Numerical Simulation of Convection-Diffusion Phenomena by Four Inverse-Quadratic-RBF Domain-Meshfree Schemes

S Kaennakham[1,a], N Chuathong[2]*

1. School of Mathematics, Institute of Science, Suranaree University of Technology, Thailand

a. Centre of Excellence in Mathematics, Bangkok, Thailand

2. Faculty of Science Environment and Energy, King Mongkut's University of Technology, Thailand

ABSTRACT

The convection-diffusion type of PDEs is numerically solved by four numerical methods in this work. These four comparatively young numerical approaches are categorized as 'domain-meshfree' as they require no internal meshing but rely only on the collocation process amongst nodes via. the inverse-quadratic radial basis function (IQ-RBF). They are; the well-known Kansa Collocation Method (KCM), the Hermite Collocation Method (HCM), the Radial Point Interpolation Method (RPIM), and the Dual Reciprocity Boundary Element Method (DRBEM). The work aims to demonstrate the use of IQ-RBF as well as to compare the practical use of the methods. Moreover, engineering senses of criteria judging the quality of the methods are considered. It is found in this work that while KCM is the simplest to construct and deploys, it is highly sensitive to the number of nodes and the IQ-RBF shape parameter. The asymmetric and populated matrix problem are alleviated when HCM or DRBEM are in use yet more CPU-time and storage seem to be the price to pay, particularly HCM. However, it is actually RPIM that has appeared to be an optimal choice under all the criteria imposed.

1. INTRODUCTION

Convection diffusion problems are known to be governed PDE mathematical models and they are found to appear in many branches of sciences and engineering such as biological, physical chemical, physical in fluid mechanics, astrophysics, meteorology, and multiphase flow in oil reservoirs, polymer flow and many other areas [1]. The unsteady-state form of the equation can be represented by;

$$\dot{\omega} + \upsilon\nabla\omega - \eta\nabla^2\omega \pm S = 0 \tag{1}$$

In the above definition, $\dot{\omega}$ represents the transient term,∇ is the gradient operator, ∇^2 is the Laplacian operator, and S is the sum of additional source and/or sink terms. Very often the dimensionless parameters, υ and η , that measure the relative strength of the diffusion to

*Corresponding Author: nissaya.c@sciee.kmutnb.ac.th

the convection is quite small leading to situations where thin boundary and interior layers are presented and singular perturbation problems arises [2].

In the study of this type of problem, the numerical solution in many cases has been tackled using traditional method such as the finite difference method (FDM), the finite element method (FEM), or the finite volume method (FVM), see the book by Zienkiewicz and Taylor [3] and references herein. In these conventional schemes, before the computing process can be performed, a mesh generation or meshing process over both the domain Ω and boundary $\partial\Omega$ is required to take place. This inevitably makes the methods difficult and time-consuming particularly when solving complicated geometries. To improve the situation, affords may be put to automatically allocating the mesh-grid in hope to obtain optimal solutions and this is known as 'grid-adaptation method' [4-5].

Due to this undesirable aspect of mesh-generated numerical methods, together with a few more (please see [6]), some alternatives have been proposed over the past decade and one of which is those based on the use of a multivariate function called 'Radial Basis Function (RBF)'. The methods under this category is known as 'Meshfree/Meshless Method' [7].

A radial basis function (RBF) can be defined as a function of the distance of the point to the origin. That is, ϕ is a RBF if $\phi(\boldsymbol{x}) = \phi(\|\boldsymbol{x}\|)$, for $\boldsymbol{x}$ is a vector in $\mathbb{R}^n$ and $\|\cdot\|$ is a chosen norm. In this work, the main attention is paid to a specific type of inverse multiquadric and it is called 'Inverse-Quadratic (IQ)', defined as follows;

$$\phi(r) = \frac{1}{(1+(\varepsilon \boldsymbol{r})^2)} = \frac{1}{(1+(\varepsilon\|\boldsymbol{x}\|_2)^2)} \tag{2}$$

With $\|\cdot\|_2$ being the Euclidean norm. The appearance of the parameter $\varepsilon > 0$ is known to have a great effect on the solution of the problem at hand. FIG. 1 demonstrates the IQ-RBF profiles influenced by different nonnegative shape ε. Under the theory of interpolation, this RBF has been proved to be 'Globally Supported, Strictly Positive Definite Functions' and this function is C^{∞} at the origin [6]. While other kinds of RBF mentioned above have been receiving a huge interest from researchers [8-10], it is interesting to see this is not the case for IQ and, therefore, it deserves more investigation.

Numerical methods which require no mesh generation, at least in the domain, have recently been more popular. With the use of RBF, one of the pioneers of this idea is the work nicely done my Kansa in 1990 [11]. In his work, a multiquadric type of RBF was introduced and applied to the interpolation problem and was extended to solving PDEs and the method is named "Kansa Method' ever since; some applications include groundwater contaminant transport [12], convection–diffusion problems [13], plate and shell analysis [14], microelectromechanical system analysis [15] and many more. Li [19] concluded that the accuracy of the RBF method was more superior than FEM.

The traditional version of Kansa Collocation Method (KCM), nevertheless, is known to severely suffer from having a fully asymmetric and populated collocation matrix, increasing the risk of being ill-conditioned. Attempts to avoid these drawbacks have been proposed and developed, see the book by Hua and Shantanu [17].

One of the studies aiming to alleviate the problems encountered in the use of KCM is that proposed by Fasshauer [18] in 1997. Fasshauer [18] described the scattered Hermite interpolation for the solution of elliptic partial differential equations by using collocation. An example of comparison studies is that done by Kazerm et al. [19]. Another attempt is that

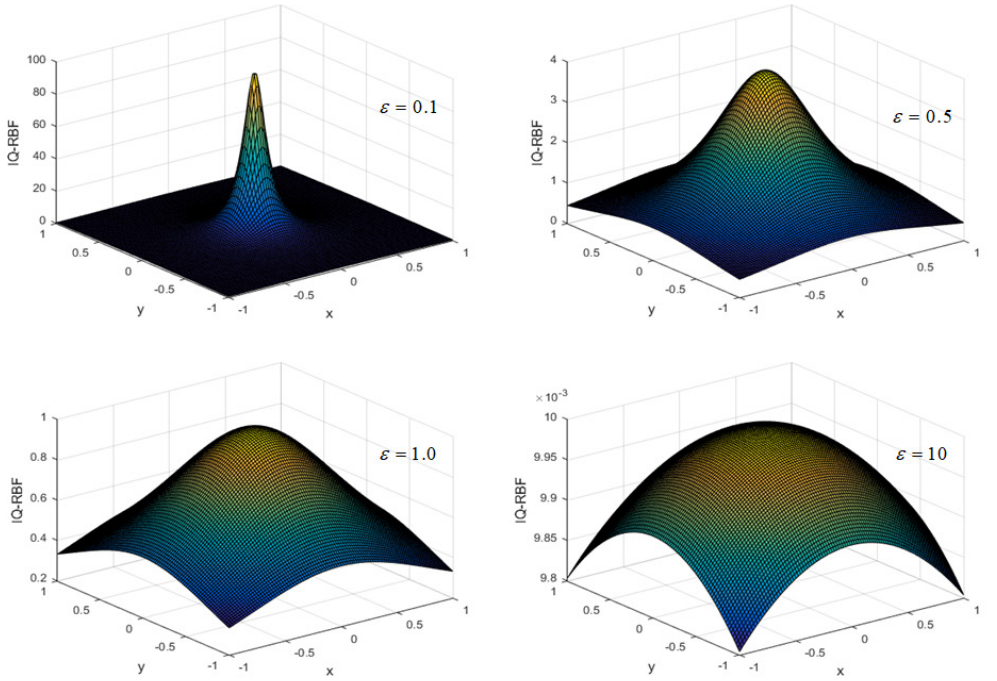

FIG. 1. Inverse-Quadratic (IQ) –RBF surface plot at different shape parameters ε.

proposed in 2002 by Wang and Liu [20] and it has been known as 'The Radial Point Interpolation Method (RPIM)'. One of the nice numerical experiments on applying this method is that carried out by Liu [21] in 2011, where it was concluded that the singularity problem can be improved (see also Bozkurt et. al. [22]). For some more recently and nicely documented, the interested reader is referred to [23] and [24].

Another comparatively new method that does not require any domain meshing is called 'Boundary Element Method (BEM)' [25]. Nevertheless, an undesirable aspect is the need to discretize the domain into a series of internal cells to deal with the terms taken to the boundary by application of the fundamental solution, inevitably destroying some attraction of the method. To remedy this drawback, one of the attempts is the so-called 'Dual Reciprocity Boundary Element Method (DRBEM)', by Nardini and Brebbia [26]. The idea has extensively been developed by many researchers such as the perturbation DRBEM [27], the separation of variables DRBEM [28], and the Laplace transform DRBEM [29].

Over the decade, even though some numerical comparison studies are available, see [30] for instance, the only criterion adopted appears to be the accuracy. This is rather impractical when there are other factors deserve to be taken into consideration as criteria as well.

In this work, four comparatively young domain-meshless numerical methods are studied and they are; the traditional or conventional Kansa Collocation Method (KCM), the Hermite Collocation Method (HCM), the Radial Point Interpolation Method (RPIM), and the Dual Reciprocity Boundary Element Method (DRBEM). The main objective is to shed more light into their overall practical aspects that should cover at least four criteria, listed below;

1. The overall accuracy and the error growth in time.
2. The CUP-time requirement in computing process.
3. The sensitivity to factors involved such as the density of nodes and shape values.
4. The simplicity to setup and deploy.

The problem chosen for this task is one of the classical convection-diffusion PDE in two dimensions.

2. MATHEMATICAL BACKGROUND

2.1. The Governing Equation of Convection-Diffusion PDE

This work aims to numerically solve two-dimensional convection-diffusion problems that are modelled and governed by the following partial differential equation;

$$\frac{\partial u}{\partial t} + V_x \frac{\partial u}{\partial x} + V_y \frac{\partial u}{\partial y} = \eta_x \frac{\partial^2 u}{\partial x^2} + \eta_y \frac{\partial^2 u}{\partial y^2} - \beta u + g(x, y, t) \tag{3}$$

Where V_x, V_y are convection coefficients, and η_x, η_y are diffusion coefficients. The last two terms βu and the source term $g(x, y, t)$ are additional and needed only in specific cases.

For the test cases studied in this work, it is set that $\eta_x = \eta_y = \eta$ the it leads to the following expression;

$$\frac{\partial u}{\partial t} + V_x \frac{\partial u}{\partial x} + V_y \frac{\partial u}{\partial y} + \beta u - g(x) = \eta \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) \tag{4}$$

Leading to;

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = \frac{1}{\eta} \left(\frac{\partial u}{\partial t} + \left(V_x \frac{\partial u}{\partial x} + V_y \frac{\partial u}{\partial y} \right) + \beta u - g(x) \right) \tag{5}$$

Subject to the initial condition $u(x, y, 0) = \psi_1(x, y)$ and the boundary condition $u(x, y, t) = \psi_2(x, y, t)$ with $t > 0$ and Ω is a domain of the problem, $\partial\Omega$ is its boundary, ψ_1, ψ_2 are known functions.

