REVISTA DE LA UNIÓN MATEMÁTICA ARGENTINA
Vol. 64, No. 2, 2023, Pages 375–396
Published online: April 25, 2023
https://doi.org/10.33044/revuma.2901

STABILIZING RADIAL BASIS FUNCTIONS TECHNIQUES FOR A LOCAL BOUNDARY INTEGRAL METHOD

LUCIANO PONZELLINI MARINELLI

Abstract. Radial basis functions (RBFs) have been gaining popularity recently in the development of methods for solving partial differential equations (PDEs) numerically. These functions have become an extremely effective tool for interpolation on scattered node sets in several dimensions. One key issue with infinitely smooth RBFs is the choice of a suitable value for the shape parameter ε, which controls the flatness of the function. It is observed that best accuracy is often achieved when ε tends to zero. However, the system of discrete equations from interpolation matrices becomes ill-conditioned. A few numerical algorithms have been presented that are able to stably compute an interpolant, even in the increasingly flat basis function limit, such as the RBF-QR method and the RBF-GA method. We present these techniques in the context of boundary integral methods to improve the solution of PDEs with RBFs. These stable calculations open up new opportunities for applications and developments of local integral methods based on local RBF approximations. Numerical results for a small shape parameter that stabilizes the error are presented. Accuracy and comparisons are also shown for elliptic PDEs.

1. Introduction

Radial basis functions (RBF) collocation methods were first developed by R. Hardy [23] for multivariate scattered data interpolation. R. Franke showed the benefits of using RBFs for interpolations in several numerical experiments [22]. In his pioneer works, E. J. Kansa used RBF meshless methods to find the numerical solution of PDEs to a wide range of problems [24, 25]. R. Franke conjectured the inversibility of the interpolation matrix with multiquadric RBF and C. Micchelli proved this conjecture improving and accelerating the use of RBFs in many fields [27]. These meshless methods can easily handle high-dimensional irregular domains and are rather easy to implement compared to mesh-based methods such as the finite element method.

2020 *Mathematics Subject Classification.* 15A12, 65D12, 65N12, 65N35, 65R20.

Key words and phrases. Meshless method, radial basis functions, local integral method, RBF-GA, RBF-QR.

Work partially supported by Secretaría de Ciencia y Tecnología, Facultad de Ciencias Exactas, Ingeniería y Agrimensura, Universidad Nacional de Rosario, and Centro Internacional Franco Argentino de Ciencias de la Información y de Sistemas (CIFASIS-CONICET-UNR).

	Name of RBF	**Abbreviation**
Smooth $\phi(r,\varepsilon)$		
$e^{-(\varepsilon r)^2}$	Gaussian	GA [2, 28, 16, 31]
$\sqrt{1+(\varepsilon r)^2}$	Multiquadric	MQ [1, 4, 6, 25]
$\left(1+(\varepsilon r)^2\right)^{\beta}$, $\beta \in \mathbb{R}\backslash\mathbb{N}_0$	Generalized Multiquadric	GMQ [19]
$1/\sqrt{1+(\varepsilon r)^2}$	Inverse Multiquadric	IMQ [6, 12, 19, 34]
Piecewise smooth $\phi(r)$		
r^{β}, $\beta \notin 2\mathbb{N}$	Radial Potential	RP [30]
$r^{2\beta}\log(r)$, $\beta \in \mathbb{N}$	Thin Plate Spline	TPS [4, 32, 33]
$r^{2\beta-1}$ or $r^{2\beta}\log(r)$, $\beta \in \mathbb{N}$	Polyharmonic Spline	PHS [3, 11, 28, 29]

Table 1. Some well-known RBFs; ε is the shape parameter.

There are many choices of RBFs $\phi(r)$, where $r = \|\mathbf{x}\|$ and ε is the shape parameter, which can be used as shown in Table 1 (see more examples in the books [10, 13]). The references in the last column of the table indicate some PDEs solved numerically for each type of RBF. In this work, we focus on the use of strictly positive definite Gaussian RBFs that depend on this ε.

Theoretically, the use of infinitely smooth radial kernels like the Gaussian RBF for interpolations gives a spectrally convergent meshless method [8]. In [20] it was shown that this RBF interpolant converges to a polynomial interpolant when ε tends to zero. In [26] the behavior of this interpolant in the limit case was studied both theoretically and numerically to achieve higher accuracy. Therefore the practical use of a small range of the shape parameter is recommended to improve the method's performance. In numerical implementations, when the shape parameter ε is small the RBFs become almost linearly dependent ('flat RBFs') giving rise to an ill-conditioned system. The rows in the interpolation matrix are almost equal, thus giving rise to a numerically ill-conditioned problem. In 1995 R. Schaback formulated this issue as the so-called uncertainty principle [35]. This principle has contributed to a mistaken conception in which the flat RBFs always lead to a numerical ill-conditioned RBF linear system assuming that the RBF interpolants are computed by solving by a standard RBF approach (often denoted as 'RBF-Direct'). The RBF interpolation problem is not itself a numerical ill-conditioned problem in the limit case. The ill-conditioning issue is specific to this RBF-Direct approach. So, several well-conditioned numerical algorithms have been developed in the last decades.

The RBFs make up an ill-conditioned basis in a good approximation space, so the purpose of an RBF stable algorithm is to stably evaluate the interpolant when

ε decreases improving the accuracy. Several stabilizing techniques have been presented so far in the literature to bypass this ill-conditioning and allow a complete range of ε that could be useful. B. Fornberg and co-authors made several contributions in this sense [15, 17, 18, 21, 36]. The RBF-QR was developed for the sphere in [18] and similar ideas followed for the RBF-QR in cartesians in [15]. The Gauss-QR method was proposed by Fasshauer [9] for the Gaussian kernels in connection to Hilbert–Schmidt (Mercer) series expansions of positive definite kernels. The RBF-GA presented in [17] avoids the calculation of truncated infinite series and uses the efficient numerical implementation of the incomplete gamma function. Another recent development is the RBF with rational approximation (RBF-RA) [36] following the ideas of the Contour–Padé (C-Padé) algorithm in [21].

Applications of the RBF-QR to Stefan problems were presented in [6]. In [2] this algorithm was used in the method of fundamental solutions for boundary value problems and in [7] for the numerical calculation of surface divergence-free RBFs on the sphere. In [31] this technique improved the results for local integral methods applied to elliptic PDEs in the limit case. However, there is no application of the RBF-GA method to local integral methods that use RBF interpolation at a local level and $\varepsilon \to 0$ to increase accuracy.

The local integral methods that use RBFs as local interpolants have been established as robust methods in the last decades. The appearance of the stable algorithms mentioned above shows that the RBF interpolation operator could be stabilized achieving accurate results, as presented in the Local Boundary-Domain Integral Method (LBDIM) with RBF-QR by the author.

In this paper a novel local integral method is presented. The LBDIM uses the RBF-GA method for strictly positive definite Gaussian RBFs to produce an interpolation matrix with acceptable condition number and stabilizes the errors. The new approach is tested for many differential problems. Its strengths and weaknesses are compared with RBF-FD and LBDIM for some numerical tests.

The paper is organized as follows. In Section 2 we provide a background in RBFs in the perspective for solving PDEs numerically. In Section 3 we introduce the LBDIM approach inspired by [31] and the stabilizing techniques to local interpolations to reduce the ill-conditioning in the local RBF linear systems. In Section 4 we provide numerical results for a Poisson PDE over a circular domain and for two linear elliptic PDEs over a square and an irregular domain. Some concluding remarks are presented in Section 5.

2. Background in RBFs

Definition 2.1 ([8])**.** A function $\varphi : \mathbb{R}^d \to \mathbb{R}$ is an *RBF* when there exists a univariate function $\phi : [0, \infty) \to \mathbb{R}$ such that

$$\varphi(\mathbf{x}) = \phi(r), \qquad r = \|\mathbf{x}\|, \tag{2.1}$$

where $\|\cdot\|$ is some norm on $\mathbb{R}^d$ (usually the Euclidean norm). When the RBF depends on a center $\mathbf{x}_0 \in \mathbb{R}^d$ we write $\varphi(\mathbf{x}) = \phi(r)$, where $r = \|\mathbf{x} - \mathbf{x}_0\|$ in (2.1).

The *Gaussian RBF* is defined as

$$\phi(r) = e^{-(\varepsilon r)^2}, \tag{2.2}$$

where ε is known as the *shape parameter*. In Figure 1 the Gaussian RBF in two dimensions is shown in the cases $\varepsilon = 2, 1, 0.5, 0.1$ over the square domain $[-2, 2] \times [-2, 2]$.

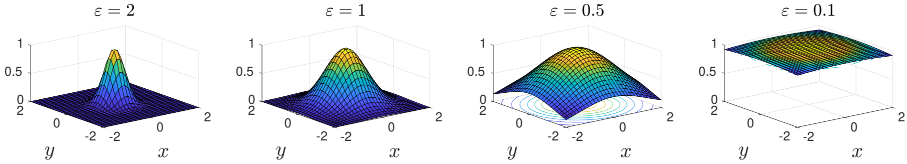

FIGURE 1. Behaviour of Gaussian RBFs $\phi(r) = e^{-(\varepsilon r)^2}$ when $\varepsilon \to 0$.

