Results, challenges and new steps on RBF approximation and computation*

Stefano De Marchi
Department of Medicine
University of Padova, Italy

Nadaniela Egidi, Josephin Giacomini, Pierluigi Maponi
School of Science and Technology
University of Camerino, Italy

Abstract

We present an up-to-date overview of approximation methods based on Radial Basis Function (RBF) techniques, which have recently attracted attention across various computational tools and application fields. This review study presents relevant results on RBF techniques, highlighting the associated computational challenges and stability issues that must be addressed when high-performance or parallel computation is required.

1 Introduction

The origins of RBF (Radial Basis Function) techniques date back to the second half of the last century and stem from a geodetic application by Hardy [1], in which irregular surfaces in cartography were obtained from sparse topographical data. In this seminal work, the representation of complex surfaces was proposed as a sum of quadric forms (such as cones or hyperboloids) centered at different data points. The simple idea of approximating data at scattered centers attracted the attention of several investigators, leading to the mathematical foundation of RBF or, more generally, to kernel-based approximation. The first relevant results about convergence, stability, and existence of solutions were provided in [2]. In [3], a critical benchmark test of various scattered data interpolation methods was presented, showing that the multiquadric method yielded the most accurate results. Another fundamental theoretical result is the proof for the solvability of the RBF interpolation matrix [4]. Furthermore, the theory of RBF interpolation can rely on a physical interpretation since the family of the

*Work partially supported by INdAM-GNCS research funds.

so-called *Thin Plate Spline* can be obtained as the unique minimizer of a specific semi-norm (representing "bending energy") [6].

These fundamental results provide a suitable computational environment for addressing various problems. RBFs are primarily valued for their mesh-free nature, which allows them to handle complex geometries and scattered data without a structured grid. For example, the Kansa method [7] allows one to compute solutions to Partial Differential Equations (PDEs) without a rigid geometric grid, unlike finite element approaches. RBFs are now the gold standard for mesh morphing [75, 8], in which an existing computational grid (e.g., around an aircraft wing) is smoothly deformed using a few control points while preserving the quality of the cells in the internal grid. RBFs are also used for elastic registration [9], in which two different images (e.g., an MRI - Magnetic Resonance Imaging - and a CT - Computed Tomography - scan of the same patient) are deformed to perfectly align the anatomical landmarks. This *meshless* approach is now standard for complex engineering and medical simulations.

RBFs also play a crucial role in scientific Machine Learning, both in Artificial Neural Networks (ANNs) and as kernels in Support Vector Machines (SVMs). The close connection between training ANNs and data-fitting processes is discussed in [10] and addressed using an RBF approach. Several contributions further strengthened this initial work on RBF networks. For instance, in [13] the authors demonstrated the efficiency of RBFs in learning complex mappings through locally tuned hidden units. The connection between RBFs and ANNs was generalized in [14], where RBF networks are shown to be universal approximators. The kernel trick was initially proposed in [15] to integrate kernel functions into the SVM framework. It demonstrated that SVMs could use RBF kernels to automatically select "supporting patterns" (support vectors) as the centers of the radial functions, allowing the model to adapt its complexity to the data.

Although RBFs are mathematically elegant approximation tools, they face several critical technical challenges that can compromise their effectiveness. The *Shape Parameter dilemma* affects most usual kernels, like Multiquadrics or Gaussians, and relies on the practical choice of the parameter that controls the basis function's width. Its choice is therefore crucial for obtaining effective implementations of RBF tools [16]. The *numerical instability* is strictly related to the shape parameter and to the ill-conditioning of the interpolation matrix as the number of data points increases [19]. The *computational complexity and scalability* are other critical aspects of RBF techniques. In fact, unlike the Finite Element Methods (FEM), classic RBF matrices are dense when global RBFs are used, so the solution of an RBF interpolation problem scales at $O(N^3)$ for the computational load and at $O(N^2)$ for the storage load, where N is the characteristic dimension of the problem. In addition, the *selection of centers* of the RBFs is a non-trivial optimization problem [42, 44, 45] that affects the accuracy of the overall

approximation process. The problem is also connected to the choice of a stable basis as discussed in [41, 43].

All these critical aspects must be properly addressed when the problem to be solved is large and/or difficult due to the problem's intrinsic instabilities. Thus, they are the key to making RBF a relevant tool in High Performance Scientific Computing (HPSC) [22]. In the next sections, we provide an overview of research efforts addressing critical aspects of RBF by selecting a few works. The selected material is also organized into two directions: those that mainly address the computational issues of RBF and those that address its stability issues. Of course, these aspects are never separable in any computational procedure, but this organization allows us to focus on the main contributions of the presented works.

The paper is organized as follows. In Section 2, we present the main achievements in the stabilization of RBF techniques. In Section 3, we discuss the results for efficient computer implementations of RBF techniques. In Section 4, we outline some conclusions and final remarks.

2 Conditioning and stabilization strategies

We start by recalling the RBF interpolation problem. Let $D \subset \mathbb{R}^d$ some subset of $\mathbb{R}^d$. Take the set of points $(\underline{x}_i, f_i)$, $i = 1, 2, \ldots, N$, being the interpolation data, where $X = \{\underline{x}_i \in D \subset \mathbb{R}^d,\ i = 1, 2, \ldots, N\}$ is the set of the centers (often corresponding to the interpolation points) and $f_i \in \mathbb{R}$, $i = 1, 2, \ldots, N$, are the interpolation values.

The interpolation problem is: given functions $\phi_i : D \to \mathbb{R}$, $i = 1, 2, \ldots, N$, compute the coefficients c_j, $j = 1, 2, \ldots, N$, such that

$$\sum_{j=1}^{N} c_j \phi_j(\underline{x}_i) = f_i, \quad i = 1, 2, \ldots, N, \tag{1}$$

of course, it is required that this solution is unique.

