International Journal of Engineering Applied Sciences and Technology, 2020
Vol. 5, Issue 1, ISSN No. 2455-2143, Pages 1-12
Published Online May 2020 in IJEAST (http://www.ijeast.com)

ON SOME POSITIVE NON-HALF-INTEGER EXPONENT RADIAL BASIS FUNCTION METHODS FOR APPROXIMATING THE SOLUTIONS OF SOME SECOND ORDER PARTIAL DIFFERENTIAL EQUATIONS

Tersoo Luga
Department of Mathematics/Statistics/Computer Science,
University of Agriculture, Makurdi, Nigeria

Abstract—**Some radial basis functions with positive non-half-integer (PNH-RBFs) exponents have been used to develop numerical methods that are claimed to produce better numerical approximations when compared to methods constructed with radial basis functions having negative-integer/half-integer exponents referred to as standard RBFs. In this paper, we develop a numerical method for approximating the solution of steady state partial differential equations (PDEs) and a radial basis function method of lines (RBF-MOLs) for solving time-dependent PDEs in two space dimensions using two PNH-RBFs. Two Poisson equations and a heat equation in two space dimensions were used as test problems to perform numerical experiments and compared with results from methods developed with the standard RBFs. From our results, all the radial basis function methods produced nearly the same accuracy regardless of the value of the exponent.**

Keywords—**Radial Basis Functions, Method of Lines and Generalized Multiquadrics**

I. INTRODUCTION

Partial differential equations (PDEs) are used for modelling real life phenomenon in different mathematical and engineering communities. Not all PDEs have analytical solutions, so numerical methods are required to provide approximate solutions in such cases. Sometimes even if there exists an analytical solution, numerical methods are developed for the purpose of comparison and to prepare for situations where the analytical solution may fail. In 1950's, PDEs were approximated mainly with finite difference method (FDM), while in 1960's, the finite element method (FEM) was mostly used since it can be applied on irregular shaped domains (Fasshauer [1]). However, these methods have algebraic convergence (Sarra and Kansa [2]) which limit their accuracy. Spectral methods were developed in 1970's to provide exponential convergence [1] to PDEs both on structured and unstructured domains.

Mesh-free methods stated appearing in literature in 1980's Fasshauer [1] and are becoming viable tools for approximating the solutions of many mathematical and engineering problems in recent times because of two major reasons: (1) the ease to generate mesh over two- and three-dimensional complicated domains which take longer time with the traditional methods such as FDM and FEM, (2) the convergence rate of the traditional methods which are typically second order and require more discretization and operations than the mesh-free methods Kansa [2]. Radial basis functions (RBFs) are mesh-free methods whose value depends only on distance from the origin or points called centres. RBFs were derived for the purpose of multivariate scattered data interpolation Hardy [3], however, the application of RBFs for solving PDEs appeared in 1990 when Kansa [4] and [5] developed a multiquadratic (MQ) collocation method and applied it to approximate the solutions of some PDEs. The breakthrough of Kansa lead to the applications of RBFs for solving PDEs in different computing communities.

One of the most frequently used RBFs are the multiquadrics (MQ) (Chenoweth [6]; Kansa [2] and Kansa and Holobrodko [7]), this may be traced to the experiments conducted by Frank [8] and other researchers such as Madych and Nelson [9] and Madych [10] that proved theoretically that multiquadrics converges exponentially. The multiquadrics are member of the generalized MQ (GMQ) family

$$\phi(r) = (1 + (\varepsilon r)^2)^{\beta} \tag{1.1}$$

where the exponent β may be any real number except non-negative integers Sarra and Kansa [11]. Some RBFs with negative integer exponent are the inverse quadratic (IQ), $\beta = -1$ and the generalized inverse MQ (GIMQ), $\beta = -2$. Commonly used half-integer exponent GMQ RBFs are the MQ, $\beta = \frac{1}{2}$ and the inverse (IMQ), $\beta = -\frac{1}{2}$. The IQ, GIMQ, MQ and IMQ have been successfully applied by Bibi [12]; Luga *et al.* [13], [14] and Sarra and Kansa [11] among others to approximate the solutions of some PDEs. Higher positive half-integer exponents of equation (1.1), $\beta = \frac{(2m-1)}{2}$, $m = 0, 1, 2, \ldots$ are becoming interesting area of research especially with advent of advanpix MATLAB tool box Kansa and Holobrodko [7]. The paper by Kansa *et al.* [15] and some experiments in the book of Sarra and Kansa [11] confirm that higher positive half-integers exponents GMQ RBFs yield good approximation. Wertz *et al*. [16] showed that the values of GMQ for $\beta \geq \frac{1}{2}$ under some conditions may produce better results than other values of β, regardless of whether the data points are randomly or uniformly distributed.

Contrary to the conclusion of Wertz *et al*. [16], some positive non-half integer exponents of equation (1.1) have been shown by some researchers to produce better results. For instance, Wang and Liu [17] proposed a radial point interpolation method (radial PIM) in which they studied the effect of shape parameters on the numerical accuracy of different exponents β of GMQ RBFs and the Gaussian RBF, from the various experiments performed, they observed that the optimal value of the shape parameters was obtained at $\beta = 1.03$. Similarly, Xiao and McCarthy [18] developed a meshless method for stress analysis of two-dimensional solids based on a local weighted residual method with the Heaviside step function over a local subdomain. Experiments were performed with equation (1.1) using $\beta = -\frac{1}{2}, \frac{1}{2}, 1.03 \text{ and } 1.99$, the difference between the relative errors and the analytical solutions confirmed that the value of $\beta = 1.99$ produced the optimal results. Furthermore, Xiao *et al*. [19] also showed that to get efficient results with RBFs, the exponent β in equation (1.1) has to be optimized, meaning that β does not have to be restricted to half-integer values. They obtained good results with the value of $\beta = 1.1$ and confirmed that non-half- integer values of β accelerated convergence.

Chenoweth [6] observed that the claims by some researchers that non-half-integer exponents GMQ RBFs produced optimal results are not proved and have no theoretical backings. She conducted series of numerical experiments on interpolation in one and two dimensions with different values of β in equation (1.1) to determine the existence of an optimal value, but concluded that the optimal value of β is problem dependent. In this paper, we used the values of $\beta = 1.03 \text{ and } 1.99$ in equation (1.1) to develop RBF collocation method and apply the methods to approximate the solutions of some steady state and a time-dependent partial differential equation in two space dimensions. The solution of the time-dependent PDEs are approximated with the method of lines (MOLs) constructed by combining the positive non-half-integer exponent GMQ RBFs with the fourth order Runge-Kutta method. The aim is to verify the claim that some non-half-integer GMQ RBFs produce better numerical results when compared to other RBFs using second order PDEs. The MQ RBF is used as a standard for our comparison.

The rest of the paper is structured as follows: Section II deals with the methods developed for performing experiments, the results are presented in Section III, while the discussion and conclusion are done in Sections IV and V respectively and the references are provided in Section VI.

II. METHODS

The Formulation of GMQ-RBF having non-standard exponents $\beta = 1.03$ and $\beta = 1.99$ for solving steady state and time-dependent PDEs is presented in this Section. For the time-dependent PDEs, non-standard exponent GMQ-RBFs are used for space discretization, while the 4th order Runge-Kutta method is used as a time-stepping method.

