Available online at www.sciencedirect.com

ScienceDirect

Applied Numerical Mathematics 00 (2015) 1–13

Appl. Numer. Math.

Partition of unity interpolation using stable kernel-based techniques

R. Cavoretto[a,*], S. De Marchi[b], A. De Rossi[a], E. Perracchione[a], G. Santin[b]

[a]*Department of Mathematics "G. Peano", University of Torino, via Carlo Alberto 10, I–10123 Torino, Italy*
[b]*Department of Mathematics, University of Padova, via Trieste 63, I–35121 Padova, Italy*

This paper is dedicated to Prof. Francesco A. Costabile on the occasion of his 70th birthday

Abstract

In this paper we propose a new stable and accurate approximation technique which is extremely effective for interpolating large scattered data sets. The Partition of Unity (PU) method is performed considering Radial Basis Functions (RBFs) as local approximants and using locally supported weights. In particular, the approach consists in computing, for each PU subdomain, a stable basis. Such technique, taking advantage of the local scheme, leads to a significant benefit in terms of stability, especially for flat kernels. Furthermore, an optimized searching procedure is applied to build the local stable bases, thus rendering the method more efficient.

Keywords: meshfree approximation, radial basis functions, partition of unity, scattered data interpolation, numerical stability, Krylov space methods.
2010 MSC: 65D05, 65D15, 65Y20.

1. Introduction

Considering the state of the art [5, 7, 12, 13, 21], we propose a new method for multivariate approximation which allows to interpolate large scattered data sets stably, accurately and with a relatively low computational cost.

The interpolant we consider is expressed as a linear combination of some basis or kernel functions. Focusing on Radial Basis Functions (RBFs), the Partition of Unity (PU) is performed by blending RBFs as local approximants and using locally supported weight functions. With this approach a large problem can be decomposed into many small problems, [2, 14, 20, 25], and therefore in the approximation process we could work with a large number of nodes.

However, in some cases, local approximants and consequently also the global one may suffer from instability due to ill-conditioning of the interpolation matrices. This is directly connected to the order of smoothness of the basis function and to the node distribution. It is well-known that the stability depends on the flatness of the RBF. More specifically, if one keeps the number of nodes fixed and considers smooth basis functions, then the problem of instability becomes evident for small values of the shape parameter. Of course, a basis function with a finite order of smoothness can be used to improve the conditioning but the accuracy of the fit gets worse. For this reason, the recent research is moved to the study of more stable bases.

[*]Corresponding author.

Email addresses: roberto.cavoretto@unito.it (R. Cavoretto), demarchi@math.unipd.it (S. De Marchi), alessandra.derossi@unito.it (A. De Rossi), emma.perracchione@unito.it (E. Perracchione), gsantin@math.unipd.it (G. Santin)

For particular RBFs, techniques allowing to stably and accurately compute the interpolant, also in the *flat limit* $\varepsilon \to 0$, have been designed in the recent years. These algorithms, named RBF-QR methods, are all rooted in a particular decomposition of the kernel, and they have been developed so far to treat the Gaussian and the Inverse Multiquadric kernel. We refer to [15, 17, 18, 19] for further details on these methods.

A different and more general approach, consisting in computing, via a truncated Singular Value Decomposition (SVD) stable bases, namely Weighted SVD (WSVD) bases, has been presented in [12]. We remark that in the cases where the RBF-QR algorithms can be applied, they produce a far more stable solution of the interpolation problem. Nevertheless, the present technique applies to *any* RBF kernel, and to any domain.

In this paper, a stable approach via the PU method, named WSVD-PU, which makes use of local WSVD bases and uses compactly supported weight functions, is presented. Thus, following [13], for each PU subdomain a stable RBF basis is computed in order to solve the local interpolation problem. Consequently, since the local approximation order is preserved for the global fit, the interpolant will result more stable and accurate. Concerning the stability, we can surely expect a more significant improvement in the stabilisation process with infinitely smooth functions than with functions characterized by a finite order of regularity. Moreover, in terms of accuracy, the benefits coming from the use of such stable bases are more significant in a local approach than in a global one. In fact, generally, while in the global case a large number of truncated terms of the SVD must be dropped to preserve stability, a local technique requires only few terms are eliminated, thus enabling the method to be much more accurate.

Concerning the computational complexity of the algorithm, the use of the so-called block-based space partitioning data structure enables us to efficiently organize points among the different subdomains, [5, 8]. Then, for each subdomain a local RBF problem is solved with the use of a stable basis. The main and truly high cost, involved in this step, is the computation of the SVD. To avoid this drawback, techniques based on Krylov space methods are employed, since they turn out to be really effective, [3, 13]. A complexity analysis supports our findings.

The guidelines of the paper are as follows. In Section 2, we present the WSVD bases, computed by means of the Lanczos algorithm, in the general context of global approximation. Such method is used coupled with the PU approach which makes use of an optimized searching procedure, as shown in Section 3. The proposed approach turns out to be stable and efficient, as stressed in Section 4. In Section 5 extensive numerical experiments, carried out with both globally and compactly supported RBFs of different orders of smoothness, support our findings. Moreover, all the Matlab codes are made available to the scientific community in a downloadable free software package:

http://hdl.handle.net/2318/1527447

2. RBF interpolation and WSVD basis

In Subsection 2.1 we briefly review the main theoretical aspects concerning RBF interpolation, [4], while the remaining subsections are devoted to the efficient computation of the WSVD basis via Krylov space methods.

2.1. RBF interpolation

Our goal is to recover a function $f : \Omega \to \mathbb{R}$, Ω being a bounded set in $\mathbb{R}^M$, using a set of samples of f on N pairwise distinct points $X_N \subset \Omega$, namely $\boldsymbol{f} = [f_1, \ldots, f_N]^T$, $f_i = f(\boldsymbol{x}_i)$, $\boldsymbol{x}_i \in X_N$. To this end, one considers a positive definite and symmetric kernel $\Phi : \Omega \times \Omega \longrightarrow \mathbb{R}$ to construct an interpolant in the form

$$R(\boldsymbol{x}) = \sum_{j=1}^{N} c_j \Phi(\boldsymbol{x}, \boldsymbol{x}_j), \quad \boldsymbol{x} \in \Omega. \tag{1}$$

The kernels we will consider are always radial, meaning that there exist a positive *shape parameter* ε and a function $\phi : \mathbb{R}_{\geq 0} \to \mathbb{R}$ such that for all $\boldsymbol{x}, \boldsymbol{y} \in \Omega$, $\Phi(\boldsymbol{x}, \boldsymbol{y}) = \phi_\varepsilon(\|\boldsymbol{x} - \boldsymbol{y}\|_2) = \phi(\varepsilon\|\boldsymbol{x} - \boldsymbol{y}\|_2)$. In Table 1 we report a list of some strictly positive definite radial kernels with their smoothness degrees. We remark that Gaussian, Inverse MultiQuadric and Matérn functions are globally supported and strictly positive definite in $\mathbb{R}^M$ for any M, whereas Wendland ones are compactly supported (whose support is $[0, 1/\varepsilon]$) and strictly positive definite in $\mathbb{R}^M$ for $M \leq 3$ (see [26]).

