Available online at www.sciencedirect.com
ScienceDirect
Computers and Mathematics with Applications 54 (2007) 136–146
An International Journal
computers & mathematics with applications
www.elsevier.com/locate/camwa

A numerical method for two-dimensional Schrödinger equation using collocation and radial basis functions

Mehdi Dehghan*, Ali Shokri

Department of Applied Mathematics, Faculty of Mathematics and Computer Science, Amirkabir University of Technology, No. 424, Hafez Ave., Tehran, Iran

Received 5 October 2006; received in revised form 28 December 2006; accepted 22 January 2007

Abstract

In this paper, we propose a numerical scheme to solve the two-dimensional (2D) time-dependent Schrödinger equation using collocation points and approximating the solution using multiquadrics (MQ) and the Thin Plate Splines (TPS) Radial Basis Function (RBF). The scheme works in a similar fashion as finite-difference methods. The results of numerical experiments are presented, and are compared with analytical solutions to confirm the good accuracy of the presented scheme.

Keywords: Two-dimensional Schrödinger equation; Collocation; Radial Basis Function (RBF); Multiquadrics (MQ); Thin Plate Splines (TPS)

1. Introduction

This paper is devoted to the numerical computation of the two-dimensional (2D) time-dependent Schrödinger equation:

$$-\mathrm{i}\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \omega(x, y)u, \tag{1.1}$$

in some continuous domain with suitable initial and Dirichlet boundary conditions and an arbitrary potential function $\omega(x, y)$. The solution for such a kind of equation is of fundamental importance in quantum mechanics (modeling of quantum devices [1]), electromagnetic wave propagation [2], underwater acoustics (paraxial approximations to the wave equation [3]) and design of certain optoelectronic devices [4] as it models an electromagnetic wave equation in a two-dimensional weakly guiding structure. It has also found its application in various quantum dynamics calculations [5,6].

Such a linear Schrödinger equation has also been represented in a hydrodynamical form. In this formulation, system (1.1) is replaced by a system of non-linear equations in terms of particle density and velocity potential, by separating the real and imaginary parts of a general solution, called a quantum hydrodynamic equation (QHD), which is formally analogous to the equations of irrotational motion in a classical fluid [7,8].

* Corresponding author. Tel.: +98 21 6406322; fax: +98 21 6497930.
E-mail addresses: mdehghan@aut.ac.ir (M. Dehghan), shokri.a@gmail.com (A. Shokri).

0898-1221/$ - see front matter
doi:10.1016/j.camwa.2007.01.038

There have been attempts to develop numerical schemes for equations similar to system (1.1). Several second-order spatial accurate schemes including (1, 5) fully explicit scheme, (1, 9) fully explicit method, (5, 1) fully implicit technique, (5, 5) fully implicit formula, Crank–Nicolson finite-difference procedure, alternating direction implicit (ADI) method, the Barakat and Clark explicit finite-difference scheme were discussed by Dehghan [9]. The authors of [8,10] studied models similar to the present problem but they used finite-difference techniques.

Finite-difference methods are known as the first technique for solving partial differential equations (PDEs). Even though the alternating direction implicit methods are very effective for solving various kinds of partial differential equations, conditional stability of explicit finite-difference procedures [11] and the need to use a large amount of CPU time in implicit finite-difference schemes [12] limit the applicability of these methods. Furthermore, these methods provide the solution of the problem on mesh points only, and the accuracy of the techniques is reduced in non-smooth and non-regular domains.

In the current work we investigate a different approach to find the solution of Eq. (1.1). This paper presents a numerical scheme to solve the two-dimensional (2D) time-dependent Schrödinger equation [13] using the collocation method and approximating the solution directly using multiquadrics and the thin plate splines radial basis function. The scheme is similar to finite-difference methods. To test the robustness, accuracy and efficiency of the scheme, it is applied to three examples having analytical solutions. Our results exhibit good comparison with analytical solutions.

