International Journal of Engineering Applied Sciences and Technology, 2019
Vol. 4, Issue 2, ISSN No. 2455-2143, Pages 7-15
Published Online June 2019 in IJEAST (http://www.ijeast.com)

RADIAL BASIS FUNCTION METHODS FOR APPROXIMATING THE TWO-DIMENSIONAL HEAT EQUATION

Tersoo Luga
Dept. of Maths/Stat/CS
Federal University of Agriculture
Makurdi, Benue State, Nigeria

Terhemen Aboiyar
Dept. of Maths/Stat/CS
Federal University of Agriculture
Makurdi, Benue State, Nigeria

Solomon O. Adee
Department of Mathematics
Moddibo Adama University of Technology, Yola, Adamawa State, Nigeria

Abstract— **Radial basis function method of lines (RBF-MOLs) for approximating the two-dimensional heat equation were formulated using two globally supported and positive radial basis functions (RBFs), namely, inverse quadratic (IQ), generalized inverse multiquadric (GIMQ) and the fourth order Runge-Kutta method. The RBFs were used for discretizing the space variables while the fourth order Runge-Kutta method was used as a time-stepping method to integrate the resulting systems of ordinary differential equations (ODEs) that emanated from the space discretization. The methods were implemented in MATLAB and compared with the multiquadric radial basis function method of lines (MQ-RBF-MOLs). The performance of the proposed RBF-MOLs was measured in terms of point-wise error and processing time (CPU Time). Our numerical results show that our proposed methods compared favourably with the MQ-RBF-MOLs**

Keywords— **Radial Basis Function, Multiquadrics, Inverse Quadratics, Generalized Inverse Multiquadrics, Positive Definite RBFs, Globally Supported RBFs, Radial Basis Function Method of Lines**

I. INTRODUCTION

Radial basis functions (RBFs) were derived for the purpose of multivariate scattered data interpolation Buhmann [1]. In 1990, Kansa [2, 3] got a breakthrough by developing a multiquadric (MQ) RBF-collocation scheme for approximating the elliptic, parabolic and hyperbolic partial differential equations (PDEs), this methodology is referred to as Kansa's method. The pioneer work of Kansa in RBF paved way for a research boom in RBFs and their numerous applications to PDEs Chen *et al.* [4]. In recent times, RBF methods are applied to numerical solutions of integral equations (IEs), Integro-partial differential equations (IPDEs), plasma fussion simulations, molecular quantum mechanics, cellular biology Kansa and Holoborodko [5], multivariate scattered data processing Iske [6], neural networks Chen *et al.* [7], machine learning Cortes and Vapnik, Burges [8, 9] etc. RBF methods are becoming viable choice numerical methods in different scientific computing communities. They are preferred to other locally based polynomial methods such as the finite difference method (FDM), finite element method (FEM), finite volume method (FVM), boundary element method (BEM) Chen [4], globally based polynomial methods such as pseudospectral methods Sarra and Kansa [10] etc. The reason for the preference of RBF methods over other numerical methods is that besides being mathematically simple, they do not require any mesh generation which is advantageous for application to higher-dimensional problems containing irregular or moving boundary, and, they also have spectral convergence Chen *et al.* [4].

Although the are many modifications and improved RBF-collocation methods that are presented in Chen *et al.* [4], Buhmann [1] G. E. Fasshauer [11] and Chen *et al.* [12], but Kansa's RBF-collocation method still remains one of the commonly used methods. The main advantage of the Kansa's method is that it uses global approximations of the space variables on both the domain and the boundary, this methodology makes the method simple but effective in dealing with higher-dimensional problems with complex domain geometry Yao *et al.* [13].

In many applications, the globally supported RBF methods are used Kansa and Holoborodko [5]. Chen *et al.* [4] highlighted some of the admirable features of the globally supported RBFs as (i) highly accurate and often converge exponentially, (ii) easy to apply to higher dimensional problems (iii) meshless in approximation of multivariate scattered data (iv) easy to improve the numerical accuracy by adding more points around large gradient regions. Most of the globally supported RBFs are continuously differentiable $(C^{\infty} - \text{RBFs})$, these group of RBFs are the best to solve numerically higher dimensional partial differential equations (PDEs) due to some of the reasons, (i) an n-dimensional problem becomes a one-dimensional radial distance problem,

(ii) the convergence rate increases with the dimensionality (iii) they have spectral convergence Kansa and Holoborodko [5].

Among the globally supported RBFs that are continuously differentiable, the generalized multiquadric RBFs (GMQ-RBFs) are commonly used, this can be traced to the work of Franke [14], he used the following criteria: timing, storage, accuracy, visual pleasantness of the surface and the ease of implementation to extensively test 29 different algorithms on some interpolation problems and he ranked the MQ-RBF and the thin plate spline radial basis function (TPS-RBF) as the best candidates. The simplicity of the interpolation matrix of the MQ-RBF makes it more attractive to many applications as compared to the TPS-RBF. Similarly, Kansa and Holoborodko [5] also observed that two commonly used globally-supported and $C^{\infty}-\text{RBFs}$ are the Gaussian and the generalized multiquadrics radial basis function GMQ-RBF.

The Gaussian is defined by

$$\phi(r) = \exp(-(\varepsilon r)^2) \qquad (1)$$

while the GMQ-RBF is defined

$$\phi(r) = (1 + (\varepsilon r)^2)^{\beta}, \qquad \beta = -2, -1, -\frac{1}{2}, \frac{1}{2}, \frac{3}{2}, \frac{5}{2}, \dots \quad (2)$$

where $r = \|\boldsymbol{x} - \boldsymbol{x}_i\|$, ε is called the shape parameter, $\|\cdot\|$ is a norm, usually the Euclidean norm.