Name	smoothness	ϕ	support	SPD
Gaussian	C^∞	e^{-r^2}	global	✓
IM	C^∞	$(1+r^2)^{-1/2}$	global	✓
Matérn	C^2	$(1+r)e^{-r}$	global	✓
Wendland	C^2	$\max\{1-r,0\}^4(4r+1)$	compact	✓
TPS	C^∞	$r^2\log(r)$	global	✗

Table 1: Most used kernel functions in RBF approximation; the acronyms IM and TPS stand for Inverse Multiquadric and Thin Plate Spline, respectively. We note that the last column reports whether they are shape parameter dependent (SPD).

In the case of RBF, the functions ϕ_j, $j = 1, 2, \ldots, N$, are generated by a unique kernel function ϕ, more precisely $\phi_j(\underline{x}) = \phi(\|\underline{x} - \underline{x}_j\|)$, $j = 1, 2, \ldots, N$, $\underline{x} \in D$, with $\phi : [0, \infty) \to \mathbb{R}$. Among the various RBFs available in the literature, we have reported in Table 1 the most frequently used in applications: the Gaussian, the Inverse Multiquadric (IM), the Matérn, the Wendland, and the Thin Plate Splines (TPS). We note that the Matérn and Wendland functions reported in the table are twice-differentiable members of their respective families. Also, TPS can be seen as a special case of the polyharmonic function family (cf. e.g. [16] for details). The shape of these functions, except for the TPS, can be easily controlled by the so-called *shape parameter* $\epsilon > 0$ through the scaling of the kernel argument, that is, by considering $\phi(\epsilon r)$. In Figure 1, we have reported the kernel graphs (extended by parity) described in Table 1, and the graphs of $\phi(\epsilon \|\underline{x}\|)$, $\underline{x} \in \mathbb{R}^2$ for different choices of the shape parameter ϵ.

Assuming that ϕ is strictly positive definite (see e.g. [16]), the interpolation problem (1) has a unique solution since the interpolation matrix Φ, where $[\Phi]_{i,j} = \phi(\|\underline{x}_i - \underline{x}_j\|) \in \mathbb{R}^{N \times N}$, is strictly positive definite. We note that this does not affect the generality of the discussion, since, by the Paley–Wiener Theorem, for every conditionally positive definite function of order k (see e.g. [16]), there exists an associated strictly positive definite function (see e.g. [44]). However, the linear system (1) is usually ill-conditioned as N grows. Several authors have investigated the spectral properties of the interpolation matrix Φ; see, for example, [24, 25, 26, 27, 28, 29]. In particular, [30] proved that the minimum eigenvalue λ of Φ is bounded as follows:

$$cN^{-\beta/d} \leq \lambda \leq CN^{-\beta/d} \tag{2}$$

where $c, C > 0$ are constants depending on the interpolation points and the kernel function ϕ, $\beta > 0$ is a smoothness parameter for ϕ. More precisely, the Fourier transform $\hat{\phi}$ satisfies the following relation:

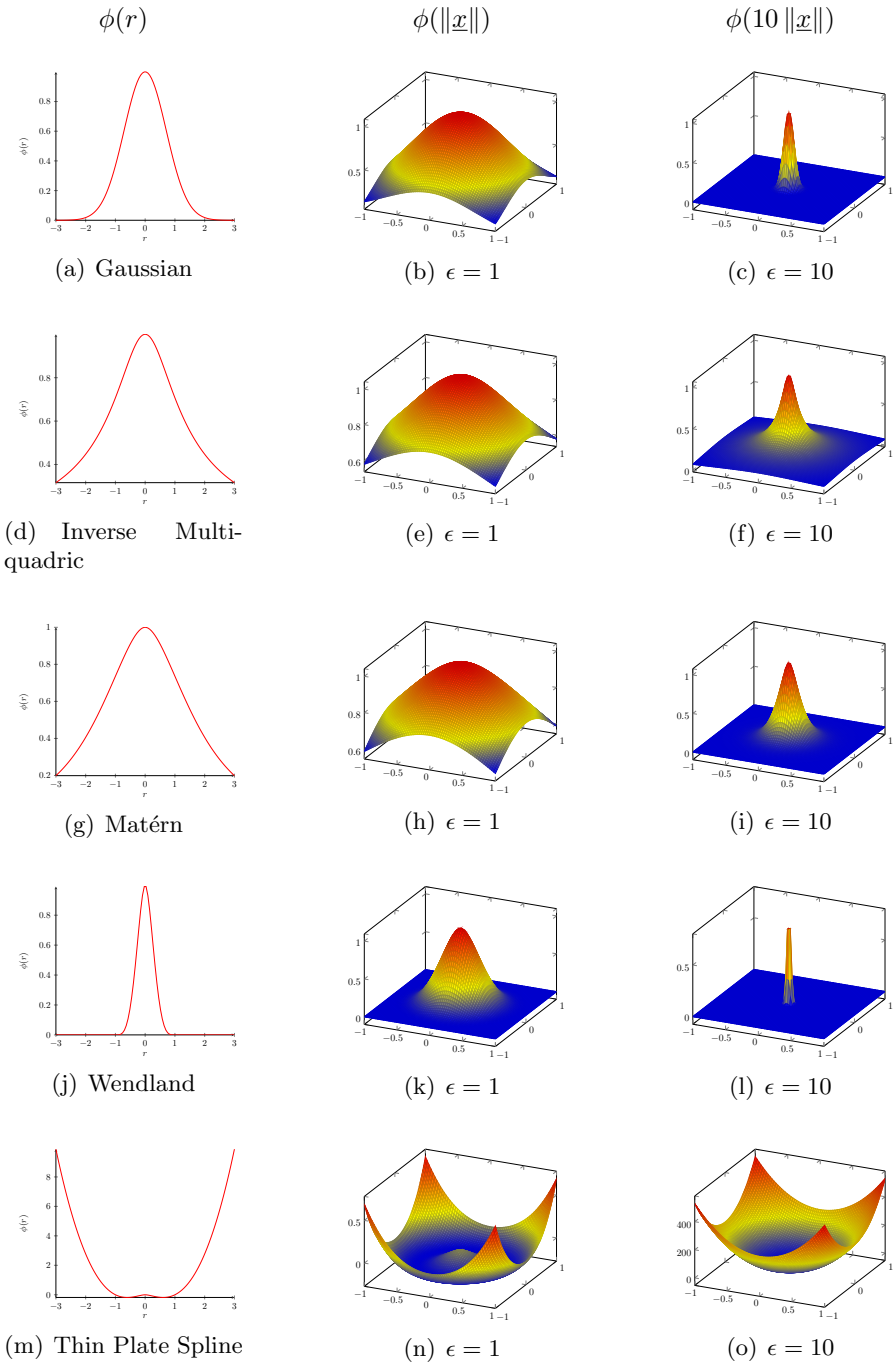
$$0 < \hat{\phi}(\omega) < A(1 + |\omega|)^{-\beta}, \quad \omega \in \mathbb{R}, \tag{3}$$

