Interpolation with Variably Scaled Kernels

MIRA BOZZINI[†]
*Dipartimento di Matematica e Applicazioni, Università di Milano–Bicocca,
Via Roberto Cozzi 53, 20125 Milano, Italy*

LICIA LENARDUZZI[‡]
*CNR, IMATI Milano Department,
via Bassini 15, 20133 Milano, Italy*

MILVIA ROSSINI[§]
*Dipartimento di Matematica e Applicazioni, Università di Milano–Bicocca,
Via Roberto Cozzi 53, 20125 Milano, Italy*

ROBERT SCHABACK[¶]
*Institut für Numerische und Angewandte Mathematik Fakultt fr Mathematik und Informatik,
Universität Göttingen Lotzestraße 16-8, D–37073 Göttingen, Germany*

[Received on 28 August 2013]

Within kernel–based interpolation and its many applications, it is a well–documented but unsolved problem to handle the scaling or the *shape parameter*. We consider native spaces whose kernels allow us to change the kernel scale of a d–variate interpolation problem locally, depending on the requirements of the application. The trick is to define a *scale function* c on the domain $\Omega \subset \mathbf{R}^d$ to transform an interpolation problem from data locations x_j in $\mathbf{R}^d$ to data locations $(x_j, c(x_j))$ and to use a fixed–scale kernel on $\mathbf{R}^{d+1}$ for interpolation there. The $(d+1)$–variate solution is then evaluated at $(x, c(x))$ for $x \in \mathbf{R}^d$ to give a d–variate interpolant with a varying scale. A large number of examples show how this can be done in practice to get results that are better than the fixed–scale technique, with respect to both condition and error. The background theory coincides with fixed–scale interpolation on the submanifold of $\mathbf{R}^{d+1}$ given by the points $(x, c(x))$ of the graph of the *scale function* c.

Keywords: positive definite radial basis functions; shape parameter; scaling.

1. Introduction

A symmetric *kernel*

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

defined on a domain $\Omega \subset \mathbf{R}^d$ is very useful for a variety of purposes from interpolation or approximation to PDE solving, if certain *nodes* or *centers* $x_1, \ldots, x_N \subset \mathbf{R}^d$ are used to define *kernel translates* $K(\cdot, x_j)$ as *trial functions*. If the kernel is positive (semi–) definite, i.e. the *kernel matrices* with elements $K(x_i, x_j)$, $1 \leqslant i, j \leqslant N$ are positive (semi–) definite for all choices of nodes, there is a *native* Hilbert

[†] Email: mira.bozzini@unimib.it

[‡] licia@mi.imati.cnr.it

[§] Corresponding author. Email: milvia.rossini@unimib.it

[¶] Email: schaback@math.uni-goettingen.de space $\mathscr{H}$ in the background in which the kernel is reproducing, i.e.

$$f(x) = (f, K(\cdot, x))_{\mathscr{H}} \text{ for all } x \in \Omega,\, f \in \mathscr{H}.$$

The use of reproducing kernels in Hilbert spaces leads to various optimality properties and plenty of applications, but we leave details to the background literature [1, 6, 25, 23, 11].

If the kernel is *translation–invariant* on $\mathbf{R}^d$, it is of the form $K(x,y) = \Phi(x-y)$ for all $x,y \in \mathbf{R}^d$. In a geostatistical context, kernels are viewed as covariances of spatial ramdom variables, and then translation–invariant kernels are called *stationary*. If the kernel is *radial*, i.e. of the form $K(x,y) = \phi(\|x-y\|_2)$ for a scalar function $\phi : [0,\infty) \to \mathbf{R}$, the function ϕ is called a *radial basis function*. There also are *compactly supported kernels* which vanish outside the unit ball, e.g. the very useful *Wendland* kernels [24, 25].

Kernels on $\mathbf{R}^d$ can be *scaled* by a positive factor c by going over to the new kernel

$$K_c(x,y) := K(x/c, y/c) \text{ for all } x,y \in \mathbf{R}^d. \tag{1.1}$$

In case of a radial kernel supported on the unit ball, the support of the scaled kernel now has support radius c. Large c increase the condition of kernel matrices, while small c let the translates turn into sharp peaks which approximate functions badly, if separated too far from each other.

The choice of a *scaling* or *shape parameter c* of a kernel or radial basis function is a problem that is around for more than two decades [15, 20, 4, 5, 10, 13]. There are purely experimental reports on the behavior of kernel–based methods under scaling, and there are optimization techniques that try to find a good compromise between bad condition and good reproduction quality. A special case of scaling is the *flat limit* $c \to \infty$ which is not considered here [9, 12, 18, 19, 21, 22].

Users would like to have the scale of a kernel translate vary with the translation. This means working with trial functions $\phi(\|x-x_j\|_2/c_j)$, $1 \leqslant j \leqslant N$ in the radial case, but it is easy to come up with examples that let interpolation in the nodes fail for certain nonuniform choices of scale.

This paper mimics the case of varying scales by letting the scale parameter be an additional coordinate. This allows varying scales without leaving the firm grounds of kernel–based interpolation. It turns out that this approach can be fully understood as the standard single–scale method applied to a certain submanifold of $\mathbf{R}^{d+1}$, and some numerical examples show that the method works quite satisfactorily in cases that are spoiled by excessive instability of the standard method.

2. Basics

Let K be a positive definite kernel on $\mathbf{R}^{d+1}$. We assume readers to be familiar with this notion and with the standard examples of radial kernels, see e.g. [25].

Definition 1 If a *scale function* $c : \mathbf{R}^d \to (0,\infty)$ is given, we can define a *variably scaled* kernel on $\mathbf{R}^d$ by

$$K_c(x,y) := K((x,c(x)),(y,c(y))) \text{ for all } x,y \in \mathbf{R}^d. \tag{2.1}$$

Note that this definition is different from (1.1). We explain the differences below for special cases of radial kernels.

The following is easy to prove.

Theorem 1 If K is positive (semi–) definite, so is K_c.
If K and c are continuous, so is K_c. □

If K is positive definite, interpolation of values $f_1, \dots, f_N$ on $X := \{x_1, \dots, x_N\}$ proceeds via solving a linear system

$$A_{c,X}a = f \in \mathbf{R}^N$$

with the kernel matrix $A_{c,X} := (K_c(x_j, x_k))_{1 \leqslant j,k \leqslant N}$, which is positive definite. The coefficient vector $a \in \mathbf{R}^N$ then allows the solution function to be written as

$$s_{c,X,f}(x) := \sum_{j=1}^{N} a_j K_c(x, x_j) = \sum_{j=1}^{N} a_j K((x, c(x)), (x_j, c(x_j))). \tag{2.2}$$

2.1 *Radial Kernels*

If the kernel is radial, i.e. $K(x,y) = \phi(\|x-y\|_2^2)$, then the kernel matrix entries are

$$\phi(\|x_j - x_k\|^2 + (c(x_j) - c(x_k))^2)$$

and the interpolant is

$$s_{c,X,f}(x) := \sum_{j=1}^{N} a_j \phi(\|x - x_j\|_2^2 + (c(x) - c(x_j))^2).$$

It is identical to the standard kernel interpolant if the scale function is constant.

