The MLPG Method Based on Radial Point Interpolation Method for Solving Coupled Nonlinear Reaction-Diffusion Equations

Kanittha Yimnak[1], Anirut Luadsong[2]

[1]Department of Mathematics and Statistics, Faculty of Applied Science,
Dhurakij Pundit University
110/1-4 Prachachuen Road, Laksi,
Bangkok, 10210, Thailand

[2]Department of Mathematics, Faculty of Science,
King Mongkut's University of Technology Thonburi (KMUTT)
126 Pracha-utid Road, Bangmod, Toongkru,
Bangkok, 10140, Thailand
Email: anirut.lua {at} gmail.com

ABSTRACT--- ***The meshless local Petrov-Galerkin (MLPG) method is developed to solve the system of coupled nonlinear reaction-diffusion equations in two dimensional spaces subjected to Dirichlet and Neumann boundary conditions on a square domain. The spatial variations are approximated by radial point interpolation method (RPIM) and the nonlinear terms are treated iteratively within each time step. Two numerical examples are considered to demonstrate the applicability and the accuracy of the proposed method is investigated by root mean square of relative error.***

Keyword--- MLPG Method, Radial Point Interpolation Method, Reaction-Diffusion Equations, Crank-Nicolson Method.

1. INTRODUCTION

Systems of reaction-diffusion, which are proposed by Alan Turing [1], have an important application in mathematics, physics, chemistry and biology. The reaction-diffusion equation is often solved by numerical methods and usually diffusion is thought to be stabilizing. The idea that diffusion could make a stable and uniform chemical state unstable was innovative. Ahmand Shirzadi [2] developed the MLPG formulation for numerical solution of the nonlinear reaction-diffusion equations. The spatial variations are approximated by moving least squares and the nonlinear term are treated iteratively within each time step. Constructing of shape functions is one of the most important issues in the MLPG method. There are many methods for constructing a shape function such as the moving least squares (MLS) and the weighted least squares (WLS) method. The most popular method is MLS. Although the MLPG method and many other meshless methods have been gradually applied to different fields, there exists inconvenience because of the difficulty in implementing some essential boundary conditions; the shape function constructed by MLS approximation does not satisfy the Kronecker delta function property. In this research, we would like to present radial point interpolation method (RPIM), which was proposed by G. R. Liu [3] for constructing nodal shape function, has the Kronecker delta property that is a good property for shape function. We shall develop the MLPG formulation with using the Heaviside step test function on local sub-domains with temporal discretization by the *Crank-Nicolson* method. These systems are solved by local integral equation formulation and one-step time discretization method. The nonlinear terms are treated iteratively within each time step. The boundary and domain integrals are calculated using the Simpson and the Gauss-Legendre quadrature rules. Two numerical examples are considered in order to verify the proposed method with testing its accuracy implementing by root mean square of relative error.

2. GOVERNING EQUATION

The numerical simulations of the coupled pair of nonlinear partial differential equation are as follow:

$$\frac{\partial u}{\partial t} = D_1 \nabla^2 u + \alpha_1 u + A(u, v) + f_1(\boldsymbol{x}, t),$$

$$\frac{\partial v}{\partial t} = D_2 \nabla^2 v + \alpha_2 v + \beta u + B(u,v) + f_2(\boldsymbol{x},t) \tag{1}$$

given initial and Dirichlet and/or Neumann boundary conditions in the two-dimensional region Ω, where $D_1, D_2, \alpha_1, \alpha_2$ and β are given constants, A and B are functions of the field variables u and v, f_1 and f_2 are assumed to be prescribed sources. In the case of two-component reaction system, $u(\boldsymbol{x},t)$ and $v(\boldsymbol{x},t)$ stand for concentrations and D_1, D_2 for the diffusion coefficients of chemical species.