The real coefficients $\boldsymbol{c} = [c_1, \ldots, c_N]^T$ in (1) are determined by solving the linear system $A\boldsymbol{c} = \boldsymbol{f}$, where the interpolation (or kernel) matrix $A \in \mathbb{R}^{N \times N}$ is given by

$$A_{ij} = \Phi(\boldsymbol{x}_i, \boldsymbol{x}_j), \quad i, j = 1, \ldots, N. \tag{2}$$

RBF	$\phi_\varepsilon(r)$
Gaussian C^∞ (GA)	$e^{-\varepsilon^2 r^2}$
Inverse MultiQuadric C^∞ (IMQ)	$(1+\varepsilon^2 r^2)^{-1/2}$
Matérn C^6 (M6)	$e^{-\varepsilon r}(\varepsilon^3 r^3 + 6\varepsilon^2 r^2 + 15\varepsilon r + 15)$
Matérn C^4 (M4)	$e^{-\varepsilon r}(\varepsilon^2 r^2 + 3\varepsilon r + 3)$
Wendland C^6 (W6)	$(1-\varepsilon r)_+^8\,(32\varepsilon^3 r^3 + 25\varepsilon^2 r^2 + 8\varepsilon r + 1)$
Wendland C^4 (W4)	$(1-\varepsilon r)_+^6\,(35\varepsilon^2 r^2 + 18\varepsilon r + 3)$

Table 1. Examples of strictly positive definite radial kernels with their orders of smoothness and shape parameter $\varepsilon > 0$; $r = \|\cdot\|_2$ is the Euclidean distance, while $(\cdot)_+$ denotes the truncated power function.

The so constructed solution R is a function of the *native Hilbert space* $\mathcal{N}_\Phi(\Omega)$ uniquely associated with the kernel, and, if $f \in \mathcal{N}_\Phi(\Omega)$, it is in particular the $\mathcal{N}_\Phi(\Omega)$-projection of f into the subspace spanned by the *standard basis*

$$\mathcal{T}_{X_N} = \{\Phi(\boldsymbol{x}, \boldsymbol{x}_j), 1 \le j \le N\}.$$

We will denote this subspace as $\mathcal{N}_\Phi(X_N)$.

Although this interpolation method is known to be highly unstable in most cases being the matrix A severely ill conditioned, it has been proven (see [11]) that the interpolation operator $f \mapsto R$ is stable as an operator in the function space $\mathcal{N}_\Phi(\Omega)$. This gap has been widely recognized to be caused by the use of the standard basis, and a lot of efforts have been made in recent years to introduce better or perfectly conditioned basis (see [21] for a general theoretical treatment of this topic, and [6, 15, 17, 18] for particular instances of stable basis; for an overview see the book [16]).

2.2. WSVD basis

We are interested here in the use of the *WSVD basis* introduced in [12], thanks to its flexibility with respect to the choice of the kernel Φ. We recall in the following some relevant properties of this basis, while we refer to the paper [12] for further details.

To construct a basis $\mathcal{U} = \{u_j\}_{j=1}^N$ of $\mathcal{N}_\Phi(X_N)$ it is enough to assign an invertible coefficient matrix $D_{\mathcal{U}} = [d_{ij}]_{i,j=1}^N$ such that

$$u_j(\boldsymbol{x}) = \sum_{i=1}^N d_{ij}\Phi(\boldsymbol{x}, \boldsymbol{x}_i),$$

or, equivalently, an invertible value matrix $V_{\mathcal{U}} = [u_j(\boldsymbol{x}_i)]_{i,j=1}^N$. The two matrices are related as $A = V_{\mathcal{U}} \cdot D_{\mathcal{U}}^{-1}$ (see [21]), and in our situation they are defined as follows (see [12]).

Definition 2.1. *A WSVD basis $\mathcal{U}$ is a basis for $\mathcal{N}_\Phi(X_N)$ characterized by the matrices*

$$D_{\mathcal{U}} = \sqrt{\tilde{W}} \cdot Q \cdot \Sigma^{-1/2} \ \ \text{and} \ \ V_{\mathcal{U}} = \sqrt{\tilde{W}^{-1}} \cdot Q \cdot \Sigma^{1/2}, \quad \text{where} \quad \sqrt{\tilde{W}} \cdot A \cdot \sqrt{\tilde{W}} = Q \cdot \Sigma \cdot Q^T,$$

is a singular value decomposition of the scaled kernel matrix $A_{\tilde{W}} = \sqrt{\tilde{W}} \cdot A \cdot \sqrt{\tilde{W}}$, and $\tilde{W}_{ij} = \delta_{ij}\tilde{w}_i$ is a diagonal matrix of positive weights.

Notice that the definition uses a set of positive weights that was employed in the original formulation as cubature weights to construct the basis. Nevertheless, these weights do not change the numerical behavior of the basis, hence we will assume from now on $\tilde{w}_i = 1/N$, $1 \le i \le N$. Moreover, for notational convenience, the diagonal elements of Σ will be denoted as $\sigma_1 \ge \cdots \ge \sigma_N$.

This basis has been introduced to mimic in a discrete sense the properties of the *eigenbasis*. The latter is constructed starting from the operator $T : L_2(\Omega) \to L_2(\Omega)$,

$$T[f](\boldsymbol{x}) = \int_\Omega \Phi(\boldsymbol{x}, \boldsymbol{y}) f(\boldsymbol{y}) d\boldsymbol{y}, \tag{3}$$

through the following Theorem (see e.g. [22]).

Theorem 2.1 (Mercer's Theorem)**.** *If the kernel Φ is continuous and positive definite on a bounded set $\Omega \subseteq \mathbb{R}^M$, the operator T has a countable set of eigenfunctions $\{\varphi_k\}_k$ and eigenvalues $\{\lambda_k\}_k$. The eigenfunctions are orthonormal in $L_2(\Omega)$ and orthogonal in $\mathcal{N}_\Phi(\Omega)$ with $\|\varphi_k\|_{\mathcal{N}_\Phi(\Omega)} = \lambda_k^{-1}$. Moreover, the kernel can be expressed in terms of the eigencouples as*

$$\Phi(\boldsymbol{x}, \boldsymbol{y}) = \sum_k \lambda_k \varphi_k(\boldsymbol{x}) \varphi_k(\boldsymbol{y}), \;\; \boldsymbol{x}, \boldsymbol{y} \in \Omega,$$

where the series converges uniformly and absolutely.

For its use in interpolation in $\mathcal{N}_\Phi(\Omega)$, it is convenient to use the basis $\{\sqrt{\lambda_k}\varphi_k\}_k$, that is normalized in $\mathcal{N}_\Phi(\Omega)$. With this normalization, the basis has the following properties.