Let $u : \Omega \to \mathbb{R}^d$. An *RBF interpolant* P_u with the given data $\Theta = \{\mathbf{x}_j\}_{j=1}^n$, $\mathbf{x}_j \in \mathbb{R}^d$, is defined as

$$u(\mathbf{x}) \approx P_u(\mathbf{x}) = \sum_{j=1}^{n} \alpha_j \varphi_j(\mathbf{x}) = \sum_{j=1}^{n} \alpha_j \phi(\|\mathbf{x} - \mathbf{x}_j\|), \tag{2.3}$$

where the unknown coefficients α_j for $j = 1, \ldots, n$ are given by imposing the collocation conditions

$$u_i = P_u(\mathbf{x}_i) = \sum_{j=1}^{n} \alpha_j \varphi_j(\mathbf{x}_i), \quad i = 1, \ldots, n.$$

In matrix form the system could be written as

$$\begin{bmatrix} \phi(\|\mathbf{x}_1 - \mathbf{x}_1\|) & \phi(\|\mathbf{x}_1 - \mathbf{x}_2\|) & \ldots & \phi(\|\mathbf{x}_1 - \mathbf{x}_n\|) \\ \phi(\|\mathbf{x}_2 - \mathbf{x}_1\|) & \phi(\|\mathbf{x}_2 - \mathbf{x}_2\|) & \ldots & \phi(\|\mathbf{x}_2 - \mathbf{x}_n\|) \\ \vdots & \vdots & \ddots & \vdots \\ \phi(\|\mathbf{x}_n - \mathbf{x}_1\|) & \phi(\|\mathbf{x}_n - \mathbf{x}_2\|) & \ldots & \phi(\|\mathbf{x}_n - \mathbf{x}_n\|) \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_n \end{bmatrix} = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{bmatrix}.$$

Taking $[\mathbf{\Phi}]_{ij} = \varphi_j(\mathbf{x}_i) = \phi(\|\mathbf{x}_i - \mathbf{x}_j\|)$, $\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_n]^T$, and $\boldsymbol{u} = [u_1, \ldots, u_n]^T$,

$$\mathbf{\Phi} \alpha = \boldsymbol{u}, \tag{2.4}$$

where we need $\mathbf{\Phi}$ to be invertible [27].

The RBF collocation method represents the approximate solution as a weighted sum of the RBFs $\varphi_j(\mathbf{x}) = \phi(\|\mathbf{x} - \mathbf{x}_j\|_2)$ where the unknown expansion coefficients are determined through solving the linear system (2.4). This is known as *RBF-Direct*.

Definition 2.2 ([8])**.** A continuous function $\varphi : \mathbb{R}^d \to \mathbb{C}$ is *positive definite* on $\mathbb{R}^d$ if and only if

$$\sum_{j=1}^{n} \sum_{k=1}^{n} c_j \overline{c_k}\, \varphi(\mathbf{x}_j - \mathbf{x}_k) \geq 0 \tag{2.5}$$

for every distinct node set $\chi = \{\mathbf{x}_1, \dots, \mathbf{x}_n\} \subset \mathbb{R}^d$ and for every $\mathbf{c} = [c_1, \dots, c_n] \in \mathbb{C}^d$, where $\overline{c_k}$ denotes the conjugate of c_k. The function φ is *strictly positive definite* on $\mathbb{R}^d$ if the quadratic form (2.5) is zero only when $\mathbf{c} = \mathbf{0}$.

Since positive definite functions satisfy the condition $\varphi(-\mathbf{x}) = \overline{\varphi(\mathbf{x})}$, it is necessary that φ be an even function. Also, the non-negativity of the quadratic form is needed. It is also possible to characterize strictly positive definite functions φ as every continuous function from the Lebesgue space $L_1(\mathbb{R}^d)$ that is bounded and whose Fourier transform is non-negative and not identically zero [8].

A standard criterion to measure the numerical stability of an approximation method is the condition number. The linear system of equations is ill-conditioned for the Gaussian RBFs defined in (2.2). The condition number of the interpolation matrix $\mathbf{\Phi}$, denoted as $\kappa(\mathbf{\Phi})$, increases fast indicating that the problem is ill-posed. For strictly positive definite RBFs, the condition number can be calculated as (see [8])

$$\kappa(\mathbf{\Phi}) = \frac{\lambda_{\max}}{\lambda_{\min}},$$

where $\lambda_{\min} > 0$ and $\lambda_{\max} > 0$ denote the smallest and the largest eigenvalues of the interpolation matrix $\mathbf{\Phi}$, which are positive since the matrix is symmetric, $[\mathbf{\Phi}]_{ij} = \phi(\|\mathbf{x}_i - \mathbf{x}_j\|)$.

For the strictly positive definite Gaussian RBF $\phi(r) = e^{-r^2\varepsilon^2}$, it is shown that $\phi(r) \leq \phi(0) = 1$. Since, for any matrix, the spectral radius is less than any matrix norm, $\rho(\mathbf{\Phi}) \leq \|\mathbf{\Phi}\|$, we have that $\lambda_{\max}$ is bounded as

$$\lambda_{\max} \leq \|\mathbf{\Phi}\|_\infty \leq n\varphi(0) = n.$$

That is, the largest eigenvalue is of order $O(n)$, where n is the number of centers of the interpolation, so $\lambda_{\max}$ approaches to infinity as n grows. If n is not too big, $\lambda_{\max}$ is acceptable, especially when the dimension of the interpolation problem, d, is not too large. Hence,

$$\kappa(\mathbf{\Phi}) \leq \frac{n}{\lambda_{\min}},$$

and therefore a lower bound for eigenvalues is needed to establish a bound for the condition number of $\mathbf{\Phi}$. The separation distance of data centers is defined as

$$q_\Theta = \frac{1}{2} \min_{i \neq j} \|\mathbf{x}_i - \mathbf{x}_j\|_2.$$

Using this distance, a lower bound for $\lambda_{\min}$ in the case of the Gaussian RBFs that depend on the shape parameter ε is

$$\lambda_{\min} \geq \frac{C_d e^{-40.71 d^2/(\varepsilon q_\Theta)^2}}{\left(\sqrt{2}\varepsilon q_\Theta\right)^d},$$

where the constant C_d depends on the problem dimension d as (see [8])

$$C_d = \frac{1}{2\Gamma\left(\frac{d+2}{2}\right)} \left(\frac{M_d}{\sqrt{8}}\right)^d, \quad M_d = 12 \left(\frac{\pi \Gamma^2\left(\frac{d+s}{2}\right)}{9}\right)^{\left(\frac{1}{d+1}\right)},$$

Γ being the gamma function.

In the case of infinitely smooth functions, such as the Gaussian RBFs, we see that, for a fixed number of nodes or at least for constant q_Θ, reducing the value of ε which improves interpolation accuracy, the lower bound for $\lambda_{\min}$ goes exponentially to zero and therefore the condition number of the interpolation matrix $\mathbf{\Phi}$ grows exponentially. This is known as the stationary approximation setting. Numerically, the rows in the interpolation matrix become similar so the matrix becomes almost singular, giving rise to ill-conditioning at the cost of numerical instability. The upper bound for the error estimate increases, i.e., the accuracy of the interpolant deteriorates. On the other hand, in the non-stationary approximation, ε is fixed and adding more interpolation nodes in order to improve interpolation accuracy makes the problem become increasingly ill-conditioned due to the decrease of the separation distance. This was presented as the uncertainty principle in [35].

2.1. **The Gaussian RBF expansion.** The Gaussian RBF $\varphi(r) = e^{-\varepsilon^2 r^2}$, centered at $\mathbf{x}_j \in \mathbb{R}^d$, that depends on the shape parameter $\varepsilon > 0$ is

$$\begin{aligned}\phi_j(\mathbf{x}) = \varphi_j(\|\mathbf{x}-\mathbf{x}_j\|) &= e^{-\varepsilon^2\|\mathbf{x}-\mathbf{x}_j\|^2}\\ &= e^{-\varepsilon^2\langle(\mathbf{x}-\mathbf{x}_j)^T,\,\mathbf{x}-\mathbf{x}_j\rangle}\\ &= e^{-\varepsilon^2\langle\mathbf{x}^T,\,\mathbf{x}\rangle}e^{-\varepsilon^2\langle\mathbf{x}_j^T,\mathbf{x}_j\rangle}e^{2\varepsilon^2\langle\mathbf{x}^T,\,\mathbf{x}_j\rangle}.\end{aligned} \tag{2.6}$$

The expansion of the Gaussian in 2D centered at $(x_j, y_j) \in \mathbb{R}^2$ from (2.6) is

$$\phi_j(x,y) = e^{-\varepsilon^2\left(x^2+y^2\right)}e^{-\varepsilon^2\left(x_j^2+y_j^2\right)}e^{2\varepsilon^2(xx_j+yy_j)},$$

where we stay in the same space ignoring the scalars $e^{-\varepsilon^2\left(x_j^2+y_j^2\right)}$, and considering the basis function

$$\tilde{\phi}_j(x,y) = e^{-\varepsilon^2\left(x^2+y^2\right)}e^{2\varepsilon^2(xx_j+yy_j)}, \tag{2.7}$$

with the last factor expressed using the Taylor expansion

$$e^{2\varepsilon^2(xx_j+yy_j)} = \sum_{k=0}^{\infty}\frac{(2\varepsilon^2)^k}{k!}(xx_j+yy_j)^k, \quad \varepsilon > 0. \tag{2.8}$$

Keeping the term $e^{-\varepsilon^2(x^2+y^2)}$ in (2.7), the Gaussian RBF depends on the basis

$$e^{-\varepsilon^2(x^2+y^2)}\{\{1\},\{x,y\},\{x^2,xy,y^2\},\{x^3,x^2y,xy^2,y^3\},\ldots\}. \tag{2.9}$$

Defining $z = 2\varepsilon^2\,(xx_j + yy_j) = 2\varepsilon^2\langle\mathbf{x},\mathbf{x}_j\rangle$, the remainder can be expressed as

$$G_k(z) = e^z - \sum_{j=0}^{k-1}\frac{z^j}{j!} = \frac{e^z}{(k-1)!}\int_0^z e^{-t}t^{k-1}\,dt = \frac{e^z}{(k-1)!}\gamma(k,z), \tag{2.10}$$

where $\gamma(k,z)$ is the inferior incomplete gamma function. We will continue this idea in the next section.