for a suitable constant $A > 0$. Besides, for smooth kernels such as Gaussian and the multiquadrics, (2) holds, with bounds that decay exponentially.

Obtained the c_j as the solution of (1), we can consider the function

$$P_f(\underline{x}) = \sum_{j=1}^{N} c_j \phi_j(\underline{x}), \quad \underline{x} \in D, \tag{4}$$

as an approximation for the function $f(\underline{x})$, $\underline{x} \in D$ that generated the interpolation data $(\underline{x}_i, f_i)$, $i = 1, 2, \ldots, N$, i.e., $f(\underline{x}_i) = f_i$, $i = 1, 2, \ldots, N$. The accuracy of P_f depends on the density of the point set X in D and on the regularity of the function f. The *fill distance* is the classical measure of the



(a) Gaussian (b) $\epsilon = 1$ (c) $\epsilon = 10$

(d) Inverse Multi-quadric (e) $\epsilon = 1$ (f) $\epsilon = 10$

(g) Matérn (h) $\epsilon = 1$ (i) $\epsilon = 10$

(j) Wendland (k) $\epsilon = 1$ (l) $\epsilon = 10$

(m) Thin Plate Spline (n) $\epsilon = 1$ (o) $\epsilon = 10$

Figure 1: Graphs of the kernels reported in Table 1. For $\mathbb{R}^2$ we used the shape parameters $\epsilon = 1$, 10.

density of the interpolation points, it is given by

$$h_{X,D} := \sup_{\underline{x} \in D} \min \left\{ \left\| \underline{x} - \underline{x}_j \right\|, \ j = 1, 2, \ldots, N \right\}, \tag{5}$$

and represents the radius of the largest ball that can be placed among the interpolation points.

Other quantities often used to analyze stability are the *separation distance*

$$q_X := \frac{1}{2} \min \left\{ \left\| \underline{x}_i - \underline{x}_j \right\|, \ 1 \leq i, j \leq N, \ i \neq j \right\}, \tag{6}$$

and the *uniformity*

$$\rho := \frac{q_X}{h_{X,D}}. \tag{7}$$

In particular, if ρ is near $1/\sqrt{d}$, the interpolation data are nearly equally distributed in the Euclidean norm on $\mathbb{R}^d$. Moreover, in contrast to polynomial interpolation, radial basis interpolants perform better when the interpolation nodes are nearly uniformly distributed [39].

Classical error estimates are usually given for functions in $N_\phi(D)$, the *native space* of ϕ which is obtained from the infinite dimensional space $H_\phi(D) = span\{\phi(\|\underline{x} - \cdot\|), \ \underline{x} \in D\}$ equipped with the scalar product:

$$\left(\sum_i p_i \phi(\|\underline{x}_i - \cdot\|), \sum_j q_j \phi(\left\|\underline{x}_j - \cdot\right\|) \right) = \sum_{i,j} p_i q_j \phi(\left\|\underline{x}_i - \underline{x}_j\right\|), \tag{8}$$

where finite or infinite summations can be considered. In particular, $N_\phi(D)$ is defined as the completion of $H_\phi(D)$ with respect to the norm $\|\cdot\|_\phi$ induced by (8). We recall that the native space is the Reproducing Kernel Hilbert Space (RKHS) associated to ϕ.

Theorem 1 *Let $\phi : D \to \mathbb{R}$ be a continuous function and a strictly positive definite kernel. Let $\underline{x}_i \in D \subset \mathbb{R}^d$, $i = 1, 2, \ldots, N$, be pairwise distinct points, and let $f \in N_\phi(D)$. Then for every $\underline{x} \in D$ we have:*

$$|f(\underline{x}) - P_f(\underline{x})| \leq P_{\phi,X}(\underline{x}) \, \|f\|_{N_\phi}, \quad \underline{x} \in D, \tag{9}$$

where $P_{\phi,X}$ is the `power function` *which depends on the interpolation points X and on the kernel ϕ, and $\|f\|_{N_\phi}$ is the native space norm of f.*

See [16, Chap. 14] for a detailed proof of this theorem.

Theorem 1 assumes the knowledge of the exact solution c_j, $j = 1, 2, \ldots, N$ of (1). However, only an approximated solution $\tilde{c}_j$, $j = 1, 2, \ldots, N$ of (1) is usually known yielding the following error inequality

$$\left| f(\underline{x}) - \tilde{P}_f(\underline{x}) \right| \leq |f(\underline{x}) - P_f(\underline{x})| + \left| P_f(\underline{x}) - \tilde{P}_f(\underline{x}) \right|, \quad \underline{x} \in D, \tag{10}$$

where $\tilde{P}_f$ is the function (4) computed by using the approximated solution $\tilde{c}_j$, $j = 1, 2, \ldots, N$, of (1). Due to the ill-conditioning of (1), the second

addendum in the right-hand side of (10) may add a non-negligible quantity to the interpolation error.

