APPLICATION OF MOVING KRIGING INTERPOLATION BASED ON THE MESHLESS LOCAL PETROV-GALERKIN (MK-MLPG) METHOD FOR THE TWO-DIMENSIONAL TIME-DEPENDENT SCHRÖDINGER EQUATION

ALI HABIBIRAD AND ESMAIL HESAMEDDINI *
DEPARTMENT OF APPLIED MATHEMATICS, FACULTY OF BASIC SCIENCES, SHIRAZ UNIVERSITY OF TECHNOLOGY, SHIRAZ, IRAN.
E-MAILS: A.HABIBIRAD@SUTECH.AC.IR, HESAMEDDINI@SUTECH.AC.IR

(Received: 11 July 2018, Accepted: 20 January 2019)

Abstract. In this article, an efficient numerical technique for solving the two-dimensional time-dependent Schrödinger equation is presented. At first, we employ the meshless local Petrov-Galerkin (MLPG) method based on a local weak formulation to construct a system of discretized equations and then the solution of time-dependent Schrödinger equation will be approximated. We use the Moving Kriging (MK) interpolation instead of Moving least Square (MLS) approximation to construct the MLPG shape functions and hence the Heaviside step function is chosen as a test function on each subdomain. In this method, no mesh is needed neither for integration of the local weak form nor construction of the shape functions. So, the MLPG is truly a meshless method. Several numerical examples are presented and the results are compared to their analytical and RBF solutions to illustrate the accuracy and capability of this algorithm.

AMS Classification: 65M60, 34A45.
Keywords: Meshless local Petrov-Galerkin (MLPG) method, Two-dimensional time-dependent Schrödinger equation, Moving Kriging interpolation.

* CORRESPONDING AUTHOR
JOURNAL OF MAHANI MATHEMATICAL RESEARCH CENTER
VOL. 7, NUMBERS 1-2 (2018) 105-125.
DOI: 10.22103/JMMRC.2019.12431.1063

1. Introduction

The nonlinear Schrödinger equation (NLSE) is an important partial differential equation (PDE) in modern physics with applications in many different branches of physics and applied mathematics, including nonlinear quantum field theory, condensed matter and plasma physics, nonlinear optics and quantum electronics, fluid mechanics, theory of turbulence and phase transitions, biophysics, star formation and so on [10].

We consider the two-dimensional time-dependent Schrödinger equation with the following form:

$$-i\frac{\partial u}{\partial t} = \nabla^2 u + \omega(x,y)u, \quad (x,y) \in \Omega, \tag{1}$$

in some connected domain with suitable initial and Dirichlet boundary conditions and an arbitrary potential function $\omega(x,y)$.

Several numerical methods are considered for (1), among others, we mention Subasi [25] using the finite difference schemes, Dehghan and Shokri [13] using collocation and radial basis functions, Kalita et al. [18] by a semi-discrete higher order compact scheme, Antoine et al [3] by a Crank-Nicolson implicit scheme, Dehghan [14] using several finite difference techniques. Authors of [2] used the dual reciprocity boundary element method to study the generalized nonlinear two-dimensional Schrödinger equation, etc.

1.1. **The meshless methods.**

In recent years, more and more attentions have been paid to meshless methods, since they do not require mesh to discrete the problem domain, and they are very flexible in solving boundary value problems, especially, problems with discontinuities, moving boundaries and severe material deformations. There are a great number of meshless methods under current development, including the reproducing kernel particle method proposed by Liu et al [21], element-free galerkin method proposed by Belytschko et al [9], point interpolation method by Liu et al [19], smooth particle hydrodynamics method by Gingold and Monaghan [15], the meshless local Petrov-Galerkin (MLPG) method by Atluri and Zhu [4, 5], radial point interpolation method by Liu and Gu [20], Wang and Liu [27], Liu et al [26] . The meshless local Petrov-Galerkin (MLPG) method is a completely meshless method, which never employs mesh divisions in analysis. The

concept of MLPG was first proposed by Atluri and Shen [6], and later discussed in depth by Atluri and Shen [6, 7] and Atluri [8].

The concept of shape function construction is one of the central and most important issues that significantly effects on the performance of meshfree methods. A number of ways to create shape functions have been proposed including the moving least squares (MLS) approximation, radial point interpolation method (RPIM), and MK interpolation. The MLS shape functions do not have the Kronecker delta property thereby making the imposition of essential boundary conditions are complicated. In order to eliminate this shortcoming of the MLS shape functions, the MK interpolation technique, which has the Kronecker delta function and consistency property, can be employed instead of the traditional MLS approximation to construct the meshless shape functions. The MK interpolation is an approximation procedure originally employed in geostatistics by using known values and a semivariogram to determine unknown values [11].

In this paper, we will provide a meshless MLPG method based on MK interpolation to obtain the numerical solution for the 2D time-dependent Schrödinger equation. The organization of this paper is as follows: In Section 2, we briefly explain the MK interpolation. In Section 3, we describe the numerical implementation of MLPG method for the linear Schrödinger equation. Moreover, in this section the construction of local weak form of Schrödinger equation and the time difference approximation are discussed. Numerical experiments for some examples are reported in Section 4. Finally, a conclusion is given in Section 5.

2. The moving Kriging interpolation approach

Similar to the MLS scheme, the moving Krigin approach approximates the distribution function $u(\mathbf{x})$ in a sub-domain Ω_s ($\Omega_s \subset \Omega$, Ω is the domain of $u(\mathbf{x})$). We use the following procedure to construct the shape functions of the MLPG method by MK interpolation. Suppose that Ω discretized by a set of suitably scattered nodes $\mathbf{x}_i$ ($i = 1, 2, \ldots, n$). It is assumed that only N nodes of this randomly located nodes are in neighbourhood of a point $\mathbf{x}$ (i.e. Ω_s) and have effect on $u(\mathbf{x})$. The MK interpolation $u^h(\mathbf{x})$ can be shown in the form of linear combination of the shape

functions defined as [11, 16]

$$u^h(\mathbf{x}) = \sum_{i=1}^{N} \Phi_i(\mathbf{x}) u_i = \mathbf{\Phi}(\mathbf{x})\mathbf{u}, \quad \mathbf{x} \in \Omega_s, \tag{2}$$

in which

$$\mathbf{\Phi}(\mathbf{x}) = \mathbf{p}^T(\mathbf{x})A + \mathbf{r}^T(\mathbf{x})B. \tag{3}$$