The most frequent used values of β are $-\frac{1}{2}, \frac{1}{2}, \frac{3}{2}, \frac{5}{2}, \dots$ Kansa and Holoborodko [5]. RBF methods can either be applied independently or combined with other numerical methods to formulate hybrid numerical methods. One of the prominent numerical methods that combines RBFs and other numerical methods is the method of lines (MOLs) referred to as radial basis function method of lines (RBF-MOLs). This method is constructed by combining an RBF method which is used for space discretization and a time-stepping method which is used for integrating the system of ODEs that emanate from the space discretization. The MOLs is a suitable numerical method for approximating some time-dependent PDEs Sarra and Kansa [10], Fasshauer and McCourt [15] and Bibi [16]. Many researchers Sarra and Kansa [10], Bibi [16] and Luga *et al.* [17] among others have applied the RBF-MOLs for the solution of some time-dependent PDEs. The performance of the RBF-MOLs measured using the point-wise error, compared with other numerical methods show that the RBF-MOLs outperformed the other numerical methods especially in terms of accuracy.

To demonstrate the ease of implementing the RBF-MOLs in higher dimensions and on irregular shaped domains, Sarra and Kansa [10] applied the MQ RBF-MOLs to approximate a two-dimensional heat equation on complex domain, the point-wise error showed that the MQ RBF-MOLs yielded accurate approximations. They also used the finite difference method of lines (FD-MOLs) to approximate the same two-dimensional heat equation and compared the result with MQ RBF-MOLs, the numerical results obtained from the MQ RBF-MOLs was better than that of FD-MOLs when compared to the exact solution.

The motivation of this work comes from Luga *et al* [17]. They formulate the inverse multiquadric (IMQ), IQ, GIMQ RBF-MOLs and applied the methods to approximate some time-dependent PDEs in one-dimension space, their test problems were obtained from Sarra and Kansa [10] and were compared in terms of point-wise error. They observed that their results compared favourably with those obtained from the MQ RBF-MOLs of Sarra and Kansa [10]. In view of the above, we combine (i) the inverse quadratic (IQ) RBF, (ii) the generalized inverse multiquadric (GIMQ) RBF with the fourth order Runge-Kutta method to propose two RBF-MOLs, namely, IQ RBF-MOLs and GIMQ RBF-MOLs for approximating the two-dimensional heat equation. The reason for the choice of these RBF-MOLs is that many researchers have focus on applying equation (2) for the values of $\beta = \frac{1}{2}$ and $-\frac{1}{2}$, recently, Kansa and Holoborodko [5] explained that using the advanpix multi-precision toolbox (AMPT), other values of β such as $\frac{3}{2}, \frac{5}{2}, \dots$ are now becoming viable choice for GMQ RBF methods. However, much is not known about IQ and the GIMQ RBF methods especially the GIMQ RBF method which are defined by equation (2) for the values of $\beta = -1$ and -2. To this end, we wish to explore these RBF methods and apply them to different problems and make comparisons with the frequently used MQ RBF methods.

The rest of the paper is organized as follows: Section II is devoted to developing the RBF-MOLs by first formulating the interpolation matrix, the evaluation matrix and the differentiation matrix of the IQ and GIMQ RBFs. The numerical testing and comparison of results is reported in Section III, discussion in section IV and finally the conclusion is presented in Section V.

II METHODS

The formulation of radial basis function method of lines (RBF-MOLs) for approximating the two-dimensional heat equation is described in this section, much emphasis is paid to the differentiation matrices of the IQ and GIMQ RBF methods which are the main tools for the space discretization. The algorithm for the space discretization using RBF methods and the algorithm for time-stepping using the fourth order Runge-Kutta method are also provided in this section. The radial basis function (RBF) interpolation method which is key to the construction of the MOLs is first explained in details.

Radial Basis Function (RBF) Interpolation Method

Suppose $u|_X = \left(u(x_1), u(x_2), \dots, u(x_N)\right)^T \in \mathbb{R}^d$ is a set of function values sampled from an unknown function $u: \mathbb{R}^d \longrightarrow \mathbb{R}$ at a scattered data set $x = \{x_1, x_2, \dots, x_N\} \subset \mathbb{R}^d, d \geq 1$. To compute an interpolant $s: \mathbb{R}^d \longrightarrow \mathbb{R}$, which is an approximation of $u|_X$ requires using the condition $s|_X = u|_X$ which is satisfied by

International Journal of Engineering Applied Sciences and Technology, 2019
Vol. 4, Issue 2, ISSN No. 2455-2143, Pages 7-15
Published Online June 2019 in IJEAST (http://www.ijeast.com)

$$s(x_i) = u(x_i), \quad \forall\, 1 \le i \le N \tag{2.1}$$

RBF interpolation scheme works with a fixed radial function $\phi: [0, \infty) \longrightarrow \mathbb{R}$, where the interpolant s in equation (2.1) is assumed to be of the form

$$s(x) = \sum_{j=1}^{M} \lambda_j \phi(\|x - x_j\|) + p(x), p \in \mathcal{P}_m^d \tag{2.2}$$

where $\|\cdot\|$ denote the Euclidean norm on $\mathbb{R}^d$ and $\mathcal{P}_m^d$ is the linear space containing all real-valued polynomials in d variables of degree at most $m-1$ respectively. $m \equiv m(\phi)$ is said to be the order of the basis function.

The IQ and GIMQ RBFs are positive definite, thus, the polynomial term in equation (2.2) is not required to make their interpolation matrices invertible Fasshauer [11], hence we shall consider an interpolant of the form

$$s(x) = \sum_{j=1}^{M} \lambda_j \phi(\|x - x_j\|) \tag{2.3}$$

Interpolation or System Matrix of Positive Definite RBF Methods

To form an interpolation matrix for a positive definite RBF method, equation (2.3) is expanded for the data points $i = 1,2, \dots, N$ and centres $j = 1,2, \dots, M$ which results to a linear combination of terms that can be written in the form

$$[A]\lambda_j = u\,, \quad i = 1,2, \dots, N \text{ and } j = 1,2, \dots, M \tag{2.4}$$

where $[A]$ is a matrix with the entries

$$a_{ij} = \phi(\|x_i - x_j\|), \quad i = 1,2, \dots, N \text{ and } j = 1,2, \dots, M \tag{2.5}$$

Interpolation or System Matrix of IQ and GIMQ RBF Methods

The basic functions and the interpolants of the IQ and GIMQ RBF methods provided in Table 1 are used to obtain their various interpolation matrices.

