Computational Methods for Differential Equations
http://cmde.tabrizu.ac.ir
Vol. 7, No. 1, 2019, pp. 138-151

Stable Gaussian radial basis function method for solving Helmholtz equations

Jalil Rashidinia*
School of Mathematics, Iran University of Science and Technology, Tehran, Iran.
E-mail: rashidinia@iust.ac.ir

Manoochehr Khasi
School of Mathematics, Iran University of Science and Technology, Tehran, Iran.
E-mail: m_khasi@iust.ac.ir

Abstract Radial basis functions (RBFs) are a powerful method for obtaining the numerical solution of high-dimensional problems. They are often referred to as a meshfree method and the spectrally accurate can be achieved by them. In this paper, we analyze a new stable method for evaluating Gaussian radial basis function interpolants based on the eigenfunction expansion. We develop our approach in two-dimensional spaces for solving Helmholtz equations. In this paper, the eigenfunction expansions are rebuilt based on Chebyshev polynomials which are more suitable in numerical computations. Numerical examples are presented to demonstrate the effectiveness and robustness of the proposed method for solving two-dimensional Helmholtz equations.

Keywords. Gaussian radial basis functions, Eigenfunction expansion, Helmholtz equations, Sylvester system.

2010 Mathematics Subject Classification. 65M99, 65N99, 76R50.

1. Introduction

Helmholtz equation is very important in a variety of science and engineering problems, for example, in physics, technology, geophysics and optical problems. There are several numerical methods for solving the Helmholtz equation. Among them we can mention the Finite element method in [2], the Finite volume method in [13, 21], the Boundary element method in [22], spectral element methods in [17, 19], radial basis function method in [15], or spectral methods in [1, 16].

Radial basis functions (RBFs) have been used in many branches of science and engineering. Today there are many books related to the theory, applications and implementations of RBFs (see [3, 6, 23, 25]). By using infinitely smooth basis functions such as Gaussians or Multiquadric , the exponential convergence rate can be achieved [9, 25]. The best accuracy can usually be obtained when the shape parameter is small. But we should mention that as the shape parameter becomes small, the interpolant

Received: 17 September 2017 ; Accepted: 27 October 2018.
∗ Corresponding author.

matrix becomes increasingly ill-conditioned. This fact has led to a growing number of stable approaches to overcome this problem, such as the Contour-Padé approach [8] or the RBF-QR method which in 2007 was introduced by Fornberg and Piret [10, 11] and later on by Larsson [14]. In [7], Fasshauer and McCourt developed a different type of the RBF-QR method by considering an eigenfunction expansion of the Gaussian RBF. They established a connection between the RBF-QR algorithm and Mercer's theorem which states any positive definite kernel such as Gaussian has an eigenfunction expansion . In our previous work [20], we changed the eigenfunction expansion approach for evaluating Gaussian RBF interpolants by taking advantage of the orthogonality of the eigenfunctions which are based on Hermite polynomials.

In this article, we study numerical solution by a stable approach based on the eigenfunction expansions of Gaussian Radial basis functions which is an extension of our previous work [20]. The eigenfunction expansions are rebuilt based on Chebyshev polynomials, which are more suitable in numerical computations.

In this paper, we consider the following two-dimensional Helmholtz equation

$$\Delta u + k^2 u = f(x,y), \qquad \text{in } \Omega, \tag{1.1}$$

where f is a smooth functions, $\Delta = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$ is Laplacin operator, Ω is a bounded domain in $\mathbb{R}^2$, k is the wavenumber defined as $k := 2\pi p/v$ with p and v stating frequency and speed, respectively, and u is the unknown solution representing a pressure field. The wavenumber k is a constant for the homogeneous medium, and varies for the heterogeneous medium. On the boundary $\partial\Omega$, a Robin boundary condition is used:

$$\alpha u(x,y) + \beta \frac{\partial u}{\partial \mathbf{n}}(x,y) = h(x,y), \qquad \text{on } \partial\Omega, \tag{1.2}$$

where $\mathbf{n}$ is the outward unit normal vector to the boundary, $h(x,y)$ is a smooth function, and α, β are non zero, simultaneously. If $\alpha \neq 0$, $\beta = 0$, the boundary conditions can be imposed as Dirichlet boundary conditions and if $\alpha = 0$, $\beta \neq 0$, they can be imposed as Neumann boundary conditions.

The remaining part of this paper is organized as follows. Section 2 is devoted to some essential concepts about the stable method for Gaussian RBF interpolation based on eigenfunction expansions. In Section 3, the solution of two-dimensional Helmholtz equations are investigated. In Section 4, some numerical experiments that illustrate the accuracy, efficiency and stability of the proposed method are included.

2. A new stable method for 2d Gaussian RBF interpolation

Before discussing the discretization of the equation (1.1), we present a brief summary of the new stable method for Gaussian RBF interpolation. Based on Mercer's theorem, every positive definite kernel $K : \Omega \times \Omega \to \mathbb{R}$ where $\Omega \subset \mathbb{R}^d$, can be explained in terms of the positive eigenvalues $\lambda_n \to 0$ and normalized eigenfunctions φ_n of an associated compact integral operator [24]. In fact

$$K(\mathbf{x},\mathbf{z}) = \sum_{n=1}^{\infty} \lambda_n \varphi_n(\mathbf{x}) \varphi_n(\mathbf{z}). \tag{2.1}$$

