# Comparative Analysis of Parameter Influence on the Performance of Two Local Meshless Methods for Convection-Diffusion Equations

PICHAPOP PAEWPOLSONG[1], DUSITA RITTHISON[2], FELIX R. SAUCEDO-ZENDEJO[3], SAYAN KAENNAKHAM[2], KRITTIDEJ CHANTHAWARA[4,*]

[1]Department of Interdisciplinary Science and Internationalization,
Institute of Science, Suranaree University of Technology,
Nakhon Ratchasima, 30000
THAILAND

[2]School of Mathematics and Geoinformatics, Institute of Science,
Suranaree University of Technology,
Nakhon Ratchasima, 30000
THAILAND

[3]Centro de Investigación en Matemáticas Aplicadas,
Universidad Autónoma de Coahuila,
Coahuila, 25280
MEXICO

[4]Department of Mathematics, Faculty of Science,
Ubon Ratchathani Rajabhat University,
Ubon Ratchathani, 34000
THAILAND

**Corresponding Author*

*Abstract:* - In this study, two local meshless methods: the Local Direct Meshless Method (LDR) and the Local Indirect Meshless Method (LiDR), are comprehensively compared for numerically approximating convection-diffusion partial differential types of equations (PDEs). Unlike other conventional collocation-related meshless methods, where the typical choice of radial basis function (RBF) is the famous Gaussian one, this work focuses on an alternative called the multiquadric RBF. The investigation tests, monitors, and records several parameters and their influences on the methods' performances using 4 benchmarking test cases. It has been revealed that the LDR is capable of providing more consistent, better accuracy, especially over a wide range of durations and varying node configurations. The LiDR, on the other hand, is found to yield reasonable but less accurate solutions, particularly in the convection-dominated scenarios. Other than this, the work also provides useful insights regarding the effects of the shape parameter and the size of the subdomain, where the LDR's improved performance makes it more promising for practical applications, which may include those related to engineering processes and environmental challenges. These findings contribute to the understanding of local meshless methods for solving convection-diffusion PDEs and offer valuable guidance for method optimization and future research.

*Key-Words:* - Differential Quadrature, Local Integration, Radial Basis Function, Multiquadric, Meshless Method, Convection-Diffusion.

Received: April 5, 2026. Revised: May 9, 2026. Accepted: June 13, 2026. Published: August 3, 2026.

## 1 Introduction

Over the past decades, it has been widely recognized that partial differential equations (PDEs) are indispensable mathematical tools in the process of understanding (i.e., dealing with) a wide range of problems encountered by humanity, ranging from scientific to engineering disciplines. They have effectively equipped researchers with the capability to model and search for solutions to several of the most challenging natural phenomena with meaningful and quantitative insights, [1], [2]. With the variations in categories of Mother Nature, these equations appear in several forms and conditions, with the ultimate aim being to most correctly mimic the true nature being investigated at hand. This includes the convection–diffusion equation, the wave equation, the Navier–Stokes equations describing fluid flows, and reaction–diffusion systems that explain the spread and evolution of chemical substances, among many others, [3].

Known as one of the most impactful types of PDEs, the convection–diffusion type appears challenging both in terms of its wide range of utility in modeling various physical and engineering phenomena and applications, and in terms of the process of reaching reliable solutions when analytical solutions are not attainable, making the attempt to numerically approximate them more hopeful. Within its structure, two main mechanisms have crucial roles to play, as the name indicates, which are convection and diffusion. Convection describes the movement of substances due to fluid flow, whereas diffusion depicts the movement of such substances from higher concentration to lower concentration areas. The integration of these two mechanisms makes solving for the solution not straightforward, bringing huge interest to researchers, [4], [5], [6]. Its diverse challenges investigated include food processing [7], atmospheric pollutant dispersion modeling [8], thermal and energy management systems [9], financial quantitative modeling [10], and broader investigations into transport, transfer, and dispersion phenomena where patterns and dynamic interactions significantly influence the overall behavior of the system [11], see more in their references.

As previously mentioned, reaching the analytical solution to such types of PDEs is not so hopeful (and even impossible in many cases), so numerical methods and strategies have become a promising alternative for obtaining such solutions. Within this scope, several approaches are already well known, including the Finite Difference Method (FDM) [12], [13], Finite Element Method (FEM) [14], [15], and Finite Volume Method (FVM) [16], which have traditionally dominated the field of PDE discretization. They, nevertheless, do not come in that handy for several applications, such as those involving numerical instabilities and spurious oscillations, particularly in problems dominated by convective behavior. To alleviate these undesirable situations, a few other choices of methods have also been proposed and tested, such as the Streamline Upwind/Petrov–Galerkin (SUPG) method [17], Local Projection techniques [18], and stencil-based schemes [19]. Additionally, Adaptive Mesh Refinement (AMR) strategies have emerged as essential tools to enhance the resolution of solutions near regions exhibiting steep gradients, albeit with the trade-off of increased computational complexity, [20].

Apart from those just referred to in the previous paragraph, there are two families of numerical methods: those where nodes and grids are constructed only on the domain boundary, and those where only nodes are needed. The first one includes methods such as the Boundary Element Method (BEM) [21], [22], and the other grid-independent approaches have demonstrated notable advantages, including flexibility in implementation and reduced geometric constraints [23], [24], and this family is what this work focuses on. With the absence of a mesh, these methods are known as “meshless” or “meshfree” methods and have been receiving a huge amount of interest from researchers in the area. Specifically, the pioneering work in [25] demonstrated the promising potential of Radial Basis Functions (RBFs), showing improved numerical results in several types of problems described by PDEs, [26]. Some recent investigations to be recommended are the compactly supported RBFs, local and global collocation strategies, and Modified Kansa methods, which have considerably enhanced stability, accuracy, and computational efficiency, [27], [28], [29].

Under the challenges of dealing with convection–diffusion phenomena, RBF-based methods have shown several desirable aspects as they avoid the intricacies and computational overhead associated with mesh generation. Nevertheless, those primary RBF-based works were designed to take into account all the nodes from the entire domain, leading to unavoidable poor scalability and computational inefficiency, particularly when dealing with real-time applications. To mitigate this not-so-impressive aspect, there have been a few alternatives, such as those focusing on the localized version and those attempting to automatically adapt the concentration of the nodes involved. This includes H-adaptive RBF finite difference schemes [30], Shishkin-type node clustering [31], optimal shape parameter tuning [32], automatic node redistribution strategies [33], and non-interior node treatments [34]. These localized methods promise enhanced computational efficiency, accuracy, and scalability.

Inspired by the potential and positive aspects of these local-based RBF strategies, this work pays special attention to two fashions of localizing and mathematical manipulating, which are the Local Direct Meshless Method (LDR) introduced in [35], and the Local Indirect Meshless Method (LiDR) formulated in [36] and extended in [37]. The main contributions of this work are as follows:

(i) We perform a systematic comparison of the LDR and LiDR methods for transient convection–diffusion problems using MQ-RBFs, covering four benchmark test cases (from pure diffusion to strongly convection-dominated regimes) [38], [39], [40], [41], [42].

(ii) We study in detail the influence of the shape parameter c and the local subdomain size n_s on the accuracy and stability of both methods.

(iii) Based on the numerical results, we provide practical guidelines for choosing c and n_s, and show that LDR is generally more accurate and robust than LiDR, especially in convection-dominated problems. effectively.

The layout of the work is that we provide the mathematical construction of the two chosen local meshless methods in Section 2, followed by their theoretical implementations to the target PDEs, which are detailed in Section 3. Section 4 then describes the main algorithm, the process of calculation, and the main results obtained for the four benchmarking test cases. We then draw some crucial and useful conclusions and findings in Section 5.

# 2 Local (RBF) Collocation Methods

## 2.1 The Local Direct RBF Collocation (LDR) Method

The formulation of this type of local RBF quadrature starts with assuming that for each subdomain $\Omega_i$ containing $n_s$ elements, the function $u(\mathbf{p}_i)$ is represented by a linear combination of a chosen RBF $\varphi_j(\mathbf{p}_i)$ and the coefficient vector, $\alpha^{[i]}$ as defined as follows.

$$u(\mathbf{p}_i)=\sum_{j=1}^{n_s}\varphi_j(\mathbf{p}_i)a_j^{[i]},\quad \mathbf{p}_i\in\Omega_i, \tag{1}$$

where $j=1,2,...,n_s$. In this work, a popular choice of RBFs known as 'multiquadric (MQ)' is under investigation, and its formula is expressed as follows

$$\varphi_j(\mathbf{p}_i)=\sqrt{\left(r_j^{[i]}\right)^2+c^2}. \tag{2}$$

The positive constant $c$ is known as the shape parameter, known to play a critical role in determining solution quality, and $r_j^{[i]}$ denotes the distance between vector points $\mathbf{p}_i$ and $\mathbf{p}_j^{[i]}$ in the local subdomain $\Omega_i$, generally taken to be the Euclidean format, i.e.,

$$r_j^{[i]}=\left\|\mathbf{p}_i-\mathbf{p}_j^{[i]}\right\|. \tag{3}$$

It follows from Eq. (1) that.

$$u_\xi^{(n)}(\mathbf{p}_i)=\sum_{j=1}^{n_s}\frac{\partial^{(n)}\varphi_j(\mathbf{p}_i)}{\partial\xi^{(n)}}a_j^{[i]},\quad \mathbf{p}_i\in\Omega_i, \tag{4}$$

and

$$u\left(\mathbf{p}_k^{[i]}\right)=\sum_{j=1}^{n_s}\varphi_j\left(\mathbf{p}_k^{[i]}\right)a_j^{[i]}, \tag{5}$$

$\mathbf{p}_k^{[i]}\in\Omega_i$, $k=1,2,...,n_s$. It is possible to write Eq. (5) as follows.

$$\mathbf{u}_{n_s}^{[i]}=\boldsymbol{\varphi}_{n_s\times n_s}^{[i]}\boldsymbol{\alpha}^{[i]}. \tag{6}$$

The coefficients $a_j^{[i]}$ are calculated by inverting the matrix $\boldsymbol{\varphi}_{n_s\times n_s}$, as shown in

$$\boldsymbol{\alpha}^{[i]}=\left[\boldsymbol{\varphi}_{n_s\times n_s}^{[i]}\right]^{-1}\mathbf{u}_{n_s}^{[i]}. \tag{7}$$

From Eq. (1), it follows that.

$$u(\mathbf{p}_i)=\left[\varphi_1(\mathbf{p}_i)\,\varphi_2(\mathbf{p}_i)...\varphi_{n_s}(\mathbf{p}_i)\right]\boldsymbol{\alpha}^{[i]},\ \mathbf{p}_i\in\Omega_i. \tag{8}$$

