IOP Conf. Series: Journal of Physics: Conf. Series **1489** (2020) 012020 doi:10.1088/1742-6596/1489/1/012020

A Numerical Study of a Compactly-Supported Radial Basis Function Applied with a Collocation Meshfree Scheme for Solving PDEs

S Tavaen [1], K Chanthawara [2] and S Kaennakham [3,*]

[1,3]School of Mathematics, Institute of Science, Suranaree University of Technology, Nakhon Ratchasima 30000, Thailand
[2] Program of Mathematics, Faculty of Science, Ubon Ratchathani Rajabhat Universit, Mueang Ubon Ratchathani, Ubon Ratchathani 34000, Thailand
[*]Centre of Excellence in Mathematics, Bangkok 10400, Thailand

E-mail: *sayan_kk@g.sut.ac.th

Abstract. It is known that all Radial Basis Function-based meshfree methods suffer from a lack of reliable judgement on the choice of shape parameter, appearing in most of the RBFs. While the popularity of meshfree/meshless numerical methods is growing fast over the past decade, the great challenge is still to find an optimal RBF form with its optimal shape parameter. In this work, the main focus is on one type of RBF namely 'Compactly-Supported (CS-RBF)' that contains no parameter, and yet has not been explored numerically as much in the past, particularly under the context of data interpolation/approximation and solving partial differential equations (PDEs). To compare the potential advantages of CS-RBF, two most popular choices of RBF widely used; Multiquadric (MQ), and Gaussian (GA) were studied parallelly. The information gathered and presented in this work shall be useful for the future users in making decision on RBF.

1. Introduction

Amongst the well-known numerical schemes; finite volume, finite difference, and finite element method that have been invented, developed, and applied in a wide range of science and engineering problems, a rather young idea was discovered and has been categorized as 'meshless/meshfree' methods [1, 2]. The methods under this category have recently become promising alternative tools for numerically solving variety of science and engineering problems. Generally, these meshless schemes can be grouped into two main classes [3];

- Strong forms ; The finite point method [4], The hp-meshless cloud method [5], the collocation method [6], and references therein.
- Weak forms ; The diffuse element method [7], The element-free Galerkin method (EFGM; [8], The point interpolation method [9], and references therein.

Each of these has its own advantages/disadvantages depending on several factors involved; domain geometry, governing equations, boundary/initial conditions, computer arithmetic, etc. Amongst those being proposed and developed nowadays, one of the well-known meshfree method is that called 'RBF-collocation' or sometimes called 'Kansa's method', where it uses a set of global approximation function

to approximate the field variables on both the domain and the boundary when solving PDEs [10] . In this work, particularly, the main attention is paid on the application of the so-called 'Collocation Method' characterized in the strong form family. The method is truly meshfree meaning no mesh is at all required at any point of the whole computing process.

For the method of collocation meshfree, there are several forms of well-known radial basis function and some are listed below, [11];

- Gaussian (GS), defined as;
$$\varphi(r)=e^{-(\varepsilon r)^2}$$
- Inverse Multiquadric (IMQ) , defined as;
$$\varphi(r)=1/\sqrt{1+(\varepsilon r)^2}$$
- Multiquadric (MQ) , defined as;
$$\varphi(r)=\sqrt{1+(\varepsilon r)^2}$$
- Inverse quadratic (IQ) , defined as;
$$\varphi(r)=1/\left(1+(\varepsilon r)^2\right)$$
- Wendland C^6 (WL) , defined as;
$$\varphi(r)=(1-\varepsilon r)^8\left(32(\varepsilon r)^3+25(\varepsilon r)^2+8\varepsilon r+1\right)$$
- Cubic Matérn (CU) , defined as;
$$\varphi(r)=e^{-\varepsilon r}\left(15+15\varepsilon r+6(\varepsilon r)^2+(\varepsilon r)^3\right)$$

Here, ε is called shape parameter, determined by the user and this exactly remains the open topic for researchers to investigate the possible optimal choice. A lot of attempts have been made to alleviate this problem but by far no universal choice has been found [12-14]. The attempt to get rid of this burden by turning attention away from these groups of RBFs to those containing no parameter is now being made.

For this purpose, this work focusses on a compactly-supported form of RBF proposed by Buhmann [15] and the effectiveness one can obtained when applied this type of RBF with the collocation-based method.

- Noted as CS1 and defined as;
$$\varphi(r)=\left(119r^{9/2}/45+16r^{7/2}/3-7r^4-14r^2/15+1/9\right)_+$$
- Noted as CS2 and defined as;
$$\varphi(r)=\left(-4r^3\log(r)/3+r^4/2+4r^3/9-r^2+1/18\right)_+$$
- Noted as CS3 and defined as;
$$\varphi(r)=\left(2r^4\log(r)-7r^4/2+16r^3/3-2r^2+1/6\right)_+$$

when $+$ indicates the cut-off function which is defined to be r , when $0\le r\le 1$ and zero elsewhere.