Now suppose $\mathcal{K}$ is a native Hilbert space of functions on Ω, $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\} \subset \Omega$ is the set of centers and $\mathcal{K}_X = \text{span}\{K(\cdot, \mathbf{x}_j), \quad x_j \in X\}$ is the subspace spanned by the basis $K(\cdot, \mathbf{x}_j), \quad 1 \leq \mathbf{x}_j \leq N$. We can write the interpolant $s_f \in \mathcal{K}_X$ of $f \in \mathcal{K}$ at $\mathbf{x}_1, \ldots, \mathbf{x}_N$ as

$$s_f(x) = \sum_{j=1}^{N} c_j K(\mathbf{x}, \mathbf{x}_j),$$

where the coefficients c_j are determined by the interpolation conditions $s_f(\mathbf{x}_j) = f(\mathbf{x}_j) : \quad j = 1, \cdots, N$; i.e., they can be obtained by solving the following $N \times N$ linear system

$$\mathbf{K}\mathbf{c} = \mathbf{f},$$

where $\mathbf{f} = (f(\mathbf{x}_1), \cdots, f(\mathbf{x}_N))^T$, $\mathbf{c} = (c_1, \cdots, c_N)^T$, and

$$\mathbf{K} = \begin{bmatrix} K(\mathbf{x}_1, \mathbf{x}_1) & \cdots & K(\mathbf{x}_1, \mathbf{x}_N) \\ \vdots & & \vdots \\ K(\mathbf{x}_N, \mathbf{x}_1) & \cdots & K(\mathbf{x}_N, \mathbf{x}_N) \end{bmatrix}.$$

In applications, we have to truncate the series in (2.1). By choosing N terms of the series (2.1), and ignoring the truncation error, we can approximate the kernel as

$$K(\mathbf{x}, \mathbf{x}_j) = \sum_{n=1}^{N} \lambda_n \varphi_n(\mathbf{x}) \varphi_n(\mathbf{x}_j).$$

Therefore the interpolant s_f becomes

$$s_f(\mathbf{x}) = \sum_{j=1}^{N} c_j \sum_{n=1}^{N} \lambda_n \varphi_n(\mathbf{x}) \varphi_n(\mathbf{x}_j) = \mathbf{V}_\Phi^T(\mathbf{x}) \ \mathbf{\Lambda}_N \ \mathbf{\Phi}_\mathbf{X} \ \mathbf{c},$$

where $\mathbf{V}_\Phi^T(\mathbf{x}) = (\varphi_1(\mathbf{x}), \ldots, \varphi_N(\mathbf{x}))$,

$$\mathbf{\Lambda}_N = \begin{bmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_N \end{bmatrix}, \quad \text{and} \quad \mathbf{\Phi}_\mathbf{X} = \begin{bmatrix} \varphi_1(\mathbf{x}_1) & \cdots & \varphi_1(\mathbf{x}_N) \\ \vdots & & \vdots \\ \varphi_N(\mathbf{x}_1) & \cdots & \varphi_N(\mathbf{x}_N) \end{bmatrix}.$$

In the same manner as in [4, 20], we can show that

$$s_f(\mathbf{x}) = \mathbf{V}_\Phi^T(\mathbf{x}) \ \mathbf{\Phi}_\mathbf{X}^{-T} \ \mathbf{f}. \tag{2.2}$$

For small values of the Gaussian shape parameter ε, the eigenvalues λ_n decrease toward zero [18] rapidly, and this causes the system to become ill-conditioned. So that $\mathbf{\Lambda}_N$, which depends on eigenvalues, is eliminated. We can conclude that one of the source of ill-conditioning is removed.

We consider the one-dimensional Gaussian RBF, which is a positive definite kernel. Based on Mercer's theorem

$$\mathrm{e}^{-\varepsilon^2(x-z)^2} = \sum_{n=1}^{\infty} \lambda_n \varphi_n(x) \varphi_n(z),$$

where the φ_n are orthogonal functions with respect to the weight function $\rho(x) = \frac{\alpha}{\sqrt{\pi}}\mathrm{e}^{-\alpha^2 x^2}$,

$$\varphi_n(x) = \sqrt{\beta}\mathrm{e}^{-\delta^2 x^2}\widetilde{H}_{n-1}(\alpha\beta x), \tag{2.3}$$

and $\widetilde{H}_n(x)$ are normalized Hermite polynomials (see [7, 20]). Also

$$\beta = \left(1 + \frac{4\varepsilon^2}{\alpha^2}\right)^{\frac{1}{4}}, \quad \delta^2 = \frac{\alpha^2}{2}(\beta^2 - 1),$$

and the eigenvalues λ_n are given by

$$\lambda_n = \sqrt{\frac{\alpha^2}{\alpha^2 + \varepsilon^2 + \delta^2}}\left(\frac{\varepsilon^2}{\alpha^2 + \varepsilon^2 + \delta^2}\right)^{n-1}, \quad n = 1, 2, \ldots.$$

By using (2.3), the vector function $\mathbf{V}_\Phi(x)$ and the matrix $\mathbf{\Phi}_X$ can be decomposed as

$$\mathbf{V}_\Phi(x) = \sqrt{\beta}\ \mathrm{e}^{-\delta^2 x^2}\ \mathbf{V}_H(x), \tag{2.4}$$

and

$$\mathbf{\Phi}_X = \sqrt{\beta}\ \mathbf{H}_X\ \mathbf{D}_X, \tag{2.5}$$

where

$$\mathbf{V}_H(x) = \begin{bmatrix} \widetilde{H}_0(\alpha\beta x) \\ \vdots \\ \widetilde{H}_{N-1}(\alpha\beta x) \end{bmatrix}, \qquad \mathbf{H}_X = \begin{bmatrix} \widetilde{H}_0(\alpha\beta x_1) & \cdots & \widetilde{H}_0(\alpha\beta x_N) \\ \vdots & & \vdots \\ \widetilde{H}_{N-1}(\alpha\beta x_1) & \cdots & \widetilde{H}_{N-1}(\alpha\beta x_N) \end{bmatrix},$$

and

$$\mathbf{D}_X = \begin{bmatrix} \mathrm{e}^{-\delta^2 x_1^2} & & 0 \\ & \ddots & \\ 0 & & \mathrm{e}^{-\delta^2 x_N^2} \end{bmatrix}.$$

