Recursive Kernels

Mohammed Mouattamid and Robert Schaback

October 31, 2007

Abstract

This paper is an extension of earlier papers [8, 9] on the "native" Hilbert spaces of functions on some domain $\Omega \subset \mathbb{R}^d$ in which conditionally positive definite kernels are reproducing kernels. Here, the focus is on subspaces of native spaces which are induced via subsets of Ω, and we shall derive a recursive subspace structure of these, leading to recursively defined reproducing kernels. As an application, we get a recursive Neville–Aitken–type interpolation process and a recursively defined orthogonal basis for interpolation by translates of kernels.

Keywords: Radial basis functions, interpolation, scattered data, kernels, multiscale methods

AMS Classification: 41A05,41063, 41065, 65D05, 65D15.

1 Positive Definite Kernels

Let Ω be a nonempty set. If we have a Hilbert space $\mathcal{H}$ over $\mathbb{R}$ consisting of functions on Ω such that the point evaluation functionals

$$\delta_x \; : \; f \mapsto f(x) \text{ for all } x \in \Omega, \; f \in \mathcal{H}$$

are continuous, there is a symmetric *reproducing kernel*

$$K_{\mathcal{H}}(x,y) := (\delta_x, \delta_y)_{\mathcal{H}} \text{ for all } x, y \in \Omega$$

with the *reproduction equation*

$$f(x) = \delta_x(f) = (f, K(x,\cdot))_{\mathcal{H}} \text{ for all } x \in \Omega, \; f \in \mathcal{H}. \tag{1}$$

Such kernels have the additional property to be *positive (semi–) definite* on Ω in a certain sense. For each such kernel, one can also construct a "native" Hilbert space in which it is reproducing. Since we need this Hilbert space and its construction principles, we have to go

into details now. Since there is some confusion in the literature concerning the definition of positive definiteness, we need some notation for a proper definition.

If $X \subseteq \Omega$ is a finite subset, we denote its cardinality by $|X|$ and use coefficient vectors $a \in I\!R^{|X|}$ to define finitely supported linear functionals by

$$\lambda_{a,X}(f) := \sum_{x_j \in X} a_j f(x_j)$$

on the space $I\!R^{\Omega}$ of real–valued functions on Ω which allow point evaluation. These functionals form a linear space

$$L_0(\Omega) := \left\{ \lambda_{a,X} \ : \ X \subseteq \Omega \subset I\!R^d, \ |X| < \infty, \ a \in I\!R^{|X|} \right\}$$

over the real numbers by standard operations.

Definition 1 *A symmetric function*

$$K \ : \ \Omega \times \Omega \to I\!R$$

is called a positive definite kernel on $L_0(\Omega)$*, if the form*

$$\begin{aligned} (\lambda_{a,X}, \lambda_{b,Y})_K &:= \lambda^s_{a,X}, \lambda^t_{b,Y} K(s,t) \\ &= \sum_{x_j \in X} \sum_{y_k \in Y} a_j b_k K(x_j, y_k) \end{aligned}$$

is a symmetric bilinear and positive definite quadratic form on

$$L_0(\Omega) := \left\{ \lambda_{a,X} \ : \ X \subseteq \Omega \subset I\!R^d, \ |X| < \infty, \ a \in I\!R^{|X|} \right\}.$$

Note that we write λ^t to indicate that a functional λ acts with respect to the variable t. Reproducing kernels of Hilbert spaces of functions with continuous point evaluation are always positive definite in the above sense. But there is also another notion:

Definition 2 *A kernel K is* positive (semi–) definite on Ω*, if for all finite sets X the quadratic form*

$$\begin{aligned} I\!R^{|X|} \ni a \ \mapsto \ & (\lambda_{a,X}, \lambda_{a,X})_K \\ & = \lambda^s_{a,X}, \lambda^t_{a,X} K(s,t) \\ & = \sum_{x_j \in X} \sum_{x_k \in X} a_j a_k K(x_j, x_k) \end{aligned}$$

is positive (semi–) definite.

These two notions are not the same, because a kernel K can be positive definite on $L_0(\Omega)$ while being only positive semi–definite on Ω. The main point is that there may be a functional of the form $\lambda_{a,X}$ which is zero as a functional without having zero coefficients. Positive definiteness on Ω thus is a stronger property, and it is satisfied iff the point evaluation functionals *separate points*, i.e. iff point evaluation functionals at different points are linearly independent. The early references [1, 4] make a similar distinction, separating positive definite kernels from what they call positive definite matrices.

If a kernel K is positive definite on $L_0(\Omega)$ without being induced by a Hilbert space $\mathcal{H}$ in which it is reproducing, one can construct an appropriate space from the kernel. In fact, the kernel induces a norm

$$\|\lambda_{a,X}\|_K^2 := (\lambda_{a,X}, \lambda_{a,X})_K \text{ for all } \lambda_{a,X} \in L_0(\Omega)$$

under which $L_0(\Omega)$ is a pre–Hilbert space which can formally be completed to form a Hilbert space $\mathcal{L}_0(\Omega)$.