Property 2.1. *The eigenbasis $\{\sqrt{\lambda_k}\varphi_k\}_k$ has the following properties:*

i. *it is $\mathcal{N}_\Phi(\Omega)$-orthonormal,*
ii. *it is $L_2(\Omega)$-orthogonal with norm λ_k,*
iii. $(\sqrt{\lambda_k}\varphi_k, f)_{L_2(\Omega)} = \lambda_k(\sqrt{\lambda_k}\varphi_k, f)_{\mathcal{N}_\Phi(\Omega)}, \forall f \in \mathcal{N}_\Phi(\Omega)$,
iv. $\lambda_k \geq \lambda_{k+1}$ *and* $\lambda_k \to 0$ *as* $k \to \infty$,
v. $\sum_k \lambda_k = \phi(0)\,\text{meas}(\Omega)$.

As proven in [12], the WSVD basis enjoys the same properties when the inner product of $L_2(\Omega)$ is replaced with its discrete version $\ell_2(X_N)$, as summarized in the following statement.

Property 2.2. *The WSVD basis $\{u_k\}_{k=1}^N$ has the following properties:*

i. *it is $\mathcal{N}_\Phi(\Omega)$-orthonormal,*
ii. *it is $\ell_2(X_N)$-orthogonal with norm σ_k,*
iii. $(u_k, f)_{\ell_2(X_N)} = \sigma_k(u_k, f)_{\mathcal{N}_\Phi(\Omega)}, \forall f \in \mathcal{N}_\Phi(\Omega)$,
iv. $\sigma_1 \geq \cdots \geq \sigma_N > 0$,
v. $\sum_{k=1}^N \sigma_k = \phi(0)\,\text{meas}(\Omega)$.

Since the interpolation is a $\mathcal{N}_\Phi(\Omega)$-projection, we can rewrite the interpolant R in terms of the $\mathcal{N}_\Phi(\Omega)$-orthonormal WSVD basis as

$$R(\boldsymbol{x}) = \sum_{k=1}^N (f, u_k)_{\mathcal{N}_\Phi(\Omega)} u_k(\boldsymbol{x}), \tag{4}$$

and, thanks to point (iii) of Property 2.2, this can be further rewritten as

$$R(\boldsymbol{x}) = \sum_{k=1}^N \sigma_k^{-1}(f, u_k)_{\ell_2(X_N)} u_k(\boldsymbol{x}). \tag{5}$$

The latter form of the interpolant shows that R is also the solution of the discrete least-squares approximation problem $\min \|f - g\|_{\ell_2(X_N)}$ among all functions in $g \in \mathcal{N}_\Phi(X_N)$. If we instead solve the minimization problem over the subspace $\text{span}\{u_1, \ldots, u_m\}$, $m \leq N$, we find a solution R^m given by the truncation of the interpolant, i.e.,

$$R^m(\boldsymbol{x}) = \sum_{k=1}^m \sigma_k^{-1}(f, u_k)_{\ell_2(X_N)} u_k(\boldsymbol{x}). \tag{6}$$

Observe that it makes sense to consider the last minimization problem and its solution R^m for some $m \leq N$ instead of the original interpolation problem, since in this way we leave out the portion of the subspace $\mathcal{N}_\Phi(X_N)$ corresponding to small singular values, and this corresponds to solve the linear system (2) by means of a low-rank approximation of the matrix A. A detailed discussion of this approach can be found in [12], but we remark here that this method strictly depends on the behavior of the singular values of A. Namely, if we consider smoother RBFs, then by using (6) instead of the standard approach a better stabilization of the interpolation process is expected.

On the other hand, this method has some disadvantages. First, it is required to compute a singular value decomposition of the (possibly large) kernel matrix, and in the end only a few elements of the decomposition are used. This is computationally expensive, but in the next section we will explain how to overcome this problem. Second, this method requires to neglect part of the information to reduce instability, and, in some cases, this removal is too big to obtain a meaningful approximant. A solution to this problem is provided by the coupling with a localization method, and it is the main topic of this paper.

2.3. Fast computation through Krylov space methods

We present here a way to compute an approximation of the WSVD basis that makes use of the Lanczos algorithm. The method is discussed in [13], and it aims at reducing the computational cost of the procedure by approximating the truncated SVD of A.

We start by a general description of the Lanczos algorithm. Further details can be found in [9, 23, 24]. Let $\mathcal{K}_m(A,\boldsymbol{f}) = \text{span}\{\boldsymbol{f}, A\boldsymbol{f}, \dots, A^{m-1}\boldsymbol{f}\}$ be the Krylov subspace of order m generated by the matrix A and the vector $\boldsymbol{f}$. The Lanczos method computes an orthonormal basis $\{\boldsymbol{p}_i\}_{i=1}^m$ of $\mathcal{K}_m(A,\boldsymbol{f})$ through a Gram-Schmidt orthonormalization, i.e., the Lanczos basis $\{\boldsymbol{p}_i\}_{i=1}^m$, is computed by the following recurrence formula:

$$\beta_{i+1}\boldsymbol{p}_{i+1} = A\boldsymbol{p}_i - \alpha_i\boldsymbol{p}_i - \beta_i\boldsymbol{p}_{i-1}, \quad \text{with} \quad \beta_1\boldsymbol{p}_0 = 0. \tag{7}$$

Letting $P_m \in \mathbb{R}^{N\times m}$ the matrix having the vectors $\boldsymbol{p}_i$ as columns, and letting H_m be the $(m+1)\times m$ tridiagonal matrix defined as

$$H_m = \begin{pmatrix} \alpha_1 & \beta_2 & \cdots & 0 \\ \beta_2 & \alpha_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \beta_{m-1} \\ 0 & \cdots & \beta_{m-1} & \alpha_m \\ 0 & \cdots & 0 & \beta_m \end{pmatrix} \tag{8}$$

the algorithm can be formulated in matrix form as

$$AP_m = P_{m+1}\bar{H}_m, \quad \bar{H}_m = \begin{pmatrix} H_m \\ \bar{h}\boldsymbol{e}_m^T \end{pmatrix}, \tag{9}$$

where $\boldsymbol{e}_m \in \mathbb{R}^m$ is the unit vector and $\bar{h}$ is a scalar value.

Once we compute the matrices, the solution of the initial system can be approximated as $\boldsymbol{x} = P_m\boldsymbol{y}$, where $\boldsymbol{y} \in \mathbb{R}^m$ is such that $\bar{H}_m\boldsymbol{y} = \|\boldsymbol{f}\|_2\boldsymbol{e}_1$.

The idea is to use the matrix $\bar{H}_m$ to approximate the SVD of A, and since A has usually a good low-rank approximation, we expect to do so with $m \ll N$. Specifically, let $\bar{H}_m = U_m\bar{\Sigma}_m V_m^T$ be a singular value decomposition of $\bar{H}_m$, where $U_m \in \mathbb{R}^{(m+1)\times(m+1)}$, $V_m \in \mathbb{R}^{m\times m}$ are unitary matrices and

$$\bar{\Sigma}_m = \begin{pmatrix} \Sigma_m \\ 0 \end{pmatrix},$$

with Σ_m the diagonal matrix with singular values σ_i, $1 \le i \le m$ on the diagonal.

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. Thus, to simplify the notation we will denote by U_m the matrix without the last column, so that the decomposition becomes $\bar{H}_m = U_m\Sigma_m V_m^T$.