2.2. The Kansa Collocation Method (KCM)

The collocation scheme starts with considering the following elliptical partial differential equation defined on a bounded and connected domain Ω;

$$\Phi[u(\mathbf{x})] = f(x) \qquad \text{for } \mathbf{x} \in \Omega \subset \mathbb{R}^n \tag{6}$$

$$B_1 u(\mathbf{x}) = g(x) \qquad \text{for } \mathbf{x} \in \Gamma_1 \tag{7}$$

$$B_2 u(\mathbf{x}) = h(x) \qquad \text{for } \mathbf{x} \in \Gamma_2 \tag{8}$$

Where $\partial\Omega$ is the domain boundary containing two non-overlap sections; Γ_1 and Γ_2 , with $\Gamma_1 \cap \Gamma_2 = \phi$. These differential operators; Φ, and B_1, B_2 are applied on the domain, and the two boundary sections respectively. Three known functions $f(\mathbf{x}), g(\mathbf{x}), h(\mathbf{x})$ can be dependent of space and/or time. Let $\mathbf{X}^c = \{\mathbf{x}_j\}_{j=1}^{N}$ be a set of randomly selected points, known as ' collocation' or 'centers', on the domain where $\{\mathbf{x}_j\}_{j=1}^{N_i}$ are those contained within, where $\{\mathbf{x}_j\}_{j=N_i+1}^{N_i+N_1}$ and $\{\mathbf{x}_j\}_{j=N_i+N_1+1}^{N}$ are those on the boundary Γ_1 and $\Gamma_2$respectively.

The collocation scheme writes the approximate solution, $\tilde{u}(\mathbf{x})$, as a linear combination of the basis function $\{\varphi(:-)\}_j^N$, shown in the following form;

$$u(\mathbf{x}) \simeq \tilde{u}(\mathbf{x}) = \sum_{j=1}^{N} \alpha_j \varphi\left(\left\|\mathbf{x} - \mathbf{x}_j\right\|_2\right) \tag{9}$$

Where α_j are coefficients and $\|.\|_2$ being the Euclidean norm. The basis function used now is the inverse-quadratic radial type as defined previously.

Applying the operators Φ, and B_1, B_2 to on both domain and boundary sections, satisfying the governing system of equations, allows the system to arrive at;

$$\mathbf{A}\boldsymbol{\alpha} = \mathbf{b} \tag{10}$$

Where $\boldsymbol{\alpha} = [\alpha_1 \quad \alpha_2 \quad \dots \quad \alpha_N]$, the known $\mathbf{b}$ vector is as follows;

$$\mathbf{b} = \left[f(\mathbf{x}_1) \quad f(\mathbf{x}_2) \quad \dots \quad f(\mathbf{x}_{N_i}) \quad g(\mathbf{x}_{N_i+1}) \quad \dots \quad g(\mathbf{x}_{N_i+N_1}) \quad h(\mathbf{x}_{N_i+N_1+1}) \quad \dots \quad h(\mathbf{x}_N) \right]^T$$

And by setting $\boldsymbol{\varphi}$ to be a matrix with entries $\varphi_{ij} = \varphi\left(\left\|\mathbf{x}_i - \mathbf{x}_j\right\|_2\right)$. Hence, for $i, j = 1,2,\dots,N$ we have;

$$\mathbf{A} = \begin{bmatrix} \Phi[\boldsymbol{\varphi}] \\ B_1[\boldsymbol{\varphi}] \\ B_2[\boldsymbol{\varphi}] \end{bmatrix} \tag{11}$$

Once α_j are obtained, the approximate solution are straightforward yielded. This method is known as 'Kansa', in honor of a great mathematician Prof. Edward Kansa who discovered the idea in 1990. The method has been applied to a wide range of problem ever since [31]. It, nevertheless, is not of no shortcoming where it is known to suffer the problem of asymmetric interpolation matrix, **A**, and the rigorous mathematical proof of its solvability is still not available [32]. It is also very often produces low quality results particularly in boundary-adjacent region [33].

The Laplacian form, LHS of equation (5), can now be expressed as follows;

$$\left(\frac{\partial^2}{\partial x^2}+\frac{\partial^2}{\partial y^2}\right)\varphi(r)=\frac{d^2}{dr^2}\varphi(r)+\frac{1}{r}\frac{d}{dr}\varphi(r) \tag{12}$$

Where the other terms in the equation can also be replaced by derivatives expressed above accordingly.

2.3. The Hermite Collocation Method (HCM)

In 1997, Fasshauer [21] proposed a way of interpolation by applying the self-adjoint operators Φ^* , and B_1^*, B_2^* to the governing system of equations and rewrite the approximate solution as;

$$u(\boldsymbol{x})\simeq\tilde{u}(\boldsymbol{x})=\sum_{j=1}^{N_i}\alpha_j\Phi^*\varphi\left(\left\|\boldsymbol{x}-\boldsymbol{x}_j\right\|_2\right)$$
$$+\sum_{j=N_i+1}^{N_i+N_1}\alpha_j B_1^*\varphi\left(\left\|\boldsymbol{x}-\boldsymbol{x}_j\right\|_2\right)+\sum_{j=N_i+N_1+1}^{N}\alpha_j B_2^*\varphi\left(\left\|\boldsymbol{x}-\boldsymbol{x}_j\right\|_2\right) \tag{13}$$

This leads to a new interpolation matrix $\boldsymbol{A}$, shown as follows;

$$\boldsymbol{A}=\begin{bmatrix}\Phi\Phi^*[\boldsymbol{\varphi}] & \Phi B_1^*[\boldsymbol{\varphi}] & \Phi B_2^*[\boldsymbol{\varphi}]\\ B_1\Phi^*[\boldsymbol{\varphi}] & B_1B_1^*[\boldsymbol{\varphi}] & B_1B_2^*[\boldsymbol{\varphi}]\\ B_2\Phi^*[\boldsymbol{\varphi}] & B_2B_1^*[\boldsymbol{\varphi}] & B_2B_2^*[\boldsymbol{\varphi}]\end{bmatrix} \tag{14}$$

An application of this Hermite type of collocation method to elastostatic problem was done by Leitao [34]. Some interesting implementations of the scheme to the transient and nonlinear plate problems can be found in [35,36] and [37].

In order to implement the Hermite concept, it is necessary that the Laplacian be applied twice resulting in the following so-called fourth-order biharmonic form;

$$\left(\frac{\partial^4}{\partial x^4}+2\frac{\partial^4}{\partial x^2\partial y^2}+\frac{\partial^4}{\partial y^4}\right)\varphi(r)=\frac{d^4}{dr^4}\varphi(r)+\frac{2}{r}\frac{d^3}{dr^3}\varphi(r)-\frac{1}{r^2}\frac{d^2}{dr^2}\varphi(r)+\frac{1}{r^3}\frac{d}{dr}\varphi(r) \tag{15}$$

Hence, the inverse-quadratic type of RBF used in this work and its first four orders of derivatives can be respectively expressed as follows;

$$\varphi\left(\left\|\mathbf{x}-\mathbf{x}_j\right\|_2\right)=\frac{1}{(1+(\varepsilon r)^2)}=\frac{1}{\left(1+\left(\varepsilon\left\|\mathbf{x}-\mathbf{x}_j\right\|_2\right)^2\right)} \tag{16}$$

Thus,

$$\frac{d}{dr}\varphi(r)=-\frac{2\varepsilon^2 r}{(1+(\varepsilon r)^2)^2} \tag{17}$$

$$\frac{d^2}{dr^2}\varphi(r)=2\varepsilon^2\frac{3(\varepsilon r)^2-1}{(1+(\varepsilon r)^2)^3} \tag{18}$$

$$\boldsymbol{\frac{d^3}{dr^3}\varphi(r)=\frac{24\varepsilon^4 r\left[1-(\varepsilon r)^2\right]}{\left(1+(\varepsilon r)^2\right)^4}} \tag{19}$$

$$\frac{d^4}{dr^4}\varphi(r) = \frac{24\varepsilon^4[1-5(\varepsilon r)^2]}{(1+(\varepsilon r)^2)^5} \tag{20}$$

Where $r_{ij} = \|\boldsymbol{x} - \boldsymbol{x}_j\|_2$, for each pair of center nodes; $\boldsymbol{x}_i, \boldsymbol{x}_j$.

2.4. The Radial Point Interpolation Method (RPIM)

The method was proposed by Wang and Liu [23], and it writes the approximate solution for a given PDE, $\tilde{u}(\boldsymbol{x})$, as the linear combination of the basis function and monomials $p_j(\boldsymbol{x})$, shown in the following form;

$$u(\boldsymbol{x}) \simeq \tilde{u}(\boldsymbol{x}) = \sum_{i=1}^{N} R(\|\boldsymbol{x} - \boldsymbol{x}_i\|_2)a_i + \sum_{j=1}^{m} p_j(\boldsymbol{x})b_j = \boldsymbol{R}^T(\boldsymbol{x})\boldsymbol{a} + \boldsymbol{P}^T(\boldsymbol{x})\boldsymbol{b} \tag{21}$$

With m representing the number of polynomial basis (usually, $m < N$). The polynomial function can be chosen from Pascal's triangle which, for 2D problems, as;

$$\boldsymbol{P^T}(\boldsymbol{x}) = [1, x, y, x^2, xy, y^2, \dots] \tag{22}$$

To ensure the unique solution of the system, additional m equations can be added as the constraint conditions, as follows;

$$\sum_{i=1}^{N} p_j(\mathbf{x}_i)a_i = \mathbf{P}_m^T\mathbf{a} = 0 \tag{23}$$

for $j = 1,2,3,\dots,m$. This leads to the following form of matrix equations describing the interpolation process for all centers all over the domain.

$$\tilde{\mathbf{U}}(\mathbf{x}) = \begin{bmatrix}\mathbf{U}(\mathbf{x}) \\ \mathbf{0}\end{bmatrix} = \begin{bmatrix}\mathbf{R}_0 & \mathbf{P}_m \\ \mathbf{P}_m^T & \mathbf{0}\end{bmatrix}\begin{bmatrix}\mathbf{a} \\ \mathbf{b}\end{bmatrix} = \mathbf{H}\boldsymbol{\alpha}_0 \tag{24}$$

Leading to

$$\boldsymbol{\alpha}_0 = \boldsymbol{H}^{-1}\tilde{\boldsymbol{U}}(\boldsymbol{x}) \tag{25}$$

Where

$$\boldsymbol{H} = \begin{bmatrix}\boldsymbol{R}_0 & \boldsymbol{P}_m \\ \boldsymbol{P}_m^T & \mathbf{0}\end{bmatrix}$$

$$\boldsymbol{\alpha}_0 = \begin{bmatrix}\mathbf{a} \\ \mathbf{b}\end{bmatrix} = \begin{bmatrix}a_1 & a_2 & \dots & a_n & b_1 & b_2 & \dots & b_m\end{bmatrix}^T$$

$$\mathbf{R}_0 = \begin{bmatrix} R_1(x_1,y_1) & R_2(x_1,y_1) & \dots & R_N(x_1,y_1) \\ R_1(x_2,y_2) & R_2(x_2,y_2) & \dots & R_N(x_2,y_2) \\ \vdots & \vdots & \ddots & \vdots \\ R_1(x_N,y_N) & R_2(x_N,y_N) & \dots & R_N(x_N,y_N) \end{bmatrix}_{N\times N}$$

and

$$\mathbf{P}_m^T = \begin{bmatrix} 1 & 1 & \dots & 1 \\ x_1 & x_2 & \dots & x_N \\ y_1 & y_2 & \dots & y_N \\ \vdots & \vdots & \ddots & \vdots \\ p_m(\mathrm{x}_1) & p_m(\mathrm{x}_2) & \dots & p_m(\mathrm{x}_n) \end{bmatrix}_{m\times N}$$

Substituting these back into the collocation equation, yielding;

$$u(\boldsymbol{x}) = [\boldsymbol{R}^T(\boldsymbol{x}) \quad \boldsymbol{P}^T(\boldsymbol{x})]\boldsymbol{H}^{-1}\widetilde{\boldsymbol{U}}(\boldsymbol{x}) \tag{26}$$

By setting the shape functions matrix, $\hat{\boldsymbol{\theta}}^T(\boldsymbol{x})$, as;

$$\hat{\boldsymbol{\theta}}^T(\boldsymbol{x}) = [\boldsymbol{R}^T(\boldsymbol{x}) \quad \boldsymbol{P}^T(\boldsymbol{x})]\boldsymbol{H}^{-1} \tag{27}$$

Then the previous equation can be re-written as;

$$u(\boldsymbol{x}) = \hat{\boldsymbol{\theta}}^T(\boldsymbol{x})\widetilde{\boldsymbol{U}}(\boldsymbol{x}) \tag{28}$$

Where the approximate solutions at each center can now be obtained.