Having the kernel K at our disposal, we can introduce functions on Ω via

$$f_{a,X}(\cdot) := \lambda_{a,X}^t K(\cdot, t) =: R_K(\lambda_{a,X}) \text{ for all } \lambda_{a,X} \in L_0(\Omega)$$

and this introduces a linear and surjective Riesz–type map

$$R_K \ : \ L_0(\Omega) \to R_K(L_0(\Omega)) =: F_0(\Omega).$$

It satisfies

$$\lambda_{b,Y}(R_K(\lambda_{a,X})) = (\lambda_{a,X}, \lambda_{b,Y})_K$$

for all $\lambda_{a,X}, \lambda_{b,Y} \in L_0(\Omega)$ and thus is injective. If we define a bilinear form on $F_0(\Omega)$ by

$$(f_{a,X}, f_{b,Y})_K := (R_K^{-1} f_{a,X}, R_K^{-1} f_{b,Y})_K = (\lambda_{a,X}, \lambda_{b,Y})_K$$

for all $\lambda_{a,X}, \lambda_{b,Y} \in L_0(\Omega)$, the map R_K is the usual Riesz isometry, and it extends to the completions as

$$\mathcal{R}_K \ : \ \mathcal{L}_0(\Omega) \to \mathcal{R}_K(\mathcal{L}_0(\Omega)) =: \mathcal{F}_0(\Omega).$$

Note that the definition of R_K is the same for all domains Ω, and thus we drop Ω in the notation for R_K and $\mathcal{R}_K$. The kernel K is reproducing on the Hilbert space $\mathcal{F}_0(\Omega)$ in the generalized sense

$$\lambda(f) = (f, \mathcal{R}_K(\lambda))_K \text{ for all } \lambda \in \mathcal{L}_0(\Omega),\ f \in \mathcal{F}_0(\Omega) \tag{2}$$

which specializes to (1) on $\mathcal{H} := \mathcal{F}_0(\Omega)$ when setting $\lambda := \delta_x$ and using $R_K(\delta_x)(y) = K(x, y)$.

Altogether, we see that there is a one–to–one connection between positive definite kernels on $L_0(\Omega)$ and "native" Hilbert spaces of the form $\mathcal{F}_0(\Omega)$ in which these kernels are reproducing. A typical instance is given by the radial kernel

$$K_{m-d/2}(x,y) := \|x-y\|_2^{m-d/2} K_{m-d/2}(\|x-y\|_2),\ x,y \in I\!R^d,\ m > d/2.$$

Starting the above construction from $K_{m-d/2}$ generates Sobolev space $\mathcal{F}_0(I\!R^d) = W_2^m(I\!R^d)$, which is not evident unless one goes backwards and uses Fourier transforms to identify the reproducing kernel of $W_2^m(I\!R^d)$. Other less standard cases arise, for instance, when using the Gaussian or inverse multiquadric kernels

$$K(x,y) := \exp(-\|x-y\|_2^2) \text{ or } K(x,y) := (1+\|x-y\|_2^2)^{-m},\ m > d/2,$$

respectively. They generate "native" Hilbert spaces of global analytic functions on $I\!R^d$.

2 Duals of Native Spaces

Let $\mathcal{H}$ be a Hilbert space of functions on Ω with a reproducing kernel $K_{\mathcal{H}}$. A functional λ in the dual $\mathcal{H}^*$ has a representer $f_\lambda = R(\lambda)$ in $\mathcal{H}$ via the Riesz map $R : \mathcal{H}^* \to \mathcal{H}$. It allows to define the function

$$\lambda^x K_{\mathcal{H}}(x,\cdot) := R(\lambda) \in \mathcal{H}.$$

This can be understood classically for $\lambda = \lambda_{a,X}$, but may be a non-classical action of λ on $K_{\mathcal{H}}$ in general.

If μ is another functional in $\mathcal{H}^*$, we have

$$\mu(\lambda^x K_{\mathcal{H}}(x,\cdot)) = \mu(R(\lambda)) = (\lambda,\mu)_{\mathcal{H}^*} =: \mu^y \lambda^x K_{\mathcal{H}}(x,y).$$

This means that the action of two functionals in $\mathcal{H}^*$ to the two arguments of the reproducing kernel is always well–defined. In special cases, e.g. for Sobolev spaces, the above argument implies that the kernel roughly has twice the (generalized) differentiability of the elements of its native Hilbert space.

But there is another important application. Consider a linear and continuous mapping L from $\mathcal{H}$ to some other Hilbert space $\mathcal{V}$. Then L^*L maps $\mathcal{H}$ back to itself, and for each $f \in \mathcal{H}$ and $x \in \Omega$ we have

$$\begin{aligned}(L^*Lf)(x) &= (L^*Lf, K_{\mathcal{H}}(x,\cdot))_{\mathcal{H}} \\ &= (Lf, L^y K_{\mathcal{H}}(x,y))_{\mathcal{V}}.\end{aligned}$$

Consequently, the adjoint L^* of L can be represented as

$$(L^*v)(x) = (v, L^y K_{\mathcal{H}}(x,y))_{\mathcal{V}} \text{ for all } x \in \Omega,\ v \in \mathcal{V}.$$

3 Subspaces of Native Spaces

We now look at two kinds of subspaces of native spaces and their connection. They are introduced by subsets $\Omega_0 \subseteq \Omega$ and defined as

$$\begin{array}{lcl} V(\Omega_0,\Omega) & := & \{f \in \mathcal{F}_0(\Omega) \,:\, f(\Omega_0) = \{0\}\} \\ F(\Omega_0,\Omega) & := & \mathcal{F}_0(\Omega_0). \end{array}$$

For the second case, we note that $\mathcal{F}_0(\Omega_0)$ is the native space for K being restricted to the subset Ω_0, and it isometrically embedded in $\mathcal{F}_0(\Omega)$ because we have

$$\begin{array}{lcl} L_0(\Omega_0) & \subseteq & L_0(\Omega) \\ F_0(\Omega_0) & \subseteq & F_0(\Omega) \end{array}$$

due to their definition, and these relations extend as isometric embeddings to the closures.