2.2 *Power Kernels*

Consider K to be a radial power kernel $\phi(r) = r^\beta$. Then

$$\begin{array}{rcl} K_c(x,y) & = & \left(\|x-y\|_2^2 + (c(x) - c(y))^2\right)^{\beta/2} \\ & = & |c(x) - c(y)|^\beta \left(\dfrac{\|x-y\|_2^2}{(c(x)-c(y))^2} + 1 \right)^{\beta/2} \end{array}$$

is close to (but not identical) to a scaled multiquadric kernel. The interpolants take the form

$$s_{c,X,f}(x) := \sum_{j=1}^{N} a_j \left(\|x_j - x\|_2^2 + (c(x_j) - c(x))^2\right)^{\beta/2}$$

and are similar to scaled multiquadrics, while identical to power interpolants if the scale function is constant.

2.3 *Multiquadrics*

Consider K to be the standard multiquadric. Then

$$K_c(x,y) \quad = \quad \left(\|x-y\|_2^2 + (c(x) - c(y))^2 + 1\right)^{\beta/2}$$

and the interpolants take the form

$$s_{c,X,f}(x) := \sum_{j=1}^{N} a_j \left(\|x_j - x\|_2^2 + (c(x_j) - c(x))^2 + 1\right)^{\beta/2}.$$

which can be seen as a variably scaled multiquadric.

2.4 *Gaussian*

Consider K to be the Gaussian. Then

$$K_c(x,y) \quad = \quad \exp(-\|x-y\|_2^2)\exp\left(-(c(x)-c(y))^2\right)$$

and the interpolants take the form

$$s_{c,X,f}(x) := \sum_{j=1}^{N} a_j \exp(-\|x_j-x\|_2^2)\exp\left(-(c(x_j)-c(x))^2\right)$$

which can be seen as a superposition of Gaussians of the same scale but with varying amplitudes for evaluation.

2.5 *Compactly Supported Radial Kernels*

Consider K to be a radial kernel $K(x,y) = \phi(\|x-y\|_2^2)$ with compact support in the unit ball of $\mathbf{R}^{d+1}$. Then the sparsity of the kernel matrix can be modified by the scale function c via

$$K_c(x,y) \quad = \quad \phi\left(\|x-y\|_2^2 + (c(x)-c(y))^2\right).$$

Since the interpolants take the form

$$s_{c,X,f}(x) := \sum_{j=1}^{N} a_j \phi\left(\|x_j-x\|_2^2 + (c(x_j)-c(x))^2\right),$$

the support radius can be influenced via c.

3. Theoretical Analysis

The map

$$C\ :\ x \mapsto (x,c(x)) \tag{3.1}$$

maps $\mathbf{R}^d$ into a d-dimensional submanifold $C(\mathbf{R}^d)$ of $\mathbf{R}^{d+1}$. A set $X = \{x_1,\dots,x_N\} \subset \Omega \subset \mathbf{R}^d$ of nodes goes into $C(X) \subset C(\Omega) \subset C(\mathbf{R}^d) \subset \mathbf{R}^{d+1}$. On $\mathbf{R}^{d+1}$ and the point set $C(X)$, interpolation by the unscaled kernel K takes place, and the resulting interpolant (2.2) takes the form

$$s_{c,X,f}(x) = s_{1,C(X),f}(x,c(x)) = s_{1,C(X),f}(C(x))$$

with the interpolant $s_{1,C(X),f}$ at scale 1 of the data of f in the points $(x_1,c(x_1)),\dots,(x_N,c(x_N))$ of $C(X)$. This means that in $\mathbf{R}^{d+1}$ the kernel (2.1) is used, and if we project points $(x,c(x)) \in \mathbf{R}^{d+1}$ back to $x \in \mathbf{R}^d$, the projection of the kernel K_c on $\mathbf{R}^{d+1}$ turns into a varying shape kernel on $\mathbf{R}^d$ whenever $c(x)$ is not constant.

Thus the analysis of error and stability of the varying–scale problem in $\mathbf{R}^d$ coincides with the analysis of a fixed–scale problem on a submanifold in $\mathbf{R}^{d+1}$. If C is assumed to be a diffeomorphism between Ω and $C(\Omega)$, and if Ω is compact, one can see $C(\Omega)$ as any other compact d–variate domain. The usual error and stability analysis focuses on *fill distance*

$$h(X,\Omega) := \sup_{y\in\Omega}\min_{x\in X}\|x-y\|_2$$

and *separation distance*

$$q(X) := \min_{X \ni x \neq y \in X} \|x-y\|_2$$

in their standard definitions. These will transform with C accordingly, and will roughly be multiplied by a factor that scales with the norm of the gradient of c or the Lipschitz constant L of c. In particular,

$$\begin{array}{rcl} \|C(x)-C(y)\|_2^2 & = & \|x-y\|_2^2+(c(x)-c(y))^2 \\ & \leqslant & \|x-y\|_2^2(1+L)^2 \\ \|C(x)-C(y)\|_2^2 & \geqslant & \|x-y\|_2^2 \end{array}$$

shows that distances will blow up with C, letting separation distances never decrease, thus enhancing stability. But fill distances will also blow up, increasing the usual error bounds.

This argument shows that one can successfully use the varying–scale (VSK) technique on points x_j and x_k that have

$$\|x_j - x_k\|_2 = q(X) << h(X,\Omega).$$

The scales should then vary like

$$|c(x_j)-c(x_k)| \approx h(X,\Omega) >> q(X),$$

and this should be done for all close–by pairs of centers until one roughly gets that

$$q(C(X)) \approx h(X,\Omega) \approx h(C(X),C(\Omega))$$

holds, i.e. the transformed centers are approximately uniformly distributed.

All of this is a special case of the following general principle. Given a kernel $K\ :\ \Omega\times\Omega\to\mathbf{R}$ and a bijective map

$$C\ :\ \Omega \mapsto C(\Omega),$$

the kernel

$$K_C(C(x),C(y)) := K(x,y) \text{ for all } x,y\in\Omega$$

now acts on $C(\Omega)$ and inherits the definiteness properties of K. This will also work "backwards", i.e. for a bijective map onto Ω.

This gives rise to two native spaces and their properties, i.e.

$$\begin{array}{rcl} f(x) & = & (f,K(x,\cdot)) \\ K(x,y) & = & (K(x,\cdot),K(y,\cdot)) \end{array}$$

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

$$\begin{array}{rcl} g(u) & = & (g,K_C(u,\cdot))_C \\ K_C(u,v) & = & (K_C(u,\cdot),K_C(v,\cdot))_C \end{array}$$

for all $u,v\in C(\Omega)$.

Theorem 2 The native spaces for K and K_C are isometric.

Proof: Functions f on Ω will map to functions $\mathscr{C}(f)$ on $C(\Omega)$ via $\mathscr{C}(f)(Cx) := f(x)$, and this map $\mathscr{C}$ is linear. Furthermore, $\mathscr{C}(K(\cdot,x))(C(y)) = K(x,y) = K_C(C(x),C(y))$ and thus also $\mathscr{C}(K(\cdot,x))(\cdot) = K_C(C(x),\cdot)$ hold. Thus we get

$$\begin{array}{rcl} K(x,y) & = & (K(x,\cdot),K(y,\cdot)) \\ & = & (\mathscr{C}(K(\cdot,x))(C(\cdot)),\mathscr{C}(K(\cdot,y))(C(\cdot))) \\ & = & K_C(C(x),C(y)) \\ & = & (K_C(C(x),\cdot),K_C(C(y),\cdot))_C \end{array}$$

for the two inner products, and this means

$$(f,g) = (\mathscr{C}(f)(C(\cdot)),\mathscr{C}(g)(C(\cdot))) = (\mathscr{C}(f),\mathscr{C}(g))_C$$

for all f,g in the span of all $K(\cdot,x)$, which are mapped to all $\mathscr{C}(f)$, $\mathscr{C}(g)$ in the span of all $\mathscr{C}(K(\cdot,x)) = K_C(C(x),\cdot)$. This proves that $\mathscr{C}$ is an isometry between the two native spaces. □

If we apply this to our special situation, we see that the native space for K_c on Ω is isometrically isomorphic to the native space for K_1 on $C(\Omega)$ for the C in (3.1).