The layout of the paper is as follows. In Section 2 we show how we use the radial basis functions to approximate the solution. In Section 3 we apply the method on the two-dimensional (2D) time-dependent Schrödinger equation. The results of numerical experiments are presented in Section 4. Section 5 is dedicated to a brief conclusion. Finally some references are introduced at the end.

2. Radial basis function approximation

To avoid mesh generation, in recent years meshless techniques have attracted the attention of researchers. In a meshless (mesh-free) method a set of scattered nodes is used instead of meshing the domain of the problem. Some meshless schemes are the element-free Galerkin method, the reproducing kernel particle, and the local point interpolation. For more descriptions see [14] and references therein.

In the last 20 years, the radial basis functions method has become known as a powerful tool for scattered data interpolation problem. The use of radial basis functions as a meshless procedure for the numerical solution of partial differential equations is based on the collocation scheme. Due to the collocation technique, this method does not need to evaluate any integral. The main advantage of numerical procedures which use radial basis functions over traditional techniques is the meshless property of these methods. Radial basis functions are used actively for solving partial differential equations.

In the last decade, the development of radial basis functions (RBFs) as a truly meshless method for approximating the solutions of PDEs has drawn the attention of many researchers in science and engineering. One of the domain-type meshless methods, the so-called Kansa's method developed by Kansa in 1990 [15,16], is obtained by directly collocating the RBFs, particularly the multiquadrics (MQ), for the numerical approximation of the solution.

Kansa's method was recently extended to solve various ordinary and partial differential equations including the 1D nonlinear Burgers equation [17] with shock wave, shallow water equations for tide and currents simulation [18], heat transfer problems [19], and free boundary problems [20,21].

In most of the cases, the accuracy of the RBF solution, however, depends heavily on the choice of a shape parameter c in the MQ or Gaussian basis functions. The choice of this optimal value is still under intensive investigation. Many authors have investigated the shape parameter. For instance, Carlson and Foley [22] found that c is problem dependent. Tarwater [23] found that, by increasing c, the Root-Mean-Square (RMS) of the error dropped to a minimum then increased sharply afterwards. In general, as c increases the system of equations to be solved becomes ill-conditioned.

The approximation of a distribution $u(\mathbf{x})$, using radial basis functions, may be written as a linear combination of N radial functions; usually it takes the following form:

$$u(\mathbf{x}) \simeq \sum_{j=1}^{N} \lambda_j \varphi(\mathbf{x}, \mathbf{x}_j) + \psi(\mathbf{x}) \quad \text{for } \mathbf{x} \in \Omega \subset \mathbb{R}^d, \tag{2.1}$$

where N is the number of data points, $\mathbf{x} = (x_1, x_2, \ldots, x_d)$, d is the dimension of the problem, the λ's are coefficients to be determined and φ is the radial basis function. Eq. (2.1) can be written without the additional polynomial ψ. In

that case φ must be *unconditionally positive definite* to guarantee the solvability of the resulting system (e.g. Gaussian or inverse multiquadrics). However, ψ is usually required when φ is *conditionally positive definite*, i.e., when φ has a polynomial growth towards infinity. Examples are thin plate splines and multiquadrics. We will use multiquadrics and thin plate splines for the numerical scheme introduced in Section 3. These are defined as:

(i) Multiquadrics

$$\varphi(\mathbf{x}, \mathbf{x}_j) = \varphi(r_j) = \sqrt{r_j^2 + c^2}, \quad c > 0, \tag{2.2}$$

(ii) Generalized Thin Plate Splines (TPS)

$$\varphi(\mathbf{x}, \mathbf{x}_j) = \varphi(r_j) = r_j^{2m} \log(r_j), \quad m = 1, 2, 3, \ldots, \tag{2.3}$$

where $r_j = \|\mathbf{x} - \mathbf{x}_j\|$ is the Euclidean norm.

Since φ given by (2.2) is C^∞ continuous, we can use it directly. But φ in (2.3) is C^{2m-1} continuous, so higher-order thin plate splines must be used for higher-order partial differential operators. For the Schrödinger equation, $m = 2$ is used for thin plate splines (i.e. second-order thin plate splines).