Radial Basis Function Interpolation Method

To obtain RBF interpolation, we first assume that if $u: \mathbb{R}^d \longrightarrow \mathbb{R}$, is an unknown function, it can be approximated with an RBF interpolant of the form $s: \mathbb{R}^d \longrightarrow \mathbb{R}$ defined by

$$u(x) = s(x) \tag{2.1}$$

The interpolant on the right hand side of equation (2.1) is given by

$$s(x) = \sum_{j=1}^{N} \alpha_j \phi(\|x - x_j\|) + p(x) \tag{2.2}$$

where $\|\cdot\|$ denotes the Euclidean norm, $p \in \mathcal{P}_m^d$ is a polynomial of degree $m-1$ and α_j $j = 1, 2, \ldots, N$ is an unknown vector to be found. If an RBF is positive definite, then equation (2.2) is used without appending the polynomial term, yet the interpolant produces an invertible interpolation matrix. On the other hand, the interpolant of a conditionally positive RBFs need to be appended with the polynomial term in order to get an invertible interpolation matrix (Fasshauer [1]). Equation (1.1) is called positive definite if $\beta < 0$, and conditionally positive definite if $\beta > 0$. Although the RBFs we are using for the space discretization of the PDEs are conditionally positive definite, we shall apply them without appending the polynomial term as explained in (Sarra and Kansa [11]). Thus equation (2.2) reduces to

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

Substituting equation (2.3) in equation (2.1) and expanding for each $x = x_i, i = 1,2,3, \ldots, N,$ $j = 1,2,3, \ldots, N$, gives the $N \times N$ interpolation matrix in $d \geq 1$ dimensions

$$\begin{bmatrix} \phi(\|x_1 - x_1\|) & \phi(\|x_1 - x_2\|) & \cdots & \phi(\|x_1 - x_N\|) \\ \phi(\|x_2 - x_1\|) & \phi(\|x_2 - x_2\|) & \cdots & \phi(\|x_2 - x_N\|) \\ \vdots & \vdots & \ddots & \vdots \\ \phi(\|x_N - x_1\|) & \phi(\|x_N - x_2\|) & \cdots & \phi(\|x_N - x_N\|) \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_N \end{bmatrix} = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_N \end{bmatrix} \tag{2.4}$$

In vector-matrix notation, equation (2.4) is written as

$$B\alpha = u. \tag{2.5}$$

The unknown vector α is obtained from equation (2.5) as shown in equation (2.6)

$$\alpha = B^{-1}u \tag{2.6}$$

The evaluation matrix is obtained by evaluating equation (2.3) for each data point $x = x_i, i = 1,2,3,\dots,M,\ j = 1,2,3,\dots,N$ centres, however, to ensure a symmetric evaluation matrix, both N data points and centres are used. The evaluation matrix is expressed as

$$\sum_{j=1}^{N} \alpha_j \phi(\|x - x_j\|) = H\alpha_j \tag{2.7}$$

where H has the entries $h_{ij} = \phi(\|x - x_j\|)$.

Differentiating (2.7), we get the differentiation matrix,

$$\sum_{j=1}^{N} \alpha_j \frac{\partial}{\partial x_i}\phi(\|x_i - x_j\|) = \frac{\partial}{\partial x_i}H\alpha_j, \tag{2.8}$$

equation (2.8) can be differentiated the number of required times to get the order of the derivative of interest.

Thus, relating equation (2.8) and equation (2.1) shows that

$$\frac{\partial}{\partial x_i}u(x_i) \approx \frac{\partial}{\partial x_i}s(x_i) = \frac{\partial}{\partial x_i}H\alpha_j. \tag{2.9}$$

Substituting equation (2.6) in equation (2.9) yields

$$\frac{\partial}{\partial x_i}u = \frac{\partial}{\partial x_i}HB^{-1}u,$$

Let

$$D = \frac{\partial}{\partial x_i}HB^{-1} \tag{2.10}$$

Equation (2.10) is called the differentiation matrix which is used for approximating the derivatives of a given PDE.

Substituting the basic functions

$$\phi(r) = \left(1 + (\varepsilon\|x_i - x_j\|)^2\right)^{1.03}, i,j = 1,2,3,\dots,N \tag{2.11}$$

and

$$\phi(r) = \left(1 + (\varepsilon\|x_i - x_j\|)^2\right)^{1.99}, \quad i,j = 1,2,3,\dots,N \tag{2.12}$$

we get the following basis functions which can be substituted to get the interpolation matrix (2.4) and evaluation matrix (2.7) which are used for the formulation of the differentiation matrix in two dimensions.

$$\begin{bmatrix} (1+(\varepsilon\|x_1-x_1\|)^2)^{1.03} & (1+(\varepsilon\|x_1-x_2\|)^2)^{1.03} & \cdots & (1+(\varepsilon\|x_1-x_N\|)^2)^{1.03} \\ (1+(\varepsilon\|x_2-x_1\|)^2)^{1.03} & (1+(\varepsilon\|x_2-x_2\|)^2)^{1.03} & \cdots & (1+(\varepsilon\|x_2-x_N\|)^2)^{1.03} \\ \vdots & \vdots & \ddots & \vdots \\ (1+(\varepsilon\|x_N-x_1\|)^2)^{1.03} & (1+(\varepsilon\|x_N-x_2\|)^2)^{1.03} & \cdots & (1+(\varepsilon\|x_N-x_N\|)^2)^{1.03} \end{bmatrix} \tag{2.13}$$

and

$$\begin{bmatrix} (1+(\varepsilon\|x_1-x_1\|)^2)^{1.99} & (1+(\varepsilon\|x_1-x_2\|)^2)^{1.99} & \cdots & (1+(\varepsilon\|x_1-x_N\|)^2)^{1.99} \\ (1+(\varepsilon\|x_2-x_1\|)^2)^{1.99} & (1+(\varepsilon\|x_2-x_2\|)^2)^{1.99} & \cdots & (1+(\varepsilon\|x_2-x_N\|)^2)^{1.99} \\ \vdots & \vdots & \ddots & \vdots \\ (1+(\varepsilon\|x_N-x_1\|)^2)^{1.99} & (1+(\varepsilon\|x_N-x_2\|)^2)^{1.99} & \cdots & (1+(\varepsilon\|x_N-x_N\|)^2)^{1.99} \end{bmatrix} \tag{2.14}$$

Existence and Uniqueness of Interpolation Matrix

For an RBF method applied for discretizing the space derivatives of a PDE to exist and be unique, the interpolation matrix of the RBF must be invertible. There are many methods for characterizing the existence and uniqueness of an interpolation matrix (Fasshauer [1]), however, we shall verify that the RBF of interest are completely monotone.

Completely Monotone Functions (Sarra and Kansa [11])

A function $\phi(r)$ is completely monotone on $[0,\infty)$ if

(i) $\phi \in C[0,\infty)$,

(ii) $\phi \in C^{\infty}(0,\infty)$,

(iiii) $(-1)^{\ell}\phi^{(\ell)}(r) \geq 0$,

where $r > 0$ and $\ell = 0,1,2,\dots$.

The RBFs we have selected are conditionally positive definite, thus we shall verify that their basic functions are completely monotone using a proof in (Fasshauer [1]).

2.2.2 Theorem 2: (Fasshauer [1])

Suppose $\phi(r)$ is completely monotone, then

$$\phi(r) = (-1)^{\lceil\beta\rceil}(1+r)^{\beta}, \qquad 0 < \beta \notin \mathbb{N},$$

imply

$$\phi^{(\ell)}(r) = (-1)^{\lceil\beta\rceil}\beta(\beta-1)\dots(\beta-\ell+1)(1+r)^{\beta-\ell}$$

so that

$$(-1)^{\lceil\beta\rceil}\phi^{(\lceil\beta\rceil)}(r) = \beta(\beta-1)\dots(\beta-\lceil\beta\rceil+1)(1+r)^{\beta-\lceil\beta\rceil} \tag{2.15}$$

where $\lceil\beta\rceil$ means the least integer greater than β.