The matrices A and B are defined as follows [12]:

$$A = (P^T R^{-1} P)^{-1} P^T R^{-1}, \tag{4}$$

$$B = R^{-1}(I - PA), \tag{5}$$

where I is an $N \times N$ unit matrix and $\mathbf{p}(\mathbf{x})$ has the following vector form

$$\mathbf{p}^T(\mathbf{x}) = [p_1(\mathbf{x}) \cdots p_m(\mathbf{x})], \tag{6}$$

where $p_j(\mathbf{x})$ is a polynomial basis function, which has monomial terms. Also, we use a linear basis in two-dimensional space in our computations as

$$\mathbf{p}^T(\mathbf{x}) = [1, x, y], \ m = 3,$$

where the quadratic polynomial basis is

$$\mathbf{p}^T(\mathbf{x}) = [1, x, y, x^2, xy, y^2], \ m = 6,$$

and the cubic polynomial basis as

$$\mathbf{p}^T(\mathbf{x}) = [1, x, y, x^2, xy, y^2, x^3, x^2y, xy^2, y^3], \ m = 10.$$

The matrix P represents the values of polynomial basis functions in Eq (6) at the given set of nodes as

$$P = \begin{bmatrix} p_1(\mathbf{x}_1) & \cdots & p_m(\mathbf{x}_1) \\ \cdots & \cdots & \cdots \\ p_1(\mathbf{x}_N) & \cdots & p_m(\mathbf{x}_N) \end{bmatrix}. \tag{7}$$

Moreover, the vector $\mathbf{r}(\mathbf{x})$ in Eq (3) is given by

$$\mathbf{r}^T(\mathbf{x}) = \begin{bmatrix} \gamma(\mathbf{x}, \mathbf{x}_1) & \cdots & \gamma(\mathbf{x}, \mathbf{x}_N) \end{bmatrix}, \tag{8}$$

where $\gamma(\mathbf{x}_i, \mathbf{x}_j)$ is the correlation function between any pair of nodes located at $\mathbf{x}_i$ and $\mathbf{x}_j$. Many different functions can be used as a correlation function $\gamma(\mathbf{x}_i, \mathbf{x}_j)$ [24]. Here, we use the Gaussian weight function, as

$$\gamma(\mathbf{x}, \mathbf{x}_i) = \begin{cases} \frac{\exp[-(d_i/c_i)^2]-\exp[-(r_i/c_i)^2]}{1-\exp[-(r_i/c_i)^2]}, & 0 \le d_i \le r_i, \\ 0, & d_i \ge r_i, \end{cases} \tag{9}$$

where $d_i = \|\mathbf{x} - \mathbf{x}_i\|$, c_i a constant controlling the shape of the weight function and r_i is size of the support domain [1].

Hence, the correlation matrix $R[\gamma(\mathbf{x}_i, \mathbf{x}_j)]_{N\times N}$ is given by

$$R = \begin{bmatrix} \gamma(\mathbf{x}_1, \mathbf{x}_1) & \cdots & \gamma(\mathbf{x}_1, \mathbf{x}_N) \\ \cdots & \cdots & \cdots \\ \gamma(\mathbf{x}_N, \mathbf{x}_1) & \cdots & \gamma(\mathbf{x}_N, \mathbf{x}_N) \end{bmatrix}. \tag{10}$$

The partial derivatives of shape function ϕ_i can be obtained as follows

$$\phi_{i,x} = \frac{\partial \Phi_k(x)}{\partial x} = \sum_j^m \frac{\partial p_j(x)}{\partial x} A_{ji} + \sum_k^N \frac{\partial r_k(x)}{\partial x} B_{ki}, \tag{11}$$

$$\phi_{i,y} = \frac{\partial \Phi_k(x)}{\partial y} = \sum_j^m \frac{\partial p_j(x)}{\partial y} A_{ji} + \sum_k^N \frac{\partial r_k(x)}{\partial y} B_{ki}. \tag{12}$$

Unlike to the shape function of MLS scheme, the shape function obtained by the MK interpolation has the following δ Kronecker property:

$$\phi_k(\mathbf{x}_j) = \begin{cases} 1, & (k = j; k, j = 1, 2, \ldots, N), \\ 0, & (k \ne j; k, j = 1, 2, \ldots, N). \end{cases} \tag{13}$$

To see the other properties of the MK interpolation one can refer to [11, 16, 12].

3. Local weak formulation

We consider the two-dimensional time-dependent Schrödinger equation with the following form ($i = \sqrt{-1}$ is imaginary number):

$$-i\frac{\partial u}{\partial t} = \nabla^2 u + \omega(x, y)u, \quad (x, y) \in \Omega, \tag{14}$$

subject to Dirichlet boundary condition

$$u(x, y, t) = g(x, y, t), \quad (x, y) \in \partial\Omega,\ 0 < t \le T,$$

and the initial condition

$$u(x, y, 0) = f(x, y), \quad (x, y) \in \Omega, \tag{15}$$

where f and ω are known functions, while the function u is unknown. The MLPG formulation constructs the weak form on local sub-domains such as Ω_s bounded by Γ_s. The local sub-domains overlap each other and covering the full domain Ω . They could be of any geometric shape and size [4], such as circles in two dimensions and spheres or cubes in three dimensions. For simplicity, we take them to be a circular shape. The local weak form of Eq (14) at each $\mathbf{x} \in \Omega_s$ can be weighted by test functions and integrated over a local sub-domain (Fig.1). Then, this equation can be written as

$$\int_{\Omega_s} (i\frac{\partial u}{\partial t} + \nabla^2 u + \omega u)\nu d\Omega = 0, \tag{16}$$

where ν is a test function. Using the divergence theorem, the following equation is resulted:

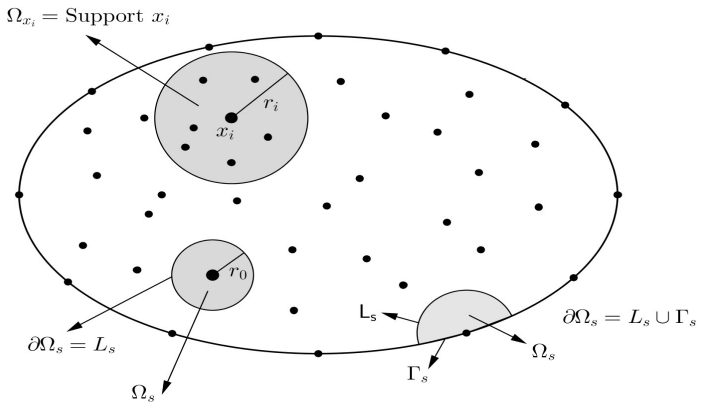

FIGURE 1. Local sub-domains Ω_s and global domain Ω.

