrix. Starting theoretically from a huge linear system (9) based on N points, the basic idea is to project the right-hand side into the column space and to select those columns that allow to reproduce the right-hand side with good accuracy. In principle, this could be done by applying a column-pivoted QR algorithm to the system

$$\begin{pmatrix} -E(f) & A \\ 1 & \mathbf{0}^T \end{pmatrix} \begin{pmatrix} 1 \\ \alpha \end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ 1 \end{pmatrix}$$

making sure that the first column is used first. Here, we used boldface to distinguish between vectors and scalars. In addition, the QR algorithm should not start from full matrices, but rather work on columns only. Again, this is well-known from numerical linear algebra, but we want to describe a different algorithm that performs data-dependent orthogonal matching pursuit in the Hilbert space $\mathcal{K}$ and uses the Newton basis.

Let X be a large set of N points $x_1, \dots, x_N$, and assume the data vector $E(f) = (f(x_1), \dots, f(x_N))^T$ to be given from a function $f \in \mathcal{K}$ we do not know explicitly, but which we want to approximate. By (2), we know that the data have the semantics

$$f(x_j) = (f, K(\cdot, x_j))_{\mathcal{K}}, \quad 1 \le j \le N$$

though we do not know f. Selecting x_1 to be the point where the data vector $E(f)$ attains its maximum absolute value means that we have selected the kernel translate that will approximate f best in $\mathcal{K}$. We now proceed like in the previous algorithm, forming the first Newton basis function $N_1 := K(\cdot, x_1)/\sqrt{K(x_1, x_1)}$. But then we replace the data of f by the data of the error $f_1 := f - (f, N_1)_{\mathcal{K}} N_1$ of its best approximation by multiples of N_1 or $K(\cdot, x_1)$ in $\mathcal{K}$. For this, we only need

$$(f, N_1)_{\mathcal{K}} = f(x_1)/\sqrt{K(x_1, x_1)}.$$

Then we proceed by choosing x_2 as the point where f_1 attains its maximum absolute value.

This algorithm constructs a Newton basis, but with a different, now f-dependent selection of points. In the notation of the above algorithm, let us assume that we already have V_m and need

$$f_m := f - \sum_{j=1}^{m} (f, N_j)_{\mathcal{K}} N_j = f_{m-1} - (f, N_m)_{\mathcal{K}} N_m \tag{23}$$

on the full set X. To perform this recursively, we look at step $m+1$ and use (22) to get

$$(f, u)_{\mathcal{K}} = f(x_{m+1}) - \sum_{j=1}^{m} (f, N_j)_{\mathcal{K}} N_j(x_{m+1})$$

and, with proper normalization,

$$(f, N_{m+1})_{\mathcal{K}} = (f, u)_{\mathcal{K}} / \|u\|_{\mathcal{K}}$$

with $\|u\|_{\mathcal{K}}^2 = z_{m=1} - w_{m+1}$ as shown before. Thus we have a recursion for these inner products, and inserting them into (23) allows us to find x_{m+1} as the point where f_m attains its maximal absolute value.

This algorithm is orthogonal matching pursuit in $\mathcal{K}$, and it should be terminated when $|(f, N_{m+1})_{\mathcal{K}}|$ is sufficiently small. By orthogonality to the span of $K(\cdot, x_j)$ for $1 \le j \le m$, the result is the interpolant to f on $x_1, \dots, x_m$. It should finally be noted that the method is a reformulation of the greedy method of [13] in terms of the Newton basis.