Table 1: The Basic Functions and Interpolants of the IQ and GIMQ RBFs

S/No.	RBF	Basic Function $\phi(r)$	Interpolant $s(x)$
1	Inverse Quadratic (IQ)	$(1+(\varepsilon r)^2)^{-1}$	$\sum_{j=1}^{N} \lambda_j \left(1 + (\varepsilon\|x - x_j\|)^2\right)^{-1}$
3	Generalized Inverse Multiquadrics (GIMQ)	$(1+(\varepsilon r)^2)^{-2}$	$\sum_{j=1}^{N} \alpha_j \left(1 + (\varepsilon\|x - x_j\|)^2\right)^{-2}$

The interpolation matrices with the entries a_{ij} for IQ and GIMQ obtained by applying the interpolation condition (2.1) to the interpolant in Table 1 are provided in equation (2.6)

$$a_{ij} = \left(1 + (\varepsilon\|x - x_j\|)^2\right)^{\beta} \tag{2.6}$$

$i = 1,2, \dots, N$ and $j = 1,2, \dots, M$.

Substituting $\beta = -1$ and -2 gives the interpolation matrices of the IQ and GIMQ RBF respectively.

Evaluation Matrix of Positive Definite RBF Methods

To get the evaluation matrices for IQ and GIMQ RBF methods, $\lambda_j, j = 1,2, \dots, M$ are obtained by solving equation (2.4) from the interpolation matrix, the interpolant (2.3) is evaluated at M points to form an $N \times M$ matrix H, called the evaluation which has the entries

$$h_{ij} = \phi(\|x_i - x_j\|), \tag{2.7}$$

$i = 1,2, \dots, N$ and $j = 1,2, \dots, M$

In this work, we ensure that the number of data point are equal to the number of centres so that we may get unique solution [15]. To this end, instead of getting an $N \times M$ matrix, we have an $N \times N$ matrix. To get the respective evaluation matrices for IQ and GIMQ, for the values of $\beta = -1$ and -2, the entries of equation (2.6) are evaluated.

Differentiation Matrix and Approximation of Derivatives using RBF Methods

Discretizing the partial derivatives of a PDE using an RBF method requires the use of a differentiation matrix, this is achieved by finding the partial derivatives of the interpolant (2.3) and expressing the differentiation matrix in terms of the interpolation and the evaluation matrices [10].

The first and second partial derivatives are given below

$$\frac{\partial}{\partial x_i} s(\boldsymbol{x}) = \sum_{j=1}^{N} \lambda_j \frac{\partial}{\partial x_i} \phi(\|\boldsymbol{x} - \boldsymbol{x}_j\|) \tag{2.8}$$

$$\frac{\partial^2}{\partial^2 x_i} s(\boldsymbol{x}) = \sum_{j=1}^{N} \lambda_j \frac{\partial^2}{\partial^2 x_i} \phi(\|\boldsymbol{x} - \boldsymbol{x}_j\|) \tag{2.9}$$

Higher order partial derivatives and mixed partial derivatives are handled in a similar manner.

Suppose equations (2.8) and (2.9) and higher order derivatives are evaluated at the centres $\{x_j\}$ and written in matrix form, we have

$$\frac{\partial^k}{\partial^k x_i} s(\boldsymbol{x}) = \frac{\partial^k}{\partial^k x_i} H \lambda_j, \qquad k = 1, 2, \dots, N \tag{2.10}$$

where $\frac{\partial^k}{\partial^k x_i} H$ are the derivatives of the evaluation matrix with entries defined by equation (2.7). We denote the derivatives of the first and second evaluation matrices as H_x and H_{xx} respectively.

Making λ_j the subject of the formula in equation (2.4) gives

$$\lambda_j = [A]^{-1} u \tag{2.11}$$

substituting equation (2.11) into (2.10) gives

$$\frac{\partial^k}{\partial^k x_i} s(\boldsymbol{x}) = \frac{\partial^k}{\partial^k x_i} H [A]^{-1} u \tag{2.12}$$

but

International Journal of Engineering Applied Sciences and Technology, 2019
Vol. 4, Issue 2, ISSN No. 2455-2143, Pages 7-15
Published Online June 2019 in IJEAST (http://www.ijeast.com)

$$\frac{\partial^k}{\partial^k x_i} u(\boldsymbol{x}) \approx \frac{\partial^k}{\partial^k x_i} s(x) = Du(x)$$

Clearly,

$$D = \frac{\partial^k}{\partial^k x_i} H[A]^{-1} \tag{2.13}$$

Equation (2.13) represents the differentiation matrix of RBF methods and it is defined if the interpolation matrix is invertible. The derivatives of the function $u(x)$ at the centres $\{x_j\}_{j=1}^N$ can be approximated using a single differentiation matrix D for linear problems, however, for nonlinear problems, D is applied to each partial derivative separately.

According to Sarra and Kansa [10], the chain rule for the first and second derivatives for any sufficient differentiable RBF $\phi(r)$, is defined by

$$\frac{\partial \phi}{\partial x_i} = \frac{d\phi}{dr}\frac{\partial r}{\partial x_i} \tag{2.14}$$

$$\frac{\partial^2 \phi}{\partial^2 x_i} = \frac{d\phi}{dr}\frac{\partial^2 r}{\partial^2 x_i} + \frac{d^2\phi}{d^2 x_i}\left(\frac{\partial r}{\partial x_i}\right)^2 \tag{2.15}$$

where

$$\frac{\partial r}{\partial x_i} = \frac{x_i}{r} \quad \text{and} \quad \frac{\partial^2 r}{\partial^2 x_i} = \frac{1-\left(\frac{\partial r}{\partial x_i}\right)^2}{r}$$

Well-Posedness of the Interpolation Problem

A problem is said to be well-posed if it exists and it is unique. The existence and uniqueness of the differentiation matrix, equation (2.13) used for the space discretization of PDEs is guaranteed if the interpolation matrix [A] is invertible. An interpolation matrix is invertible if the basis functions which are the entries of the matrix is completely monotone and radial Fasshauer [11], since the basis functions are generated from a basic function, we shall use the theorem below to establish that the basic functions of IQ and GIMQ are invertible and consequently their differentiation matrices.