$$\int_{\Omega_s} (i\frac{\partial u}{\partial t} + \omega u)\nu d\Omega - \int_{\Omega_s} (\frac{\partial u}{\partial x}\frac{\partial \nu}{\partial x} + \frac{\partial u}{\partial y}\frac{\partial \nu}{\partial y})d\Omega + \int_{\partial\Omega_s} (n_x\frac{\partial u}{\partial x}\nu + n_y\frac{\partial u}{\partial y}\nu)d\Gamma = 0, \tag{17}$$

where $\partial\Omega_s$ is the boundary of local sub-domain Ω_s. We take Heaviside step function

$$\nu = \begin{cases} 1, & \mathbf{x} \in \Omega_s, \\ 0, & \mathbf{x} \notin \Omega_s, \end{cases} \tag{18}$$

as the test function in any sub-domain [22, 17], then the local weak form of Eq (17) arrive to the following integral equation

$$\int_{\Omega_s}(i\frac{\partial u}{\partial t}+\omega u)d\Omega+\int_{\partial\Omega_s}(n_x\frac{\partial u}{\partial x}+n_y\frac{\partial u}{\partial y})d\Gamma=0, \tag{19}$$

where $\partial\Omega_s=\Gamma_s\cup L_s$, L_s is the subsection of the local boundary over which no boundary conditions are specified and Γ_s is the subscribe of the local boundary and the global boundary (Fig.1). So, the local weak form of Eq (19) arrive at

$$\int_{\Omega_s} i\frac{\partial u}{\partial t}d\Omega+\int_{\Omega_s}\omega u d\Omega+\int_{\Gamma_s}(n_x\frac{\partial u}{\partial x}+n_y\frac{\partial u}{\partial y})d\Gamma+\int_{L_s}(n_x\frac{\partial u}{\partial x}+n_y\frac{\partial u}{\partial y})d\Gamma=0. \tag{20}$$

To construct the Dirichlet boundary conditions, the delta function property of shape functions will be used. The solution of Eq (1) is a function with both the spatial coordinates and time. Suppose that only N nodes are in neighbourhood of point $\mathbf{x}$ have effect on the numerical solution, so the MK interpolation implies that

$$u(\mathbf{x},t)=\sum_{j=1}^{N}\phi_j(\mathbf{x})u_j(t)=\Phi(\mathbf{x})\mathbf{u}(t). \tag{21}$$

Substituting this relation in the local weak-form of Eq (20), we obtain the following discrete equations for all nodes

$$-iC\dot{\mathbf{u}}=(B+K)\mathbf{u}, \tag{22}$$

in which

$$\begin{aligned} C_{ij}&=\int_{\Omega_s}\phi_j\ d\Omega,\\ B_{ij}&=\int_{L_s}\left(n_x\frac{\partial\phi_j}{\partial x}+n_y\frac{\partial\phi_j}{\partial y}\right)\ d\Gamma+\int_{\Gamma_s}\left(n_x\frac{\partial\phi_j}{\partial x}+n_y\frac{\partial\phi_j}{\partial y}\right)\ d\Gamma,\\ K_{ij}&=\int_{\Omega_s}\omega\phi_j\ d\Omega. \end{aligned} \tag{23}$$

Using the backward difference technique to approximate the first order derivative in Eq (22), results in

$$\dot{\mathbf{u}}=\frac{\partial\mathbf{u}}{\partial t}=\frac{\mathbf{u}^{n+1}-\mathbf{u}^n}{dt}.$$

Also, the finite difference approximation for u is as follows

$$u=\frac{\mathbf{u}^{n+1}+\mathbf{u}^n}{2}.$$

Therefore, Eq (22) can be written as

$$-iC\frac{\mathbf{u}^{n+1}-\mathbf{u}^{n}}{dt}=(B+K)\frac{\mathbf{u}^{n+1}+\mathbf{u}^{n}}{2}, \tag{24}$$

or

$$(iC+\frac{dt}{2}(B+K))\mathbf{u}^{n+1}=(iC-\frac{dt}{2}(B+K))\mathbf{u}^{n}, \tag{25}$$

where $\mathbf{u}^n = \mathbf{u}(n.dt)$. By solving this system the numerical solution for $\mathbf{u}$ at the nodal points will be determined.

4. Numerical examples

To support our method, we present numerical results of the MK-MLPG for the $2D$ time-dependent Schrödinger equation on five examples. In these examples we report the L_∞ and maximum errors which are defined by

$$\begin{aligned} L_\infty &= \left\|u^{exact}-u^{numerical}\right\|_\infty = \max_i \left|u_i^{exact}-u_i^{numerical}\right|, \\ \epsilon &= \frac{\left\|u^{exact}-u^{numerical}\right\|_\infty}{\left\|u^{exact}\right\|_\infty}, \end{aligned}$$

and in some examples, we report the order of our method in space or time variables [23] with

$$c-\text{order}=\frac{\log\left(\frac{E_1}{E_2}\right)}{\log\left(\frac{h_1}{h_2}\right)},$$

to evaluate the efficiency of our method (where E_1 and E_2 are maximum errors correspond to h_1 and h_2, respectively) . Note that 8×8 Gauss points are used in each local sub-domain Ω_s, 7 Gauss points are used on each local boundary Γ_s and boundary section L_s for the numerical Gauss integration. Also, we take $d_i = dx/2$, $r_i = 4d_i$ and $c_i = 4r_i$ in domains and sub-domains for Gaussian weight function (9).

Remark. We performed our computations by using $MATLAB$ software on a $Core\ i3-2100\ PC$ with a $3.10-GHz\ CPU$ and $2-GB$ of memory.

4.1. **Example 4.1.** In this example, we consider the time-dependent Schrödinger Eq (1) in $(x, y) \in \Omega = [0, \pi] \times [0, \pi]$, subject to the following initial condition

$$u(x, y, 0) = \sin(x) \sin(y), \tag{26}$$

and Dirichlet boundary conditions which are zero on all sides, with the given potential function as

$$\omega(x, y) = 3, \qquad (x, y) \in \Omega. \tag{27}$$

The exact solution of this problem is

$$u(x, y, t) = e^{it} \sin(x) \sin(y). \tag{28}$$

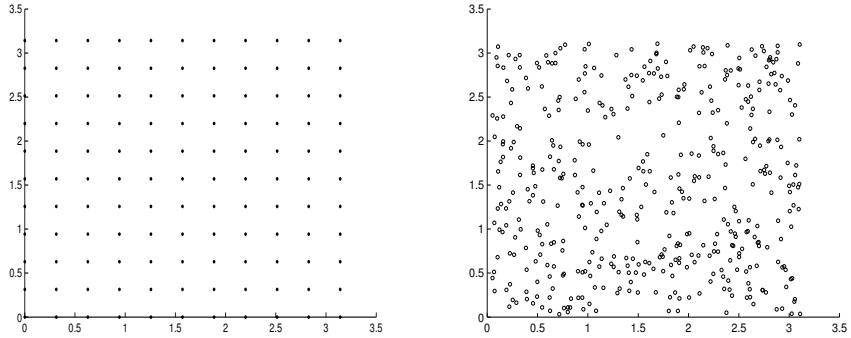

FIGURE 2. The regular (121 nodes) and irregular (441 nodes) domains in example 4.1.