3. RADIAL POINT INTERPOLATION METHOD

The radial point interpolation method (RPIM) $u^h(\boldsymbol{x})$ at point $\boldsymbol{x}$ is defined as presented in [3,4]. Consider the function $u(\boldsymbol{x})$ defined in the domain Ω discretized by a set of properly scattered nodes $\boldsymbol{x}_i, (i = 1,2,\dots,n)$, where n is the total number of nodes in the whole domain. It is assumed that only N nodes surrounding point $\boldsymbol{x}$ have the effect on $u(\boldsymbol{x})$. The sub-domain Ω_x that encompasses these surrounding nodes is called the interpolation domain of point $\boldsymbol{x}$. The $u^h(\boldsymbol{x})$ at $\boldsymbol{x}_q$ is approximated in the form of

$$u^h(\boldsymbol{x},\boldsymbol{x}_q) = \sum_{i=1}^{N} R_i(\boldsymbol{x}) a_i + \sum_{j=1}^{m} P_j(\boldsymbol{x}) b_j = \boldsymbol{R}^T(\boldsymbol{x})\boldsymbol{a} + \boldsymbol{P}^T(\boldsymbol{x})\boldsymbol{b}, \tag{2}$$

where $R_i(\boldsymbol{x})$ is a radial basis function, $P_j(\boldsymbol{x})$ is monomial in the space coordinates $\boldsymbol{x}_i = (x_i, y_i)^T$, m is the number of polynomial basis functions. Coefficients a_i and b_j are constants yet to be determined. We also choose $m < N$ to have better stability of the interpolating function.

In general, a linear basis in two-dimensional space is:

$\boldsymbol{P}^T(\boldsymbol{x}) = (1, x, y), \quad m = 3,$

a quadratic basis is given as

$$\boldsymbol{P}^T(\boldsymbol{x}) = (1, x, y, x^2, xy, y^2), \quad m = 6$$

and a cubic basis is given as

$$\boldsymbol{P}^T(\boldsymbol{x}) = (1, x, y, x^2, xy, y^2, x^3, x^2y, xy^2, y^3), \qquad m = 10.$$

For give $\boldsymbol{x}$, we have

$\boldsymbol{a} = [a_1, a_2, \dots, a_N]^T, \qquad \boldsymbol{b} = [b_1, b_2, \dots, b_m]^T$

$\boldsymbol{R}^T(\boldsymbol{x}) = [R_1(\boldsymbol{x}), R_2(\boldsymbol{x}), \dots, R_N(\boldsymbol{x})], \qquad \boldsymbol{P}^T(\boldsymbol{x}) = [P_1(\boldsymbol{x}), P_2(\boldsymbol{x}), \dots, P_m(\boldsymbol{x})]$

Typically, in two-dimensional problems

$$\boldsymbol{R}_i(\boldsymbol{x}) = \boldsymbol{R}_i(r_i) = R_i(x,y) \tag{3}$$

$$\boldsymbol{r}_i(\boldsymbol{x}) = \sqrt{(x - x_i)^2 + (y - y_i)^2} \tag{4}$$

Now enforcing equation (2) to be satisfied at nodes to determine the coefficients a_i and b_i. The matrix form is :

$$\boldsymbol{U}_s = \boldsymbol{R}_Q \boldsymbol{a} + \boldsymbol{P}_m \boldsymbol{b} \tag{5}$$

where $\boldsymbol{U}_s = [u_1, u_2, \dots, u_N]^T$ Matrix $\boldsymbol{R}_Q$ is given by

$$\boldsymbol{R}_Q = \begin{bmatrix} R_1(\boldsymbol{x}_1) & R_2(\boldsymbol{x}_1) & \cdots & R_N(\boldsymbol{x}_1) \\ R_1(\boldsymbol{x}_2) & R_2(\boldsymbol{x}_2) & \cdots & R_N(\boldsymbol{x}_2) \\ \vdots & \vdots & \vdots & \vdots \\ R_1(\boldsymbol{x}_N) & R_2(\boldsymbol{x}_N) & \cdots & R_N(\boldsymbol{x}_N) \end{bmatrix}_{N\times N}. \tag{6}$$

The matrix $\boldsymbol{P}_m$ is a $N \times m$ matrix given by

$$\boldsymbol{P}_m = \begin{bmatrix} P_1(\boldsymbol{x}_1) & P_2(\boldsymbol{x}_1) & \cdots & P_m(\boldsymbol{x}_1) \\ P_1(\boldsymbol{x}_2) & P_2(\boldsymbol{x}_2) & \cdots & P_m(\boldsymbol{x}_2) \\ \vdots & \vdots & \vdots & \vdots \\ P_1(\boldsymbol{x}_N) & P_2(\boldsymbol{x}_N) & \cdots & P_m(\boldsymbol{x}_N) \end{bmatrix}_{N\times m}. \tag{7}$$