By substituting $\boldsymbol{\alpha}^{[i]}$ from Eq. (7) into Eq. (8), the following is obtained for $\mathbf{p}_i\in\Omega_i$,

$$u(\mathbf{p}_i)=\left[\varphi_1(\mathbf{p}_i)\varphi_2(\mathbf{p}_i)...\,\varphi_{n_s}(\mathbf{p}_i)\right]\left[\boldsymbol{\varphi}_{n_s\times n_s}^{[i]}\right]^{-1}\mathbf{u}_{n_s}^{[i]}. \tag{9}$$

Consequently,

$$u_\xi^{(n)}(\mathbf{p}_i)=\left[\frac{\partial^{(n)}\varphi_1(\mathbf{p}_i)}{\partial\xi^{(n)}}\frac{\partial^{(n)}\varphi_2(\mathbf{p}_i)}{\partial\xi^{(n)}}...\frac{\partial^{(n)}\varphi_{n_s}(\mathbf{p}_i)}{\partial\xi^{(n)}}\right]\times\left[\boldsymbol{\varphi}_{n_s\times n_s}^{[i]}\right]^{-1}\mathbf{u}_{n_s}^{[i]}. \tag{10}$$

Which can be rewritten as follows

$$u_\xi^{(n)}(\mathbf{p}_i)=\frac{\partial^{(n)}\vec{\varphi}(\mathbf{p}_i)}{\partial\xi^{(n)}}\left[\boldsymbol{\varphi}_{n_s\times n_s}^{[i]}\right]^{-1}\mathbf{u}_{n_s}^{[i]}. \tag{11}$$

This completes the derivation of the local direct meshless method or LDR formulation.

## 2.2 The Local Indirect RBF Collocation (LiDR) Method

In contrast with the local direct method described above, the indirect local RBF collocation (Local Indirect, LiDR) method starts with the approximation of the highest order of derivative occurring in the governing equation. This means that for a second-order derivative $u''(\mathbf{p})$, its approximation can be defined under this concept as

$$u_{\xi}^{(n)}(\mathbf{p}_i)=\sum_{j=1}^{n_s}\varphi_j(\mathbf{p}_i)a_j^{[i]},\quad \mathbf{p}_i\in\Omega_i.$$

From this point, an integration procedure with respect to $x$ can be performed as expressed below.

$$\begin{aligned}u_x^{(1)}(\mathbf{p}_i)&=\int u_x^{(2)}(\mathbf{p}_i)dx\\&=\int\sum_{j=1}^{n_s}\varphi_j(\mathbf{p}_i)a_j^{[i]}dx\\&=\sum_{j=1}^{n_s}a_j^{[i]}\int\varphi_j(\mathbf{p}_i)dx\\&=\sum_{j=1}^{n_s}a_j^{[i]}H_j(\mathbf{p}_i)+C_2(y).\end{aligned}\tag{12}$$

The same procedure is repeated to yield the following.

$$\begin{aligned}u(\mathbf{p}_i)&=\int u_x^{(1)}(\mathbf{p}_i)dx\\&=\int\left(\sum_{j=1}^{n_s}a_j^{[i]}H_j(\mathbf{p}_i)+C_2(y)\right)dx\\&=\sum_{j=1}^{n_s}a_j^{[i]}\int H_j(\mathbf{p}_i)dx+C_2(y)x+C_1(y)\\&=\sum_{j=1}^{n_s}a_j^{[i]}\bar{H}_j(\mathbf{p}_i)+C_2(y)x+C_1(y),\end{aligned}\tag{13}$$

where $H_j(\mathbf{p}_i)=\int\varphi_j(\mathbf{p}_i)dx$, and $a_j^{[i]}$ are their coefficients. Note that the integration with respect to $y$ can be done similarly. The additional polynomial term in both Eq. (12) and Eq. (13) can be achieved by minimizing the sum of squared errors, as nicely done in [36]. Despite this, the work provided in [41] has shown that the method still performs well without the polynomial term. We have verified in preliminary tests (not shown) that including polynomials does not significantly improve the error in our benchmark problems, but can worsen the conditioning of the local matrices, so it is not included in this study.

As previously stated, only the multiquadric MQ-RBF is considered and according to [41], the expressions for the integrated MQ-RBF forms can be expressed as follows:

$$\begin{aligned}H_j^{(\xi)}(\mathbf{p}_i)&=\int\varphi_j(\mathbf{p}_i)d\xi\\&=\frac{\left(\xi_{\mathbf{p}_i}-\xi_{\mathbf{p}_j^{[i]}}\right)\sqrt{\left(r_j^{[i]}\right)^2+c^2}}{2}\\&\quad+\frac{\left(r_j^{[i]}\right)^2-\left(\xi_{\mathbf{p}_i}-\xi_{\mathbf{p}_j^{[i]}}\right)^2+c^2}{2}\\&\quad\times\ln\left(\xi_{\mathbf{p}_i}-\xi_{\mathbf{p}_j^{[i]}}+\sqrt{\left(r_j^{[i]}\right)^2+c^2}\right),\end{aligned}$$

and

$$\begin{aligned}\bar{H}_j^{(\xi)}(\mathbf{p}_i)&=\int H_j^{(\xi)}(\mathbf{p}_i)d\xi\\&=\frac{\left(\left(r_j^{[i]}\right)^2+c^2\right)^{\frac{3}{2}}}{6}\\&\quad+\frac{\left(r_j^{[i]}\right)^2-\left(\xi_{\mathbf{p}_i}-\xi_{\mathbf{p}_j^{[i]}}\right)^2+c^2}{2}\left(\xi_{\mathbf{p}_i}-\xi_{\mathbf{p}_j^{[i]}}\right)\\&\quad\times\ln\left(\xi_{\mathbf{p}_i}-\xi_{\mathbf{p}_k}+\sqrt{\left(r_j^{[i]}\right)^2+c^2}\right)\\&\quad-\frac{\left(r_j^{[i]}\right)^2-\left(\xi_{\mathbf{p}_i}-\xi_{\mathbf{p}_j^{[i]}}\right)^2+c^2}{2}\sqrt{\left(r_j^{[i]}\right)^2+c^2}.\end{aligned}$$

As a result, the original $u(\mathbf{p}_i)$ can be given by the following equation.

$$u(\mathbf{p}_i)=\sum_{j=1}^{n_s}\bar{H}_j^{(\xi)}(\mathbf{p}_i)a_j^{[i]},\quad \mathbf{p}_i\in\Omega_i.\tag{14}$$

Eq. () indicates that.

$$u\left(\mathbf{p}_k^{[i]}\right)=\sum_{j=1}^{n_s}\bar{H}_j^{(\xi)}\left(\mathbf{p}_k^{[i]}\right)a_j^{[i]},$$

for $\mathbf{p}_k^{[i]}\in\Omega_i$, $k=1,2,...,n_s$. This can be expressed as a $n_s\times n_s$ linear system as

$$\mathbf{u}_{n_s}^{[i]}=\bar{\mathbf{H}}_{n_s\times n_s}^{[i]}\boldsymbol{\alpha}^{[i]}.$$

Inverting the matrix $\bar{\mathbf{H}}_{n_s\times n_s}^{[i]}$ yields the coefficients $a_j^{[i]}$ in the following form.

$$\boldsymbol{\alpha}^{[i]}=\left[\bar{\mathbf{H}}_{n_s\times n_s}^{[i]}\right]^{-1}\mathbf{u}_{n_s}^{[i]}.\tag{15}$$

From Eq. (14), it is possible to expand the matrix as follows.

$$u(\mathbf{p}_i)=\left[\bar{H}_1^{(\xi)}(\mathbf{p}_i)\,\bar{H}_2^{(\xi)}(\mathbf{p}_i)....\bar{H}_{n_s}^{(\xi)}(\mathbf{p}_i)\right]\boldsymbol{\alpha}^{[i]}\;,\mathbf{p}_i\in\Omega_i.\tag{16}$$

Then Eq.(16) can now be updated using Eq.(15), resulting in the following statement, defined for $\mathbf{p}_i \in \Omega_i$,

$$u(\mathbf{p}_i) = \left\lfloor \bar{H}_1^{(\xi)}(\mathbf{p}_i)\,\bar{H}_2^{(\xi)}(\mathbf{p}_i)\ldots\bar{H}_{n_s}^{(\xi)}(\mathbf{p}_i)\right\rfloor \times \left[\bar{\mathbf{H}}_{n_s\times n_s}^{[i]}\right]^{-1}\mathbf{u}_{n_s}^{[i]}.$$

Which can be rewritten in a clearer format as below

$$u_\xi^{(n)}(\mathbf{p}_i) = \left\lfloor \frac{\partial^{(n)}\bar{H}_1^{(\xi)}(\mathbf{p}_i)}{\partial\xi^{(n)}}\,\frac{\partial^{(n)}\bar{H}_2^{(\xi)}(\mathbf{p}_i)}{\partial\xi^{(n)}}\ldots\frac{\partial^{(n)}\bar{H}_{n_s}^{(\xi)}(\mathbf{p}_i)}{\partial\xi^{(n)}}\right\rfloor \times \left[\bar{\mathbf{H}}_{n_s\times n_s}^{[i]}\right]^{-1}\mathbf{u}_{n_s}^{[i]}.$$

It is then possible to write it as follows.

$$u_\xi^{(n)}(\mathbf{p}_i) = \frac{\partial^{(n)}\vec{\bar{H}}^{(\xi)}(\mathbf{p}_i)}{\partial\xi^{(n)}}\left[\bar{\mathbf{H}}_{n_s\times n_s}^{[i]}\right]^{-1}\mathbf{u}_{n_s}^{[i]}. \tag{17}$$

This completes the derivation of the local indirect meshless method or LiDR formulation.

In the next section, these two local meshless methods are implemented for the target form of PDEs with are described as 'transient convection-diffusion equation' in two dimensions.