If $\mathcal{P}_q^d$ denotes the space of d-variate polynomials of order not exceeding than q, and letting the polynomials $P_1, \ldots, P_m$ be the basis of $\mathcal{P}_q^d$ in $\mathbb{R}^d$, then the polynomial $\psi(\mathbf{x})$, in Eq. (2.1), is usually written in the following form:

$$\psi(\mathbf{x}) = \sum_{i=1}^{m} \zeta_i P_i(\mathbf{x}), \tag{2.4}$$

where $m = (q - 1 + d)!/(d!(q - 1)!)$.

To determine the coefficients $(\lambda_1, \ldots, \lambda_N)$ and $(\zeta_1, \ldots, \zeta_m)$, the collocation method is used. However, in addition to the N equations resulting from collocating (2.1) at the N points, an extra m equations are required. This is ensured by the m conditions for (2.1),

$$\sum_{j=1}^{N} \lambda_j P_i(\mathbf{x}_j) = 0, \quad i = 1, \ldots, m. \tag{2.5}$$

In a similar representation as (2.1), for any linear partial differential operator $\mathcal{L}$, $\mathcal{L}u$ can be approximated by

$$\mathcal{L}u(\mathbf{x}) \simeq \sum_{j=1}^{N} \lambda_j \mathcal{L}\varphi(\mathbf{x}, \mathbf{x}_j) + \mathcal{L}\psi(\mathbf{x}). \tag{2.6}$$

3. The two-dimensional Schrödinger equation

Let us consider the following two-dimensional time-dependent Schrödinger equation:

$$-\mathrm{i}\frac{\partial u}{\partial t} = \nabla^2 u + \omega(x, y)u, \quad (x, y) \in \Omega \subset \mathbb{R}^2, \ 0 < t \leq T, \tag{3.1}$$

with the initial condition

$$u(x, y, 0) = u_0(x, y), \quad (x, y) \in \Omega, \tag{3.2}$$

and Dirichlet boundary condition

$$u(x, y, t) = g(x, y), \quad (x, y) \in \partial\Omega, \ 0 < t \leq T, \tag{3.3}$$

where u_0, g and ω are known functions, ∇^2 is equal to $(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2})$, and the function u is unknown.

First, let us discretize (3.1) according to the following θ-weighted scheme:

$$-\mathrm{i}\frac{u(x,y,t+\mathrm{d}t)-u(x,y,t)}{\mathrm{d}t} = \theta\left[\nabla^2 u(x,y,t+\mathrm{d}t)+\omega(x,y)u(x,y,t+\mathrm{d}t)\right] + (1-\theta)\left[\nabla^2 u(x,y,t)+\omega(x,y)u(x,y,t)\right], \tag{3.4}$$

where $0 \le \theta \le 1$, and $\mathrm{d}t$ is the time step size. Rearranging (3.4), using the notation $u^n = u(x,y,t^n)$ where $t^n = t^{n-1} + \mathrm{d}t$, we obtain

$$-\mathrm{i}u^{n+1} - \theta\mathrm{d}t\left[\nabla^2 u^{n+1} + \omega(x,y)u^{n+1}\right] = -\mathrm{i}u^n + (1-\theta)\mathrm{d}t\left[\nabla^2 u^n + \omega(x,y)u^n\right]. \tag{3.5}$$

Assuming that there are a total of $(N-3)$ interpolation points, $u(x,y,t^n)$ can be approximated by

$$u^n(x,y) \simeq \sum_{j=1}^{N-3} \lambda_j^n \varphi(r_j) + \lambda_{N-2}^n x + \lambda_{N-1}^n y + \lambda_N^n. \tag{3.6}$$