However, there are $N + m$ variables in equation (5), but only have N equations, so it is an undetermined equations, solving the above equation (5) needs to impose a constraint equation

$$\boldsymbol{P}_m^T \boldsymbol{a} = 0. \tag{8}$$

Combing equations (5) and (8), the matrix form becomes

$$\begin{bmatrix} \boldsymbol{R}_Q & \boldsymbol{P}_m \\ \boldsymbol{P}_m^T & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{a} \\ \boldsymbol{b} \end{bmatrix} = \begin{bmatrix} \boldsymbol{U}_s \\ 0 \end{bmatrix}. \tag{9}$$

Solving equation (9), we can obtain

$$\boldsymbol{b} = \boldsymbol{S}_b \boldsymbol{U}_s, \tag{10}$$

$$\boldsymbol{a} = \boldsymbol{S}_a \boldsymbol{U}_s, \tag{11}$$

where $\boldsymbol{S}_b = \left[\boldsymbol{P}_m^T \boldsymbol{R}_Q^{-1} \boldsymbol{P}_m\right]^{-1} \boldsymbol{P}_m \boldsymbol{R}_Q^{-1}, \quad \boldsymbol{S}_a = \boldsymbol{R}_Q^{-1} - \boldsymbol{R}_Q^{-1} \boldsymbol{P}_m \boldsymbol{S}_b.$

Substituting $\boldsymbol{a}, \boldsymbol{b}$ back into equation (2), we obtain

$$\boldsymbol{U}^h(\boldsymbol{x}, \boldsymbol{x}_q) = [\boldsymbol{R}^T \boldsymbol{S}_a + \boldsymbol{P}^T \boldsymbol{S}_b]\boldsymbol{U}_s$$

$$= \sum_{i=1}^{N} \boldsymbol{\Phi}_i(\boldsymbol{x})\, u_i = \boldsymbol{\Phi}(\boldsymbol{x})\boldsymbol{U}_s\,, \tag{12}$$

where shape function $\boldsymbol{\Phi}(\boldsymbol{x})$ with the size $1 \times N$ is given by

$$\boldsymbol{\Phi}(\boldsymbol{x}) = \left[\boldsymbol{\Phi}_1(\boldsymbol{x}), \boldsymbol{\Phi}_2(\boldsymbol{x}), \dots, \boldsymbol{\Phi}_N(\boldsymbol{x})\right]. \tag{13}$$

A radial basis function is defined as: Gaussian (EXP):

$$R_i(\boldsymbol{x}) = e^{(-cr^2)} = e^{(-c[(x-x_i)^2+(y-y_i)^2])}, \tag{14}$$

where c is all called shape-parameter.

The derivatives of shape functions can be easily obtained as

$$\frac{\partial \Phi_k}{\partial x} = \sum_{i=1}^{N} \frac{\partial R_i}{\partial x} S_{ik}^a + \sum_{j=1}^{N} \frac{\partial P_j}{\partial x} S_{jk}^b\,,$$

$$\frac{\partial \Phi_k}{\partial y} = \sum_{i=1}^{N} \frac{\partial R_i}{\partial y} S_{ik}^a + \sum_{j=1}^{N} \frac{\partial P_j}{\partial y} S_{jk}^b\,. \tag{15}$$

4. LOCAL INTEGRAL FORMULATION

The local integral formulation of equation (1) can be written as:

$$\int_{\Omega_s^i} \frac{\partial u}{\partial t} w_i\, d\Omega = D_1 \int_{\Omega_s^i} (\nabla^2 u)\, w_i\, d\Omega + \alpha_1 \int_{\Omega_s^i} u\, w_i\, d\Omega + \int_{\Omega_s^i} A(u, v)\, w_i\, d\Omega + \int_{\Omega_s^i} f_1(\boldsymbol{x}, t) w_i\, d\Omega, \tag{16}$$