Among the most popular methods for stabilizing the interpolant, many belong to the following categories.

1. **RBF-QR method**: it is rooted in a particular decomposition of the kernel, and it has been developed so far to treat the Gaussian kernel [20, 21, 17, 23].

2. **Hilbert-Schmidt Singular Value Decomposition (HS-SVD)**: it has been developed to stably compute the RBF interpolants [11, 18]. In principle, this technique can be applied to any kernel, provided that the HS eigenvalues and eigenvectors are known. However, these quantities are far from being easy to compute, and in practice, they only work for the Gaussian function.

3. **Weighted Singular Value Decomposition (WSVD) bases**: it is a more general approach that applies to any RBF, consisting of computing a weighted SVD which produces stable bases [12].

We recall that when we have flat or nearly flat RBFs, a severe ill-conditioning of the RBF interpolation matrix occurs, even if it generally guarantees the best accuracy for smooth solutions. The *RBF-QR method* tries to reduce the error due to the ill-conditioning when the shape parameter approaches zero, so the RBF becomes flat. In this case, the error term (9) is usually small, but the second addendum in the right-hand side of (10) is usually large as a consequence of the ill-conditioning of (1). The core idea of RBF-QR is to change the basis of the approximation space to eliminate numerical instability; this is done by expanding the RBF kernel into a series of more stable functions, such as spherical harmonics [31] or Chebyshev polynomials [32]. A QR decomposition of the representation matrix arising from these expansions can gather the ill-conditioning in simple matrices that can be inverted almost analytically. A similar idea is proposed in [33] for the Gaussian kernel, for which a Mercer expansion is considered together with the QR decomposition of the sampled eigenfunctions used in the expansion.

Stabilization of RBF interpolation is also possible using greedy techniques [40], which are iterative algorithms for selecting an optimal subset of points (centers) from a large dataset X to construct an efficient and stable interpolant. Several variants have been proposed. The *p-greedy* [41, 42] is a data-independent method that iteratively selects the next point where the power function $P_{\phi,X}$ is maximized, ensuring in this way that the points are "well-spread" and minimize the worst-case error for any function in $N_\phi(D)$. The *f-greedy* [43, 44] is a data-dependent method that selects the next point where the current interpolation residual attains the maximum value, so this is highly effective for functions with localized features like sharp gradients.

The *f/p-greedy* [43] is a hybrid selection strategy designed to maximize the benefits of both data-independent (p-greedy) and data-dependent (f-greedy) methods. The *β-greedy* [45] algorithm is a unified theoretical framework that encompasses the entire family of greedy selection strategies for RBF interpolation; it is formally introduced to study how different selection criteria affect the convergence rate and numerical stability of the interpolant.

We recall that the *Lebesgue constant*, Λ_N, is an indicator that measures the propagation of the interpolation error, indeed, we have

$$\|f - P_f\|_\infty \leq \Lambda_N E_N^*(f),$$

where $\|f - P_f\|_\infty = \max\{|f(\underline{x}) - P_f(\underline{x})|, \underline{x} \in D\}$, and as usual $E_N^*(f)$ is the sup norm error of the best polynomial approximation of f of degree at most N. It grows as the number of data points N increases, but high-quality point sets X, such as those generated by the p-greedy algorithm, exhibit slow growth of the Lebesgue constant; see [44] for details. We also note that p-greedy or their generalization known as β-greedy strategies ensure a good separation distance among points in X, preventing in this way a fast growing of the condition number of the system (1), see [45, 46] for details.

A similar result for greedy strategies is obtained in [47], where the statistical concepts of Design of Experiments are used to improve the numerical stability of the interpolation matrix and the interpolation error, thereby mitigating the *Shape Parameter Dilemma*. It is worth noting that this yields a point distribution similar to Chebyshev or Leja points, which are denser near the boundaries of the domain to avoid the *Runge effect*; moreover, a similar approach is used in the solution of partial differential equations. To open a brief discussion about the Shape Parameter Dilemma, which strongly affects both the effectiveness and the accuracy of an RBF interpolant, Figure 2 shows a trade-off in the shape parameter choice for Gaussian RBF. Both the flat limit region and the tiny variance region must be avoided due to ill-conditioning or scarce accuracy, respectively. The "optimal" choice is the central region where the condition number is limited and the interpolation error is satisfactory.

We conclude this section with some words about *Partition of Unity (PU)* techniques [34, 35], as a "trait d'union" with the next section, since these techniques are able to stabilize problem (1) but also to provide a more efficient solution of RBF interpolation by transforming a single, often ill-conditioned global problem into a series of smaller, numerically manageable local subproblems. The main components of these techniques are: **i)** the reduction of ill-conditioning by dividing the domain into overlapping subdomains (patches), where small local matrices have to be inverted with a consequent reduction of numerical instability; **ii)** local management of the problem by adapting the solution procedure to each patch, for example, by using local shape parameters; **iii)** smooth blending by compactly supported weight functions to ensure a smooth, continuous transition between

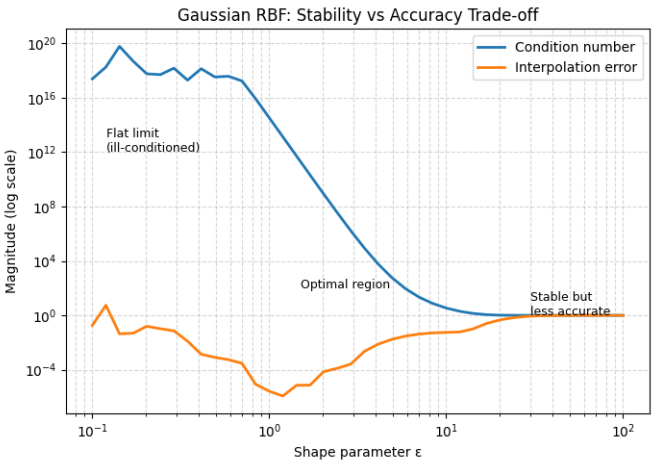


Figure 2: Gaussian function. Conditioning Vs Accuracy. The trade-off principle or the Shape Parameter Dilemma.

patches without introducing artificial oscillations. After its introduction, PU has been used in several concrete applications, such as, for example, 3D surface reconstruction from laser scans [36], Interpolation of Earth's topography data [37], and smooth transfer of aerodynamic loads from a dense fluid mesh to a sparser structural wing mesh [38].