2.5 The Dual Reciprocity Boundary Element Method (DRBEM)

The mathematical construction of the dual reciprocity boundary element method can start with the Poisson equation as follows;

$$\nabla^2 u = b(x,y) \tag{29}$$

Which as its equivalent integral form, given by [29], as;

$$c_i u_i + \int_\Gamma q^* u d\Gamma - \int_\Gamma u^* q d\Gamma = \sum_{j=1}^{N+L} \alpha_j \left(c_i \hat{u}_{ij} + \int_\Gamma q^* \hat{u}_j d\Gamma - \int_\Gamma u^* \hat{q}_j d\Gamma \right) \tag{30}$$

Where u^* is the fundamental solution and the term $\hat{q}_j$ is defined as $\hat{q}_j = \frac{\partial \hat{u}_j}{\partial n}$, where $\mathbf{n}$ is the unit outward normal to Γ, and can be written as;

$$\hat{q}_j = \frac{\partial \hat{u}_j}{\partial x}\frac{\partial x}{\partial n} + \frac{\partial \hat{u}_j}{\partial y}\frac{\partial y}{\partial n} \tag{31}$$

With N and L being the number of boundary and internal nodes respectively, b can be now approximated by;

$$b_i(x,y) \approx \sum_{j=1}^{N+L} \alpha_j \varphi_{ij}(x,y) \tag{32}$$

Here, the function φ is the radial basis function, then;

$$\nabla^2 \hat{u}_j = f_j \tag{33}$$

For some particular solution $\hat{u}_j$. By applying Green's theorem, the boundary element approximation to then it becomes, at a node i[th];

$$c_i u_i + \sum_{k=1}^{N} H_{ik} u_k - \sum_{k=1}^{N} G_{ik} q_k = \sum_{j=1}^{N+L} \alpha_j \left(c_i \hat{u}_{ij} + \sum_{k=1}^{N} H_{ik} \hat{u}_{kj} - \sum_{k=1}^{N} G_{ik} \hat{q}_{kj}\right) \tag{34}$$

Where the definition of the terms H_{ik} and G_{ik} can be found in [28]. The index k is used for the boundary nodes which are the field points. Using a collocation technique, the above equation can be compactly expressed in matrix form as follows;

$$\boldsymbol{Hu - Gq} = \left(\boldsymbol{H\hat{U} - G\hat{Q}}\right)\alpha \tag{35}$$

By substituting $\alpha = \boldsymbol{F^{-1}b}$ into equation (35) making the right-hand side) a known vector. Therefore, it can be rewritten as;

$$\boldsymbol{Hu - Gq = d} \tag{36}$$

Where $\boldsymbol{d} = \left(\boldsymbol{H\hat{U} - G\hat{Q}}\right)\boldsymbol{F^{-1}b}$.

Applying boundary condition(s) to equation (36) then it can be seen as the simple form as follows;

$$\boldsymbol{AX = Y} \tag{37}$$

Where $\boldsymbol{X}$ contains N unknown boundary values of u's and q's and the resulting linear system can be solved by Gaussian elimination scheme. Therefore, the solution is obtained via;

$$u_i = -\sum_{k=1}^{N} H_{ik}u_k + \sum_{k=1}^{N} G_{ik}q_k = \sum_{j=1}^{N+L} \alpha_j\left(c_i\hat{u}_{ij} + \sum_{k=1}^{N} H_{ik}\hat{u}_{kj} - \sum_{k=1}^{N} G_{ik}\hat{q}_{kj}\right) \tag{38}$$

When dealing with convection-diffusion forms of PDEs, it can be done by setting;

$$V_x\frac{\partial u}{\partial x} = \boldsymbol{V_x}\frac{\partial F}{\partial x}\boldsymbol{F^{-1}u} \tag{39}$$

And

$$V_y\frac{\partial u}{\partial y} = \mathbf{V_y}\frac{\partial F}{\partial y}\mathbf{F^{-1}u} \tag{40}$$

Also, we set the RHS of equation (5) as;

$$b = \frac{1}{\varepsilon}\left(\frac{\partial u}{\partial t} + \left(V_x\frac{\partial u}{\partial x} + V_y\frac{\partial u}{\partial y}\right) + \beta u - g(x)\right) \tag{41}$$

Therefore, from equation (5), equation (36), and equation (41), the governing equation can be rewritten in the following form;

$$\boldsymbol{Hu - Gq = S\left(\frac{1}{\varepsilon}\left(\dot{u} + \left(V_x\frac{\partial F}{\partial x}F^{-1}u + V_y\frac{\partial F}{\partial y}F^{-1}u\right) + \beta u - g(x)\right)\right)} \tag{42}$$

Where $\dot{u}$ is the transient term which will be discretized using RK4 as detailed in the following section.

3. COMPUTATION SETUPS

All numerical solutions obtained from the whole experiment are validated mainly by comparing to the exact or analytical solutions. For this, the following error norms are used;

1. Maximum Error (L_∞) ;

$$L_\infty = \max_{1\le i\le N} |\tilde{u}(x_i) - u(x_i)| \tag{43}$$

2. Root-Mean-Square Error (L_{rms});

$$L_{rms} = \left(\frac{1}{N}\sum_{j=1}^{N}\left(\tilde{u}(x_j) - u(x_j)\right)^2\right)^{1/2} \tag{44}$$

3. Percentage Relative Error ($Rel.Err.$)

$$Rel.Err.(\%) = \frac{|u_i^{ext} - u_i^{appx}|}{|u_i^{ext}|} \times 100 \tag{45}$$

4. Absolute Error ($Abs.Err.$)

$$\text{Abs. Err.} = \left|u_i^{ext} - u_i^{appx}\right| \tag{46}$$

The transient term of the governing equation is tackled numerically by using the well-known Runge-Kutta (RK4) method for all schemes under investigation. The process begins by setting;

$$\frac{\mathrm{du}}{\mathrm{dt}} = \mathrm{F}(\mathbf{u}) \tag{47}$$

For $\boldsymbol{u}(\boldsymbol{x}, t_n) = \boldsymbol{u}^n, \boldsymbol{x} \in \mathbb{R}^d$. Then let the following;

$$\begin{aligned} k_1 &= \Delta t F(\mathbf{u}^n, t^n), \\ k_2 &= \Delta t F\left(\mathbf{u}^n + \frac{k_1}{2}, t^n + \frac{\Delta t}{2}\right), \\ k_3 &= \Delta t F\left(\mathbf{u}^n + \frac{k_2}{2}, t^n + \frac{\Delta t}{2}\right), \\ k_4 &= \Delta t F(\mathbf{u}^n + k_3, t^n + \Delta t). \end{aligned} \tag{48}$$

Then, the time-increased $\boldsymbol{u}^{n+1}$ is obtained by;

$$\boldsymbol{u}^{n+1} = \boldsymbol{u}^n + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) \tag{49}$$

4. NUMERICAL EXPERIMENTS AND GENERAL DISCUSSION

Example 4.1: Interpolation with Inverse-Quadratic RBF

To study the impact of the shape parameter,ε,can be done by investigating an interpolation problem of a benchmark and well-known Franke-type function [38] defined on a unit square-domain, as;

$$\begin{aligned} f(x,y) = 0.75\,exp\left[-\frac{(9x-2)^2}{4} - \frac{(9y-2)^2}{4}\right] + 0.75\,exp\left[-\frac{(9x+1)^2}{49} - \frac{(9y+1)}{10}\right] + \\ 0.5\,exp\left[-\frac{(9x-7)^2}{4} - \frac{(9y-3)^2}{4}\right] - 0.2\,exp[-(9x-4)^2 - (9y-7)^2] \end{aligned} \tag{50}$$

With a set of N computational nodes uniformly-distributed over the unit square (i.e. the summation of internal and boundary nodes). The interpolation function, $\tilde{f}(\boldsymbol{x}_i)$ at the i-th center node $\boldsymbol{x}_i$, is defined as a linear combination of the inverse quadratic radial basis function as;

$$\tilde{f}(\boldsymbol{x}_i) = \sum_{j=1}^{N} \alpha_j \varphi\left(\left\|\boldsymbol{x}_i - \boldsymbol{x}_j\right\|_2\right) = \sum_{j=1}^{N} \alpha_j \left(\frac{1}{\left(1+\left(\varepsilon\left\|\boldsymbol{x}_i-\boldsymbol{x}_j\right\|_2\right)^2\right)}\right) \tag{51}$$

With $\tilde{f}(\boldsymbol{x}_i) = f(\boldsymbol{x}_i)$ for all $i = 1,2,\ldots,N$. By imposing this function on all nodes, it leads to the linear system expressed as;

$$\boldsymbol{A\alpha} = \boldsymbol{F} \tag{52}$$

Where

$$\mathbf{A} = \begin{bmatrix} \varphi\left(\left\|\mathbf{x}_1 - \mathbf{x}_1\right\|_2\right) & \varphi\left(\left\|\mathbf{x}_1 - \mathbf{x}_2\right\|_2\right) & \ldots & \varphi\left(\left\|\mathbf{x}_1 - \mathbf{x}_N\right\|_2\right) \\ \varphi\left(\left\|\mathbf{x}_2 - \mathbf{x}_1\right\|_2\right) & \varphi\left(\left\|\mathbf{x}_2 - \mathbf{x}_2\right\|_2\right) & \ldots & \varphi\left(\left\|\mathbf{x}_2 - \mathbf{x}_N\right\|_2\right) \\ \vdots & \vdots & \ddots & \vdots \\ \varphi\left(\left\|\mathbf{x}_N - \mathbf{x}_1\right\|_2\right) & \varphi\left(\left\|\mathbf{x}_N - \mathbf{x}_2\right\|_2\right) & \ldots & \varphi\left(\left\|\mathbf{x}_N - \mathbf{x}_N\right\|_2\right) \end{bmatrix}$$

Functions' and the coefficient matrix; $\boldsymbol{\alpha} = [\alpha_1 \quad \alpha_2 \quad \ldots \quad \alpha_N]^T$, with the known vector function $\mathbf{F} = \begin{bmatrix} \tilde{f}\left(\mathbf{x}_1\right) & \tilde{f}\left(\mathbf{x}_2\right) & \ldots & \tilde{f}\left(\mathbf{x}_N\right) \end{bmatrix}^T$. By the well-known Gauss-Seidel method, the coefficient matrix $\boldsymbol{\alpha}$ is easily obtained and it will then be used to interpolate the value of $f(x, y)$ at a new set of interpolation nodes via. the same linear summation as defined above.