$$\int_{\Omega_s^i} \frac{\partial v}{\partial t} w_i\, d\Omega = D_2 \int_{\Omega_s^i} (\nabla^2 v) w_i\, d\Omega + \alpha_2 \int_{\Omega_s^i} v\, w_i\, d\Omega + \beta \int_{\Omega_s^i} u\, w_i\, d\Omega + \int_{\Omega_s^i} B(u, v)\, w_i\, d\Omega + \int_{\Omega_s^i} f_2(\boldsymbol{x}, t) w_i\, d\Omega, \tag{17}$$

where w_i is a Heaviside step used as the test function:

$$w_i(\boldsymbol{x}) = \begin{cases} 1\,, & \boldsymbol{x} \in \Omega_s^i \\ 0\,, & \boldsymbol{x} \notin \Omega_s^i, \end{cases}$$

u and v are trial functions, and instead of the entire domain Ω_s^i we have considered a sub-domain Ω_s^i located entirely domain Ω, $\boldsymbol{x} = (x, y)^T \in \Omega \subset R^2$. The domain Ω is enclosed by $\Gamma = \Gamma_D \cup \Gamma_N$, with boundary conditions:

$$u = \bar{u} \quad \text{and} \quad v = \bar{v} \quad \text{on } \Gamma_D\,, \tag{18}$$

$$\boldsymbol{n} \cdot \nabla u = \bar{q}_u \quad \text{and} \quad \boldsymbol{n} \cdot \nabla v = \bar{q}_v \quad \text{on} \quad \Gamma_N\,, \tag{19}$$

where $\boldsymbol{n} = (n_1, n_2)^T$ is an outward unit normal vector of the boundary and $\boldsymbol{n} \cdot \nabla u \equiv \frac{\partial u}{\partial n}$. The condition (18) is often referred to as the Dirichlet boundary condition; and (19) as the Neumann boundary condition. Let $\tilde{u}(\boldsymbol{x}, t)$ and $\tilde{v}(\boldsymbol{x}, t)$ which substitute $u(\boldsymbol{x}, t)$ and $v(\boldsymbol{x}, t)$ respectively be the trial solutions:

$$\tilde{u}(\boldsymbol{x}, t) = \sum_{j=1}^{N} \phi_j(\boldsymbol{x})\hat{u}_j(t), \quad \tilde{v}_j(\boldsymbol{x}, t) = \sum_{j=1}^{N} \phi_j(\boldsymbol{x})\hat{v}_j(t),$$

$$\tilde{A} = A\left(\tilde{u}(\boldsymbol{x}, t), \tilde{v}(\boldsymbol{x}, t)\right), \quad \tilde{B} = B\left(\tilde{u}(\boldsymbol{x}, t), \tilde{v}(\boldsymbol{x}, t)\right).$$

For internal nodes, from local integral equations (16) and (17) and using the RPIM (2), we have the following nonlinear equations:

$$\sum_{j=1}^{N} \left[\int_{\Omega_s^i} \phi_j(\boldsymbol{x}) d\Omega\right] \frac{\partial \hat{u}_j}{\partial t} = \sum_{j=1}^{N} \left[D_1 \left(\int_{L_s^i} \phi_{j,n}(\boldsymbol{x}) d\Gamma + \int_{\Gamma_{sD}^i} \phi_{j,n}(\boldsymbol{x}) d\Gamma\right) + \left(\alpha_1 \int_{\Omega_s^i} \phi_j(\boldsymbol{x}) d\Omega\right)\right] \hat{u}_j$$

$$+ D_1 \int_{L_{sN}^i} \bar{q}_u d\Gamma + \int_{\Omega_s^i} \tilde{A}\, d\Omega + \int_{\Omega_s^i} f_1(\boldsymbol{x}, t) d\Omega, \tag{20}$$

$$\sum_{j=1}^{N} \left[\int_{\Omega_s^i} \phi_j(\boldsymbol{x}) d\Omega\right] \frac{\partial \hat{v}_j}{\partial t} = \sum_{j=1}^{N} \left[D_2 \left(\int_{L_s^i} \phi_{j,n}(\boldsymbol{x}) d\Gamma + \int_{\Gamma_{sD}^i} \phi_{j,n}(\boldsymbol{x}) d\Gamma\right) + \left(\alpha_2 \int_{\Omega_s^i} \phi_j(\boldsymbol{x}) d\Omega\right)\right] \hat{v}_j$$

$$+ \sum_{j=1}^{N} \left[\beta \int_{\Omega_s^i} \phi_j(\boldsymbol{x}) d\Omega\right] \hat{u}_j + \int_{\Omega_s^i} \tilde{B}\, d\Omega + D_2 \int_{L_{sN}^i} \bar{q}_v d\Gamma + \int_{\Omega_s^i} f_2(\boldsymbol{x}, t) d\Omega. \tag{21}$$