3. Stabilizing techniques for a local integral method

3.1. **The LBDIM-GA method.** Let us consider the following elliptic problem on a bounded open domain Ω:

$$\begin{cases} \mathcal{L}\left[u(\mathbf{x})\right] = f(\mathbf{x}), & \mathbf{x} \in \Omega, \\ \mathcal{B}\left[u(\mathbf{x})\right] = g(\mathbf{x}), & \mathbf{x} \in \Gamma = \partial\Omega, \end{cases} \tag{3.1}$$

where $\mathcal{L}[\cdot]$ is an elliptic operator and $\mathcal{B}[\cdot]$ is a classical boundary operator related to different kinds of boundary conditions (Dirichlet, Neumann or Robin).

We assume that the partial differential equation can be written as

$$\Delta u\left(\mathbf{x}\right) = b\left(\mathbf{x}, u(\mathbf{x}), \nabla u(\mathbf{x})\right), \tag{3.2}$$

and also in this work we consider that equation (3.2) can be expressed as

$$b\left(\mathbf{x}, u(\mathbf{x}), \nabla u(\mathbf{x})\right) = f(\mathbf{x}) + \widetilde{b}\left(u(\mathbf{x}), \nabla u(\mathbf{x})\right), \tag{3.3}$$

where the source function f is known and $\tilde{b}$ is linear in the variables u and ∇u.

According to the integral representation formula presented in [31] for the PDE (3.3), using the corresponding Green's second identity and the well-known Dirichlet Green's function (DGF), we have

$$u\left(\xi\right) = \int_{\Gamma_i} Q\left(\mathbf{x}, \xi\right) u(\mathbf{x})\, d\Gamma_i + \int_{\Omega_i} \tilde{b}\; G\left(\mathbf{x}, \xi\right) d\Omega_i + \int_{\Omega_i} f\left(\mathbf{x}\right) G\left(\mathbf{x}, \xi\right) d\Omega_i, \tag{3.4}$$

where ξ is the interior source point, also referred to as the collocation point, $G\left(\mathbf{x}, \xi\right)$ is the DGF on the circular integral subdomain or subregion Ω_i, and $Q\left(\mathbf{x}, \xi\right)$ its corresponding normal derivative. By definition, over the boundary Γ_i the value of $G\left(\mathbf{x}, \xi\right)$ is identically zero.

This approach is the main idea of the meshless boundary-domain integral method used in this work. Formula (3.4) is applied over each local integral subdomain Ω_i, embedded into interpolation stencils Θ_i, which are heavily overlapped as shown in Figure 2. In this figure we have a schematic representation of local stencils and local circular subdomains in two dimensions for different local integral formulations.

The field variable $u(\mathbf{x})$ is approximated by an RBF basis using the corresponding nodes of the interpolation stencil, plus some boundary points if the stencil is next to the global boundary Γ. The set $\{(\mathbf{x}_j, u(\mathbf{x}_j))\}_{j=1}^{n_i}$ is formed by the internal nodes $\mathbf{x}_j$ and the corresponding unknown nodal values $u(\mathbf{x}_j)$ for $j = 1, \ldots, n_i$, as in the case of Θ_i in Figure 2. The other set $\{(\mathbf{x}_j, g\left(\mathbf{x}_j\right))\}_{j=n_i+1}^{n}$ has the boundary nodes $\mathbf{x}_j$ and the boundary data $g\left(\mathbf{x}_j\right)$ for $j = n_i + 1, \ldots, n$ with $n = n_i + n_b$ the total number of the local stencils. These are the cases of Θ_j and Θ_k shown in Figure 2.

Thus the local approximation is given by

$$u(\mathbf{x}) = \sum_{j=1}^{n} \alpha_j \varphi_j(\mathbf{x}), \tag{3.5}$$

where α_j are the coefficients from the interpolation, $\varphi_j(\mathbf{x}) = \phi\left(\|\mathbf{x} - \mathbf{x}_j\|\right)$ the RBF interpolants, and $n = n_i + n_b$, as mentioned before.

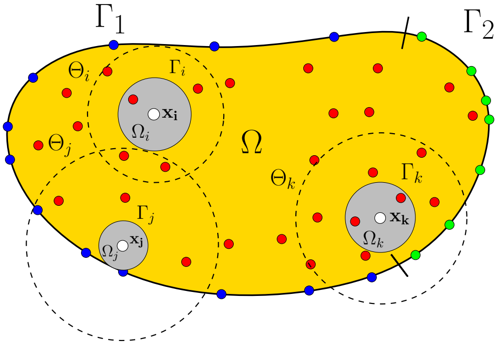

FIGURE 2. Schematic representation of the local stencils Θ_i, Θ_j, Θ_k (nodes inside the dotted line) and the local circular subdomains Ω_i, Ω_j, Ω_k with boundary Γ_i, Γ_j, Γ_k in two dimensions for local integral meshless methods.

When the interpolation stencil Θ_i has nodes on the contour Γ of the global domain Ω, we consider interpolation points on the boundary for the stencil. In Figure 3 different cases of stencils and their respective subregions of integration Ω_i are shown: in case (a), when they are totally internal to the domain; and in cases (b) and (c), when the stencils have contact with the boundary of the global domain. In this last case we include data points on the contour with the boundary conditions imposed on the boundary value problem under study. Case (b) shows a stencil near a smooth part of Γ, and case (c) near an angular point at the boundary.

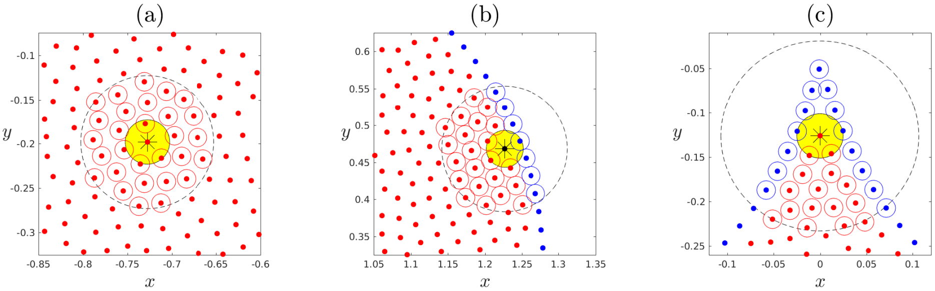

FIGURE 3. Local stencils and integration subdomains for $n = 30$.

In cases (b) and (c) from Figure 3, the RBF interpolant (2.3) is a Hermite-based approach (see [8]). The field is approximated as

$$u(\mathbf{x}) \approx P_u(\mathbf{x}) = \sum_{j=1}^{n_i} \alpha_j \phi\left(\|\mathbf{x}-\mathbf{x}_j\|\right) + \sum_{j=n_i+1}^{n_i+n_b} \alpha_j \mathcal{B}_\xi \phi\left(\|\mathbf{x}-\xi\|\right)|_{\xi=\mathbf{x}_j}.$$

The operator $\mathcal{B}_{\mathbf{x}_j}$ is the boundary operator with respect to nodes $\mathbf{x}_j$, and $\mathcal{B}_{\mathbf{x}}$ with respect to $\mathbf{x}$, which is the identity operator in Dirichlet boundary conditions. This type of interpolant has the advantage, in elliptic PDEs, of ensuring the invertibility of the interpolation matrix for the Gaussian RBFs used this in paper. Also, in the case of Dirichlet boundary conditions, the matrix will turn out to be symmetric, which allows an efficient implementation.

The Hermite interpolation matrix is of the form

$$\mathbf{A} = \begin{bmatrix} \mathbf{\Phi} & \mathcal{B}_\xi \mathbf{\Phi} \\ \mathcal{B}_\mathbf{x} \mathbf{\Phi} & \mathcal{B}_\mathbf{x} \mathcal{B}_\xi \mathbf{\Phi} \end{bmatrix}, \tag{3.6}$$

where $[\mathbf{\Phi}] \in \mathbb{R}^{n_i \times n_i}$, $[\mathcal{B}_{\mathbf{x}_j} \mathbf{\Phi}] \in \mathbb{R}^{n_i \times n_b}$, $[\mathcal{B}_\mathbf{x} \mathbf{\Phi}] \in \mathbb{R}^{n_b \times n_i}$, and $[\mathcal{B}_\mathbf{x} \mathcal{B}_{\mathbf{x}_j} \mathbf{\Phi}] \in \mathbb{R}^{n_b \times n_b}$. The coefficients for the four submatrices are

$$\begin{aligned} [\mathbf{\Phi}]_{ij} &= \phi\left(\|\mathbf{x} - \mathbf{x}_j\|\right)|_{\mathbf{x}=\mathbf{x}_i}, && i,j = 1,\dots,n_i, \\ [\mathcal{B}_\xi \mathbf{\Phi}]_{ij} &= \mathcal{B}_\xi \phi\left(\|\mathbf{x}_i - \xi\|\right)|_{\xi=\mathbf{x}_j}, && i = 1,\dots,n_i,\ i = 1,\dots,n_b, \\ [\mathcal{B}_\mathbf{x} \mathbf{\Phi}]_{ij} &= \mathcal{B}_\mathbf{x} \phi\left(\|\mathbf{x} - \mathbf{x}_j\|\right)|_{\mathbf{x}=\mathbf{x}_i}, && i = 1,\dots,n_b,\ j = 1,\dots,n_i, \\ [\mathcal{B}_\mathbf{x} \mathcal{B}_\xi \mathbf{\Phi}]_{ij} &= \mathcal{B}_\mathbf{x} \mathcal{B}_\xi \phi\left(\|\mathbf{x} - \xi\|\right)|_{\mathbf{x}=\mathbf{x}_i,\, \xi=\mathbf{x}_j}, && i,j = 1,\dots,n_b. \end{aligned}$$

The coefficients α from the interpolation are obtained from

$$\mathbf{A}\alpha = \mathbf{d}. \tag{3.7}$$

The unknown field in the internal nodes is $\mathbf{u} = [u(\mathbf{x}_1), \dots, u(\mathbf{x}_{n_i})]$ and the boundary data in the boundary nodes is $g(\mathbf{u}) = [g(\mathbf{x}_{n_i+1}), \dots, g(\mathbf{x}_n)]$, where g is from equation (3.1).