Since the Hermite polynomials values can grow dramatically, the algorithm for evaluation of Gaussian RBFs can become unstable. Therefore in the following we show that the eigenfunctions can be rebuilt according to any other orthogonal polynomials, like Chebyshev polynomials, which are more stable for numerical computation. Let $\{p_n(x)\}_{n=0}^{\infty}$ be a family of polynomials, since $\{p_0, \cdots, p_n\}$ is a basis for π_n (the space of all polynomials of degree at most n), $\widetilde{H}_n(\alpha\beta x)$ can be presented as a linear combination of its members as follows:

$$\widetilde{H}_n(\alpha\beta x) = \sum_{k=0}^{n} c_{n,k} p_k(x), \tag{2.6}$$

so

$$\begin{bmatrix} \widetilde{H}_0(\alpha\beta x) \\ \widetilde{H}_1(\alpha\beta x) \\ \vdots \\ \widetilde{H}_{N-1}(\alpha\beta x) \end{bmatrix} = \begin{bmatrix} c_{00} & & & \\ c_{10} & c_{11} & 0 & \\ \vdots & \vdots & \ddots & \\ c_{N-1,0} & c_{N-1,1} & \cdots & c_{N-1,N-1} \end{bmatrix} \begin{bmatrix} p_0(x) \\ p_1(x) \\ \vdots \\ p_{N-1}(x) \end{bmatrix}.$$

The above formula can be written in matrix-vector form as $\mathbf{V}_H(x) = \mathbf{C}\ \mathbf{V}_p(x)$ and consequently $\mathbf{H}_X = \mathbf{C}\ \mathbf{P}_X$ where

$$\mathbf{P}_X = \begin{bmatrix} p_0(x_1) & \cdots & p_0(x_N) \\ \vdots & & \vdots \\ p_{N-1}(x_1) & \cdots & p_{N-1}(x_N) \end{bmatrix},$$

is a polynomial Vandermonde-type matrix [5, 12].

Therefore, from the relations (2.2), (2.4), and (2.5), we have

$$\begin{aligned} s_f(x) &= \sqrt{\beta}\ \mathrm{e}^{-\delta^2 x^2}\ \mathbf{V}_H^T(x)\ \frac{1}{\sqrt{\beta}}\mathbf{H}_X^{-T}\ \mathbf{D}_X^{-1}\ \mathbf{f} \\ &= \mathrm{e}^{-\delta^2 x^2}\ \mathbf{V}_p^T(x)\ \mathbf{C}^T\ \mathbf{C}^{-T}\ \mathbf{P}_X^{-T}\ \mathbf{D}_X^{-1}\ \mathbf{f} \\ &= \mathrm{e}^{-\delta^2 x^2}\ \mathbf{V}_p^T(x)\ \mathbf{P}_X^{-T}\ \mathbf{D}_X^{-1}\ \mathbf{f}, \end{aligned}$$

so we rebuild eigenfunctions according to $p_n(x)$, consequently we rebuild $\varphi_n(x)$ as

$$\varphi_n(x) = \sqrt{\beta}\mathrm{e}^{-\delta^2 x^2} p_{n-1}(x).$$

Now we can generalize the above discussion for two-dimensional interpolation formula for $f : [a,b] \times [c,d] \to \mathbb{R}$. Let $Y = \{y_1, \cdots, y_M\}$ be a set of arbitrary grid points on $[c,d]$ and $\bar{\Omega} := X \times Y = \{(x_i, y_j) :\ i = 1,2,\cdots,N,\ j = 1,2,\cdots,M\}$ be the tensor product grid points on $[a,b]\times[c,d]$. By using the tensor product form of the Gaussian kernel, we have

$$s_f(x,y) = \sum_{j=1}^{N} c_j \sum_{n=1}^{N}\sum_{m=1}^{M} \lambda_m\lambda_n\varphi_n(x)\varphi_n(x_j)\varphi_m(y)\varphi_m(y_j),$$

where

$$\varphi_m(y) = \sqrt{\beta}\ \mathrm{e}^{-\delta^2 y^2} p_{m-1}(x), \qquad m = 1,\cdots,M, \qquad y \in [c,d].$$

Just as in [20], we can show that

$$s_f(x,y) = \mathbf{V}_\Phi^T(x)\ \mathbf{\Phi}_X^{-T}\ \mathbf{F}\ \mathbf{\Phi}_Y^{-1}\ \mathbf{V}_\Phi(y), \tag{2.7}$$

where

$$\mathbf{F} = \begin{bmatrix} f(x_1,y_1) & \cdots & f(x_1,y_M) \\ \vdots & & \vdots \\ f(x_N,y_1) & \cdots & f(x_N,y_M) \end{bmatrix}.$$

Similarly, by using (2.4) and (2.5), we can decompose $\mathbf{V}_\Phi(y)$ and $\mathbf{\Phi}_Y$ as

$$\mathbf{V}_\Phi(y) = \sqrt{\beta}\ \mathrm{e}^{-\delta^2 y^2}\mathbf{V}_P(y),$$

and

$$\mathbf{\Phi}_Y = \sqrt{\beta}\ \mathbf{P}_Y\ \mathbf{D}_Y.$$

By these decompositions, the formula (2.7) can be obtained as:

$$s_f(x,y) = \mathrm{e}^{-\delta^2(x^2+y^2)}\ \mathbf{V}_P^T(x)\ \mathbf{P}_X^{-T}\ \bar{\mathbf{F}}\ \mathbf{P}_Y^{-1}\ \mathbf{V}_P(y), \tag{2.8}$$

where $\bar{\mathbf{F}} = \mathbf{D}_X^{-1}\ \mathbf{F}\ \mathbf{D}_Y^{-1}$.

Remark 1. *Up until now the polynomials p_n were arbitrary. In particular we can, use Chebyshev polynomials (also shifted Chebyshev polynomials) instead of Hermite polynomials. Therefore, in this paper, when we approximate $f : [a,b] \to \mathbb{R}$ by Gaussian RBFs, the $\varphi_n(x)$'s are modified as*

$$\varphi_n(x) = \sqrt{\beta}\ \mathrm{e}^{-\delta^2 x^2}\widehat{T}_{n-1}(\alpha_1 x + \alpha_2), \qquad x \in [a,b],$$

where

$$\widehat{T}_n(x) = \begin{cases} \frac{2}{\sqrt{N}}, & n = 0, \\ \sqrt{\frac{2}{N}}T_n(x), & n \geq 1, \end{cases}$$

and the parameters α_1, α_2, which control the Chebyshev polynomials, can be chosen arbitrary. While if we choose $\alpha_1 = \frac{2}{b-a}$, $\alpha_2 = -\frac{a+b}{b-a}$, then the shifted Chebyshev polynomials become bounded such that $|T_n(\alpha_1\ x + \alpha_2)| \leq 1$ for $x \in [a,b]$.