The L_∞ and maximum absolute errors for the real and imaginary parts at $t = 0.1, 0.3, 0.5, 0.7, 1$ with $dt = 0.01$ on 11×11 uniformed nodes (Fig.2 left) are reported in Table 1. The last column of Table 1 shows the CPU times.

In Table 2, we report the L_∞ error for the real and imaginary parts, maximum error for different numbers of nodes in the space variables for the MK-MLPG scheme and $c-$ order of our method with $t = 1$ and $dt = 0.01$. Considering the irregular domain of Fig.2 (right), the maximum error (ϵ) and CPU times calculated by the presented method (MK-MLPG) are listed in Table 3. Fig.3, shows a comparison between numerical results obtained by MK-MLPG method with the exact solution of Real (left) and Imaginary (right) parts of the temperature at $y = \pi/2$ for different times. In Fig.4, one can observe a comparison between numerical results obtained

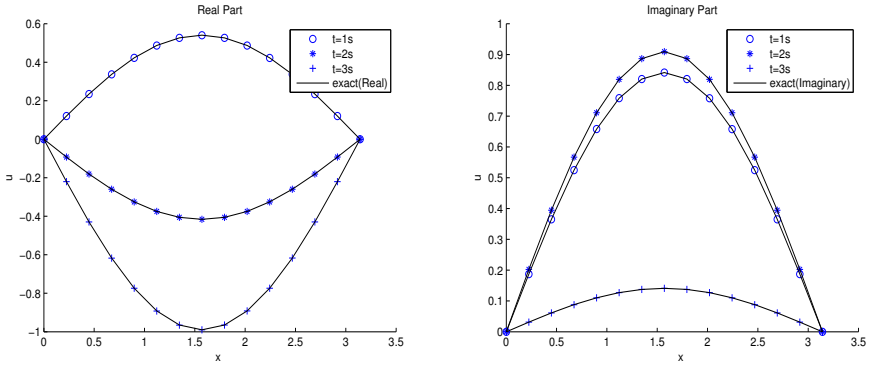

FIGURE 3. comparison between the exact solution and the results of present method at $y = \pi/2$, for example 4.1.

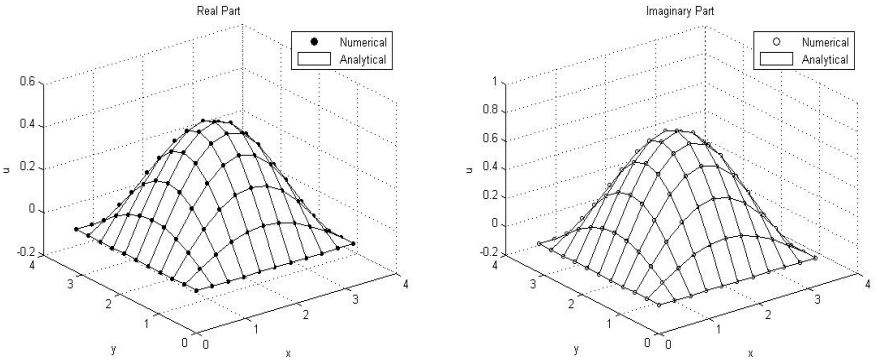

FIGURE 4. Plot of the Real (left) and Imaginary (right) parts of numerical and exact solutions for example 4.1.

by MK-MLPG method with the exact solution for the real and imaginary parts at $t = 1s$.

These tables and graphs reveal that the results obtained by the MK-MLPG scheme, are in a good agreement with the exact solutions.

4.2. **Example 4.2.** We consider Eq (1) in a square domain $\Omega = [0, 1] \times [0, 1]$ with the following potential function

$$\omega(x, y) = 1 - \frac{2}{x^2} - \frac{2}{y^2}, \tag{29}$$

TABLE 1. L_∞ , Maximum error and CPU times for example 4.1.

	L_∞error		Max-error(ϵ)	CPU time (s)
t	$Real$	$Imaginary$		
0.1	1.02(−4)	4.82(−5)	1.02(−4)	0.4
0.3	1.24(−4)	8.17(−5)	1.34(−4)	0.4
0.5	1.14(−4)	7.93(−5)	1.15(−4)	0.6
0.7	7.88(−5)	1.19(−4)	1.32(−4)	0.7
1	1.08(−4)	1.25(−4)	1.25(−4)	1.0

TABLE 2. L_∞ and Maximum error at $t = 1$ with $dt = 0.01$ for example 4.1.

Number of points	L_∞error		Max-error(ϵ)	$c-$ order
	$Real$	$Imaginary$		
$(11 \times 11,\ h = \pi/10)$	1.08(−4)	1.25(−4)	1.25(−4)	–
$(15 \times 15,\ h = \pi/14)$	6.98(−5)	6.85(−5)	7.40(−5)	1.55
$(21 \times 21,\ h = \pi/20)$	3.52(−5)	4.51(−5)	4.31(−5)	1.51
$(41 \times 41,\ h = \pi/40)$	2.43(−5)	3.49(−5)	2.13(−5)	1.02

TABLE 3. The maximum error(ϵ) for irregular 441 nodes Fig.2 (right) with $dt = 0.01$ in example 4.1.

time(s)	Max-error (ϵ)	CPU Time (s)
0.10	7.18(−2)	5
0.50	1.71(−2)	8
1.00	2.81(−2)	12

subject to the initial and boundary conditions as

$$u(x, y, 0) = x^2 y^2, \quad (x, y) \in \Omega, \tag{30}$$

and

$$\begin{aligned} &u(0, y, t) = 0, \qquad u(1, y, t) = y^2 \exp(it), \\ &u(x, 0, t) = 0, \qquad u(x, 1, t) = x^2 \exp(it). \end{aligned} \tag{31}$$