Now we want to define an approximation of the WSVD basis using the approximate SVD of A. We define a set of functions $\{\bar{u}_k\}_{k=1}^m \in \mathcal{N}_\Phi(X_N)$ which shows similarities with the WSVD basis, even if it does not form a basis, since they do not span $\mathcal{N}_\Phi(X_N)$. Anyway, we will go on calling such set of functions, with abuse of notation, basis.

Definition 2.2. *The approximate WSVD basis is characterized by the matrices*

$$D_{\bar{\mathcal{U}}_m} = P_m \cdot V_m \cdot \Sigma_m^{-1/2} \;\; \textit{and} \;\; V_{\bar{\mathcal{U}}_m} = P_{m+1} \cdot U_m \cdot \Sigma_m^{1/2}.$$

where $AP_m = P_{m+1}\bar{H}_m$ is the Lanczos decomposition of A of order m and $\bar{H}_m = U_m\Sigma_m V_m^T$ is a singular value decomposition of $\bar{H}_m$.

The next statement clarifies the connection between the two basis. As it is evident by construction, the basis strongly depends on the particular function $f \in \mathcal{N}_\Phi(\Omega)$.

Property 2.3. *The approximate WSVD basis $\{\bar{u}_k\}_{k=1}^m$ has the following properties:*

i. *it is near $\mathcal{N}_\Phi(\Omega)$-orthonormal, meaning that its $\mathcal{N}_\Phi(\Omega)$-Gramian is the identity matrix plus a rank one matrix,*
ii. *it is $\ell_2(X_N)$-orthogonal with norm σ_k,*
iii. *$(\bar{u}_k, f)_{\ell_2(X_N)} = \sigma_k(\bar{u}_k, f)_{\mathcal{N}_\Phi(\Omega)}$ if f is the function used to construct the basis,*
iv. *$\sigma_1 \geq \cdots \geq \sigma_m > 0$,*
v. *it coincides with the WSVD basis if $m = N$.*

This basis allows to solve again the least square approximation problem. Namely, if $f \in \mathcal{N}_\Phi(\Omega)$ is the function used for the Lanczos algorithm, the approximant $\bar{R}^m$ defined as

$$\bar{R}^m(\boldsymbol{x}) = \sum_{k=1}^m \sigma_k^{-1}(f, \bar{u}_k)_{\ell_2(X_N)}\bar{u}_k(\boldsymbol{x}),$$

minimizes the distance $\|f - g\|_{\ell_2(X_N)}$ for $g \in \text{span}\{\bar{u}_1, \ldots, \bar{u}_m\}$, $m \leq N$. Moreover, thanks to property (iii) of Property 2.3, $\bar{R}^m$ can be written in terms of $\mathcal{N}_\Phi(\Omega)$-inner products as

$$\bar{R}^m(\boldsymbol{x}) = \sum_{k=1}^m (f, \bar{u}_k)_{\mathcal{N}_\Phi(\Omega)}\bar{u}_k(\boldsymbol{x}). \tag{10}$$

Notice that property (v) of Property 2.3 proves that $\bar{R}^N = R^N = R$.

Approximating R^m with its fast computable version $\bar{R}^m$ solves efficiently the problems. We will see in the following sections how to successfully couple this technique with a fast domain decomposition method.

3. Partition of unity method using stable bases

The main idea is to use the stable basis introduced in the previous section in order to generate local stable approximants and accumulate them into the global fit.

3.1. A stable computation of the PU interpolant

Let $\Omega \subseteq \mathbb{R}^M$ be an open and bounded domain, and let $\{\Omega_j\}_{j=1}^d$ be an open and bounded covering of Ω satisfying some mild overlap condition among the subdomains (or patches) Ω_j. The set $I(\boldsymbol{x}) = \{j : \boldsymbol{x} \in \Omega_j\}$, for $\boldsymbol{x} \in \Omega$, is uniformly bounded on Ω, with $\Omega \subseteq \bigcup_{j=1}^d \Omega_j$. Associated with the subdomains we choose partition of unity weight functions W_j, i.e. a family of compactly supported, nonnegative and continuous functions subordinate to the subdomain Ω_j, such that $\sum_{j=1}^d W_j(\boldsymbol{x}) = 1$ on Ω and $\text{supp}(W_j) \subseteq \Omega_j$.

In order to have a better stabilization of the global fit, our goal is to define stable approximants of the form (10) on each subdomain Ω_j. In other words, for each (local) matrix $A_j \in \mathbb{R}^{N_j \times N_j}$, i.e. the j-th interpolation matrix associated with the subdomain Ω_j, a low-rank approximation is computed and thus the so-called WSVD-PU approximant is given by:

$$\bar{\mathcal{I}}(\boldsymbol{x}) = \sum_{j=1}^d \bar{R}_j^{m_j}(\boldsymbol{x})W_j(\boldsymbol{x}), \quad \boldsymbol{x} \in \Omega, \tag{11}$$

where $W_j : \Omega_j \longrightarrow \mathbb{R}$ is a partition of unity weight function and $X_{N_j} = X_N \cap \Omega_j$. As evident, $\bar{R}_j^{m_j}$ defines a local stable RBF approximant on Ω_j of the form:

$$\bar{R}_j^{m_j}(\boldsymbol{x}) = \sum_{k=1}^{m_j} (\sigma_k^{(j)})^{-1}(f_{|\Omega_j}, \bar{u}_k^{(j)})_{\ell_2(X_{N_j})}\bar{u}_k^{(j)}(\boldsymbol{x}), \quad \boldsymbol{x} \in \Omega_j. \tag{12}$$

According to [25], if we assume to have a k-stable partition of unity, then the derivatives of the weight functions satisfy

$$||D^{\beta} W_j||_{L^{\infty}(\Omega_j)} \leq \frac{C_{\beta}}{\delta_j^{|\beta|}}, \quad |\beta| \leq k, \quad \forall \beta \in \mathbb{N}^M,$$

where δ_j is the diameter of Ω_j and $C_{\beta} > 0$ is a constant. As nonnegative functions $W_j \in C^k(\mathbb{R}^M)$, we may consider a *Shepard weight*, i.e.,

$$W_j(\boldsymbol{x}) = \frac{\varphi_j(\boldsymbol{x})}{\sum_{k \in I(\boldsymbol{x})} \varphi_k(\boldsymbol{x})}, \quad j = 1, \ldots, d,$$

$\varphi_j(\boldsymbol{x})$ being compactly supported functions with support on Ω_j, such as the Wendland functions [26].

