

Fast computation of orthonormal basis for RBF spaces through Krylov space methods

Stefano De Marchi · Gabriele Santin

Received: 7 January 2014 / Accepted: 18 November 2014 / Published online: 9 December 2014
© Springer Science+Business Media Dordrecht 2014

Abstract In recent years, in the setting of radial basis function, the study of approximation algorithms has particularly focused on the construction of (stable) bases for the associated Hilbert spaces. One of the ways of describing such spaces and their properties is the study of a particular integral operator and its spectrum. We proposed in a recent work the so-called WSVD basis, which is strictly connected to the eigen-decomposition of this operator and allows to overcome some problems related to the stability of the computation of the approximant for a wide class of radial kernels. Although effective, this basis is computationally expensive to compute. In this paper we discuss a method to improve and compute in a fast way the basis using methods related to Krylov subspaces. After reviewing the connections between the two bases, we concentrate on the properties of the new one, describing its behavior by numerical tests.

Keywords Interpolation · Positive definite functions · Matrix factorization · Fast computation

Mathematics Subject Classification 41A05 · 42A82 · 15A23 · 65Y20

Communicated by Michiel Hochstenbach.

S. De Marchi · G. Santin (✉)
Department of Mathematics, University of Padova, via Trieste 63, Padova, Italy
e-mail: gsantin@math.unipd.it

S. De Marchi
e-mail: demarchi@math.unipd.it

1 Introduction

In what follows K is a strictly positive definite and radial kernel defined on a set $\Omega \subset \mathbb{R}^s$, and X is a finite set of $n \in \mathbb{N}$ distinct points in Ω . Since the kernel is radial, there exist a unique function $\phi : [0, +\infty) \rightarrow \mathbb{R}$ and a real number $\varepsilon > 0$ such that $K(x, y) = K_\varepsilon(x, y) = \phi(\varepsilon\|x - y\|) \forall x, y \in \Omega$, with $\|\cdot\|$ the usual Euclidean norm. The number ε is called the *shape parameter* and plays an important role in the approximation process.

We denote by $\mathcal{N}_K(\Omega)$ the *native space* of K on Ω , while $\mathcal{N}_K(X)$ is the subspace of $\mathcal{N}_K(\Omega)$ spanned by $\{K(\cdot, x), x \in X\}$. The unique interpolant in $\mathcal{N}_K(X)$ of a function $f \in \mathcal{N}_K(\Omega)$ on X will be denoted by $s_X(f)$.

It can be observed that the so-called *direct method* to compute the interpolant can be really unstable in most cases, and this observation lead to a wide literature of methods which discuss strategies to overcome the problem through the optimization of various parameters involved in the approximation process (cf. e.g. [9, 11]). Nevertheless, it has been proved in [4] that the *interpolation operator* itself is stable in the function space $\mathcal{N}_K(\Omega)$. This result suggests that the focus should be on the design of different algorithms to compute in stable way the same interpolant.

In this view, one of the main research topic in the present study of RBF approximation is the construction of different *bases* of the same space $\mathcal{N}_K(\Omega)$, which avoids the use of the ill-conditioned basis of translates of the kernel K on the data set X .

The construction of such bases and the corresponding algorithms used to approximate functions, are known as *stable algorithms*. This kind of approach has been introduced in [11] for the particular case of multiquadric kernels, and then extended to a wider family of kernels on the sphere in [10] and to different domains (see [6, 8, 9]) for the Gaussian kernel. All these algorithms allow to compute an effective interpolant also in the case of the flat limit $\varepsilon \rightarrow 0$. This confirms that the problem is intrinsically well posed and that the instability is a numerical issue. In fact, except for the previous cases, the direct method is still the most used approach for RBF approximation for general Ω and kernels. In all these cases, our algorithm reduces significantly the instability of the problem.

Another approach for the construction of a better basis has been introduced in [16] (extending an idea already explored in [14] for the so-called Newton basis). Namely, the authors provided a way to produce and characterize different kinds of orthonormal bases, i.e. sets $\mathcal{U} = \{u_j\}_{1 \leq j \leq n} \subset \mathcal{N}_K(X)$ such that $\text{span}\{u_j, 1 \leq j \leq n\} = \mathcal{N}_K(X)$ and $(u_i, u_j) = \delta_{ij}$ for $1 \leq i, j \leq n$. This kind of construction can be applied in the same way to any RBF kernel on any domain Ω and characterizes all the possible orthonormal bases which span the same space $\mathcal{N}_K(\Omega)$.

In that paper has been remarked that the key tool to understand the behavior of these different bases is the *eigenbasis* associated to a particular integral operator acting on the space $\mathcal{N}_K(\Omega)$.

Using this idea, in [3] we built a particular orthonormal basis, that we called *weighted singular value decomposition basis* (shortly WSVD basis), which provides a meaningful discrete approximation of the eigenbasis. In the setting of [16], the WSVD basis allows to study the different eigenbases regardless to the particular kernel or

domain. Moreover, the WSVD basis allows to interpolate functions in $\mathcal{N}_K(\Omega)$ in a more stable way.

Since the main issue of the WSVD basis was the computational cost, here we present an efficient method to compute a (slightly) modified version of it, which turns out to enjoy some further interesting properties. The method is based on Krylov spaces techniques, which have already been proved to be an effective tool in connection with RBF approximation (see e.g. [2, 7]).

The paper is organized as follows. In Sect. 2 we recall the fundamental facts about the WSVD basis together with a couple of new properties that we did not notice before. In Sect. 3 we present the new results and in the last Section we discuss some numerical experiments which show the good behavior and effectiveness of the new basis.

2 WSVD bases

In [3], we have introduced the so-called WSVD basis. The basis has been constructed by following the ideas already outlined in [16], and intended as a discrete approximation of a particular basis of the native space $\mathcal{N}_K(\Omega)$. To be more precise, it is well-known from the Mercer’s theorem (see e.g. [17, §10.4] for a discussion of this construction) that the compact and self adjoint operator

$$\begin{aligned}
 T_K &: \mathcal{N}_K(\Omega) \rightarrow \mathcal{N}_K(\Omega) \\
 T_K[f](\cdot) &= \int_{\Omega} f(y)K(\cdot, y)dy,
 \end{aligned}
 \tag{2.1}$$

has a countable set of eigenvectors $\{\varphi_j\}_{j>0}$ which form a basis of $\mathcal{N}_K(\Omega)$, that is the *eigenbasis*. The basis can be chosen to be $\mathcal{N}_K(\Omega)$ -orthonormal and $L_2(\Omega)$ -orthogonal, with $\|\varphi_j\|^2 = \lambda_j$, $\{\lambda_j\}_{j>0}$ being the set of eigenvalues of T_K , and $\lambda_j \rightarrow 0$ as $j \rightarrow \infty$.