When the stencils are inside the domain, vector $\mathbf{d}$ is given as

$$\mathbf{d^T} = [\mathbf{u}]^T$$

for the system (3.7), while when stencils are near the global boundary, vector $\mathbf{d}$ is

$$\mathbf{d^T} = [\mathbf{u}, g(\mathbf{u})]^T,$$

with Hermite interpolation matrix from (3.6).

From the reconstruction formula for the unknown field u over a subdomain, we have

$$u(\mathbf{x}) = \Phi^T(\mathbf{x})\alpha = \Phi^T(\mathbf{x})\mathbf{A}^{-1}\mathbf{d},$$

where the vector $\Phi(\mathbf{x})$ has the Gaussian RBF basis. For the interior stencils Θ_i, the vector $\Phi(\mathbf{x})$ is

$$\Phi^T(\mathbf{x}) = \left[\phi\left(\|\mathbf{x} - \mathbf{x}_1\|\right), \dots, \phi\left(\|\mathbf{x} - \mathbf{x}_n\|\right)\right]^T,$$

while in stencils near the domain,

$$\begin{aligned} &\Phi^T(\mathbf{x}) \\ &\quad = \Big[\phi\left(\|\mathbf{x} - \mathbf{x}_1\|\right), \dots, \phi\left(\|\mathbf{x} - \mathbf{x}_n\|\right), \mathcal{B}_{\mathbf{x}_1} \phi\left(\|\mathbf{x} - \mathbf{x}_1\|\right), \dots, \mathcal{B}_{\mathbf{x}_{n_b}} \phi\left(\|\mathbf{x} - \mathbf{x}_{n_b}\|\right)\Big]^T. \end{aligned}$$

The linear term $\tilde{b}$ is locally interpolated with RBFs similar to the approach in (3.7):

$$\tilde{b}\left(u(\mathbf{x}), \nabla u(\mathbf{x})\right) \approx \sum_{j=1}^{n} \beta_j \varphi_j(\mathbf{x}), \tag{3.8}$$

where β_j are the coefficients from the approximation and φ_j are RBF interpolants.

Using equation (2.3) in the integral formula (3.4) with $\xi = \mathbf{x}_i$ for the interior of the subdomain Ω_i with boundary Γ_i, we get the following local integral formula:

$$\begin{aligned} u(\xi) = \sum_{j=1}^{n} \alpha_j \left\{ \int_{\Gamma_i} Q(\mathbf{x},\xi)\varphi_j(\mathbf{x})\, d\Gamma_{\mathbf{x}} \right\} + \sum_{j=1}^{m} \beta_j \left\{ \int_{\Omega_i} G(\mathbf{x},\xi)\varphi_j(\mathbf{x})\, d\Omega_{\mathbf{x}} \right\} \\ + \int_{\Omega_i} G(\mathbf{x},\xi) f(\mathbf{x})\, d\Omega_{\mathbf{x}}. \end{aligned} \tag{3.9}$$

The discretized form of the unknown field $u_i = u\left(\mathbf{x}_i\right)$ from (3.9) is

$$u_i = \sum_{j=1}^{n} \alpha_j \widetilde{h}_{ij} + \sum_{j=1}^{m} \beta_j \widetilde{g}_{ij} + \widetilde{f}_i,$$

with α_j and β_j from equations (3.7) and (3.8), and the coefficients $\widetilde{h}_{ij}$, $\widetilde{g}_{ij}$, and $\widetilde{f}_i$ are

$$\begin{aligned} \widetilde{h}_{ij} &= \int_{\Gamma_i} Q\left(\mathbf{x},\mathbf{x}_i\right) \varphi_j(\mathbf{x})\, d\Gamma_{\mathbf{x}}, \\ \widetilde{g}_{ij} &= \int_{\Omega_i} G\left(\mathbf{x},\mathbf{x}_i\right) \varphi_j(\mathbf{x})\, d\Omega_{\mathbf{x}}, \\ \widetilde{f}_i &= \int_{\Omega_i} G\left(\mathbf{x},\mathbf{x}_i\right) f(\mathbf{x})\, d\Omega_{\mathbf{x}}, \end{aligned}$$

where the subscript i corresponds to the node set distribution enumeration and j indexes the local stencil enumeration. These boundaries and domain integrals are calculated numerically using the Gauss–Legendre quadrature.

In matrix form,

$$u_i = \widetilde{\mathbf{h}}_i^T \alpha + \widetilde{\mathbf{g}}_i^T \beta + \widetilde{f}_i, \tag{3.10}$$

with $\alpha = [\dots,\alpha_j,\dots]^T \in \mathbb{R}^n$ and $\beta = [\dots,\beta_j,\dots]^T \in \mathbb{R}^m$ column vectors with RBF interpolation coefficients, $\widetilde{\mathbf{h}}_i = [\dots,\widetilde{h}_{ij},\dots]^T \in \mathbb{R}^n$, $\widetilde{\mathbf{g}}_i = [\dots,\widetilde{g}_{ij},\dots]^T \in \mathbb{R}^m$ column vectors, and $\widetilde{f}_i \in \mathbb{R}$ scalar data.

The ill-conditioning of the linear systems to obtain α and β could be relevant when $\varepsilon \to 0$. Instead, we make a local change of basis for the LBDIM using the RBF-GA numerical technique to improve the condition number. Following [17] we consider the new RBF-GA basis $\{\psi_j\}$ instead of the Gaussian RBF basis $\{\varphi_j\}$ for local interpolation for the field u in (3.5) and the linear term $\tilde{b}$ in (3.8) that spans the same space as the Gaussian RBFs. The construction of this basis is based on the expansion $\{\tilde{\phi}_j\}$ in (2.8) and (2.9).

For $k = 0$ (adding the monomial $\{1\}$):

$$\left[\, \tilde{\varphi}_1(\mathbf{x}) \,\right] = e^{-\varepsilon^2 \|\mathbf{x}\|} \left(1 + G_0(2\varepsilon^2 \langle \mathbf{x},\mathbf{x}_1 \rangle)\right).$$

For $k = 1$ (adding monomials $\{x, y\}$):

$$\begin{bmatrix} \tilde{\varphi}_2(\mathbf{x}) \\ \tilde{\varphi}_3(\mathbf{x}) \end{bmatrix} = e^{-\varepsilon^2 \|\mathbf{x}\|} \left(\begin{bmatrix} 1 \\ 1 \end{bmatrix} + \frac{1}{1!} \begin{bmatrix} 2\varepsilon^2 \langle \mathbf{x},\mathbf{x}_2 \rangle \\ 2\varepsilon^2 \langle \mathbf{x},\mathbf{x}_3 \rangle \end{bmatrix} + \begin{bmatrix} G_1(2\varepsilon^2 \langle \mathbf{x},\mathbf{x}_2 \rangle) \\ G_1(2\varepsilon^2 \langle \mathbf{x},\mathbf{x}_3 \rangle) \end{bmatrix} \right).$$

For $k = 2$ (adding monomials $\{x^2, xy, y^2\}$):

$$\begin{bmatrix} \tilde{\varphi}_4(\mathbf{x}) \\ \tilde{\varphi}_5(\mathbf{x}) \\ \tilde{\varphi}_6(\mathbf{x}) \end{bmatrix} = e^{-\varepsilon^2\|\mathbf{x}\|} \left(\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} + \frac{1}{1!} \begin{bmatrix} 2\varepsilon^2\langle \mathbf{x}, \mathbf{x}_4\rangle \\ 2\varepsilon^2\langle \mathbf{x}, \mathbf{x}_5\rangle \\ 2\varepsilon^2\langle \mathbf{x}, \mathbf{x}_6\rangle \end{bmatrix} \right.$$
$$\left. + \frac{1}{2!} \begin{bmatrix} 4\varepsilon^4\langle \mathbf{x}, \mathbf{x}_4\rangle^2 \\ 4\varepsilon^4\langle \mathbf{x}, \mathbf{x}_5\rangle^2 \\ 4\varepsilon^4\langle \mathbf{x}, \mathbf{x}_6\rangle^2 \end{bmatrix} + \begin{bmatrix} G_2(2\varepsilon^2\langle \mathbf{x}, \mathbf{x}_4\rangle) \\ G_2(2\varepsilon^2\langle \mathbf{x}, \mathbf{x}_5\rangle) \\ G_2(2\varepsilon^2\langle \mathbf{x}, \mathbf{x}_6\rangle) \end{bmatrix} \right).$$

And so on for $k \geq 3$, where $G_k(\cdot)$ comes from (2.10).

The RBF-GA basis functions $\{\psi_j\}$ are obtained cancelling analytically all the Taylor coefficients in (2.8) and finding an orthonormal basis for the null space of a matrix of data from nodes $\mathbf{x}_j$.