# 3 Local (RBF) Collocation Methods for Transient Convection-Diffusion Problems

In this work, the two-dimensional convection-diffusion problems defined on a domain $\Omega$ with a boundary $\Gamma$ that are modeled and governed by the following partial differential equation are to be numerically solved. The governing equation can be expressed as follows:

$$\frac{\partial u}{\partial t} + V_x\frac{\partial u}{\partial x} + V_y\frac{\partial u}{\partial y} = \varepsilon_x\frac{\partial^2 u}{\partial x^2} + \varepsilon_y\frac{\partial^2 u}{\partial y^2} - \beta u + g(x,y,t), \tag{18}$$

where $\varepsilon_x$ and $\varepsilon_y$ are diffusion coefficients and $V_x$ and $V_y$ are convection coefficients. The final two terms $\beta$ and the source term $g(x,y,t)$ are optional and only required in certain cases. The following initial condition is given.

$$u(\mathbf{p},t_0) = \psi_1(\mathbf{p}),\ \mathbf{p}\in\Omega\cup\Gamma,$$

and the boundary condition is expressed as follows.

$$u(\mathbf{p},t) = \psi_2(\mathbf{p},t),\ t\ge t_0\ ,\mathbf{p}\in\Gamma.$$

In this work, the main attention is paid to problems in two dimensions only, hence for $\mathbf{p}\in\Omega\cup\Gamma$,

$$\mathbf{p} = (x,y).$$

The two local collocation meshfree methods, LDR and LiDR, are now implemented to solve Eq.(18). For the transient term, the time discretization for $t\in(t_0, t_0+\Delta t]$ is approximated as follows:

$$\frac{\partial u}{\partial t} \approx \frac{u_t - u_{t-\Delta t}}{\Delta t},\quad \frac{\partial u}{\partial x} \approx \frac{\partial u_{t-\Delta t}}{\partial x},$$
$$\frac{\partial u}{\partial y} \approx \frac{\partial u_{t-\Delta t}}{\partial y},\ \frac{\partial^2 u}{\partial x^2} \approx \frac{\partial^2 u_{t-\Delta t}}{\partial x^2},$$
$$\frac{\partial^2 u}{\partial y^2} \approx \frac{\partial^2 u_{t-\Delta t}}{\partial y^2}.$$

By substituting back into Eq.(18), the following statement can be further proceeded.

$$\begin{aligned} u(\mathbf{p},t) &= u(x,y,t) \\ &\approx u_{t-\Delta t} \\ &+\Delta t\left\{\varepsilon_x\frac{\partial^2 u_{t-\Delta t}}{\partial x^2} + \varepsilon_y\frac{\partial^2 u_{t-\Delta t}}{\partial y^2}\right. \\ &-V_x\frac{\partial u_{t-\Delta t}}{\partial x} - V_y\frac{\partial u_{t-\Delta t}}{\partial y} \\ &\left.-\beta u_{t-\Delta t} + g_{t-\Delta t}(\mathbf{p})\right\} \end{aligned} \tag{19}$$

Substituting Eq. (11) into the above equation provides the following equation for the local direct method (LDR)

$$\begin{aligned} u(\mathbf{p}_i,t) &\approx u_{t-\Delta t}(\mathbf{p}_i) \\ &+\Delta t\left\{\varepsilon_x\frac{\partial^{(2)}\vec{\varphi}(\mathbf{p}_i)}{\partial x^{(2)}}\left[\boldsymbol{\varphi}_{n_s\times n_s}^{[i]}\right]^{-1}\mathbf{u}_{n_s}^{[i]}\right. \\ &+\varepsilon_y\frac{\partial^{(2)}\vec{\varphi}(\mathbf{p}_i)}{\partial y^{(2)}}\left[\boldsymbol{\varphi}_{n_s\times n_s}^{[i]}\right]^{-1}\mathbf{u}_{n_s}^{[i]} \\ &-V_x\frac{\partial\vec{\varphi}(\mathbf{p}_i)}{\partial x}\left[\boldsymbol{\varphi}_{n_s\times n_s}^{[i]}\right]^{-1}\mathbf{u}_{n_s}^{[i]} \\ &-V_y\frac{\partial\vec{\varphi}(\mathbf{p}_i)}{\partial y}\left[\boldsymbol{\varphi}_{n_s\times n_s}^{[i]}\right]^{-1}\mathbf{u}_{n_s}^{[i]} \\ &\left.-\beta u_{t-\Delta t}(\mathbf{p}_i) + g_{t-\Delta t}(\mathbf{p}_i)\right\} \end{aligned} \tag{20}$$

However, with the local indirect method (LiDR), by substituting Eq.(17) into Eq.(19), it yields the following equation.

$$
\begin{aligned}
u\left(\mathbf{p}_i,t\right) &\approx u_{t-\Delta t}\left(\mathbf{p}_i\right) \\
&+\Delta t\Bigg\{\varepsilon_x \frac{\partial^{(2)}\vec{\bar{H}}^{(x)}\left(\mathbf{p}_i\right)}{\partial x^{(2)}}\left[\bar{\mathbf{H}}^{[i]}_{n_s\times n_s}\right]^{-1}\mathbf{u}^{[i]}_{n_s} \\
&+\varepsilon_y \frac{\partial^{(2)}\vec{\bar{H}}^{(y)}\left(\mathbf{p}_i\right)}{\partial y^{(2)}}\left[\bar{\mathbf{H}}^{[i]}_{n_s\times n_s}\right]^{-1}\mathbf{u}^{[i]}_{n_s} \\
&-V_x \frac{\partial\vec{\bar{H}}^{(x)}}{\partial x}\left[\bar{\mathbf{H}}^{[i]}_{n_s\times n_s}\right]^{-1}\mathbf{u}^{[i]}_{n_s} \\
&-V_y \frac{\partial\vec{\bar{H}}^{(y)}\left(\mathbf{p}_i\right)}{\partial y}\left[\bar{\mathbf{H}}^{[i]}_{n_s\times n_s}\right]^{-1}\mathbf{u}^{[i]}_{n_s} \\
&-\beta u_{t-\Delta t}\left(\mathbf{p}_i\right)+g_{t-\Delta t}\left(\mathbf{p}_i\right)\Bigg\}
\end{aligned}
\tag{21}
$$

for $i=1,2,...,N.$

With the implementation of the methods to the convection-diffusion problem described, the next step is to numerically study and analyse the effectiveness of these methods with some benchmark examples.

# 4 Computational Approach and Numerical Results

## 4.1 Step-by-Step Implementation of Meshless Algorithms

**Step 1.** Generate the set of points $\{\mathbf{p}_i\}_{i=1}^{N}$ in the domain $\Omega\cup\Gamma$ and divide it into two non-intersecting sets: the set of interior points on $\Omega$ ($\{\mathbf{p}_i\}_{i=1}^{N}$, containing $N_i$ elements), and the set of boundary points on $\Gamma$ ($\{\mathbf{p}_i\}_{i=N_i+1}^{N_i+N_b}$, containing $N_b$ elements), where $N_i+N_b=N.$

**Step 2.** For each $\mathbf{p}_i=(x_i,y_i)\in\Omega\cup\Gamma$ $i=1,2,...,N$ .

2.1. Create a local influence domain, $\Omega_i=\left\{\mathbf{p}_k^{[i]}\right\}_{k=1}^{n_s}$, which includes the nearest $n_s-1$ points to $\mathbf{p}_i$ and itself.

2.2. Calculate $u_x^{(1)}\left(\mathbf{p}_i\right)$, $u_x^{(2)}\left(\mathbf{p}_i\right)$, $u_y^{(1)}\left(\mathbf{p}_i\right)$ and $u_y^{(2)}\left(\mathbf{p}_i\right)$ for the LDR method, using Eq. (11). However, if the LiDR is taken into account, they will take the form of Eq.(17).

**Step 3.** The following operations are carried out during each iteration.

3.1. Apply the initial condition at $t=t_0$ by specifying $u\left(\mathbf{p}_i\right)=\psi_1(\mathbf{p}_i)$, for $i=1,2,...,N.$

3.2. Apply the boundary conditions $u\left(\mathbf{p}_i\right)=\psi_2(\mathbf{p}_i,t)$, for $i=N_i+1,N_i+2,...,N_i+N_b$ , and $t\geq t_0$.

3.3. Approximate the value of $u\left(\mathbf{p}_i,t\right)$, for $i=1,2,...,N$ , use Eq.(20) for the LDR method or Eq.(21) for the LiDR method.

3.4. For the next time step, setting $u\left(\mathbf{p}_i,t-\Delta t\right)\leftarrow u\left(\mathbf{p}_i,t\right)$ for $i=1,2,...,N,$ and $t-\Delta t\leftarrow t.$

**Step 4.** The calculation comes to an end at $t=t_{end}$, the approximate values of $u\left(\mathbf{p}_i,t_{end}\right)$ is now obtained.

**Step 5.** Validate the approximate solution using the criteria provided in the next section.

## 4.2 Validation Criteria and Error Analysis

Regarding the quality of the solutions obtained from the two proposed meshless methods, the first criterion is on the accuracy and two error measurements (defined for a location $\mathbf{p}_i$ at a time $t$) are employed.

1. The relative error norm, defined as:

$$
Rel.Err=\sqrt{\frac{\sum_{i=1}^{N}(u^{exact}(\mathbf{p}_i,t)-u^{num}(\mathbf{p}_i,t))^2}{\sum_{i=1}^{N}\left(u^{exact}(\mathbf{p}_i,t)\right)^2}}.
$$

2. The root mean square error, defined as

$$
\mathrm{RMSE}=\sqrt{\frac{\sum_{i=1}^{N}\left(u^{exact}(\mathbf{p}_i,t)-u^{num}(\mathbf{p}_i,t)\right)^2}{N}}
$$

Here $u^{exact}(\mathbf{p}_i,t)$ is the exact solution and $u^{exact}(\mathbf{p}_i,t)$ it is the numerical one, both measured at the node $\mathbf{p}_i$ and at a target/final time $t$. Additionally, for comparative analysis, the sensitivity to multi-quadratic shape parameters, the size of the subdomain, and convection coefficients are observed and compared.

## 4.3 Numerical Benchmarks and Performance Evaluation

### 4.3.1 Test Case 1: Diffusion Equation Validation

Let's first examine a dimensionless diffusion equation, as explored in [43], to evaluate the two meshless methods

$$
\frac{\partial u}{\partial t}=\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}.
$$

The assumption is that the problem domain is square $\mathbf{p} \in [-0.5, 0.5] \times [-0.5, 0.5]$, the initial condition is $u(\mathbf{p}, t_0) = 1$ and the Dirichlet jump boundary condition is $u(\mathbf{p}, t) = 0.$ . The exact solution is given in [44], expressed as follows

$$u(\mathbf{p}, t) = \frac{16}{\pi^2} u_{exact}(x, t) u_{exact}(y, t),$$

with $\xi = x$ or $y$. The following is obtained.

$$u_{exact}(\xi, t) = \sum_{i=0}^{\infty} \frac{(-1)^i \mathrm{e}^{\left[-(2i+1)^2 \pi^2 t\right]} \cos\left[(2i+1)\pi\xi\right]}{2i+1}.$$

Our investigation starts with paying attention to the influence of the multiquadric shape parameter (c) on the numerical performance of the two local methods under study. To receive sufficient information from such inquiries, the parameter values in the range (c∈(0,10]) were taken into consideration, where the overall performance was measured carefully using the Root Mean Square Error (RMSE). The final target time was set to (t = 0.01) for all simulations with time increment (Δt = 0.0001), executing on two subdomain sizes $n_s = 5$ and $n_s = 9$. Figure 1 in the Appendix depicts the numerical results, which reveal that promising results are found to have been produced within the specific range of such parameter (c ≈ [0.5,3.0]). Outside this optimal range, the numerical results noticeably suffered in terms of precision. It really stresses the parameter's delicate character and the need to choose it very wisely.