To determine the interpolation coefficients $(\lambda_1, \lambda_2, \ldots, \lambda_{N-2}, \lambda_{N-1}, \lambda_N)$, the collocation method is used by applying (3.6) at every point (x_i, y_i), $i = 1, 2, \ldots, N-3$. Thus we have

$$u^n(x_i,y_i) \simeq \sum_{j=1}^{N-3} \lambda_j^n \varphi(r_{ij}) + \lambda_{N-2}^n x_i + \lambda_{N-1}^n y_i + \lambda_N^n, \tag{3.7}$$

where $r_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$. The additional conditions due to (2.5) are written as

$$\sum_{j=1}^{N-3} \lambda_j^n = \sum_{j=1}^{N-3} \lambda_j^n x_j = \sum_{j=1}^{N-3} \lambda_j^n y_j = 0. \tag{3.8}$$

Writing (3.7) together with (3.8) in a matrix form we have

$$[u]^n = \mathbf{A}[\lambda]^n, \tag{3.9}$$

where $[u]^n = \begin{bmatrix} u_1^n u_2^n \ldots u_{N-3}^n & 0 & 0 & 0 \end{bmatrix}^{\mathrm{T}}$, $[\lambda]^n = \begin{bmatrix} \lambda_1^n & \lambda_2^n & \ldots & \lambda_N^n \end{bmatrix}^{\mathrm{T}}$ and $\mathbf{A} = [a_{ij}, 1 \le i, j \le N]$ is given by

$$\mathbf{A} = \begin{bmatrix} \varphi_{11} & \cdots & \varphi_{1(N-3)} & x_1 & y_1 & 1 \\ \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ \varphi_{(N-3)1} & \cdots & \varphi_{(N-3)(N-3)} & x_{N-3} & y_{N-3} & 1 \\ x_1 & \cdots & x_{N-3} & 0 & 0 & 0 \\ y_1 & \cdots & y_{N-3} & 0 & 0 & 0 \\ 1 & \cdots & 1 & 0 & 0 & 0 \end{bmatrix}. \tag{3.10}$$

Assuming that there are $p < (N-3)$ internal points and $(N-3-p)$ boundary points, then the $(N \times N)$ matrix $\mathbf{A}$ can be split into: $\mathbf{A} = \mathbf{A}_d + \mathbf{A}_b + \mathbf{A}_e$, where

$$\begin{aligned} &\mathbf{A}_d = [a_{ij} \text{ for } (1 \le i \le p, 1 \le j \le N) \text{ and 0 elsewhere}], \\ &\mathbf{A}_b = [a_{ij} \text{ for } (p+1 \le i \le N-3, 1 \le j \le N) \text{ and 0 elsewhere}], \\ &\mathbf{A}_e = [a_{ij} \text{ for } (N-2 \le i \le N, 1 \le j \le N) \text{ and 0 elsewhere}]. \end{aligned} \tag{3.11}$$

Using the notation $\mathcal{L}\mathbf{A}$ to designate the matrix of the same dimension as $\mathbf{A}$ and containing the elements $\tilde{a}_{ij}$, where $\tilde{a}_{ij} = \mathcal{L}a_{ij}$, $1 \le i, j \le N$, then Eq. (3.5) together with (3.3) can be written, in matrix form, as

$$[-\mathrm{i}A_d + B][\lambda]^{n+1} = [-\mathrm{i}A_d + C][\lambda]^n + [G]^{n+1}, \tag{3.12}$$

where

$$B = -\theta\delta t\left(\nabla^2 A_d + W * A_d\right) + A_b + A_e, \qquad C = (1-\theta)\mathrm{d}t\left(\nabla^2 A_d + W * A_d\right),$$

$$[G]^n = \begin{bmatrix} 0 & \dots & 0 & G^n_{p+1} & \dots & G^n_{N-3} & 0 & 0 & 0 \end{bmatrix}^{\mathrm{T}},$$

and $W = \begin{bmatrix} \omega_1 & \omega_2 & \dots & \omega_p & 0 & \dots & 0 \end{bmatrix}^{\mathrm{T}}$. The symbol "$*$" means that the ith component of vector W is multiplied to all components of ith row of matrix A_d. Eq. (3.12) is obtained by combining (3.5), which applies to the domain points, while (3.3) applies to the boundary points.