3. Implementation of the method

In this section, we want to solve two dimensional Helmholtz equation (1.1) on $\Omega = [a,b] \times [c,d]$ by a collocation method based on Gaussian eigenfunctions. The boundary conditions (1.2) can be considered as

$$\begin{cases} \alpha_1 u(a,y)\ + \beta_1 u_x(a,y)\ = h_1(a,y), \\ \alpha_2 u(b,y) + \beta_2 u_x(b,y) = h_2(b,y), \\ \alpha_3 u(x,c)\ + \beta_3 u_y(x,c)\ = h_3(x,c), \\ \alpha_4 u(x,d) + \beta_4 u_y(x,d) = h_4(x,d), \end{cases} \tag{3.1}$$

Suppose that the approximate solution of (1.1) is

$$U(x,y) = \mathbf{V}_\Phi^T(x)\ \mathbf{\Phi}_X^{-T}\ \mathbf{U}\ \mathbf{\Phi}_Y^{-1}\ \mathbf{V}_\Phi(y), \tag{3.2}$$

where $[\mathbf{U}]_{ij} = U(x_i, y_j)$. For solving the equation and illustrating the algorithm, it is necessary to decompose and rearrange the matrix $\mathbf{U}$ as

$$\mathbf{U} = \left[\begin{array}{cc|ccc} u_{11} & u_{1M} & u_{12} & \cdots & u_{1,M-1} \\ u_{N,1} & u_{N,M} & u_{N2} & \cdots & u_{N,M-1} \\ \hline u_{21} & u_{2,M} & u_{22} & \cdots & u_{2,M} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ u_{N-1,1} & u_{N-1,M} & u_{N-1,2} & \cdots & u_{N-1,M-1} \end{array}\right] = \left[\begin{array}{c|c} \mathbf{U}_{\mathcal{B}}^{cor} & \mathbf{U}_{\mathcal{B}}^{row} \\ \hline \mathbf{U}_{\mathcal{B}}^{col} & \mathbf{U}_{\mathcal{I}} \end{array}\right].$$

In fact, we decompose $\mathbf{U}$ to four matrices $\mathbf{U}_{\mathcal{B}}^{cor}$, $\mathbf{U}_{\mathcal{B}}^{row}$, $\mathbf{U}_{\mathcal{B}}^{col}$, $\mathbf{U}_{\mathcal{I}}$ which are the four corners, the up and down boundaries, the left and right boundaries, and the interior entries of $\mathbf{U}$, respectively. We should mention that it is necessary to rearangement the other matrices in the formula (3.2) as $\mathbf{U}$, to avoid deformation of that. For the Dirichlet boundary conditions, all of the entries along the boundary of the matrix $\mathbf{U}$ are known, so it is necessary to obtain only $\mathbf{U}_{\mathcal{I}}$ by simplifying the equation with some matrix algebraic operations. For the Neumann boundary conditions, the problem becomes a little complicated.

Collocating the equation at the interior points and using the boundary data at the boundary points leads to the following system of equations

$$\begin{aligned} \mathbf{V}_{\Phi}^{''T}(x_i) \quad & \mathbf{\Phi}_X^{-T}\ \mathbf{U}\ \mathbf{\Phi}_Y^{-1}\ \mathbf{V}_{\Phi}(y_j) + \mathbf{V}_{\Phi}^{T}(x_i)\ \mathbf{\Phi}_X^{-T}\ \mathbf{U}\ \mathbf{\Phi}_Y^{-1}\ \mathbf{V}_{\Phi}''(y_j) \\ + \quad & k^2\ \mathbf{U} = f(x_i, y_j), \end{aligned} \tag{3.3}$$

for $1 < i < N,\ 1 < j < M$, associated with boundary conditions

$$\begin{cases} \alpha_1 u(x_1, y_j) + \beta_1 u_x(x_1, y_j) = h_1(x_1, y_j), \\ \alpha_2 u(x_N, y_j) + \beta_2 u_x(x_N, y_j) = h_2(x_N, y_j), \qquad i = 2, \cdots, N-1, \\ \alpha_3 u(x_i, y_1) + \beta_3 u_y(x_i, y_1) = h_3(x_i, y_1), \\ \alpha_4 u(x_i, y_M) + \beta_4 u_y(x_i, y_M) = h_4(x_i, y_M), \qquad j = 1, \cdots, M. \end{cases} \tag{3.4}$$

By denoting the following

$$\widetilde{\mathbf{\Psi}}_X^T = \begin{bmatrix} \mathbf{V}_{\Phi}^{''T}(x_2) \\ \vdots \\ \mathbf{V}_{\Phi}^{''T}(x_{N-1}) \end{bmatrix}, \quad \widetilde{\mathbf{\Phi}}_Y = \begin{bmatrix} \mathbf{V}_{\Phi}(y_2) & \cdots & \mathbf{V}_{\Phi}(y_{M-1}) \end{bmatrix},$$

and $\mathbf{A} = [a_{ij}] = \widetilde{\mathbf{\Psi}}_X^T\ \mathbf{\Phi}_X^{-T}$, we consider the first term on the left-hand side of (3.3). In the same manner as in [20] we can show that

$$\widetilde{\mathbf{\Psi}}_X^T\ \mathbf{\Phi}_X^{-T}\ \mathbf{U}\ \mathbf{\Phi}_Y^{-1}\ \widetilde{\mathbf{\Phi}}_{\mathbf{Y}}$$