4. Numerical Examples

We now provide some examples that show the different roles of the variable scale parameter: it may affect both the stability and the accuracy. In 4.1 we will show how its appropriate choice enhances stability, while the rest of this section demonstrates that one can significantly improve the recovery quality, in particular by preserving shape properties in a much better way than for interpolation with constant scale.

4.1 *Stability*

4.1.1 *Chebyshev Points* . For univariate functions on $\Omega = [-1,+1]$, interpolation by polynomials should be done in Chebyshev points $x_j = -\cos(\pi(j-1)/(N-1))$, $1 \leqslant j \leqslant N$ rather than in equidistant points. But the fill distance of these points behaves like $1/N$, while the separation distance behaves like $1/N^2$. If kernel–based methods are used, this leads to a very large condition in the kernel matrices, no matter which kernel is chosen. To cope with this, we map the interval $\Omega = [-1,+1] \subset \mathbf{R}$ to the semi–circle $C(\Omega) \subset \mathbf{R}^2$ via $C(x) = (x,\sqrt{1-x^2})$. Then the resulting points are equidistant, and we can work with a single–scale kernel in $\mathbf{R}^2$ for interpolation in $C(X)$. The separation distance will now behave like $1/N$ like the fill distance.

As a specific case, we chose the Gaussian kernel at fixed scale $0.1\cdot\sqrt{2}$ and took 55 Chebyshev points to work with. For comparison, we also used 55 equidistant points. The L_∞ errors for interpolating the Runge function $f(x) = 1/(1+25x^2)$ are given in Table 1(`tabRunge`). If there is no noise, the single scale methods are still superior. A possible explanation is that linear systems with badly conditioned kernel matrices and interpolation data from functions in the "native" Hilbert space have good approximate solutions based on a selection of columns, thus avoiding the influence of small separation distances. If additive random noise of maximal value ±0.001 is added, the above argument fails and the bad condition spoils the results seriously.

Results for other kernels and point sets were similar. If there is no noise and the data are from a smooth function, single scale methods often work fine even if the condition is beyond the limit. The MATLAB

Points and scaling	Condition	no noise	0.001 noise
equidistant, single scale	$5 \cdot 10^{14}$	$7 \cdot 10^{-5}$	0.0331
Chebyshev, single scale	$1 \cdot 10^{16}$	$1.1 \cdot 10^{-5}$	1.4294
Chebyshev, variable scale	$8 \cdot 10^{5}$	$1.3 \cdot 10^{-4}$	0.0012

Table 1. Interpolation of Runge function by Gaussians

backslash operator provides good approximate solutions in many cases. But as soon as noise and a bad condition of the single scale problem occur together, the variable scale tends to save the situation. In contrast to to polynomial interpolation, the equidistant case often is not seriously inferior to using Chebyshev nodes.

4.1.2 *Monotonic Node Transformation in 1D.* For scattered nodes $-1 \leqslant x_1 < \ldots < x_N \leqslant 1$ in 1D one can find a transformation C that maps them into points $z_j \in \mathbf{R}^2$ which are equidistant. If h is the fill distance of the given points, one can guarantee

$$\|z_{j+1} - z_j\|_2^2 = (x_{j+1} - x_j)^2 + (c_{j+1} - c_j)^2 = 4h^2$$

by choosing

$$c_{j+1} = c_j + \sqrt{4h^2 - (x_{j+1} - x_j)^2},$$

starting with $c_1 = 0$. This gives a monotonic sequence, and one can use piecewise linear interpolation to construct a continuous map C with $C(x_j) = z_j,\ 1 \leqslant j \leqslant N$ of the form $C(x) = (x, c(x))$ with c being monotonic.

Experiments with this strategy show a similar behavior like the previous example, but the non–differentiability of the map C spoils convergence rates. For a setting with 55 scattered points in $[-1, +1]$, Figure 1(`figC`) shows the monotonic function c that increases sharply where there are close–by points, while it is flat over holes in the node set. Table 2(`tabC`) shows the corresponding conditions and errors, again for the Runge function. Readers should remember that the variable scale interpolation is a fixed scale interpolation along the curve in Figure 1(`figC`). The maximal errors are attained in all cases near the gap in the data near $x = 0.1$. The example shows that the current method (VSK) is more stable and robust.

Points and scaling	Condition	no noise	0.001 noise
scattered, single scale	$2.6 \cdot 10^{16}$	$1.3 \cdot 10^{-5}$	0.0344
scattered, variable scale	$3.8 \cdot 10^{8}$	0.0126	0.0125

Table 2. Interpolation of Runge function by Gaussians, scattered nodes

4.1.3 *Cluster of data.* We take $N = 47$ points $x_i \in [-1, 1]$ so that 41 nodes are equispaced in the interval and 6 close to 0.4, with mutual distance $q = 10^{-4}$. As $c(x)$, we consider the *skew-Gaussian* [17] depicted in Fig. 2

$$\begin{array}{rcl} t(x) & := & 2 \cdot (x - 4.0005 \cdot 10^{-1})/(3.5 \cdot 10^{-4}) \\ c(x) & := & \exp(-(t^2(x) + 5 \cdot (1/\pi \arctan(t(x) + 1) + 0.5))) \end{array} \tag{4.1}$$