Assume that $[\lambda]^n = [\lambda_r]^n + i[\lambda_i]^n$ and $[G]^n = [G_r]^n + i[G_i]^n$. The r and i indices mean the real part and imaginary part of complex vectors, respectively. Using this notation, (3.12) can be written as

$$\begin{aligned} &B[\lambda_r]^{n+1} + A_d[\lambda_i]^{n+1} + \mathrm{i}\left(-A_d[\lambda_r]^{n+1} + B[\lambda_i]^{n+1}\right) \\ &\quad = C[\lambda_r]^n + A_d[\lambda_i]^n + [G_r]^{n+1} + \mathrm{i}\left(-A_d[\lambda_r]^n + C[\lambda_i]^n + [G_i]^{n+1}\right). \end{aligned} \tag{3.13}$$

The complex equation (3.13) can be [9] rewritten in the following real variable form:

$$\begin{bmatrix} B & A_d \\ -A_d & B \end{bmatrix} \begin{bmatrix} \lambda_r \\ \lambda_i \end{bmatrix}^{n+1} = \begin{bmatrix} C & A_d \\ -A_d & C \end{bmatrix} \begin{bmatrix} \lambda_r \\ \lambda_i \end{bmatrix}^{n} + \begin{bmatrix} G_r \\ G_i \end{bmatrix}^{n+1}. \tag{3.14}$$

Note that (3.14) is obtained by equalizing the real and imaginary parts of the right and left sides.

Thus, the solution of the complex system has been reduced to solving the real variable system. Since the coefficient matrix is unchanged in time steps, we use the LU factorization to the coefficient matrix only once and use this factorization in our algorithm.

Remark. Although Eq. (3.12) is valid for any value of $\theta \in [0, 1]$, we will use $\theta = 1/2$ (the famous Crank–Nicolson scheme).

4. Numerical experiments

In order to study the validity and effectiveness of the proposed scheme, it is applied to three examples, two with the presence of the potential function, and the third one without it.

4.1. Example 1

We consider Eq. (3.1) in the region $0 \le x, y \le 1$ with potential function

$$\omega(x, y) = 3 - 2tgh^2 x - 2tgh^2 y. \tag{4.1}$$

The analytical solution of the equation is [9]

$$u(x, y, t) = \frac{\mathrm{i}\exp(\mathrm{i}t)}{\cosh(x)\cosh(y)}. \tag{4.2}$$

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

$$u(x, y, 0) = \frac{\mathrm{i}}{\cosh(x)\cosh(y)}, \quad 0 \le x, y \le 1, \tag{4.3}$$

and

$$\begin{aligned} u(0, y, t) &= \frac{\mathrm{i}\exp(\mathrm{i}t)}{\cosh(y)}, \qquad u(1, y, t) = \frac{\mathrm{i}\exp(\mathrm{i}t)}{\cosh(1)\cosh(y)}, \\ u(x, 0, t) &= \frac{\mathrm{i}\exp(\mathrm{i}t)}{\cosh(x)}, \qquad u(x, 1, t) = \frac{\mathrm{i}\exp(\mathrm{i}t)}{\cosh(x)\cosh(1)}. \end{aligned} \tag{4.4}$$

Table 1 presents the maximum absolute error for the real part and imaginary part of solution at different times up to $t = 1.0$, using multiquadrics and the thin plate spline as the radial basis function for the discussed scheme. These

Table 1
Maximum absolute error and Max error of multiquadrics and thin plate spline based scheme at different times for example 1 with $dx = dy = 0.1$, $dt = 0.001$, and $c = 0.7$ for MQ