For $k = 0$:

$$[\,\psi_1(\mathbf{x})\,] = e^{-\varepsilon^2\|\mathbf{x}\|} \frac{1}{\varepsilon^0} \mathbf{B}_0 \left[\, G_0(2\varepsilon^2\langle \mathbf{x}, \mathbf{x}_1\rangle)\,\right], \quad \text{with } \mathbf{B}_0 = [1].$$

For $k = 1$:

$$\begin{bmatrix} \psi_2(\mathbf{x}) \\ \psi_3(\mathbf{x}) \end{bmatrix} = e^{-\varepsilon^2\|\mathbf{x}\|} \frac{1}{\varepsilon^2} \mathbf{B}_1 \begin{bmatrix} G_1(2\varepsilon^2\langle \mathbf{x}, \mathbf{x}_1\rangle) \\ G_1(2\varepsilon^2\langle \mathbf{x}, \mathbf{x}_2\rangle) \\ G_1(2\varepsilon^2\langle \mathbf{x}, \mathbf{x}_3\rangle) \end{bmatrix}, \quad \text{with } \mathbf{B}_1 = \big[\text{null}\begin{bmatrix} 1 & 1 & 1 \end{bmatrix}\big]^T.$$

For $k = 2$:

$$\begin{bmatrix} \psi_4(\mathbf{x}) \\ \psi_5(\mathbf{x}) \\ \psi_6(\mathbf{x}) \end{bmatrix} = e^{-\varepsilon^2\|\mathbf{x}\|} \frac{1}{\varepsilon^4} \mathbf{B}_2 \begin{bmatrix} G_2(2\varepsilon^2\langle \mathbf{x}, \mathbf{x}_1\rangle) \\ G_2(2\varepsilon^2\langle \mathbf{x}, \mathbf{x}_2\rangle) \\ G_2(2\varepsilon^2\langle \mathbf{x}, \mathbf{x}_3\rangle) \\ G_2(2\varepsilon^2\langle \mathbf{x}, \mathbf{x}_4\rangle) \\ G_2(2\varepsilon^2\langle \mathbf{x}, \mathbf{x}_5\rangle) \\ G_2(2\varepsilon^2\langle \mathbf{x}, \mathbf{x}_6\rangle) \end{bmatrix},$$
$$\text{with } \mathbf{B}_2 = \left[\text{null} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ x_1 & x_2 & x_3 & x_4 & x_5 & x_6 \\ y_1 & y_2 & y_3 & y_4 & y_5 & y_6 \end{bmatrix}\right]^T.$$

And so on for $k \geq 3$. The construction of this basis using null-spaces and the values of the inferior gamma function in $G_k(z)$ are available in an efficient way without numerical cancellations.

Finally, with this new basis $\{\psi_j\}$, the discretized matrix form for u_i in (3.10) is

$$u_i = f_i + \left(\mathbf{h}_i^T \mathbf{A}^{-1} + \widetilde{\mathbf{h}}_i^T \widetilde{\mathbf{A}}^{-1} \mathbf{A}_{\mathbf{b}} \mathbf{A}^{-1}\right) \mathbf{d}, \tag{3.11}$$

where the matrix $\mathbf{A}_{\mathbf{b}}$ corresponds to calculation of the vector $\tilde{b}$. Following [31] to avoid the numerical computation of the inverse matrix $\mathbf{A}^{-1}$ from equation (3.7) and the interpolation matrix from equation (3.8), $\widetilde{\mathbf{A}}^{-1}$, we rewrite expression (3.11) as

$$u_i = f_i + \mathbf{z}_i^T \mathbf{d},$$

where the algorithmic procedure to calculate this equation is

Step 1. Solve $\widetilde{\mathbf{A}}\widetilde{\mathbf{w}}_i = \widetilde{\mathbf{h}}_i$ (since $\widetilde{\mathbf{A}}$ is symmetric).
Step 2. Calculate $\mathbf{w}_i = \mathbf{h}_i + {\mathbf{A}_\mathbf{b}}^T \widetilde{\mathbf{w}}_i$.
Step 3. Solve $\mathbf{A}^T \mathbf{z}_i = \mathbf{w}_i$.
Step 4. Calculate $u_i = f_i + \mathbf{z}_i^T \mathbf{d}$.

Thus, equation (3.11) is collocated at each trial point at each stencil to form a global sparse matrix system. This system of equations is solved efficiently using the GMRES iterative method.

3.2. **The LBDIM-QR method.** The LBDIM-QR method was presented in [31] as LBDIM-St. In this case the local RBF interpolants for the field u and the term $\tilde{b}$ were developed using the RBF-QR basis. This approximation follows a similar integral formula approach to the LBDIM and uses the RBF-QR basis $\{\psi_j\}$ for the local interpolations. It was shown that it reduces the numerical error due to the instabilities of the low shape parameter ε without the drawback of ill-conditioning at local level for the Gaussian RBFs. Also, it makes a stable computation which circumvents the ill-conditioning of local systems of linear equations arising from RBF interpolations. It was applied in several elliptic PDEs with Dirichlet and Neumann boundary conditions over different 2D domains obtaining the best results for quasi-uniform node distributions. Nevertheless, the calculation of the expansion of the Gaussian RBFs in the RBF-QR method has a computational cost.

4. Numerical results

In this section we adopt benchmark solutions defined over the unit disk, the unit square and an irregular domain in two dimensions. We present numerical results for the novel LBDIM-GA and compare them with the LBDIM and the LBDIM-QR formulations presented in [31]. Moreover, we present comparisons with other numerical methods. We study the variation of the shape parameter ε, the stencil size n, and N, the total number of nodes in the interior domain.

The domains Ω considered are discretized using scattered nodes in 2D from the repellent distribution presented in [3] and from the quasi-uniform node distributions presented in [14]. The latter algorithm generates a set of nodes with an advancing front method using a density function starting from the boundary Γ into the domain.

In order to show the solution accuracy, we define the *L_2-norm error*

$$L_2\text{-error} = \sqrt{\frac{\sum_{i=1}^{N} \left(u_i^{\text{exac}} - u_i^{\text{approx}}\right)^2}{\sum_{i=1}^{N} \left(u_i^{\text{exac}}\right)^2}}$$

and the *root mean square error*

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{N} \left(u_i^{\text{exac}} - u_i^{\text{approx}}\right)^2}{N}}.$$

4.1. **Poisson's equation over the unit disk.** Let us consider the following Poisson equation with Dirichlet boundary conditions defined in the domain $\Omega = \{(x, y) \mid x^2 + y^2 \leq 1\}$ shown in Figure 4:

$$\nabla^2 u\,(x, y) = (-200) \sin[10(x + y)]. \tag{4.1}$$

The exact solution to this problem is $u\,(x, y) = \sin[10(x + y)]$, which has an oscillatory behavior. This problem was solved by Bayona et al. [3] using the RBF-generated finite difference (RBF-FD) method, combining polyharmonic splines (see Table 1) with multivariate polynomials.

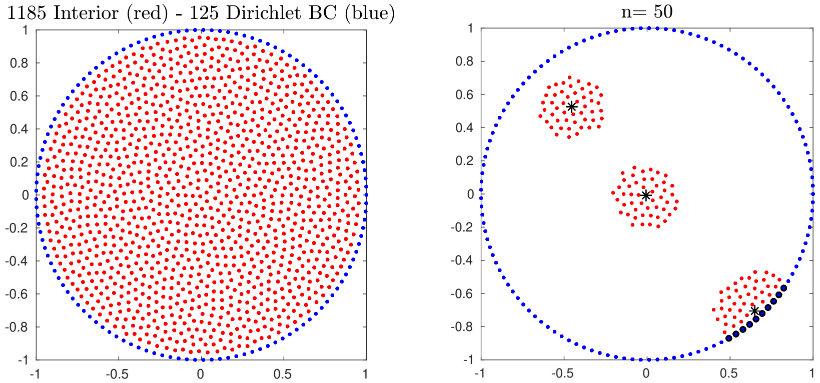

FIGURE 4. Repel node distribution and local stencils.

The numerical example (4.1) is solved for three different domain discretizations based on the repel algorithm mentioned in the aforementioned paper, corresponding to $h = 0.05$, 0.025, 0.01. These result in $N = 1185, 4880, 9639$ interior nodes, and $N_b = 125, 251, 349$ boundary nodes, respectively. The left subplot in Figure 4 shows the repel node distribution for $N = 1185$; the structure for $N = 4880, 96390$ is similar but denser. The right subplot in the figure shows the local stencils with $n = n_i + n_b = 50$ over the unit disk, one of them close to the boundary with the data value from the Dirichlet boundaries conditions.

The objective of this example is to explore the effect of increasing the number of stencil nodes from $n = 10$ to $n = 90$ with steps of 10 and also varying $\varepsilon \in [1, 10]$.

4.1.1. *Accuracy isolines.* In Figure 5 we show accuracy isolines ($\log_{10}(L_2$-error)) for $N = 1185, 4880, 9639$ in rows. In the first column the LBDIM with Gaussian RBF interpolations $\phi(r) = e^{-(\varepsilon r)^2}$ is presented. In the second column we show the LBDIM-GA and in the third column the LBDIM-QR. These are the LBDIM versions with the two local stabilizing techniques: the RBF-GA and the RBF-QR algorithms.