Starting with this setting we considered a cubature formula with the set of nodes being X itself, and we approximated the eigenvectors of T_K by the symmetric Nyström method (see e.g. [1, §11.4]).

It turned out that this basis can be described in terms of the notation introduced in [16]. We recall that each basis for $\mathcal{N}_K(X)$ can be parametrized by a matrix of change of basis, say $C_{\mathcal{U}} = (c_{ij})_{1 \leq i, j \leq n}$, such that $u_j = \sum_{i=1}^n c_{ij}K(\cdot, x_i)$, $1 \leq j \leq n$, or otherwise by the matrix of its values at X , say $V_{\mathcal{U}} = (u_j(x_i))_{1 \leq i, j \leq n}$. In both cases $V_{\mathcal{U}} \cdot C_{\mathcal{U}}^{-1} = A$, where $A = (K(x_i, x_j))_{1 \leq i, j \leq n}$ is the so-called kernel matrix.

In view of this, we defined the WSVD basis as follows.

Definition 2.1 A weighted SVD basis \mathcal{U} is a basis for $\mathcal{N}_K(X)$ characterized by the matrix of change of basis

$$C_{\mathcal{U}} = \sqrt{W} \cdot Q \cdot \Sigma^{-1}$$

and the matrix $V_{\mathcal{U}} = (u_j(x_i))_{1 \leq i, j \leq n}$ at X given by

$$V_{\mathcal{U}} = \sqrt{W^{-1}} \cdot Q \cdot \Sigma,$$

where

$$\underbrace{\sqrt{W} \cdot A \cdot \sqrt{W}}_{A_W} = Q \cdot \Sigma^2 \cdot Q^T$$

is a singular value decomposition (and a unitary diagonalization) of the scaled kernel matrix A_W . To be more precise, Σ is a diagonal matrix with $\Sigma_{jj} = \sigma_j$, $j = 1, \dots, n$ and $\sigma_1^2 \geq \dots \geq \sigma_n^2 > 0$ are the singular values of A_W , and W is a diagonal matrix where $W_{jj} = w_j$, $j = 1, \dots, n$ are the positive weights of the cubature rule $(X, \mathcal{W})_n$.

Remark 2.1 While our main interest here is the solution of standard interpolation problems, it is worth mentioning that the same approach can be used when dealing with different approximation problems, namely in the case of generalized approximation. In this latter setting the data are provided by the evaluation of some linear functionals $L_i \in \mathcal{N}_K(\Omega)^*$, $1 \leq i \leq n$ and, under some assumptions on the linear independence of the functionals, the problem can be solved as a standard interpolation problem using the basis $\{L_i K(\cdot, x), 1 \leq i \leq n\}$ instead of $\{K(\cdot, x_i), 1 \leq i \leq n\}$ (see e.g. [17, Ch.16]).

2.1 Properties

The basis so defined has some interesting properties. First notice that, given a cubature rule $(X, \mathcal{W})_n$, we can define the $\ell_{2,w}(X)$ inner product as

$$(f, g)_{\ell_{2,w}(X)} = \sum_{i=1}^n w_i f(x_i)g(x_i), \quad f, g \in \mathcal{N}_K(\Omega), \tag{2.2}$$

which is a discrete version of $(\cdot, \cdot)_{L_2(\Omega)}$. Hence, the operator

$$\begin{aligned} T_n : \mathcal{N}_K(\Omega) &\rightarrow \mathcal{N}_K(X) \\ T_n[f](x) &= (f, K(\cdot, x))_{\ell_{2,w}(X)} \end{aligned} \tag{2.3}$$

which is the discrete version of T_K , is defined just replacing the $L_2(\Omega)$ product with its discrete version $\ell_{2,w}(X)$ in (2.1). In fact the operator is a compact (its range is $\mathcal{N}_K(X)$) and self adjoint, as can be seen by direct computation. It has n eigenvectors, which turns out to be exactly the basis $\{u_j\}_{j=1}^n$. The functions u_j are essentially a discrete version of the $\{\varphi_j\}_{j>0}$, as summarized in the following theorem (whose proof is in [3]).

Theorem 2.2 *Every weighted SVD basis \mathcal{U} satisfy the following properties:*

- (a) $T_n[u_j] = \sigma_j u_j$, $1 \leq j \leq n$;
- (b) \mathcal{U} is $\mathcal{N}_K(\Omega)$ -orthonormal;
- (c) \mathcal{U} is $\ell_{2,w}(X)$ -orthogonal;
- (d) $\|u_j\|_{\ell_{2,w}(X)}^2 = \sigma_j^2 \quad \forall u_j \in \mathcal{U}$;
- (e) $\sum_{j=1}^n \sigma_j^2 = \phi(0) |\Omega|$.

For a general set of weights we simply require that they are non negative and sum up to the measure $|\Omega|$. It is also possible to use a set of weights which does not provide a cubature rule, but in this way we loose the connection with the eigenbasis.

Using the orthonormality property of the basis, the interpolant can then be written as

$$s_X(f) = \sum_{j=1}^n (f, u_j) u_j \quad \forall f \in \mathcal{N}_K(\Omega) \tag{2.4}$$

and the corresponding power function $\mathcal{P}_{K,X}(x)$, which is the operator norm of the pointwise error functional $\mathcal{E}_x : \mathcal{N}_K(\Omega) \rightarrow \mathbb{R}$, $\mathcal{E}_x(f) = f(x) - s_X(f)(x)$, has the simple representation

$$\mathcal{P}_{K,X}(x) = \left(\phi(0) - \sum_{j=1}^n u_j(x)^2 \right)^{1/2} \quad \forall x \in \Omega. \tag{2.5}$$

This basis allows to go a step further and to consider another kind of approximation. Indeed, as a consequence of the property (a), we have that $(f, u_j)_{\ell_{2,w}(X)} = \sigma_j^2(f, u_j) \quad \forall f \in \mathcal{N}_K(\Omega)$ and $\forall j = 1, \dots, n$.