The following abbreviations have been used for the integral term:

$$k_{ij} = \int_{\Omega_s^i} \phi_j(\boldsymbol{x})\, d\Omega,$$

$$h_{ij}^1 = D_1 \left(\int_{L_s^i} \left(\frac{\partial \phi_j(\boldsymbol{x})}{\partial \boldsymbol{n}} \right) d\Gamma + \int_{L_{sD}^i} \left(\frac{\partial \phi_j(\boldsymbol{x})}{\partial \boldsymbol{n}} \right) d\Gamma \right) + \alpha_1 \int_{\Omega_s^i} \phi_j(\boldsymbol{x}) d\Omega,$$

$$h_{ij}^2 = D_2 \left(\int_{L_s^i} \left(\frac{\partial \phi_j(\boldsymbol{x})}{\partial \boldsymbol{n}} \right) d\Gamma + \int_{L_{sD}^i} \left(\frac{\partial \phi_j(\boldsymbol{x})}{\partial \boldsymbol{n}} \right) d\Gamma \right) + \alpha_2 \int_{\Omega_s^i} \phi_j(\boldsymbol{x}) d\Omega,$$

$$l_{ij} = \beta \int_{\Omega_s^i} \phi_j(\boldsymbol{x}) d\Omega,$$

$$f_{1i} = D_1 \int_{L_{sN}^i} \bar{q}_u d\Gamma + \int_{\Omega_s^i} \check{A}\, d\Omega + \int_{\Omega_s^i} f_1(\boldsymbol{x}, t) d\Omega,$$

$$f_{2i} = D_2 \int_{L_{sN}^i} \bar{q}_v d\Gamma + \int_{\Omega_s^i} \check{B}\, d\Omega + \int_{\Omega_s^i} f_2(\boldsymbol{x}, t) d\Omega.$$

The boundary and domain integrals are calculated by the Simpson and the Gauss-Legendre quadrature rules. We can rewrite Eqs. (20) and (21) as:

$$\sum_{j=1}^{N} k_{ij} \frac{\partial \hat{u}_j}{\partial t} = \sum_{j=1}^{n} h_{ij}^1 \hat{u}_j + f_{1i}, \tag{22}$$

$$\sum_{j=1}^{N} k_{ij} \frac{\partial \hat{v}_j}{\partial t} = \sum_{j=1}^{n} \left(h_{ij}^2 \hat{v}_j + l_{ij} \hat{u}_j \right) + f_{2i}. \tag{23}$$

Temporal discretization

The finite-difference approximation of the time derivatives of Eqs.(22) and (23) in case of the Crank-Nicolson method is given as follows:

$$(2\boldsymbol{K} - \Delta t \boldsymbol{H_1}) \hat{\boldsymbol{U}}^{k+1} = (2\boldsymbol{K} + \Delta t \boldsymbol{H_1}) \hat{\boldsymbol{U}}^{k} + \Delta t (\boldsymbol{F_1}^{k+1} + \boldsymbol{F_1}^{k}), \tag{24}$$

and

$$(2\boldsymbol{K} - \Delta t \boldsymbol{H_2}) \hat{\boldsymbol{V}}^{k+1} = (2\boldsymbol{K} + \Delta t \boldsymbol{H_2}) \hat{\boldsymbol{V}}^{k} + \Delta t \left(\boldsymbol{L} \hat{\boldsymbol{U}}^{k+1} + \boldsymbol{L} \hat{\boldsymbol{U}}^{k} \right) + \Delta t (\boldsymbol{F_2}^{k+1} + \boldsymbol{F_2}^{k}). \tag{25}$$

where $\boldsymbol{K} = \left[k_{ij}\right]_{N\times N}, \boldsymbol{H_1} = \left[h_{ij}^1\right]_{N\times N}, \boldsymbol{H_2} = \left[h_{ij}^2\right]_{N\times N}, \boldsymbol{L} = \left[l_{ij}\right]_{N\times N}$,