When looking along the left edge of the subplot in the first column corresponding to LBDIM, we observe an unstable region for small values of ε and high values of n (yellow region in the figure). On the other hand, for LBDIM-GA in the second column and for LBDIM-QR in the third column, we observe that instability issues

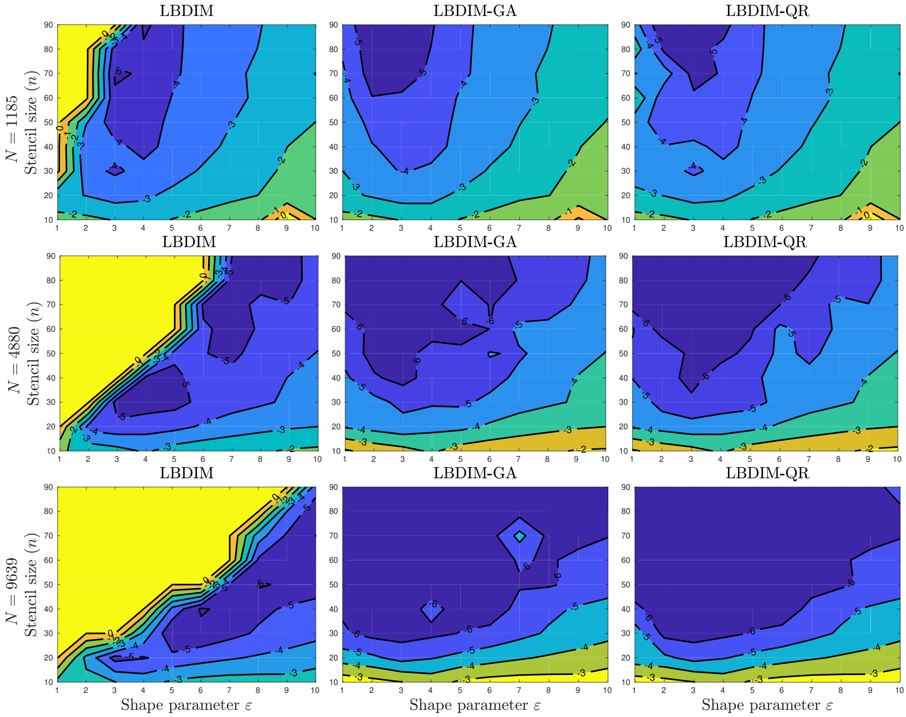

FIGURE 5. Accuracy isolines ($\log_{10}(L_2\text{-error})$) with $N = 1185, 4880, 9639$.

are no longer present. We obtain a great stability area (blue region in the figure) for a small ε regime and for stencils with a higher number of nodes. The best order of accuracy is 1×10^{-5} for both stable methods when $N = 1185$.

The middle row in Figure 5 presents results for $N = 4880$. Accuracy results for LBDIM present stability issues in the whole upper triangular part of the left subplot. There is a tradeoff between n and ε when using LBDIM. For LBDIM-GA and LBDIM-QR, we obtain great areas of stability attaining the best results for an order of accuracy of 1×10^{-7}.

The bottom row in Figure 5 gives results for $N = 9639$. The L_2-error for tLBDIM is unstable in the whole upper triangular part of the left subplot. Besides, when using both stabilizing techniques, we obtain a great blue stable region with a best order of accuracy of 1×10^{-7}. We observe that the L_2-error decreases as the node distribution is refined. These results for tLBDIM-GA improved the numerical L_2-error presented in [3] using the the RBF-FD method with polynomial augmentation.

4.1.2. *Conditioning isolines.* In Figure 6 we show the condition number isolines $(\log_{10}(\kappa(\mathbf{A}_i)))$ for the interpolation matrix (3.7) for $N = 1185, 4880, 9639$, row-wise. As before, in the first column we have the condition number for LBDIM, in the second column for LBDIM-GA, and in the third one for LBDIM-QR.

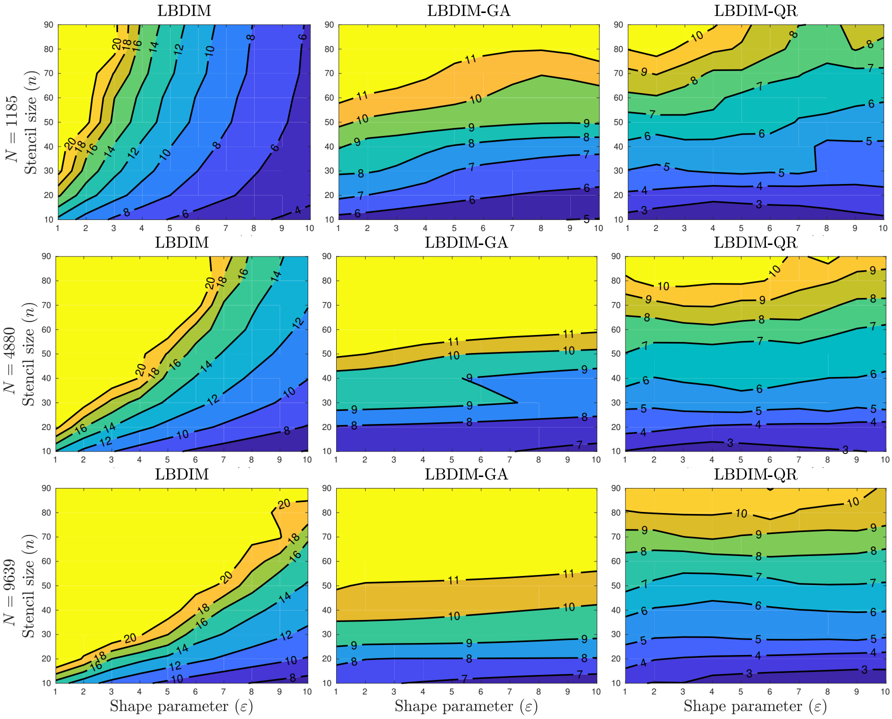

FIGURE 6. Condition number isolines ($\log_{10}(\kappa(\mathbf{A}_i))$) with $N = 1185, 4880, 9639$.

In the first column with double precision, when using Gaussian RBFs without employing a stable algorithm, we obtain a great region with condition number greater than $1 \times 10^{+20}$. This occurs when the shape parameter of the Gaussians decreases and n increases. When N increases from 1185 to 9639, the worst condition number increases in the figure (yellow area). This means that the high condition number encountered is harmless in actual numerical computation of LBDIM's approximate solution. This corresponds to the unstable areas from Figure 5.

In the second column we present the conditioning for LBDIM-GA. In this case the condition number of the interpolation matrix with RBF-GA increases as the stencil size increases. The last two rows of subplots show that, for $n \geq 50$, the condition number is $1 \times 10^{+11}$ and also the condition number tends to be independent of the shape parameter since the isolines are almost horizontal.

In the last column we give results for the conditioning of LBDIM-QR. A similar behavior is obtained but with a condition number of order $1 \times 10^{+10}$.

4.1.3. *Computational time.* The numerical experiments were performed on a PC with 7.5 GB of RAM and an Intel Core i7-7500U 7th generation CPU running at 2.70 GHz. All timings were performed using the MATLAB R2017a implementations of the 2D algorithms. Figure 7 illustrates the computational cost of the

LBDIM compared with the LBDIM-GA and the LBDIM-QR for $N = 1185$. The subplots show the computational time for $\varepsilon = 2, 3, 6$.

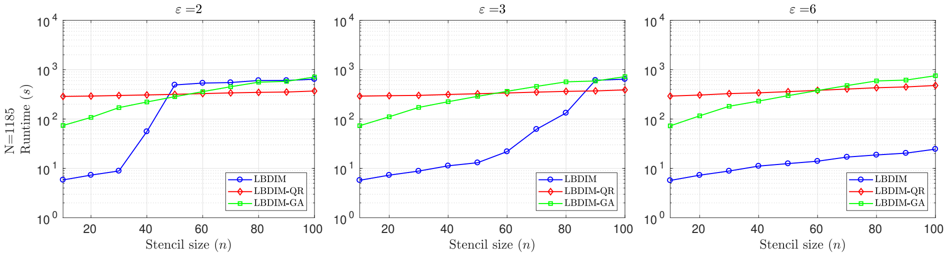

FIGURE 7. CPU runtime as function of n for $N = 1185$ and $\varepsilon = 2, 3, 6$.

The cost for the LBDIM increases for higher number of stencil sizes n when ε decreases and, for the LBDIM-QR, the CPU time remains the same for different values of the shape parameter. For higher stencil sizes and low shape parameter corresponding to best accuracy, the LBDIM-QR cost is less than that for the LBDIM. The CPU time for computing the LBDIM-GA is similar to the cost of the LBDIM-QR but, for small stencil size, the stabilizing RBF-GA technique is faster.

It is well known that the computational cost of the RBF-GA algorithm is dominated by the QR factorization to obtain the B_k matrices and the computation of the incomplete gamma function. Also, the RBF-QR method is slower than the RBF-GA since the CPU time lies in the expansion of the new basis. This is shown in all subfigures for $n \leq 50$.

Local interpolations with RBF-QR and RBF-GA algorithms overcome the instability of the LBDIM at the cost of more computational time. However, introducing these stabilizing techniques into a local integral method represents an improvement since we increase the accuracy.

4.2. **Convection-diffusion-reaction equation with variable coefficients.**

For this test, we consider the scalar two-dimensional problem studied in [5] with Dirichlet boundary conditions in the square domain $\{(x, y) \mid 0 \leq x \leq 1,\ 0 \leq y \leq 1\}$,

$$\nabla^2 u + \left(xy + x^2\right) \frac{\partial u}{\partial x} + x \sinh(y) \frac{\partial u}{\partial y} + \sin(x + y)u = f(x, y), \quad (x, y) \in \Omega. \tag{4.2}$$

The analytical solution of this problem is $u = y \sin(\pi x) + x \cos(\pi y)$ and the term $f(x, y)$ in (4.2) is forced to satisfy the exact solution. The stencil size n is fixed in 20 and 30. The node distributions in the 2D domain Ω are quasi-uniform [14].