Before computing the global fit by mean of local stable approximants obtained with the Lanczos procedure, we briefly sketch in the sequel some relevant properties. Since point (v) of Property 2.3 implies $\overline{R}^{N_j} = R^{N_j} = R_j$, we can recover the PU interpolant, by considering $\overline{R}^{N_j}$, i.e.:

$$\mathcal{I}(\boldsymbol{x}) = \sum_{j=1}^{d} \overline{R}_j^{N_j}(\boldsymbol{x}) W_j(\boldsymbol{x}) = \sum_{j=1}^{d} \sum_{k=1}^{N_j} (\sigma_k^{(j)})^{-1} (f_{|\Omega_j}, \overline{u}_k^{(j)})_{\ell_2(X_{N_j})} \overline{u}_k^{(j)}(\boldsymbol{x}), \quad \boldsymbol{x} \in \Omega. \tag{13}$$

Remark 3.1. *If the functions $\overline{R}_j^{N_j}$, $j = 1, \ldots, d$, satisfy the interpolation conditions $\overline{R}_j^{N_j}(\boldsymbol{x}_i) = f(\boldsymbol{x}_i)$ for each $\boldsymbol{x}_i \in \Omega_j$, then the global PU approximant inherits the interpolation property of the local interpolants, i.e.*

$$\mathcal{I}(\boldsymbol{x}_i) = \sum_{j=1}^{d} \overline{R}_j^{N_j}(\boldsymbol{x}_i) W_j(\boldsymbol{x}_i) = \sum_{j \in I(\boldsymbol{x}_i)} f(\boldsymbol{x}_i) W_j(\boldsymbol{x}_i) = f(\boldsymbol{x}_i).$$

In order to be able to formulate error bounds, we need some further assumptions on regularity of Ω_j and define the *fill distance*

$$h_{X_N,\Omega} = \sup_{\boldsymbol{x} \in \Omega} \min_{\boldsymbol{x}_i \in X_N} ||\boldsymbol{x} - \boldsymbol{x}_i||_2.$$

Specifically, we require that an open and bounded covering $\{\Omega_j\}_{j=1}^d$ is *regular* for (Ω, X_N). This means to fulfill the following properties [25]:

i. for each $\boldsymbol{x} \in \Omega$, the number of subdomains Ω_j with $\boldsymbol{x} \in \Omega_j$ is bounded by a global constant C;
ii. there exists a constant $C_r > 0$ and an angle $\theta \in (0, \pi/2)$ such that every subdomain Ω_j satisfies an interior cone condition with angle θ and radius $r = C_r h_{X_N,\Omega}$;
iii. the local fill distances $h_{X_{N_j},\Omega_j}$ are uniformly bounded by the global fill distance $h_{X_N,\Omega}$.

Remark 3.2. *The first property ensures that* (13) *is actually a sum over at most C summands. Moreover, it is crucial for an efficient evaluation of the global approximant that only a constant number of local interpolants has to be evaluated. It follows that it should be possible to locate those C indices in constant time. The second and third properties are significant for estimating errors of RBF interpolants.*

After defining the space $C_{\nu}^k(\mathbb{R}^M)$ of all functions $f \in C^k$ whose derivatives of order $|\beta| = k$ satisfy $D^{\beta} f(\boldsymbol{x}) = \mathcal{O}(||\boldsymbol{x}||_2^{\nu})$ for $||\boldsymbol{x}||_2 \longrightarrow 0$, we consider the following convergence result [14, 26]:

Theorem 3.1. *Let $\Omega \subseteq \mathbb{R}^M$ be open and bounded and suppose that $X_N = \{\boldsymbol{x}_i, i = 1, \ldots, N\} \subseteq \Omega$. Let $\phi \in C_{\nu}^k(\mathbb{R}^M)$ be a strictly positive definite function. Let $\{\Omega_j\}_{j=1}^d$ be a regular covering for (Ω, X_N) and let $\{W_j\}_{j=1}^d$ be k-stable for $\{\Omega_j\}_{j=1}^d$. Then the error between $f \in \mathcal{N}_{\phi}(\Omega)$, where $\mathcal{N}_{\phi}$ is the native space of ϕ, and its PU interpolant* (13) *can be bounded by:*

$$|D^{\beta} f(\boldsymbol{x}) - D^{\beta} \mathcal{I}(\boldsymbol{x})| \leq C h_{X_N,\Omega}^{\frac{k+\nu}{2} - |\beta|} |f|_{\mathcal{N}_{\phi}(\Omega)},$$

for all $\boldsymbol{x} \in \Omega$ and all $|\beta| \leq k/2$.

3.2. The PU algorithm: the Lanczos procedure

For each subdomain, in order to generate the local stable approximation matrix, the Lanczos method is applied to the matrix $A_j \in \mathbb{R}^{N_j \times N_j}$ and to the function values $\boldsymbol{f}_j \in \mathbb{R}^{N_j}$ associated with the subdomain Ω_j. In this way the matrix H_{m_j} and the Lanczos basis $\{\boldsymbol{p}_i^{(j)}\}_{i=1}^{m_j}$ are computed for each subdomain. Then, for each interpolation problem a local stable basis is formed.

By using a different stopping criterion in the Lanczos algorithm, with respect to the one employed in [13], we can compute stable bases for a wider family of RBFs, both globally defined and compactly supported. The main problem in the Lanczos procedure concerns the stopping criterion, (see `Step 3` of the `Lanczos Algorithm`). From Property 2.2 (point (v)) and Property 2.3 (point (v)) a reliable one is:

$$\left| \phi(0) - \frac{1}{N_j} \sum_{k=1}^{m_j} \alpha_k^{(j)} \right| < \tau, \tag{14}$$

for a certain fixed tolerance τ, which is supposed to be equal for all the subdomains.

From Property 2.3 (point (v)), the fact that we impose as maximum number of iterations, in the `Lanczos Algorithm` at `Step 2`, exactly the number of nodes in Ω_j, i.e. N_j, naturally follows.

```
INPUTS: $N_j$, number of data in $\Omega_j$; $A_j$, the local interpolation matrix;
        $\boldsymbol{f}_j$, the function values associated to $\Omega_j$;
        $\tau$, the tolerance used as stopping criterion; $\phi$, the radial basis
        function.
OUTPUTS: $\boldsymbol{p}_1^{(j)}, \ldots, \boldsymbol{p}_m^{(j)}$, the new basis in $\Omega_j$; $H_{m_j}$, the tridiagonal matrix.
Step 1: Set $\beta_1^{(j)} = 0$; $\boldsymbol{p}_0^{(j)} = 0$; $\boldsymbol{p}_1^{(j)} = \frac{\boldsymbol{f}_j}{||\boldsymbol{f}_j||_2}$.
Step 2: For $i = 1 : N_j$
        $\tilde{\boldsymbol{p}}_i^{(j)} = A_j \boldsymbol{p}_i^{(j)} - \beta_i \boldsymbol{p}_{i-1}^{(j)}$
        $\alpha_i^{(j)} = (\tilde{\boldsymbol{p}}_i^{(j)}, \boldsymbol{p}_i^{(j)})$
        $\tilde{\boldsymbol{p}}_i^{(j)} = \tilde{\boldsymbol{p}}_i^{(j)} - \alpha_i^{(j)} \boldsymbol{p}_i^{(j)}$
        $\beta_{i+1}^{(j)} = ||\tilde{\boldsymbol{p}}_i^{(j)}||_2$
          Step 3: If $\beta_{i+1}^{(j)} = 0$ or $|\phi(0) - \frac{1}{N_j} \sum_{k=1}^{i} \alpha_k^{(j)}| < \tau$
                  break
        $\boldsymbol{p}_{i+1}^{(j)} = \tilde{\boldsymbol{p}}_i^{(j)} / \beta_{i+1}^{(j)}$
```

Table 2. The `Lanczos Algorithm`. Routine performing the Lanczos procedure.