The analytical solution of this equation is given in [25] as

$$u(x, y, t) = x^2 y^2 \exp(it). \tag{32}$$

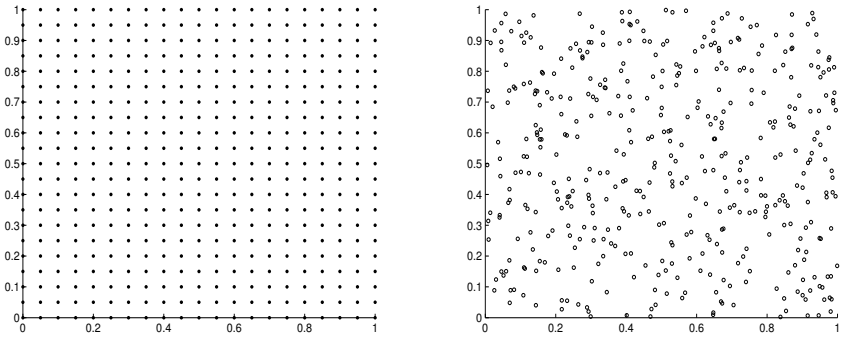

FIGURE 5. The regular (441 nodes) and irregular (441 nodes) domains considered in example 4.2.

TABLE 4. L_∞ and Maximum error (ϵ) with $dt = 0.0005$ for example 4.2.

	L_∞ *Re part*		L_∞ *Im part*		Max-error(ϵ)	
t	MK-MLPG	RBF[13]	MK-MLPG	RBF[13]	MK-MLPG	RBF[13]
0.1	2.79(−4)	4.04(−4)	2.34(−4)	3.57(−4)	3.18(−4)	4.14(−4)
0.3	2.11(−4)	5.12(−4)	2.31(−4)	3.05(−4)	2.43(−4)	5.13(−4)
0.5	1.81(−4)	4.63(−4)	3.01(−4)	3.95(−4)	3.37(−4)	4.97(−4)
0.7	2.62(−4)	3.89(−4)	1.80(−4)	4.16(−4)	3.03(−4)	5.70(−4)
1	2.51(−4)	3.72(−4)	3.48(−4)	4.12(−4)	4.02(−4)	4.26(−4)

TABLE 5. L_∞ and Maximum error for $t = 1$ with $dt = 0.01$ for example 4.2.

Number of points	L_∞error		Max-error(ϵ)	$c-$ order
	Real	*Imaginary*		
$(11 \times 11,\ h = \frac{1}{10})$	7.43(−4)	3.36(−4)	3.02(−4)	–
$(21 \times 21,\ h = \frac{1}{20})$	4.25(−4)	6.38(−5)	7.84(−5)	1.94
$(41 \times 41,\ h = \frac{1}{400})$	1.54(−5)	2.28(−5)	2.70(−5)	1.53

The second and the third columns of Table 4 present L_∞ error for the real part by MK-MLPG and RBF methods [13]. Also, two next columns show L_∞ error

of imaginary part by our method and RBF method [13]. The last two columns show maximum error of proposed method and RBF scheme [13]. These results are obtained for 121 uniformed nodes ($dx = dy = 0.1$) at different times up to $t = 1$ (first column of Table 4) with $dt = 0.0005$. As one can see from this table, the obtained solutions are superior to the results of [13].

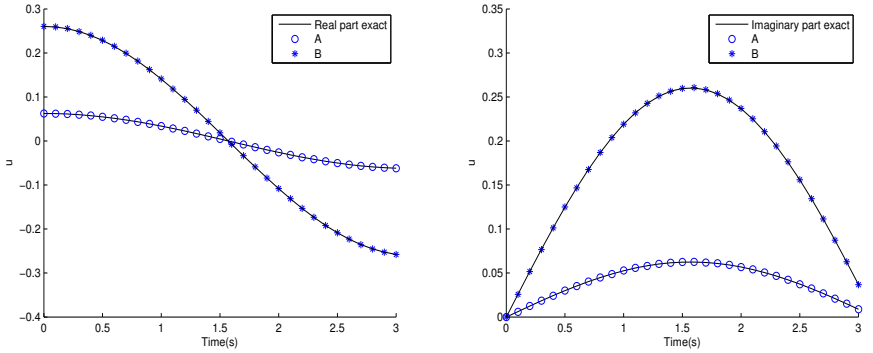

FIGURE 6. Temperature evolution at points A(0.5,0.5) and B($\frac{10}{14}$,$\frac{10}{14}$), Real (left) Imaginary (right) parts for example 4.2.

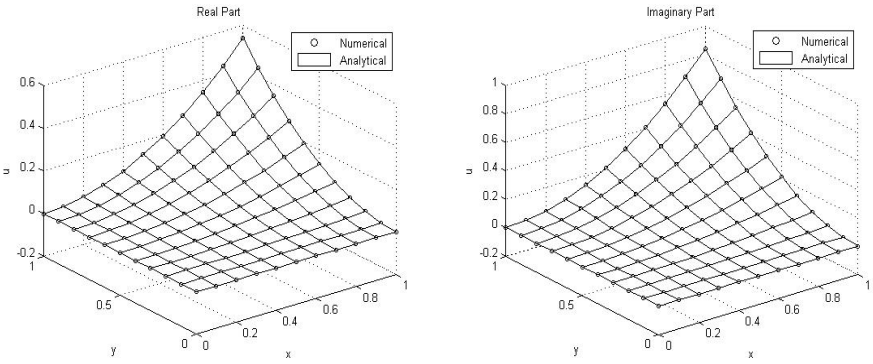

FIGURE 7. Real (left) and Imaginary (right) parts of numerical and exact solutions of example 4.2 at $t = 1s$.

TABLE 6. The Maximum errors for irregular 441 nodes Fig.2 (right) with $dt = 0.001$ for example 4.2.

time(s)	Max-error (ϵ)	CPU Time (s)
0.10	1.20(−3)	7
0.50	4.29(−2)	10
1.00	5.96(−2)	14

As in the previous example, in Table 5 we report the L_∞ error for the real and imaginary parts and the maximum error of solution for different numbers of uniformed nodes (see Fig.5, left) in the space variable with $t = 1$ and $dt = 0.01$. Also, the last column of this table show the c-order of our scheme.

Table 6, shows the maximum error and CPU times in irregular domain (Fig.5 (right)) with 441 nodes and $t = 0.1, 0.5, 1$, $dt = 0.001$ for this example. Fig.6, shows a comparison between the numerical results obtained by MK-MLPG scheme with the exact solution of Real (left) and Imaginary (right) parts of the temperature at two points $A(0.5, 0.5)$ and $B(\frac{10}{14}, \frac{10}{14})$ for different times. The graphs of real and imaginary parts of the estimated and analytical solutions at time $t = 1$ with $dt = 0.001$ are depicted in Fig.7.