If $\beta = 1.03$ and $\beta = 1.99$, then both $\lceil 1.03\rceil$ and $\lceil 1.99\rceil$ are equal to 2.

Thus equation (2.15) in both cases reduce to

$$(-1)^2\phi^{(2)}(r) = (1.03)(0.03)\frac{1}{(1+r)^{0.97}} \geq 0 \tag{2.16}$$

and

$$(-1)^2\phi^{(2)}(r) = (1.99)(0.99)\frac{1}{(1+r)^{0.01}} \geq 0 \tag{2.17}$$

Equation (2.16) and (2.17) shows that Theorem 1 is verified.

Algorithm for Discretizing Steady State Partial Differential Equations

Let N denote distinct data points and centres that are divided into two subsets, one containing the interior data points and centres while the other is used to enforce boundary conditions denoted by

$$E = [X_I, X_B].$$

Let $\mathcal{L}$ and $\mathcal{B}$ denote the differential operators on the interior, X_I and boundary points, X_B, applying the operators, we get

$$\mathcal{L}u(x_i) = \sum_{j=1}^{N} \alpha_j \mathcal{L}\phi(\|x_i - x_j\|), \quad i = 1,2,\dots,N_I \tag{2.18}$$

and

$$\mathcal{B}u(x_i) = \sum_{j=1}^{N} \alpha_j \mathcal{B}\phi(\|x_i - x_j\|), \quad i = N_{I+1},\dots,N \tag{2.19}$$

The right hand side of equations (2.18) and (2.19) can be written in matrix notation as $\frac{\partial}{\partial x}H\alpha$, where the evaluation matrix H that discretizes the PDE consist of

$$\frac{\partial}{\partial x}H = [\mathcal{L}\phi + \mathcal{B}\phi]. \qquad (2.20)$$

The entries of (2.20) are defined by

$(\mathcal{L}\phi)_{ij} = \mathcal{L}\phi(\|x_i - x_j\|), \quad i = 1, 2, \dots, N_I, j = 1, 2, \dots, N$

$(\mathcal{B}\phi)_{ij} = \mathcal{B}\phi(\|x_i - x_j\|), \quad i = N_{I+1}, \dots, N, j = 1, 2, \dots, N.$

Multiplying B^{-1} to equation (2.20) yields the differentiation matrix used for discretizing steady state PDEs which only have space derivatives, i.e. equation (2.10). The implementation is done using a MATLAB programme.

2.4 Algorithm for Discretizing Time-Dependent Partial Differential Equations

For the time-dependent PDEs considered in this paper, once the space discretization is performed, the PDE can be written as

$$\frac{\partial u}{\partial t} + [\mathcal{L} + \mathcal{B}] = 0 \qquad (2.21)$$

where

$$\mathcal{L}u(x_i) = \sum_{j=1}^{N} \alpha_j(t)\mathcal{L}\phi(\|x_i - x_j\|), \quad i = 1, 2, \dots, N_I, j = 1, 2, \dots, N \qquad (2.22)$$

and

$$\mathcal{B}u(x_i) = \sum_{j=1}^{N} \alpha_j(t)\mathcal{B}\phi(\|x_i - x_j\|), \quad i = N_{I+1}, \dots, N, j = 1, 2, \dots, N. \qquad (2.23)$$

Equation (2.21), (2.22) and (2.23) can be written as a single equation as shown below

$$\frac{du}{dt} = \frac{\partial}{\partial x}H\alpha \qquad (2.24)$$

For time-dependent PDEs, equation (2.5) takes the form

$$B\alpha = u, \quad u = \{u_1(t), u_2(t), \dots, u_N(t)\} \qquad (2.25)$$

by making α the subject of the formula gives equation (2.6). Substituting equation (2.6) in (2.24) yields

$$\frac{du}{dt} = \frac{\partial}{\partial x}HB^{-1}u \qquad (2.26)$$

or

$$\frac{du}{dt} = Du \qquad (2.27)$$

where D is differential matrix.

Equation (2.27) is a system of ODEs which can be integrated using a suitable time-stepping method. In this paper, the 4th order Runge-Kutta method is used as a time-stepping method.

Discretizing Derivatives with Radial Basis Functions

To discretize a derivative using RBFs $\phi(r)$, the chain rule for the first two derivatives according to Sarra and Kansa [11] are given as

$$\frac{\partial\phi}{\partial x_i} = \frac{d\phi}{dr}\frac{\partial r}{\partial x_i} \qquad (2.28)$$

and

$$\frac{\partial^2\phi}{\partial x_i^2} = \frac{d\phi}{dr}\frac{\partial^2 r}{\partial x_i^2} + \frac{d^2\phi}{\partial r^2}\left(\frac{\partial\phi}{\partial x_i}\right)^2 \qquad (2.29)$$

where

$$\frac{\partial r}{\partial x_i} = \frac{x_i}{r}$$

and

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

For the GMQ RBF having the value of $\beta = 1.03$,

$$\frac{d\phi}{dr} = 2.06\varepsilon^2 r(1 + \varepsilon^2 r^2)^{0.03}$$

and

$$\frac{d^2\phi}{\partial r^2} = 2.06\varepsilon^2 r(1 + \varepsilon^2 r^2)^{0.03} + \frac{0.1236\varepsilon^4 r^2}{(1 + \varepsilon^2 r^2)^{0.97}}$$

Similarly, for the GMQ RBF having the value of $\beta = 1.99$

$$\frac{d\phi}{dr} = 3.98\varepsilon^2 r(1 + \varepsilon^2 r^2)^{0.99}$$

and

$$\frac{d^2\phi}{dr^2} = 3.98\varepsilon^2 r(1 + \varepsilon^2 r^2)^{0.99} + \frac{7.8804\,\varepsilon^4 r^2}{(1 + \varepsilon^2 r^2)^{0.01}}.$$

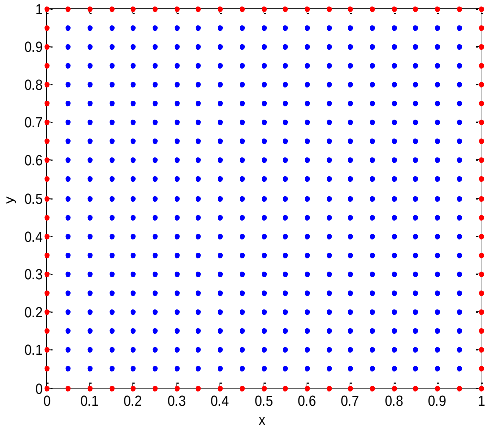
III. RESULTS

In the section, we present the numerical results of two steady state PDEs and a two-dimensional heat equation solved by the methods formulated and described in Section 2. These methods are implemented in MATLAB, while the results are displayed in Tables and graphs for comparison discussion and conclusion. All the programmes are written in Windows 8 Operating system using MATLAB 2017b. The test problems and parameter values are drawn from Sarra and Kansa [11].