Thus for each $f \in \mathcal{N}_K(\Omega)$ we can define a discrete least-squares approximant $s_X^m(f)$ as the minimizer of $\|f - g\|_{\ell_{2,w}(X)}$ among all functions $g \in span\{u_1, \dots, u_m\}$, and thanks to the latter property we have that $s_X^m(f)$ is obtained from the interpolant $s_X(f)$ simply truncating the sum to the first m terms, i.e.

$$s_X^m(f)(x) = \sum_{j=1}^m \sigma_j^{-2}(f, u_j)_{\ell_{2,w}(X)} u_j(x) = \sum_{j=1}^m (f, u_j) u_j(x). \tag{2.6}$$

This kind of approximation corresponds to the use of a low rank approximation of the kernel matrix A , and provides a way to overcome the instability of the problem, while solving just a little sub problem of the original one. This approach make sense since in the singular values $\{\sigma_j^2\}_{1 \leq j \leq n}$ of A accumulate to zero very fast, being in particular a discrete approximation of the eigenvalues of the compact operator (2.1). Of course the rate of decay of the singular values, and hence the effectiveness of the method, strictly depends on the particular problem. Namely, it is well-known that the decay is faster if the kernel is smoother and if the shape parameter is smaller, so in these cases we can expect a better stabilization of the approximation process. An in-deep study of the influence of the shape parameter ε on the decay of the singular values of the matrix A can be found in [12, Sect. 5].

Hence, the truncated basis provides a good approximation and allows to deal with a smaller problem.

Remark 2.2 One may argue that the same result can be obtained by choosing a proper subset of samples, $X_m \subset X_n$, $m < n$. The question is “how to select such a subset?” This is in general an hard task. A known approach is by means of greedy algorithms

that recursively select a good interpolation set of samples, such as the ones described in [5, 16]. These algorithms are based on similar ideas, that is the extraction of a good sub-basis starting from the complete data set. We have already compared them with our method in [3].

Remark 2.3 When dealing with generalized interpolation problems (see Remark 2.1) this truncation still works, but we need to take into account the action of the truncation in the cases when the problem involves functionals of different kind. A typical example is the solution of a PDE with boundary conditions, where we have to consider two different sets of functionals: $\{L_i\}_{i=1}^{n_1} \in \mathcal{N}_K(\Omega)^*$ in the interior and $\{B_i\}_{i=1}^{n_2} \in \mathcal{N}_K(\Omega)^*$ on the boundary. In this case our method extracts a reduced basis of the subspace $\text{span}\{L_i K(\cdot, x), 1 \leq i \leq n_1\} \cup \text{span}\{B_i K(\cdot, x), 1 \leq i \leq n_2\} \subset \mathcal{N}_K(\Omega)$, not two bases of the subspaces spanned by the different sets of functionals.

The fast computation of the most significant part of the basis, the set $\{u_1, \dots, u_m\}$ for certain $m < n$, is the main topic of the next sections.

2.2 Stability and convergence

Before moving to the computational aspects, we stress here some results about the stability and the convergence of this approximation.

Concerning the convergence we can define the analogous of the power function, say $\mathcal{P}_{K,X}^m(x)$, so that

$$|f(x) - s_X^m(f)(x)| \leq \mathcal{P}_{K,X}^m(x) \|f\|, \quad \forall x \in \Omega, \tag{2.7}$$

where, thanks to (2.6), we have the explicit formula

$$\mathcal{P}_{K,X}^m(x)^2 = \phi(0) - \sum_{j=1}^m u_j(x)^2, \quad \forall x \in \Omega. \tag{2.8}$$

We may expect that, for a proper choice of $m < n$, the terms $\|u_j^2(x)\|_{L_\infty(\Omega)}$ are small for $j > m$. We did not prove this fact yet, while it has clear numerical evidence as we observed by many numerical tests.

To measure the stability we use two different estimates, which correspond to two different norms used to measure the norm of $f \in \mathcal{N}_K(\Omega)$. If it is viewed as a function of the native space, we can consider the bound (see [3, 16])

$$|s_X^m(f)(x)| \leq \left(\sum_{j=1}^m u_j(x)^2 \right)^{\frac{1}{2}} \|f\| \leq \sqrt{\phi(0)} \|f\| \quad \forall x \in \Omega. \tag{2.9}$$

On the other hand, we can look at the sequence $f|_X$ and provide bounds in terms of $\|f\|_{\ell_\infty(X)}$. To do this, we define a *pseudo Lagrange basis* as follows.

Definition 2.3 Let $\{l_i\}_{i=1}^n$ be the unique Lagrange (or cardinal) basis of $\mathcal{N}_K(X)$. The pseudo Lagrange basis $\{\tilde{l}_i\}_{i=1}^n$ of order m is given by $\tilde{l}_i = s_X^m(l_i)$, $1 \leq i \leq n$.

For $m = n$ we obviously have $\tilde{l}_i = l_i$, $1 \leq i \leq n$.

Now we can give another bound on the stability of $s_X^m(f)$, based on the ‘‘pseudo-Lebesgue constant’’, which is equivalent to the usual one when using l_i instead of \tilde{l}_i .

Theorem 2.4 Let $\{\tilde{l}_i\}_{i=1}^n$ be the pseudo Lagrange basis associated to $\{u_j\}_{j=1}^m$. Thus, for $1 \leq i \leq n$,

$$\tilde{l}_i = w_i \sum_{j=1}^m \sigma_j^{-2} u_j(x_i) u_j. \tag{2.10}$$

Furthermore, $\forall f \in \Omega$ the following bound holds:

$$\|s_X^m(f)\|_{L_\infty(\Omega)}, \leq \tilde{\Lambda}_X \|f\|_{\ell_\infty(X)}, \tilde{\Lambda}_X = \max_{x \in \Omega} \sum_{i=1}^n |\tilde{l}_i(x)| \tag{2.11}$$

where $\tilde{\Lambda}_X$ is the pseudo-Lebesgue constant.

Proof From Definition 2.3 we have, for $1 \leq i \leq n$,

$$\tilde{l}_i = s_X^m(l_i) = \sum_{j=1}^m \sigma_j^{-2} (l_i, u_j)_{\ell_{2,w}(X)} u_j = w_i \sum_{j=1}^m \sigma_j^{-2} u_j(x_i) u_j,$$

which proves the first statement.

To prove the second one it suffices to rewrite the inner product in $s_X^m(f)$ and get the approximant in cardinal form

$$s_X^m(f) = \sum_{j=1}^m \frac{(f, u_j)_{\ell_{2,w}(X)}}{\sigma_j^2} u_j = \sum_{i=1}^n f(x_i) w_i \sum_{j=1}^m \sigma_j^{-2} u_j(x_i) u_j = \sum_{i=1}^n f(x_i) \tilde{l}_i$$

which gives the desired bound in terms of $\tilde{\Lambda}_X$. □

3 The new basis

As seen in the previous section, we can compute an orthonormal basis \mathcal{U} of $\mathcal{N}_K(X)$. Its main property is that it allows to extract a sub basis $\{u_1, \dots, u_m\}$, with $m < n$, which gives an approximant $s_X^m(f)$ as good as the interpolant, while being much more stable.