4.3. **Example 4.3.** Consider Eq (1) in a square domain $\Omega = [0, 1] \times [0, 1]$ with the following potential function

$$\omega(x, y) = 3 - 2\tanh^2 x - 2\tanh^2 y. \tag{33}$$

The analytical solution of this equation is [14]

$$u(x, y, t) = \frac{i \exp(it)}{\cosh(x)\cosh(y)}. \tag{34}$$

The initial and boundary conditions can be found from the analytical solution as

$$u(x, y, 0) = \frac{i}{\cosh(x)\cosh(y)}, \quad (x, y) \in \Omega, \tag{35}$$

and

$$\begin{aligned} u(0, y, t) &= \frac{i \exp(it)}{\cosh(y)}, \qquad u(1, x, t) = \frac{i \exp(it)}{\cosh(1)\cosh(y)}, \\ u(x, 0, t) &= \frac{i \exp(it)}{\cosh(x)}, \qquad u(x, 1, t) = \frac{i \exp(it)}{\cosh(1)\cosh(x)}. \end{aligned} \tag{36}$$

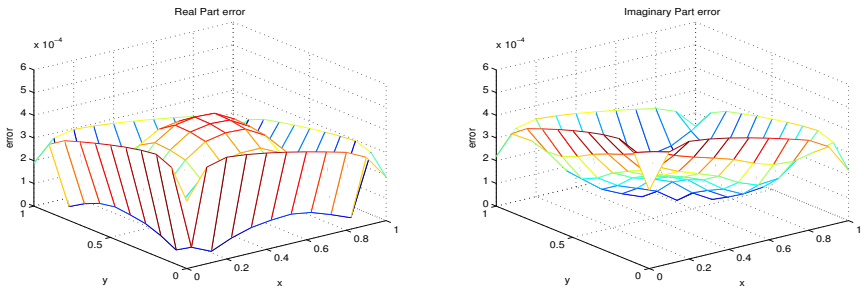

FIGURE 8. Plot of the absolute errors of Real part (left) and Imaginary part (right) at $t = 4s$ for example 4.3.

Here, we use 11×11 regular nodes ($dx = dy = 0.1$) in the problem domain as shown in Fig.2 (left). Table 7, compares the numerical results with RBF method [13] and shows the superiority of our algorithm. In this table, the first column is time and two next columns are L_∞ of real part errors by MK-MLPG and RBF methods [13], columns four and five represent the L_∞ of imaginary part errors by MK-MLPG and RBF algorithms [13] respectively, the last two columns are maximum errors by proposed method and RBF scheme [13]. Table 8, show the MK-MLPG results and the exact solutions on some selected points at $t = 4s$. This table reveals that, the MK-MLPG results are in a good agreement with the exact solution even at the large values of t.

Also, the graph of absolute error between the analytical and numerical (error in Real (left) and Imaginary (right) parts) solutions at $t = 4s$ are depicted in Fig.8.

TABLE 7. L_∞ and Maximum error (ϵ) for example 4.3 with $dt = 0.001$.

	L_∞ *Re part*		L_∞ *Im part*		Max-error(ϵ)	
t	MK-MLPG	RBF[13]	MK-MLPG	RBF[13]	MK-MLPG	RBF[13]
0.1	2.33(−6)	2.44(−5)	2.38(−5)	2.99(−5)	2.97(−5)	3.01(−5)
0.3	1.23(−6)	2.95(−5)	2.48(−6)	2.38(−5)	2.74(−6)	3.45(−5)
0.5	2.17(−5)	2.74(−5)	3.15(−5)	3.40(−5)	3.21(−5)	3.67(−5)
0.7	1.09(−5)	2.54(−5)	1.65(−5)	1.86(−5)	1.67(−5)	3.16(−5)
1	1.36(−5)	2.94(−5)	1.40(−5)	2.42(−5)	1.96(−5)	3.32(−5)

TABLE 8. Numerical and exact solutions for some selected points at time $t = 4s$ in example 4.3.

(x, y)	Exact	Numerical
$(0.1, 0.1)$	$0.7493 - 0.6472i$	$0.7493 - 0.6476i$
$(0.2, 0.2)$	$0.7273 - 0.6282i$	$0.7272 - 0.6285i$
$(0.3, 0.3)$	$0.6926 - 0.5982i$	$0.6923 - 0.5984i$
$(0.4, 0.4)$	$0.6475 - 0.5593i$	$0.6472 - 0.5595i$
$(0.5, 0.5)$	$0.5952 - 0.5141i$	$0.5948 - 0.5142i$
$(0.6, 0.6)$	$0.5385 - 0.4651i$	$0.5381 - 0.4651i$
$(0.7, 0.7)$	$0.4804 - 0.4149i$	$0.4801 - 0.4148i$
$(0.8, 0.8)$	$0.4231 - 0.3654i$	$0.4229 - 0.3654i$
$(0.9, 0.9)$	$0.3685 - 0.3183i$	$0.3685 - 0.3184i$

4.4. **Example 4.4.** In this case, we consider $\omega(x, y) = 0$ in Eq (1) with the following initial condition

$$u(x, y, 0) = e^{-(x^2+y^2)-ik_0x}, \tag{37}$$

which generates transient Gaussian distribution as follows

$$u(x, y, t) = \frac{i}{i - 4t} e^{-\left((x^2+y^2)+ik_0x+ik_0^2t)/(i-4t)\right)}, \tag{38}$$

initially centred at $(0, 0)$ and then moving along the negative $x-$direction as time progresses [18]. Here, k_0 is the wave number and we take it 2.5. Hence, this is an open domain problem, and we limited our computational domain to the square $2.5 \leq x, y \leq 2.5$. The boundary conditions from the exact solution of Eq (25) will be determined as in the previous examples.

TABLE 9. L_∞ and Maximum error with $dt = 0.001$ for example 4.4.