With starting the interpolation process by utilizing $N_{\text{ctr}} = 10 \times 10$ centers, FIG. 2. shows the RMS error produced at a wide range of shape parameter when using $N_{int} = 7 \times 7$ interpolation nodes. It is observed that the error is clearly varies with the shape and the best interpolation solution is seen to take place when $\varepsilon \in (2.0,5.0)$. At $\varepsilon \approx 3.5$ in particular, the results obtained are in good agreement with the exacts as illustrated in FIG. 3. The comparison of absolute error ($Abs.Err.$) produced when using two values of shape, $\varepsilon = 3.5$ and $\varepsilon = 10.0$, is provided in Tab. 1. It can be seen from this Table. that the shape has great effect on the final numerical solution, both locally and globally, of the domain. Nodes (0.333,0.000), (1.000,0.333), and (1.000,0.666) are some of the locations where the highest growths in $Abs.Err.$ are found; from $\approx 9.00E-06$ up to $\approx 1.00E-01$.

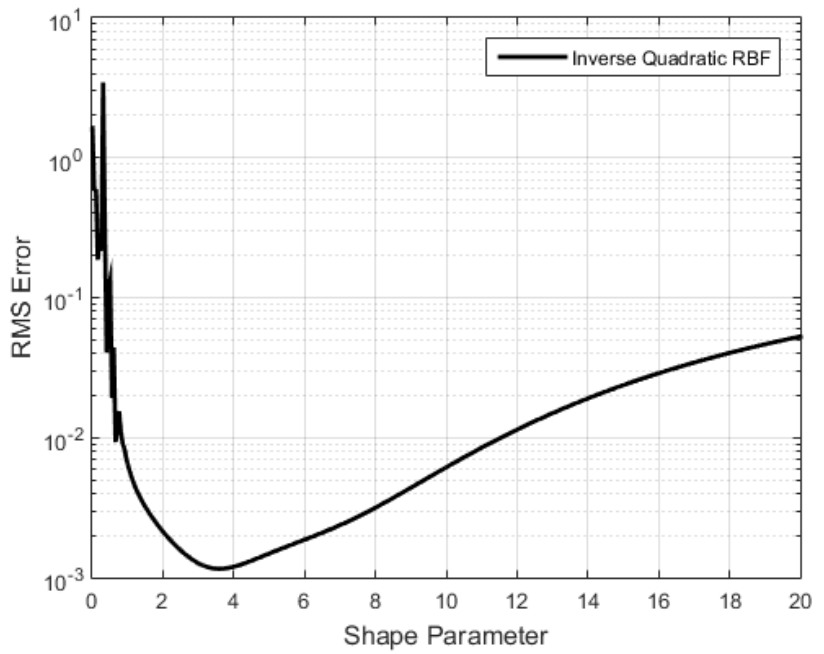

FIG. 2. RMS obtained shape parameter computed with $N_{\text{ctr}} = 10 \times 10$centers and $N_{int} = 7 \times 7$ interpolation nodes.

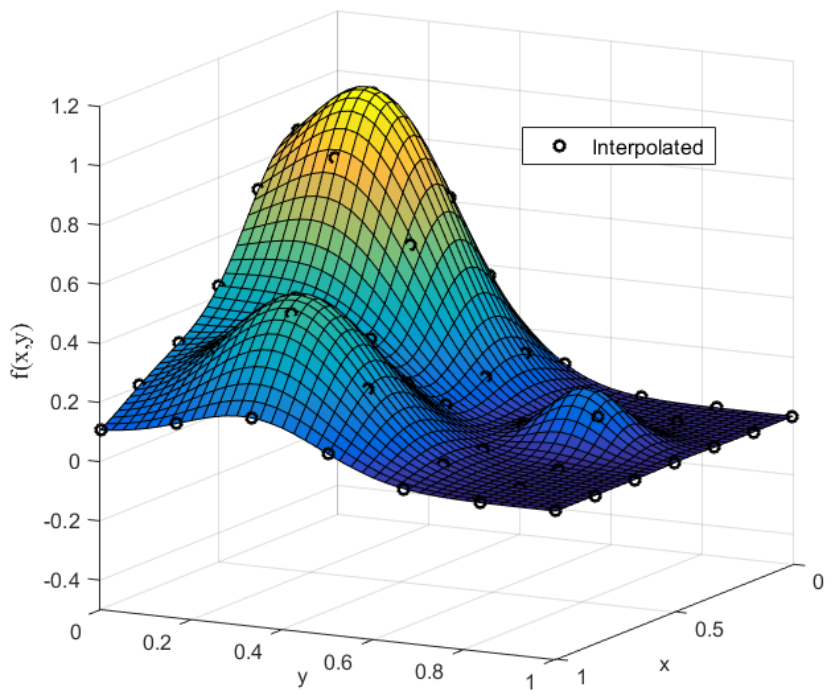

FIG. 3. Interpolated values with the exact surface computed with $\varepsilon = 3.5$.

Tab. 1. Absolute error of inverse quadratic-RBF interpolation using $\varepsilon = 3.5$ and $\varepsilon = 10.0$.

(x, y)	Abs. Error		(x, y)	Abs. Error	
	$\boldsymbol{\varepsilon = 3.5}$	$\boldsymbol{\varepsilon = 10.0}$		$\boldsymbol{\varepsilon = 3.5}$	$\boldsymbol{\varepsilon = 10.0}$
(0.000,0.666)	1.70E-05	3.37E-01	(0.666,0.333)	2.00E-05	2.61E-02
(0.000,1.000)	1.64E-05	2.21E-01	(0.666,0.666)	5.30E-05	2.72E-01
(0.333,0.000)	9.99E-06	4.09E-01	(0.666,1.000)	2.46E-05	5.12E-01
(0.333,0.333)	9.99E-06	3.22E-02	(1.000,0.000)	9.99E-06	7.44E-02
(0.333,0.666)	6.29E-05	2.66E-01	(1.000,0.333)	9.99E-06	1.68E-01
(0.333,1.000)	9.20E-06	2.09E-02	(1.000,0.666)	6.00E-06	3.29E-01

Example 4.2: Poisson with Nonrectangular Domain

The equation is shown below;

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = -x^2 \tag{53}$$

This is defined on the domain with an elliptical boundary, expressed as;

$$\frac{x^2}{4} + y^2 = 1 \tag{54}$$

Where the boundary condition is taken directly from the exact solution which is expressed as follows;

$$u(x, y) = -\frac{1}{246}(50x^2 - 8y^2 + 33.6)\left(\frac{x^2}{4} + y^2 - 1\right) \tag{55}$$

In this example, the shape values expected to lead to reasonably good results are adopted from the findings in example 1, i.e. $\varepsilon \in (2.0,5.0)$. The main focus is paid to the effect of the density of nodes contained in the domain and for this, at least 4 levels of nodes density are investigated. FIG. 4 displays two levels of density where the number of boundary nodes is kept constant at 40 for all cases.

Both RMS and L_∞ – error norms are carefully monitored, and the results are in Tab. 2.

As the number of nodes increases, it is found from both error norms that all the four numerical schemes under investigation produce results with more accuracy. Nevertheless, RPIM is seen to provide results with the comparatively lowest error at every level of nodes density. At the highest density of nodes, while KCM, HCM, and DRBEM have $L_\infty > 1.0E-03$, it is RPIM that produces solutions with only $L_\infty \approx 3.25E-04$. However, all errors revealed from all schemes in this example and at all levels of density of nodes, are found to remain under $5.00E-03$ which is reasonably well. An example of solutions obtained from DRBEM is plotted against the exact solution and is depicted in FIG. 5.

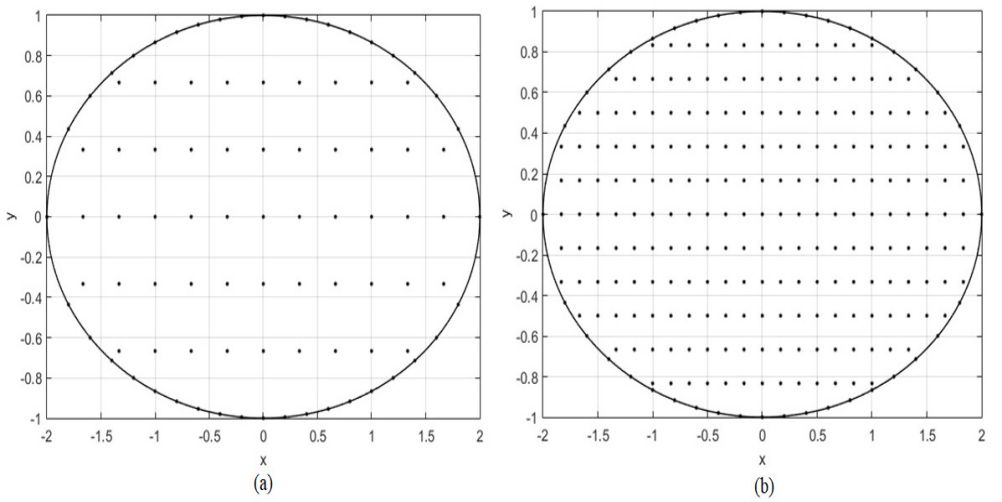

FIG. 4. Two levels of nodes density with a fixed number of boundary nodes of 40 and with; (a) 51 internal nodes, and (b) 217 internal nodes.

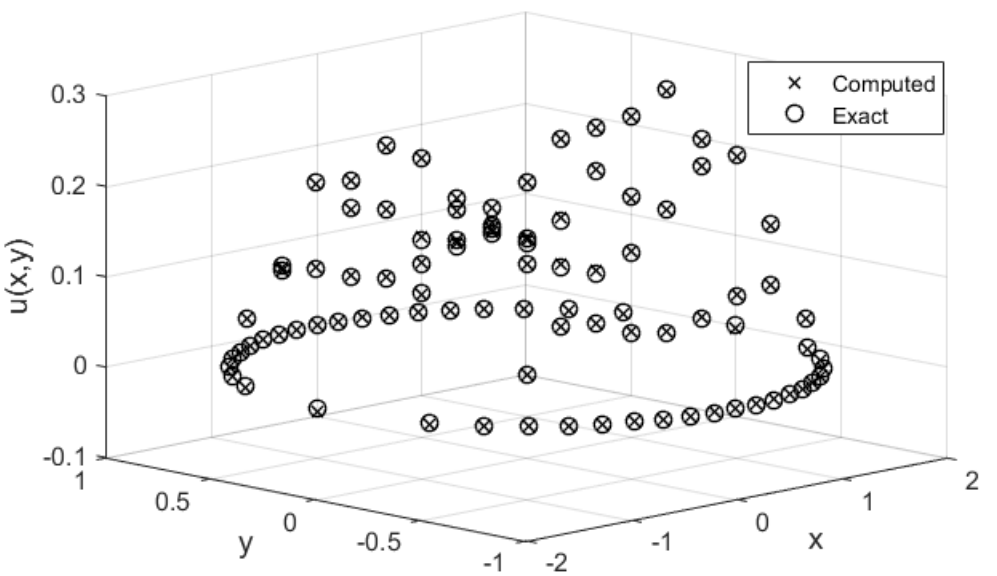

FIG. 5. DRBEM-solution plot against corresponding exact ones; 81 computational nodes and Inverse-Quadratic shape $\varepsilon = 3.0$.