Theorem 2.1: Completely Monotone Functions Fasshauer **[11]**

A function $\Phi: [0,\infty) \longrightarrow \mathbb{R}$ that is $C[0,\infty) \cap C^\infty[0,\infty)$ and satisfies

$$(-1)^\ell \phi^\ell(r) \geq 0, \quad \ell = 0,1,2,\ldots \tag{2.16}$$

is completely monotone on $[0,\infty)$.

Theorem 2.2: Multiply Monotone Functions Fasshauer **[11]**

A function $\Phi: (0,\infty) \longrightarrow \mathbb{R}$ which is $C^{d-2}(0,\infty), d \geq 0$ and for which $(-1)^\ell \phi^\ell(r) \geq 0$, non-increasing and convex for $\ell = 0,1,2,\ldots,d-2$ is called d times monotone on $(0,\infty)$. In case $d = 1$, the only requirement is that $\Phi \in (0,\infty)$ be non-negative and non-increasing.

Invertibility of the interpolation or System Matrices of IMQ, IQ and GIMQ

Using the basic functions provided in Table 1 and applying Theorems 2.1 and 2.2, we obtain the following results that show that the interpolation matrices of IQ and GIMQ are invertible and consequently exist and are unique.

IQ: $$(-1)^\ell \phi^\ell(r) = \ell!\,\varepsilon^{2\ell}(1+\varepsilon^2 r)^{-(\ell+1)} \geq 0, \tag{2.17}$$

GIMQ: $$(-1)^\ell \phi^\ell(r) = (\ell+1)!\,\varepsilon^{2\ell}(1+\varepsilon^2 r)^{-(\ell+2)} \geq 0. \tag{2.18}$$

$$\ell = 0,1,2,\ldots$$

Algorithm for Discretizing the Space Variable(s) using RBF Methods

To apply meshless method of lines technique using radial basis function, we consider a PDE of the form

$$\frac{\partial u}{\partial t} + L(u) = 0, \quad \boldsymbol{x} \in \Omega, \quad t \geq 0 \tag{2.19}$$

where

$u = u(\boldsymbol{x}, t)$,

$\Omega = \Omega_1 \cup \Omega_2$,

$\Omega_1 =$ interior data points

$\Omega_2 =$ boundary data points

$L =$ spatial derivatives operator.

One advantage of the Kansa's method is that both the interior and boundary data points are discretized at once and in a similar manner, however, the interior points are arranged before the boundary points.

Let the centres in the problem domain be given as $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N \in \Omega \subset \mathbb{R}^d$, $(d = 1,2,3)$. Since we are dealing with two-dimensional problems, we restrict $d = 2$.

The approximate solution $u(\boldsymbol{x}, t)$ for a time dependent PDE can be expressed as

$$Lu(\boldsymbol{x}) = \sum_{j=1}^{N} \lambda_j(t)\, \left(L\phi\left(\|\boldsymbol{x} - \boldsymbol{x}_j\|\right)\right). \tag{2.21}$$

In matrix form, equations (2.20) and (2.21) can be written as

$$[A]c = u,$$
$$u = [u_1(t), u_2(t), \ldots, u_N(t)]^T \tag{2.22}$$

and

$$[B]c = L(u) \tag{2.23}$$

where A has entries of the form

$$[A] = \phi\left(\|\boldsymbol{x} - \boldsymbol{x}_j\|\right), \qquad i,j = 1,2,\ldots,N$$

and B is the matrix with entries of the form

$$[B] = L\phi\left(\|\boldsymbol{x} - \boldsymbol{x}_j\|\right)_{\boldsymbol{x}=x_i}, \qquad i,j = 1,2,\ldots,N$$

From equation (2.21), (2.22) and (2.23), we have

$$L(u) = (BA^{-1})u, \tag{2.24}$$
$$L(u) = Du, \tag{2.25}$$

where

$$D = [B][A]^{-1}$$

After discretizing a PDE in space with radial basis functions, equation (2.19) is transformed into a semi discretized system given by

$$\frac{du}{dt} = Du \qquad (2.26)$$

The system of ODEs (2.26) are integrated using a suitable ODE solver. In this work after discretizing the space derivatives using the differentiation matrices of IQ and GIMQ, the resulting systems of ODEs are integrated using the fourth order explicit Runge-Kutta method.

Algorithm for Fourth Order Runge-Kutta Method used as a Time-Stepping Method

The explicit fourth-order Runge-Kutta (RK4) algorithm for integrating the system of ODEs is described below. It is assumed that $u(t_n, \boldsymbol{x}_n)$, $\boldsymbol{x}_n \in \mathbb{R}^2$,

$$k_1 = \Delta t F(t_n, \boldsymbol{x}_n),$$

$$k_2 = \Delta t F\left(t_n + \frac{\Delta t}{2}, \boldsymbol{x}_n + \frac{k_1}{2}\right),$$

$$k_3 = \Delta t F\left(t_n + \frac{\Delta t}{2}, \boldsymbol{x}_n + \frac{k_2}{2}\right),$$

$$k_4 = \Delta t F(t_n + \Delta t, \boldsymbol{x}_n + k_3).$$

Therefore,

$$u_{n+1} = u_n + \tfrac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4).$$

III. EXPERIMENT AND RESULTS

In this Section, we provide the numerical results for a two-dimensional heat equation using the IQ RBF-MOLs and GIMQ RBF-MOLs formulated and described in Section 2. These methods are implemented in MATLAB, while the results are displayed in Tables and graphs for analysis, discussion and conclusion. The CPU time for this Example is also provided for comparison with the MQ RBF-MOLs. All the programmes are written in Windows 8 operating system using MATLAB 2007b. The test problem and the parameter values are drawn from the work of Sarra and Kansa [10].

Example: A Two-Dimensional Heat Equation

Consider the two-dimensional heat equation

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}, \qquad (3.1)$$

the Dirichlet boundary conditions are specified using the exact solution

$$u(x, y, t) = \exp(-5t\pi^2)\sin(\pi x)\sin(2\pi y),$$

while the initial condition is given as

$$u(x, y, 0) = \sin(\pi x)\sin(2\pi y).$$

Equation (3.1) is solved on three domains containing different pattern of spaced data points namely, equally spaced data points, scattered data points and scattered points on a complex domain as shown in Figure 1. The space derivatives are discretized using the inverse quadratic (IQ) and the generalized inverse multiquadric (GIMQ) RBFs, while the resulting system of ODEs are integrated using the fourth order Runge-Kutta method. This Example is implemented in MATLAB using a time step size of $\Delta t = 5.0 \times 10^{-4}$ and $\Delta t = 5.0 \times 10^{-5}$ respectively to advance the solution up to the final time $FT = 0.1$, the result is recorded in Table 2, while the graphical display of the initial profile and the different RBF-MOLs are displayed in Figures $2-5$.