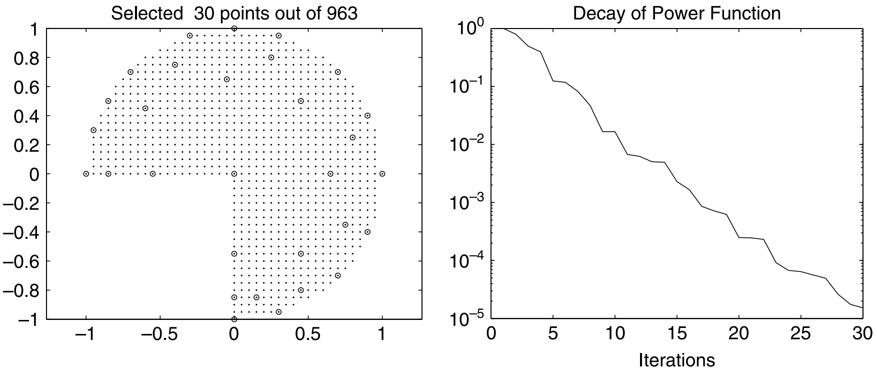

Fig. 1. Selected points and power function decay for the Gaussian.

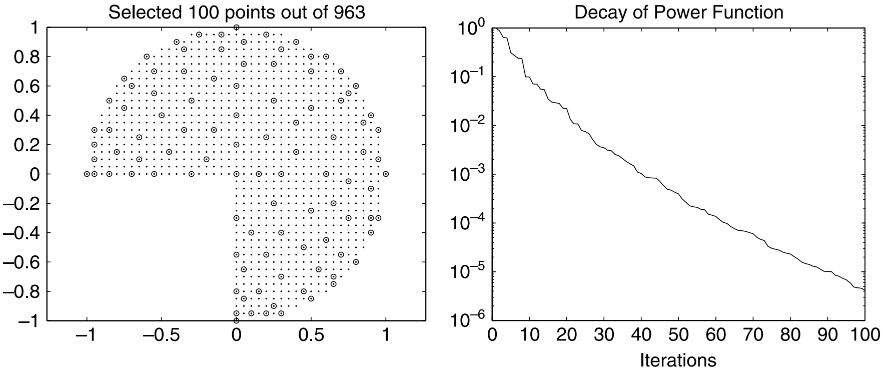

Fig. 2. Selected points and power function decay for an inverse multiquadric.

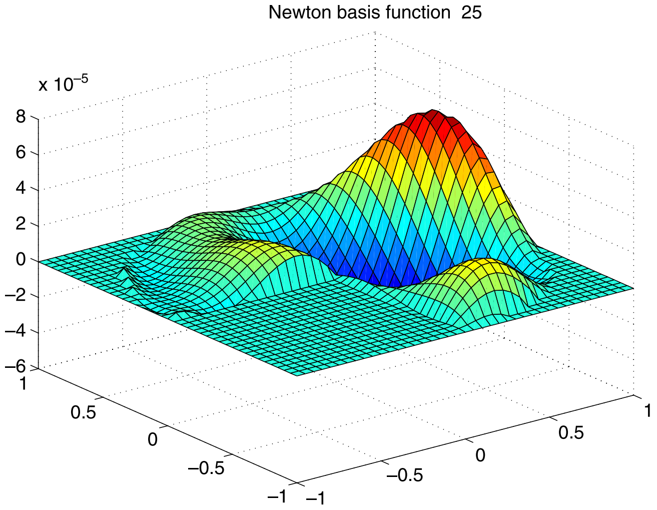

Fig. 3. Newton basis function v_{25} for the Gaussian.

11. Numerical examples

We consider the domain Ω defined by the unit disk with the third quadrant cut away. We select a large set X of points on a fine grid on $[-1, 1]^2$ that fall into Ω. Then we run the adaptive algorithm of Section 10 to generate a selection of well-distributed points. For the Gaussian at scale 2, Fig. 1 shows the first 30 selected points and the decay of the maximum of the power function for that case. For 100 points and inverse multiquadrics of the radial form

$$\phi(r) = (1 + r^2/8)^{-2},$$

similar plots are in Fig. 2. The Newton basis function v_{25} for the Gaussian at scale 2 is in Fig. 3.