In Figure 8 we show the RMSE versus ε for several numbers of nodes $N = 443, 960, 1678$, with quasi-uniform distributions using $n = 20$, 30 stencil size. The error on the LBDIM-Direct solution (solid blue line) converges only for large values of ε and starts diverging as soon as ε becomes too small. The error on the LBDIM-QR solution (green dashed line) converges for the small range of the shape parameter. Furthermore, when increasing the stencil size from 20 to 30, we observe that the divergence of the RMSE starts for higher ε since the RBF interpolation matrices are larger.

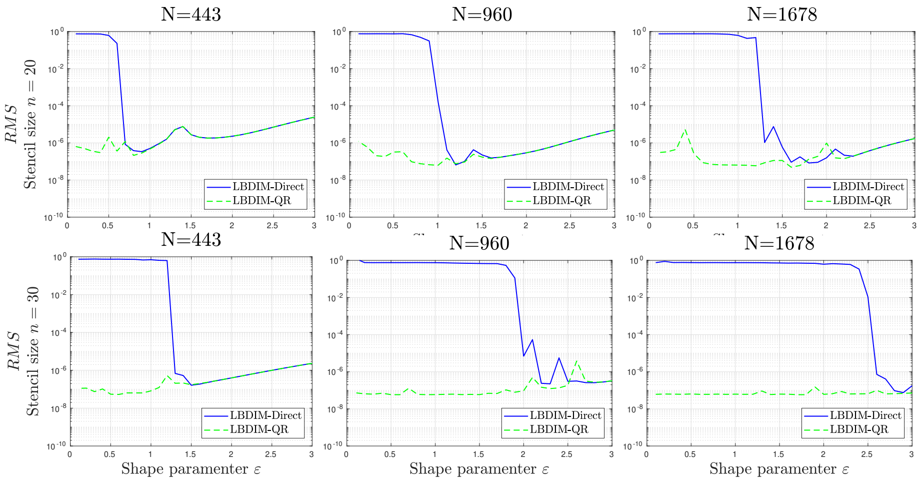

FIGURE 8. RMSE comparison between LBDIM and LBDIM-QR with Gaussian RBFs for $N = 443, 960, 1678$ quasi-uniform nodes.

Comparing the results obtained through LBDIM-QR against those obtained by Castro et al. [5] with the Method of Approximate Particular Solutions (MAPS) using multiquadric RBFs (see Table 1) in two dimensions, we can say that the stabilized LBDIM-QR improves the RMSE in four orders of magnitude. The RMSE for the MAPS on cartesian node distributions with $N = 443, 960, 1678$ total points is of order 1×10^{-4}, while the LBDIM-QR achieves an order of 1×10^{-8} on quasi-uniform node distributions as in Figure 9 with stencil size $n = 30$. These results were obtained by trial and error in order to find an optimum shape parameter ε_{opt}, which produces the lowest error for each N.

4.3. **General elliptic equation over an irregular domain.** Finally, let us consider the following elliptic equation with Dirichlet boundary conditions defined on the irregular domain $\Omega \subset \mathbb{R}^2$ with a hole and sharp boundaries illustrated in Figure 10. The PDE is given by

$$\nabla^2 u + \alpha(x,y)\frac{\partial u}{\partial x} + \beta(x,y)\frac{\partial u}{\partial y} + \gamma(x,y)u = f(x,y), \quad (x,y) \in \Omega. \tag{4.3}$$

The oscillatory coefficients in equation (4.3) are given by $\alpha = e^{-y^2+\cos(4\pi x)\sin(3\pi y)}$, $\beta = -y\sin(4\pi x)$, and $\gamma = x^2 y$. The exact solution to this problem is $u(x,y) = \sin(2\pi y^2 + 3\pi x) - \cos(\pi y - 2\pi x^2)$. A similar PDE over this region was numerically solved using the RBF-FD method in [3], and the same PDE was solved in [31] for the LBDIM-QR.

We study the accuracy in double precision for L_2-error as a function of the stencil size n and the shape parameter ε. Similarly to subsection 4.1.1, in Figure 11

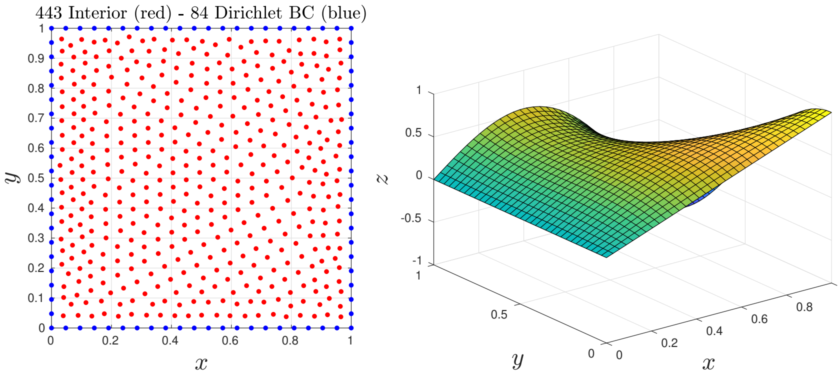

FIGURE 9. Quasi-uniform node distribution and exact solution.

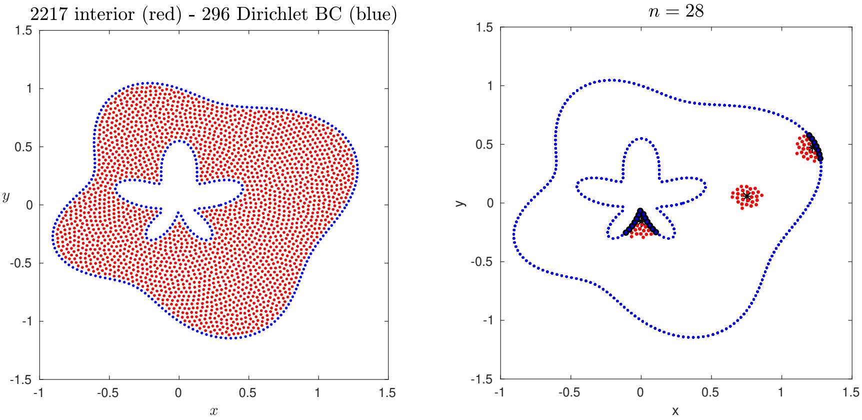

FIGURE 10. Domain discretization of irregular domain Ω with a repel node distribution and local stencils for $n = 28$.

we show the accuracy isolines for the LBDIM (first column) and the LBDIM-GA (second column) with $N = 1146, 2217$ repel nodes. The stencil size varies as $n = 10, 15, 21, 28, 36, 45, 55, 66, 78$.

In the first column on the top left of each subfigure, we observe the unstable region (yellow area in the subfigure) when the shape parameter decreases and the stencil size increases. In the case $N = 1146$ we obtain a great stable region (blue area in the subfigure) of order 1×10^{-5} that extends for $2 \leq \varepsilon \leq 6$ and $n \geq 30$. When $N = 2217$ the stable region is reduced due to ill-conditioning of local Gaussian RBF-Direct interpolations, although we get an order 1×10^{-6} for the L_2-error.

For the LBDIM-GA in the second column we observe that the stability region extends over reaching an L_2-error order of 1×10^{-5} and 1×10^{-6} with $N = 1146$

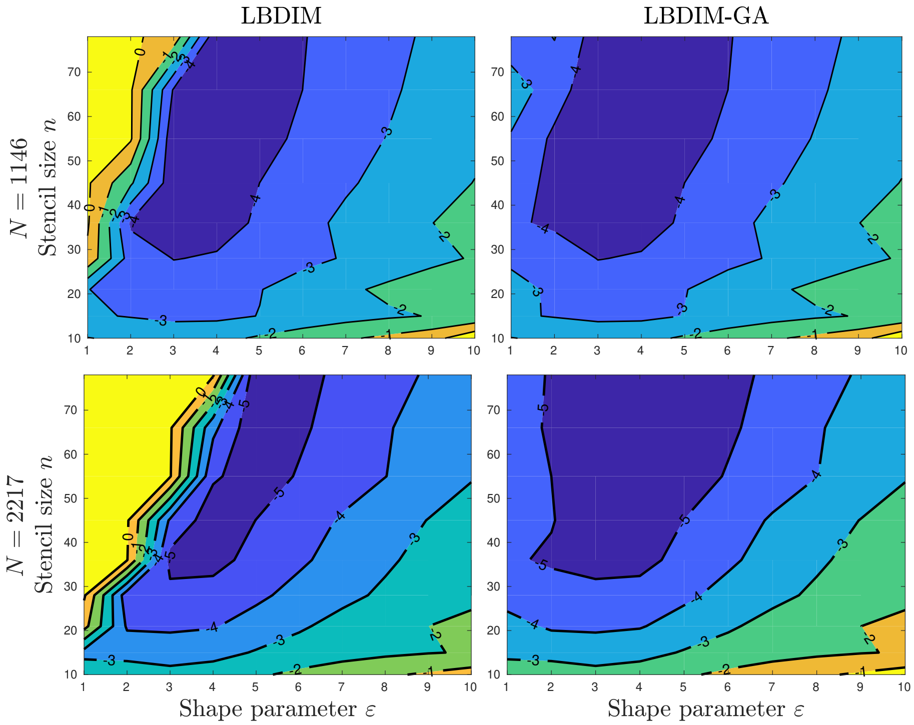

Figure 11. Accuracy isolines ($\log_{10}(L_2\text{-error})$) for LBDIM and LBDIM-GA with $N = 1146, 2217$.

and 2217 respectively. Thus, this approach for reaching high accuracy spans the same function space for Gaussian RBFs and extends the range for a small ε regime.

An interesting observation is that the stencil size may be variable over the domain Ω. In Figure 10 we get different shapes for the stencils when the collocation points are near a smooth part of the boundary, near a sharp point or totally inside. Although not studied here, a possible strategy is to use adaptive meshless centers for the RBF stencils in this type of domain.