With the discovery of optimal shape parameters, we went on to examine the accuracy and stability of the methods. This was done by comparing their approximate solutions to the analytical ones documented in [44]. It is found, as illustrated in Figure 2 (Appendix), that the numerical solutions provided by the two local methods are in reasonably good agreement with the exact ones, with only small discrepancies detected in some areas.

Obtained from this first investigation, the slight performance advantages of the LDR were detected, especially when it comes to numerical accuracy and solution consistency across the domain. The key finding of this test emphasized the important impact of the shape parameter on the overall performance of both local methods, highlighting how a poorly chosen value can significantly degrade the quality of the solution. These informative insights provide useful guidance for further studies, which might involve more challenging scenarios, higher Peclet numbers, and increased complexity of the problems at hand, as well as more refined parameter tuning strategies.

### 4.3.2 Test Case 2: Convection-Diffusion Scenario with Constant Parameters

In this numerical experiment, we examine the performance of the LDR and LiDR methods in solving a convection–diffusion problem characterized by constant convection and diffusion coefficients, as previously studied in [34], [45]. Specifically, constant values of convection coefficients $Vx = Vy = 0.8$ and diffusion coefficients $\varepsilon_x = \varepsilon_y = 0.01$ and with $g(x) = \beta = 0$ were employed. The exact analytical solution, clearly defined over a square computational domain, $(x, y) \in [0.5, 2] \times [0.5, 2]$, served as the reference for accuracy assessment and was also used to set both initial and boundary conditions.

$$u(p,t) = \frac{1}{4t+1} \times \exp\left( -\left[ \frac{(x - 0.8t - 0.5)^2}{0.01(4t+1)} + \frac{(y - 0.8t - 0.5)^2}{0.01(4t+1)} \right] \right). \quad (22)$$

Following the insights gained from Test Case 1 regarding the critical role of the MQ shape parameter (c), this scenario extended the investigation into additional parameters crucial for meshless methods: the local subdomain size $n_s$ and time step size $\Delta t$. A series of systematic numerical experiments was conducted with three distinct subdomain sizes ($n_s = 5, 9, 13$), a discretization grid of 15×15, and a sufficiently small time increment ($\Delta t = 0.0001$) to ensure temporal accuracy.

Figure 3 in the Appendix clearly depicts the progression of relative errors (Rel.Err) over simulation time ($t \in (0,1]$). From this figure, it is observed that accuracy generally deteriorates gradually as simulation time increases for both methods. However, a notable contrast between the two methods emerges, with LDR consistently outperforming LiDR across all evaluated times. Remarkably, LDR not only maintained but improved its accuracy over longer simulation periods, achieving its lowest relative error (7.89E-02) at the final simulation time ($t = 1.0$ with optimal parameters $n_s = 9$ and $c = 0.33$). In contrast, LiDR exhibited the highest relative error ($5.18$ at $t = 1.0$, with $n_s = 5$ and $c = 0.93$), indicating that the accuracy of LiDR is more sensitive to subdomain size and shape parameter selection, particularly for extended simulations.

Further visual insights into the numerical performance of both methods are presented in Figure 4 (Appendix), showing solution profiles at representative simulation times ($t = 0.1,\ 0.5,\ 1.0$). These profiles clearly illustrate that LDR consistently provides more accurate approximations to the analytical solution, particularly in capturing the spatial-temporal dynamics of the convection–diffusion process. LiDR, while still adequately representing the general solution behavior, shows less precision, particularly evident at later simulation stages and larger subdomain sizes.

Summarizing these results, several critical conclusions and practical recommendations emerge from this scenario. First, the results underscore the superiority of the LDR method in capturing the complex dynamics of convection-diffusion processes, especially over longer simulation durations. Secondly, they highlight the paramount importance of appropriate parameter selection (both the shape parameter cc and local subdomain size $n_s$). While smaller subdomain sizes generally offer better computational efficiency, an optimal balance is required since excessively small or large values may negatively impact accuracy and stability. Lastly, these findings reaffirm that adaptive or dynamic parameter adjustment strategies could be highly beneficial in practical, real-world implementations of these local meshless methods.

### 4.3.3 Test Case 3: Convection-Dominant Cases

The governing equation in this example is the same as that investigated in Test Case 2, except that $\beta = 0$ with the following source term, [46].

$$g(\mathbf{p},t) = \cos\left(\varepsilon_x x + \varepsilon_y y\right)\left(\varepsilon_x V_x + \varepsilon_y V_y\right) \times \exp\left(-t\left[\left\{\varepsilon_x\right\}^3 + \left\{\varepsilon_y\right\}^3\right]\right). \tag{23}$$

All boundary conditions are taken from the analytical solution, which is given as follows,

$$u(\mathbf{p},t) = \sin\left(\varepsilon_x x + \varepsilon_y y\right) \times \exp\left(-t\left[\left\{\varepsilon_x\right\}^3 + \left\{\varepsilon_y\right\}^3\right]\right).$$

The problem is defined on the computational domain $\Omega \cup \Gamma = [0,2]\times[0,2]$.

With one of the ultimate goals of this work is to evaluate how well the local methods can handle situations where the convection force is dominating, known as convection-dominated phenomena. This kind of mechanism is well known to be challenging for many traditional numerical methods, including those mentioned in Section 1. In particular, in this third case, the problem is set up similarly to that of Test Case 2 but now includes a carefully designed source term to reinforce the convection-dominated behavior and to better assess the robustness of the two local methods under such demanding conditions.

To achieve such discovery, the numerical experiments were set up at four values of the convection coefficients: ($V = 100,\ 200,\ 300,\ 400$), while the diffusion coefficients were fixed at $\varepsilon_x = \varepsilon_y = 1$. Three subdomain sizes were taken into consideration for this test, ($n_s = 5,9,13$), where all the approximate solutions were measured using the Relative Error (Rel.Err) and the computation time (CTm). As can be seen in Table 1 and Table 2 in the Appendix, the two local methods were found to be able to maintain reasonable accuracy for most tested conditions, with only slight decreases in accuracy as the convection force strengthened. With this said, it has to be noticed that LDR is observed to have produced more consistency in accuracy over LiDR. Furthermore, it was observed that the computational time increased slightly with rising convection coefficients, as expected, and was also influenced by the size of the local subdomain.

To elaborate this further from a theoretical point of view, the LDR method involves slightly larger local systems than LiDR, because the unknowns are determined directly from the differential operator discretization. This suggests that the per-subdomain computational cost of LDR may be higher. However, in our implementation, the total CPU time is often lower for LDR (see Table 2 in the Appendix), because LDR is more stable and requires fewer time steps/iterations to achieve the same error level. In contrast, LiDR may need smaller time steps or more iterations due to its higher sensitivity to the choice of $c$ and $n_s$.

For both LDR and LiDR, the dominant cost is solving local linear systems of size. $n_s \times n_s$on each subdomain. If a direct dense solver is used, the cost per subdomain is $O(n_s^3)$. The number of subdomains is approximately equal to the number of nodes, so the total complexity scales as $O(Nn_s^3)$ for both methods. The main difference is that in LDR, the system is assembled directly from the differential operator, whereas in LiDR, an additional reconstruction step is needed, which can increase the constant factors in the total cost

To further investigate the robustness of the methods under even more difficult conditions, the work also considered two additional values of the convection coefficients: $V = 500,\ 600$. As shown in Figure 5 and Figure 6 (Appendix), both of these methods were still capable of capturing the crucial phenomena, provided that the optimal parameters were chosen, which include the multiquadric shape parameter, the subdomain size, and the time step size.

Test Case 3 finished our study, giving us valuable insights into how the LDR and LiDR meshless methods perform when conditions are difficult and convection is very strong. Both methods proved to be stable and gave reasonable accuracy. However, LDR consistently showed better accuracy and overall robustness, no matter how strong the convection was or how the subdomains were configured. For this reason, in practical situations where convection is dominant, we recommend using the LDR method, provided you carefully choose or adaptively adjust the parameters. These findings also confirm once more that using adaptive strategies to pick parameters is very important for making meshless methods more reliable and practical for use in engineering calculations.

#### 4.3.4 Test Case 4: Advection-Diffusion Example with Analytical Solution

To conclude, let us consider the linear advection-diffusion equation in its specific form, [47]. The equation is expressed as follows.

$$\frac{\partial u}{\partial t} = \varepsilon_x \frac{\partial^2 u}{\partial x^2} + \varepsilon_y \frac{\partial^2 u}{\partial y^2} + V_x \frac{\partial u}{\partial x} + V_y \frac{\partial u}{\partial y}, \qquad (24)$$

where each of $V_x = V_y = V$, $\varepsilon_x = \varepsilon_y = \varepsilon$ is a constant. The problem domain is assumed to be a square domain $\mathbf{p} \in [0,1] \times [0,1]$. The initial condition is $u(\mathbf{p}, t_0) = \exp(-C_x x) + \exp(-C_y y)$ and the Dirichlet boundary conditions are defined as follows:

$$u(\mathbf{p}, t) = \exp(t)\left(\exp(-C_x x) + \exp(-C_y y)\right),$$

where $C_x = \left(V_x + \sqrt{(V_x)^2 + 4\varepsilon_x}\right) / 2\varepsilon_x$ and $C_y = \left(V_y + \sqrt{(V_y)^2 + 4\varepsilon_y}\right) / 2\varepsilon_y$.

For the last experiment, we investigated the problem where the related values were set to $Vx = Vy = 1$ and $\varepsilon_x = \varepsilon_y = 0.8$ and in this case the analytical solutions were published by [47], against which we compared the obtained results. Table 3 and Table 4 in the Appendix show the main results, where it can be seen that both of the local methods manage to produce reliable accuracy when compared to both the analytical solutions and those documented by other works in the literature, such as the Dual Reciprocity Boundary Element Method (DRBEM), the Inverse Multiquadric (IMQ) method, and Thin Plate Splines (TPS).

A detailed study of how the shape parameter($c$) affects the numerical accuracy of both meshless methods was performed, as shown in Figure 7 (Appendix). This figure clearly indicates that the accuracy of LDR and LiDR stays stable only within a certain optimal interval for this shape parameter. This result strongly supports our previous conclusion that the shape parameter must be chosen with great care to ensure both accuracy and stability in the numbers. Furthermore, the robustness of the methods was demonstrated in even more challenging scenarios with stronger convection forces, presented clearly in Figure 8 and Figure 9 (Appendix) through detailed solution surface plots and contour profiles. These plots confirm that both methods can manage complex advection–diffusion problems, provided that all parameters, the shape parameter, subdomain size, and time increment, are selected optimally and correctly.