3 Computational strategies: fast and compression

We premise that any strategy to reduce the computational cost of the RBF interpolation should be implemented with special care, especially when these strategies are based on an approximation of the RBF matrix; in fact, any small modifications of the RBF matrix can produce large perturbations in the solution of (1) as a consequence of its ill conditioning.

3.1 Classical approaches

The *Fast Multipole Method (FMM)* is exactly of this type; it is used to accelerate the evaluation of the RBF interpolant in the solution of the linear system (1). This is a dense linear system, so direct methods (like LU decomposition) require $O(N^3)$ operations, the Conjugate Gradient Method (CGM) converges in at most N iterations, because the matrix is symmetric. A single matrix-vector multiplication (in iterative methods such as CGM or GMRES or BiCGSTAB) requires $O(N^2)$ operations. So, reducing the computational cost of matrix-vector multiplication is fundamental for obtaining efficient iterative methods for solving (1). The FMM reduces the cost of matrix-vector multiplication in iterative methods by approximating

the action of the RBF matrix. This approximation uses ideas from classical scattering theory [48] and can compute matrix-vector multiplication in $O(N \log N)$ operations. In [52], a fast algorithm for the matrix-vector multiplication is obtained for the inverse multiquadric; the computational cost of this algorithm has an upper bound of $O(N \log N)$. In [49], the Chebyshev interpolation is used to replace the analytical expansion of the kernel ϕ. In [50], the analytical expansion of the kernel ϕ is calculated in terms of equivalent densities on surfaces enclosing the domain D; this approach has general validity and allows the use of different kernels ϕ. In [51], the acceleration of the iterative solver for RBFs using FMM has been provided. In [53], using a technique similar to FMM and a local low-rank representation of the interpolation matrix, a preconditioning matrix is obtained to stabilize the numerical solution of (1).

A similar method that allows both efficient matrix-vector multiplication and an explicit representation of the matrix, which can be profitably used to construct a preconditioner, is the *hierarchical representation* [54]; in this paper, different approaches to obtain low-rank and block-wise approximations for different RBF functions are analyzed. Among the various approaches considered, a Taylor-based representation undoubtedly benefits from its generality, which guarantees its applicability to any conditionally positive definite RBF.

High-performance computing software has been developed to implement FFM, such as, for example, [55] demonstrates RBF interpolation for datasets with several million points using parallel FMM and the PETSc library; [56] provides a computer procedure in the Rust language [57], combining FMM and domain decomposition for datasets with over one million points.

3.2 Samplets: a new compression strategy

A similar result can be obtained by *samplets* [58], which are a specific class of multiscale basis functions designed to compress the dense matrices that arise in large-scale problems. In particular, they have been introduced as a specialized alternative to standard wavelets, specifically tailored for scattered data and integral operators. Samplets are a specialized multiresolution analysis (MRA) technique designed for data-adapted decomposition of localized signed measures, frequently used to handle scattered (unstructured) data in arbitrary dimensions. They represent a generalization of the Tausch-White multi-wavelet framework to discrete settings, enabling data compression, feature detection, and numerical approximation [58], and we may construct them using the usual vanishing moment framework. A visual representation on a multiresolution by samplets is given in Figure 3.

This abstract setting allows us to construct them on very general domains (hence data sets). By "thresholding" (setting these small values to zero), the dense matrix becomes sparse; in this way, the matrix-vector mul-

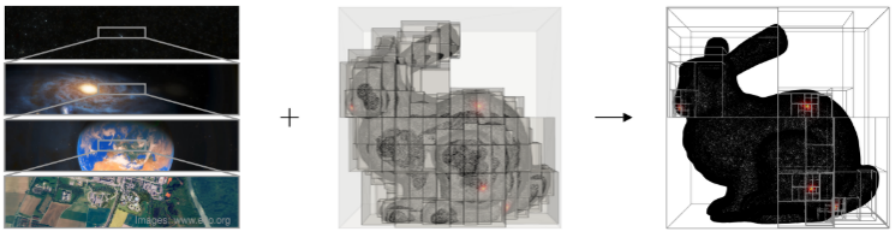


Figure 3: Left: multiresolution of the Earth from the space. Center: the bunny overlayed by samplets. Right: the sum shows how the multiresolution of samplets works. Image courtesely provided by Sara Avesani - Università della Svizzera Italiana.

tiplication requires $O(N \log N)$ operations that can be profitably used in the iterative solution of (1). The samplets framework has been extended to efficiently compute complex algebraic operations on the RBF matrix; see [59] for details. In [60], a study of sample properties for functions in $N_\phi(D)$ and some applications of RBF in sparse data interpolation and surface reconstruction are reported.

3.3 The Nyström method

The *Nyström method* provides another approximation technique to reduce the computational cost of (1). In this technique, the kernel matrix Φ is approximated using only m columns, where $m \ll N$. The selection of these landmarks from the dataset X is usually done by random sampling, clustering of K-means, or more advanced techniques such as the *Leverage Score sampling* [61]. The eigendecomposition of the selected m columns provides a low-rank representation of the kernel matrix

$$\Phi \approx \tilde{\Phi} = UEU^t, \tag{11}$$

where $U \in R^{N \times m}$, $E \in R^{m \times m}$. From (11), the Woodbury formula can be used to solve the linear system (1) efficiently. A statistical analysis and error bounds for the method are provided in [63]; moreover, a detailed analysis of the strategy to select better landmark points is proposed in [64].