2. Mathematical Background

Radial Basis Functions (RBF), φ , are commonly found as multivariate functions whose values are dependent only on the distance from the origin. This means that $\varphi(\mathbf{x})=\varphi(r)\in R$ with $\mathbf{x}\in\mathbb{R}^n$ and $r\in R$; or, in other words, on the distance from a point of a given set $\{\mathbf{x}_j\}$, and $\varphi(\mathbf{x}-\mathbf{x}_j)=\varphi(r_j)\in R$ where can normally be defined as follows;

$$r=\left\|\mathbf{x}-\mathbf{x}^{\Theta}\right\|_2=\sqrt{(x_1-x_1^{\Theta})^2+(x_2-x_2^{\Theta})^2+...+(x_n-x_n^{\Theta})^2} \tag{1}$$

2.1. RBF collocation scheme for PDFs

For and some fixed points $\mathbf{x}\in\mathbb{R}^n$. Nevertheless, in this work, $r_j=\left\|\mathbf{x}-\mathbf{x}_j\right\|_2$ is the Euclidean distance and any function φ satisfying $\varphi(\mathbf{x})=\varphi(\|\mathbf{x}\|_2)$, is called a radial function. For the methodology of RBF-collocation meshless method for numerically solving PDEs, it begins with considering a linear elliptic partial differential equation with boundary conditions, where $g(\mathbf{x})$ and $f(\mathbf{x})$ are known. We seek $u(\mathbf{x})$ from;

$$Lu(\mathbf{x})=f(\mathbf{x}),\qquad \mathbf{x}\ \text{in}\ \Omega \tag{2}$$

$$Mu(\mathbf{x})=g(\mathbf{x}),\qquad \mathbf{x}\ \text{on}\ \partial\Omega \tag{3}$$

where $\Omega\in\mathbb{R}^d$, $\partial\Omega$ denotes the boundary of domain Ω, L and M are the linear elliptic partial differential operators and operating on the domain Ω and boundary domain $\partial\Omega$, respectively. For Kansa's method, it represents the approximate solution $u(\mathbf{x})$ by the interpolation, using an RBF interpolation as the following;

$$u(\mathbf{x})=\sum_{j=1}^{N}c_j\varphi_j(r)=\sum_{j=1}^{N}c_j\varphi\left(\left\|\mathbf{x}-\mathbf{x}_j\right\|\right) \tag{4}$$

We can see that N linear dependent equations are required for solving N unknowns of c_j.

Substituting $u(\mathbf{x})$ into equation (2) and equation (3), we obtain the system of equations as follows;

$$L\left(\sum_{j=1}^{N_I}c_j\varphi\left\|\mathbf{x}-\mathbf{x}_j\right\|\right)=\sum_{j=1}^{N_I}c_jL\varphi\left(\left\|\mathbf{x}-\mathbf{x}_j\right\|\right)=f(\mathbf{x}_i),\ \ i=1,\ldots,N_I \tag{5}$$

$$M\left(\sum_{j=N_I+1}^{N}c_j\varphi\left\|\mathbf{x}-\mathbf{x}_j\right\|\right)=\sum_{j=N_I+1}^{N}c_jM\varphi\left(\left\|\mathbf{x}-\mathbf{x}_j\right\|\right)=g(\mathbf{x}_i),\ \ i=N_I+1,\ldots,N \tag{6}$$

Above equations, we choose N collocation points on both domain Ω and boundary domain $\partial\Omega$, and divide it into N_I interior points and N_B boundary points $(N=N_I+N_B)$. Let $X=\{\mathbf{x}_1,\mathbf{x}_2,\ldots,\mathbf{x}_N\}$ denotes the set of collocation points, $I=\{I_1,\ldots,I_{N_I}\}$ denotes the set of interior points and $B=\{B_1,\ldots,B_{N_B}\}$ the set of boundary points. The centers $\mathbf{x}_j$ used in equation (5) and equation (6) are chosen as collocation points. The previous substituting yields a system of linear algebraic equations which can be solved for seeking coefficient $\mathbf{c}$ by rewriting equation (5) and equation (6) in matrix form as;

$$\mathbf{Ac}=\mathbf{F} \tag{7}$$

where $\mathbf{A}=\begin{bmatrix}\mathbf{A}_L\\ \mathbf{A}_M\end{bmatrix}$ and $\mathbf{F}=\begin{bmatrix}f(\mathbf{x}_i)\\ g(\mathbf{x}_i)\end{bmatrix}$ with the following detail;

$$(\mathbf{A}_L)_{ij}=L\varphi\left(\left\|\mathbf{x}_i-\mathbf{x}_j\right\|\right)\ ,\ \mathbf{x}_i\in I,\ \mathbf{x}_j\in X\ \ ,\ i=1,2,\ldots,N_I,\ \ j=1,2,\ldots,N$$

$$(\mathbf{A}_M)_{ij}=M\varphi\left(\left\|\mathbf{x}_i-\mathbf{x}_j\right\|\right)\ \ ,\ \mathbf{x}_i\in B,\ \mathbf{x}_j\in X\ \ ,\ i=N_I+1,\ldots,N,\ \ \ j=1,2,\ldots,N,$$

$f(\mathbf{x}_i); \mathbf{x}_i \in I, \quad i = 1, 2, \ldots, N_I,$
$g(\mathbf{x}_i); \mathbf{x}_i \in B, \quad i = N_I + 1, \ldots, N.$

Equation (7), the coefficient $\mathbf{c}'s$ are computed from the following system;

$$\mathbf{c} = \mathbf{A}^{-1}\mathbf{F} \tag{8}$$

Therefore, the matrix $\mathbf{c}$ is substituted into (4) and the approximate solution of $u(\mathbf{x})$ can be determined by;

$$u(\mathbf{x}) = \sum_{j=1}^{N} c_j \varphi\left(\left\|\mathbf{x} - \mathbf{x}_j\right\|\right) \tag{9}$$

The system is known to provide solution if and only if the matrix $\mathbf{A}$ is non-singular, its inverse exists.