In summary, Test Case 4 successfully confirms that both the LDR and LiDR methods are effective and highly accurate for solving advection–diffusion problems that have analytical solutions for comparison. The detailed numerical experiments highlighted important insights about parameter sensitivity, reinforced the recommendation to use adaptive parameter selection strategies, and showed the robust potential of these methods. This confirms they are reliable numerical tools for practical applications, especially in engineering and environmental modeling.

Further tests regarding convergence and stability were done by changing the density of the nodes and varying the Peclet number. These experiments confirmed that both the LDR and LiDR methods show consistent convergence patterns. However, LDR demonstrated greater robustness when the conditions were dominated by convection. For the sake of brevity, we have not included the complete numerical results in this section, but they are available to be provided as supplementary material if requested.

## 5 Conclusion

This research conducted a comprehensive numerical evaluation and comparative analysis of the Local Direct Radial Basis Function (LDR) and the Local

Indirect Radial Basis Function (LiDR) meshless methods for solving various convection–diffusion problems. Across 4 benchmark tests ranging from pure diffusion to conditions dominated by strong convection, both methods proved to possess robust and reliable numerical capabilities. However, the LDR method consistently provided superior accuracy and stability, which was most clear in cases with strong convection, longer simulation times, and varying node arrangements. The LiDR method also offered generally satisfactory numerical approximations and remained stable across the tests, but it typically showed lower accuracy compared to LDR, particularly when convection was very high.

Crucially, the study emphasized a significant sensitivity to key numerical parameters, especially the multiquadric shape parameter, c, and the local subdomain size $n_s$. The best numerical performance was tightly linked to a careful tuning of parameters, making clear the necessity for well-informed or adaptive strategies for selection. Given the slightly higher computational resources for LDR, its better accuracy and robustness make this exchange worthwhile, especially in tasks where high precision is critical, such as environmental modeling or engineering design. Therefore, LDR emerges as the recommended method for practical high-precision applications, while LiDR is still useful for less demanding or quick approximation work. Future efforts should be directed to developing intelligent adaptive algorithms for automatic parameter selection, and also extending these frameworks to three-dimensional problems and integrating them with advanced computational methods like parallel or GPU-based structures.

*Acknowledgement:*
The corresponding author would like to express his sincere appreciation to Department of Mathematics, Faculty of Science, Ubon Ratchathani Rajabhat University for its kind support.

*References:*

[1] E. Goncalvès, and D. Zeidan, Numerical simulation of unsteady cavitation in liquid hydrogen flows, Int. J. Eng. Syst. Model. Simul. 9(1),41-52(2017), https://www.inderscienceonline.com/doi/abs/10.1504/IJESMS.2017.081736

[2] K.M. Agbavon, A.R. Appadu, and M. Khumalo, On the numerical solution of Fisher's equation with coefficient of diffusion term much smaller than coefficient of reaction term, Adv. Differ. Equ. 2019(1), 1-33 (2019), https://doi.org/10.1186/s13662-019-2080-x

[3] C. Franke, and R. Schaback, Convergence order estimates of meshless collocation methods using radial basis functions, Adv. Comput. Math. 8, 381-399 (1998), https://doi.org/10.1023/A:1018916902176

[4] J. Lin, S.Y. Reutskiy, and J. Lu, A novel meshless method for fully nonlinear advection–diffusion-reaction problems to model transfer in anisotropic media, Appl. Math. Comput. 339, 459-476 (2018), https://doi.org/10.1016/j.amc.2018.07.045

[5] F. Wang, W. Chen, A. Tadeu, and C.G. Correia, Singular boundary method for transient convection–diffusion problems with time-dependent fundamental solution, Int. J. Heat Mass Transf. 114, 1126-1134 (2017), https://doi.org/10.1016/j.ijheatmasstransfer.2017.07.007

[6] Z.J. Fu, Q. Xi, W. Chen, and A.H.D. Cheng, A boundary-type meshless solver for transient heat conduction analysis of slender functionally graded materials with exponential variations, Comput. Math. Appl. 76(4), 760-773 (2018), https://doi.org/10.1016/j.camwa.2018.05.017

[7] B.M. Chen, and H.V. Kojouharov, Non-standard numerical methods applied to subsurface biobarrier formation models in porous media, Bull. Math. Biol. 61(4), 779-798 (1999), https://doi.org/10.1006/bulm.1999.0113

[8] Z.J. Zhou, and N.N. Yan, A survey of numerical methods for convection–diffusion optimal control problems, J. Numer. Math. 22(1),61-85 (2014),https://doi.org/10.1515/jnum-2014-0003

[9] A. Fallahzadeh, and K. Shakibi, A method to solve Convection-Diffusion equation based on homotopy analysis method, J. Interpolat. Approx. Sci. Comput. 2015(1), 1-8 (2015), https://d-nb.info/1078099669/34 (Accessed Date 24 Sep 2025)

[10] A. Safdari-Vaighani, A. Heryudono, and E. Larsson, A radial basis function partition of unity collocation method for convection–diffusion equations arising in financial applications, J. Sci. Comput. 64(2), 341-367 (2015),https://doi.org/10.1007/s10915-014-9935-9

[11] S. Sengupta, N.A. Sreejith, P. Mohanamuraly, G. Staffelbach, and L. Gicquel, Global spectral analysis of the Lax–Wendroff-central difference scheme applied to Convection–

Diffusion equation, Comput. Fluids 242, 1-22 (2022), https://doi.org/10.1016/j.compfluid.2022.105508

[12] S. Zhai, L. Qian, D. Gui, and X. Feng, A block-centered characteristic finite difference method for convection-dominated diffusion equation, Int. Commun. Heat Mass Transf. 61, 1-7 (2015), https://doi.org/10.1016/j.icheatmasstransfer.2014.11.003

[13] D. Funaro, A new scheme for the approximation of advection-diffusion equations by collocation, SIAM J. Numer. Anal. 30(6), 1664-1676 (1993), https://doi.org/10.1137/0730085

[14] Z. Si, X. Feng, and A. Abduwali, The semi-discrete streamline diffusion finite element method for time-dependented convection–diffusion problems, Appl. Math. Comput. 202(2), 771-779 (2008), https://doi.org/10.1016/j.amc.2008.03.021

[15] H. Rui, and M. Tabata, A second-order characteristic finite element scheme for convection-diffusion problems, Numer. Math. 92(1), 161-177 (2002), https://doi.org/10.1007/s002110100364

[16] S. Wang, Solving convection-dominated anisotropic diffusion equations by an exponentially fitted finite volume method, Comput. Math. Appl. 44(8-9), 1249-1265 (2002), https://doi.org/10.1016/S0898-1221(02)00230-4

[17] A. Russo, Streamline-upwind Petrov/Galerkin method (SUPG) vs residual-free bubbles (RFB), Comput. Methods Appl. Mech. Eng. 195(13-16), 1608-1620 (2006), https://doi.org/10.1016/j.cma.2005.05.031

[18] P. Knobloch, A generalization of the local projection stabilization for convection-diffusion-reaction equations, SIAM J. Math. Anal. 48(2), 659-680 (2010), https://doi.org/10.1137/090767807

[19] S.H. Zhang, and W.Q. Wang, A stencil of the finite-difference method for the 2D convection diffusion equation and its new iterative scheme, Int. J. Comput. Math. 87(11), 2588-2600 (2010), https://doi.org/10.1080/00207160802691637

[20] A. Falini, and M.L. Sampoli, Adaptive Refinement in Advection–Diffusion Problems by Anomaly Detection: A Numerical Study, Algorithms, 14(328), 1-20 (2021), https://doi.org/10.3390/a14110328

[21] A. Bokota, and S. Iskierka, An analysis of the diffusion-convection problem by the boundary element method, Eng. Anal. Bound. Elem. 15(3), 267-275 (1995), https://doi.org/10.1016/0955-7997(95)00031-I

[22] K. Chanthawara and S. Kaennakham, A Hybrid Shapeless Radial Basis Function Applied With the Dual Reciprocity Boundary Element Method, WSEAS Trans. Math. 20, 159-170 (2021), https://doi.org/10.37394/23206.2021.20.17(Accessed Date 12 Dec 2025)

[23] X.Y. Liu, A. Karageorghis, and C.S. Chen, A Kansa-Radial Basis Function Method for Elliptic Boundary Value Problems in Annular Domains, J. Sci. Comput. 65, 1240-1269 (2015), https://doi.org/10.1007/s10915-015-0009-4

[24] E.J. Kansa, H. Power, G.E. Fasshauer, and L. Ling, A volumetric integral radial basis function method for time-dependent partial differential equations: I. Formulation, Eng. Anal. Bound. Elem. 28, 1191–1206 (2004), https://doi.org/10.1016/j.enganabound.2004.01.004

[25] E.J. Kansa, Multiquadrics - a scattered data approximation scheme with applications to computational fluid dynamics – II, Comput. Math. Appl. 19(8-9), 147-161(1990), https://doi.org/10.1016/0898-1221(90)90271-K

[26] S. Kaennakham, and K. Chanthawara, Numerical solution to nonlinear transient coupled-PDE by the modified multiquadric meshfree method, Rom. J. Phys. 66(108), 1-14 (2021). https://rjp.nipne.ro/2021_66_5-6.html

[27] A. Fallah, E. Jabbari, and R. Babaee, Development of the Kansa method for solving seepage problems using a new algorithm for the shape parameter optimization, Comput. Math. Appl. 77, 815-829 (2019), https://doi.org/10.1016/j.camwa.2018.10.021

[28] R. Verma, and S. Kumar, Computational study on constant and sinusoidal heating of skin tissue using radial basis functions, Comput. Biol. Med. 121, 1-10 (2020), https://doi.org/10.1016/j.compbiomed.2020.103808

[29] W. Chen, Y. Hong, and J. Lin, The sample solution approach for determination of the optimal shape parameter in the multiquadric function of the Kansa method, Comput. Math. Appl. 75(8), 2942–2954 (2018), https://doi.org/10.1016/j.camwa.2018.01.023

[30] J. Li, S. Zhai, Z. Weng, and X. Feng, H -adaptive RBF-FD method for the high-dimensional convection-diffusion equation, Int. Commun. Heat Mass Transf. 89, 139-146 (2017), https://doi.org/10.1016/j.icheatmasstransfer.2017.06.001