$\boldsymbol{F_1} = [f_{1i}]_{N\times 1}, \boldsymbol{F_2} = [f_{2i}]_{N\times 1}, \hat{\boldsymbol{U}} = [\hat{u}_1\ \hat{u}_2\ \cdots\ \hat{u}_N]^T$ and $\hat{\boldsymbol{V}} = [\hat{v}_1\ \hat{v}_2\ \cdots\ \hat{v}_N]^T$

Because of A and B are nonlinear function of u and v, we solve them iteratively in each time step with replacing A^{k+1} and B^{k+1} by A^k and B^k, respectively at zeroth iteration. Eqs.(24) and (25) are converted into a set of nonlinear algebraic equation for unknowns $\hat{\boldsymbol{U}}^{k+1}$ and $\hat{\boldsymbol{V}}^{k+1}$.

5. NUMERICAL EXPERIMENTS

The analyzed domain is $\Omega = [0,1] \times [0,1]$. The error of u and v, which are presented in the numerical results, are represented by root mean square of relative error (RMSRE) of u and v, respectively where

$$RMSRE_u = \sqrt{\frac{1}{N} \sum_{i=1}^{N} \left(\frac{u_i - \hat{u}_i}{u_i} \right)^2}, i = 1,2,\ldots,N$$

u_i and $\hat{u}_i$ are the exact and computed values of u at point $\boldsymbol{x_i}$, respectively, and N is the number of nodes.

5.1 Example1

The system of nonlinear PDEs in the region $\Omega = [0,1] \times [0,1]$ is given as follows:

$$\frac{\partial u}{\partial t} = \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) + u - u^2 v^2 + g_1(x,y,t), \qquad \frac{\partial v}{\partial t} = \left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} \right) + v - u^2 + uv + g_2(x,y,t),$$

where

$g_1(x,y,t) = e^{4x-6t} - 5e^{-2t+x+y}$, $g_2(x,y,t) = -4e^{-t+x-y} + e^{-4t+2x+2y} - e^{2x-3t}$.

The initial and Dirichlet boundary conditions are chosen in such a way that the exact solution is

$u(x,y,t) = e^{-2t+x+y}$, $v(x,y,t) = e^{-t+x-y}$.

The show results have been obtained using $N = 25, 81, 256$ and 441 nodal points, respectively, and $\Delta t = 0.1$ at time instant t =1. Fig.1. (a) and (b) show that the RMSRE of u and v using $m = 3,6$ and 10 decrease by increasing the number of nodal points. Moreover, the errors of u and v using $m = 6$ and 10 are less than the errors of u and v using $m = 3$. The profile of trial solution of u and v is similar to the profile of exact solution of u and v (see in Fig.2.(a),(b),(c) and (d)).

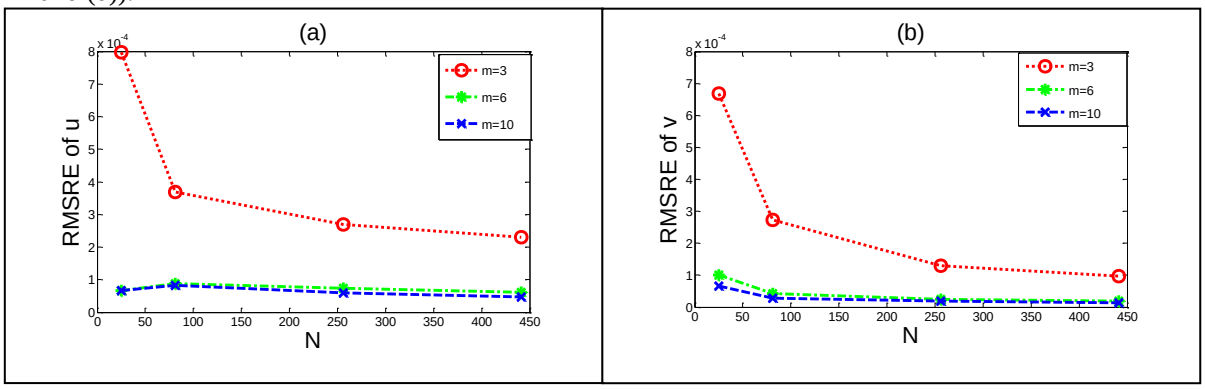

Fig.1 Error profiles of u and v against the number of nodal points obtained at time instant t=1 and Δt=0.1: (a) RMSRE of u; (b) RMSRE of v.