The main problem is the efficiency of this computation. Since the basis is found by a singular value decomposition, we have to compute all the elements u_j , $1 \leq j \leq n$,

and then to leave out the last $n - m$ terms, the ones for which $\sigma_j^2 < \tau$, with τ a prescribed tolerance.

Here we present a way to slightly modify our basis in order to compute its most significant part. Although the connection between the two bases will be discussed in details, we stress here that the new basis has some characteristic features coming from its dependence on the data. Moreover, since we want to use our basis with more general data sets X than those associated to a cubature rule, we will omit the dependence on the weights $\{w_i\}_{1 \leq i \leq n}$, which is equivalent to take all weights $w_i = 1/n, \forall i$.

The proposed method makes use of some tools from the theory of Krylov subspaces, recalled in the next subsection for readers convenience.

3.1 The Lanczos method and the approximation of the SVD

Our goal is finding the solution of the linear system $Ax = b$ with A the kernel matrix and $b = (f(x_i))_{1 \leq i \leq n}$.

To this aim, let $\mathcal{K}_m(A, b) = \text{span}\{b, Ab, \dots, A^{m-1}b\}$ be the Krylov subspace of order m generated by A and b . The Lanczos method computes an orthonormal basis $\{p_1, \dots, p_m\}$ of $\mathcal{K}_m(A, b)$ through a Gram-Schmidt orthonormalization. If we denote by $P_m \in \mathbb{R}^{n \times m}$ the matrix having the vectors p_i as columns, we can write the algorithm in matrix form as

$$AP_m = P_{m+1}\bar{H}_m, \quad \bar{H}_m = \begin{bmatrix} H_m \\ \bar{h}e_m^T \end{bmatrix}, \tag{3.1}$$

where H_m is a $m \times m$ tridiagonal matrix (because of the symmetry of the kernel), $\bar{h} \in \mathbb{R}$ and $e_m \in \mathbb{R}^m$ is the m -th unit vector.

Once these matrices have been computed, the solution x can be approximated as $x = P_m y$, where $y \in \mathbb{R}^m$ is such that $\bar{H}_m y = \|b\|_2 e_1$.

If A has a good low-rank approximation, which is our case, we expect that a good approximation of x can be computed using m components with $m \ll n$.

This decomposition can be used for other purposes. Indeed, in the recent paper [15] the authors studied the relation between the singular values of A and those of \bar{H}_m . In particular, $\bar{H}_m = U_m \bar{\Sigma}_m V_m^T$ is a singular value decomposition of \bar{H}_m , where $U_m \in \mathbb{R}^{(m+1) \times (m+1)}$ and $V_m \in \mathbb{R}^{m \times m}$ are unitary matrices and

$$\bar{\Sigma}_m = \begin{bmatrix} \Sigma_m^2 \\ 0 \end{bmatrix},$$

Σ_m^2 being the diagonal matrix having the singular values as its entries. The authors then analyzed the relationship between the SVD decomposition of A and the decomposition $(P_{m+1}U_m, \bar{\Sigma}_m, P_m V_m)$. Although the paper shows a close connection between the two decompositions, together with other results, for our purposes we use the second which is indeed a good approximation of the SVD and coincide when $m = n$.

Remark 3.1 Since the last row of $\bar{\Sigma}_m$ is the zero vector, the decomposition does not change if we remove this row and the last column of U_m . To simplify our notation,

from now onward we will denote by U_m the matrix without the last column, so that the decomposition becomes $\bar{H}_m = U_m \Sigma_m^2 V_m^T$. Moreover, the j -th diagonal element of Σ_m^2 will be denoted by σ_j^2 instead of $\sigma_{j,m}^2$ when no confusion arises.

In view of this result, our idea is using the SVD of \bar{H}_m for constructing the new basis.

3.2 Construction of the basis

Using the above observations, we can construct a new set of functions $\{\tilde{u}_j\}_{1 \leq j \leq m} \in \mathcal{N}_K(X)$ similarly to the construction of the WSVD basis. The set will be orthogonal with respect to the $\ell_{2,w}(X)$ -inner product and near-orthonormal in $\mathcal{N}_K(\Omega)$, in a way that we will specify later. Moreover, this set can be computed iteratively, i.e. we can choose to compute just the first m elements, the ones we are interested in, except for the case when the complete WSVD basis is needed.

We stress that the set $\{\tilde{u}_j\}_{1 \leq j \leq m}$ does not span $\mathcal{N}_K(X)$ (unlike $m = n$, when the SVD of \bar{H}_m equals to that of A). This says that the term ‘‘basis’’ is not the most appropriate to identify these functions, even if we will use it (with an abuse of language) with the aim to avoid a more complicated statement.

Using the notation in [16], the basis can be defined as follows.

Definition 3.1 Let A be the kernel matrix for the set X of n distinct points. Let $\bar{H}_m, P_m, V_m, U_m, \Sigma_m^2$ be as introduced above. Hence, the basis $\bar{\mathcal{U}}_m = \{\bar{u}_1, \dots, \bar{u}_m\}$ is characterized by the matrix of change of basis

$$C_{\bar{\mathcal{U}}_m} = P_m V_m \Sigma_m^{-1} \tag{3.2}$$

or by the collocation matrix

$$V_{\bar{\mathcal{U}}_m} = P_{m+1} U_m \Sigma_m. \tag{3.3}$$

In this case the basis strongly depends on the particular function $\bar{f} \in \mathcal{N}_K(\Omega)$ used to construct the Krylov subspace. This dependence influences the behavior of the approximant, as it will be more clear from the following properties of the basis.

3.3 Properties

We start by proving this Lemma.

Lemma 3.2 Let \tilde{U}_m be the square matrix obtained from U_m removing the last row u_m^T . Then \tilde{U}_m and V_m coincide except for the last row, namely only the m -th row d_m^T of the difference is a non zero row vector.