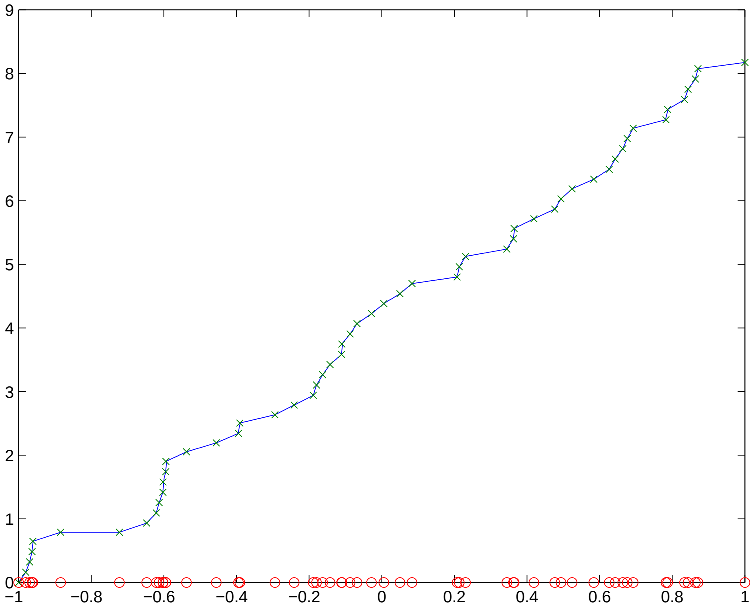

FIG. 1. Monotonic function c for $C(x) = (x, c(x))$

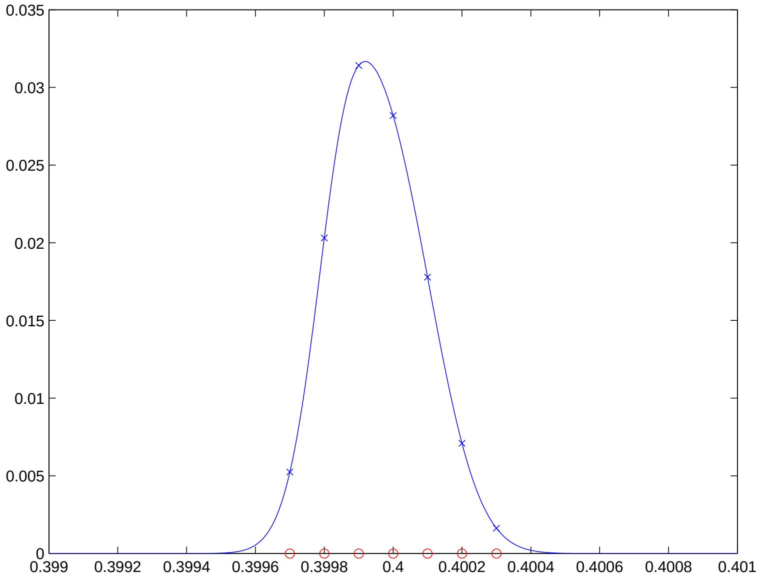

FIG. 2. $C(x) = (x, c(x))$ with some $c(x_i)$ values

In Table 3, we show the condition numbers and the L_∞ errors obtained interpolating the Runge function by the Gaussian kernel at fixed scale $0.1 \cdot \sqrt{2}$ and by the proposed technique. In Fig. 3 the plots of the absolute error for the two interpolants are depicted.

Points and scaling	Condition	error	0.001 noise
cluster, single scale	$3.5 \cdot 10^{16}$	$6.02 \cdot 10^{-5}$	$5.4 \cdot 10^{-1}$
cluster, variable scale	$6.9 \cdot 10^{10}$	$9.4 \cdot 10^{-6}$	$3.0 \cdot 10^{-3}$

Table 3. Interpolation of Runge function by Gaussians, cluster nodes

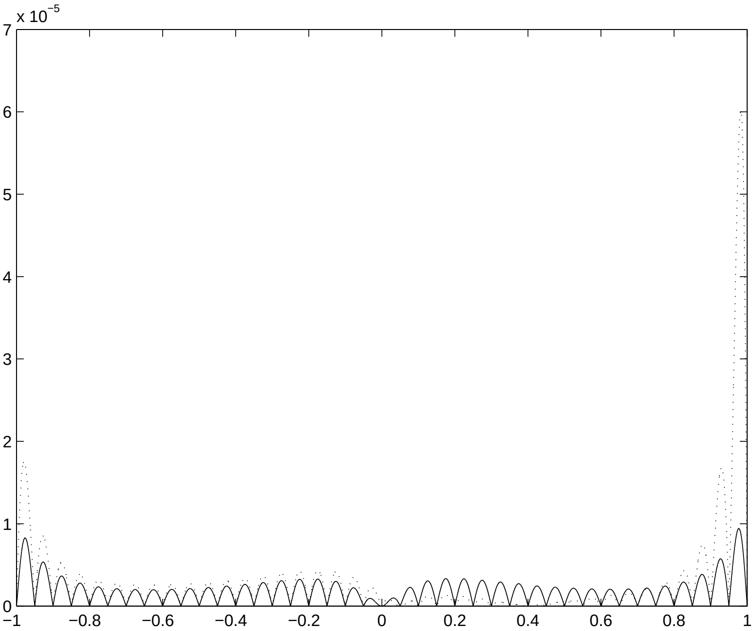

FIG. 3. No noise case; absolute error for the classic case dotted; absolute error for the VSK-interpolant in full line

4.2 *Enhancing reproduction quality*

In this section, we deal with the problem of obtaining interpolants which reproduce faithfully the underlying functions. (see e.g. [3], [7]). In the following examples, we compare the classical interpolant provided by the C^2 Wendland kernel with support radius 1 with the VSK interpolation provided by the d-variate C^2 Wendland kernel with support radius $\mu(C(\Omega))^{1/d}$, where μ is the length or area of $C(\Omega)$.

4.2.1 *The logistic function.* We consider the logistic function

$$f(x) = (1 + 2 \cdot \exp(p(x)))^{-0.5}, \tag{4.2}$$

where $p(x) = -3 \cdot (10\sqrt{2x^2} - 6.7)$. We take $N = 11$ nodes with a carefully chosen distribution in the interval $\Omega = [0,1]$. In particular, we want a higher density where the function changes more quickly and a lower density where it is nearly constant. In this case we chose as variable scale $c(x)$ the multiquadric interpolant s_{MQ} of the given data (with $\delta = 0.03$) multiplied by a tension coefficient. For this example,

$$c(x) = 2 \cdot s_{MQ}(x), \tag{4.3}$$

and its plot is shown in Fig. 4, together with the values $c(x_i)$ marked.

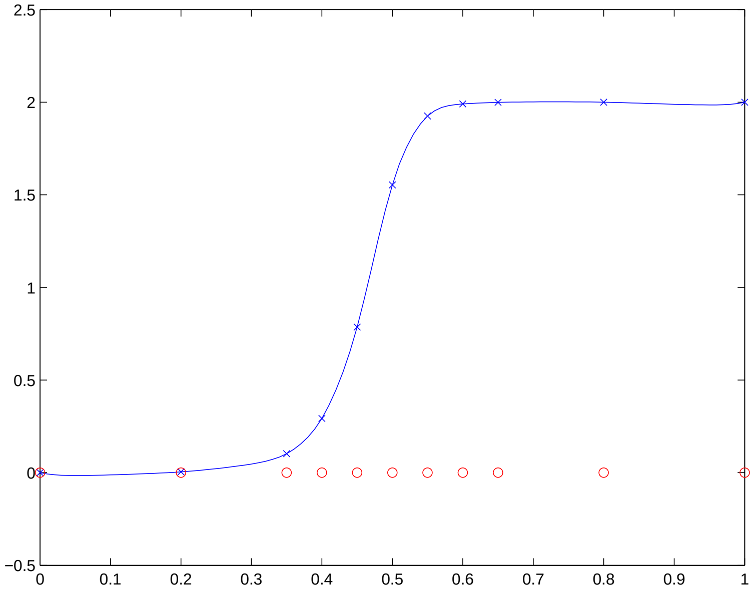

FIG. 4. $C(x) = (x, c(x))$

Note that the classical interpolant (Fig. 5) has near the right boundary an oscillation not consistent with the underlying function shape, while the VSK interpolant (Fig. 6) is a faithful recovery of (4.2). The absolute interpolation errors for the two interpolants are shown in Fig. 7.

The L_∞ errors are $2.5 \cdot 10^{-2}$ for the fixed–scale case and $6.4 \cdot 10^{-3}$ for the variable–scale case.