[31] N. Li, H. Su, D. Gui, and X. Feng, Multiquadric RBF-FD method for the convection-dominated diffusion problems base on Shishkin nodes, Int. J. Heat Mass Transf. 118, 734-745 (2018), https://doi.org/10.1016/j.ijheatmasstransfer.2017.11.011

[32] N. Chuathong, and S. Kaennakham, Numerical solution to coupled burgers' equations by gaussian-based hermite collocation scheme, J. Appl. Math. 2018, 1-18 (2018), https://doi.org/10.1155/2018/3416860

[33] S. Kaennakham, and N. Chuathong, An automatic node-adaptive scheme applied with a RBF-collocation meshless method, Appl. Math. Comput. 348, 102-125 (2019), https://doi.org/10.1016/j.amc.2018.11.066

[34] S. Kaennakham, and N. Chuathong, Numerical simulation of convection diffusion phenomena by four inverse-quadratic-RBF domain-meshfree schemes, Int. J. Multiphysics, 13(1), 1-30 (2019), https://www.themultiphysicsjournal.com/index.php/ijm/article/view/13-1-1

[35] Y. Wu, and C. Shu, Development of RBF-DQ method for derivative approximation and its application to simulate natural convection in concentric annuli, Comput. Mech. 29, 477–485 (2002), https://doi.org/10.1007/s00466-002-0357-4

[36] N. Mai-Duy, and T. Tran-Cong, An efficient indirect RBFN- based method for numerical solution of PDEs, Numer. Methods Partial Differ. Equ. 21(4), 770-790 (2005), https://doi.org/10.1002/num.20062

[37] Y. Wu, and C. Shu, Development of RBF-DQ method for derivative approximation and its application to simulate natural convection in concentric annuli, Comput. Mech. 29, 477–485 (2002), https://doi.org/10.1007/s00466-002-0357-4

[38] C.S. Chen, M.A. Jankowska, and A. Karageorghis, RBF–DQ algorithms for elliptic problems in axisymmetric domains, Numer. Algorithms 1-31 (2022), https://doi.org/10.1007/s11075-021-01105-w

[39] F. Motaman, G.R. Rakhshandehroo, N.R. Hashemi, and M. Niazkar, Application of RBF-DQ method to time-dependent analysis of unsaturated seepage, Transp. Porous Media 125, 543-564 (2018), https://doi.org/10.1007/s11242-018-1138-7

[40] K. Parand, and S. Hashemi, RBF-DQ method for solving non-linear differential equations of Lane-Emden type, Ain Shams Eng. J. 9(4), 615-629 (2018), https://doi.org/10.1016/j.asej.2016.03.010

[41] C. Shu, and Y.L. Wu, Integrated radial basis functions- based differential quadrature method and its performance, Int. J. Numer. Methods Fluids, 53(6) 969-984 (2007), https://doi.org/10.1002/fld.1315

[42] A. Ebrahimijahan, M. Dehghan, and M. Abbaszadeh, Numerical simulation of shallow water waves based on generalized equal width (GEW) equation by compact local integrated radial basis function method combined with adaptive residual subsampling technique, Nonlinear Dyn. 105(4), 3359-3391 (2021), https://doi.org/10.1007/s11071-021-06733-4

[43] G. Yao, B. Šarler, and C.S. Chen, A comparison of three explicit local meshless methods using radial basis functions, Eng. Anal. Bound. Elem. 35(3), 600-609 (2011), https://doi.org/10.1016/j.enganabound.2010.06.022

[44] H.S. Carslaw and J.C. Jaeger, Condution of heat in solids, (Oxford University Press, United Kingdom, 1986), https://doi.org/10.1115/1.3225900

[45] H.W. Sun, and L.Z. Li, A CCD–ADI method for unsteady convection–diffusion equations, Comput. Phys. Commun. 185, 790-797 (2014), https://doi.org/10.1016/j.cpc.2013.11.009

[46] B. Nayroles, G. Touzot, and P. Villon, Generalizing the finite element method: diffuse approximation and diffuse elements, Comput. Mech. 10, 307-318 (1992), https://doi.org/10.1007/BF00364252

[47] M. Dehghan, A. Mohebbi, High-order compact boundary value method for the solution of unsteady convection-diffusion problems, Math. Comput. Simul. 79, 683–699(2008), https://ideas.repec.org/a/eee/matcom/v79y2008i3p683-699.html

# APPENDIX

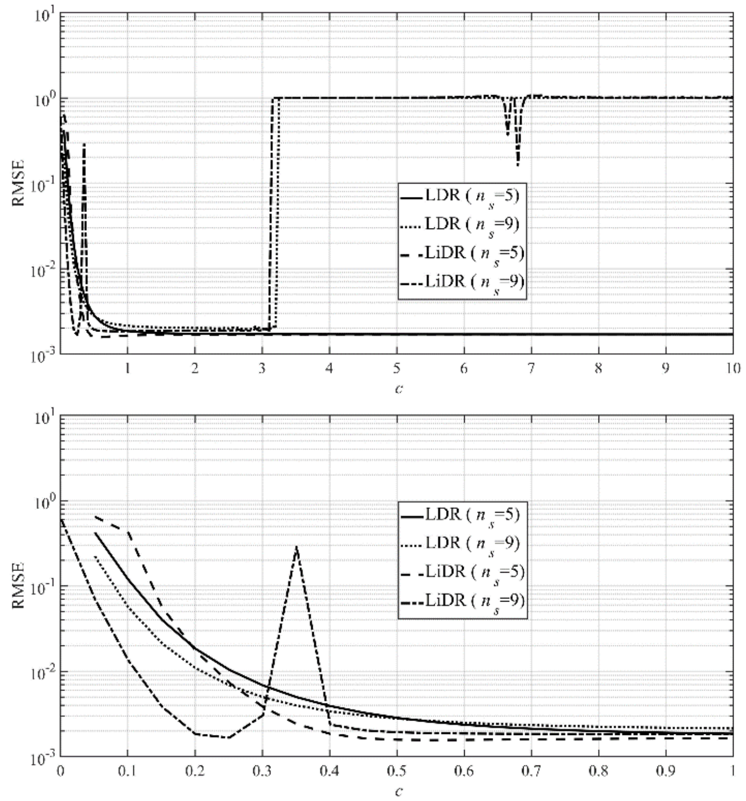


Fig. 1: Root mean square error (RMSE) measured at different shape values $(c)$ produced by the two local meshless methods for $n_s = 5$ and 9 at $t = 0.01$ with $\Delta t = 0.0001$.

*Source: created by the authors*

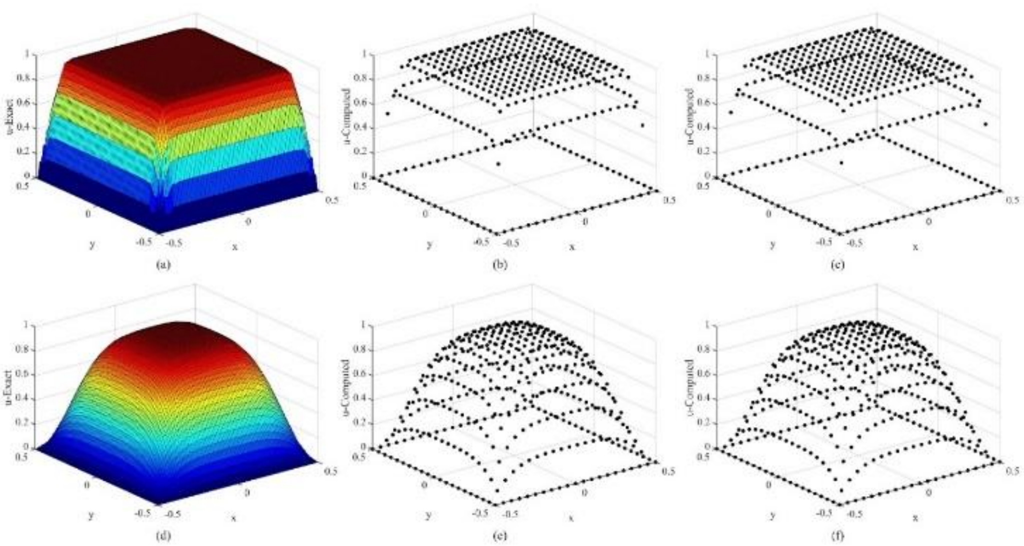

Fig. 2: Solution profiles using $\Delta t = 0.0001$, $N = 21 \times 21$ and $n_s = 5$: First row at $t = 0.001$; (a) the exact, (b) LDR with $c = 9.901$ and (c) LiDR with $c = 0.201$. Second row at $t = 0.01$ (d) the exact, (e) LDR with $c = 9.901$ and (f) LiDR with $c = 0.551$.

*Source: created by the authors*

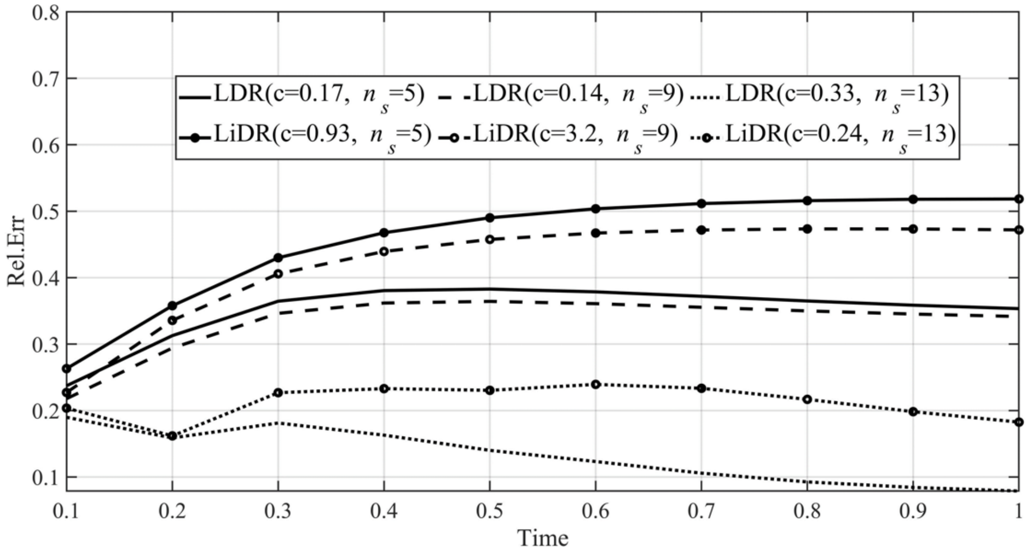


Fig. 3: Relative error progression measured at different target times for the two local meshless methods at three levels of the nearest points, $n_s = 5$, 9 and 13 with $N = 15 \times 15$