2.2. Numerical Treatment for Nonlinear PDE

The following procedure was presented by Fasshauer [16] and is being briefly revisited here. The computation process starts with considering the nonlinear PDEs of the form

$$Lu = f \tag{10}$$

With the following algorithm;

1. Create the collocation point sets $X \subset \Omega$ and start with an initial guess u_0
2. For $k = 1, 2, \ldots, K$
   (a) The linearized problem
   $$\Lambda_{u_{k-1}} v = f - L u_{k-1} \quad \text{on} \quad X \tag{11}$$
   (b) Perform the Newton update
   $$\tilde{v} = v \tag{12}$$
   (c) Update the previous approximation
   $$u_k = u_{k-1} + \tilde{v} \tag{13}$$

In this algorithm $\Lambda_{u_{k-1}}$ is the linearization of the nonlinear differential operator L at u_{k-1} .

Here we provide an example in order to elaborate the algorithm described above.

Considering the nonlinear PDE of the form[2];

$$-\mathrm{E}^2\nabla^2 u - u + u^3 = f \quad \text{in } \Omega = (0,1)\times(0,1) \tag{14}$$

and

$$u(x, y) = 0 \quad \text{on } \partial\Omega \tag{15}$$

On the right hand side, f is chosen so that equation (14) has an analytic solution of the form

$$u(x, y) = \psi(x)\psi(y) \tag{16}$$

with $\psi(t) = 1 + e^{-1/\mathrm{E}} - e^{t/\mathrm{E}} - e^{(t-1)/\mathrm{E}}$, (x, y) denotes the Cartesian coordinates of $\mathbf{x} \in \mathbb{R}^2$, and the parameter ε determines the size of the boundary layers near the edges of the domain Ω . We use a value of $\mathrm{E} = 0.1$

For this model problem the linearization $\Lambda_{u_{k-1}}$ of L is given by

$$\Lambda_{u_{k-1}} v = -\mathrm{E}^2\nabla^2 v + \left(3u_{k-1}^2 - 1\right)v \tag{17}$$

and therefore the equation to be solved in step 2a) of the algorithm is of the form;

$$-\mathrm{E}^2\nabla^2 v+\left(3u_{k-1}^2-1\right)v=f+\mathrm{E}^2\nabla^2 u_{k-1}+u_{k-1}-u_{k-1}^3 \tag{18}$$

When perform step 2a), it is necessary to solve the following linear system, arising from the nonsymmetrical collocation procedure

$$\sum_{j=1}^{n^{(k)}} c_j^{(k)} \Lambda_{u_{k-1}}\left[\varphi\left(\left\|\mathbf{x}-\mathbf{x}_j^{(k)}\right\|\right)\right]_{x=x_j^{(k)}} = f+\Lambda u_{k-1}\left(\mathbf{x}_i^{(k)}\right),\ i=1,\ldots,n_I^{(k)} \tag{19}$$

$$\sum_{j=1}^{n} c_j^{(k)}\left[\varphi\left(\left\|\mathbf{x}_i^{(k)}-\mathbf{x}_j^{(k)}\right\|\right)\right]=0 \quad ,\ i=n_I^{(k)}+1,\ldots,n^{(k)} \tag{20}$$

By using equation (11-13) , the above then becomes;

$$\sum_{j=1}^{n} c_j\left[-\mathrm{E}^2\left[\nabla^2\varphi\right]_{ij}+\left(3u_i^2-1\right)\varphi_{ij}\right]=f+\mathrm{E}^2\left[\tilde{\nabla}^2 u\right]_i+u_i-u_i^3 \quad ,\ i=1,\ldots,n_I \tag{21}$$

$$\sum_{j=1}^{n} c_j\varphi_{ij}=0 \quad ,\ i=n_I+1,\ldots,n_I+n_B=n \tag{22}$$

where, for transparency, the index $k's$ are being omitted on the quantities n, n_I, n_B , and c_j . The Newton update $\tilde{v}=v$ used in step 2c) of the algorithm is then given by;

$$v(\mathbf{x})=\sum_{j=1}^{n^{(k)}} c_j^{(k)}\varphi\left(\left\|\mathbf{x}-\mathbf{x}_j^{(k)}\right\|\right) \tag{23}$$

The next section, the whole process of numerical computing is implemented for solving nonlinear partial differential equations and for this 4 problems are tested. All solutions obtained from this investigation are justified by compared with their corresponding exacts.

3. Numerical Measurement Norms

All computing experiments were carried out on the same computer; Intel(R) Core(TM) i7-5500U CPU @ 2.40GHz with 8.00 GB of RAM and 64-bit Operating System. All numerical solutions obtained from the whole study are validated by comparing against the analytical solutions using different types of error norms as listed in Table 1, or some other numerical works where available in literature.

Table 1. Error Norms adopted in this work.

Error Norm	Symbol Defined	Mathematical Formula
Maximum	L_∞	$\max\limits_{1\le i\le N}\left\|u^{ext.}(\mathbf{x}_i)-u^{appx.}(\mathbf{x}_i)\right\|$
Root-Mean-Square	L_{RMS}	$\left(\frac{1}{N}\sum\limits_{j=1}^{N}\left(u^{ext.}(\mathbf{x}_i)-u^{appx.}(\mathbf{x}_i)\right)^2\right)^{1/2}$
Absolute	L_{Abs}	$\left\|u^{ext.}(\mathbf{x}_i)-u^{appx.}(\mathbf{x}_i)\right\|$