A similar result is obtained by the *Structured Kernel Interpolation* (SKI) [70] that approximates the kernel matrix Φ using an interpolation on a regular grid, in this way

$$\Phi \approx \tilde{\Phi} = WKW^t, \tag{12}$$

where W is a sparse matrix containing the interpolation weights and K has a Toeplitz structure; so the solution of (1) can be obtained by iterative

methods with a nearly linear computational cost. In [71], it is proposed to improve the efficiency of SKI by a decomposition of the kernel into products of lower-dimensional structures. Moreover, [72] has provided a rigorous theoretical error bound for SKI, confirming that the approximation error decays cubically relative to the grid density when using cubic interpolation.

3.4 Other approaches

The conceptual basis of *Random Fourier Feature* (RFF) [65] is the integral Bochner Theorem, which is used to approximate the kernel ϕ by a finite number of randomized trigonometric features and the explicit mapping of the data to a low-dimensional Euclidean space. In this way, the solution of (1) is obtained as a least squares problem in a reduced number of variables. The theoretical analysis for the uniform convergence of random features is provided in [66]. Several authors have studied this method in detail, analyzing uniform error bounds [67], the approximation quality and statistical efficiency [68], and the risk of learning with RFF for different loss functions [69].

Among the various approximation techniques for RBF matrix Φ, it is worth mentioning also the so-called *Adaptive Cross Approximation* (ACA), initially proposed in the context of the solution of integral equations [73] and then generalized for RBF interpolation matrices [74]. This algorithm constructs low-rank approximations of a given matrix by selecting rows and columns, calculating the residual, and updating the matrices until a tolerance threshold is met. ACA does not use analytical expansions of the kernel function ϕ, so, in principle, it can work with different kernels even if it is developed for matrices generated by sampling smooth functions.

4 Conclusions

We conclude this review paper with some final remarks. We have tried to provide a big-picture overview of the current state of RBF approximation methods, particularly by outlining the main ideas that make this technique stable, efficient, and implementable in HPSC architectures. Indeed, these contributions have made RBF a powerful tool in high-performance computing [76, 75, 77], including machine learning applications and data analysis [78, 79, 5]. The success of RBF in easily tackling high-dimensional problems has led to the development of software libraries for high performance computing, such as, for example, RBF interpolation and RBF solution of PDE in Python [81], RBF interpolation in Julia programming language [82, 83], and in PETSc library [55], as well as parallel hardware dedicated to RBF computations [84, 85]. We hope that this short survey will serve as a useful guide to the various directions in which scientific research on the RBF approximation is developing.

Acknowledgments. We acknowledge the INdAM-GNCS group and the topical group on "Approximation Theory and Applications" of the Italian Mathematical Union and the topical group ANA&A of SIMAI.

References

[1] R.L.Hardy, Multiquadric equations of topography and other irregular surfaces. Journal of Geophysical Research, **76**(1971) 1905-1915.

[2] J. Meinguet, Multivariate interpolation at arbitrary points made simple. Zeitschrift für angewandte Mathematik und Physik (ZAMP), **30**(1979) 292-304.

[3] R. Franke, Scattered data interpolation: tests of some methods. Mathematics of Computation, **38**(1982), 181-200.

[4] C.A. Micchelli, Interpolation of scattered data: Distance matrices and conditionally positive definite functions. Constructive Approximation, **2**(1986), 11-22.

[5] Cinzia Bandiziol, Stefano De Marchi, Persistence symmetric kernels for classification: A comparative study. Symmetry, **16**(9) (2024), 1236.

[6] J. Duchon. Splines minimizing rotation-invariant semi-norms in Sobolev spaces. In: W. Schempp, K. Zeller (eds) Constructive Theory of Functions of Several Variables. Lecture Notes in Mathematics, vol 571, 1977.

[7] E.J. Kansa, Multiquadrics—A scattered data approximation scheme with applications to computational fluid-dynamics—II. Computers & Mathematics with Applications, **19**(1990) 147-161.

[8] A. de Boer, M.S. van der Schoot, H. Bijl, Mesh deformation based on radial basis function interpolation, Computers & Structures, **85**(2007) 784-795.

[9] K. Rohr, H.S. Stiehl, R. Sprengel, T.M. Buzug, J. Weese, M.H. Kuhn, Landmark-based elastic registration using approximating thin-plate splines. IEEE Transactions on Medical Imaging. **20**(2001) 526-534.

[10] D.S. Broomhead, D. Lowe, Multivariable functional interpolation and adaptive networks. Complex Systems, **2**(1988) 321-355.

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

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

[13] J. Moody, C.J. Darken, Fast learning in networks of locally-tuned processing units. Neural Computation **1**(1989) 281-294.

[14] T. Poggio, F. Girosi, Networks for approximation and learning. Proceedings of the IEEE, **78**(1990) 1481-1497.

[15] B.E. Boser, I.M. Guyon, V.N. Vapnik, A training algorithm for optimal margin classifiers COLT '92: Proceedings of the fifth annual workshop on Computational learning theory (1992) 144-152.

[16] G.E. Fasshauer, Meshfree Approximation Methods with MATLAB. Interdisciplinary Mathematical Sciences, Vol. 6, 2007, World Scientific Publishers.

[17] G.E. Fasshauer and M. Mc Court, Stable evaluation of Gaussian radial basis function interpolants, SIAM J. Sci. Comput., 34 (2012), pp. A737–A762.

[18] G.E. Fasshauer and M. Mc Court, Kernel-based Approximation Methods with MATLAB. Interdisciplinary Mathematical Sciences, Vol. 19, 2015, World Scientific Publishers.