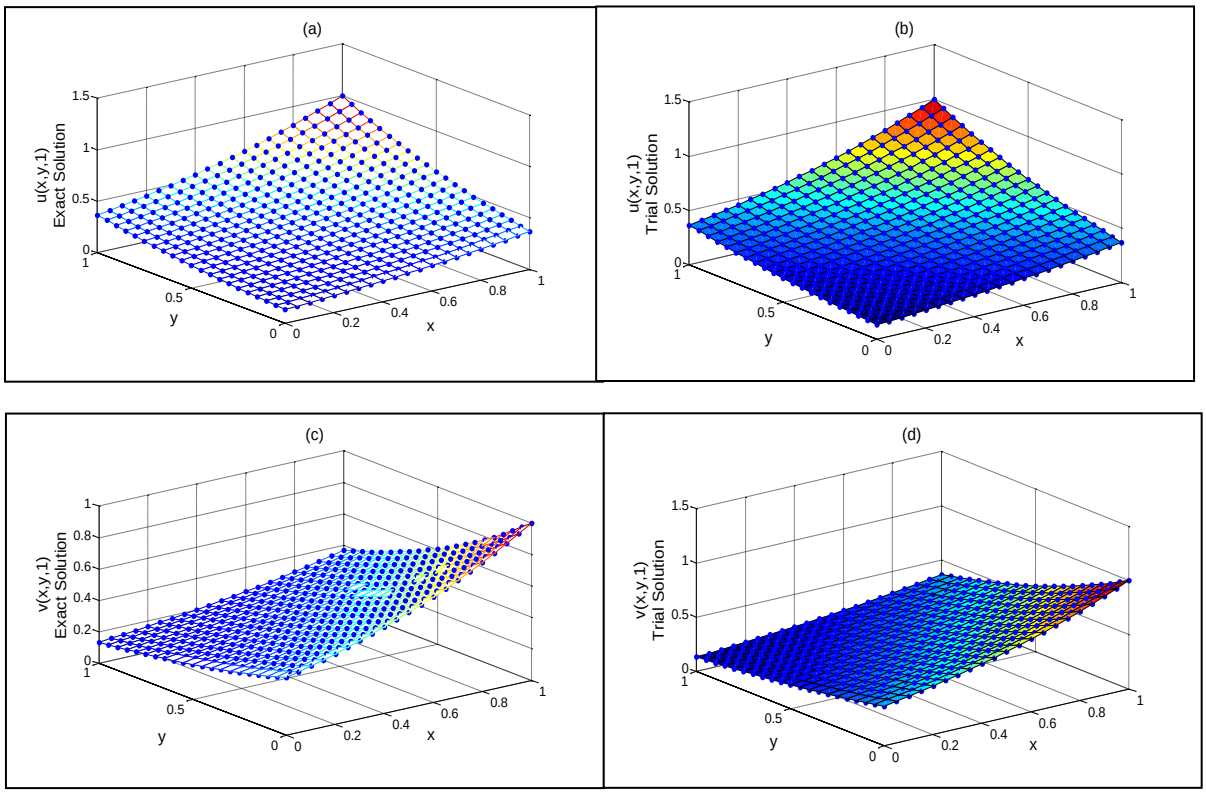

Fig.2 Exact and trial solutions of u and v by using N=441, Δt=0.1 and t=1: (a) exact solution of u ; (b) trial solution of u; (c) exact solution of v ; (d) trial solution of v.