4. Numerical Experiments

4.1. Test 1 : Interpolation Problem

The interpolation problem starts with a set of discrete data $\mathbf{X}=\{\mathbf{x}_i\}_{i=1}^N, \mathbf{x}_i\in\mathbb{R}^{\mathrm{d}}$ where for each $\mathbf{x}_i$ there is its corresponding real value $y_i\in\mathbb{R}$, then the task is to construct a continuous function $\Phi(\mathrm{x}):\mathbb{R}^{\mathrm{d}}\to\mathbb{R}$ such that;

$$\Phi(\mathbf{x}_i)=y_i \tag{24}$$

For all $i = 1, 2, ..., N$ .The first validation of the scheme proposed in this work is done using a benchmark function defined on a square-domain, $[0, 4\pi] \times [0, 4\pi]$, as follows.

$$f(x, y) = \sin(x)\cos(y) \tag{25}$$

As regarded as one of the most influential factors, the shape parameter is firstly investigated. For this purpose simulations using the Multiquadric (MQ) and Gaussian (GA) were under the investigation. Figure 1. clearly show the strong effects the shape parameter has for both RBFs. When using different numbers of nodes, it also can be seen that the errors norms can be affected as well.

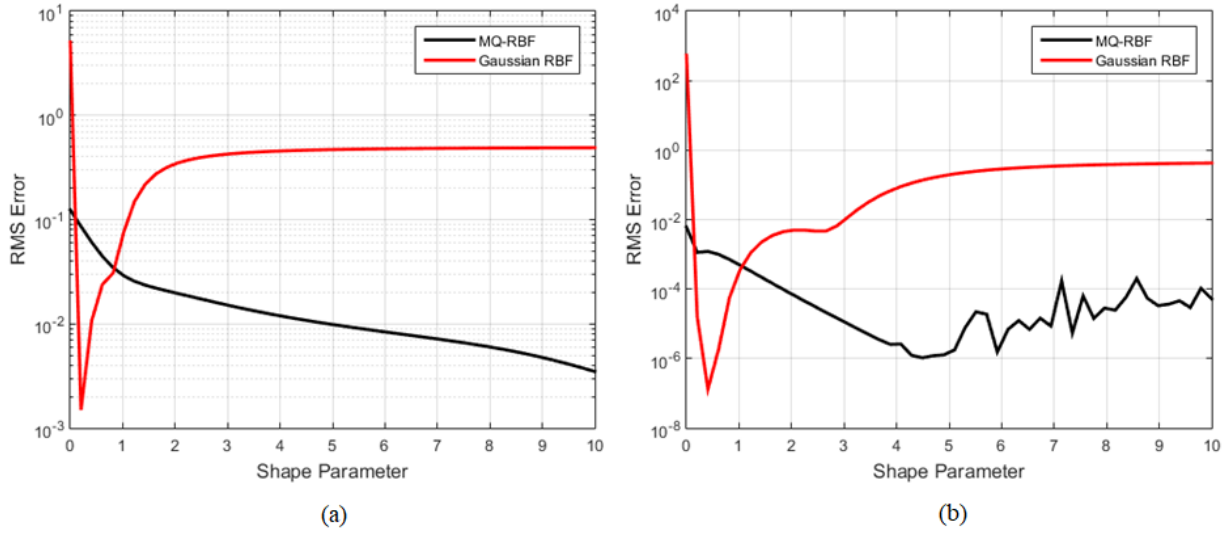


Figure 1. L_{RMS} measured at different values of shape parameters; left) with 10×10 interpolation nodes/centres, and right) with 30×30 interpolation nodes/centres.

At the node density of 15×15, Table 2 provides the absolute errors measured at different locations all over the domain. The information contained in this table show significant accuracy obtained from all types of RBFs. Nevertheless, for MQ and GA, a caution has been previously taken on fining the optimal shape values and they were found to be in $(4, 6.5)$ for MQ, $(0.1 - 0.8)$ and for GA. It must be noted that no extra treatment was needed for the cases of using compactly-supported RBFs. Solutions produced by all the cases were plotted against one another as shown in figure 2. and figure 3.

Table 2. L_{Abs} measured at different locations over the computational domain with 15×15 interpolation nodes/centres

x	y	MQ ($\varepsilon = 5.00$)	GA ($\varepsilon = 0.50$)	CS1	CS2	CS3
0.000	0.000	8.59E-10	1.57E-12	1.46E-11	3.55E-11	1.46E-11
0.000	1.396	1.87E-06	2.73E-12	4.17E-04	1.09E-02	5.42E-04
0.000	12.566	1.32E-09	3.64E-13	5.46E-11	7.84E-12	6.37E-12
1.396	0.000	7.06E-06	1.60E-05	6.66E-04	7.37E-05	3.86E-03
2.793	0.000	2.56E-06	5.86E-06	3.49E-04	1.41E-03	1.07E-02
2.793	1.396	7.73E-06	1.13E-05	8.23E-04	5.66E-03	9.36E-04
2.793	12.566	2.55E-06	5.86E-06	3.49E-04	1.41E-03	1.07E-02
4.189	0.000	1.12E-06	2.43E-06	5.30E-04	1.21E-02	1.55E-02
4.189	12.566	1.12E-06	2.43E-06	5.30E-04	1.21E-02	1.55E-02
5.585	12.566	3.32E-07	6.88E-07	3.51E-04	6.51E-03	1.64E-02
6.981	0.000	3.32E-07	6.88E-07	3.51E-04	6.51E-03	1.64E-02
6.981	12.566	3.32E-07	6.88E-07	3.51E-04	6.51E-03	1.64E-02
8.378	0.000	1.12E-06	2.43E-06	5.30E-04	1.21E-02	1.55E-02
8.378	12.566	1.12E-06	2.43E-06	5.30E-04	1.21E-02	1.55E-02
9.774	0.000	2.55E-06	5.86E-06	3.49E-04	1.41E-03	1.07E-02
9.774	11.170	7.73E-06	1.13E-05	8.23E-04	5.66E-03	9.36E-04
9.774	12.566	2.56E-06	5.86E-06	3.49E-04	1.41E-03	1.07E-02
11.170	0.000	7.07E-06	1.60E-05	6.66E-04	7.37E-05	3.86E-03
11.170	12.566	7.06E-06	1.60E-05	6.66E-04	7.37E-05	3.86E-03
12.566	0.000	1.09E-09	3.76E-13	1.20E-10	6.24E-11	2.64E-11
12.566	12.566	2.26E-09	9.46E-13	5.91E-13	5.34E-11	4.06E-11