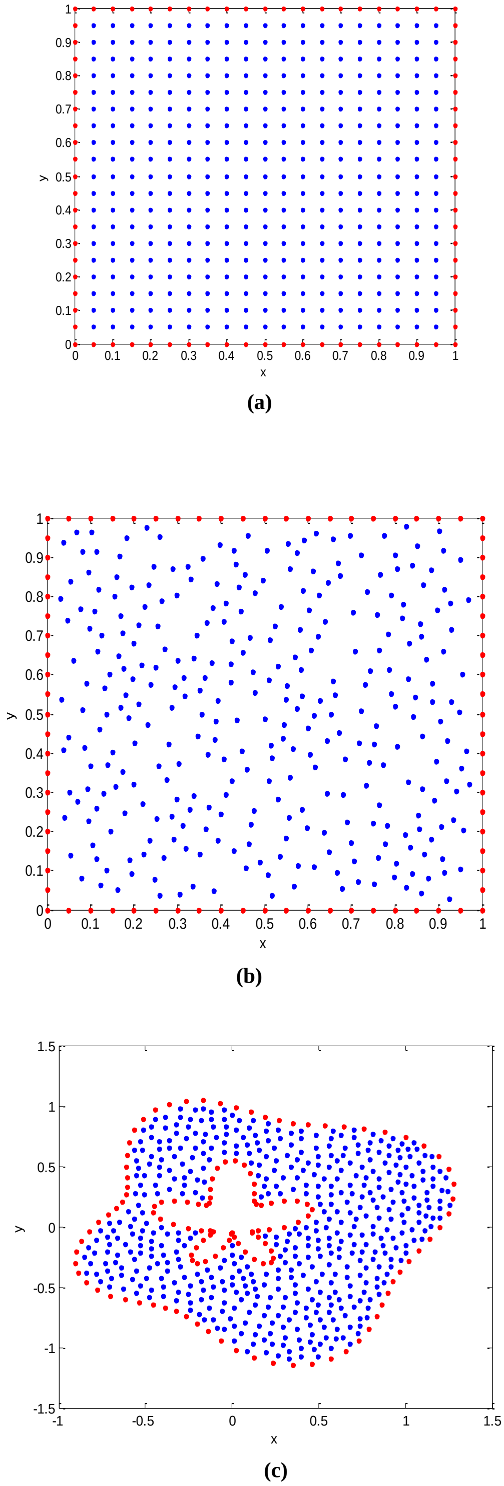
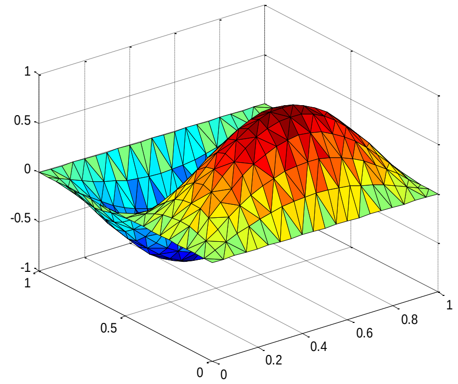
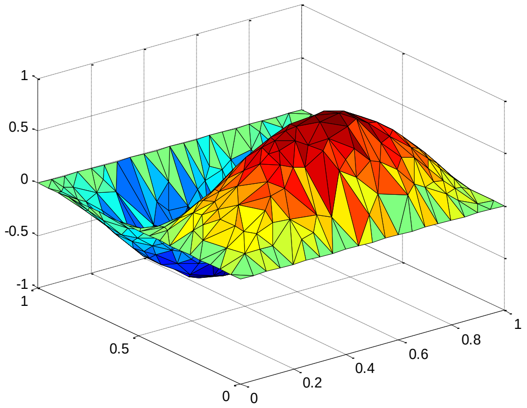


Fig. 1: Computational Domain and Data Points Locations on (a) Equally Spaced Data Points (b) Scattered Data Points (c) Complex Domain

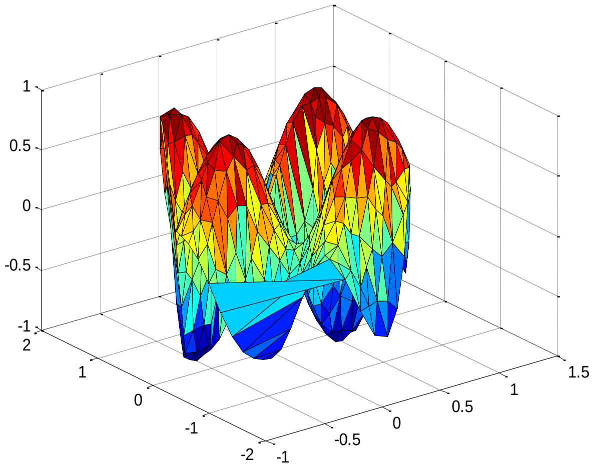
International Journal of Engineering Applied Sciences and Technology, 2019
Vol. 4, Issue 2, ISSN No. 2455-2143, Pages 7-15
Published Online June 2019 in IJEAST (http://www.ijeast.com)



(a)

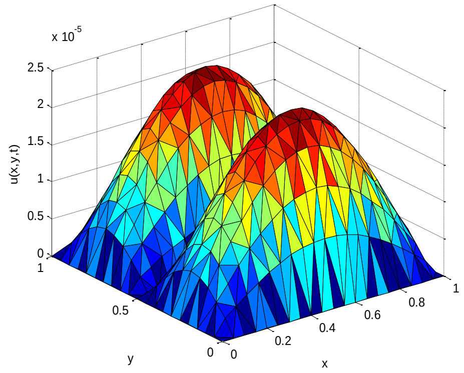


(b)