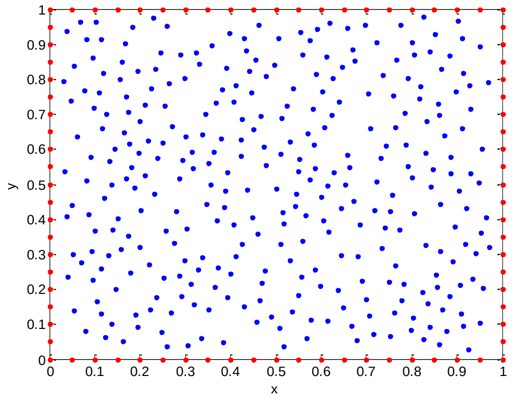
Domains and Data Points Distribution

First, we provide three different domains with data points that are also used as centres for approximating the solutions of the PDEs used as test problems. Figure 1(a) is a domain that consists of 441 points, 80 boundary points and 361 interior points. Similarly, the domain of Figure 1(b) is made up 399 points, 80 boundary points and 319 scattered (Halton) interior points. Figure 1(c) is complex domain with a total of 635 points, 310 boundary points and 505 interior points adopted from Sarra and Kansa [11].

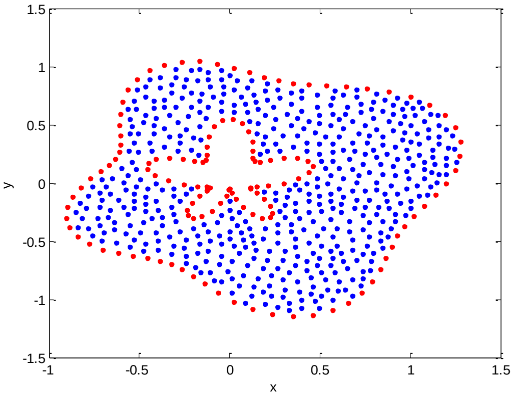
International Journal of Engineering Applied Sciences and Technology, 2020
Vol. 5, Issue 1, ISSN No. 2455-2143, Pages 1-12
Published Online May 2020 in IJEAST (http://www.ijeast.com)



(a)



(b)



(c)

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

Example 3.1

Consider the Poisson equation

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

where

$f(x, y) = a - b + c,$

$$a = \frac{130}{(65 + (x-0.2)^2 + (y-0.1)^2)^3 (2x-0.4)^2}$$

$$b = \frac{260}{(65 + (x-0.2)^2 + (y-0.1)^2)^2}$$

$$c = \frac{130}{(65 + (x-0.2)^2 + (y-0.1)^2)^3 (2y-0.2)^2}.$$

The exact solution is given by

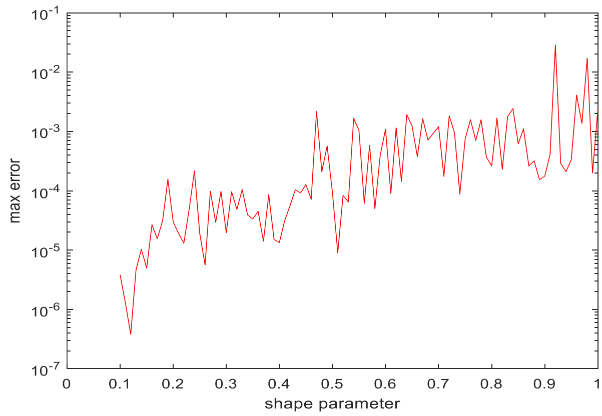
$$u(x, y) = \frac{65}{(65 + (x-0.2)^2 + (y-0.1)^2)}.$$

The Dirichlet boundary conditions, $u(x,0), u(x,1), u(0,y)$ and $u(1,y)$ are chosen to satisfy the exact solution.

This problem is implemented in MATLAB using equation (1.1) with $\beta = \frac{1}{2}, \beta = 1.03$ and $\beta = 1.99$ on the domains displayed in Fig. 1(a) and (b) and recorded in Table 1 and Figs. 2-5.

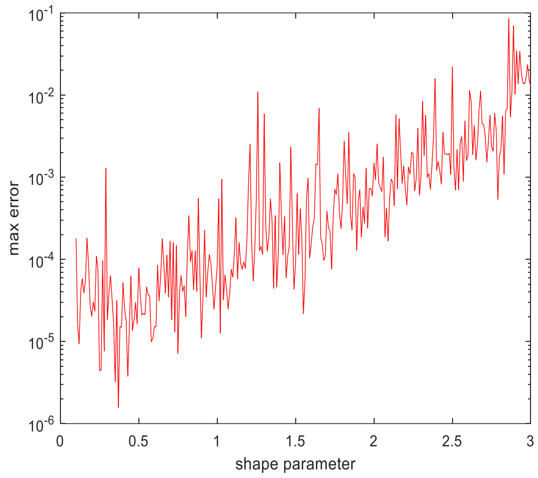
Table 1: Comparison of MQ, GMQ$(\beta = 1.03)$ and GMQ$(\beta = 1.99)$ for Example 3.1

S/N	RBF	Data Distribution	Shape Parameter (ε)	MPE
1	MQ	Equally Spaced	0.12	3.8161×10^{-6}
		Scattered	0.12	3.6105×10^{-7}
2	GMQ $(\beta = 1.03)$	Equally Spaced	0.4	1.8051×10^{-4}
		Scattered	0.5	1.1305×10^{-5}
3	GMQ $(\beta = 1.99)$	Equally Spaced	0.3	5.0062×10^{-4}
		Scattered	0.6	6.7215×10^{-5}

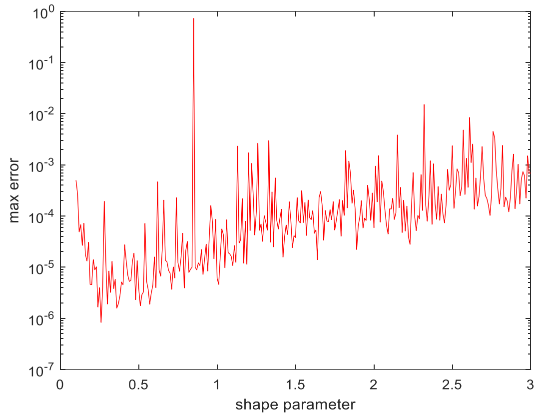


(a)

International Journal of Engineering Applied Sciences and Technology, 2020
Vol. 5, Issue 1, ISSN No. 2455-2143, Pages 1-12
Published Online May 2020 in IJEAST (http://www.ijeast.com)

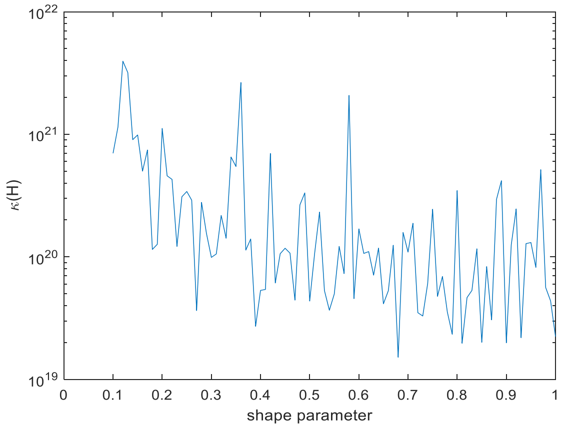


(b)