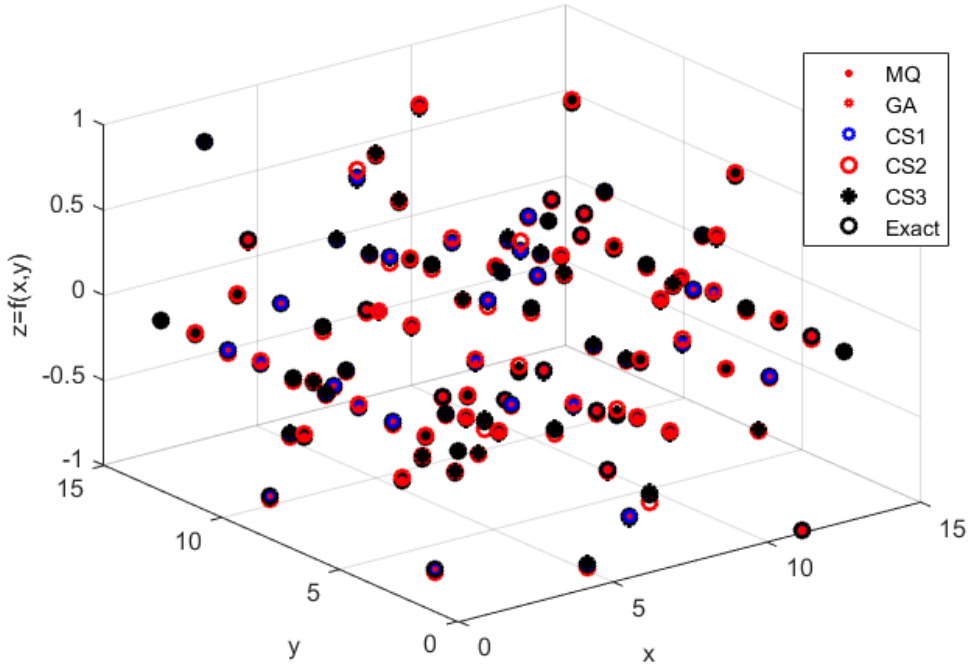


Figure 2. Solution approximation at selected locations obtained from each RBF type, compared again the exact solution.

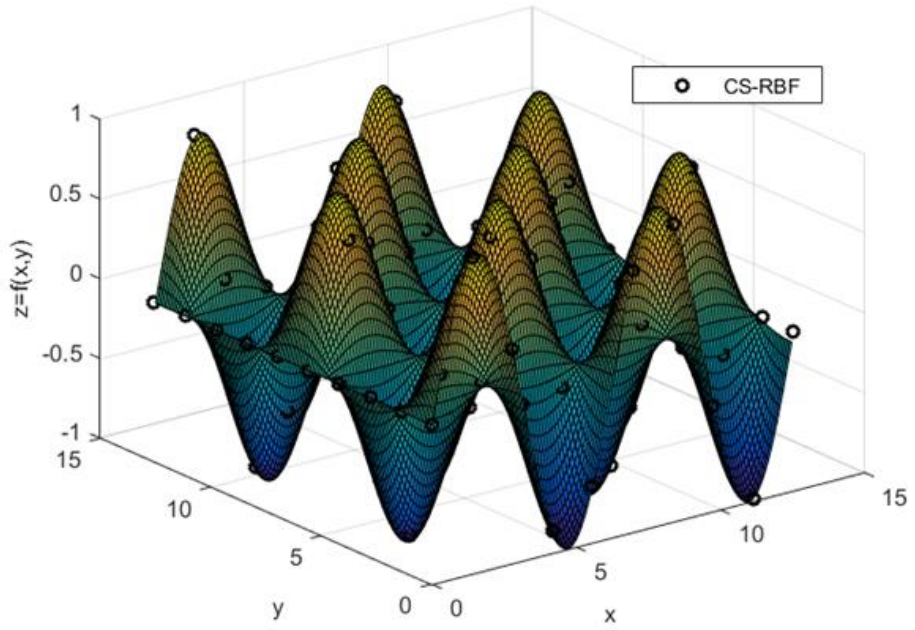


Figure 3. Solution profile comparison between that produced by CS1-RBF and that of the exact one.