5. Conclusions

We proposed a new algorithm for generating numerical approximations for solving boundary value problems with Dirichlet conditions. This method, named LBDIM-GA, is a local integral method that improves the numerical stability of the error for low ε range using the same space as the Gaussian RBFs, but the space is better conditioned. The local RBF interpolation matrices are well-conditioned, providing high accuracy for general linear elliptic PDEs with Dirichlet boundary conditions over general domains. The presented experiments show that the RBF-GA algorithm works effectively in local interpolation for numerical integral methods in 2D. We also compared results with the LBDIM-QR established in the literature (presented as LBDIM-St in [31]), where the LBDIM-GA method improves the computational time. In double precision, using a stable algorithm such as RBF-QR or RBF-GA on a local integral method such as LBDIM gives numerical access to small ε-values. It remains an open task to provide further stable implementations for other RBF stabilizing techniques in local integral methods. Also, the extension of implementations to higher dimensions and non-stationary problems is currently under research.

Acknowledgments

I would like to thank Bengt Fornberg and Natasha Flyer from the University of Colorado at Boulder for the valuable discussions and suggestions they provided during my stay at that institution in 2018. I am also grateful for the support from the Centro Internacional Franco Argentino de Ciencias de la Información y de Sistemas (CIFASIS-CONICET-UNR) in Rosario, Argentina.

References

[1] M. Ahmad, Siraj-ul-Islam, and E. Larsson, Local meshless methods for second order elliptic interface problems with sharp corners, *J. Comput. Phys.* **416** (2020), 109500, 17 pp. MR 4094820.

[2] P. R. S. Antunes, Reducing the ill conditioning in the method of fundamental solutions, *Adv. Comput. Math.* **44** (2018), no. 1, 351–365. MR 3755753.

[3] V. Bayona, N. Flyer, B. Fornberg, and G. A. Barnett, On the role of polynomials in RBF-FD approximations: II. Numerical solution of elliptic PDEs, *J. Comput. Phys.* **332** (2017), 257–273. MR 3591180.

[4] N. Caruso, M. Portapila, and H. Power, An efficient and accurate implementation of the localized regular dual reciprocity method, *Comput. Math. Appl.* **69** (2015), no. 11, 1342–1366. MR 3339046.

[5] D. A. Castro, W. F. Florez, M. Portapila, N. Caruso, C. A. Bustamante, R. Posada, and J. M. Granados, Numerical examination of the effect of different boundary conditions on the method of approximate particular solutions for scalar and vector problems, *Eng. Anal. Bound. Elem.* **127** (2021), 75–90. MR 4239676.

[6] M. Dehghan and M. Najafi, Numerical solution of a non-classical two-phase Stefan problem via radial basis function (RBF) collocation methods, *Eng. Anal. Bound. Elem.* **72** (2016), 111–127. MR 3550377.

[7] K. P. Drake and G. B. Wright, A stable algorithm for divergence-free radial basis functions in the flat limit, *J. Comput. Phys.* **417** (2020), 109595, 9 pp. MR 4106713.

[8] G. E. Fasshauer, *Meshfree Approximation Methods with MATLAB*, Interdisciplinary Mathematical Sciences, 6, World Scientific, Hackensack, NJ, 2007. MR 2357267.

[9] G. E. Fasshauer and M. J. McCourt, Stable evaluation of Gaussian radial basis function interpolants, *SIAM J. Sci. Comput.* **34** (2012), no. 2, A737–A762. MR 2914302.

[10] G. E. Fasshauer and M. J. McCourt, *Kernel-Based Approximation Methods Using MATLAB*, World Scientific, Hackensack, NJ, 2015. https://doi.org/10.1142/9335.

[11] N. Flyer, G. A. Barnett, and L. J. Wicker, Enhancing finite differences with radial basis functions: experiments on the Navier-Stokes equations, *J. Comput. Phys.* **316** (2016), 39–62. MR 3494343.

[12] N. Flyer and E. Lehto, Rotational transport on a sphere: local node refinement with radial basis functions, *J. Comput. Phys.* **229** (2010), no. 6, 1954–1969. MR 2586231.

[13] B. Fornberg and N. Flyer, *A Primer on Radial Basis Functions with Applications to the Geosciences*, CBMS-NSF Regional Conference Series in Applied Mathematics, 87, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2015. MR 3450067.

[14] B. Fornberg and N. Flyer, Fast generation of 2-D node distributions for mesh-free PDE discretizations, *Comput. Math. Appl.* **69** (2015), no. 7, 531–544. MR 3320269.

[15] B. Fornberg, E. Larsson, and N. Flyer, Stable computations with Gaussian radial basis functions, *SIAM J. Sci. Comput.* **33** (2011), no. 2, 869–892. MR 2801193.

[16] B. Fornberg and E. Lehto, Stabilization of RBF-generated finite difference methods for convective PDEs, *J. Comput. Phys.* **230** (2011), no. 6, 2270–2285. MR 2764546.

[17] B. Fornberg, E. Lehto, and C. Powell, Stable calculation of Gaussian-based RBF-FD stencils, *Comput. Math. Appl.* **65** (2013), no. 4, 627–637. MR 3011446.

[18] B. Fornberg and C. Piret, A stable algorithm for flat radial basis functions on a sphere, *SIAM J. Sci. Comput.* **30** (2007/08), no. 1, 60–80. MR 2377431.

[19] B. Fornberg and C. Piret, On choosing a radial basis function and a shape parameter when solving a convective PDE on a sphere, *J. Comput. Phys.* **227** (2008), no. 5, 2758–2780. MR 2388503.

[20] B. Fornberg, G. Wright, and E. Larsson, Some observations regarding interpolants in the limit of flat radial basis functions, *Comput. Math. Appl.* **47** (2004), no. 1, 37–55. MR 2062724.

[21] B. Fornberg and G. Wright, Stable computation of multiquadric interpolants for all values of the shape parameter, *Comput. Math. Appl.* **48** (2004), no. 5-6, 853–867. MR 2105258.

[22] R. Franke, Scattered data interpolation: tests of some methods, *Math. Comp.* **38** (1982), no. 157, 181–200. MR 0637296.

[23] R. L. Hardy, Theory and applications of the multiquadric-biharmonic method. 20 years of discovery 1968–1988, *Comput. Math. Appl.* **19** (1990), no. 8-9, 163–208. MR 1040159.

[24] E. J. Kansa, Multiquadrics—a scattered data approximation scheme with applications to computational fluid-dynamics. I. Surface approximations and partial derivative estimates, *Comput. Math. Appl.* **19** (1990), no. 8-9, 127–145. MR 1040157.

[25] E. J. Kansa, Multiquadrics—a scattered data approximation scheme with applications to computational fluid-dynamics. II. Solutions to parabolic, hyperbolic and elliptic partial differential equations, *Comput. Math. Appl.* **19** (1990), no. 8-9, 147–161. MR 1040158.

[26] E. Larsson and B. Fornberg, Theoretical and computational aspects of multivariate interpolation with increasingly flat radial basis functions, *Comput. Math. Appl.* **49** (2005), no. 1, 103–130. MR 2123189.

[27] C. A. Micchelli, Interpolation of scattered data: distance matrices and conditionally positive definite functions, *Constr. Approx.* **2** (1986), no. 1, 11–22. MR 0891767.

[28] P. K. Mishra, G. E. Fasshauer, M. K. Sen, and L. Ling, A stabilized radial basis-finite difference (RBF-FD) method with hybrid kernels, *Comput. Math. Appl.* **77** (2019), no. 9, 2354–2368. MR 3945118.

[29] E. H. Ooi and V. Popov, An efficient implementation of the radial basis integral equation method, *Eng. Anal. Bound. Elem.* **36** (2012), no. 5, 716–726. MR 2880719.

[30] P. W. Partridge, Radial basis approximation functions in the boundary element dual reciprocity method, *Trans. Model. Simul.* **23** (1999), 325–334.

[31] L. Ponzellini Marinelli, N. Caruso, and M. Portapila, A stable computation on local boundary-domain integral method for elliptic PDEs, *Math. Comput. Simulation* **180** (2021), 379–400. MR 4152797.

[32] V. Popov and T. Thanh Bui, A meshless solution to two-dimensional convection-diffusion problems, *Eng. Anal. Bound. Elem.* **34** (2010), no. 7, 680–689. MR 2639905.

[33] M. Portapila and H. Power, Iterative schemes for the solution of system of equations arising from the DRM in multi domain approach, and a comparative analysis of the performance of two different radial basis functions used in the interpolation, *Eng. Anal. Bound. Elem.* **29** (2005), no. 2, 107–125. https://doi.org/10.1016/j.enganabound.2004.08.008.

[34] S. A. Sarra, Integrated multiquadric radial basis function approximation methods, *Comput. Math. Appl.* **51** (2006), no. 8, 1283–1296. MR 2235828.

[35] R. Schaback, Error estimates and condition numbers for radial basis function interpolation, *Adv. Comput. Math.* **3** (1995), no. 3, 251–264. MR 1325034.

[36] G. B. Wright and B. Fornberg, Stable computations with flat radial basis functions using vector-valued rational approximations, *J. Comput. Phys.* **331** (2017), 137–156. MR 3588686.

Luciano Ponzellini Marinelli
Departamento de Matemática, Facultad de Ciencias Exactas, Ingeniería y Agrimensura,
Rosario, Santa Fe, Argentina
Facultad de Química e Ingeniería, Pontificia Universidad Católica Argentina, Rosario, Santa Fe, Argentina
luciano@fceia.unr.edu.ar, lucianoponzellini@uca.edu.ar

Received: September 30, 2021
Accepted: April 1, 2022