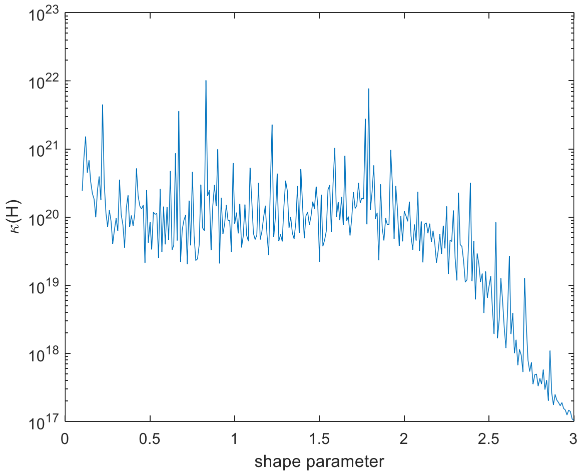


(c)

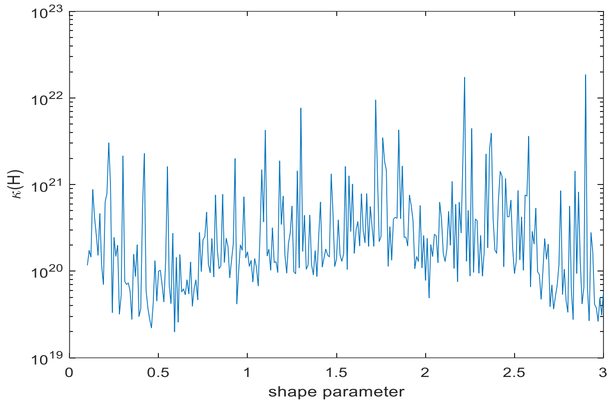
Fig. 2: Maximum Error versus the Shape Parameter for Example 3.1 on Equally Spaced Data Points using (a) MQ (b) GMQ($\beta = 1.03$) and (c) GMQ($\beta = 1.99$)



(a)

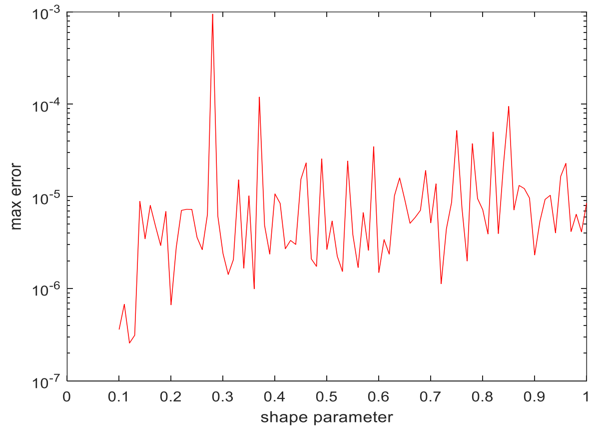


(b)

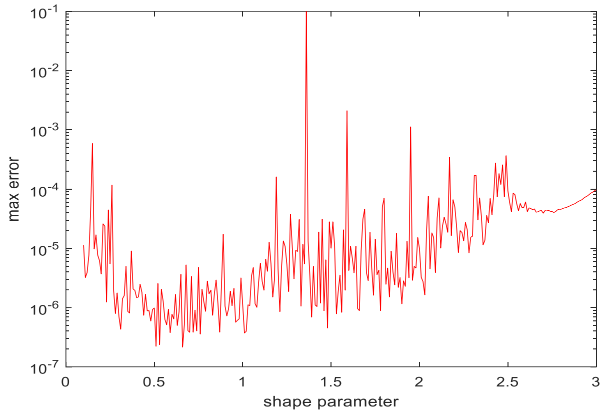


(c)

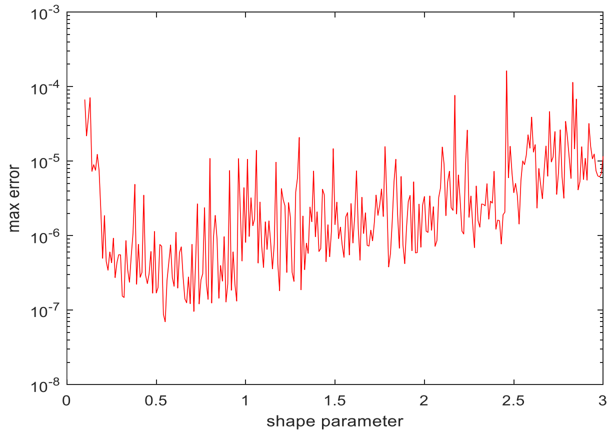
Fig. 3: Condition Number of the System Matrix for Example 3.1 on Equally Spaced Data Points using (a) MQ (b) GMQ($\beta = 1.03$) and (c) GMQ($\beta = 1.99$)



(a)

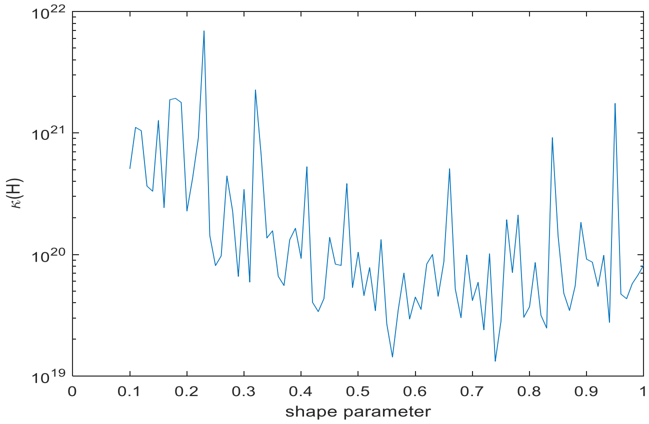


(b)

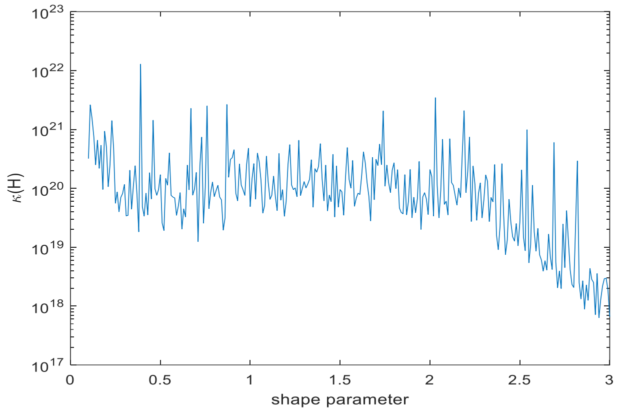


(c)

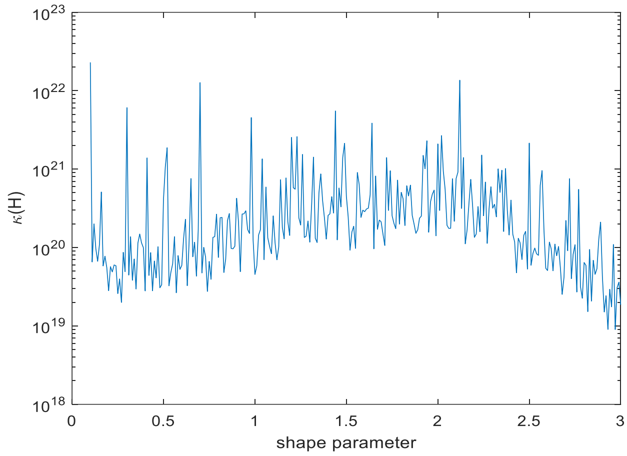
Fig. 4: Maximum Error versus the Shape Parameter for Example 3.1 on Scattered Data Points using (a) MQ (b) GMQ($\beta = 1.03$) and (c) GMQ($\beta = 1.99$)