Tab. 2. RMS- and L_∞-error, produced at 4 levels of densities of internal nodes.

Method	5×5		10×10		15×15		20×20	
	RMS	L_∞	**RMS**	L_∞	**RMS**	L_∞	**RMS**	L_∞
KCM	1.71E-03	2.15E-03	1.22E-03	2.03E-03	1.04E-03	1.19E-03	9.41E-04	1.27E-03
HCM	2.01E-03	2.74E-03	1.41E-03	2.46E-03	1.01E-03	2.05E-03	8.55E-04	1.18E-03
RPIM	9.44E-04	1.85E-03	8.62E-04	1.54E-03	5.23E-04	1.75E-03	3.86E-04	3.25E-04
DRBEM	1.64E-03	1.99E-03	1.37E-03	2.04E-03	1.22E-03	1.78E-03	7.07E-04	1.01E-03

Example 4.3: With Zero Source Term

In this example, the governing equation explained in Section 2.1 is solved with the main objective of investigating the error accumulated in time where the best shape parameter is still believed to follow the findings from example 4.1. In this case, we set $V_x = V_y = 0.8$ and $\eta_x = \eta_y = 0.01$ with zero sink and zero source terms, i.e. $\beta u = g(x) = 0$ .The governing equation is of the form as shown below;

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = \frac{1}{0.01}\left(\frac{\partial u}{\partial t} + 0.8\left(\frac{\partial u}{\partial x} + \frac{\partial u}{\partial y}\right)\right) \tag{56}$$

Where $(x, y) \in D = \{(x, y) \in \mathbb{R}^2 | 0.5 < x, y < 2\}$, $t \in 0, T$. Its exact solution is defined as follows;

$$u(x, y, t) = \frac{1}{4t+1} exp\left[-\frac{(x-0.8t-0.5)^2}{0.01(4t+1)} - \frac{(y-0.8t-0.5)^2}{0.01(4t+1)}\right] \tag{57}$$

Both initial condition and boundary conditions come from the exact solution. FIG. 6 depicts the exact solution profile at $\Delta t = 0.001$ computed using 20×20 computation nodes.

Tab. 3 compares RMS errors measured at chosen 7 internal nodes obtained at two time levels, $t = 0.5$ and $t = 1.0$, using $\Delta t = 0.001$ and 12×12 computational nodes. At the smaller $t = 0.5$, KCM and DRBEM are found to have approximately the same error magnitude where the other two, HCM and RPIM are close to each other with comparatively lower error range,$\approx 5.00E-04$ to $9.00E-04$. When the time level increases to $t = 1.0$, it is interesting to have found that while KCM, HCM, and DRBEM methods are seen to produce solutions with noticeably lower accuracy, it is RPIM that has been only slightly affected, with the RMS remaining in the range, i.e.$< 8.00E-04$.

When time increases even further, see FIG. 7, it can be clearly seen that all four domain-meshfree methods have growing their own $L_\infty -$ error norm. The highest magnitude in error is found to have been produced by KCM with above $L_\infty \approx 1.00E-01$ at $t = 2.00$. DRBEM is also found to have approximately the same growth rate in error as KCM, but is only slightly lower in overall. On the other hand, HCM and RPIM are found to remain at lower level of this type of error norm with the lowest value of $L_\infty -$ error is found to be obtained from RPIM, i.e.$L_\infty < 1.00E-02$.

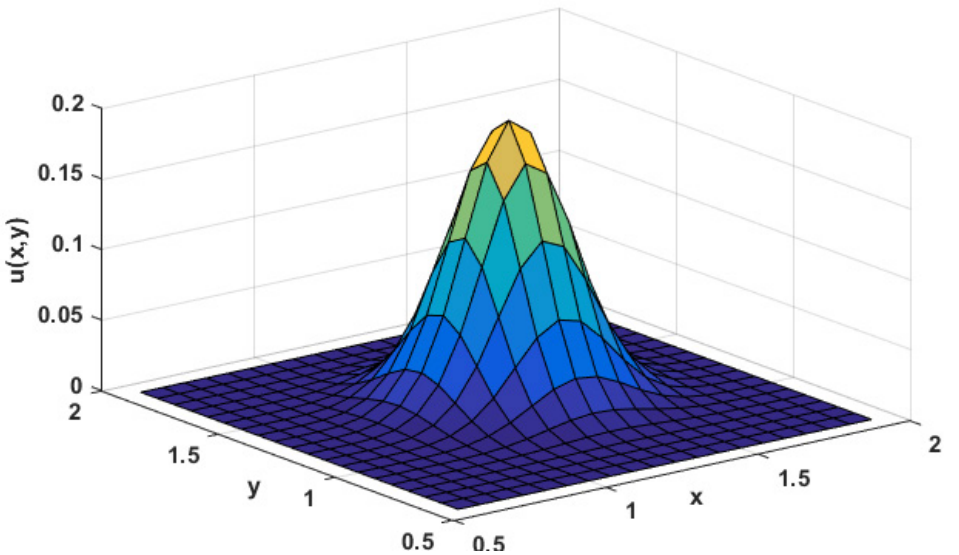

FIG. 6. Exact solution profile obtained with $\Delta t = 0.001$ at $t = 1.00$ and 20×20 nodes.

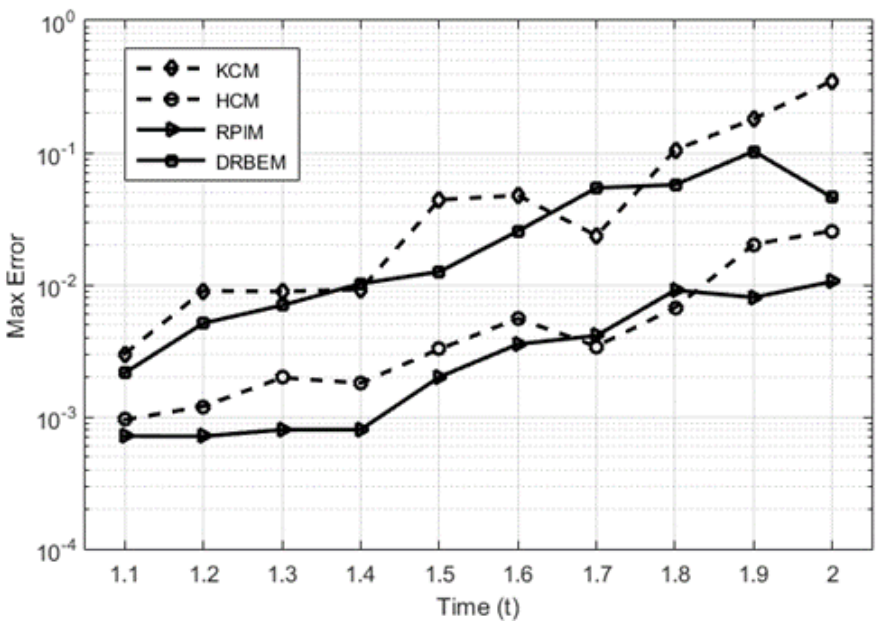

FIG. 7. L_∞ – Error measurement obtained at each time level $1.1 \leq t \leq 2.0$ with $\Delta t = 0.005$, $N = 12 \times 12$ and a fixed $\varepsilon = 3.5$.

Tab. 3. RMS error comparison at two time levels with $\Delta t = 0.001$ and using 12×12 nodes.

time	Center	KCM	HCM	RPIM	DRBEM
	(0.6,0.6)	1.641E-03	8.622E-04	6.024E-04	1.221E-03
	(0.8,0.8)	1.352E-03	8.045E-04	5.110E-04	1.039E-03
	(1.0,1.0)	1.547E-03	9.144E-04	6.117E-04	1.085E-03
$t = 0.5$	(1.2,1.2)	1.594E-02	8.014E-04	6.405E-04	1.002E-03
	(1.4,1.4)	2.054E-02	8.556E-04	5.759E-04	7.065E-03
	(1.6,1.6)	1.544E-03	9.012E-04	6.641E-04	2.112E-03
	(1.8,1.8)	1.219E-03	8.510E-04	5.882E-04	1.403E-03
	(0.6,0.6)	1.971E-03	9.844E-04	6.775E-04	2.004E-03
	(0.8,0.8)	1.774E-03	1.022E-03	7.059E-04	1.995E-02
	(1.0,1.0)	1.690E-03	2.012E-03	7.215E-04	1.704E-03
$t = 1.0$	(1.2,1.2)	1.704E-02	9.571E-04	6.994E-04	8.112E-02
	(1.4,1.4)	2.204E-02	1.102E-03	6.024E-04	1.295E-02
	(1.6,1.6)	1.821E-03	1.507E-03	7.302E-04	7.045E-02
	(1.8,1.8)	1.604E-03	9.991E-04	6.230E-04	9.881E-02

Example 4.4: With Nonzero Source Term

The governing equation is set to contain $\beta = 0$ and the source term is given by;

$$g(x, y, t) = (\eta_x V_x + \eta_y V_y)(e^{-t(\eta_x^3 + \eta_y^3)})cos(\eta_x x + \eta_y y) \tag{58}$$

The computational domain is $(x, y) \in D = \{(x, y) \in \mathbb{R}^2 | 0 < x, y < 2\}$, $t \in 0, T$ and all boundary conditions are taken from the analytical solution which is provided as follows;

$$u(x, y, t) = e^{-t(\eta_x^3 + \eta_y^3)} sin(\eta_x x + \eta_y y) \tag{59}$$

On this regular domain, it is set that the total number of centers or computational nodes is defined as the sum of the number of internal nodes and the boundary nodes or $N = N_{int} + N_{bdy}$. In order to cover as wide aspect; accuracy, computational time, node density and parameter effect, as possible, a large amount of numerical experiments were carried out and the main results are listed in Tab. 4. In this Table., only comparatively best results for each case produced by a certain value of shape parameter, ε_{opl}, are presented for each node density level.

In terms of effect due to the shape value, it is clearly seen that it is very much varying for each numerical method and no noticeable correlation can be established. When compare between DRBEM and KCM method, for instance, while the optimal values of shape used in DRBEM remain under $1.00E-01$, those found for KCM seem to increase beyond $1.00E+01$ as $r_{ij}=\left\|\mathbf{x}_i-\mathbf{x}_j\right\|_2$ is reduced, for all $V_x=V_y=V=10,10^2$ and 10^3. Moreover, the optimal shape does not always increase when the number of nodes creases and this aspect can be seen in the case of HCM and RPIM. When setting $V_x = V_y$ = 5,000, $\eta_x = \eta_y$ = 1 and with $N=19\times19$, FIG. 8 clearly shows that KCM is highly sensitive to the change of shape parameter value. In this FIGURE., the good quality of solutions produced by HCM, RPIM, and DRBEM can still be expected while utilizing the same shape values as listed in Tab. 4, yet this is not the case for KCM. Another evidence supporting this assumption is what is displayed in FIG. 9. It shows that the solution quality obtained from KCM is significantly improved when an optimal, ε_{opl} , is found and used for that particular case.