We now state a simple but important relation between these spaces.

Lemma 1 *Both spaces $V(\Omega_0,\Omega)$ and $F(\Omega_0,\Omega)$ are closed subspaces of $\mathcal{F}_0(\Omega)$, and they are mutually orthogonal.*

Proof: The first space is closed because point evaluation functionals are continuous, and the second is closed by definition. If we have some $f \in V(\Omega_0,\Omega)$ and some $f_{a,X} \in F_0(\Omega_0)$ with $X \subseteq \Omega_0$, the reproduction equation gives

$$\begin{array}{rcl} (f, f_{a,X})_K & = & \lambda_{a,X}(f) \\ & = & \displaystyle\sum_{x_j \in X \subset \Omega_0} a_j f(x_j) \\ & = & 0 \end{array}$$

and this extends to the closure of $F_0(\Omega_0)$. □

For notational convenience, we introduce the complementary orthogonal projectors

$$\begin{array}{rcl|rcl} \Pi_{\Omega_0} & : & \mathcal{F}_0(\Omega) \to \mathcal{F}_0(\Omega), & \Pi_{\Omega_0}(\mathcal{F}_0(\Omega)) & = & \mathcal{F}_0(\Omega_0) \\ (Id - \Pi_{\Omega_0}) & : & \mathcal{F}_0(\Omega) \to \mathcal{F}_0(\Omega), & (Id - \Pi_{\Omega_0})(\mathcal{F}_0(\Omega)) & = & V(\Omega_0,\Omega). \end{array} \tag{3}$$

Example 1 *Assume that a kernel K is positive definite on $\mathbb{R}^d$, like the* radial basis functions

$$K(x,y) := \phi(\|x-y\|_2) \text{ for all } x, y \in \mathbb{R}^d$$

with the Gaussian

$$\phi(r) := \exp(-r^2),\ r \geq 0$$

or the inverse multiquadric

$$\phi(r) := (1 + r^2)^{-\beta/2},\ \beta > 0,\ r \geq 0$$

or the Sobolev/Matern function

$$\phi(r) := K_{m-d/2}(r) r^{m-d/2},\ m > d/2,\ r \geq 0$$

or Wendland's function

$$\phi(r) := (1 - r)_+^4 (1 + 4r),\ r \geq 0, d \geq 3.$$

Then the local native space $\mathcal{F}_0(\Omega)$ *with respect to some general subset* Ω *of* $\mathbb{R}^d$ *is the orthogonal complement in* $\mathcal{F}_0(\mathbb{R}^d)$ *of the functions vanishing outside* Ω*.*

In the last two cases, the global native spaces on $\mathbb{R}^d$ are global Sobolev spaces of orders $m - d/2$ and $(d + 3)/2$, respectively. Then the local native spaces on subsets are completely characterized by Lemma 1. But one needs additional hypotheses on the set Ω to conclude that the local native spaces on subsets are local Sobolev spaces.

Example 2 *If* Ω_0 *is a finite set* $X = \{x_1, \ldots, x_N\}$ *of points of* Ω*, the space* $\mathcal{F}_0(X) = F_0(X)$ *is the space spanned by the functions* $K(\cdot, x_j),\ 1 \leq j \leq N$*, and the projector* Π_X *associates to each function* f *its interpolant* $\Pi_X(f)$ *on* X*. Lemma 1 then restates the standard fact that interpolants based on the functions* $K(\cdot, x_j),\ 1 \leq j \leq N$ *furnish norm–minimal interpolants within the native space.*

But X *need not be finite. For functions* f *in the native space, one can pose transfinite interpolation problems on* arbitrary *sets* X*, and their theoretical solution will be provided by the projector* Π_X*. Note that these transfinite interpolation processes are uniquely solvable if the data are exact and come from a function in the native space. For other data, solvability is not guaranteed.*

Example 3 *Asssume that* Ω *carries a topology and* K *is continuous. By the standard argument*

$$\begin{array}{rcl} f(x) - f(y) & = & (f, K(x,\cdot) - K(y,\cdot))_K \\ |f(x) - f(y)|^2 & \leq & \|f\|_K^2 \|K(x,\cdot) - K(y,\cdot)\|_K^2 \\ & = & \|f\|_K^2 (K(x,x) - 2K(x,y) + K(y,y)) \end{array}$$

this implies that all functions in $\mathcal{F}_0(\Omega)$ *are continuous. In view of the previous example, we can look at infinite sets* $X \subset \Omega$*, and it is then clear that* $V(X, \Omega) = V(\ clos\,(X), \Omega)$ *holds. Furthermore, Lemma 1 implies that an interpolant to* f *on* X *will automatically interpolate also on* $clos\,(X)$*.*

4 Kernels for Subspaces

The previous section gave us some closed subspaces of native spaces. Since these are again Hilbert spaces of functions with continuous point evaluation, we know that they have a unique reproducing kernel. Using the complementary orthogonal projectors of (3), we now want to construct the reproducing kernels on both subspaces. We do this in general and under simplified notation.