Then, once the matrix H_{m_j} is found for Ω_j the stable basis is computed by calculating the singular value decomposition of H_{m_j} and a local approximant on each subdomain in the form (12) is computed. Then the local fits are accumulated into the global one.

The use of stable bases by decomposing the initial problem into many small ones leads to a larger benefit in terms of accuracy than employing a global approach. In fact if one uses a global method the approximant results stable, but a large number of terms in the Lanczos procedure are neglected. This surely leads to a decrease of the fit accuracy. Whereas the local method turns out to be really accurate since, dealing with small problems, less terms in the computation of the basis are eliminated to preserve stability.

Extensive numerical experiments support our findings.

3.3. The PU algorithm: the block-based algorithm structure

The key step of the PU method consists in organizing the scattered data among the subdomains. To this aim the kd-tree partitioning structures are widely used, [14]. However, they are not specifically implemented for the PU method.

Here a novel partitioning procedure, specifically the so-called *block-based partitioning structure*, built for bivariate and trivariate interpolation in order to determine the points belonging to the different PU subdomains, is considered, [8]. Even if such partitioning structure is robust enough to work on 2D or 3D irregular domains, we present such efficient technique for a scattered data set lying in the unit square, i.e. $\Omega = [0, 1]^2$.

At first, a partition of unity structure, composed by d circular patches Ω_j of radius:

$$\delta = \sqrt{\frac{2}{d}}, \tag{15}$$

and whose centres $\bar{\boldsymbol{x}}_j$, $j = 1, \ldots, d$, are a grid of points on Ω, is generated. As in [14], the number of PU subdomains is chosen so that $N/d \approx 4$. This choice and (15) lead to a reliable partition of unity structure since, in this way, patches form a covering of the domain Ω.

In order to find the points belonging to the different subdomains and consequently solve, with the use of stable bases, d small interpolation problems, we propose a new partitioning structure. It leads to a natural searching procedure that turns out to be really cheap in terms of computational complexity. To this aim we first cover Ω with q^2 square blocks, where the number q of blocks along one side of the unit square is:

$$q = \left\lceil \frac{1}{\delta} \right\rceil. \tag{16}$$

In this way the width of blocks is equal to the subdomain radius. This choice can appear trivial, but on the contrary it enables us to consider in the searching process an optimized number of blocks.

Blocks are numbered from 1 to q^2 (bottom to top, left to right). Thus, with a repeated use of a quicksort routine the set X_N is partitioned by the block-based partitioning structure into q^2 subsets X_{N_k}, $k = 1, \ldots, q^2$, where X_{N_k} are the points stored in the k-th *neighbourhood*, i.e. in the k-th block and in its eight neighbouring blocks. In such framework, we will be able to get an optimal procedure to find the nearest points. In fact, given a subdomain Ω_j, whose centre belongs to the k-th block, we search for all data lying in the j-th subdomain only among those lying in the k-th neighbourhood.

Remark 3.3. *The same partitioning structure, in case of Compactly Supported RBFs (CSRBFs), must be considered locally for each subdomain. In fact, in order to build the j-th stable approximation matrix, among all points lying in the j-th subdomain, only those belonging to the support of the CSRBF must be considered.*

Remark 3.4. *Among several routines which can be employed to determine the neighboring points, we choose the block-based data structure. Anyway, we stress that the algorithm, here proposed, works in any dimension M, while the block-based data structure is only implemented for $M = 2, 3$, [8]. Thus in higher dimensions such structure must be replaced by standard routines, such as kd-trees, [1, 10, 14].*

4. Complexity analysis

Since the stable WSVD-PU algorithm is characterized by the construction of local RBF stable approximants, we consider the local data sets, composed by N_j points, $j = 1, 2, \ldots, d$. Thus, the complexity of this algorithm is influenced by the following computational issues:

i. organize by means of a partitioning structure the nodes among the subdomains,
ii. compute the stable basis on each subdomain

Concerning the efficient organization of points, an extensive complexity analysis, briefly shacked in Subsection 4.2, can be found in [8]. The cost associated to the computation of a local stable basis is investigated in Subsection 4.1.

4.1. Computation of a stable basis

Performing the Lanczos procedure on a matrix $B \in \mathbb{R}^{n\times n}$ requires $\mathcal{O}(kn^2)$, where k is the number of vectors computed by the algorithm, i.e. k is the *good* low rank approximation, (a priori unknown in our case), [9].

Given $A_j \in \mathbb{R}^{N_j\times N_j}$ the interpolation matrix defined on the Ω_j, the Lanczos method forms the matrix H_{m_j} for Ω_j after m_j iterations. Usually we have $m_j \ll N_j$, but in some cases the maximum number of iterations N_j can be reached and so, in a more general setting, $m_j \leq N_j$. This routine requires:

$$\mathcal{O}(m_j N_j^2) \leq \mathcal{O}(N_j^3), \tag{17}$$

time complexity. Thus for each subdomain the upper bound for the computational time of the Lanczos procedure is given by the right-hand side of (17).

In case of sparse matrices, such as the ones arising from the use of CSRBFs, the Lanczos procedure can be performed in: $\mathcal{O}(m_j(N_j + \tilde{n}))$ time complexity, where $\tilde{n}$ is the number of non-zero entries.

Then a singular value decomposition is applied to the matrix H_{m_j}. We remark that performing a singular value decomposition on a matrix $B \in \mathbb{R}^{n\times k}$ requires $\mathcal{O}(4n^2k + 8nk^2 + 9k^3)$ time complexity.

The singular value decomposition for each subdomain is applied to the matrix H_{m_j}; once more we stress that $m_j \ll N_j$. Thus for each subdomain the singular value decomposition can be performed in:

$$\mathcal{O}(4m_j^2 m_j + 8m_j m_j^2 + 9m_j^3) \approx \mathcal{O}(m_j^3) \tag{18}$$

time complexity.

4.2. The partitioning structure

Let us now focus on the block-based partitioning structure used to organize the N data sites in blocks. We remark that such efficient organization of points is specifically implemented for 2D data sets. Anyway, the proposed WSVD-PU algorithm is robust enough to work in any dimension M, provided that a different partitioning structure is performed.

Let $\overline{n}_s$ be the number of data sites belonging to a strip. The procedure used to store the points among the different subdomains is based on recursive calls to a *quicksort* routine which requires $\mathcal{O}(\overline{n}\log\overline{n})$, where $\overline{n}$ is the number of elements to be sorted. Thus, letting N/q the average number of points lying in a strip, the computational cost needed to organize the N points among the different subdomains is:

$$\mathcal{O}\Big(N\log N + \sum_{s=1}^{q} \overline{n}_s \log \overline{n}_s\Big) \approx \mathcal{O}\Big(\frac{3}{2} N \log N\Big). \tag{19}$$

Concerning the searching procedure, for each subdomain a quicksort procedure is used to order distances. Thus observing that the data sites in a neighbourhood are about $N/(3q)^2$ and taking into account the definitions of q and δ, the complexity can be estimated by:

$$\mathcal{O}\Big(\frac{N}{(3q)^2}\log\frac{N}{(3q)^2}\Big) \approx \mathcal{O}\Big(\frac{2N}{9d}\log\frac{2N}{9d}\Big) \approx \mathcal{O}(1). \tag{20}$$

The estimate (20) follows from the fact that we built a partitioning structure strictly related to the size of the subdomains and ad hoc for the PU method.