Tab. 4. Average Relative error ($Avg.Rel.Err.$, %) and computational time (CPU-T) at t = 0.75 at different values of $V_x = V_y = V$ for three node-density levels, computed using Δt = 0.0005, with its corresponding optimal shape ε_{opl} .

V	Method	$N=5\times5$ ($N_{int}=9, N_{bdy}=25$)			$N=11\times11$ ($N_{int}=94, N_{bdy}=27$)			$N=25\times25$ ($N_{int}=529, N_{bdy}=96$)		
		ε_{opl}	$Avg.Rel.Err.$ (%)	CPU-T	ε_{opl}	$Avg.Rel.Err.$ (%)	CPU-T	ε_{opl}	$Avg.Rel.Err.$ (%)	CPU-T
10	KCM	12.0	1.1127	16.422	15.5	1.4012	24.504	35.0	2.0025	27.844
	HCM	6.50	1.0122	55.201	10.0	1.0504	78.021	25.0	1.5800	102.001
	RPIM	5.50	1.2015	45.877	9.50	0.5488	71.778	30.0	0.4014	84.021
	DRBEM	0.05	1.2014	22.074	0.06	0.8441	33.851	0.09	0.5101	36.235
10^2	KCM	11.4	1.5412	20.104	16.0	1.1001	32.280	19.0	0.9211	45.561
	HCM	7.80	0.8925	62.502	14.5	0.8122	84.533	6.0	0.8001	121.025
	RPIM	10.5	0.7458	41.211	5.00	0.8014	70.266	19.0	0.7418	92.815
	DRBEM	0.04	1.0215	24.025	0.05	0.8010	34.002	0.08	0.8700	37.105
10^3	KCM	22.5	0.2154	28.524	32.0	0.1805	33.212	38.0	0.2058	51.299
	HCM	18.5	0.2001	80.221	6.50	0.0511	102.085	11.0	0.1844	144.544
	RPIM	15.0	0.0895	57.205	30.0	0.0219	86.260	25.0	0.0878	108.992
	DRBEM	0.05	0.1202	26.118	0.06	0.1054	34.108	0.10	0.1955	38.211

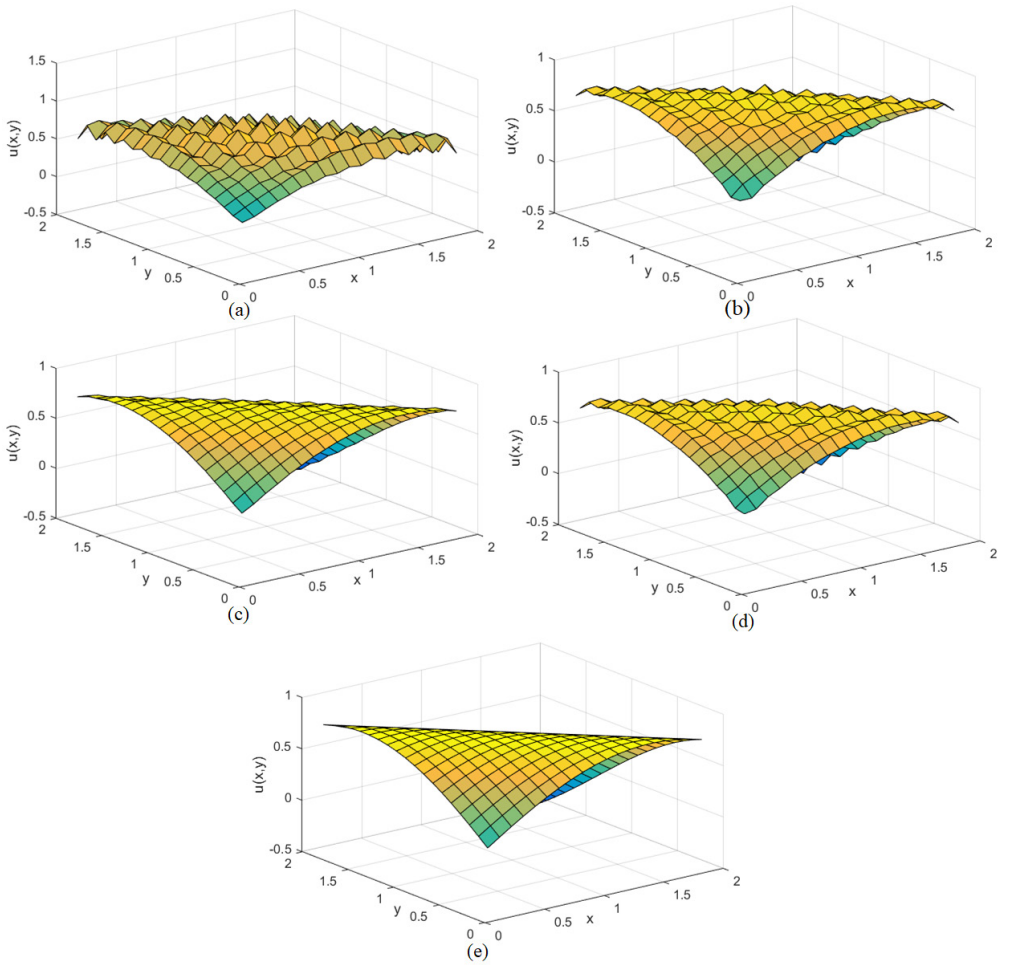

FIG. 8. Solution surface plot computed using $V_x = V_y = 5{,}000$, $\eta_x = \eta_y = 1$ at $t = 0.01$, using $\Delta t = 0.001$ with $N = 19 \times 19$; (a) KCM ($\varepsilon_{opl} = 38.0$), (b) HCM ($\varepsilon_{opl} = 11.0$), (c) RPIM ($\varepsilon_{opl} = 25.0$), (d) DRBEM ($\varepsilon_{opl} = 0.10$), and (e) Exact .

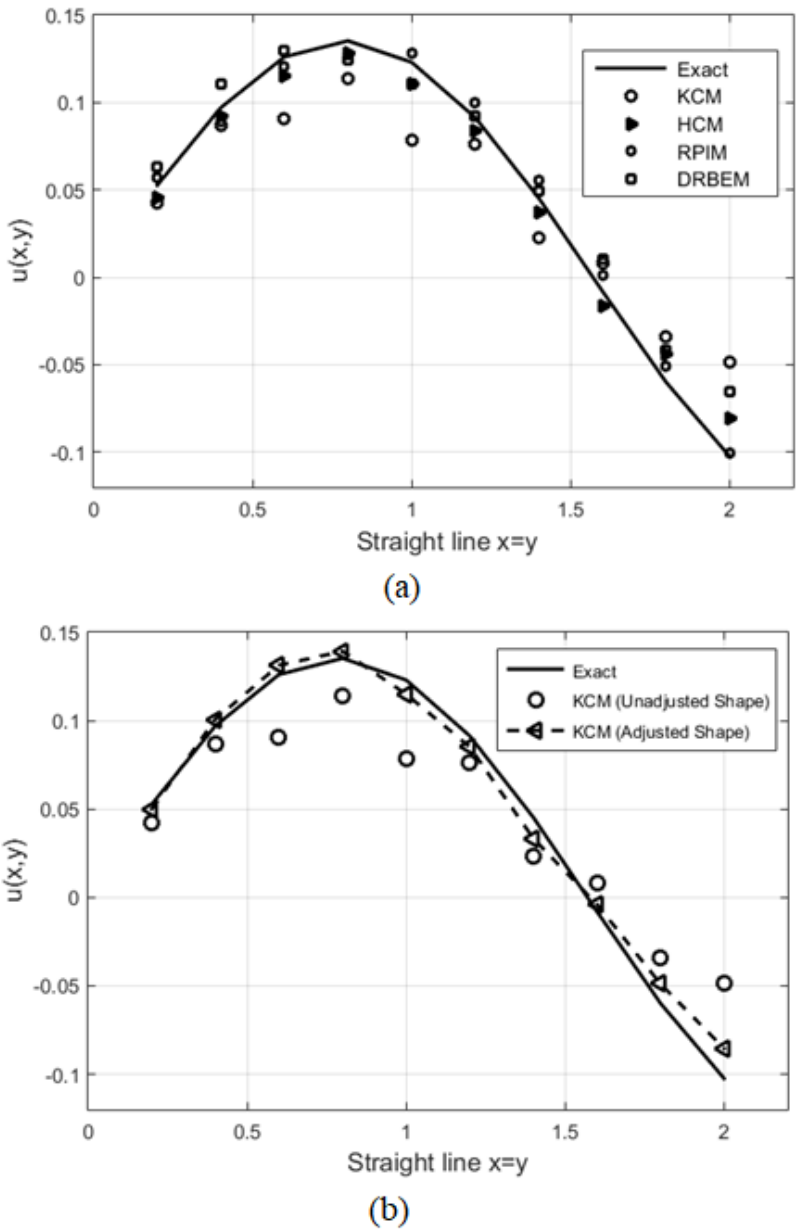

FIG. 9. Solution comparison $u(x, y)$; measured on the straight line $x = y$ across the unit domain for $V_x = V_y = 10{,}000, \eta_x = \eta_y = 1$ at $t = 1.00, \Delta t = 0.005$ with $N = 20 \times 20$; (a) KCM with unadjusted shape ($\varepsilon_{opl} = 38.0$), and (b) KCM with an adjusted shape ($\varepsilon_{opl} = 43.5$).

Example 4.5: Transient with Dirichlet boundary conditions

The special form of the problem is considered by setting;

$$c_x = \frac{V_x + \sqrt{V_x^2 + 4\eta_x}}{2\eta_x} \quad \text{and} \quad c_y = \frac{V_y + \sqrt{V_y^2 + 4\eta_y}}{2\eta_y} \tag{60}$$

The exact solution is as follows;

$$u(x, y, t) = e^t(e^{-c_x x} + e^{-c_y y}) \tag{61}$$

The initial and boundary conditions are then imposed using the above exact forms;

$$u(x, y, 0) = e^{-c_x x} + e^{-c_y y} \tag{62}$$

The Dirichlet boundary conditions;

$$u(0, y, t) = e^t(1 + e^{-c_y y}), \quad u(1, y, t) = e^t(e^{-c_x} + e^{-c_y y})$$

$$u(x, 0, t) = e^t(1 + e^{-c_x x}), \quad u(x, 1, t) = e^t(e^{-c_x x} + e^{-c_y}) \tag{63}$$

At $V_x = V_y = 1.0, \eta_x = \eta_y = 0.8$ and time $t = 0.05$,

Tab. 5 contains the numerical solutions at 8 points over the domain, obtained from all 4 schemes.

When time increases, nevertheless, from $t = 0.05$ to $t = 0.5$, all four methods have been found to have lost their capability as clearly displayed in FIG. 10 . KCM is found to strongly be affected by the time-increment while HCM and RPIM are much less affected. When time has been even further increased, it is very interesting to have seen that it is actually DRBEM that can provide good results. FIG. 11. confirms this argument where DRBEM leads to approximately the same quality of accuracy as those produced by RPIM. As the computation process continues from $t = 1.0$ to $t = 2.0$, however, the only scheme that can remain desirable solution accuracy while keeping the same ε_{opl} is actually RPIM and the evidence of this is shown in FIG. 12.