Theorem 1 *Let the Hilbert space $\mathcal{H}$ have a reproducing kernel $K_{\mathcal{H}}$ on some set Ω, and consider a subspace $\mathcal{V}$ of $\mathcal{H}$ with the canonical orthogonal projector Π. Then the reproducing kernel on $\mathcal{V} = \Pi(\mathcal{H})$ is*

$$K_{\mathcal{V}}(x,y) := \Pi_y^s \Pi_x^t K_{\mathcal{H}}(t,s),\ x,y \in \Omega \tag{4}$$

and is reproducing $\mathcal{V} = \Pi(\mathcal{H})$ on Ω. Here we used the notation Q_x^t if an operator Q acts with respect to the variable t and then sets $t = x$ in the result.

Proof: Since projectors are self–adjoint, we have

$$\begin{aligned} \Pi_x^t f(t) &= (\Pi_\cdot^t f(t), K_{\mathcal{H}}(x,\cdot))_{\mathcal{H}} \\ &= (f, \Pi_\cdot^t K_{\mathcal{H}}(x,t))_{\mathcal{H}} \\ &= (\Pi_\cdot^s f(s) + (Id - \Pi)_\cdot^s f(s), \Pi_\cdot^t K_{\mathcal{H}}(x,t))_{\mathcal{H}} \\ &= (\Pi_\cdot^s f(s), \Pi_\cdot^t K_{\mathcal{H}}(x,t))_{\mathcal{H}}. \end{aligned}$$

Since $K_{\mathcal{H}}$ is symmetric, we can use $\Pi_\cdot^t K_{\mathcal{H}}(x,t) = \Pi_x^t K_{\mathcal{H}}(t,\cdot)$ to get

$$\begin{aligned} \Pi_x^t f(t) &= (\Pi_\cdot^s f(s), \Pi_x^t K_{\mathcal{H}}(t,\cdot))_{\mathcal{H}} \\ &= (f, \Pi_\cdot^s \Pi_x^t K_{\mathcal{H}}(t,s))_{\mathcal{H}} \\ &= (\Pi_\cdot^s f(s), \Pi_\cdot^s \Pi_x^t K(t,s))_{\mathcal{H}\ \mathcal{H}} \end{aligned}$$

proving (4) because reproducing kernels are unique [1, 4]. □

Note that [1, 4] pursue the opposite direction: they show that if two subspaces are mutually orthogonal, the sum of their respective reproducing kernels is equal to the reproducing kernel for the full space. In the case above, the two orthogonal subspaces are $\mathcal{V} = \Pi(\mathcal{H})$ and $\mathcal{V}^\perp = (Id_{\mathcal{H}} - \Pi)(\mathcal{H})$. Consequently, the kernel $K_{\mathcal{H}}$ on $\mathcal{H}$ has the splitting

$$\begin{aligned} K_{\mathcal{H}}(x,y) &= K_{\mathcal{V}}(x,y) &+ K_{\mathcal{V}^\perp}(x,y) \\ &= \Pi_y^s \Pi_x^t K_{\mathcal{H}}(t,s) &+ (Id_{\mathcal{H}} - \Pi)_y^s (Id_{\mathcal{H}} - \Pi)_x^t K_{\mathcal{H}}(t,s) \end{aligned}$$

with the somewhat strange consequence

$$2\Pi_y^s \Pi_x^t K_{\mathcal{H}}(t,s) = \Pi_x^t K_{\mathcal{H}}(t,y) + \Pi_y^s K_{\mathcal{H}}(x,s)$$

for all $x, y \in \Omega$.

5 Recursive Kernels

We now focus on subspaces induced by subsets. If Ω_0 is s subset of Ω, we denote the projector onto $\mathcal{F}_0(\Omega_0)$ by $\Pi_{\Omega_0,K}$ taking the kernel K into the notation. If we want to interpolate a function $f \in \mathcal{F}_0(\Omega)$ on Ω_0, we can take the function $\Pi_{\Omega_0,K}(f)$. We now have the residual function

$$f_0 := f - \Pi_{\Omega_0,K}(f) \in V(\Omega_0, \Omega),$$

and we can work on the residual f_0 in the space $V(\Omega_0, \Omega) = \mathcal{F}_0(\Omega_0)^\perp$ from now on. As we know from the previous section, the reproducing kernel there is

$$K_{\Omega_0,K}(x, y) := (Id - \Pi_{\Omega_0,K})^s_y (Id - \Pi_{\Omega_0,K})^t_x K(s, t).$$