$$= \underbrace{\begin{bmatrix} a_{1,1} & a_{1,N} \\ a_{2,1} & a_{2,N} \\ \vdots & \vdots \\ a_{N-2,1} & a_{N-2,N} \end{bmatrix}}_{=\mathbf{A}_{\mathcal{B}}} \underbrace{\begin{bmatrix} u_{1,2} & u_{1,3} & \cdots & u_{1,M-1} \\ u_{N,2} & u_{N,3} & \cdots & u_{N,M-1} \end{bmatrix}}_{=\mathbf{U}_{\mathcal{B}}^{row}}$$

$$+\underbrace{\begin{bmatrix} a_{1,2} & \cdots & a_{1,N-1} \\ a_{2,2} & \cdots & a_{2,N-1} \\ \vdots & & \vdots \\ a_{N-2,2} & \cdots & a_{N-2,N-1} \end{bmatrix}}_{=\mathbf{A}_{\mathcal{I}}} \underbrace{\begin{bmatrix} u_{2,2} & u_{2,3} & \cdots & u_{2,M-1} \\ \vdots & \vdots & & \vdots \\ u_{N-1,2} & u_{N-1,3} & \cdots & u_{N-1,M-1} \end{bmatrix}}_{=\mathbf{U}_{\mathcal{I}}}. \tag{3.5}$$

Using the first two boundary conditions, we have

$$\underbrace{\begin{bmatrix} \alpha_1 & 0 \\ 0 & \alpha_2 \end{bmatrix}}_{=\alpha_{12}} \begin{bmatrix} \mathbf{U}_1^{row} \\ \\ \mathbf{U}_N^{row} \end{bmatrix} + \underbrace{\begin{bmatrix} \beta_1 & 0 \\ 0 & \beta_2 \end{bmatrix}}_{=\beta_{12}} \tilde{\mathbf{A}}\ \mathbf{U} = \begin{bmatrix} h_1(x_1, y_1) & \cdots & h_1(x_1, y_M) \\ h_2(x_N, y_1) & \cdots & h_2(x_N, y_M) \end{bmatrix},$$

where

$$\tilde{\mathbf{A}} = \begin{bmatrix} \mathbf{V}_{\Phi}'^{T}(x_1) \\ \\ \mathbf{V}_{\Phi}'^{T}(x_N) \end{bmatrix} \mathbf{\Phi}_X^{-T}.$$

therefore

$$\alpha_{12} \left[\mathbf{U}_{\mathcal{B}}^{cor} | \mathbf{U}_{\mathcal{B}}^{row}\right] + \beta_{12}\ \tilde{\mathbf{A}}\ \mathbf{U} = \left[\mathbf{H}^{cor} | \mathbf{H}^{row}\right].$$

Now by decomposing $\tilde{\mathbf{A}}$ and $\mathbf{U}$, we want to obtain $\mathbf{U}_{\mathcal{B}}^{row}$ and $\mathbf{U}_{\mathcal{B}}^{cor}$, in fact

$$\begin{aligned} \tilde{\mathbf{A}}\ \mathbf{U} &= \left[\ \tilde{\mathbf{A}}_{\mathcal{B}} \ \middle|\ \tilde{\mathbf{A}}_{\mathcal{I}}\ \right] \left[\begin{array}{c|c} \mathbf{U}_{\mathcal{B}}^{cor} & \tilde{\mathbf{U}}_{\mathcal{B}}^{row} \\ \hline \mathbf{U}_{\mathcal{B}}^{col} & \mathbf{U}_{\mathcal{I}} \end{array}\right] \\ &= \left[\ \tilde{\mathbf{A}}_{\mathcal{B}}\ \mathbf{U}_{\mathcal{B}}^{cor} + \mathbf{A}_{\mathcal{I}}\ \mathbf{U}_{\mathcal{B}}^{col} \ \middle|\ \tilde{\mathbf{A}}_{\mathcal{B}}\ \mathbf{U}_{\mathcal{B}}^{row} + \tilde{\mathbf{A}}_{\mathcal{I}}\ \mathbf{U}_{\mathcal{I}}.\ \right] \end{aligned}$$

Now by using two above equations

$$\alpha_{12}\ \mathbf{U}_{\mathcal{B}}^{cor} + \beta_{12}\ \left[\tilde{\mathbf{A}}_{\mathcal{B}}\ \mathbf{U}_{\mathcal{B}}^{cor} + \tilde{\mathbf{A}}_{\mathcal{I}}\ \mathbf{U}_{\mathcal{B}}^{col}\right] = \mathbf{H}^{cor},$$

and

$$\alpha_{12}\ \mathbf{U}_{\mathcal{B}}^{row} + \beta_{12}\ \left[\tilde{\mathbf{A}}_{\mathcal{B}}\ \mathbf{U}_{\mathcal{B}}^{row} + \tilde{\mathbf{A}}_{\mathcal{I}}\ \mathbf{U}_{\mathcal{I}}\right] = \mathbf{H}^{row}.$$

So by denoting $\mathbf{K}_1 = \alpha_{12} + \beta_{12}\ \tilde{\mathbf{A}}_{\mathcal{B}}$, we have

$$\mathbf{U}_{\mathcal{B}}^{cor} = \mathbf{K}_1^{-1}\left[\mathbf{H}^{cor} - \beta_{12}\ \tilde{\mathbf{A}}_{\mathcal{I}}\ \mathbf{U}_{\mathcal{B}}^{col}\right], \tag{3.6}$$

and

$$\mathbf{U}_{\mathcal{B}}^{row} = \mathbf{K}_1^{-1}\left[\mathbf{H}^{row} - \beta_{12}\ \tilde{\mathbf{A}}_{\mathcal{I}}\ \mathbf{U}_{\mathcal{I}}\right]. \tag{3.7}$$