(a)



(b)



(c)

Fig. 5: Condition Number of the System Matrix for Example 3.1 on Scattered Data Points using (a) MQ (b) GMQ($\beta = 1.03$) and (c) GMQ($\beta = 1.99$)

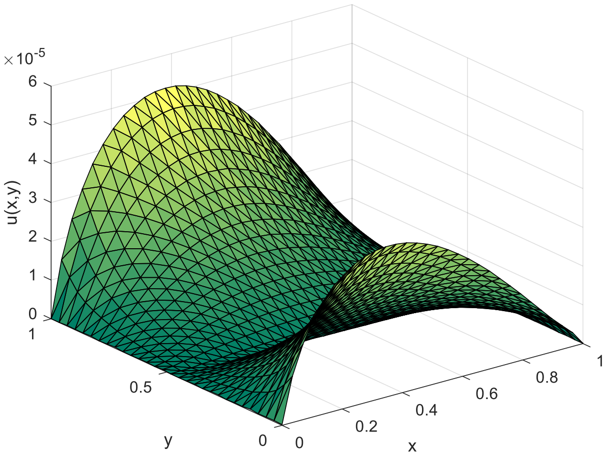
Example 3.2

Consider equation (3.1) such that
$f(x,y) = -2(2y^3 - 3y^2 + 1) + 6(1 - x^2)(2y - 1)$.
The exact solution is given by
$u(x,y) = (1 - x^2)(2y^3 - 3y^2 + 1)$,
the boundary conditions are given as
$u(0,y) = 2y^2 - 3y^2 + 1$
$u(1,y) = 0$
$\frac{\partial u(x,0)}{\partial y} = 0$ and
$\frac{\partial u(x,1)}{\partial y} = 1.$

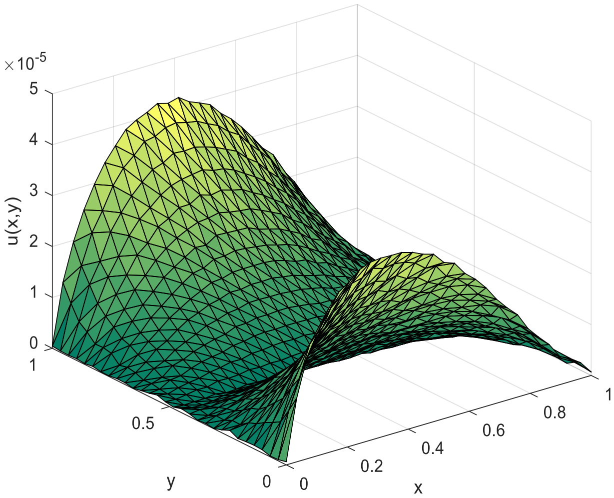
MATLAB programmes are used to implement this problem using equation (1.1) with $\beta = \frac{1}{2}, \beta = 1.03$ and $\beta = 1.99$ on the domains provided in Figs. 1(a) and 1(b) and recorded in Table 2 and Figs. 6 and 7.

Table 2: Comparison of MQ, GMQ$(\beta = 1.03)$ and GMQ$(\beta = 1.99)$ for Example 3.2

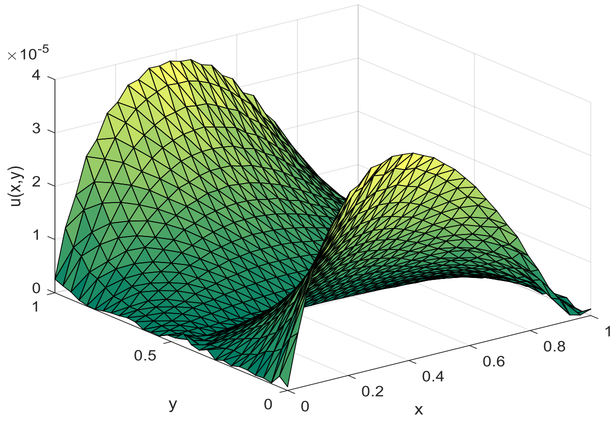
S/N	RBF	Data Distribution	Shape Parameter (ε)	MPE
1	MQ	Equally Spaced	2.0	6.7305×10^{-5}
		Scattered	2.5	5.1496×10^{-5}
2	GMQ $(\beta = 1.03)$	Equally Spaced	2.7	4.2046×10^{-5}
		Scattered	2.1	6.5898×10^{-5}
3	GMQ $(\beta = 1.99)$	Equally Spaced	3.5	3.9050×10^{-5}
		Scattered	2.6	6.4347×10^{-5}



(a)

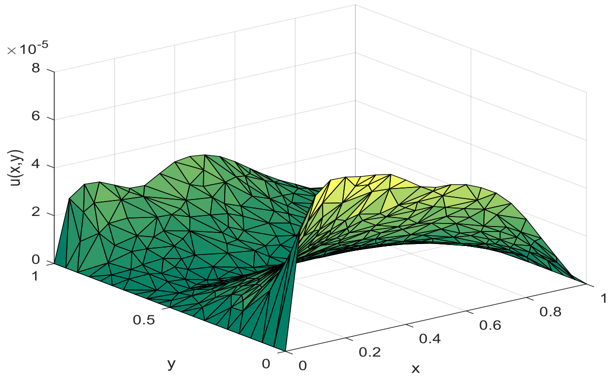


(b)

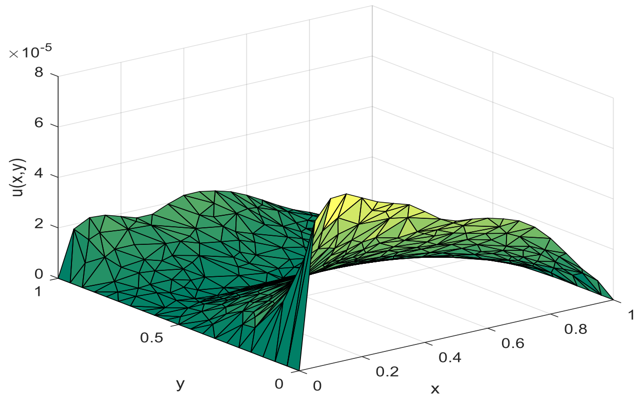


(c)

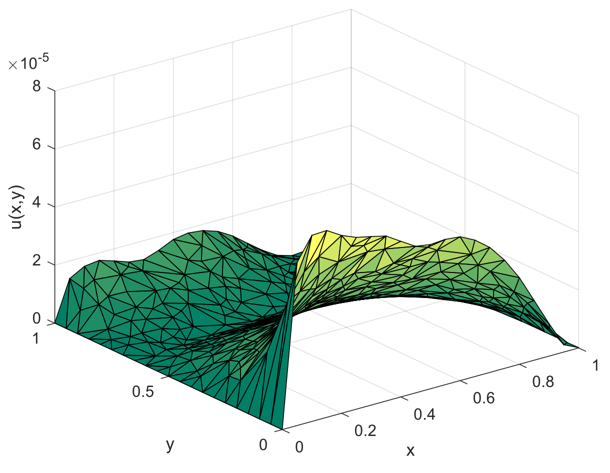
Figure 6: Numerical Solution of Example 3.2 on Equally Spaced Data Points using (a) MQ (b) GMQ$(\beta = 1.03)$ and (c) GMQ$(\beta = 1.99)$



(a)



(b)



(c)

Fig. 7: Numerical Solution of Example 3.2 on Scattered Data Points using (a) MQ (b) GMQ($\beta = 1.03$) and (c) GMQ($\beta = 1.99$)