Now assume that we enlarge the set Ω_0 to some set $\Omega_1 := \Omega_0 \cup X_0 \subseteq \Omega$ by adding a set X_0 of points not in Ω_0. We interpolate f_0 on X_0 by translates of $K_{\Omega_0,K}$, and this is formally done by the projector $\Pi_{X_0,K_{\Omega_0,K}}$. The interpolant to f on Ω_1 can then be written as

$$\begin{array}{rcl} & \Pi_{\Omega_0,K}(f) + \Pi_{X_0,K_{\Omega_0,K}}(f_0) \\ = & \Pi_{\Omega_0,K}(f) + \Pi_{X_0,K_{\Omega_0,K}}(f - \Pi_{\Omega_0,K}(f)) \\ = & \Pi_{\Omega_1,K}(f). \end{array}$$

Thus we can write the new projector onto Ω_1 as

$$\begin{array}{rcl} \Pi_{\Omega_1,K} & = & \Pi_{\Omega_0,K} + \Pi_{X_0,K_{\Omega_0,K}}(Id - \Pi_{\Omega_0,K}) \\ Id - \Pi_{\Omega_1,K} & = & (Id - \Pi_{X_0,K_{\Omega_0,K}})(Id - \Pi_{\Omega_0,K}) \end{array}$$

and the residual is

$$\begin{array}{rcl} f_1 & := & f_0 - \Pi_{X_0,K_{\Omega_0,K}}(f_0) \\ & = & (Id - \Pi_{X_0,K_{\Omega_0,K}})(f_0) \\ & = & (Id - \Pi_{X_0,K_{\Omega_0,K}})(Id - \Pi_{\Omega_0,K})(f) \\ & = & (Id - \Pi_{\Omega_1,K})(f). \end{array}$$

The reproducing kernel on the space $V(\Omega_1, \Omega)$ then is

$$\begin{array}{rcl} K_{\Omega_1,K}(x, y) & := & (Id - \Pi_{\Omega_1,K})^s_y (Id - \Pi_{\Omega_1,K})^t_x K(s, t) \\ & = & (Id - \Pi_{X_0,K_{\Omega_0,K}})^s_y (Id - \Pi_{X_0,K_{\Omega_0,K}})^t_x K_{\Omega_0,K}(s, t) \\ & = & ((Id - \Pi_{X_0,K_{\Omega_0,K}})(Id - \Pi_{\Omega_0,K}))^s_y \\ & & ((Id - \Pi_{X_0,K_{\Omega_0,K}})(Id - \Pi_{\Omega_0,K}))^t_x K(s, t). \end{array}$$

This allows to calculate these kernels recursively.

If we proceed stepwise, adding a single point at each step, we can define $\Omega_0 := \{z\}$ for some $z \in \Omega$. Then we have

$$(\Pi_{\Omega_0,K}(f))(x) = \frac{K(x, z)}{K(z, z)} f(z)$$

due to the interpolation property at z, and the new kernel is

$$\begin{aligned}
K_{\{z\},K}(x,y) &:= (Id - \Pi_{\Omega_0,K})^s_y (Id - \Pi_{\Omega_0,K})^t_x K(s,t) \\
&= K(x,y) - (\Pi_{\Omega_0,K})^s_y K(x,s) - (\Pi_{\Omega_0,K})^t_x K(y,t) \\
&\quad + (\Pi_{\Omega_0,K})^s_y (\Pi_{\Omega_0,K})^t_x K(s,t) \\
&= K(x,y) - \frac{K(y,z)}{K(z,z)} K(x,z) - \frac{K(x,z)}{K(z,z)} K(z,y) \\
&\quad + \frac{K(y,z)}{K(z,z)} K(z,z) \frac{K(x,z)}{K(z,z)} \\
&= K(x,y) - \frac{K(y,z)K(x,z)}{K(z,z)}.
\end{aligned}$$

If we recursively use points $z_1, \ldots, z_n, \ldots$ we can define

$$\begin{aligned}
K_0(x,y) &:= K(x,y) \\
K_j(x,y) &:= K_{j-1}(x,y) - \frac{K_{j-1}(y,z_j)K_{j-1}(x,z_j)}{K_{j-1}(z_j,z_j)}
\end{aligned}$$

to define kernels K_j that vanish if one of the arguments is in $\{z_1, \ldots, z_j\}$ and which are the kernels we would have called $K_{\Omega_j,K}$ with $\Omega_j = \{z_1, \ldots, z_j\}$ above.

6 Recursive Interpolation

We can now use the above arguments to define a recursive Neville–Aitken type of interpolation without solving linear systems [7]. Assume we are given a domain Ω, a kernel K on Ω, a function f on Ω and points $z_1, \ldots, z_j, \ldots \in \Omega$ to interpolate f in. The algorithm below can be carried out either for a fixed $x \in \Omega$ or in a space of functions on Ω when treating x as a variable.