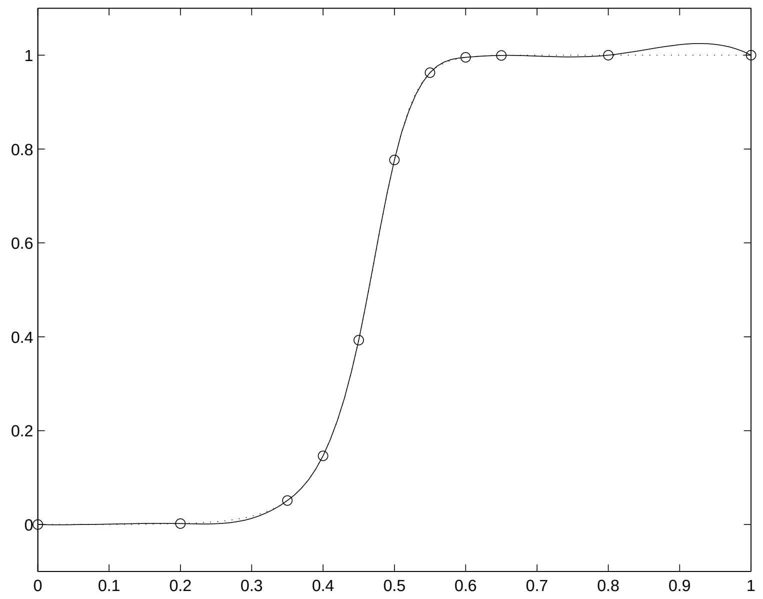

FIG. 5. Classical Wendland interpolant full lined; logistic function dotted

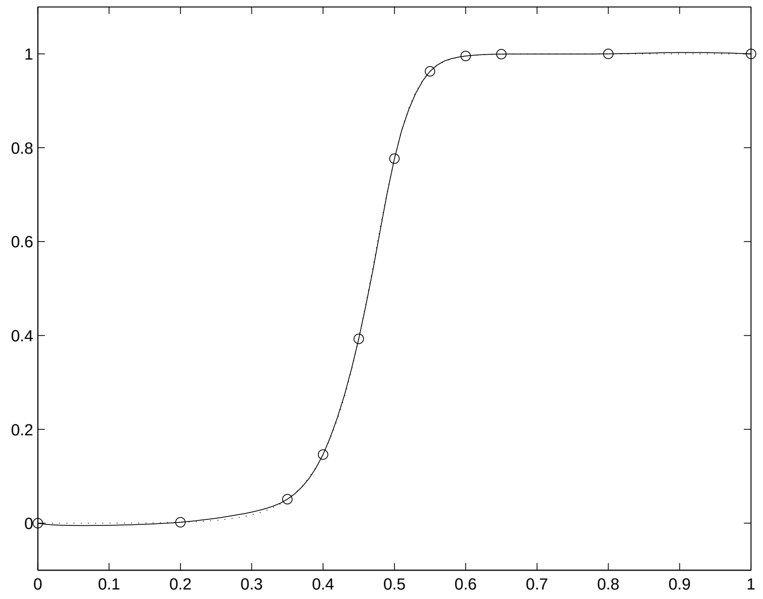

FIG. 6. VSK-interpolant full lined; logistic function dotted

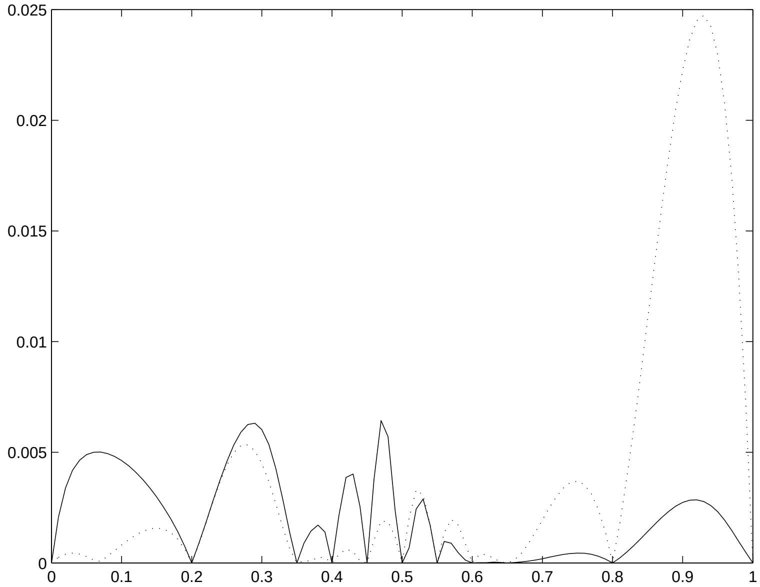

FIG. 7. Absolute error for the classic interpolant dotted; absolute error for the VSK interpolant full lined

4.2.2 *The sigmoid function.* We consider the function

$$f(x,y) = (1+2\cdot\exp(-3(9\sqrt{x^2+y^2}-6.7)))^{-0.5} \quad (x,y)\in\Omega=[0,1]^2, \tag{4.4}$$

and a configuration of $N = 89$ nodes in Ω (see Fig. 8) with higher density where the function changes most. As in the previous example, we consider the variable scale (see Fig. 9, with marks on the transformed interpolation points)

$$c(x,y) = 2.5\, s_{MQ}(x,y), \tag{4.5}$$

where s_{MQ} is the multiquadric interpolant to the given data with $\delta = 0.15$.

The classical interpolant and the VSK-interpolant are shown respectively in Figs. 10, 11 and the interpolation errors in Figs. 12, 13.

The L_∞ errors are $1.3\cdot 10^{-1}$ for the fixed–scale case and $2.5\cdot 10^{-2}$ for the variable–scale case.