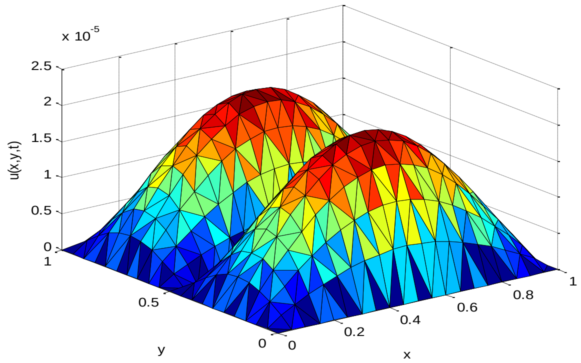


(c)

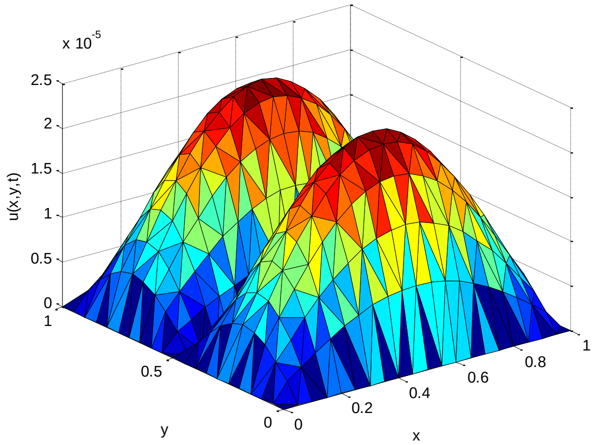
Fig. 2: Initial Profile for the two-dimensional heat equation on (a) Equally Spaced Data Points (b) Scattered Data Points (c) Complex Domain



(a)



(b)



(c)

Fig. 3: Solution of the two-dimensional heat equation using (a) MQ, (b) IQ, (c) GIMQ RBF-MOLs at the Final Time, $FT = 0.1$ on Equally Spaced Data Points

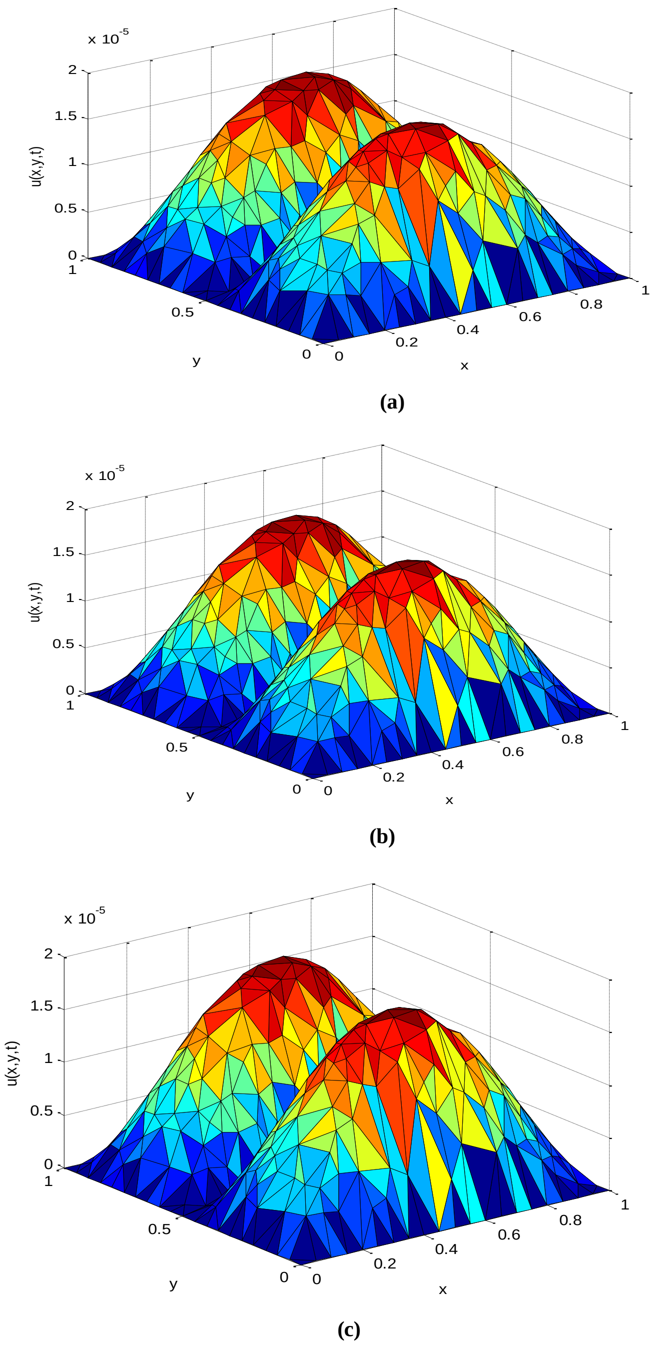


Fig. 4: Solution of the two-dimensional heat equation using (a) MQ, (b) IQ, (c) GIMQ RBF-MOLs at the Final Time, $FT = 0.1$ on Scattered Spaced Data Points