Proof Using the SVD of \bar{H}_m , it is immediate to see that

$$\left[\begin{array}{c|c} H_m^2 & \bar{h}H_me_m \\ \hline \bar{h}e_m^T H_m & \bar{h}^2 \end{array} \right] = \bar{H}_m \bar{H}_m^T = \left[\begin{array}{c|c} \tilde{U}_m \Sigma_m^4 \tilde{U}_m^T & \tilde{U}_m \Sigma_m^4 u \\ \hline u^T \Sigma_m^4 \tilde{U}_m & u^T \Sigma_m^4 u \end{array} \right],$$

hence $H_m^2 = \tilde{U}_m \Sigma_m^4 \tilde{U}_m^T$. On the other hand

$$H_m^2 + \bar{h}e_me_m^T = \bar{H}_m^T \bar{H}_m = V_m \Sigma_m^4 V_m.$$

From these two equalities easily follows

$$(V_m - \tilde{U}_m) \Sigma_m^4 (V_m - \tilde{U}_m)^T = \bar{h}e_me_m^T.$$

We can conclude that each row of the difference matrix, $D_m := V_m - \tilde{U}_m$, is the zero vector except for the last one, say d_m^T , which satisfies $d_m^T \Sigma_m^4 d_m = \bar{h}$. \square

The new constructed basis enjoys the following properties summarized in the next Theorem.

Theorem 3.3 *Let the basis $\bar{\mathcal{U}}_m$ be defined as in the Definition 3.1. Then*

- (i) *the basis is $\ell_{2,w}(X)$ -orthogonal with $\|\bar{u}_j\|_{\ell_{2,w}(X)}^2 = \sigma_j^2$;*
- (ii) *the basis is near-orthonormal on $\mathcal{N}_K(\Omega)$, meaning that $(\bar{u}_i, \bar{u}_j) = \delta_{ij} + r_{ij}^{(m)}$ where $(R^{(m)})_{ij} := r_{ij}^{(m)}$ is a rank one matrix for $1 \leq m < n$, and $r_{ij}^{(m)} = 0$ when $m = n$;*
- (iii) *when $m = n$, $\bar{\mathcal{U}}_m = \mathcal{U}$.*

Proof It suffices to compute the gramian matrix of the basis with respect to the $\ell_{2,w}(X)$ and $\mathcal{N}_K(\Omega)$ inner products, that is $\Gamma_{\bar{\mathcal{U}}_m}$ and $G_{\bar{\mathcal{U}}_m}$, respectively. Using formulas in [16] and the Definition 3.1 we get

$$\Gamma_{\bar{\mathcal{U}}_m} = (V_{\bar{\mathcal{U}}_m}^T) V_{\bar{\mathcal{U}}_m} = \Sigma_m U_m^T P_{m+1}^T P_{m+1} U_m \Sigma_m = \Sigma_m^2,$$

and

$$\begin{aligned} G_{\bar{\mathcal{U}}_m} &= (C_{\bar{\mathcal{U}}_m}^T) V_{\bar{\mathcal{U}}_m} = \Sigma_m^{-1} V_m^T P_m^T P_{m+1} U_m \Sigma_m \\ &= \Sigma_m^{-1} V_m^T [I_m \mid 0] U_m \Sigma_m = \Sigma_m^{-1} V_m^T \tilde{U}_m \Sigma_m. \end{aligned}$$

From the above Lemma 3.2, $\tilde{U}_m = V_m + e_m d_m^T$, hence if v_m^T is the last row of V_m ,

$$V_m^T \tilde{U}_m = V_m^T (V_m + e_m d_m^T) = I_m + v_m d_m^T.$$

This allows to conclude that

$$G_{\tilde{\mathcal{U}}_m} = \Sigma_m^{-1} \left(I_m + v_m d_m^T \right) \Sigma_m = I_m + R_m$$

with $R_m = \Sigma_m^{-1} v_m d_m^T \Sigma_m$.

The last statement easily follows from the fact that the SVD of A is exactly computed by the Lanczos method when $m = n$ and from Definition 3.1. \square

Note that $r_{ij} = (R_m)_{ij}$ is in general non vanishing if $m < n$. For $m = n$ it vanishes for all $1 \leq i, j \leq m$ since $\bar{h} = 0$. It is also worth noticing that from all the numerical experiments we have done, the magnitude of r_{ij} is close to the machine precision except if both i and j are close to m . That is to say that the first elements of the basis are orthonormal from a numerical point of view (see next Sect. 4).

3.4 Approximation

It comes now easy to compute the approximant obtained by solving the linear system. If we take the function $\bar{f} \in \mathcal{N}_K(\Omega)$ from which the basis is constructed, we get the approximant as a projection with respect to the $\ell_{2,w}(X)$ inner product

$$s_X^m(\bar{f})(x) = \sum_{j=1}^m \sigma_j^{-2} (\bar{f}, \bar{u}_j)_{\ell_{2,w}(X)} \bar{u}_j(x) \quad \forall x \in \Omega. \tag{3.4}$$

This follows straightforwardly from the fact that $\Sigma_m^{-2} V_{\tilde{\mathcal{U}}_m}^T$ is a left inverse of $V_{\tilde{\mathcal{U}}_m}$. What is a bit surprising is that it can be expressed in terms of the $\mathcal{N}_K(\Omega)$ inner product as in (2.6).

Theorem 3.4 *If the basis $\tilde{\mathcal{U}}_m$ is constructed from $\bar{f} \in \mathcal{N}_K(\Omega)$, for all $j = 1, \dots, m$ we have*

$$(\bar{f}, \bar{u}_j)_{\ell_{2,w}(X)} = \sigma_j^2 (\bar{f}, \bar{u}_j). \tag{3.5}$$

Hence the approximant $s_X^m(\bar{f})$ is a projection of $\mathcal{N}_K(\Omega)$, or equivalently

$$s_X^m(\bar{f})(x) = \sum_{j=1}^m (\bar{f}, \bar{u}_j) \bar{u}_j(x) \quad \forall x \in \Omega. \tag{3.6}$$

Proof Observe that the coefficient $(f, \bar{u}_j)_{\ell_{2,w}(X)}$, for a generic $f \in \mathcal{N}_K(\Omega)$, is the j -th column of $f_X^T V_{\tilde{\mathcal{U}}_m}$ (f_X is the column vector of the evaluations of f at X). Using again the Lemma 3.2 and denoting by u_m^T the last row of U_m , we get

$$\begin{aligned} V_{\tilde{\mathcal{U}}_m} &= P_m \tilde{U}_m \Sigma_m + p_{m+1} u_m^T \Sigma_m = P_m V_m \Sigma_m + p_m d_m^T \Sigma_m + p_{m+1} u_m^T \Sigma_m \\ &= C_{\tilde{\mathcal{U}}_m} \Sigma_m^2 + p_m d_m^T \Sigma_m + p_{m+1} u_m^T \Sigma_m. \end{aligned}$$

If we take $f = \tilde{f}$, since \tilde{f}_X is orthogonal to the vectors $\{p_2, \dots, p_{m+1}\}$, we have $f_X^T V_{\tilde{\varphi}_m} = f_X^T C_{\tilde{\varphi}_m} \Sigma_m^2$, i.e. the equality (3.5) holds. This implies in particular that the formula (3.6) is equivalent to (3.4). □

Remark 3.2 If we take another function $f \in \mathcal{N}_K(\Omega)$, the equality (3.5) holds with a residual term on the right hand side. This is due to the fact that in the case $f \neq \tilde{f}$ the terms depending on p_m, p_{m+1} are not deleted.