Remark 4.1. *The same computational cost* (20)*, in case of CSRBFs, must be considered locally for each subdomain, to build the sparse interpolation and evaluation matrices. In such steps we usually have a relatively small number of nodes N_j, with $N_j \ll N$, where the index j identifies the j-th subdomain.*

5. Numerical experiments

This section is devoted to point out, by means of extensive numerical simulations, stability and accuracy of the WSVD-PU interpolant. To this aim comparisons with the standard PU interpolant will be carried out.

Experiments are performed considering $N = (2^k + 1)^2$, $k = 6, 7, 8$, uniformly random Halton nodes, a grid of $d = \lfloor \sqrt{N}/2 \rfloor^2$ subdomain centres and a grid of $s = 40 \times 40$ evaluation points, which are contained in the unit square $\Omega = [0, 1] \times [0, 1]$.

In order to show the high stability of the proposed method, we compute the Root Mean Square Error (RMSE), i.e.

$$\text{RMSE} = \sqrt{\frac{1}{s}\sum_{i=1}^{s} |f(\tilde{\boldsymbol{x}}_i) - \bar{\mathcal{I}}(\tilde{\boldsymbol{x}}_i)|^2}, \tag{21}$$

for different values of the shape parameter in the range $\varepsilon = [10^{-3}, 10^2]$. Moreover, in order to point out the versatility of the proposed method, different kernels with different order of smoothness are considered, see Table 3. The error (21) is computed using as test function the well-known Franke's function:

$$\begin{aligned} f(x,y) = {} & \frac{3}{4}\exp\left[-\frac{(9x-2)^2+(9y-2)^2}{4}\right] + \frac{3}{4}\exp\left[-\frac{(9x+1)^2}{49} - \frac{9y+1}{10}\right] \\ & + \frac{1}{2}\exp\left[-\frac{(9x-7)^2+(9y-3)^2}{4}\right] - \frac{1}{5}\exp\left[-(9x-4)^2-(9y-7)^2\right]. \end{aligned}$$

In Figure 1 we compare the RMSEs obtained by means of the WSVD-PU interpolant (solid line) with the ones obtained performing the classical PU method (dashed line). As tolerance value in (14) we set 10^{-14}. These graphs point out that the use of the WSVD-PU local approach reveals a larger stability than the standard PU interpolant. Moreover, the use of a local method enables us to improve the RMSE for the optimal shape parameter in case of flat kernels, see Figure 1 and Table 3. This is consistent with the fact that in a local stable method, differently from [13], we have to solve small linear systems and therefore few terms are neglected in (12). Furthermore, from Figure 1 we can note that the WSVD-PU method turns out to be more effective with flat kernels, while for more picked bases the improvement of using stable bases becomes negligible as the order of bases function decreases. Thus, from our numerical experiments, we can observe three kinds of behavior depending on different RBF regularity classes. Specifically, the features of such classes, which differ both in terms of stability and accuracy from the standard basis, can be summarized as:

i. for C^∞ kernels: improvement of stability and of the optimal accuracy;

ii. for C^k kernels, with $k \geq 1$: improvement of stability and same optimal accuracy;

iii. for C^0 kernels: same stability and same optimal accuracy.

Moreover, since we are interested in pointing out the efficiency of the proposed WSVD-PU algorithm, in Table 4 we also report the CPU times obtained by using our stable interpolation method with the Gaussian RBF as local approximant, for each of the three different data sets. Tests have been carried out on a Intel(R) Core(TM) i3 CPU M330 2.13 GHz processor.

Acknowledgements

The first, third and fourth authors are partially supported by the University of Torino through research project "Metodi numerici nelle scienze applicate". The second and fifth authors are partially supported by the funds of the University of Padova, project CPDA124755 "Multivariate approximation with application to image reconstruction".