	L_∞ *Re part*		L_∞ *Im part*		Max-error(ϵ)	
t	MK-MLPG	RBF[13]	MK-MLPG	RBF[13]	MK-MLPG	RBF[13]
0.10	1.65(−5)	9.58(−5)	2.48(−4)	1.37(−4)	1.59(−4)	1.81(−4)
0.25	2.73(−3)	3.00(−3)	2.59(−3)	2.78(−3)	2.18(−3)	2.29(−3)
0.50	2.57(−3)	3.69(−3)	2.39(−3)	3.60(−3)	1.66(−3)	8.26(−3)
0.75	2.79(−3)	3.69(−3)	3.91(−3)	4.34(−3)	1.94(−2)	1.60(−2)

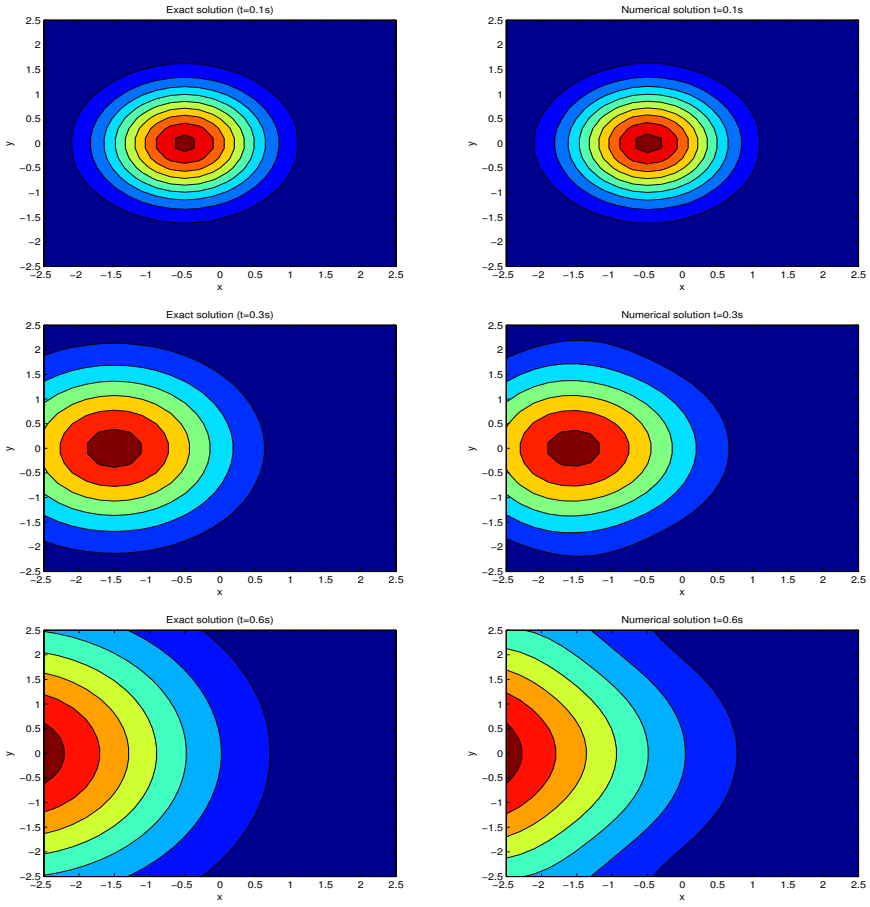

FIGURE 9. Contour plots of modulus for the exact solutions (left) and numerical solutions (right) at $t = 0.1, 0.3$ and 0.6, for example 4.4.

The numerical results of MK-MLPG and RBF [13] methods for this example are reported in Table 9. These results reveal that our method for solving this problem is superior to RBF method. The contour plot of modulus of exact and MK-MLPG solutions at $t = 0.1, 0.3$ and 0.6 with 26×26 uniformed nodes ($dx = dy = 0.2$), $dt = 0.001$, are depicted in Fig.9.

4.5. **Example 4.5.** Finally, we consider the following two-dimensional time-dependent Schrödinger equation

$$-i\frac{\partial u}{\partial t} = \nabla^2 u - u, \quad (x, y) \in [0, 1] \times [0, 1], \tag{39}$$

TABLE 10. MK-MLPG and FD[14] solutions for some selected points at time $t = 1s$ in example 4.5.

(x, y)	MK-MLPG	FD[14]
$(0.1, 0.1)$	$0.5432 + 0.8451i$	$0.5430 + 0.8456i$
$(0.2, 0.2)$	$0.5518 + 0.8581i$	$0.5511 + 0.8583i$
$(0.3, 0.3)$	$0.5647 + 0.8798i$	$0.5648 + 0.8796i$
$(0.4, 0.4)$	$0.5849 + 0.9095i$	$0.5841 + 0.9096i$
$(0.5, 0.5)$	$0.6098 + 0.9483i$	$0.6092 + 0.9488i$
$(0.6, 0.6)$	$0.6404 + 0.9977i$	$0.6405 + 0.9975i$
$(0.7, 0.7)$	$0.6782 + 1.0561i$	$0.6781 + 1.0562i$
$(0.8, 0.8)$	$0.7225 + 1.1251i$	$0.7226 + 1.1254i$
$(0.9, 0.9)$	$0.7741 + 1.2055i$	$0.7743 + 1.2059i$

subject to the initial condition

$$u(x, y, 0) = \cosh(x)\cosh(y), \tag{40}$$

and the boundary conditions

$$\begin{aligned} &u(0, y, t) = \exp(it)\cosh(y), \qquad u(1, x, t) = \exp(it)\cosh(1)\cosh(y), \\ &u(x, 0, t) = \exp(it)\cosh(x), \qquad u(x, 1, t) = \exp(it)\cosh(x)\cosh(1). \end{aligned} \tag{41}$$

The exact solution for this example is

$$u(x, y, 0) = \exp(it)\cosh(x)\cosh(y).$$

This problem will be solved by our method and Finite difference (FD) method (5,1) [14]. We apply 121 uniform nodes in the domain $[0, 1] \times [0, 1]$ by using $t = 1s$ and $dt = 0.01$. Table 10 shows the results of these methods for some nodes. One can see that, the results of MK-MLPG and FD [14] are almost the same. The Maximum absolute errors between exact solution and MK-MLPG in Real and Imaginary parts for different times are shown in columns two and four. Also, the errors of FD [14] are in columns three and five . Moreover, the last two columns of this table show the Maximum absolute errors for MK-MLPG and FD [14] methods.

In Table 12, we report the $c-$order in time variable for our method. These tables reveal that, the MK-MLPG results are in a good agreement with the FD [14] and exact solution at different values of t.

TABLE 11. L_∞ between MK-MLPG and FD[14] methods with $dt = 0.01$ for example 4.5.