Start: Define

$$\begin{aligned}
s_0(x) &:= 0, \\
r_0(x) &:= f(x) - s_0(x) = f(x) \\
K_0(x,y) &:= K(x,y).
\end{aligned}$$

Iteration: For $j = 1, 2, \ldots ...$

$$\begin{aligned}
s_j(x) &:= s_{j-1}(x) + r_{j-1}(z_j) \frac{K_{j-1}(x,z_j)}{K_{j-1}(z_j,z_j)} \\
r_j(x) &:= f(x) - s_j(x) \\
&= f(x) - s_{j-1}(x) - r_{j-1}(z_j) \frac{K_{j-1}(x,z_j)}{K_{j-1}(z_j,z_j)} \\
&= r_{j-1}(x) - r_{j-1}(z_j) \frac{K_{j-1}(x,z_j)}{K_{j-1}(z_j,z_j)} \\
K_j(x,y) &:= K_{j-1}(x,y) - \frac{K_{j-1}(y,z_j)K_{j-1}(x,z_j)}{K_{j-1}(z_j,z_j)}.
\end{aligned}$$

In the course of this algorithm, the functions f and r_j are evaluated only at the data locations. The function s_j will be the interpolant to f on $z_1, \ldots, z_j$. In matrix–vector form, and with evaluation to be done for points $x_1, \ldots, x_m \in \Omega$, the algorithm should work with

$$\begin{array}{rcl} S_j & := & (s_j(x_1), \ldots, s_j(x_m))^T \in I\!R^m \\ R_j & := & (r_j(z_1), \ldots, r_j(z_n))^T \in I\!R^n \\ & = & (0, \ldots, 0, r_j(z_{j+1}), \ldots, r_j(z_n))^T \\ C_j & := & (K_j(z_i, z_k))_{1 \le i,k \le n} \in I\!R^{n,n} \\ B_j & := & (K_j(x_i, z_k))_{1 \le i \le m,\ 1 \le j \le n} \in I\!R^{m,n} \end{array} \tag{5}$$

and proceed in MATLAB–type notation via the in–place iteration

$$\begin{array}{rcl} S_j & = & S_{j-1} + B_{j-1}(:,j) * R_{j-1}(j)/C_{j-1}(j,j) \\ R_j & = & R_{j-1} - C_{j-1}(:,j) * R_{j-1}(j)/C_{j-1}(j,j) \\ B_j & = & B_{j-1} - B_{j-1}(:,j) * C_{j-1}(j,:)/C_{j-1}(j,j) \\ C_j & = & C_{j-1} - C_{j-1}(:,j) * C_{j-1}(j,:)/C_{j-1}(j,j) \end{array} \tag{6}$$

starting from

$$\begin{array}{rcl} S_0 & := & 0 \in I\!R^m \\ R_0 & := & (f(z_1), \ldots, f(z_n))^T \in I\!R^n \\ C_0 & := & (K(z_i, z_k))_{1 \le i,k \le n} \in I\!R^{n,n} \\ B_0 & := & (K(x_i, z_k))_{1 \le i \le m,\ 1 \le k \le n} \in I\!R^{m,n}. \end{array}$$

The kernels K_j were called **power kernels** in [5] because they are related to the **power function** $P_{\Omega_j,K}(x)$ allowing to bound the pointwise interpolation error by

$$|f(x) - s_j(x)| \le P_{\Omega_j,K}(x) \|f\|_{\mathcal{H}} \text{ for all } x \in \Omega,\ f \in \mathcal{H}.$$

The connection [7, 5] is

$$P^2_{\Omega_j,K}(x) := K_j(x,x) \text{ for all } x \in \Omega.$$

If the user does not want the value $s_j(x)$ but rather some derivative of s_j at x or the value of some general linear functional λ on s_j, one can extend the iteration on the $s_j(x)$ by

Start: $\lambda_0 := \lambda(s_0) = 0$

Iteration:

$$\lambda_j := \lambda(s_j) \quad = \quad \lambda_{j-1} + r_{j-1}(z_j) \frac{\lambda^x K_{j-1}(x, z_j)}{K_{j-1}(z_j, z_j)}.$$

This allows any derivative of the interpolant to be recursively calculated "*entirely in terms of nodes*", as postulated for **meshless methods** [2], but without solving a linear system and without excessive storage.

However, we do not claim that the above algorithm is computationally more efficient or more stable than the standard approaches. It can be viewed as a disguised form of Cholesky decomposition, if we introduce the matrices (5) and see that they have the in–place Cholesky–type recursion given by the last line of (6). This reduces the original kernel matrix C_0 stepwise to matrices C_j which have the first j rows and columns set to zero, the remaining square part still being positive definite. It is remarkable that orthogonality arguments in Hilbert space naturally lead to such an algorithm.

There is another observation to be made. The above procedure generates a new basis

$$K_0(\cdot, z_1), \ldots, K_{n-1}(\cdot, z_n)$$

of the space

$$\text{span } \{K(\cdot, z_j) \ : \ 1 \leq j \leq n\}.$$

and this new basis is orthogonal in the native Hilbert space. Indeed, for $1 \leq j < k \leq n$ we have

$$\begin{aligned}
(K_{j-1}(\cdot, z_j), K_{k-1}(\cdot, z_k))_{\mathcal{H}} &= \left(\sum_{i=1}^{j} \alpha_i K(\cdot, z_i), K_{k-1}(\cdot, z_k) \right)_{\mathcal{H}} \\
&= \sum_{i=1}^{j} \alpha_i \left(K(\cdot, z_i), K_{k-1}(\cdot, z_k) \right)_{\mathcal{H}} \\
&= \sum_{i=1}^{j} \alpha_i K_{k-1}(z_i, z_k) \\
&= 0,
\end{aligned}$$

but we already know this from our construction principle based on orthogonality. An investigation of bases with similar orthogonality properties is on its way [6].