*Source: created by the authors*

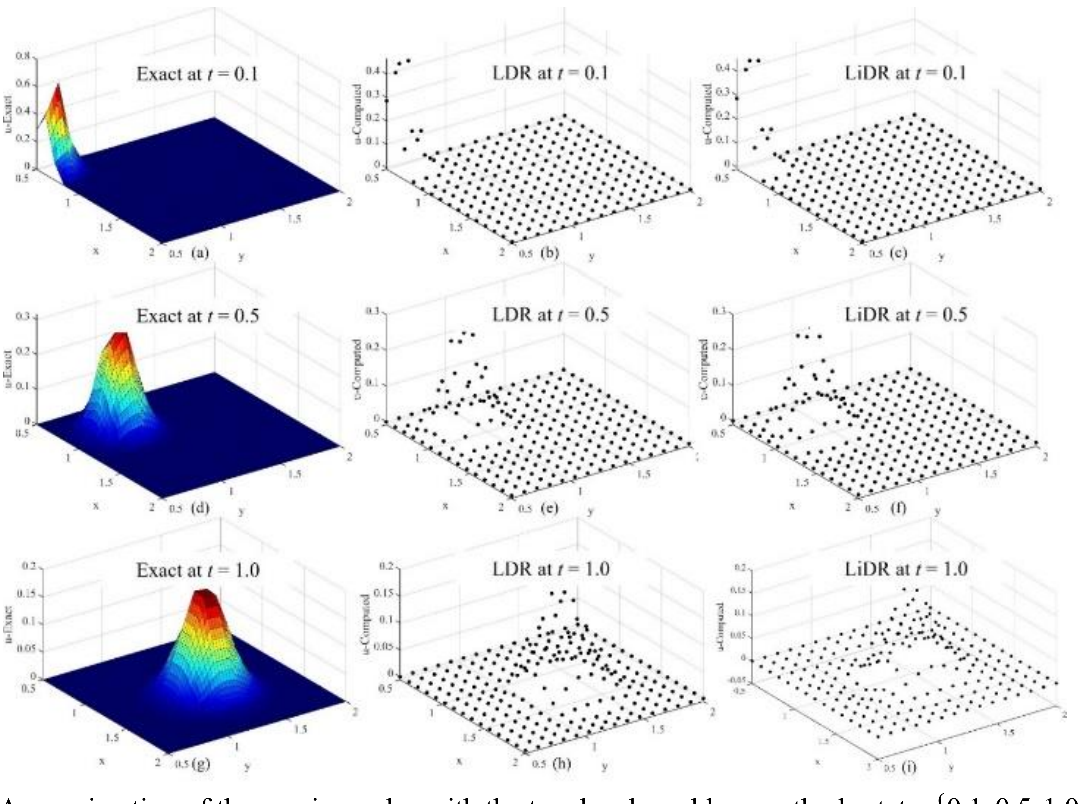


Fig. 4: Approximation of the moving pulse with the two local meshless methods at $t = \{0.1, 0.5, 1.0\}$ using $\Delta t = 0.0001$, $N = 15 \times 15$ and $n_s = 13$; (a-d-g) Exact (b-e-h) LDR with $c = 0.33$ and (c-f-i) LiDR with $c = 0.24$

*Source: created by the authors*

Table 1. Relative error (*Rel.Err*) revealed from each RBF at different values $V_x = V_y = V$, $\varepsilon_x = \varepsilon_y = 1$ and nearest points refinement ($n_s$) at $t = 0.01$, $\Delta t = 0.0001$, and $N = 10\times10$

| *Rel.Err* | $n_s = 5$ | | $n_s = 9$ | | $n_s = 13$ | |
|---|---|---|---|---|---|---|
| | LDR $(c=1.69)$ | LiDR $(c=10.09)$ | LDR $(c=2.97)$ | LiDR $(c=3.21)$ | LDR $(c=9.83)$ | LiDR $(c=6.54)$ |
| $V=100$ | 1.74E-04 | 7.71E-03 | 5.25E-05 | 2.00E-03 | 1.06E-04 | 1.65E-04 |
| $V=200$ | 1.16E-04 | 8.25E-03 | 3.43E-05 | 1.50E-03 | 1.22E-04 | 1.91E-04 |
| $V=300$ | 8.05E-05 | 8.60E-03 | 2.02E-05 | 1.53E-03 | 1.51E-04 | 2.63E-04 |
| $V=400$ | 5.05E-05 | 8.69E-03 | 2.48E-05 | 1.54E-03 | 1.36E-03 | 4.36E-03 |

*Source: created by the authors*

Table 2. Computational time (CTm.) (in seconds) revealed from each RBF at different values $V_x = V_y = V$, $\varepsilon_x = \varepsilon_y = 1$ and nearest points refinement ($n_s$) at $t = 0.01$, $\Delta t = 0.0001$ and $N = 10\times10$

| CTm. | $n_s = 5$ | | $n_s = 9$ | | $n_s = 13$ | |
|---|---|---|---|---|---|---|
| | LDR $(c=1.69)$ | LiDR $(c=10.09)$ | LDR $(c=2.97)$ | LiDR $(c=3.21)$ | LDR $(c=9.83)$ | LiDR $(c=6.54)$ |
| $V=100$ | 6.00E-01 | 1.11E+00 | 1.49E+00 | 2.63E+00 | 3.07E+00 | 5.89E+00 |
| $V=200$ | 6.82E-01 | 1.11E+00 | 1.47E+00 | 2.65E+00 | 3.07E+00 | 5.96E+00 |
| $V=300$ | 6.72E-01 | 1.11E+00 | 1.45E+00 | 2.64E+00 | 3.01E+00 | 5.89E+00 |
| $V=400$ | 6.95E-01 | 1.28E+00 | 1.73E+00 | 3.23E+00 | 3.05E+00 | 7.36E+00 |

*Source: created by the authors*

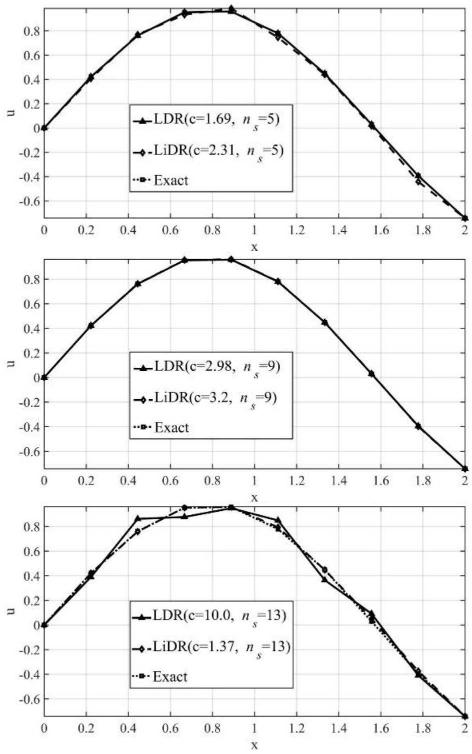


Fig. 5: Solution comparison computed on the $x = y$ straight line using $V_x = V_y = 500, \varepsilon_x = \varepsilon_y = 1$ recorded at $t = 0.01$ using $\Delta t = 0.0001$ and $N = 10\times10$ with three levels of nearest point densities

*Source: created by the authors*

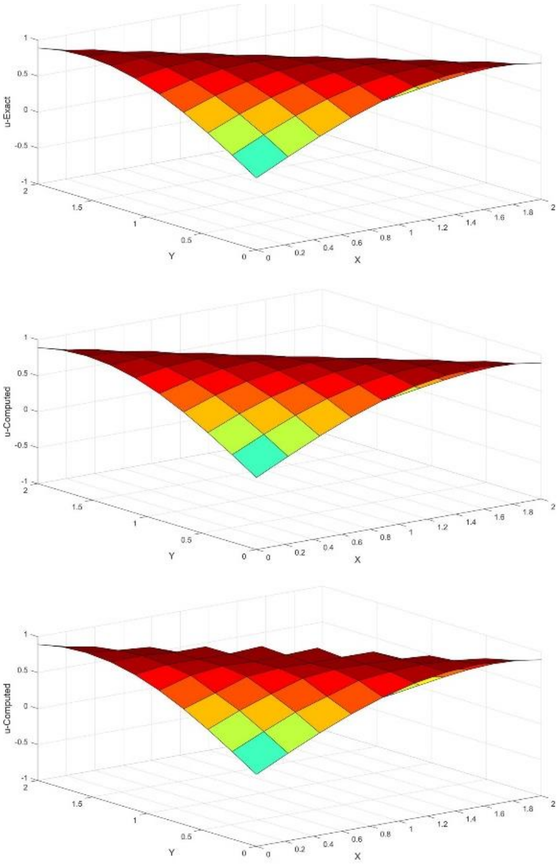


Fig. 6: Solution surface plots computed using $V_x = V_y = 600$, $\varepsilon_x = \varepsilon_y = 1$ at $t = 0.01$ and $\Delta t = 0.0001$ with $N = 10 \times 10$ and $n_s = 5$; (top) exact, (middle) LDR, and (bottom) LiDR

*Source: created by the authors*

Table 3. Numerical solutions produced by each method (this work and together with some from literature) compared to the exacts using $\Delta t = 0.001$ for $V_x = V_y = 1$, $\varepsilon_x = \varepsilon_y = 0.8$ and $t = 0.05$ using $n_s = 13$ and $N = 11 \times 11$

| Internal Node | This work | | From literature | | | Exact |
|---|---|---|---|---|---|---|
| | LDR ($c = 5.00$) | LiDR ($c = 4.26$) | DRBEM [34] | IMQ ($c = 0.104$) [47] | TPS [47] | |
| (0.1,0.1) | 1.737772 | 1.737767 | - | 1.732981 | 1.737215 | 1.737696 |
| (0.5,0.1) | 1.27429 | 1.274270 | 1.26601 | 1.265541 | 1.273859 | 1.274227 |
| (0.9,0.1) | 1.058041 | 1.058042 | 1.04109 | 1.053769 | 1.058015 | 1.057985 |
| (0.3,0.3) | 1.186976 | 1.186965 | - | 1.169126 | 1.186137 | 1.186949 |
| (0.7,0.3) | 0.870388 | 0.870375 | 0.89901 | 0.859154 | 0.870916 | 0.870372 |
| (0.1,0.5) | 1.274286 | 1.274275 | 1.27012 | 1.265541 | 1.273859 | 1.274227 |
| (0.5,0.5) | 0.810755 | 0.810749 | - | 0.799093 | 0.811537 | 0.810757 |