[19] B. Fornberg, G. Wright, Stable computation of multiquadric interpolants for all values of the shape parameter. Computers & Mathematics with Applications **48**(2004) 853-867

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

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

[22] V. Eijkhout, E. Chow, R. van de Geijn, Introduction to High Performance Scientific Computing, CC BY 3.0, 3rd edition 2020

[23] K. Kormann, C. Lasser, and A. Yurova, *Stable interpolation with isotropic and anisotropic Gaussians using Hermite generating function*, SIAM J. Sci. Comput., **41** (2019), pp. A3839–A3859.

[24] F.J. Narcowich, J.D. Ward, Norm of inverses and condition numbers for matrices associated with scattered data. J. Approx. Theory **64**(1991) 69–94.

[25] F.J. Narcowich, J.D. Ward, Norms of inverses for matrices associated with scattered data, in:Curves and Surfaces, eds. P.J. Laurent, A. Le Méhauté and L.L. Schumaker (Academic Press, Boston, 1991) pp. 341–348.

[26] F.J. Narcowich, J.D. Ward, Norm estimates for the inverses of a general class of scattered-data radial-function interpolation matrices, J. Approx. Theory **69** (1992) 84–109.

[27] K. Ball, N. Sivakumar, J.D. Ward, On the sensitivity of radial basis interpolation to minimal data separation distance, Constr. Approx. **8** (1992) 401–426.

[28] K. Ball, Eigenvalues of Euclidean distance matrices. J. Approx. Theory **68**(1992) 74-82.

[29] M.L. Braun, Accurate Error Bounds for the Eigenvalues of the Kernel Matrix. Journal of Machine Learning Research **7** (2006) 2303-2328.

[30] R. Schaback, Error estimates and condition numbers for radial basis function interpolation. Adv Comput Math **3**(1995) 251–264.

[31] B. Fornberg and C. Piret, A stable algorithm for flat radial basis functions on a sphere, SIAM Journal on Scientific Computing, **30** (2007) 60–80.

[32] B. Fornberg, E. Larsson, N. Flyer. Stable computations with gaussian radial basis functions. SIAM Journal on Scientific Computing, **33**(2011) 869–892.

[33] G.E. Fasshauer, M.J. McCourt, Stable evaluation of Gaussian radial basis function interpolants, IAM Journal on Scientific Computing, **34**(2012) A737–A762.

[34] I. Babuška, J.M. Melenk , The partition of unity method. International Journal for Numerical Methods in Engineering, **40**(1997) 727-758.

[35] H. Wendland, "Fast evaluation of radial basis functions: Methods based on partitions of unity". Approximation Theory X: Wavelets, Splines, and Applications, 2002.

[36] Y. Ohtake, A. Belyaev, M. Alexa, G. Turk, H.-P. Seidel, Multi-level partition of unity implicits, ACM Transactions on Graphics, **22**(2003) 463-470.

[37] R. Cavoretto, A. De Rossi, A meshless interpolation algorithm using a cell-based searching procedure,Computers & Mathematics with Applications, **67**(2014) 1024-1038.

[38] A. Beckert, H. Wendland, Multivariate interpolation for fluid-structure-interaction using radial basis functions. Aerospace Science and Technology, **5**(2001) 125-134.

[39] S. De Marchi, R. Schaback, H. Wendland, Near-Optimal Data-independent Point Locations for Radial Basis Function Interpolation. Adv. Comput. Math., **23**(2005) 317-330.

[40] R. Schaback, H. Wendland, Adaptive greedy techniques for approximate solution of large RBF systems. Numerical Algorithms, **24**(2000) 239–254 .

[41] S. Müller, "Adaptive fitting of scattered data by locally customized kernel bases", Foundations of Computational Mathematics, Eds. R.A. DeVore, A. Iserles e E. Süli, 2002, 233–254. Cambridge University Press. (CONTROLLARE)

[42] S. De Marchi, R. Schaback, H. Wendland, Near-optimal search of centers for RBF prediction models, Computers & Mathematics with Applications, Volume **50**(2005) 369–380.

[43] S. Müller, Komplexität und Stabilität von kernbasierten Rekonstruktionsmethoden. PhD thesis, University of Göttingen, (2009)

[44] S. De Marchi, Geometric greedy and greedy points for RBF interpolation. Proceedings of the International Conference on Computational and Mathematical Methods in Science and Engineering, 30 June, 1–3 July 2009.

[45] T. Wenzel, G. Santin, B. Haasdonk, Analysis of Target Data-Dependent Greedy Kernel Algorithms. Constructive Approximation, **55**(2022) 871-911.

[46] G. Santin, B. Haasdonk, Convergence rate of the data-independent P-greedy algorithm in kernel-based approximation. Dolomites Research Notes on Approximation, **10**(2017) 68-78.

[47] E. Larsson, R. Schaback, A Design of Experiments Approach to Point Selection in Radial Basis Function Methods, Engineering Analysis with Boundary Elements, **31**(2007) 64–72.

[48] L.F. Greengard, V. I. Rokhlin, A Fast Algorithm for Particle Simulation. Journal of Computational Physics **73**(2001) 325-348.

[49] W. Fong, E. Darve, The black-box fast multipole method. Journal of Computational Physics **228**(2009) 8712–8725.

[50] L. Ying, G. Biros, D. Zorin, A kernel-independent adaptive fast multipole algorithm in two and three dimensions. Journal of Computational Physics **196**(2004) 591–626.

[51] N. Gumerov, R. Duraiswami, Fast Radial Basis Function Interpolation via Preconditioned Krylov Iteration. SIAM Journal on Scientific Computing **29**(2007) 1876-1899.

[52] S. De Marchi, N. Egidi, J. Giacomini, P. Maponi, A. Perticarini, Computational issues by interpolating with inverse multiquadrics: a solution. Dolomites Research Notes on Approximation **15**(2022) 56-64.