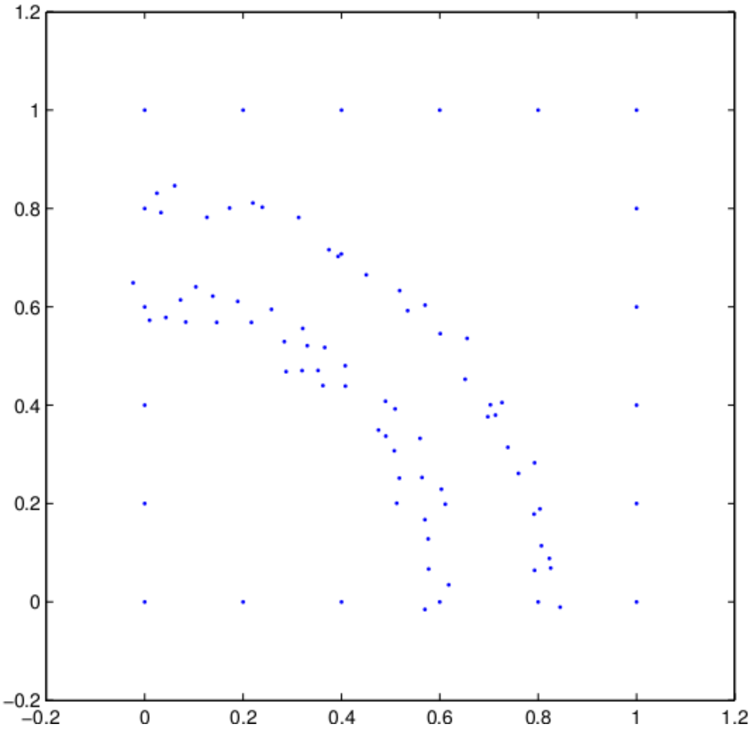

FIG. 8. Given nodes

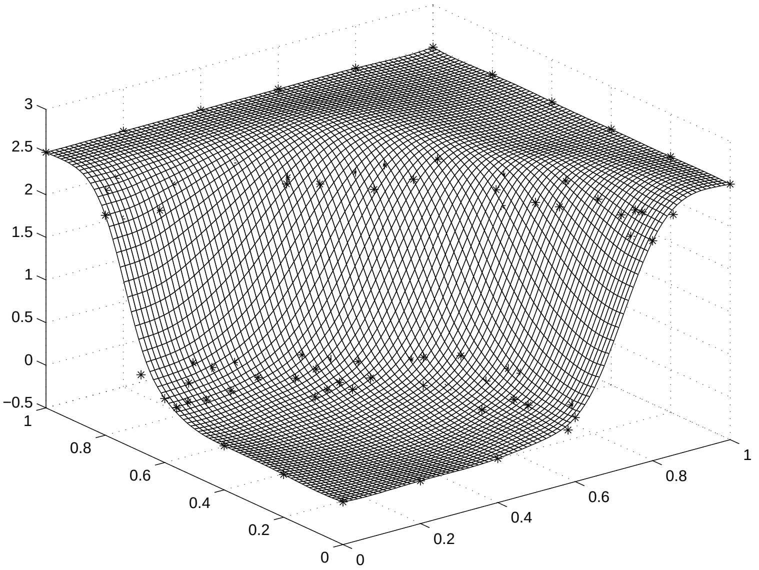

FIG. 9. $C(x,y) = (x,y,c(x,y))$

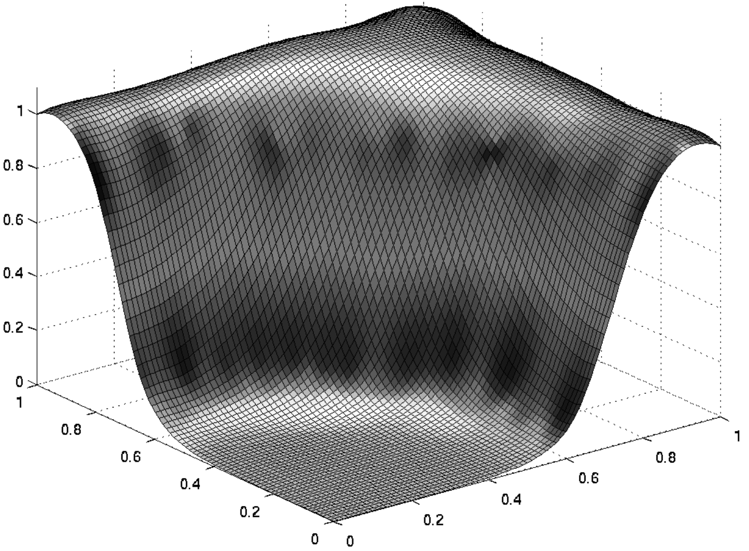

FIG. 10. Classical interpolation

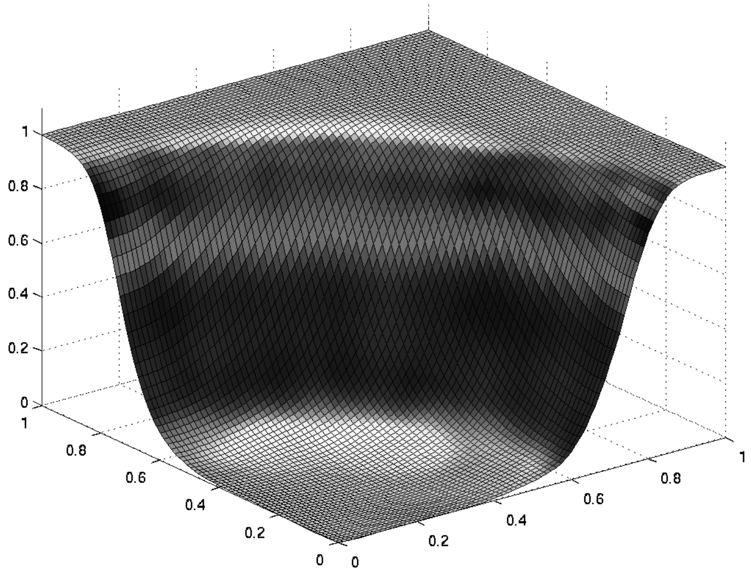

FIG. 11. $s_{c,X,f}(x)$

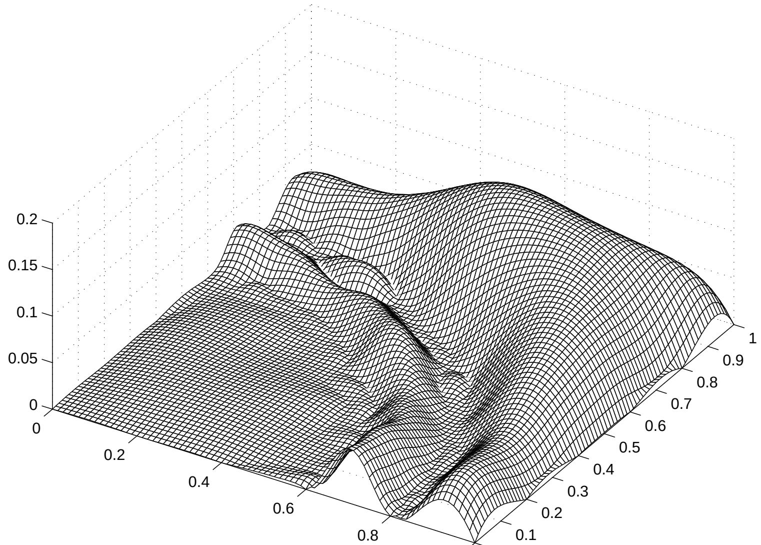

FIG. 12. Absolute interpolation error; classical case

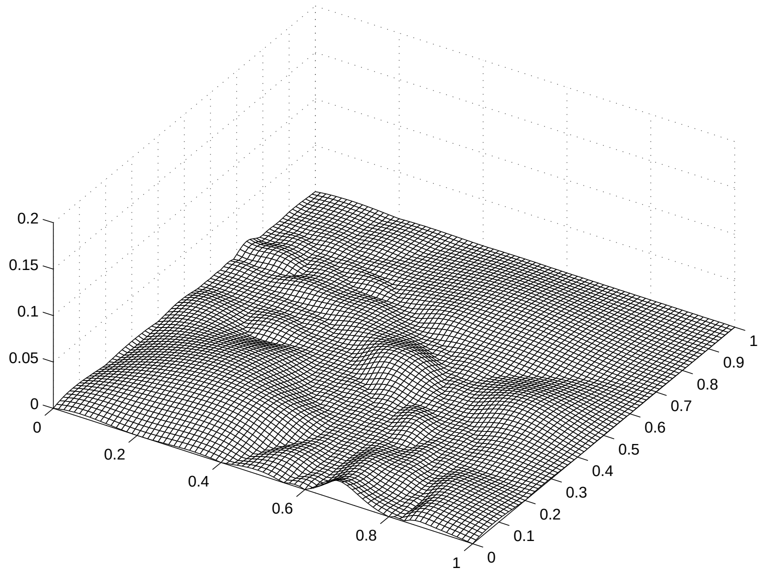

FIG. 13. Absolute error for the VSK-interpolant

4.3 *Interpolant with optimal node configuration*

In [8], the authors remarked that good node configurations X are those with

$$q(X) \approx h(X, \Omega).$$

So our aim is to find a variable scale that transforms the given node set $X = \{x_i\}_{i=1}^N$ into a set $C(X)$ satisfying