Now by denoting

$$\widetilde{\mathbf{\Phi}}_X^T = \begin{bmatrix} \mathbf{V}_{\Phi}^T(x_2) \\ \vdots \\ \mathbf{V}_{\Phi}^T(x_{N-1}) \end{bmatrix}, \quad \widetilde{\mathbf{\Psi}}_Y = \begin{bmatrix}\ \mathbf{V}_{\Phi}''(y_2) & \cdots & \mathbf{V}_{\Phi}''(y_{M-1})\ \end{bmatrix},$$

and $\mathbf{B} = [b_{ij}] = \mathbf{\Phi}_Y^{-1}\ \widetilde{\mathbf{\Psi}}_Y$, we can similarly obtained

$$\widetilde{\mathbf{\Phi}}_X^T\ \mathbf{\Phi}_X^{-T}\ \mathbf{U}\ \mathbf{\Phi}_Y^{-1}\ \widetilde{\mathbf{\Psi}}_Y = \underbrace{\begin{bmatrix} u_{2,1} & u_{2,M} \\ u_{3,1} & u_{3,M} \\ \vdots & \vdots \\ u_{N-1,1} & u_{N-1,M} \end{bmatrix}}_{=\mathbf{U}_{\mathcal{B}}^{col}} \underbrace{\begin{bmatrix} b_{1,1} & b_{1,2} & \cdots & b_{1,M-2} \\ b_{M,1} & b_{M,2} & \cdots & b_{M,M-2} \end{bmatrix}}_{=\mathbf{B}_{\mathcal{B}}}$$

$$+ \underbrace{\begin{bmatrix} u_{2,2} & u_{2,3} & \cdots & u_{2,M-1} \\ \vdots & \vdots & & \vdots \\ u_{N-1,2} & u_{N-1,3} & \cdots & u_{N-1,M-1} \end{bmatrix}}_{=\mathbf{U}_{\mathcal{I}}} \underbrace{\begin{bmatrix} b_{2,1} & b_{2,2} & \cdots & b_{2,M-2} \\ \vdots & \vdots & & \vdots \\ b_{M-1,1} & b_{M-1,2} & \cdots & b_{M-1,M-2} \end{bmatrix}}_{=\mathbf{B}_{\mathcal{I}}}, \quad (3.8)$$

and

$$\mathbf{U}_{\mathcal{B}}^{col} = \left[\mathbf{H}^{col} - \mathbf{U}_{\mathcal{I}}\ \tilde{\mathbf{B}}_{\mathcal{I}}\ \beta_{34}\right] \mathbf{K}_2^{-1}, \quad (3.9)$$

where $\mathbf{K}_2 = \alpha_{34} + \tilde{\mathbf{B}}_{\mathcal{B}}\ \beta_{34}$.

Now, by substituting (3.7) in (3.5), and (3.9) in (3.8), we obtain the following Sylvester system

$$\mathbf{A}\ \mathbf{U}_{\mathcal{I}} + \mathbf{U}_{\mathcal{I}}\ \mathbf{B} + \mathbf{C} = \mathbf{0}, \quad (3.10)$$

where

$$\begin{aligned} \mathbf{A} &= \mathbf{A}_{\mathcal{I}} + \frac{k^2}{2}\mathbf{I}, \\ \mathbf{B} &= \mathbf{B}_{\mathcal{I}} + \frac{k^2}{2}\mathbf{I}, \\ \mathbf{C} &= \mathbf{A}_{\mathcal{B}}\ \mathbf{G}_{\mathcal{B}}^{row} + \mathbf{G}_{\mathcal{B}}^{col}\ \mathbf{B}_{\mathcal{B}} - \widetilde{\mathbf{F}}, \end{aligned}$$

and $[\widetilde{\mathbf{F}}]_{i,j} = f(x_i, y_j), \qquad i = 2, \cdots, N,\ j = 2, \cdots, M.$

The equation (3.10) can be solved in MATLAB by using the command `sylvester`, which first transforms the $\mathbf{A}$ and $\mathbf{B}$ matrices to complex Schur form, then computes the solution of the resulting triangular system and finally transforms the solution back.

After solving (3.10), $\mathbf{U}_{\mathcal{I}}$ will be obtained (the solution in interior points), then by using (3.7) and (3.9), $\mathbf{U}_{\mathcal{B}}^{row}$ and $\mathbf{U}_{\mathcal{B}}^{col}$ will be obtained, and finally by using (3.6), $\mathbf{U}_{\mathcal{B}}^{cor}$ will be obtained. Also by using (3.2), we can obtain a closed form solution for the Helmholtz equation.

4. Numerical Results

In this section, several numerical experiments are reported to illustrate the accuracy and efficiency of the proposed approach to solve two-dimensional Helmholtz equations. The numerical experiments are performed in MATLAB 2014 on PC computer with an

TABLE 1. The local and global error for different values of grid points for Example 1.

$N \times M$	Uniform points		Chebyshev points	
	Local error	Global error	Local error	Global error
11×11	3.79×10^{-02}	4.02×10^{-01}	3.09×10^{-04}	1.42×10^{-02}
13×13	1.97×10^{-03}	1.72×10^{-02}	5.84×10^{-06}	2.99×10^{-05}
15×15	5.14×10^{-05}	3.85×10^{-04}	6.20×10^{-08}	3.41×10^{-06}
17×17	3.72×10^{-07}	2.45×10^{-06}	2.12×10^{-10}	1.33×10^{-08}
19×19	1.40×10^{-08}	7.98×10^{-08}	2.21×10^{-12}	1.03×10^{-10}
21×21	3.25×10^{-09}	3.37×10^{-08}	1.38×10^{-13}	1.37×10^{-12}
23×23	1.78×10^{-07}	7.22×10^{-07}	1.43×10^{-13}	8.47×10^{-13}

Intel(R) Core(TM) i5-4460 processor (3.20GHz CPU), a 64-bit Windows 7 operating system, and a 16 GB internal memory . In all examples, we used $\varepsilon = 1$, $\alpha = 1.9$ and we used uniform grid points and Chebyshev collocation points.