Example 3.3

Consider the two-dimensional heat equation

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \qquad (3.3)$$

The Dirichlet boundary conditions are specified using the exact solution

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

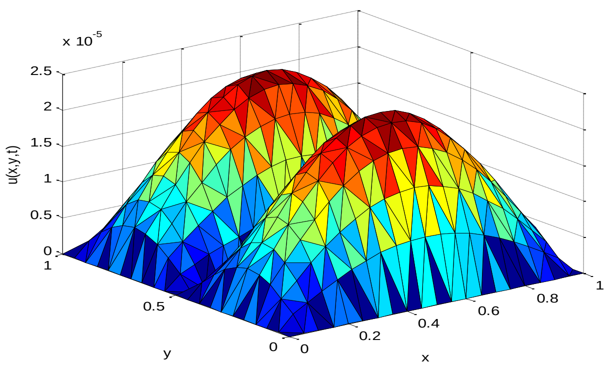
while the initial condition is given as

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

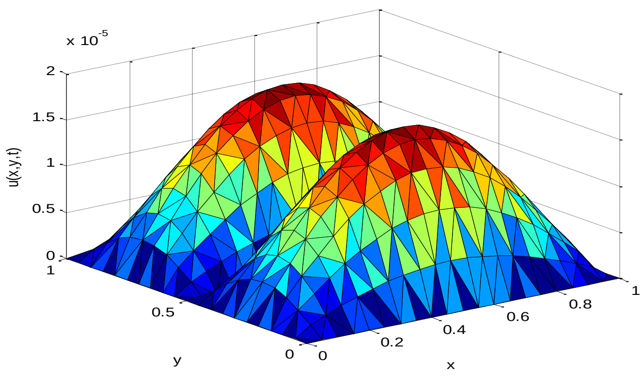
This problem is solved on three domains containing different patterns of spaced data points namely, equally spaced data points, Fig. 1(a), scattered data points, Fig. 1(b) and scattered points on a complex domain, Fig. 1(c) as shown in Fig. 1. The space derivatives are discretized using GMQ RBF for the values of the exponents $\beta = 1.03$ and $\beta = 1.99$ while the resulting system of ODEs are integrated using the fourth order Runge-Kutta method. The numerical results are displayed in Table 3 and Figs. 8 and 9.

Table 3: Comparison of MQ, IMQ, IQ, GIMQ, GMQ ($\beta = 1.03$) and GMQ ($\beta = 1.03$) RBF-MOLs for Example 3.3

S/N	RBF-MOLs	N	Δt	FT	ε	MPE	SOURCE
(a)	Solution on the Domain containing Equally Spaced Data Points						
1	MQ	441	5×10^{-5}	0.1	3.0	1.9418×10^{-5}	Sarra and Kansa[11]
2	IMQ	441	5×10^{-5}	0.1	2.5	2.0788×10^{-5}	Luga [20]
3	IQ	441	5×10^{-5}	0.1	2.5	2.3923×10^{-5}	Luga [20]
4	GIMQ	441	5×10^{-5}	0.1	2.0	2.0685×10^{-5}	Luga [20]
5	GMQ($\beta = 1.03$)	441	5×10^{-5}	0.1	3.0	1.7653×10^{-5}	
6	GMQ ($\beta = 1.99$)	441	5×10^{-5}	0.1	4.0	1.6944×10^{-5}	
(b)	Solution on the Domain containing Scattered Data Points						
1	MQ	399	5×10^{-5}	0.1	3.0	1.8416×10^{-5}	Sarra and Kansa [11]
2	IMQ	399	5×10^{-5}	0.1	2.0	1.7635×10^{-5}	Luga [20]
3	IQ	399	5×10^{-5}	0.1	2.0	1.7790×10^{-5}	Luga [20]
4	GIMQ	399	5×10^{-5}	0.1	2.0	1.8526×10^{-5}	Luga [20]
5	GMQ ($\beta = 1.03$)	399	5×10^{-5}	0.1	2.5	1.7475×10^{-5}	
6	GMQ ($\beta = 1.99$)	399	5×10^{-5}	0.1	3.5	1.7458×10^{-5}	
(c)	Solution on the Complex Domain Containing Scattered Data Points						
1	MQ	635	5×10^{-4}	0.1	2.0	3.7527×10^{-6}	Sarra and Kansa[11]
2	IMQ	635	5×10^{-4}	0.1	1.5	1.4914×10^{-6}	Luga [20]
3	IQ	635	5×10^{-4}	0.1	1.5	2.1306×10^{-6}	Luga [20]
4	GIMQ	635	5×10^{-4}	0.1	1.5	3.5593×10^{-6}	Luga [20]
5	GMQ ($\beta = 1.03$)	635	5×10^{-4}	0.1	2.5	7.2991×10^{-6}	
6	GMQ ($\beta = 1.99$)	635	5×10^{-4}	0.1	2.5	2.0994×10^{-6}	

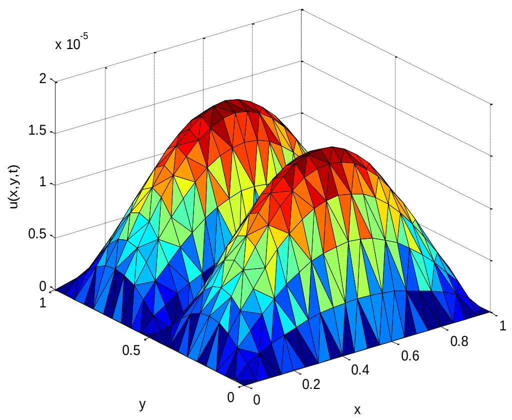


(a)



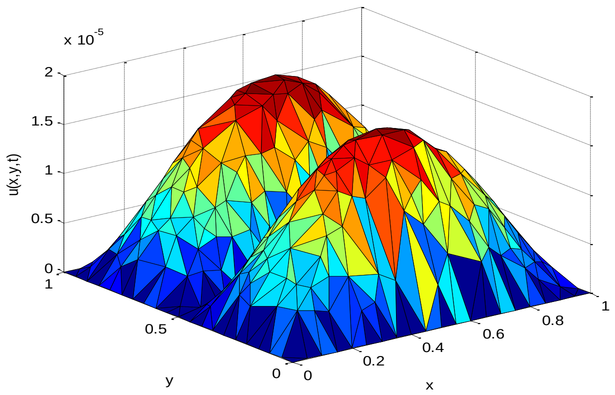
(b)

International Journal of Engineering Applied Sciences and Technology, 2020
Vol. 5, Issue 1, ISSN No. 2455-2143, Pages 1-12
Published Online May 2020 in IJEAST (http://www.ijeast.com)

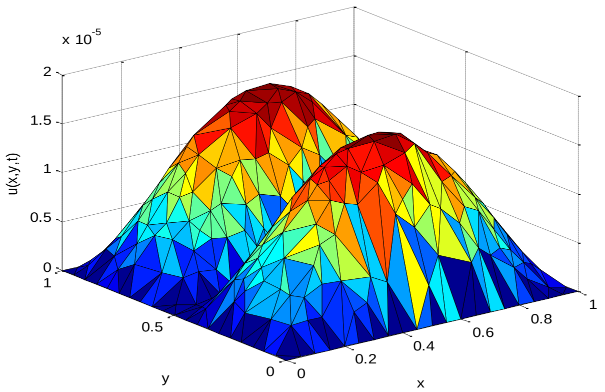


(
(c)