	L_∞ *Re part*		L_∞ *Im part*		Max-error(ϵ)	
t	MK-MLPG	FD[14]	MK-MLPG	FD[14]	MK-MLPG	FD[14]
0.75	4.92(−3)	5.41(−3)	4.01(−3)	3.54(−3)	1.69(−3)	1.81(−3)
1.00	8.26(−3)	8.37(−3)	8.58(−3)	8.53(−3)	1.81(−4)	1.65(−4)
2.00	7.41(−4)	3.12(−3)	2.32(−3)	2.77(−3)	2.65(−4)	2.94(−4)
3.00	2.61(−4)	1.45(−3)	4.55(−4)	2.81(−3)	3.12(−4)	3.50(−4)

TABLE 12. The maximum error(ϵ) and c-order for 221 regular nodes Fig.2 (left) for $t = 1s$ in example 4.5.

dt	Max-error (ϵ)	c-error
0.08	5.51(−3)	–
0.04	2.63(−3)	1.07
0.02	6.08(−4)	2.11
0.01	1.81(−4)	1.74

5. Conclusion

In this paper, the numerical solutions of the two dimensional time-dependent Schrödinger equation are discussed. We proposed a numerical method to solve this partial differential equation by using the moving Kriging interpolation based on the meshless local Petrov-Galerkin (MK-MLPG) method. This method has been successfully applied to this equation. Numerical results were obtained for five examples and were compared to their analytical and RBF solutions in some examples and by FD solution in the last example, to confirm the efficiency and capability of this scheme.

References

[1] S. Abbasbandy and A. Shirzadi, MLPG method for two-dimensional diffusion equation with Neumanns and non-classical boundary conditions. Applied Numerical Mathematics, Vol. 61, 170-180 (2011)

[2] W. T. Ang and K. C. Ang, A dual reciprocity boundary element solution of a generalized nonlinear Schrödinger equation. Numer Methods Partial differential Equations, Vol. 20, 843-854 (2004)

[3] X. Antoine, C. Besse and V. Mouysett, Numerical schemes for the simulation of the two-dimensional Schrödinger equation using non-reflecting boundary conditions. Math Comput, Vol. 73, 1779-1799 (2004)

[4] S. N. Atluri and T. Zhu, A new meshless local Petrov Galerkin (MLPG)approach in computational mechanics, Comput. Mech, Vol. 22, 117-127 (1998)

[5] S. N. Atluri and T. L. Zhu, A new meshless local Petrov-Galerkin (MLPG) approach to nonlinear problems in computer modeling and simulation. Comput Model Simul Eng, Vol. 3, Number 3, 187-196 (1998)

[6] S. N. Atluri and S. Shen, The meshless local Petrov-Galerkin (MLPG) method: a simple and less-costly alternative to the finite element and boundary element methods. CMES: Comput Modeling Eng Sci, Vol. 3, Number 1, 11-51 (2002)

[7] S. N. Atluri and S. Shen, The meshless local Petrov-Galerkin (MLPG) method. Tech Science Press (2002)

[8] S. N. Atluri, The Meshless Method (MLPG) for Domain and BIE Discretizations. Tech Science Press (2004)

[9] T. Belytschko and Y. Y. Lu and L.Gu, Element-free Galerkin methods. International Journal for Numerical Methods in Engineering,Vol. 37, 229-256 (1994)

[10] L. Biao and C. Yong, A Truncated *Painlev'e* Expansion and Exact Analytical Solutions for the Nonlinear Schrödinger Equation with Variable Coefficients. Z. Naturforsch, Vol. 60, 768 - 774 (2005)

[11] L. Chen and K. M. Liew, A local Petrov-Galerkin approach with moving Kriging interpolation for solving transient heat conduction problems, Int. J. Numer. Meth, Vol. 47, 455-467 (2011)

[12] B. Dai , B. Zheng , Q. Liang and L. Wang, Numerical solution of transient heat conduction problems using improved meshless local Petrov-Galerkin method. Applied Mathematics and Computation, Vol. 219, 10044-10052 (2013)

[13] M. Dehghan and A. Shokri, A numerical method for two-dimensional Schrödinger equation using collocation and radial basis functions. Comput Math Appl, Vol. 54, 136-146 (2007)

[14] M. Dehghan, Finite difference procedures for solving a problem arising in modeling and design of certain optoelectronic devices, Math. Comput. Simulation, Vol. 71, 16-30 (2006)

[15] R. A. Gingold and J. J. Monaghan, Smoothed particle hydrodynamics: theory and application to non-spherical stars. Mon. Not. Roy. Astron. Soc, Vol. 181, 375-389 (1977)

[16] L. Gu, Moving Kriging interpolation and element-free Galerkin method, Int. J Numer. Meth. Eng, Vol. 56, 1-11 (2003)

[17] D. A. Hu, S. Y. Long, K. Y. Liu and G. Y. Li, A modified meshless local Petrov-Galerkin method to elasticity problems in computer modelling and simulation, Eng. Anal. Bound. Elem, Vol. 30, 399-404 (2006)

[18] J. C. Kalita , P. Chhabra and S. Kumar, A semi-discrete higher order compact scheme for the unsteady two-dimensional Schrodinger equation. J Comput Appl Math, Vol. 197, 141-149 (2006)

[19] G. R. Liu and Y. T. Gu, A point interpolation method for two dimensional solid. Int. J. Numer. Meth. Eng, Vol. 50, Number 4, 937-951 (2001)

[20] G. R. Liu and Y. T. Gu, A local radial point interpolation method (LRPIM) for free vibration analyses of 2-D solids. J Sound Vib, Vol. 246, 29-46 (2001)

[21] W. K. Liu, S. Jun and Y. F. Zhang, Reproducing kernel particle methods. Int. J. Nume. Methods Fluids, Vol. 20, 1081-1106 (1995)

[22] D. Mirzaei and M. Dehghan, Meshless local Petrov-Galerkin (MLPG) approximation to the two dimensional sine-Gordon equation, J. Comput. Appl. Math, Vol. 233, 2737-2754 (2010)

[23] A. Mohebbi , M. Dehghan The use of compact boundary value method for the solution of two-dimensional Schrodinger equation. J. Comput. Appl. Math. Vol. 225, 124-134 (2009)

[24] J. Sacks, W. J. Welch, T. J. Mitchell and H. P. Wynn, Design and analysis of computer experiments. Stat Sci, Vol 4. 409-435 (1989)

[25] M. Subasi, On the finite-difference schemes for the numerical solution of two dimensional Schrödinger equation, Numer. Methods Partial Differential Equations, Vol. 18, 752-758 (2002)

[26] J. G. Wang and G. R. Liu, A point interpolation meshless method based on radial basis functions. Int. J. Numer. Meth. Eng, Vol. 54, Number 11, 1623-1648 (2002)

[27] J. G. Wang and G. R. Liu, Radial point interpolation method for elastoplastic problems. In: Proceedings of the 1st international conference on structural stability and dynamics, Taipei, Taiwan, 703-708 (2000)