The accuracy is measured by computing the local and global error of

$$\|u - U\|_{\infty,\bar{\Omega}} = \max_{(x_i,y_j)\in\bar{\Omega}} |u(x_i, y_j) - U_{i,j}|,$$

and

$$\|u - U\|_{\infty,\Omega} = \max_{(x,y)\in\Omega} |u(x, y) - U(x, y)|,$$

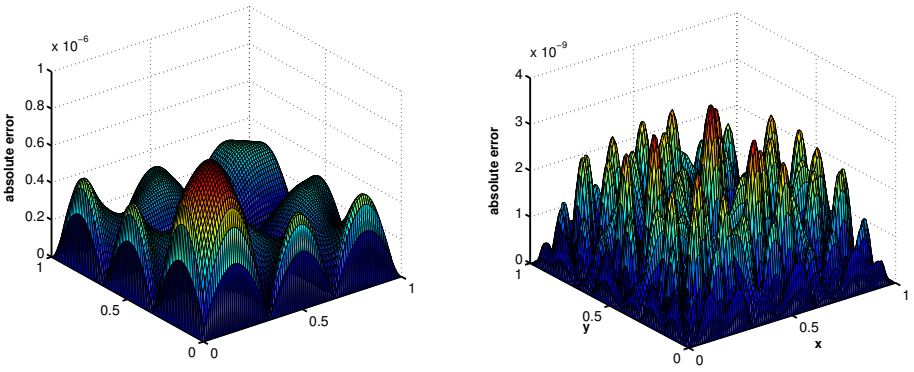
where the second norm is approximated on $300\times(b_1-a_1+b_2-a_2)$ evenly spaced points.

Example 1. Consider the following two-dimensional Helmholtz equation

$$\begin{cases} \frac{\partial^2 u(x,y)}{\partial x^2} + \frac{\partial^2 u(x,y)}{\partial y^2} + 9\pi^2\, u(x, y) = -9\pi^2 \sin(3\pi x) \sin(3\pi y), & (x, y) \in \Omega \\ u(x, y) = 0, & (x, y) \in \partial\Omega \end{cases}$$

where $\Omega = [0, 1] \times [0, 1]$ and the exact solution is $u(x, y) = \sin(3\pi x) \sin(3\pi y)$. We applied our method to this example with different values of uniform and Chebyshev grid points. The arrising Sylvester system can be solved and the numerical solution of this example can be obtained. The local and global errors are tabulated in Table 1. Also, absolute error has been plotted for 17×17 space grid points in Figure 1. From the data in Table 1 we can see that Chebyshev grid points yield a smaller error than uniform grid points.

FIGURE 1. Error with the 17×17 uniform grid point (left) and Chebyshev grid points (right) for Example 1.



Example 2. Consider the following two-dimensional Helmholtz equation from [15]

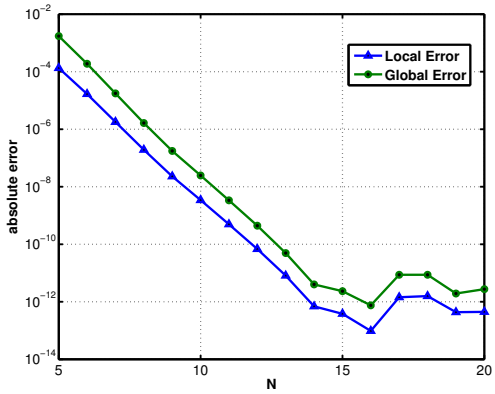
$$\begin{cases} \frac{\partial^2 u(x,y)}{\partial x^2} + \frac{\partial^2 u(x,y)}{\partial y^2} + 2\,u(x,y) = 2x - 4y, & \text{in } \Omega = [0,1] \times [0,1] \\ u_y(x,y) = h_2(x,y), & \text{on } \Gamma_2 = [0,1] \times \{0\} \\ u(x,y) = h_1(x,y), & \text{on } \Gamma_1 = \partial\Omega \backslash \Gamma_2 \end{cases}$$

where the exact solution is $u(x,y) = \sin(\sqrt{3}x)\sinh(y) + \cos(\sqrt{2}y) + x - 2y$, and $h_1(x,y)$, $h_2(x,y)$ are chosen so as comply with the exact solution. The solution of this equation is obtained by solving the Sylvester system of (3.10). We use uniform and Chebyshev grid points. In Table 2, the local and global errors are shown for different values of uniform and Chebyshev grid points. In Figure 2, absolute error is shown as function of N for Chebyshev grid points. It can be seen that spectral convergence can be achieved for $N \leq 16$. For larger N, by using higher precision, like Multiprecision Computing Toolbox by ADVANPIX, spectral convergence can be obtained. Also in Figure 3, the absolute error is plotted versus the number of grid points ($N \times M$) for our method and traditional MQ RBf method. Superiority of our method with respect to traditional MQ RBf method is clear.

TABLE 2. The local and global error for different values of grid points for Example 2.

$N \times M$	Uniform points			Chebyshev points		
	Local error	Global error	Runtime	Local error	Global error	Runtime
11×11	4.37×10^{-05}	1.89×10^{-02}	0.1824	1.34×10^{-04}	1.74×10^{-03}	0.1895
13×13	1.39×10^{-06}	4.12×10^{-04}	0.1813	1.79×10^{-06}	1.76×10^{-05}	0.1937
15×15	8.45×10^{-08}	1.09×10^{-05}	0.1776	2.96×10^{-08}	1.75×10^{-07}	0.1865
17×17	3.32×10^{-09}	5.94×10^{-07}	0.1779	4.94×10^{-10}	3.37×10^{-09}	0.1910
19×19	1.69×10^{-09}	2.06×10^{-08}	0.1804	8.10×10^{-12}	4.94×10^{-11}	0.1927
21×21	2.56×10^{-09}	9.37×10^{-09}	0.1795	3.81×10^{-13}	2.33×10^{-12}	0.1953
23×23	6.71×10^{-08}	1.38×10^{-08}	0.1875	1.43×10^{-12}	8.68×10^{-12}	0.1905

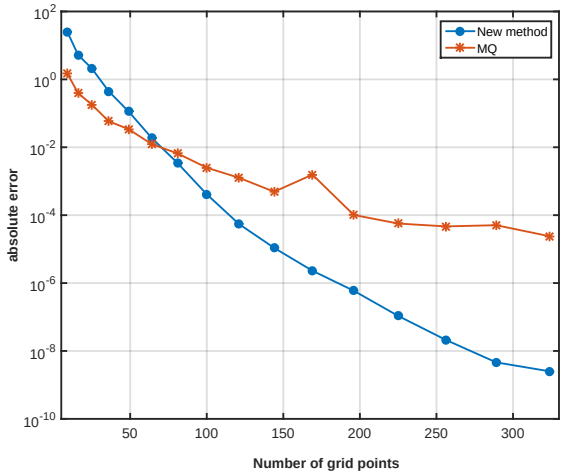
FIGURE 2. Absolute errors as function of N for Chebyshev grid points for Example 2.