[53] N. Egidi, J. Giacomini, P. Maponi, Preconditioning strategies for RBF interpolation. Yaroslav D. Sergeyev, Dmitri E. Kvasov, and Annabella Astorino, editors, Numerical Computations: Theory and Algorithms, Springer Nature, Switzerland, (2025) 246–253.

[54] A. Iske, S. Le Borne, and M. Wende, Hierarchical matrix approximation for kernel-based scattered data interpolation. SIAM journal on scientific computing **39(5)**(2017) A2287–A2316.

[55] R. Yokota, L.A. Barba, M.G. Knepley, PetRBF — A parallel O(N) algorithm for radial basis function interpolation with Gaussians. Computer Methods in Applied Mechanics and Engineering **199**(2010) 1793-1804.

[56] D. Owen, ferreus_rbf, Maptek Pty Ltd, 2025. Accessed March 12, 2026. `https://docs.rs/ferreus_rbf/latest/ferreus_rbf/#references`

[57] The Rust Project Developers. "The Rust Programming Language." Accessed March 12, 2026. `https://www.rust-lang.org/`

[58] H. Harbrecht, M. Multerer, Samplets: Construction and scattered data compression. Journal of Computational Physics **471**(2022) 111616.

[59] H. Harbrecht, M. Multerer, O. Schenk, Ch. Schwab, Multiresolution kernel matrix algebra. Numerische Mathematik **156**(2024) 1085–1114.

[60] D. Baroli, H. Harbrecht, M. Multerer, Samplet Basis Pursuit: Multiresolution Scattered Data Approximation With Sparsity Constraints. IEEE Transactions on Signal Processing, **72**(2024) 1813-1823.

[61] P. Drineas, M.W. Mahoney, S. Muthukrishnan, Sampling algorithms for ℓ_2 regression and applications. In Proceedings of the 17th Annual ACM-SIAM Symposium on Discrete Algorithms, pages 1127–1136, 2006

[62] C.K.I. Williams, M. Seeger, Using the Nyström method to speed up kernel machines. In Advances in Neural Information Processing Systems (2001) 682-688

[63] P. Drineas, M.W. Mahoney (2005). On the Nyström method for approximating a Gram matrix for improved kernel-based learning. Journal of Machine Learning Research 6 (2005) 2153–2175.

[64] K. Zhang, I.W. Tsang, J.T. Kwok, Improved Nyström low-rank approximation and error analysis. In Proceedings of the 25th International Conference on Machine Learning, Helsinki, Finland, 2008.

[65] A. Rahimi, B. Recht, Random Features for Large-Scale Kernel Machines Advances in Neural Information Processing Systems 20(2007) 1177–1184

[66] A. Rahimi, B. Recht, Uniform approximation of functions with random bases, In Proceedings of the46th Annual Allerton Conference on Communication, Control, and Computing, 2008, 555–561.

[67] D.J. Sutherland, J. Schneider, On the Error of Random Fourier Features. Proceedings of the 31st Conference on Uncertainty in Artificial Intelligence (UAI), 2015, 862–871.

[68] B.K. Sriperumbudur, Z. Szabó, Optimal rates for Random Fourier features. Proceedings of the 29th International Conference on Neural Information Processing Systems - Volume 1, 2015, 1144–1152.

[69] Z. Li, J. Ton, D. Oglic, D. Sejdinovic, Towards a Unified Analysis of Random Fourier Features. Journal of Machine Learning Research, **22**(2021) 1–51.

[70] A.G. Wilson, H. Nickisch, Hannes, Kernel interpolation for scalable structured Gaussian processes (KISS-GP). In Proceedings of the 32nd International Conference on Machine Learning - Volume 37, 2015, 1775–1784.

[71] J. Gardner, G. Pleiss, R. Wu, K. Weinberger, A. Wilson, Product Kernel Interpolation for Scalable Gaussian Processes. In Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics, 2018, 1407-1416.

[72] A. Moreno, J. Xiao, J. Mei, The Price of Linear Time: Error Analysis of Structured Kernel Interpolation. In Proceedings of the Forty-Second International Conference on Machine Learning, 2025.

[73] M. Bebendorf, Approximation of boundary element matrices. Numerische Mathematik **86**(2000) 565–589.

[74] R. Grzhibovskis, M. Bambach, S. Rjasanow, G. Hirt, Adaptive cross-approximation for surface reconstruction using radial basis functions. Journal of Engineering Mathematics **62**(2008) 149–160.

[75] M.E. Biancolini, Fast Radial Basis Functions for Engineering Applications, 2018. Springer International Publishing.

[76] B. Fornberg, N. Flyer, A Primer on Radial Basis Functions with Applications to the Geosciences. Society for Industrial and Applied Mathematics, 2015.

[77] J.-P. Li, Q.-H. Qin, Radial Basis Function Methods For Large-Scale Wave Propagation, 2021. Bentham Science publishers.

[78] R. Schaback, H. Wendland, Kernel techniques: from machine learning to meshless methods Acta numerica 15(2006) 543-639

[79] S.S. Haykin, Neural Networks: A Comprehensive Foundation (2nd ed.), 1998. Prentice Hall.

[80] N. Cristianini, J. Shawe-Taylor, An introduction to Support Vector Machines and other kernel-based learning methods, 2000. Cambridge University Press.

[81] T. Hines, RBF: A Python package for radial basis functions, 2022. GitHub.

[82] J. Crean, RadialBasisFunctions.jl: Radial basis functions for JuliaJ, 2024. GitHub.

[83] T.-A. Bird, RadialBasisFiniteDifferences.jl: High-performance RBF-generated finite differences in Julia, 2024. GitHub.

[84] General Vision Inc. (2017). CM1K - NeuroMem Chip with 1024 Neurons Datasheet.

[85] SK hynix Inc. (2022). SK hynix Develops PIM, Next-Generation AI Accelerator. SK hynix Newsroom