So far, it has been clearly seen that RPIM is capable of producing the best quality of numerical results and is least sensitive to the change of the shape parameter. The last experiment is on the effect on nodes density in the computational domain. FIG. 13 depicts the RMS error generated by all four method when the nodes increase, for the case of $V_x = V_y = 10, \eta_x = \eta_y = 0.8$ at $t = 1.5$ when using $\Delta t = 0.005$. Once again, RPIM is seen to be slightly affected by the distance between nodes while strong fluctuations in error clearly appear in the case of DRBEM and KCM. When comparing RPIM with HCM, the results show that HCM is even less sensitive to nodes density particularly during $50 < N < 250$. Nevertheless, in terms of the accuracy, RPIM is noticeably more accurate with RMS as low as $\approx 1.00E-05$.

Tab. 5. Numerical solutions produced by each RBF compared to the exact using $\Delta t = 0.001$ for $V_x = V_y = 1.0$, $\eta_x = \eta_y = 0.8$ and time $t = 0.05$.

Center	KCM	HCM	RPIM	DRBEM	Exact
(0.5,0.1)	1.27112	1.25998	1.27125	1.26601	1.274227
(0.9,0.1)	1.05401	1.15020	1.00523	1.04109	1.057985
(0.7,0.3)	0.86901	0.89012	0.87004	0.89901	0.870372
(0.1,0.5)	1.26801	1.21205	1.27311	1.27012	1.274227
(0.9,0.5)	0.58821	0.60012	0.59701	0.60123	0.594516
(0.3,0.7)	0.86810	0.79910	0.86905	0.86801	0.870372
(0.1,0.9)	1.10521	1.00214	1.07012	1.05007	1.057985
(0.5,0.9)	0.60188	0.59001	0.59102	0.59199	0.594516

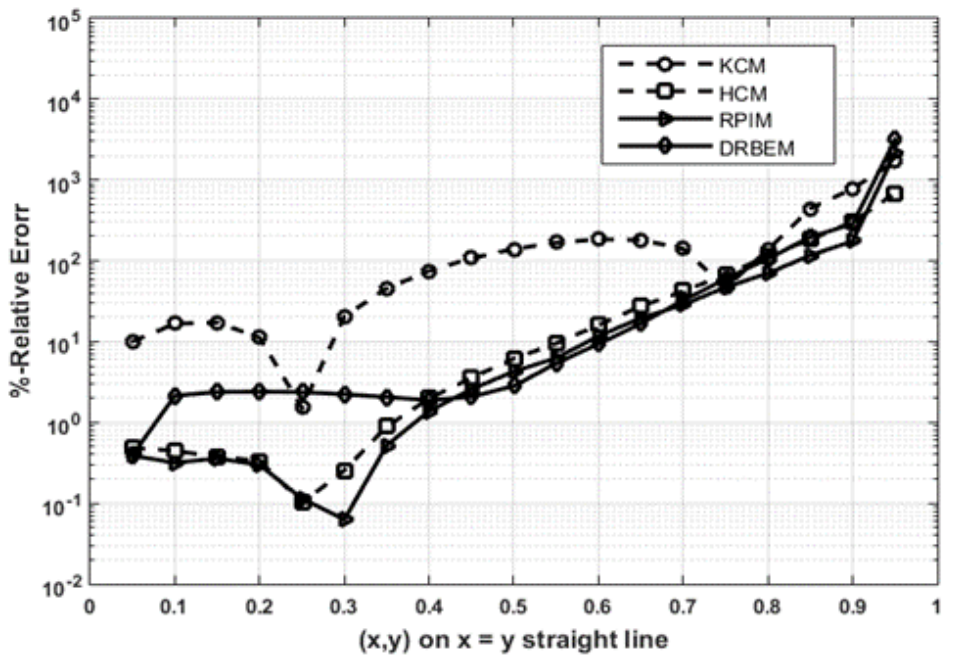

FIG. 10. Relative error percentage of solutions computed by the four methods for $V_x = V_y = 10, \eta_x = \eta_y = 1.0$ on the straight line $x = y$ at $t = 0.5$ using $\Delta t = 0.001$, and $N = 21 \times 21$.

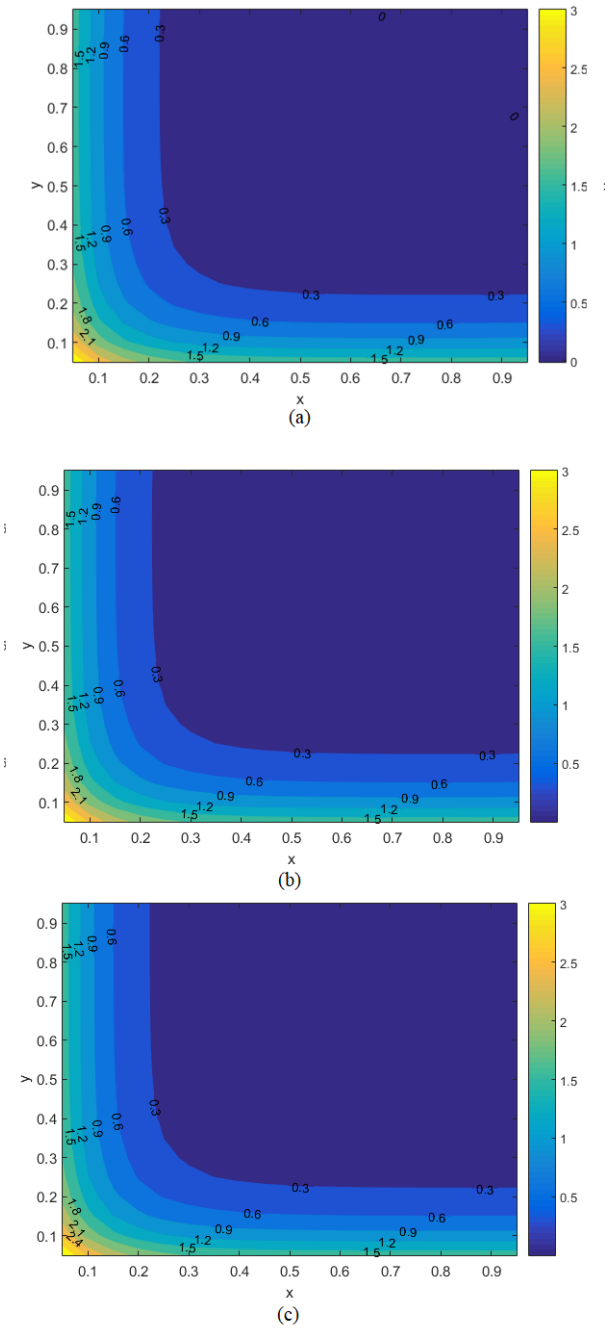

FIG. 11. Solution contours for $V_x = V_y = 10, \eta_x = \eta_y = 1.0$ at $t = 1.0$ when using $\Delta t = 0.001$, and $N = 21 \times 21$; (a) DRBEM, (b) RPIM, and (c) Exact.

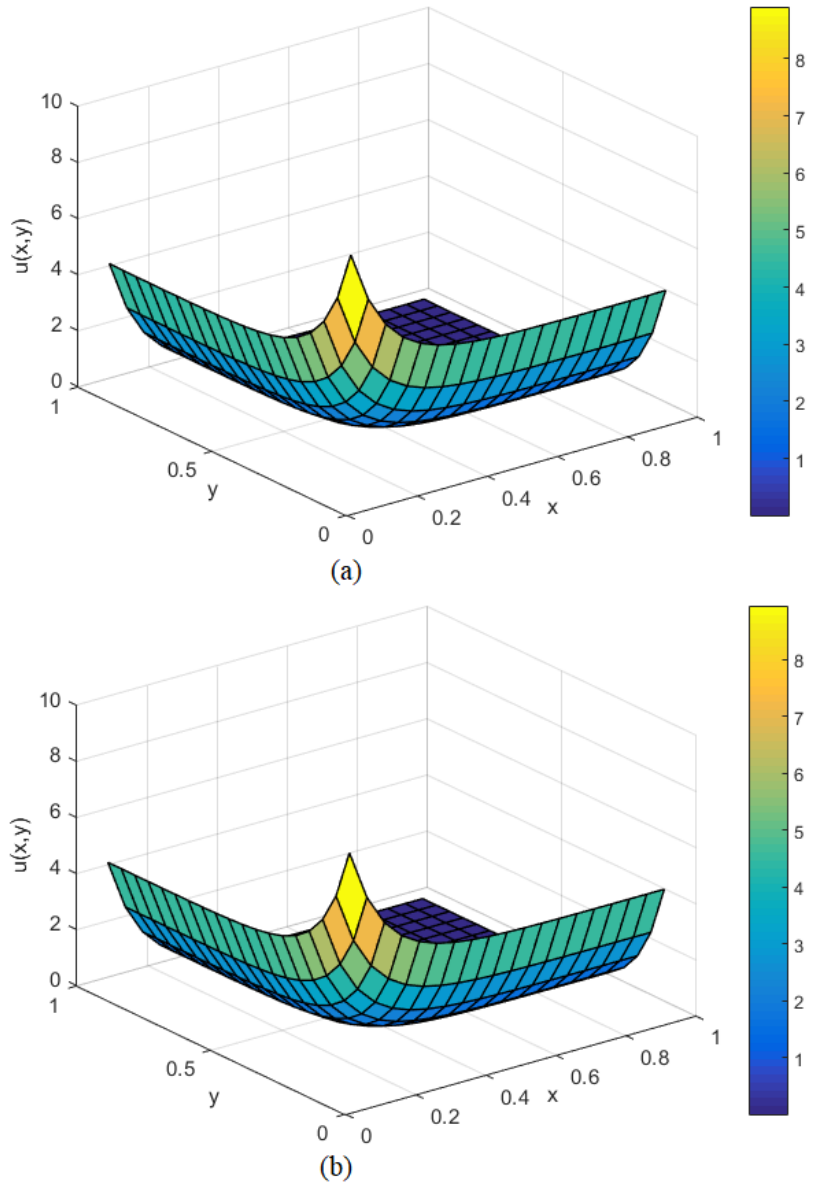

FIG. 12. Solution profiles for $V_x = V_y = 10, \eta_x = \eta_y = 1.0$ at $t = 2.0$ when using $\Delta t = 0.005$, and $N = 21 \times 21$; (a) RPIM and (b) Exact.