4.2. Test 2 : Poisson Problem with Non-rectangular domain

In this second test case, the Poisson equation shown below is numerically solved by the collocation meshfree method.

$$\nabla^2 u = \left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \right) u = -x^2 \tag{26}$$

This is defined on the domain with an elliptical boundary expressed as $\left(x^2/4\right)+y^2=1$. Where the boundary condition is taken directly from the exact solution which is expressed as follows;

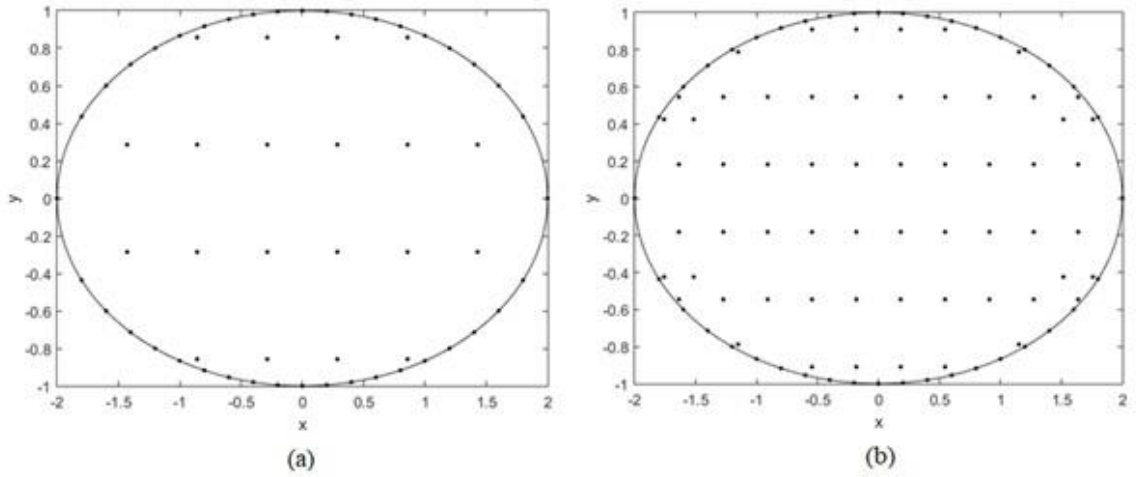
$$u(x,y)=-\frac{1}{246}\left(50x^2-8y^2+33.6\right)\left(\frac{x^4}{4}+y^2-1\right) \tag{27}$$

Based on the promising results obtained from the previous example, only CS1 is now presented here and it is marked as 'CS' from hereon.

For this example, L_∞ and L_{RMS} were utilized to measure the accuracy at different node density of 64, 144, 225, and 400. Table 3 clearly shows that the accuracy obtained from using MQ and GA is greatly influenced by the number of nodes involved in the system. This figure is more obvious when using 225 nodes with GA-RBF where a strong fluctuation of L_{RMS} occurs. This is not, however, the case when using CS where the solutions were found to remain almost intact with the change of node density. Figure 4. depicts the node distributions and the corresponding solution profile. It should also be remarked here that the optimal values of shape parameters shown in the table were obtained from a series of simulations taking place beforehand.

Table 3. Errors produced by each RBF when using different levels of nodes density.

Number of Computational Nodes	Error Norms	MQ ($\varepsilon=0.1$)	GA ($\varepsilon=0.1$)	CS
64	L_∞	8.6331e-04	9.4614e-04	0.0157
	L_{RMS}	3.8422e-04	3.5081e-04	0.0063
144	L_∞	5.7818e-04	0.0036	0.0077
	L_{RMS}	2.0858e-04	0.0014	0.0039
225	L_∞	0.0057	0.0028	0.0048
	L_{RMS}	0.0024	9.4496e-04	0.0027
400	L_∞	0.0034	0.0039	0.0032
	L_{RMS}	0.0012	0.0014	0.0015



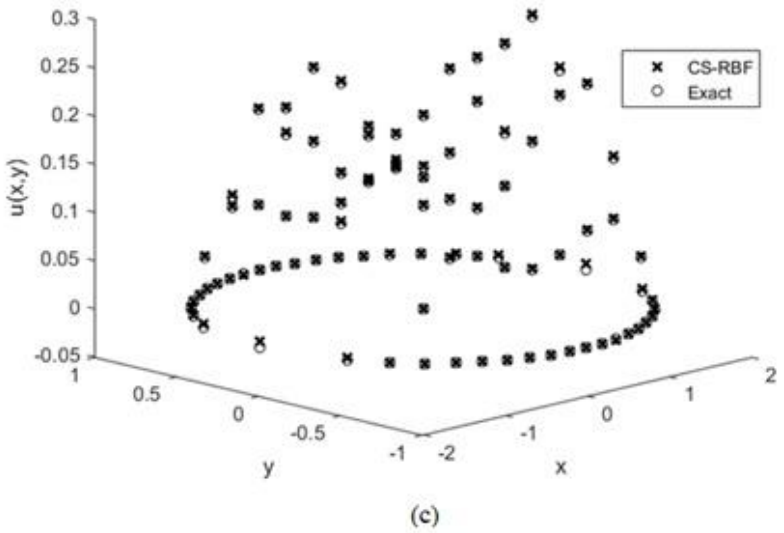


Figure 4. (a) and (b) node distribution at two levels of density, and (c) solution comparison between that of CS-RBF and the exact one.

4.3. Test 3 : Nonlinear Problem

The nonlinear PDE as given in GE. Fasshauer[5] on a square domain $0 < x < 1,\ 0 < y < 1$ is taken a look at. The governing equation is as follows;

$$-\varepsilon^2 \nabla^2 u - u + u^3 = f \tag{28}$$

with the boundary condition $u = 0$ on $\partial\Omega$ and the right hand side of the equation is chosen from the analytical solution of form;

$$u(x, y) = \psi(x)\psi(y) \tag{29}$$

with $\psi(t) = 1 + e^{-1/\varepsilon} - e^{-t/\varepsilon} - e^{(t-1)/\varepsilon}$,and (x, y) denotes the Cartesian coordinate of $\mathbf{x} \in \mathbb{R}^2$.