*Source: created by the authors*

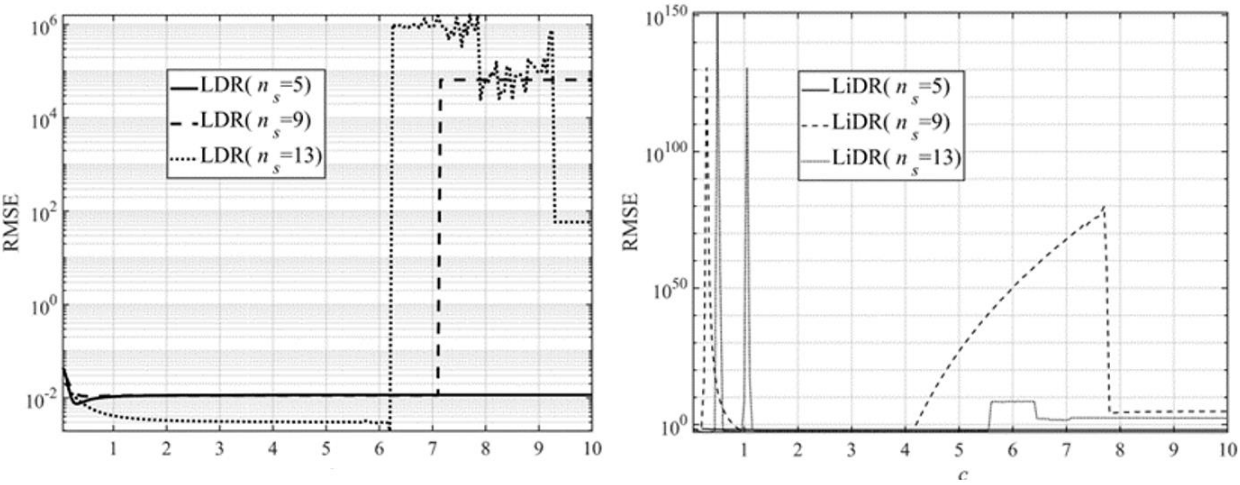


Fig. 7: Root mean square error (RMSE) measured at different shape values for the two local meshless methods at three levels of nearest-point densities: 5, 9 and 13

*Source: created by the authors*

Table 4. Error measurements recorded at two target times $(t = 1.00, 1.50)$ using two time step sizes $(\Delta t = 1.00E-04,\ 5.00E-05)$ for different levels of convection effect $\left(V = V_x = V_y \in \{2, 4, 6, 8\}\right)$ and $\varepsilon_x = \varepsilon_y = 1.4$ simulations produced by a fixed node size of $13\times13$ and $n_s = 13$ using the same shape value $c = 1.00$

| $V$ | Error norms | $\Delta t$ | $t = 1.00$ | | $t = 1.50$ | |
|---|---|---|---|---|---|---|
| | | | LDR | LiDR | LDR | LiDR |
| 2 | *Rel.Err* | 0.0001 | 5.52E-05 | 1.83E-05 | 5.52E-05 | 1.83E-05 |
| | | 0.00005 | 5.56E-05 | 1.83E-05 | 5.56E-05 | 1.83E-05 |
| | RMSE | 0.0001 | 1.51E-04 | 5.03E-05 | 2.50E-04 | 8.29E-05 |
| | | 0.00005 | 1.53E-04 | 5.02E-05 | 2.52E-04 | 8.28E-05 |
| 4 | *Rel.Err* | 0.0001 | 2.92E-04 | 1.35E-04 | 2.92E-04 | 1.35E-04 |
| | | 0.00005 | 2.92E-04 | 1.35E-04 | 2.92E-04 | 1.35E-04 |
| | RMSE | 0.0001 | 6.12E-04 | 2.83E-04 | 1.01E-03 | 4.67E-04 |
| | | 0.00005 | 6.13E-04 | 2.83E-04 | 1.01E-03 | 4.67E-04 |
| 6 | *Rel.Err* | 0.0001 | 8.71E-04 | 5.29E-04 | 8.71E-04 | 5.29E-04 |
| | | 0.00005 | 8.72E-04 | 5.29E-04 | 8.72E-04 | 5.29E-04 |
| | RMSE | 0.0001 | 1.53E-03 | 9.27E-04 | 2.52E-03 | 1.53E-03 |
| | | 0.00005 | 1.53E-03 | 9.27E-04 | 2.52E-03 | 1.53E-03 |
| 8 | *Rel.Err* | 0.0001 | 1.88E-03 | 1.33E-03 | 1.88E-03 | 1.33E-03 |
| | | 0.00005 | 1.88E-03 | 1.33E-03 | 1.88E-03 | 1.33E-03 |
| | RMSE | 0.0001 | 2.92E-03 | 2.06E-03 | 4.82E-03 | 3.40E-03 |
| | | 0.00005 | 2.92E-03 | 2.06E-03 | 4.82E-03 | 3.40E-03 |

*Source: created by the authors*

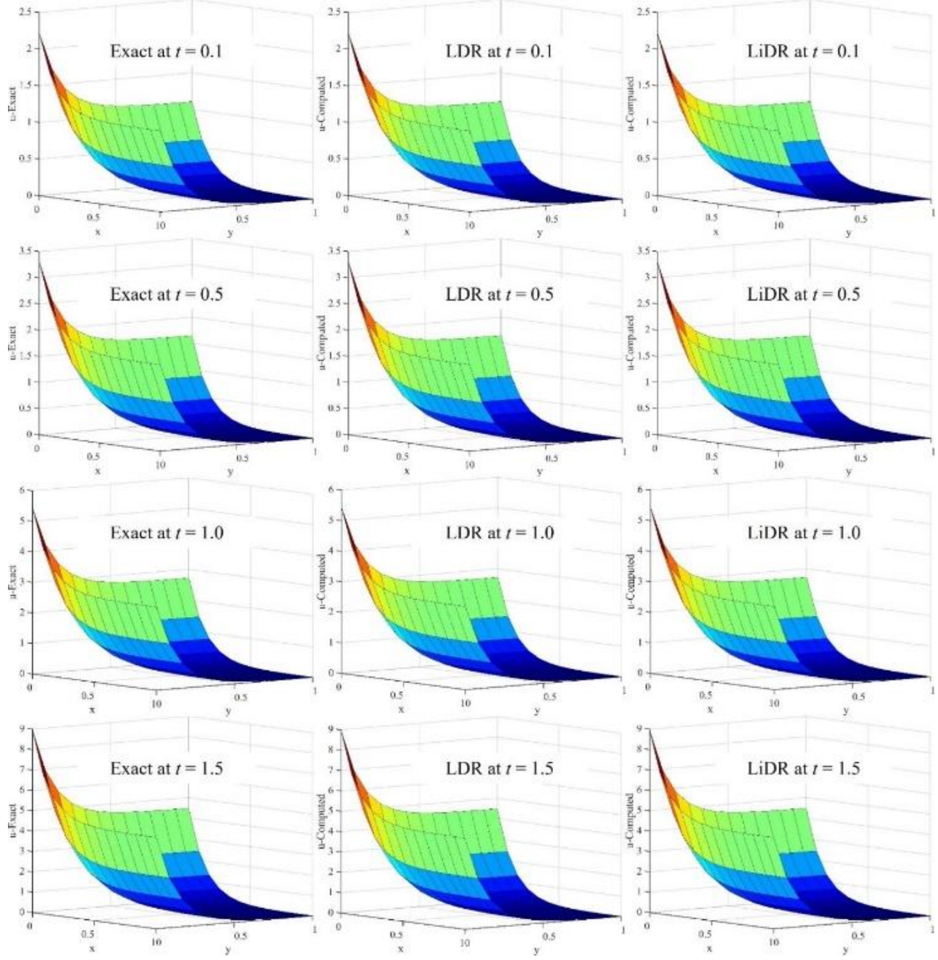


Fig. 8: Solution surface plot computed using $V_x = V_y = 10$, $\varepsilon_x = \varepsilon_y = 1.4$ at $t = 0.1$, $t = 0.5$, $t = 1$, $t = 1.5$ and $\Delta t = 0.0001$ with $n_s = 13$ and $N = 13 \times 13$. (LDR with $c = 3.06$ and LiDR with $c = 0.34$).

*Source: created by the authors*

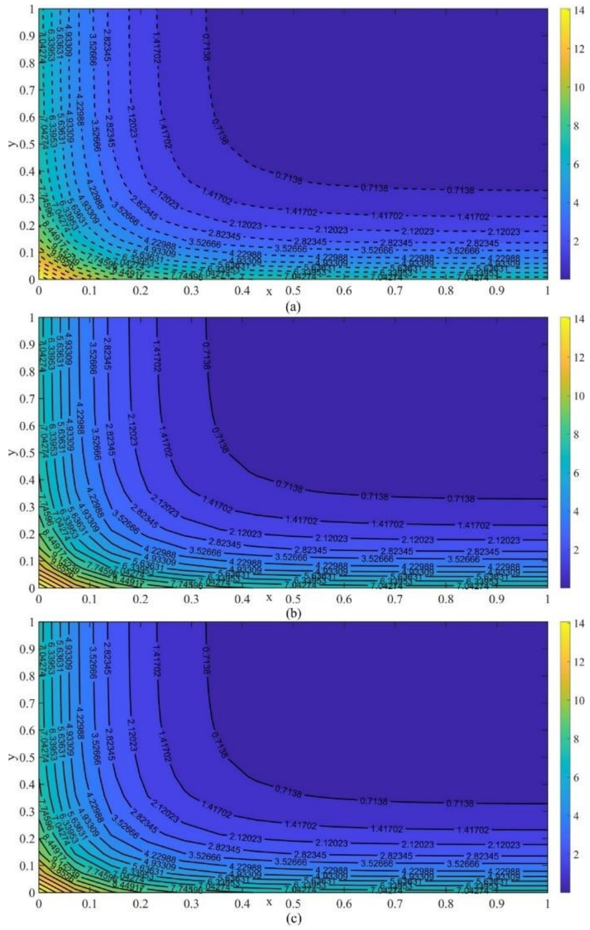

Fig. 9: Solution contours obtained for $V_x = V_y = 10$, $\varepsilon_x = \varepsilon_y = 1.4$ at $t = 2.0$ using $\Delta t = 0.0001$, $n_s = 13$ and $N = 15 \times 15$; (a) exact, (b) LDR with $c = 3.06$ and (c) LiDR with $c = 0.34$

*Source: created by the authors*

**Contribution of Individual Authors to the Creation of a Scientific Article (Ghostwriting Policy)**

The authors equally contributed in the present research, at all stages from the formulation of the problem to the final findings and solution. The corresponding author is Krittidej Chanthawara.

**Sources of Funding for Research Presented in a Scientific Article or Scientific Article Itself**

No funding was received for conducting this study.

**Conflicts of Interest**

The authors have no conflicts of interest to declare that are relevant to the content of this article.