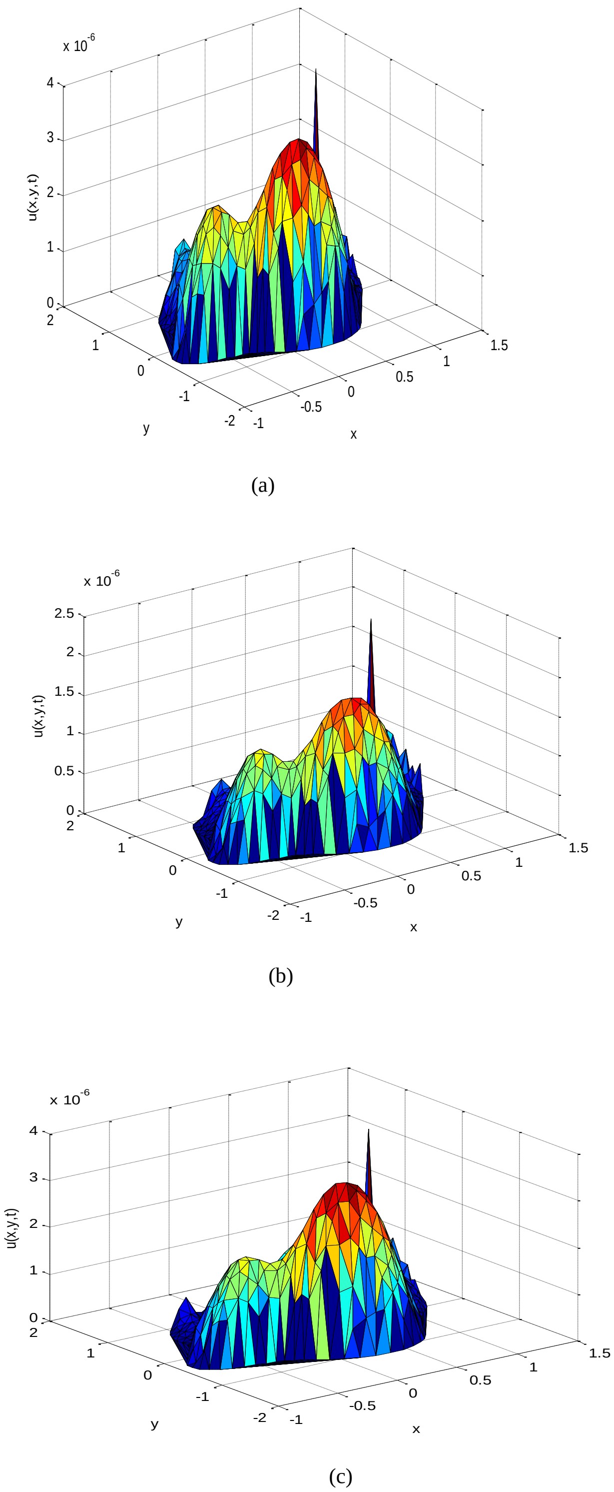


Fig. 5: Solution of the two-dimensional heat equation using (a) MQ, (b) IQ, (c) GIMQ RBF-MOLs at the Final Time, $FT = 0.1$ on a Complex Domain

International Journal of Engineering Applied Sciences and Technology, 2019
Vol. 4, Issue 2, ISSN No. 2455-2143, Pages 7-15
Published Online June 2019 in IJEAST (http://www.ijeast.com)

Table 2: Summary of MQ, IQ and GIMQ RBF-MOLs for the 2D Heat Equation

S/N	RBF-MOLs	N	Δt	FT	ε	MPE	$\kappa(B)$
(a)	Solution on the Domain Containing Equally Spaced Data Points						
1	MQ	441	5.0×10^{-5}	0.1	3.0	1.9418×10^{-5}	2.6181×10^{15}
2	IQ				2.5	2.3923×10^{-5}	6.5877×10^{18}
3	GIMQ				2.0	2.0685×10^{-5}	7.0100×10^{16}
(b)	Solution on the Domain Containing Scattered Data Points						
1	MQ	399	5.0×10^{-5}	0.1	3.0	1.8416×10^{-5}	1.1269×10^{14}
2	IQ				2.0	1.7790×10^{-5}	6.5616×10^{15}
3	GIMQ				2.0	1.8526×10^{-5}	2.5277×10^{14}
(c)	Solution on the Complex Domain Containing Scattered Data Points						
1	MQ	635	5.0×10^{-4}	0.1	2.0	3.7527×10^{-6}	2.2420×10^{15}
2	IQ				1.5	2.1306×10^{-6}	5.0443×10^{15}
3	GIMQ				1.5	3.5593×10^{-6}	1.7610×10^{14}

Table 3: Average CPU Time in Seconds for the 2D Heat Equation

Domain/Data Points	MQ	IQ	GIMQ
Regular/Equally Spaced Points	26.127121	24.729393	24.034108
Halton/Scattered Spaced Points	24.777299	22.431606	22.273862
Complex/Scattered Spaced Points	8.951064	5.969877	7.607321

IV. DISCUSSION

In this Section, the results of our findings are discussed and compared with the MQ RBF-MOLs of Sarra and Kansa [10], a conclusion is also drawn based on our findings.

To approximate the two-dimensional heat equation using RBF-MOLs**,** both the Dirichlet boundary conditions and the initial condition of the two-dimensional heat equation were obtained from the analytic solution. Sarra and Kansa [10] solved this particular problem using MQ RBF-MOLs on a complex shaped domain comprising 310 boundary data points and 505 interior data points. In this work, the same problem is used as a test problem on the IQ and GIMQ RBF-MOLs formulated. Besides applying this problem on the complex domain, a domain consisting of equally spaced data points made up of 80 boundary points and 361 interior points was used. Also, a domain containing 80 equally spaced boundary points and 319 scattered data points produced using the Halton sequence was used. The three domains containing the three data points are provided in Figure 1.

The space derivatives were discretised using the IQ and GIMQ RBF methods. We observed that discretizing the two-dimension space derivatives was similar to the discretization of the space derivatives in one dimension. Just a little more computational effort was required unlike using the polynomial based methods such as the FDM, FVM or FVM. The fourth order Runge-Kutta method was used in a similar way as in the case of one dimensional time-dependent PDEs to integrate the resulting systems of ODEs using a time step size of $\Delta t = 5.0 \times 10^{-5}$ on the domains containing equally spaced and scattered spaced data points, while the time step, $\Delta t = 5.0 \times 10^{-4}$ was used on the complex domain to advance the solution up to the final time FT=0.1. The results for the two-dimensional heat equation is displayed in Table 2 and Figures 2-5.

We observed that the results in Figure 5 produced on the nearly optimal scattered data points was better than the results produced on the other two data points. Also, the results produced in Figure 4 on scattered that points was slightly better than those produced on the equally spaced domain displayed in Figure 3.Comparing the results of IQ and GIMQ RBF-MOLs with the MQ RBF-MOLs of Sarra and Kansa [10] using the maximum point-wise error in Table 2 shows that the MQ RBF-MOLs produced the best results on the domain containing equally spaced data points, while the results for the IQ RBF-MOLs yielded the smallest point-wise error on the domain containing scattered data points and on the complex domain containing scattered data points.