$$q(C(X)) \approx h(C(X), C(\Omega)) \approx h(X, \Omega).$$

Let us consider the well–known humps and dips function by Franke [14] and $N = 131$ nodes belonging to a domain $\Omega_1 \supset \Omega = [0,1]^2$. The nodes are more dense in the zones of the maxima and of the minimum (see Fig. 14). We define the variable scale (see Fig. 15) as

$$c(x,y) = \sum_{i=1}^{N} p_i(x,y), \tag{4.6}$$

where the functions

$$p_i(x,y) := (1/\pi \cdot \arctan(a_i(x-x_i)) + 1/2) \cdot exp(-5 \cdot (y-y_i)^2)$$

increase more steeply at x_i with a larger value of a_i, if the local density of the data locations is large at x_i. This choice allows us to increase the separation distance q where the node density is high without increasing too much the fill distance h, achieving or goal this way. In Fig. 16 we show the classical interpolant performed by using a Gaussian kernel with $\delta = 0.25$ and in in Fig. 17 the VSK-interpolant. The corresponding interpolation errors are depicted in Figs. 18 and 19. The L_∞ errors are $2.1 \cdot 10^{-1}$ for the fixed-scale case and $4.2 \cdot 10^{-2}$ for the variable–scale case.

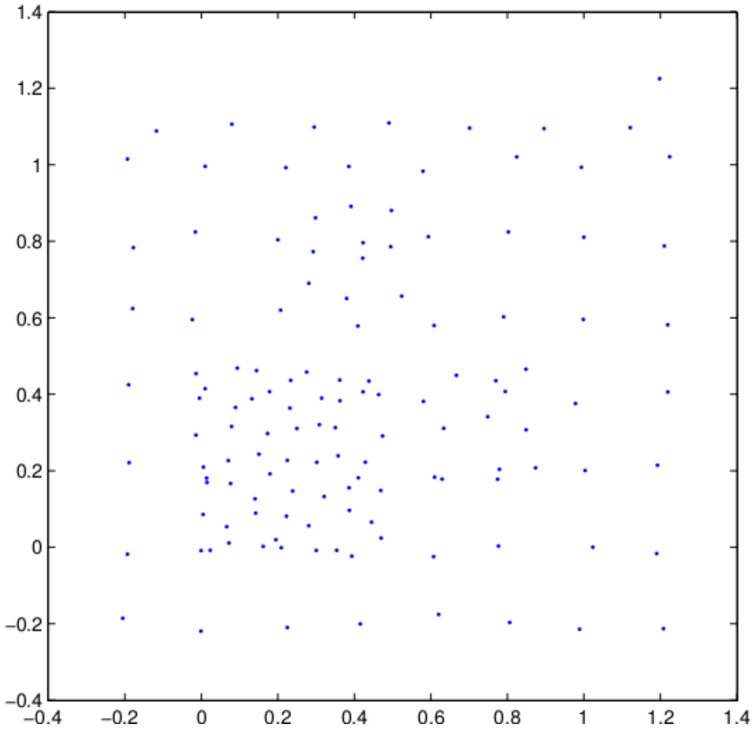

FIG. 14. Given nodes

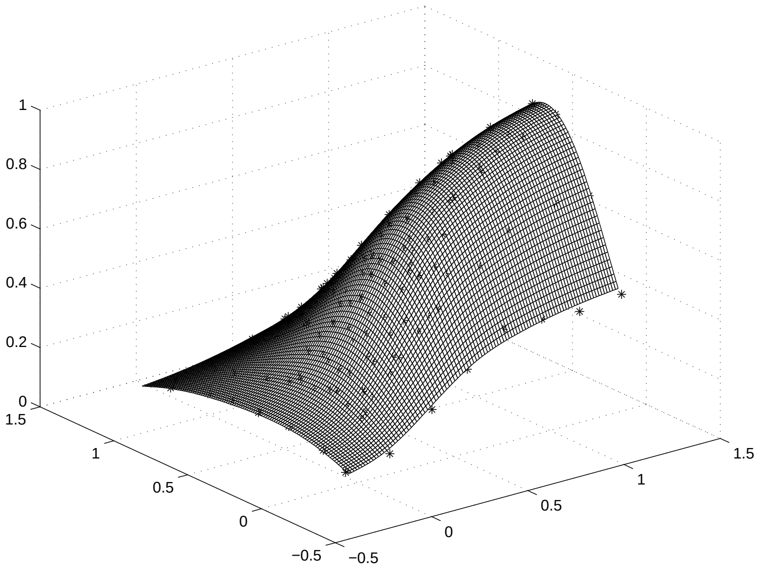

FIG. 15. $C(x,y) = (x, y, c(x,y))$

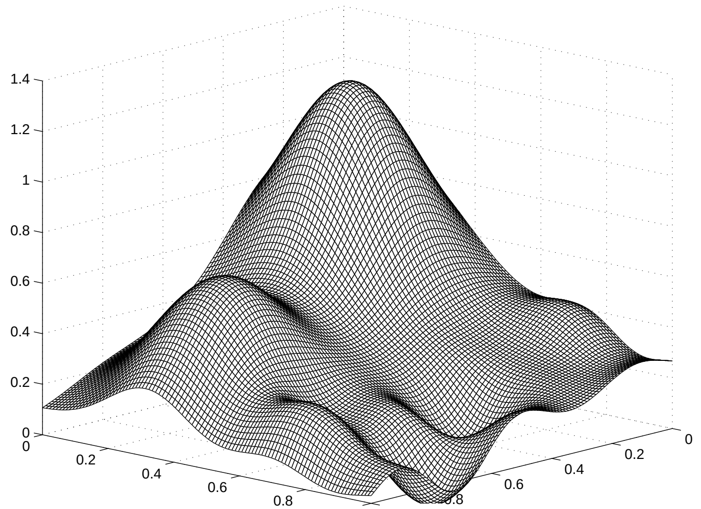

FIG. 16. Classical interpolant

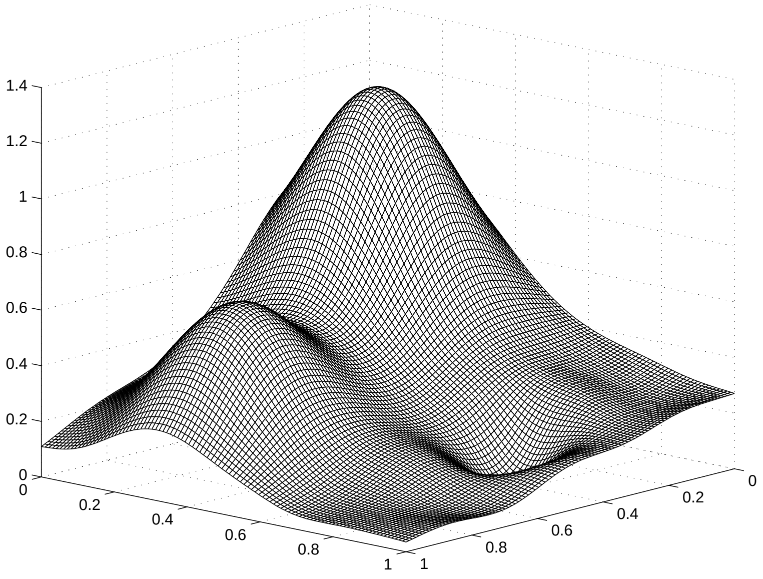

FIG. 17. VSK-interpolant

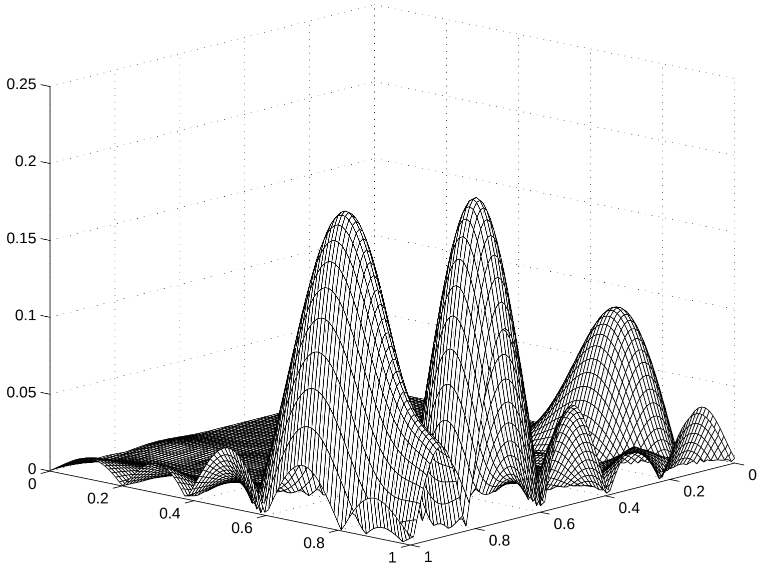

FIG. 18. Absolute error for the classical interpolant

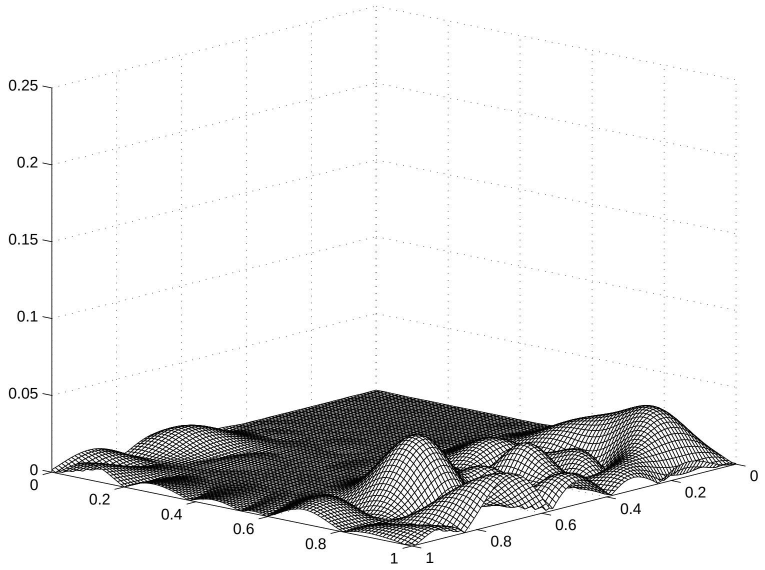

FIG. 19. Absolute error for the VSK-interpolant

5. Conclusions

The examples show how the proper use of the variable scale kernel K_C can lead to a more stable and better shape–preserving interpolant. In particular when the variable scale function c is chosen to depend on critical shape properties of the data, the interpolant reproduces the underlying phenomenon in a much more faithful way, and the interpolation error can be reduced in the critical regions to show a much more uniform behaviour in the whole domain.

REFERENCES

Aronszajn N. (1950) Theory of reproducing kernels. *Trans. Amer. Math. Soc.*, **68**, 337–404.

Bozzini M., Lenarduzzi L. (2005) Reconstruction of surfaces from a not large data set by interpolation. *Rendiconti di Matematica, Serie VII*, **25**, 223–239.

Bozzini M., Lenarduzzi L. (2003) Bivariate knot selection procedure and interpolation perturbed in scale and shape. *Curve and Surface Design*, T. Lyche, M. L. Mazure and L. Schumaker eds, 1–10.

Bozzini M., Lenarduzzi L., Rossini M., Schaback R. (2004) Interpolation by basis functions of different scales and shapes. *Calcolo*, **41** (2), 77–87.