The computational domain for this example is shown in figure 5. where some extra nodes are required for the computing algorithm for nonlinear case as explained in Section 2. Table 4 provide strong evidence confirming that the solution quality produced by CS can be as good as those obtained from the two popular choices of RBFs. Solution profiles are plotted against the exact ones and shown in figure 6.

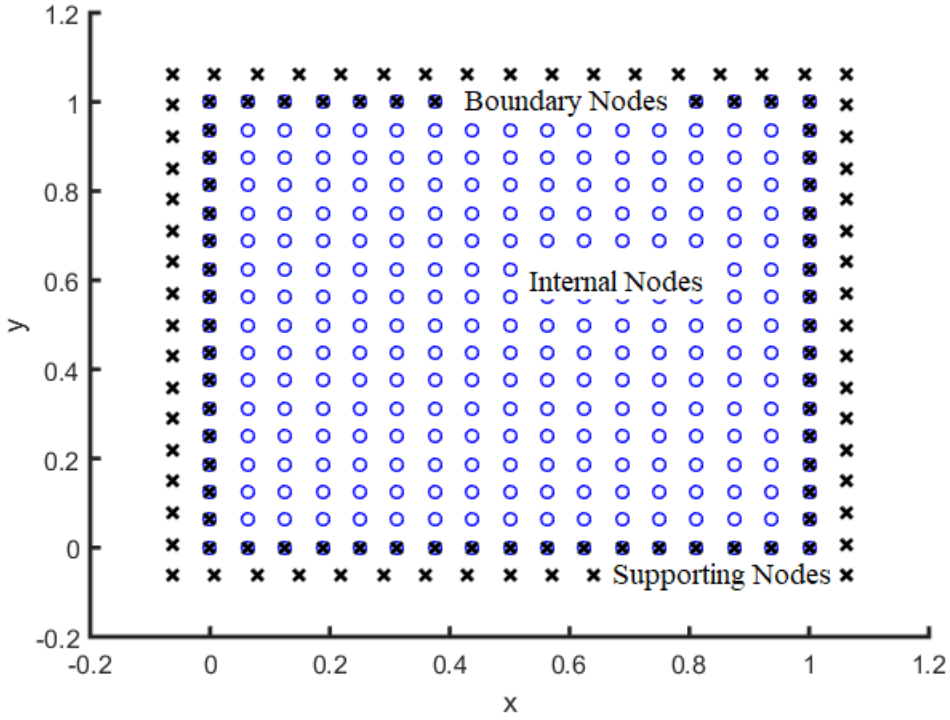


Figure 5. Node being uniformly-distributed over the computational domain, together with supported nodes needed for nonlinear computing process.

Table 4. Numerical solutions comparison with the exacts.

Points	MQ	GA	CS	Exact
(0.2,0.2)	0.750708	0.751436	0.753032	0.747144
(0.2,0.4)	0.848307	0.849172	0.849450	0.846440
(0.2,0.6)	0.848307	0.849172	0.849450	0.846440
(0.4,0.2)	0.848307	0.849172	0.849450	0.846440
(0.4,0.4)	0.959267	0.960383	0.959399	0.958933
(0.4,0.6)	0.959267	0.960383	0.959399	0.958933
(0.6,0.2)	0.848307	0.849172	0.849450	0.846440
(0.6,0.4)	0.959267	0.960383	0.959399	0.958933
(0.6,0.6)	0.959267	0.960383	0.959399	0.958933
(0.6,0.8)	0.848307	0.849172	0.849450	0.846440
(0.8,0.2)	0.750708	0.751436	0.753032	0.747144
(0.8,0.4)	0.848307	0.849172	0.849450	0.846440
(0.8,0.6)	0.750708	0.751436	0.753032	0.846440
(0.8,0.8)	0.750708	0.751436	0.753032	0.747144