An estimate for the CPU time for the two-dimensional heat equation was obtained by finding the average of three different readings and is provided in Table 3. The CPU time for the two-dimensional heat equation performed on three different domains/different patterns of data spacing revealed that the GIMQ RBF-MOLs had the smallest CPU time recorded on both the equally spaced data points and scattered data points, while the IQ RBF-MOLs recorded the least CPU time on the complex domain. It is also important to note from Table 3 that for all the three RBF-MOLs, the CPU time for the two-dimensional heat equation performed on the complex domain with scattered data points have the least timing followed by the domain containing Halton points with scattered points and finally on the domain containing equally spaced data points.

Evaluating the performance of the RBF-MOLs using the point-wise error and CPU time shows that the RBF methods performed better on scattered data points compared to uniform data points. Also, since RBFs are evaluated on data points, these points can be scattered and arranged to form different patterns to give an optimal result on a complex domain which is difficult with the other polynomial based methods.

V. CONCLUSION

Two globally supported positive definite and infinitely continuous radial basis functions (c^{∞} −RBFs), namely, the inverse quadratic (IQ) and the generalized inverse multiquadric (GIMQ) RBFs were combined with the fourth order Runge-Kutta method to construct radial basis function method of lines (RBF-MOLs). The IQ RBF-MOLs and GIMQ RBF-MOLs constructed were used to approximate the heat equation in two space dimensions and compared the numerical results with the MQ RBF-MOLs. The performance of the proposed RBF-MOLs measured in terms of the point-wise error indicate that the methods performed comparably with the MQ RBF-MOLs of Sarra and Kansa [10]. There were some instances where the proposed RBF-MOLs had smaller point-wise errors than the MQ RBF-MOLs. Also, the performance of the CPU time for the two-dimensional heat equation showed that the constructed RBF-MOLs performed comparably with the MQ RBF-MOLs. The performance of IQ and GIMQ RBF-MOLs used for solving the heat equation agrees with the work of Luga, *et al* .[17], they proposed the IMQ, IQ and GIMQ RBF-MOLs for approximating some time-dependent PDEs in one space dimension and observed that these RBF-MOLs compared favourably with the MQ RBF-MOLs of Sarra and Kansa [10].

Based on the results obtained from the proposed IQ and GIMQ RBF-MOLs, we recommend that these methods be applied for the solution of some time-dependent PDEs in two or higher dimensional space in different fields of sciences and engineering. Also, the IQ and GIMQ RBF methods could be used independently or combined with other numerically methods to constructed accurate hybrid numerical methods, since the proposed RBF-MOLs compared favourably with the MQ RBF-MOLs.

VI. REFERENCE

[1] Buhmann, M. D. (2003). *Radial Basis Functions: Theory and Implementations*, Cambridge University Press, Cambridge.

[2] Kansa, E. J (1990a). Multiquadrics-A Scattered Data Approximation Scheme with Application to Computational Fluid Dynamics-I: Surface Approximations and Surface Estimates *Computer and Mathematics with Applications,* 19(8-9): 127-145.

[3] Kansa, E. J. (1990b). Multiquadrics-A Scattered Data Approximation Scheme with Applications to Computational Fluid Dynamics II: Solutions to Parabolic, Hyperbolic and Elliptic Partial Differential Equations. *Computers and Mathematics with Applications*, 19(8-9): 147-161.

[4] Chen, W. Fu, Z.-J and C. Chen, C. S. (2014). *Recent Advances in Radial Basis Function Collocation Method*. New York, Springer Heidelberg.

[5] Kansa, E. J and Holoborodko, P. (2017). On the Ill-Nature of C^{∞} RBF Strong Collocation. *Engineering Analysis with Boundary Elements,* 78: 26-30.

[6] Iske, A. (2002). *Scattered Data Modelling using Radial Basis Functions. In: Tutorials on Multiresolution in Geometric Modelling, Eds. by Iske, A., Quack, E. and Floater, M. S.* Springer-Verlag, Heidelberg, pp. 205-242.

[7] Chen, S. Cowan, C. F. N. and Grant, P. M. (1991). Orthogonal Least-Square Learning Algorithm for Radial Basis Function Networks. *IEEE Trans. Neural Netw.*, 2(2): 302-309.

[8] Cortes, C. and Vapnik, V. (1995). Support-Vector Network. *Mach. Learning* 20(3): 273-297.

[9] Burges, C. J. C. (1998). A Tutorial on Support Vector Machines for Pattern Recognition. Data Min. Knowl. Disc. 2(2): 121-167.

[10] Sarra, S. A. and Kansa, E. J. (2009). *Multiquadric Radial Basis Function Approximation Methods for the Numerical Solution of Partial Differential Equations,* California, Technical Science Press.

[11] Fasshauer, G. E. (2007). *Meshfree Approximation Methods with MATLAB*, Singapore, World Scientific Publishing Co. Ltd.

[12] Chen, C. S. Hon, Y. C. and Schaback, R. S. (2007). *Radial Basis Functions with Scientific Computations*. Department of Mathematics, University of Southern Mississippi, Hattiesburg.

[13] Yao, G. Sarler, B. and Chen, C. S. (2011). A Comparison of Three Explicit Local Meshless Methods using Radial Basis Functions. *Engineering Analysis with Boundary Elements.* 35: 600-609.

[14] Franke, R. (1982). Scattered Data Interpolation: Test of Some Methods. *Mathematics of Computation*, 38 (157): 181-200.

[15] Fasshauer, G. and McCourt, M. (2015). *Kernel-based Approximation Methods using MATLAB.* Singapore, World Scientific Publishing Co. Ltd, Singapore.

[16] Bibi, N. (2011). Meshless Method of Lines for Numerical Solutions of Nonlinear Time-Dependent Partial Differential Equations. Ph. D. Thesis, Ghulam Ishaq Khan Institute of Engineering Science and Technology, Topi, Swabi, Pakistan.

[17] Luga, T. Aboiyar, T. and Adee, S. O. (2017). Radial Basis Functions for Approximating Time-Dependent Partial Differential Equations in One Space Dimension. *Nigerian Journal of Pure and Applied Sciences*, 9(1):182-200.