t	Maximum absolute error		Max error ϵ	CPU time used
	Real part	Imaginary part		
0.1	2.4407×10^{-5}	2.9974×10^{-5}	3.0030×10^{-5}	0
	7.8895×10^{-5}	9.8635×10^{-5}	9.9331×10^{-5}	0
0.3	2.9466×10^{-5}	2.3861×10^{-5}	3.4597×10^{-5}	1
	1.0368×10^{-4}	8.6876×10^{-5}	1.2952×10^{-4}	1
0.5	2.7468×10^{-5}	3.4044×10^{-5}	3.6720×10^{-5}	2
	7.7545×10^{-5}	9.1676×10^{-5}	1.0837×10^{-4}	2
0.7	2.5495×10^{-5}	1.8694×10^{-5}	3.1614×10^{-5}	3
	8.9137×10^{-5}	7.7454×10^{-5}	1.0639×10^{-4}	3
1.0	2.9444×10^{-5}	2.4222×10^{-5}	3.3253×10^{-5}	4
	1.0626×10^{-4}	9.3425×10^{-5}	1.2058×10^{-4}	4

For every value of t, the first and second rows of data correspond to the use of MQ and TPS as the radial basis function in the scheme, respectively.

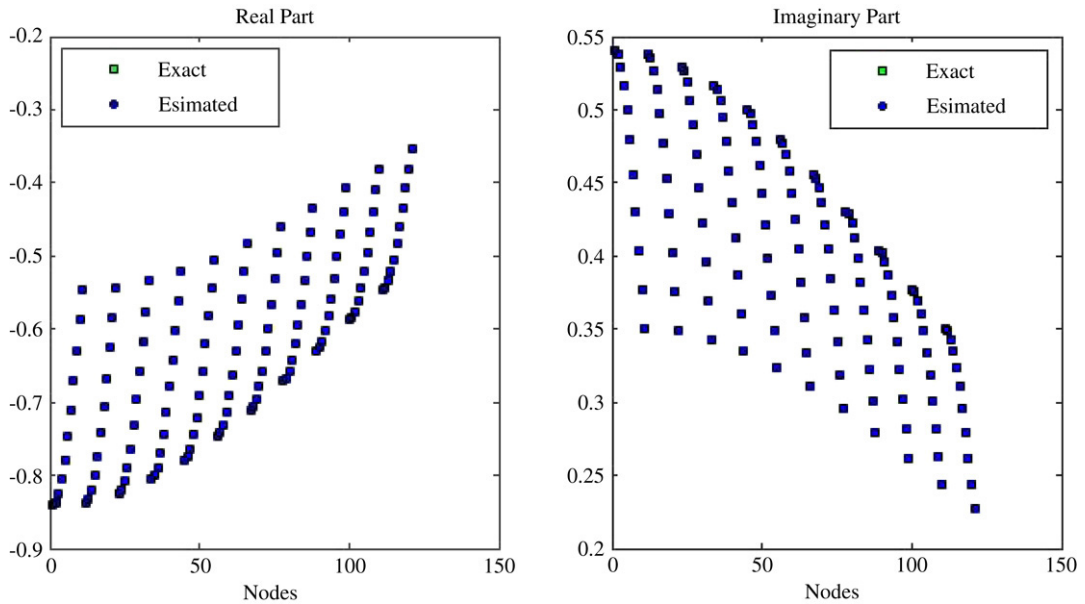

Fig. 1. Real and imaginary parts of numerical and analytical solutions at time $t = 1$ with $dx = dy = 0.1$, $dt = 0.001$ and using MQ (with $c = 0.7$) as the radial basis function for example 1.

results are obtained for $dx = dy = 0.1$, and $dt = 0.001$, and $c = 0.7$ for MQ. Also, for each method we report the maximum error defined as

$$\epsilon = \frac{\mathrm{Max}_{(x,y)\in\Omega}|u_{\mathrm{exact}}(x, y, t) - u_{\mathrm{approximate}}(x, y, t)|}{\mathrm{Max}_{(x,y)\in\Omega}|u_{\mathrm{exact}}(x, y, t)|}. \tag{4.5}$$

The graphs of the real part and the imaginary part of numerical and analytical solutions at time $t = 1$ with $dx = dy = 0.1, dt = 0.001$ and using MQ (with $c = 0.7$) as the radial basis function for all nodes are given in Fig. 1.