5. CONCLUSION

This paper enhanced a stable method based on an eigenfunction expansion of the Gaussian RBF which is applied to solving two-dimensional Helmholtz equations. The eigenfunction expansions are rebuilt based on Chebyshev polynomials which are more suitable in numerical computation. After discretization of the equation, we obtain a Sylvester system and solve it. Employing the Chebyshev basis allows our approach to become accurate and stable. Moreover, the two-dimensional formula (3.2) which is presented in [20] reduces the computational complexity for solving two-dimensional Helmholtz equations.

FIGURE 3. Absolute errors as function of N for Chebyshev grid points for Example 2.



REFERENCES

[1] K. Atkinson, O. Hansen, and D. Chien, *A spectral method for elliptic equations: The Neumann problem*, Adv. Comput. Math., *34*(3) (2011), 295–317.

[2] I. Babuska, F. Ihlenburg, E. T. Paik, and S. A. Sauter, *A generalized finite element method for solving the helmholtz equation in two dimensions with minimal pollution*, Comput. Method. Appl. Mech. engg., *128*(3-4) (1995), 325–359.

[3] M. D. Buhmann, *Radial basis functions: theory and implementations*, Cambridge university press, *12* (2003).

[4] R. Cavoretto, G. E. Fasshauer, and M. Mccourt, *An introduction to the Hilbert-Schmidt SVD using iterated Brownian bridge kernels*, Numer. Algor., *68*(2) (2015), 393-422.

[5] J. Demmel and P. Koev, *Accurate SVDs of polynomial Vandermonde matrices involving orthonormal polynomials*, Lin. Algeb. Applic., *417*(2-3) (2006), 382–396.

[6] G. E. Fasshauer, *Meshfree approximation methods with Matlab:(With CD-ROM)*, World Scientific Publishing Co Inc, *6*, (2007).

[7] G. E. Fasshauer and M. J. Mccourt, *Gaussian radial basis function interpolants*, SIAM J. Sc. Comput., *34*(2) (2012), A737–A762.

[8] B. Fornberg and G. Wright, *Stable computation of multiquadric interpolants for all values of the shape parameter*, Comput. Math. Appli., *48*(5) (2004), 853–867.

[9] B. Fornberg and N. Flyer, *Accuracy of radial basis function interpolation and derivative approximations on 1-D infinite grids*, Adv. Comput. Math., *23*(1-2) (2005), 5–20.

[10] B. Fornberg and C. Piret, *A stable algorithm for flat radial basis functions on a sphere*, SIAM J. Sc. Comput., *30*(1) (2007), 60–80.
[11] B. Fornberg, E. Larsson, and N. Flyer, *Stable computations with Gaussian radial basis functions*, SIAM J. Sc. Comput., *33*(2) (2011), 869–892.
[12] W. Gautschi, *The condition of Vandermonde-like matrices involving orthogonal polynomials*, Lin. Algeb. Appl., *52* (1983), 293–300.
[13] A. Handlovičová and I. Riečanová, *Numerical solution to the complex 2D Helmholtz equation based on finite volume method with impedance boundary conditions*, Open Phy., *14*(1) (2016), 436–443.
[14] E. Larsson, E. Lehto, A. Heryudono, and B. Fornberg, *Stable computation of differentiation matrices and scattered node stencils based on Gaussian radial basis functions*, SIAM J. Sc. Comput., *35*(4) (2013), A2096–A2119.
[15] , J. Lin, W. Chen, and K. Y. Sze, *A new radial basis function for Helmholtz problems*, Eng. Anal. Bound. Elem., *36*(12) (2012), 1923–1930.
[16] P. Martinsson, *A direct solver for variable coefficient elliptic PDEs discretized via a composite spectral collocation method*, J. Comput. Phy., *242* (2013), 460–479.
[17] A. T. Patera, *A spectral element method for fluid dynamics: laminar flow in a channel expansion*, J. Comput. Phy., *54*(3) (1984), 468–488.
[18] M. Pazouki and R. Schaback, *Bases for kernel-based spaces*, J. Comput. Appl. Math., *236*(4) (2011), 575–588.
[19] C. Pozrikidis, *Introduction to finite and spectral element methods using MATLAB*, CRC Press, 2005.
[20] J. Rashidinia, G. E. Fasshauer, and M. Khasi, *A stable method for the evaluation of Gaussian radial basis function solutions of interpolation and collocation problems*, Comput. Math. Appl., *72*(1) (2016), 178–193.
[21] I. Riečanová, *Finite volume method scheme for the solution of Helmholtz equation*, Proceedings of Aplimat, 14-th Conference on Applied Mathematics, Institute of Mathematics and Physics, Faculty of Mechanical Engineering STU in Bratislava, 2015.
[22] S. A. Sauter and C. Schwab, *Boundary element methods*, Springer-Verlag, Berlin, 2010.
[23] R. Schaback, *A practical guide to radial basis functions*,Electronic Resource, *11*, 2007.
[24] E. Schmidt, *Über die Auflösung linearer Gleichungen mit unendlich vielen Unbekannten*, Rendiconti del Circolo Matematico di Palermo (1884-1940), *25*(1) (1908), 53–77.
[25] H. Wendland, *Scattered data approximation*, Cambridge university press, *17* (2004).