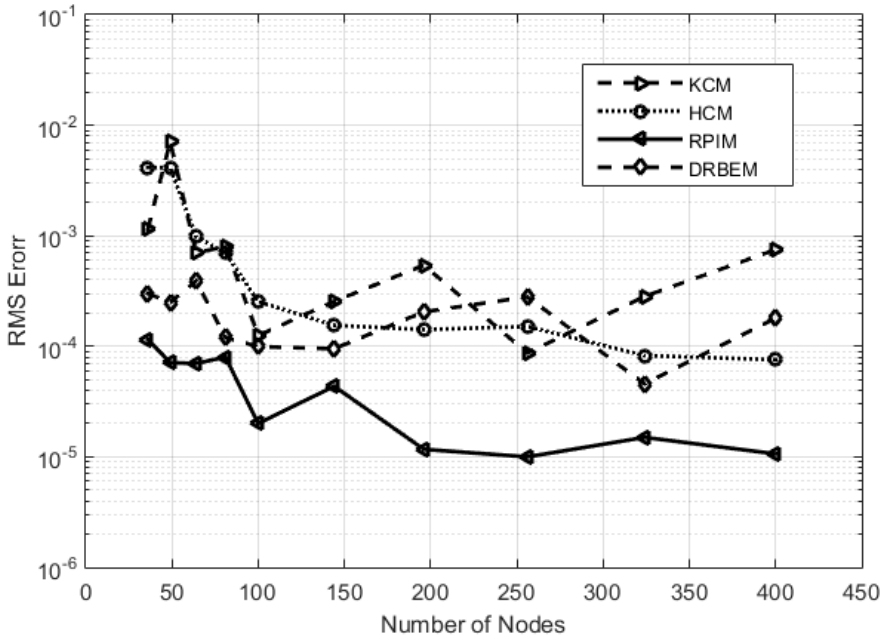

FIG. 13. RMS error progression in terms of number of computational nodes for $V_x = V_y = 10, \eta_x = \eta_y = 0.8$ at $t = 1.5$ when using $\Delta t = 0.005$.

5. CONCLUSION

In this study, one of the classical PDEs namely convection-diffusion is numerical solved by four numerical domain-meshfree approaches. The common and crucial component of all the methods is radial basis function and the inverse-quadratic (IQ) type is chosen. To reach the main purposes; to shed more light into the practical use of IQ-RBF and to compare the effectiveness of the four methods, a large series of numerical experiments were carries out and 4 aspects of each method are measured. The computed results have revealed the following conclusions;

1. In terms of the overall accuracy and error growth in time; all four schemes are found to have approximately the same error growth rate but different in magnitude, i.e. KCM and DRBEM are roughly one-order of magnitude higher than the other two.
2. In the time-consuming aspect, it appears that the Hermite type of collocation requires comparatively noticeably higher CPU-time. KCM is, on the other hand, found to be the fastest approach.
3. The impact caused by the number of nodes, N, and the change of shape parameter, ε, are seen to be very high under the context of KCM. DRBEM and RPIM are found the be much less sensitive to those factors.
4. In terms of the simplicity to construct and deploy, this can easily be judged by the way of approximating the solution $\tilde{u}(x_i)$. Based on this, it is concluded in this work that KCM is the simplest while HCM is the most complicated one followed by DRBEM and RPIM respectively.

All in all, if a good shape parameter can be achieved, this work has proved that KCM is the most suitable. for most practical use. Under the situation where internal nodes are less needed, DRBEM is an obvious choice where only boundary nodes play the major rules. For large and multidimensional domains, while KCM might encounter the problem of asymmetric and populated collocation matrix and while DRBEM can no longer handle properly, HCM can well be another alternative (provided that CPU-time has no limitation). In summary, it has been revealed that RPIM is the optimal approach in terms of all aspects and criteria mentioned above.

ACKNOWLEDGEMENT

The corresponding author would like to express his sincere appreciation to the Centre of Excellence in Mathematics, Thailand, for their kind support.

REFERENCES

[1] Djidjeli K., Nair P. B., Chinchapatnam P. P. and Price W. G., Global and compact meshless schemes for the unsteady convection–diffusion equation, In International symposium on health care and biomedical research interaction, Oujda: University of Southampton, 2004, p. 1-8.

[2] Morton K. W., Numerical solution of convection–diffusion problems, Chapman and Hall, London, 1996.

[3] Zienkiewicz O. C., Taylor R. L. and Nithiarasu P., The Finite Element Method for Fluid Dynamics, Butterworth-Heinemann, 2014.

[4] Kaennakham S. and Moatamedi M., An automatic mesh adaptation algorithm and its performance for simulation of flow over a circular cylinder at Re = 1.4 × 10^5, Int. J. Comput. Sci. Eng., 2014, vol. 9, p. 257-273.

[5] Kaennakham S., Holdø A. E., and Lambert C., A new simple h-mesh adaptation algorithm for standard Smagorinsky LES: a first step of Taylor scale as a refinement variable, Int. J. Multiphysics, 2010, vol. 4, p. 33-50.

[6] Liu G. R., Meshfree methods moving beyond the finite element method, CRC Press, Boca Raton, 2003.

[7] Chuathong N., Kaennakham S. and Toutip W., An automatic centroid-node adaptive meshless (CNAM) method for solving convection-diffusion problems, J. Eng. Appl. Sci., 2017, vol. 12, p. 6554-6291.

[8] Chuathong N. and Kaennakham S., A Numerical Investigation on Variable Shape Parameter Schemes in a Meshfree Method Applied to a Convection-Diffusion Problem, International Journal of Applied Engineering Research, 2017, vol. 12, p. 4162-4170.

[9] Kaennakham S. and Chuathong N., Solution to a Convection-Diffusion Problem Using a New Variable Inverse-Multiquadric Parameter in a Collocation Meshfree Scheme, International Journal of Multiphysics, 2017, vol. 11, p. 359-374.

[10] Chanthawara K., Kaennakham S. and Toutip W., The Numerical Study and Comparison of Radial Basis Functions in Applications of the Dual Reciprocity Boundary Element Method to Convection-Diffusion Problems, Progress in Applied Mathematics in Sciences and Engineering, AIP Conference Proceedings, 2016, vol. 020029, p. 1705.

[11] Kansa E. J., Multiquadrics–a scattered data approximation scheme with applications to computational fluid–dynamics–II solutions to parabolic, hyperbolic and elliptic partial differential equations, Comput Math Appl; 1990, vol. 19, p. 147–61.

[12] Chandhini G. and Sanyasiraju Y. V. S. S., Local RBF-FD solutions for steady convection-diffusion problems, Int. J. Numer. Methods Eng., 2007, vol. 72, p. 352-378.

[13] Chinchapatnam P. P., Djidjeli K. and Nair P. B., Unsymmetric and symmetric meshless schemes for the unsteady convection–diffusion equation, Comput. Methods Appl. Mech. Eng., 2006, vol. 195, p. 2432-2453.

[14] Ferreira A. J. M., Roque C. M. C. and Martins P. A. L. S., Radial basis functions and higher-order shear deformation theories in the analysis of laminated composite beams and plates, Compos. Struct., 2004, vol. 66, p. 287-293.

[15] Liu Y., Liew K. M., Hon Y. C. and Zhang X., Hon Y.C. and Zhang, X., Numerical simulation and analysis of an electroactuated beam using a radial basis function, Smart Materials and Structures, 2005, vol. 14, p. 1163-1171.

[16] Cheng A. H .D., Golberg M. A., Kansa E. J. and Zammito G., Exponential convergence and H-c multiquadric collocation method for partial differential equations, Numer. Meth. Part. D. E., 2003, vol. 19, p. 571-594.

[17] Li H. and Mulay S. S., Meshless Methods and Their Numerical Properties, CRC Press, Boca Raton, 2013.

[18] Fasshauer G. E., Solving partial differential equations by collocation with radial basis functions, Proceedings of Chamonix, 1996, vol. 1997 ,p. 1-8.

[19] Chen R. and Wu Z., Solving hyperbolic conservation laws using multiquadric quasi-interpolation, Numer. Meth. Part. D. E., 2006, vol. 22, p. 776-796.

[20] Wang J. G. and Liu G. R., A point interpolation meshless method based on radial basis functions, Int. J. Numer. Methods Eng., 2002, vol. 54, p. 1623-1648.

[21] Liu X., Liu G. R., Tai K. and Lam K. Y., radial basis point interpolation collocation method for 2-d solid problem, In Advances in Meshfree and X-FEM Methods, World scientific, 2011, p. 35-40.

[22] Bozkurt O. Y., Kanber B. and ASIk M. Z., Assessment of RPIM shape parameters for solution accuracy of 2D geometrically nonlinear problems, Int. J. Comput. Methods, 2013, vol. 10 (3), p. 1350003.

[23] Ghaffarzadeh H., Barghian M., Mansouri A. and Sadeghi M. H., Study of Meshfree Hermite Radial Point Interpolation Method for Flexural Wave Propagation Modeling and Damage Quantification. Lat. Am. J. Solid Struct., 2016, vol. 13, p. 2606-2627.

[24] Ma J., Wei G., Liu D. and Liu G., The numerical analysis of piezoelectric ceramics based on the Hermite-type RPIM, Appl. Math. Comput., 2017, vol. 309, p. 170-182.

[25] Toutip W., The dual reciprocity boundary element method for linear and nonlinear, in Department of Mathematics. University of Hertfordshire: United Kingdom, 2001.

[26] Nardini D. and Brebbia C. A., A new approach to free vibration analysis using boundary elements, Appl. Math. Modell., 1983, vol. 7, p. 157-162.

[27] Liu H. W., Numerical modeling of the propagation of ocean waves, University of Wollongong, 2001.

[28] Bruntona I. L. and Pullan A. J., A semi-analytic boundary element method for parabolic problem, Eng. Anal. Boundary Elem., 1996, vol. 18, p. 253-264.

[29] Zhu S.-P., Liu H.–W. and Lu X.–P., A combination of LTDRM and ATPS in solving diffusion problems, Eng. Anal. Boundary Elem., 1998, vol. 21, p. 285-289.

[30] Chuathong N. and Toutip W., An accuracy comparison of solutions between Boundary element method and meshless method for Laplace equation, In Proceedings of the 16th Annual Meeting in Mathematics (AMM2011), Khon Kaen University, Khon Kaen, 2011, p. 29-42.

[31] Li J., Chen Y. and Pepper D., Radial basis function method for 1-D and 2-D groundwater contaminant transport modeling, Computational Mechanics, 2003, vol. 32, p. 10-15.

[32] Hon Y. C. and Schaback R., On unsymmetric collocation by radial basis functions, Applied Mathematics and Computation, 2001, vol. 119, p. 177-186.

[33] Islam S., Šarler B., Vertnik R. and Kosec G., Radial basis function collocation method for the numerical solution of the two–dimensional transient nonlinear coupled Burgers' equations, Appl Math Model; 2012, vol. 36 (3), p. 1148–60.

[34] Leitao V. M. A., RBF-based meshless methods for 2D elastostatic problems, Engineering Analysis with Boundary Elements, 2004, vol. 28, p. 1271-1281.

[35] Rocca A. L., Power H., Rocca V. L. and Morale M., A meshless approach based upon radial basis function Hermite collocation method for predicting the cooling and the freezing times of foods, CMC Computers Materials and Continua, 2005, vol. 2, p. 239-250.

[36] Rosales A. H., Rocca A. L. and Power H., Radial basis function Hermite collocation approach for the numerical simulation of the effect of precipitation inhibitor on the crystallization process of an over-saturated solution, Numerical Methods for Partial Differential Equations, 2006, vol. 22, p. 361-380.

[37] Naffaa M. and Gahtani H. J. A., RBF-based meshless method for large deflection of thin plates, Engineering Analysis with Boundary Elements, 2007, vol. 31, p. 311-317.

[38] Franke R., Scattered data interpolation: tests of some methods, Math Comput, 1982, vol. 38, p. 181–200.