In Table 2 we report the maximum absolute error for the real and imaginary parts of the solution and maximum error of solution for various values of the space variables for the TPS scheme with $t = 1$ and $dt = 0.001$. From this table how the differences between the exact and estimated solutions decrease as the mesh sizes reduce can be observed.

Table 2
Maximum absolute error and Max error of thin plate spline based scheme for different spatial distances for example 1 for $t = 1$ with $dt = 0.001$

$dx(= dy)$	Maximum absolute error		Max error ϵ
	Real part	Imaginary part	
0.1	1.0626×10^{-4}	9.3425×10^{-5}	1.2058×10^{-4}
0.05	1.3829×10^{-5}	1.5102×10^{-5}	1.6889×10^{-5}
0.025	1.8563×10^{-6}	2.2814×10^{-6}	2.3130×10^{-6}

Table 3
Maximum absolute error and Max error of multiquadrics and thin plate spline based scheme at different times for example 2 with $dx = dy = 0.1$, $dt = 0.0005$, and $c = 0.45$ for MQ

t	Maximum absolute error		Max error ϵ	CPU time used
	Real part	Imaginary part		
0.1	4.0410×10^{-4}	3.5722×10^{-4}	4.1475×10^{-4}	1
	8.6297×10^{-4}	8.3522×10^{-4}	1.0159×10^{-3}	1
0.3	5.1291×10^{-4}	3.0509×10^{-4}	5.1328×10^{-4}	2
	8.0754×10^{-4}	7.1756×10^{-4}	8.0797×10^{-4}	2
0.5	4.6396×10^{-4}	3.9520×10^{-4}	4.9727×10^{-4}	4
	5.0822×10^{-4}	7.7982×10^{-4}	9.1785×10^{-4}	4
0.7	3.8999×10^{-4}	4.1646×10^{-4}	5.7056×10^{-4}	6
	7.5356×10^{-4}	9.2228×10^{-4}	1.1819×10^{-3}	6
1.0	3.7209×10^{-4}	4.1267×10^{-4}	4.2618×10^{-4}	8
	6.5917×10^{-4}	8.9195×10^{-4}	1.0889×10^{-3}	8

For every value of t, the first and second rows of data correspond to the use of MQ and TPS as the radial basis function in the scheme, respectively.

The results obtained show the very good accuracy and efficiency of the new approximate scheme. Note that we cannot distinguish the exact solution from the estimated solution in Fig. 1.

It is worth pointing that we employed this test problem which is taken from the literature [9]. This enables us to compare our results with the earlier works.

4.2. Example 2

We consider Eq. (3.1) in the region $0 \le x, y \le 1$ with potential function

$$\omega(x, y) = 1 - \frac{2}{x^2} - \frac{2}{y^2}, \tag{4.6}$$

and initial and boundary conditions

$$u(x, y, 0) = x^2 y^2, \quad 0 \le x, y \le 1, \tag{4.7}$$

and

$$u(0, y, t) = 0, \qquad u(1, y, t) = y^2 \exp(\mathrm{i}t),$$

$$u(x, 0, t) = 0, \qquad u(x, 1, t) = x^2 \exp(\mathrm{i}t). \tag{4.8}$$

The analytical solution of the equation is given in [10] as

$$u(x, y, t) = x^2 y^2 \exp(\mathrm{i}t). \tag{4.9}$$

Table 3 presents the maximum absolute error for the real part and imaginary part of solution and maximum error at different times up to $t = 1.0$, using multiquadrics and the thin plate spline as the radial basis function for the method discussed in this paper. These results are obtained for $dx = dy = 0.1$, and $dt = 0.0005$, and $c = 0.45$ for MQ. The graphs of the real part and the imaginary part of the numerical and analytical solutions at time $t = 1$ with