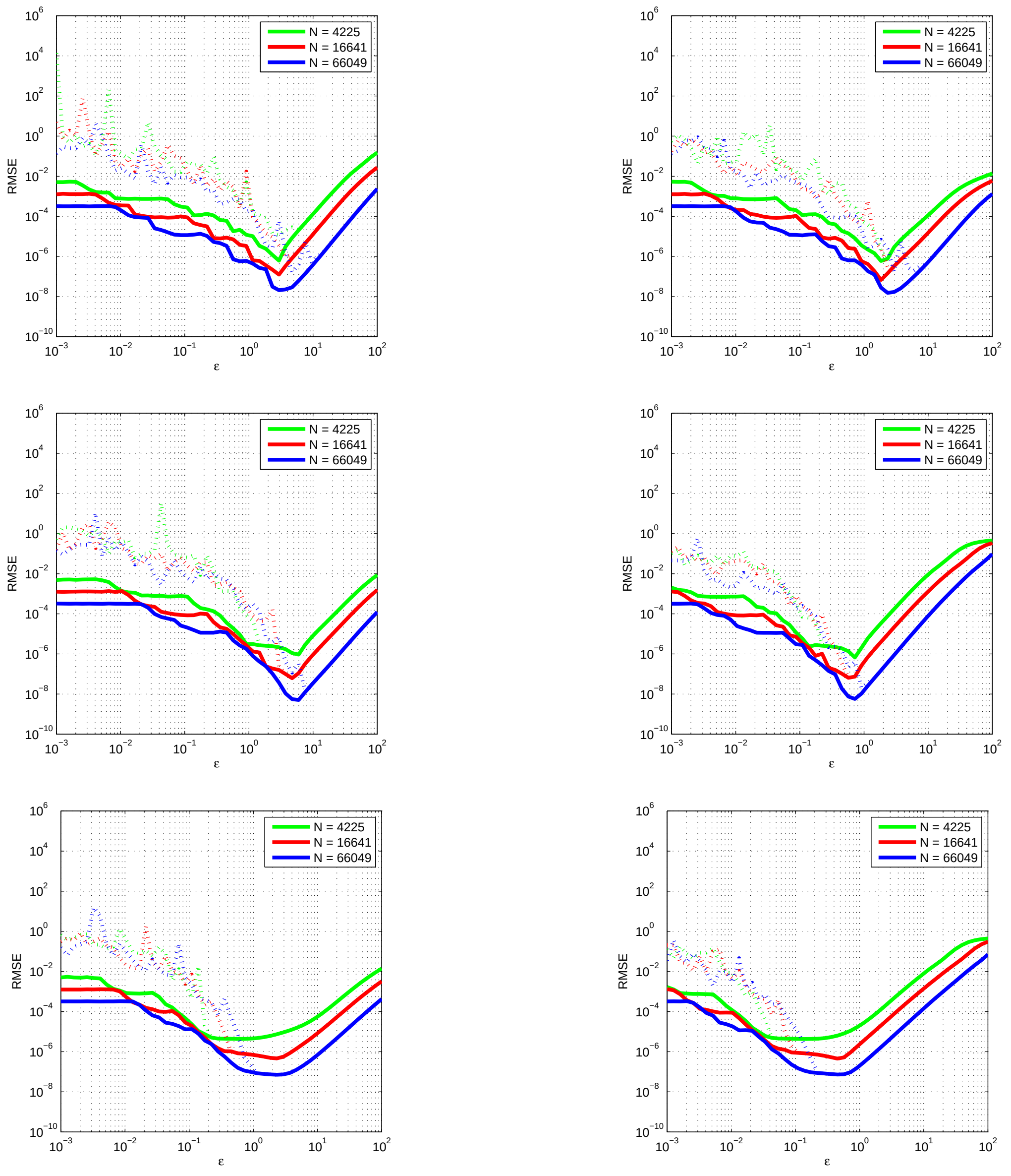

Figure 1. RMSEs obtained by varying ε for C^{∞}, C^6 and C^4 kernels. The classical PU interpolant is plotted with dashed line and the WSVD-PU approximant with solid line. From left to right, top to bottom we consider the Gaussian, Inverse MultiQuadric, Matérn C^6, Wendland C^6, Matérn C^4 and Wendland C^4 kernels, respectively.

References

[1] S. Arya, D. M. Mount, N. S. Netanyahu, R. Silverman, A. Y. Wu, An optimal algorithm for approximate nearest neighbor searching in fixed dimensions, J. ACM **45** (1998), 891–923.

[2] I. Babuška, J. M. Melenk, The partition of unity method, Internat. J. Numer. Methods Engrg. **40** (1997), 727–758.

[3] R. K. Beatson, J. B. Cherrie, C. T. Mouat, Fast fitting of radial basis functions: Methods based on preconditioned GMRES iteration, Adv. Comput. Math. **11** (1999), 253–270.

[4] M.D. Buhmann, Radial basis functions: theory and implementation, Cambridge Monogr. Appl. Comput. Math., vol. 12, Cambridge Univ. Press, Cambridge, 2003.

[5] R. Cavoretto, A. De Rossi, A meshless interpolation algorithm using a cell-based searching procedure, Comput. Math. Appl. **67** (2014), 1024–1038.

N	method	GA		IMQ		M6		W6	
		RMSE	ε_{opt}	RMSE	ε_{opt}	RMSE	ε_{opt}	RMSE	ε_{opt}
4225	PU	1.16E − 5	2.95	8.20E − 7	2.33	9.34E − 7	5.96	6.64E − 7	0.72
	WSVD-PU	6.20E − 7	2.95	5.98E − 7	1.84	9.34E − 7	5.96	6.64E − 7	0.72
16641	PU	9.70E − 7	3.73	2.94E − 7	2.33	6.18E − 8	4.71	6.44E − 8	0.57
	WSVD-PU	1.25E − 7	2.95	6.78E − 8	1.84	6.20E − 8	4.71	6.49E − 8	0.57
66049	PU	1.64E − 7	4.71	1.78E − 7	2.94	1.28E − 8	7.54	2.03E − 8	0.91
	WSVD-PU	2.09E − 8	2.95	1.54E − 8	2.33	5.10E − 9	5.96	5.70E − 9	0.72

Table 3. RMSEs obtained by using optimal values of ε for C^∞ and C^6 kernels.

N	4225	16642	66049
CPU [s]	3.94	14.69	57.73

Table 4. CPU times (in seconds) for the PU-WSVD method.

[6] R. Cavoretto, G. E. Fasshauer, M. McCourt, An introduction to the Hilbert-Schmidt SVD using iterated Brownian bridge kernels, Numer. Algorithms **68** (2015), 393–422.

[7] R. Cavoretto, A. De Rossi, A trivariate interpolation algorithm using a cube-partition searching procedure, SIAM J. Sci. Comput. **37** (2015), A1891–A1908.

[8] R. Cavoretto, A. De Rossi, E. Perracchione, Efficient computation of partition of unity interpolants through a block-based searching technique, submitted (2015).

[9] J. Chen, Y. Saad, Lanczos vectors versus singular vectors for effective dimension reduction, IEEE Transactions on Knowledge and Data Engineering **21** (2009), 1091–1103.

[10] M. De Berg, M. Van Kreveld, M. Overmars, O. Schwarzkopf, Computational geometry, Berlin, Springer (1997).

[11] S. De Marchi, R. Schaback, Stability of kernel-based interpolation, Adv. Comput. Math. **32** (2010), 155–161.

[12] S. De Marchi, G. Santin, A new stable basis for radial basis function interpolation, J. Comput. Appl. Math. **253** (2013), 1–13.

[13] S. De Marchi, G. Santin, Fast computation of orthonormal basis for RBF spaces through Krylov space methods, to appear in BIT (2015). DOI: 10.1007/s10543-014-0537-6.

[14] G. E. Fasshauer, Meshfree Approximation Methods with Matlab, World Scientific, Singapore, 2007.

[15] G. E. Fasshauer, M. J. McCourt, Stable evaluation of Gaussian radial basis function interpolants, SIAM J. Sci. Comput. **34** (2012), A737–A762.

[16] G. E. Fasshauer, M. J. McCourt, Kernel-based Approximation Methods using Matlab, World Scientific, Singapore, 2015.

[17] B. Fornberg, G. Wright, Stable computation of multiquadrics interpolants for all values of the shape parameter, Comput. Math. Appl. **47** (2004), 497–523.

[18] B. Fornberg, C. Piret, A stable algorithm for flat radial basis functions on a sphere, SIAM J. Sci. Comput. **30** (2007/08), 60–80.

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

[20] J. M. Melenk, I. Babuška, The partition of unity finite element method: basic theory and applications, Comput. Methods. Appl. Mech. Engrg. **139** (1996), 289–314.

[21] M. Pazouki, R. Schaback, Bases for kernel-based spaces, J. Comput. Appl. Math. **236** (2011), 575–588.

[22] W. Pogorzelski, Integral Equations and Their Applications. Vol. I, Translated from the Polish by Jacques J. Schorr-Con, A. Kacner and Z. Olesiak. International Series of Monographs in Pure and Applied Mathematics, Vol. 88. Pergamon Press, Oxford, 1966.

[23] H. D. Simon, H. Zha, Low-rank matrix approximation using the Lanczos bidiagonalization process with applications, SIAM J. Sci. Comput. **21** (2000), 2257–2274.

[24] S. Wei, Z. Lin, Accelerating iterations involving eigenvalue or singular value decomposition by block Lanczos with warm start, Microsoft Technical Report No. MSR-TR-2010-162.

[25] H. Wendland, Fast evaluation of radial basis functions: Methods based on partition of unity, in Approximation Theory X: Wavelets, Splines, and Applications, C. K. Chui, L. L. Schumaker, J. Stöckler (Eds.), Vanderbilt Univ. Press, Nashville, TN, 2002, pp. 473–483.

[26] H. Wendland, Scattered Data Approximation, Cambridge Monogr. Appl. Comput. Math., vol. 17, Cambridge Univ. Press, Cambridge, 2005.