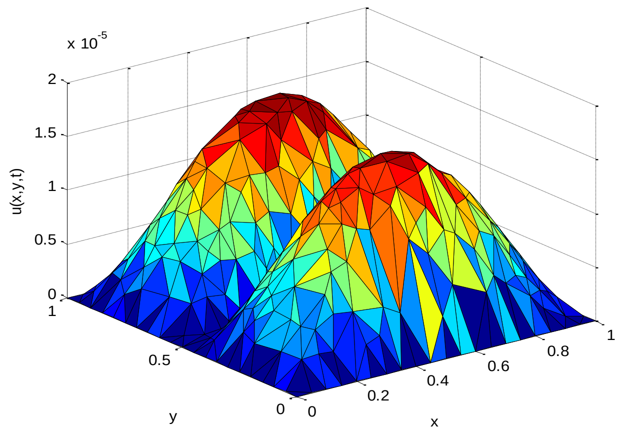
Fig. 8: Numerical Solution of Example 3.3 on Equally Spaced Data Points using (a) MQ (b) GMQ($\beta = 1.03$) and (c) GMQ($\beta = 1.99$)



(a)



(b)



(c)

Fig. 9: Numerical Solution of Example 3.3 on Scattered Data Points using (a) MQ (b) GMQ($\beta = 1.03$) and (c) GMQ($\beta = 1.99$)

IV. DISCUSSION

Steady State Partial Differential Equations

The solutions of two Poisson partial differential equations in two space dimensions were approximated using the GMQ-RBFs having the exponents, $\beta = \frac{1}{2}$, $\beta = 1.03$ and $\beta = 1.99$. The first problem has Dirichlet boundary conditions, while the second contains Neumann boundary conditions. Both problems were approximated using the data points on the domains, $\Omega = [0,1]$ as shown in Figs. 1(a) and (b). The shape parameters (ε) were obtained by plotting the maximum error versus the shape parameter where the minimum points of the plots as displayed in Figs. 2(a), (b), (c), 4(a), (b) and (c) were chosen as suitable estimates for the shape parameters to ensure stable and accurate numerical approximations. The numerical results for Example 3.1 recorded in Table 1 shows that the maximum point-wise error (MPE) from MQ-RBF method yielded the least error on both domains containing equally spaced data points and scattered data points. For Example 3.2, Table 2 shows that all the RBFs used for approximating solution of this PDE yielded approximately the same MPE on both data points. The plots for the various RBFs are provided in Figs. 6 and 7.

The Two-Dimensional Heat Equation

Example 3.3 is a two-dimensional heat equation. Sarra and Kansa [11] solved this particular problem using MQ RBF-MOLs on a complex domain, Figure 1(c). Also, Luga [20] solved the same problem using IMQ, IQ and GIMQ RBF-MOLs on all the domains displayed in Figures 1(a), (b) and (c). In this paper, we approximated the solution of this problem using two positive non-half-integer/non-integer exponent GMQ RBF-MOLs with the exponents $\beta = 1.03$ and

$\beta = 1.99$ on all the domains provided in Section III. The space derivatives were discretized using GMQ with $\beta = 1.03$ and $\beta = 1.99$, while the fourth order Runge-Kutta method was used to integrate the resulting system of ODEs. A step size of $\Delta t = 5.0 \times 10^{-5}$ was applied on the domains in Figs. 1(a) and (b), while $\Delta t = 5.0 \times 10^{-4}$ was used on the complex domain, Fig. 1(c) to advance the solution up to the final time $FT = 0.1$. The numerical solutions for the two-dimensional heat equation is displayed in Table 3 and Figs. 8 and 9. Comparing the numerical results obtained from MQ RBF-MOLs of Sarra and Kansa [11], IMQ, IQ and GIMQ RBF-MOLs of Luga [20] with GMQ RBF-MOLs constructed with GMQ RBFs having $\beta = 1.03$ and $\beta = 1.99$ using maximum point-wise error (MPE) revealed that all the RBF-MOLs considered yielded approximately the same accuracy.

V. CONCLUSION

The GMQ-RBFs with the values of the exponents $\beta = 1.03$ and $\beta = 1.99$ were used to develop numerical methods for approximating the solutions of two steady state PDEs and a heat equation in two-dimensions. The accuracy of the RBF methods were measured in terms of maximum point-wise errors (MPE) and compared with some standard GMQ RBFs. Our numerical experiments show that all the GMQ RBF methods produced nearly the same relative accuracy, regardless of the value of the exponent β. Our results also suggest that the claims that some values of β produce optimal results is not true, instead the value of β in equation (1.1) is problem dependent.

VI. REFERENCES

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

[2] Kansa, E. J. (2007). Explicit Time Integration of Hyperbolic Partial Differential Equations with Mesh Free Radial Basis Functions. *Engineering Analysis with Boundary Elements*, 31:577-585.

[3] Hardy, R. L. (1971). Multiquadric Equations of Topography and Other Irregular Surfaces. *Journal of Geographical Research*, 76: 1905-1915.

[4] 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.

[5] 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.

[6] Chenoweth, M. (2009). A Numerical Study of Generalized Multiquadric Radial Basis Function Interpolation. *Submitted to SIAM Undergraduate Research Online*, 70pp.

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

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

[9] Madych, W. R. and Nelson, S. A. (1988). Multivariate Interpolation and Conditionally Positive Definite Functions. *Approximation Theory and its Applications*, 4: 77-89.

[10] Madych, W. R. (1992). Miscellaneous Error Bounds for Multiquadric and Related Interpolators. *Computer and Mathematics with Applications*, 241(12):121-138.

[11] 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.

[12] 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. 139 pp.

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

[14] Luga, T, Aboiyar, T. and Onah, E. S. (2017b) Radial Basis Function Methods for Approximating the Cubic Nonlinear Schrödinger Equation in One Space Dimension. *Journal of the Nigerian Association of Mathematical Physics*, 43: 325-332.

[15] Kansa, E. J., Aldredge, R. C. and Ling, L. (2009). Numerical Simulation of Two-Dimensional Combustion using Mesh-Free Methods. *Engineering Analysis with Boundary Elements*, 33(7): 940-950.

[16] Wertz, J., Kansa, E. J. and Ling, L. (2006). The Role of the Multiquadric Shape Parameter in Solving Elliptic Partial Differential Equations. *Computer and Mathematics with Applications*, 51(8):1335-1348.

[17] Wang, J. G. and Liu, G. R. (2002). On the Optimal Shape Parameters of Radial Basis Functions used for 2-D Meshless Methods. *Computer Methods in Applied Mechanics and Engineering*, 191: 2611-2630.

[18] Xiao, J. R. and McCarthy, M. A. (2003). A Local Heaviside Weighted Meshless Method for Two-Dimensional Solids using Radial Basis Functions. *Computational Mechanics*, 31: 301-315.

[19] Xiao, J. R., Gama, B. A., Gillespie (Jr.), J. W. and Kansa, E. J. (2005). Meshless Solutions of 2D Contact Problems by Subdomain Variational Inequality and Meshless Local Petrov-Galerkin Method with Radial

Basis Functions. *Engineering Analysis with Boundary Elements*, 29: 95-106.

[20] Luga (2018). Solving Time-Dependent Partial Differential Equations in One and Two Space Dimensions by Radial Basis Function Method of Lines. Ph. D. Thesis submitted to the Department of Mathematics/Statistics/Computer Science, University of Agriculture, Makurdi, Nigeria. 164 pp.