At this point, we suppress numerical examples, because it turned out that implementation of the recursive method does not lead to unexpected results. If not too many evaluation points x_i are required (i.e. $m \leq \mathcal{O}(n)$), the algorithm is roughly comparable to the standard approach in both efficiency and stability, provided that it is written in matrix form like (6) and with certain simple enhancements to avoid repeated calculations of already known values or calculation of zeros known beforehand. A MATLAB program can be obtained from the second author on request.

7 Generalized Interpolation

We can extend the above algorithm to general data. Instead of given points $z_1, \ldots, z_j, \ldots$ we assume data in the form of linear functionals

$\lambda_1, \ldots, \lambda_j, \ldots$ from the dual of the native Hilbert space $\mathcal{H}$ of the kernel K on Ω. The range of admissible functionals is discussed in section 2.

Generalized interpolation of a function $f \in \mathcal{H}$ using these data then means to find a function $s_j \in \mathcal{H}$ such that

$$\lambda_k(s_j) = \lambda_k(f),\ 1 \le k \le j,$$

and the previous situation is the special case $\lambda_j = \delta_{z_j}$. Instead of calculating a single function value $s_j(x)$, we can calculate a general data value $\mu(s_j)$ for some functional $\mu \in \mathcal{H}^*$. The algorithm then is

Start: Define

$$\begin{aligned} s_0 &:= 0, \\ \mu_0 := \mu(s_0) &:= 0, \\ r_0(z_1) &:= f(z_1) \\ K_0(x,y) &:= K(x,y). \\ \mu^x K_0(x,y) &:= \mu^x K(x,y). \end{aligned}$$

Iteration: For $j = 1, 2, \ldots...$ and $j < i, k \le n$

$$\begin{aligned} \mu_j := \mu(s_j) &:= \mu_{j-1} + \lambda_j^t r_{j-1}(t) \frac{\mu^x \lambda_j^t K_{j-1}(x,t)}{\lambda_j^s \lambda_j^t K_{j-1}(s,t)} \\ \lambda_i^x r_j(x) &= \lambda_i^x r_{j-1}(x) - \lambda_j^t r_{j-1}(t) \frac{\lambda_i^x \lambda_j^t K_{j-1}(x,t)}{\lambda_j^s \lambda_j^t K_{j-1}(s,t)} \\ \lambda_i^x \lambda_k^y K_j(x,y) &:= \lambda_i^x \lambda_k^y K_{j-1}(x,y) - \frac{\lambda_j^t \lambda_k^y K_{j-1}(y,t) \lambda_i^x \lambda_j^s K_{j-1}(s,x)}{\lambda_j^s \lambda_j^t K_{j-1}(s,t)}. \\ \lambda_i^x \mu^y K_j(x,y) &:= \lambda_i^x \mu^y K_{j-1}(x,y) - \frac{\lambda_j^t \mu^y K_{j-1}(y,t) \lambda_i^x \lambda_j^s K_{j-1}(s,x)}{\lambda_j^s \lambda_j^t K_{j-1}(s,t)} \end{aligned}$$

where we suppressed the recursions

$$\begin{aligned} s_j(x) &:= s_{j-1}(x) + \lambda_j^t r_{j-1}(t) \frac{\lambda_j^t K_{j-1}(x,t)}{\lambda_j^s \lambda_j^t K_{j-1}(s,t)} \\ r_j(x) &:= f(x) - s_j(x) \\ &= f(x) - s_{j-1}(x) - \lambda_j^t r_{j-1}(t) \frac{\lambda_j^t K_{j-1}(x,t)}{\lambda_j^s \lambda_j^t K_{j-1}(s,t)} \\ &= r_{j-1}(x) - \lambda_j^t r_{j-1}(t) \frac{\lambda_j^t K_{j-1}(x,t)}{\lambda_j^s \lambda_j^t K_{j-1}(s,t)} \\ K_j(x,y) &:= K_{j-1}(x,y) - \frac{\lambda_j^t K_{j-1}(y,t) \lambda_j^s K_{j-1}(s,x)}{\lambda_j^s \lambda_j^t K_{j-1}(s,t)} \end{aligned}$$

needed to understand how the algorithm works. Note that the functions $f,\ s_j$ and r_j need not be calculated at any point. We just need the data $\lambda_k(f),\ 1 \le k \le n$ and calculate the data $\lambda_i(r_j),\ 1 \le j < i \le n$.

Formally, the function s_j will be the interpolant to f using the generalized data. When redefining the matrices suitably, the iteration of this algorithm in vector–matrix form coincides with (6), but we leave details to the reader.

8 Polynomials

In order to deal with *conditionally* positive definite kernels in the next section, we need some basic facts on polynomials, functions, and functionals. We fix a set $\Omega \subseteq \mathbb{R}^d$ and denote the space of d–variate polynomials of order at most $m \geq 0$ on $\mathbb{R}^d$ by $\mathbb{P}_m^d$. For the rest of the paper, the integers $d \geq 1$ and $m \geq 0$ are fixed, and we suppress them in the notation from now on, using $\mathbb{P} := \mathbb{P}_m^d$.