On the other hand, the Eq. (3.4) depends only on the relation between the left inverse of $V_{\tilde{\varphi}_m}$ and its transpose, not on the connection with $C_{\tilde{\varphi}_m}$. This means that we can compute the approximant of a function $f \neq \tilde{f}$ also using the basis constructed starting from \tilde{f} . Although possible, this seems not a good idea, since the construction of the basis is fast and we can obviously expect a better convergence for a data-dependent basis.

To conclude, if the WSVD basis is replaced with the new one, the estimates (2.8) and 2.11 hold also in this case. It is important to notice that we do not have yet theoretical results about their behavior. In the next section we will show some numerical computations of the power function and the Lebesgue function.

4 Numerical experiments

In this section we show some examples in order to test the features of our approximation scheme.

Since we have already analyzed the connection between the standard basis of translates and the WSVD basis in [3], here we focus only on the new features of the enhanced basis. We simply observe that in all experiments we have obtained better approximation results.

We show two examples. In the first we approximate a function from the native space of a Gaussian kernel with the aim of understanding the behavior of the method in a known setting. In the second example we approximate real data by an inverse multiquadric kernel.

All experiments have been performed with Matlab 7.14.0.739 (R2012a) on a Intel Celeron Dual-Core CPU T3100, 1.90GHz, with 3 Gb of RAM.

4.1 A native space example

We consider the Gaussian kernel

$$K_1(x, y) = e^{-\varepsilon^2 \|x-y\|_2} \quad \forall x, y \in \Omega_1,$$

where Ω_1 is the unit circle in \mathbb{R}^2 , and the shape parameter is $\varepsilon = 1$.

The function f_1 we want to approximate is defined for all $x \in \mathbb{R}^2$ as

$$f_1(x) = K_1(x, p_1) + 2 K_1(x, p_2) - 2 K_1(x, p_3) + 3 K_1(x, p_4),$$

$$p_1 = (0, -1.2), p_2 = (-0.4, 0.5), p_3 = (-0.4, 1.1), p_4 = (1.2, 1.3),$$

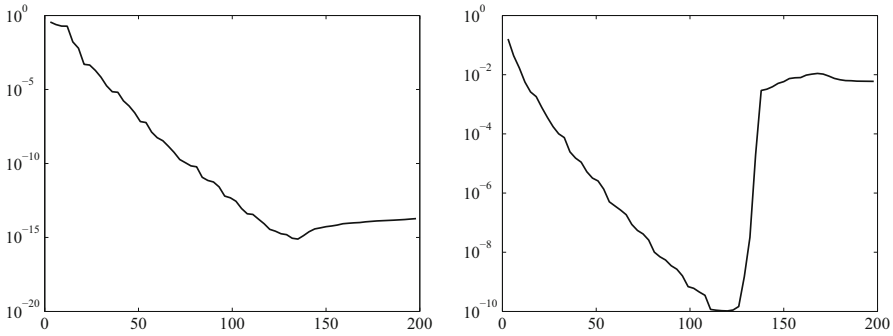


Fig. 1 Decay of the residual described in (4.1) compared with the corresponding RMSE for the example of Sect. 4.1

which is clearly a function in the native space $\mathcal{N}_{K_1}(\Omega_1)$. The set of samples X_n is chosen as the set of equally spaced points of $[-1, 1]^2$ restricted to Ω_1 , so that $\text{card}(X_n) = n$ with $n \in \mathbb{N}$.

The first problem is to choose how to stop the Lanczos iteration. For sure this is a key point of this approximation and only few sophisticated stopping criteria are known. On the other hand, we would like to use a stopping rule which tells us something on the approximation process from a functional point of view. A reasonably good choice for stopping the Lanczos iteration is when, for a certain tolerance $\tau > 0$, we have

$$\left| \frac{1}{n} \sum_{j=1}^m (H_m)_{jj} - 1 \right| < \tau, \tag{4.1}$$

which is motivated by the Property (e) of Theorem 2.2. This is a rough criterion, but it seems good enough to control the iterations in the case of functions lying in the native space of the kernel. This behavior is shown in Figure 1 in the case of $\tau = 1 \times 10^{-15}$.

In Fig. 2 we show the plots of the basis elements $u_1, u_{11}, u_{21}, u_{31}$. Notice, as happens by increasing the index j for the WSVD basis, the number of oscillations of the sign of the basis increases, while its maximum value decreases.

To analyze the computational time required to solve the approximation problem, we compared it with that required by the WSVD basis. We chose a restricted set of X_n equally spaced points in Ω_1 , namely for $n = 1, \dots, 40^2$ (this restriction is due to the slowness of the computation of the WSVD basis). For each n we computed the optimal m using the new algorithm with tolerance $\tau = 1 \times 10^{-14}$. Then we used this value to select the corresponding number of WSVD basis elements. In Table 1 we display this comparison, only for some values of n , in order to show the performance of the method. It is worth to mention that the computational times of Table 1 are those required by the truncated SVD, which are, in this case, computed from a full SVD. The computational cost of the full and the truncated SVD are essentially the same. We can also see that for all choices of n the method takes less than 7 s to construct the approximant, which is a reasonable time for the architecture we used and the number of points considered. Moreover, despite the approximation introduced by the