Bozzini M., Lenarduzzi L., Schaback R. (2002) Adaptive interpolation by scaled multiquadrics. *Advances in Computational Mathematics*, **16** 375–387.

Buhmann M. D. (2003) *Radial Basis Functions, Theory and Implementations*. Cambridge University Press.

Casciola G., Lazzaro D., Montefusco L. B., Morigi S.(2006) Shape preserving reconstruction using locally anisotropic radial basis function interpolants. *Computer and Mathematics with Applications*, **1**, 1185–1198.

De Marchi S., Schaback R., Wendland H. (2005) Near-optimal data-independent point locations for radial basis function interpolation. *Adv. Comput. Math.*, **23**(3), 317–330.

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

Fasshauer G., Zhang J.G. (2007) On choosing optimal shape parameters for rbf approximation. *Numerical Algorithms*, **45**, 345–368.

Fasshauer G. (2007) *Meshfree Approximation Methods with MATLAB*, volume 6 of *Interdisciplinary Mathematical Sciences*. World Scientific Publishers, Singapore.

Fornberg B., Wright G., Larsson L. (2004) Some observations regarding interpolants in the limit of flat radial basis functions. *Computers & Mathematics with Applications*, **47**, 37–55.

Fornberg B., Zuev J. (2007) The Runge phenomenon and spatially variable shape parameters in RBF interpolation. *Comput. Math. Appl.*, **54** (3), 379–398.

www.math.nps.navy.mil/ rfranke.

Kansa E. J., Carlson R.E. (1992) Improved accuracy of multiquadric interpolation using variable shape parameters. *Comput. Math. Appl.*, **24**, 99–120.

Iske A. (2004) *Multiresolution Methods in Scattered Data Modelling*, volume 37 of *Lecture Notes in Computational Science and Engineering*. Springer.

Jamshidi A. A., Kirby M. J. (2009) Skew-radial basis function expansions for empirical modeling. *SIAM J. Sci. Comp.* **31**, 4715–4743.

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

Lee Y.J., Yoon G.J., Yoon J. (2007) Convergence property of increasingly flat radial basis function interpolation to polynomial interpolation. *SIAM J. Mathematical Analysis*, **39**, 470-488.

Rippa S. (1999) An algorithm for selecting a good value for the parameter c in radial basis function interpolation. *Adv. Comput. Math.*, **11** (2-3), 193–210.

Schaback R. (2005) Multivariate interpolation by polynomials and radial basis functions. *Constructive Approximation*, **21**, 293–317.

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

Schaback R., Wendland H. (2006) Kernel techniques: from machine learning to meshless methods. *Acta Numerica*, **15**, 543–639.

Wendland H. (1995) Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree. *Advances in Computational Mathematics*, **4**, 389–396.

Wendland H. (2005) *Scattered Data Approximation*. Cambridge University Press.