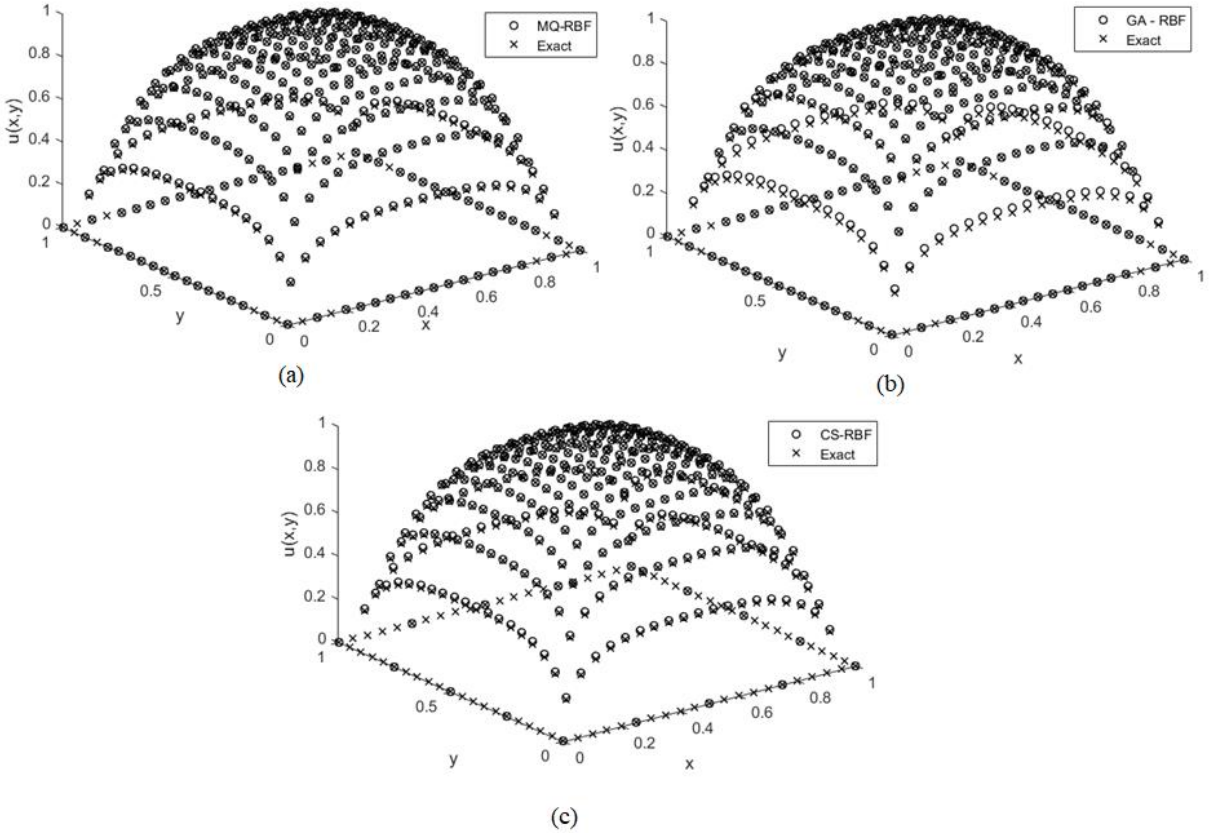


Figure 6. Solution comparisons; (a) using MQ-RBF, (b) using GA-RBF, and (c) using CS-RB

5. Conclusion

In this work, the main objective is to numerically study the effectiveness of one type of radial basis functions that has not been explored as much and it is called 'compacted-supported , CS- RBF'. This is done under the context of a numerical method that requires no mesh or grid so it is called 'meshfree/meshless' with collocation on computational nodes. The CS-RBF is applied to several benchmark test cases and its potential uses have been monitored. The main findings obtained from all the experiments are as follows;

1. When compared to the use of two popular RBFs; Multiquadric (MQ) and Gaussian (GA), the burden of finding an optimal shape parameter normally encountered in MQ and GA is completely omitted while the solutions produced are still in a good agreement.
2. The numerical solutions are found to me significantly affected by the distance between centres in the domain. This is not the case for all the shape-parameter-based RBFs such as MQ and GA.
3. The mathematical structure of CS-RBF is much less complex and much simpler when it comes to finding its first and second order of derivatives for solving PDEs.

This all indicates its promising future for further applications in more complex problems and it remains as our future investigation.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgement
The corresponding author would like to express his sincere appreciation to the Centre of Excellence in Mathematics for their kind support for this research project.

References

[1] Kansa E J 1990 A scattered data approximation scheme with applications to computational fluid-dynamics—II solutions to parabolic, hyperbolic and elliptic partial differential equations *Computers & Mathematics with Applications* **19 8** 147-61
[2] Sharan M, Kansa E J and Gupta S 1997 Application of the multiquadric method for numerical solution of elliptic partial differential equations *Applied Mathematics and Computation* **84** 275-302
[3] Chuathong N and Kaennakham S 2018 Numerical Solution to Coupled Burgers' Equations by Gaussian-Based Hermite Collocation Scheme *Journal of Applied Mathematics* **2018** 18
[4] Onate E, Idelsohn S,Zienkiewicz O C and Taylor R L 1996 A finite point method in computational mechanics. Applications to convective transport and fluid flow *International Journal for Numerical Methods in Engineering* **39 22** 3839-66
[5] Liszka T J, Duarte C A M and Tworzydlo W W 1996 hp-Meshless cloud method *Computer Methods in Applied Mechanics and Engineering* **139 1-4** 263-88
[6] Zhang X, Liu X H, Song K Z and Lu M W 2001 Least-squares collocation meshless method *International Journal for Numerical Methods in Engineering* **51 9** 1089-1100
[7] Nayroles B,Touzot G ang Villon P 1992 Generalizing the finite element method: Diffuse approximation and diffuse elements *Computational Mechanics* **10 5** 307-18
[8] Belytschko T, Gu L and Lu Y Y 1994 Fracture and crack growth by element free Galerkin methods *Modelling and Simulation in Materials Science and* Engineering **2** 519-34
[9] Liu G R and Gu Y T 2001 A point interpolation method for two-dimensional solids *International Journal for Numerical Methods in Engineering* **50 4** 937-51
[10] Kaennakham S and Chuathong N 2019 An Automatic Node-Adaptive Scheme applied with a RBF-Collocation Meshless Method *Applied Mathematics and Computation* **348(c)** 102-25
[11] Kaennakham S and Chuathong N 2017 Solution to a Convection-Diffusion Problem Using a New Variable Inverse-Multiquadric Parameter in a Collocation Meshfree Scheme *International Journal of Multiphysics* **11 4** 359-74
[12] Rippa S 1999 An algorithm for selecting a good value for the parameter c in radial basis function interpolation *Advances in Computational Mathematics* **11 2-3** 193-210
[13] Hardy R L 1971 Multiquadric equations of topography and other irregular surfaces *Journal of Geophysical Research* **76 8** 1896-1977
[14] Franke C and Schaback R 1998 Solving partial differential equations by collocation with radial basis functions *Applied Mathematics and Computation* **93 1** 73-82
[15] Buhmann M D 1998 Radial functions on compact support, *in Proceedings of the Edinburgh Mathematical society (Series 2)* **41 1** 33-46
[16] Fasshauer G E 2002 Newton iteration with multiquadrics for the solution of nonlinear PDEs *Computers & Mathematics with Applications* **43 3-5** 423-38