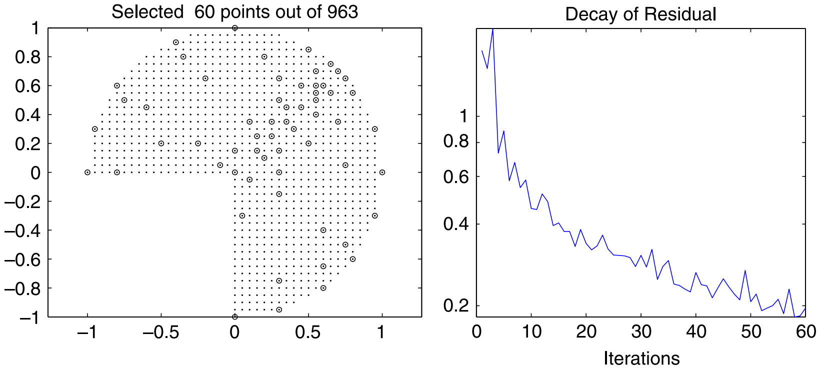

Fig. 4. Data-dependent greedy point selection.

We now turn to the f-dependent point selection strategy. Fig. 4 shows the results for the function $f(x,y) := \exp(|x-y|) - 1$ on the same domain. One can see the accumulation of selected points close to the derivative discontinuity at $x = y$. MATLAB programs are available via the homepage of the second author.

12. Conclusion and outlook

We now have a thorough account of the possibilities to construct bases of data-dependent subspaces of reproducing kernel Hilbert spaces. However, there are many open problems concerning connections between these bases and to the eigenfunction basis. The Newton basis seems to be a particularly good choice, since it is $\mathcal{K}$-orthonormal, allows stable evaluation in the sense of (11) and can be calculated recursively and adaptively. Depending on point selections, convergence rates of series like (18) and (20) should be investigated further.

Acknowledgments

Special thanks go to Davoud Mirzaei for careful proofreading, and to the referees for their comments and suggestions. The first author's work is sponsored by Deutsche Forschungsgemeinschaft via Graduiertenkolleg 1023.

References

[1] M.D. Buhmann, Radial Basis Functions, in: Cambridge Monographs on Applied and Computational Mathematics, Cambridge University Press, 2004.
[2] H. Wendland, Scattered Data Approximation, Cambridge University Press, 2005.
[3] G.F. Fasshauer, Meshfree Approximation Methods with MATLAB, in: Interdisciplinary Mathematical Sciences, vol. 6, World Scientific Publishers, Singapore, 2007.
[4] St. De Marchi, R. Schaback, Stability of kernel-based interpolation, Adv. Comput. Math. 32 (2010) 155–161.
[5] J.C. Mairhuber, On Haar's theorem concerning Chebychev approximation problems having unique solutions, Proc. Amer. Math. Soc. 7 (1956) 609–615.
[6] M.J.D. Powell, A new iterative algorithm for thin plate spline interpolation in two dimensions, Ann. Numer. Math. 4 (1997) 519–526.
[7] A.C. Faul, M.J.D. Powell, Proof of convergence of an iterative technique for thin plate spline interpolation in two dimensions, Appl. Math. Comput. 11 (2–3) (1998) 183–192.
[8] C. de Boor, What is the inverse of a basis? BIT 41 (2001) 880–890. ftp://cs.wisc.edu/Approx/inverse_basis.pdf.
[9] St. Müller, R. Schaback, A Newton basis for kernel spaces, J. Approx. Theory 161 (2009) 645–655. doi:10.1016/j.jat.2008.10.014.
[10] M. Mouattamid, R. Schaback, Recursive kernels, Anal. Theory Appl. 25 (2009) 301–316.
[11] T. Belytschko, Y. Krongauz, D.J. Organ, M. Fleming, P. Krysl, Meshless methods: an overview and recent developments, Comput. Methods Appl. Mech. Eng. 139 (1996) 3–47 (special issue).
[12] S.G. Mallat, Z. Zhang, Matching pursuits with time-frequency dictionaries, IEEE Trans. Signal Process. (1993) 3397–3415.
[13] R. Schaback, H. Wendland, Numerical techniques based on radial basis functions, in: A. Cohen, C. Rabut, L.L. Schumaker (Eds.), Curve and Surface Fitting, Vanderbilt University Press, 2000, pp. 359–374.