5.2 Example 2, application for Brusselator system

The developed formulation from this research can solve a real world application example, let us consider the nonlinear reaction-diffusion Brusselator system in the two- dimensional region, [5], $\Omega = [0,1] \times [0,1]$:

$$\frac{\partial u}{\partial t} = D\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) - (A+1)u + u^2 v + B, \qquad \frac{\partial v}{\partial t} = D\left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}\right) + Au - u^2 v,$$

with $D = 0.002$, $A = \frac{1}{2}$, $B = 1$, initial conditions

$$u(x,y,0) = \frac{1}{2}x^2 - \frac{1}{3}x^3, \qquad v(x,y,0) = \frac{1}{2}y^2 - \frac{1}{3}y^3,$$

and Neumann boundary conditions:

$$\frac{\partial u(0,y,t)}{\partial x} = \frac{\partial u(1,y,t)}{\partial x} = \frac{\partial u(x,0,t)}{\partial y} = \frac{\partial u(x,1,t)}{\partial y} = 0,$$
$$\frac{\partial v(0,y,t)}{\partial x} = \frac{\partial v(1,y,t)}{\partial x} = \frac{\partial v(x,0,t)}{\partial y} = \frac{\partial v(x,1,t)}{\partial y} = 0,$$

for which the exact solution is unknown. For small values of the diffusion coefficient D, if $1 - A + B^2 > 0$ then the numerical solution of the Brusselator system converges to an equilibrium point (*B,A/B*). The experimental results for maximum and minimum values of the exact solution are presented in Table 1. According to table 1, it is found that the values of the exact solution tend to the steady state values of $(u^*, v^*) = (B, A/B) = \left(1, \frac{1}{2}\right)$. The experimental results are similar to those previously reported [6,5].

6. CONCLUSIONS

The developed formulation to obtain solutions for the two numerical experiments based on radial point interpolation method with temporal discretization by the Crank-Nicolson method has been presented. From the first example, the results show that the developed formulation works well and has accuracy of numerical results for these nonlinear equations. In addition, the number of basis function affect to accuracy. For the Brusselator model problem with the unknown exact solution, we found that the developed formulation obtained the steady state values.

Table1 The maximum and minimum values of the exact solution using $m = 10$, $\Delta t = 0.1$ and N=25.

t	u		v	
	u_{max}	u_{min}	v_{max}	v_{min}
1	0.586990455543974	0.52570531828970	0.317453110899033	0.138575008840007
2	0.755398496431872	0.694315822255890	0.466091284654928	0.346031634562740
3	0.870052342895777	0.805997309899885	0.533089537602479	0.479636570004167
4	0.949956398388360	0.898538087878307	0.539545939554035	0.530734881411691
5	0.991848818860750	0.963171718671362	0.530895214926649	0.521270589706326
6	1.005315506953328	0.995328634596902	0.516252746934818	0.507206553463093
7	1.006027994009456	1.004821765640326	0.505162623302219	0.500689091173829
8	1.004604289686220	1.003094432015647	0.500283668899898	0.498955282042240
9	1.002363425529594	1.001049649952028	0.499094568813038	0.499006982656086
10	1.000755019975633	1.000108074159773	0.499539238141448	0.499266022818354
↓	↓	↓	↓	↓
∞	1	1	0.5	0.5

7. ACKNOWLEDGEMENT

This research is partially supported by Dhurakij Pundit University, Thailand.

8. REFERENCES

[1] Turing A. The chemical basis of morphogenesis. *Philos Trans R Soc*, London B, 1952, 237(641), pp. 37-72.
[2] Shirzadi A., Sladek V. and Sladek J**.** A local integral equation formulation to solve coupled nonlinear reaction-diffusion equations by using moving least square approximation, *Engineering Analysis with Boundary Elements*, 2013, 37(1), pp. 8-14.
[3] Liu G.R., Zhang G., Wang Y. Y., Zhong Z.H., Li G.Y. and Han X., A nodal integration technique for meshfree radial point interpolation method (NI RPIM), *International Journal of Solids and Structures*, 2007, 44(1), pp.3840–3860.
[4] Yu C., Maohui X., Dehua W. and Dongmei L., Application of radial point interpolation method to temperature field, *Journal of Mathematics Research*, 2010, 2(1), pp. 139-142.
[5] Twizell E., Gumel A., and Cao Q. A second-order scheme for the Brusselator reaction-diffusion system, *J Math Chem*, 1999, 26(1), pp.297-316.
[6] Ul Islam S. , Ali A. and Haq S., A computational modeling of the behavior of the two-dimentional reaction diffusion Brusselator system, *Appl Math Modell*, 2010, 34(1),. pp. 3896-3909.