A subset X of $\mathbb{R}^d$ is called $\mathbb{P}$–unisolvent, if

$$p(X) = \{0\} \text{ implies } p = 0,$$

and we use the shorthand notation $X \in U$ for this. If X is finite, we denote its cardinality by $|X|$ and use coefficient vectors $a \in \mathbb{R}^d$ to define finitely supported linear functionals by

$$\lambda_{a,X}(f) := \sum_{x_j \in X} a_j f(x_j)$$

on the space $\mathbb{R}^\Omega$ of real–valued functions on Ω which allow point evaluation. Then we define

$$L(\Omega) := \left\{ \lambda_{a,X} \ : \ \begin{array}{l} X \subseteq \Omega \subset \mathbb{R}^d,\ |X| < \infty,\ X \in U \\ a \in \mathbb{R}^{|X|},\ \lambda_{a,X}(\mathbb{P}) = \{0\} \end{array} \right\}$$

as a space of finitely supported functionals on $\mathbb{R}^\Omega$. Note that this is a vector space over $\mathbb{R}$ with the usual operations, and it is a subspace of the algebraic dual $(\mathbb{R}^\Omega)'$ of $\mathbb{R}^\Omega$.

For later use, we need a special tool we introduce now. If X is $\mathbb{P}$–unisolvent, we can reproduce any polynomial $p \in \mathbb{P}$ uniquely from its values on X by a formula like

$$p(x) = \sum_{x_j \in X_0} p_j^X(x) p(x_j) \text{ for all } x \in \mathbb{R}^d,\ p \in \mathbb{P} \tag{7}$$

using a Lagrange–type basis consisting of polynomials $p_j^X \in \mathbb{P}$ for points x_j from some *finite* $\mathbb{P}$–unisolvent subset X_0 of X.

9 Conditionally Positive Definite Kernels

Definition 3 *A function*

$$K \ : \ \Omega \times \Omega \to I\!R$$

is called a conditionally positive definite kernel of order m on Ω, if the form

$$\begin{aligned}(\lambda_{a,X}, \lambda_{b,Y})_K &:= \lambda^s_{a,X}, \lambda^t_{b,Y} K(s,t) \\ &= \sum_{x_j \in X} \sum_{y_k \in Y} a_j b_k K(x_j, y_k)\end{aligned}$$

is a symmetric bilinear and definite quadratic form on $L(\Omega)$.

Note that we write λ^t to indicate that a functional λ acts with respect to the variable t.

If a kernel K satisfies the above definition, it induces a norm

$$\|\lambda_{a,X}\|^2_K := (\lambda_{a,X}, \lambda_{a,X})_K \text{ for all } \lambda_{a,X} \in L(\Omega)$$

under which $L(\Omega)$ is a pre–Hilbert space which can formally be completed to form a Hilbert space $\mathcal{L}(\Omega)$.

To extend the recursive construction of kernels and interpolants to the conditionally positive definite case, we fix a minimal $I\!P$–unisolvent set $\Xi \subseteq \Omega$ and define the projector

$$\Pi_\Xi(f) := \sum_{\xi_j \in \Xi} p^\Xi_j(\cdot) f(\xi_j)$$

using the terminology of (7).

Theorem 2 *[3, 8] Under the above assumptions, the kernel*

$$K_\Xi(x,y) := (Id - \Pi_\Xi)^s_x (Id - \Pi_\Xi)^t_y K(s,t)$$

is positive definite on $\Omega \setminus \Xi$. □

This makes it easy to deal with conditionally positive definite kernels. Given a large set X of data points, we first select a minimal $I\!P$–unisolvent subset Ξ and perform the step above. Then we continue to work on $\Omega \setminus \Xi$ using the data points from $X \setminus \Xi$ and the kernel K_Ξ. This can be done along the lines of section 6. In other words, conditional positive definiteness just calls for an initial step towards positive definiteness.

References

[1] N. Aronszajn. Theory of reproducing kernels. *Trans. Amer. Math. Soc.*, 68:337–404, 1950.

[2] T. Belytschko, Y. Krongauz, D.J. Organ, M. Fleming, and P. Krysl. Meshless methods: an overview and recent developments. *Computer Methods in Applied Mechanics and Engineering, special issue*, 139:3–47, 1996.

[3] W.A. Light and H. Wayne. Spaces of distributions, interpolation by translates of a basis function and error estimates. *Numer. Math.*, 81:415–450, 1999.

[4] H. Meschkowski. *Hilbertsche Räume mit Kernfunktion*. Springer, Berlin, 1962.

[5] M. Mouattamid. Theory of power kernels. Doctoral dissertation, Universität Göttingen, 2005.

[6] St. Müller and R. Schaback. Newton bases for kernel spaces. Preprint Göttingen, 2007.

[7] R. Schaback. Reconstruction of multivariate functions from scattered data. Manuscript, available via http://www.num.math.uni-goettingen.de/schaback/research/group.html, 1997.

[8] R. Schaback. Native Hilbert spaces for radial basis functions I. In M.D. Buhmann, D. H. Mache, M. Felten, and M.W. Müller, editors, *New Developments in Approximation Theory*, number 132 in International Series of Numerical Mathematics, pages 255–282. Birkhäuser Verlag, 1999.

[9] R. Schaback. A unified theory of radial basis functions (Native Hilbert spaces for radial basis functions II). *J. Comp. Appl. Math.*, 121:165–177, 2000.