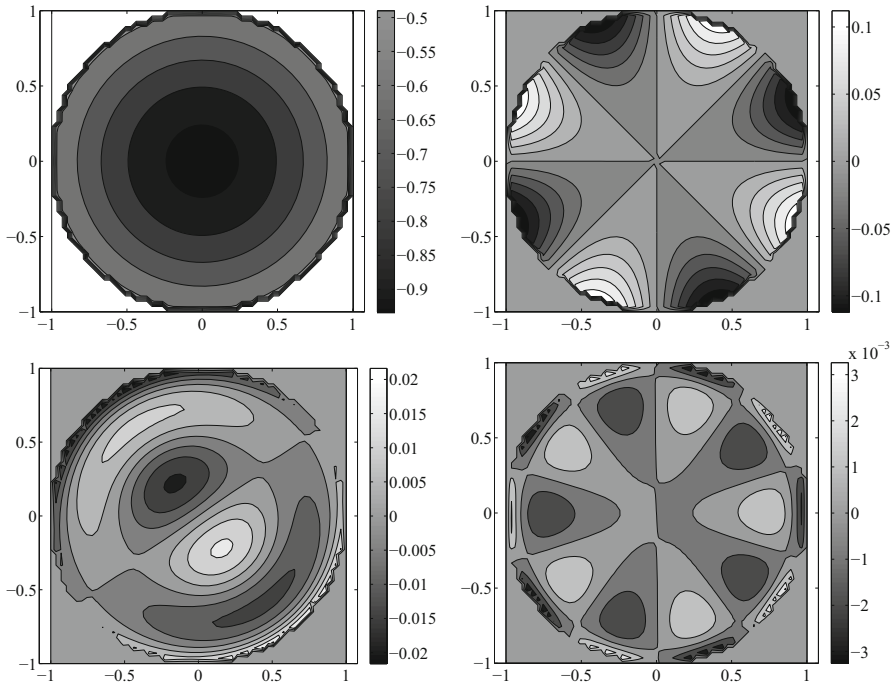


Fig. 2 Basis functions computed in the first example of Sect. 4.1. From *top left* to *bottom right* $u_1, u_{11}, u_{21}, u_{31}$

Table 1 Comparison of the WSVD basis and the new basis. Computational time in seconds and corresponding RMSE for the example in Sect. 4.1, restricted to $n = 15^2, 23^2, 31^2, 39^2$ equally spaced points

n	225	529	961	1521
m	110	114	115	116
RMSE new	3.4×10^{-10}	6.7×10^{-11}	5.5×10^{-11}	3.4×10^{-11}
RMSE WSVD	3.3×10^{-9}	1.1×10^{-9}	8.3×10^{-10}	7.9×10^{-10}
Time new	3.4×10^{-1}	1.0×10^0	2.6×10^0	6.5×10^0
Time WSVD	7.2×10^{-1}	4.2×10^0	2.5×10^1	1.1×10^2

new truncated basis, the computed errors are comparable or even better to the one of the original basis. This behavior is due to the function-dependent construction of the basis.

Using these observations we can construct the approximation on the full data set. In this case, choosing $\tau = 1 \times 10^{-14}$ we obtain $m = 115$ basis elements out of $3,600 = 60^2$ points with $RMSE = 1.03 \times 10^{-10}$. The overall computation takes 45 seconds, with the first 40 s used to construct the basis, and the remaining 5 s required to compute the approximant.

The approximated power function and the approximated generalized Lebesgue function (introduced in Sect. 2) are depicted in Fig. 3. It is clear that both the quantities

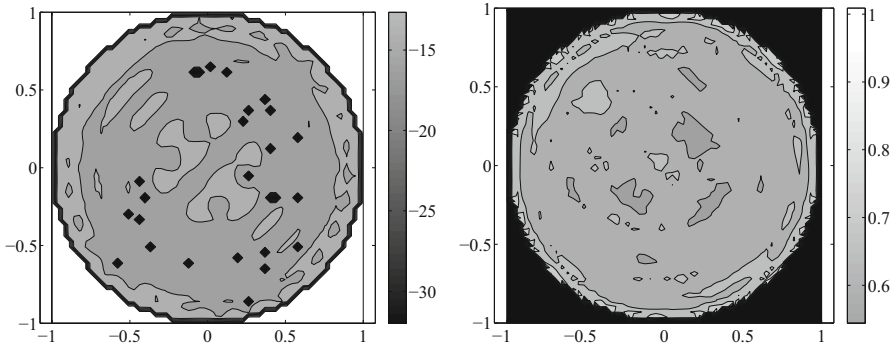


Fig. 3 Approximated power function (left, logarithmic plot) and approximated Lebesgue function (right) associated to the example of Sect. 4.1

Table 2 Number m of selected bases and the corresponding RMSE by decreasing ε

ε	2	2^{-2}	2^{-4}	2^{-6}	2^{-7}
RMSE	1.0×10^{-8}	2.1×10^{-6}	1.7×10^{-3}	6.5×10^{-2}	3.75×10^{-1}
m	249	64	26	16	12

These experiments refer to the Example of Sect. 4.1 for $n = 25^2$ equally spaced points

are quite uniform in the domain Ω_1 . The maximum value of the approximated power function on the grid is of order 10^{-13} , while the value of $\tilde{\Lambda}_X$ is 10.20.

To conclude this experiment, we show in Table 2 the effect of the shape parameter ε . In particular, we use the example of this section, with $n = 25^2$ equally spaced points, to show how the number m of selected bases, and thus the RMSE, changes as $\varepsilon \rightarrow 0$. It is clear that the number of selected bases decreases with ε , affecting also the corresponding RMS error.

4.2 A general example

In this case we take the inverse multiquadric kernel (IMQ)

$$K_2(x, y) = \left(\sqrt{1 + \varepsilon^2 \|x - y\|^2} \right)^{-1} \quad \forall x, y \in \Omega_2,$$

where $\varepsilon = 2$ and Ω_2 is a lune, namely, the set defined by the difference of two disks of radius 0.5 with centers in $(0, 0)$ and $(0.5, 0.5)$.

We tried to approximate the well-known Franke’s test function (see [13]), say f_2 , plotted in Fig. 4, top left. To make the test as general as possible, we used a set of randomly distributed data sites X_n with $n = 60^2$.

The results of the test are depicted in Fig. 4. The method used $m = 176$ basis elements selected among 60^2 points giving a $RMSE = 1.38 \times 10^{-6}$ and the pointwise error as in Fig. 4 (bottom right). The whole computation required one minute of CPU.

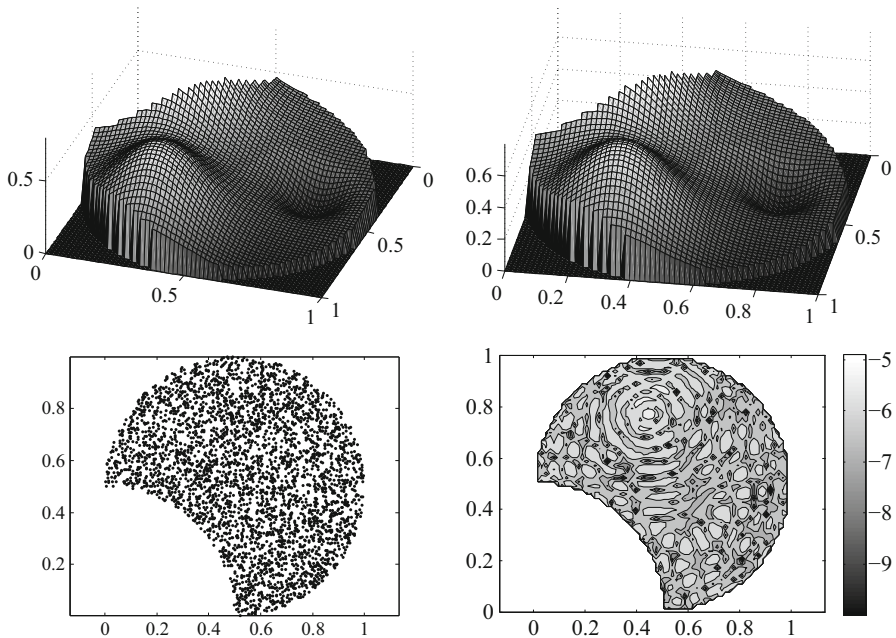


Fig. 4 Franke's Function (*top left*); approximation obtained with a IMQ kernel with $\varepsilon = 2$ (*top right*) using 60^2 randomly distributed data points (*bottom left*); pointwise error computed on a grid of 60×60 equally spaced points (*bottom right*)

It is important to remark that in this case the tolerance of the stopping rule have been set to $\tau = 1 \times 10^{-10}$, since a smaller value led to an increase of the RMS error. The failure of the stopping rule in this case is due to the fact that the function f_2 does not belong to the native space of the kernel.

5 Conclusions

The method here presented for computing a new approximation of the WSVD basis seems to be very effective both from the point of view of the computational time and the approximation properties. Actually, the reasons for the study of this new method were mainly motivated by computational issues: the computation of the original WSVD basis was too slow and costly.

The new computation of the basis is based on the Lanczos algorithm, which is used in order to provide an approximation of the truncated SVD of the kernel matrix A . This introduces an error in the construction, but, thanks to the fact that the new basis is built using information coming from the sampling of the unknown function, we are now able to obtain a better approximation.

Both this new approach and the one proposed in [3] for the WSVD basis, aim to reduce the instability of the direct method for the computation of RBF approximants. Several stable algorithms are known in literature for solving this task in particular

instances, as recalled in Sect. 1. In these particular cases we are not able to reach the same stable convergence of the aforementioned algorithms, since both our methods rely on a low-rank approximation of A by omitting the elements of the spectrum corresponding to small eigenvalues, not on an “a priori” knowledge of a stable basis. This is particularly evident in the “flat limit” case, as discussed in Sect. 4.

Nevertheless, in all other cases where an explicit basis is not known, our methods provide an effective way to face the approximation problems, and the omission of some redundant information results in an improved and more stable process.

Some features of the basis require further investigations. In particular, the stopping rule we used in the numerical experiments, seems to be too rough and needs to be improved.

For interested readers, at the web page of the second author, <http://www.math.unipd.it/~gsantin/soft.html>, it is available a set of Matlab programs which contains the functions implementing the new basis as well as the codes used to produce the examples here presented.

Acknowledgments We would like to thank the anonymous referees who, thanks to their suggestions, allow to significantly improve the paper and the results here presented. The authors have been supported by the funds 2012 of the University of Padua, project CPDA124755 “Multivariate approximation with application to image reconstruction”. Both authors are members of GNCS-INdAM.

References

1. Atkinson, K., Han, W.: Theoretical Numerical Analysis, vol. 39 of Texts in Applied Mathematics, 3rd edn. Springer, Dordrecht (2009)
2. Beatson, R.K., Cherrie, J.B., Mouat, C.T.: Fast fitting of radial basis functions: methods based on preconditioned GMRES iteration. *Adv. Comput. Math.* **11**(2–3), 253–270 (1999)
3. De Marchi, S., Santin, G.: A new stable basis for radial basis function interpolation. *J. Comput. Appl. Math.* **253**, 1–13 (2013)
4. De Marchi, S., Schaback, R.: Stability of kernel-based interpolation. *Adv. Comput. Math.* **32**(2), 155–161 (2010)
5. De Marchi, S., Schaback, R., Wendland, H.: Near-optimal data-independent point locations for radial basis function interpolation. *Adv. Comput. Math.* **23**(3), 317–330 (2005)
6. Fasshauer, G.E., McCourt, M.J.: Stable evaluation of Gaussian radial basis function interpolants. *SIAM J. Sci. Comput.* **34**(2), A737–A762 (2012)
7. Faul, A.C., Powell, M.J.D.: Krylov subspace methods for radial basis function interpolation. In: Numerical analysis 1999 (Dundee), vol. 420 of Chapman & Hall/CRC Res. Notes Math., pp. 115–141. Chapman & Hall/CRC, Boca Raton (2000)
8. Fornberg, B., Larsson, E., Flyer, N.: Stable computations with Gaussian radial basis functions. *SIAM J. Sci. Comput.* **33**(2), 869–892 (2011)
9. Fornberg, B., Lehto, E., Powell, C.: Stable calculation of Gaussian-based RBF-FD stencils. *Comput. Math. Appl.* **65**(4), 627–637 (2013)
10. Fornberg, B., Piret, C.: A stable algorithm for flat radial basis functions on a sphere. *SIAM J. Sci. Comput.* **30**(1), 60–80 (2007/2008)
11. Fornberg, B., Wright, G.: Stable computation of multiquadric interpolants for all values of the shape parameter. *Comput. Math. Appl.* **48**(5–6), 853–867 (2004)
12. Fornberg, B., Zuev, J.: The Runge phenomenon and spatially variable shape parameters in RBF interpolation. *Comput. Math. Appl.* **54**(3), 379–398 (2007)
13. Franke, R.: A critical comparison of some methods for interpolation of scattered data. Tech. Rep. NPS-53-79-003. Naval Postgraduate School, Monterey, Calif (1979)
14. Müller, S., Schaback, R.: A Newton basis for kernel spaces. *J. Approx. Theory* **161**(2), 645–655 (2009)

15. Novati, P., Russo, M.R.: A GCV based Arnoldi-Tikhonov regularization method. *BIT* **54**(2), 501–521 (2014)
16. Pazouki, M., Schaback, R.: Bases for kernel-based spaces. *J. Comput. Appl. Math.* **236**(4), 575–588 (2011)
17. Wendland, H.: *Scattered Data Approximation*, vol. 17 of Cambridge Monographs on Applied and Computational Mathematics. Cambridge University Press, Cambridge (2005)