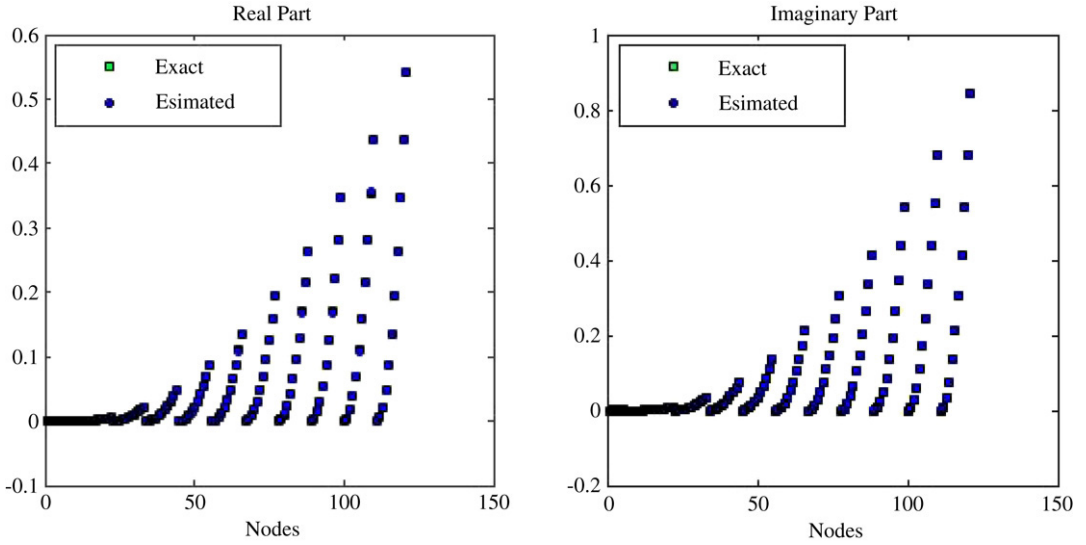

Fig. 2. Real and imaginary parts of numerical and analytical solutions at time $t = 1$ with $dx = dy = 0.1, dt = 0.0005$ and using MQ (with $c = 0.45$) as the radial basis function for example 2.

Table 4
Maximum absolute error and Max error of thin plate spline based scheme for different spatial distances for example 2 for $t = 1$ with $dt = 0.0005$

$dx(= dy)$	Maximum absolute error		Max error ϵ
	Real part	Imaginary part	
0.1	6.5917×10^{-4}	8.9195×10^{-4}	1.0889×10^{-3}
0.05	1.0926×10^{-4}	1.5588×10^{-4}	1.5625×10^{-4}
0.025	1.5771×10^{-5}	1.9652×10^{-5}	2.0828×10^{-5}

$dx = dy = 0.1, dt = 0.0005$ and using MQ (with $c = 0.45$) as the radial basis function for all nodes are given in Fig. 2.

Similar to the previous example, in Table 4 we report the maximum absolute error for the real and imaginary parts of the solution and the maximum error of solution for various values of the space variable for $t = 1$ with $dt = 0.0005$.

This table indicates the effect of reduction of the size of the space variables on increasing the accuracy of the method. Thus it gives an overview of the order of convergence (experimentally) with respect to the space variables.

4.3. Example 3

In this case $\omega(x, y) = 0$ in Eq. (3.1) with initial condition

$$u(x, y, 0) = e^{-(x^2+y^2)-ik_0x}, \tag{4.10}$$

that generates the transient Gaussian distribution

$$u(x, y, 0) = \frac{i}{i - 4t} e^{-i\left((x^2+y^2+ik_0x+ik_0^2t)/(i-4t)\right)}, \tag{4.11}$$

initially centered at (0, 0) and then moving along the negative x-direction as time progresses [8]. Here k_0 is the wave number, which we take as 2.5. Although this is an open domain problem, we have limited our computational domain to the square $-2.5 \le x, y \le 2.5$ and we extract the boundary conditions from the exact solution (4.11) as was done in previous examples.

In Table 5, we present the maximum absolute error for the real part and imaginary part of solution and Max error at different times up to $t = 0.75$, using multiquadrics and the thin plate spline as the radial basis function in the described scheme. These results are obtained for $dx = dy = 0.2$, and $dt = 0.001$, and $c = 0.7$ for MQ. The

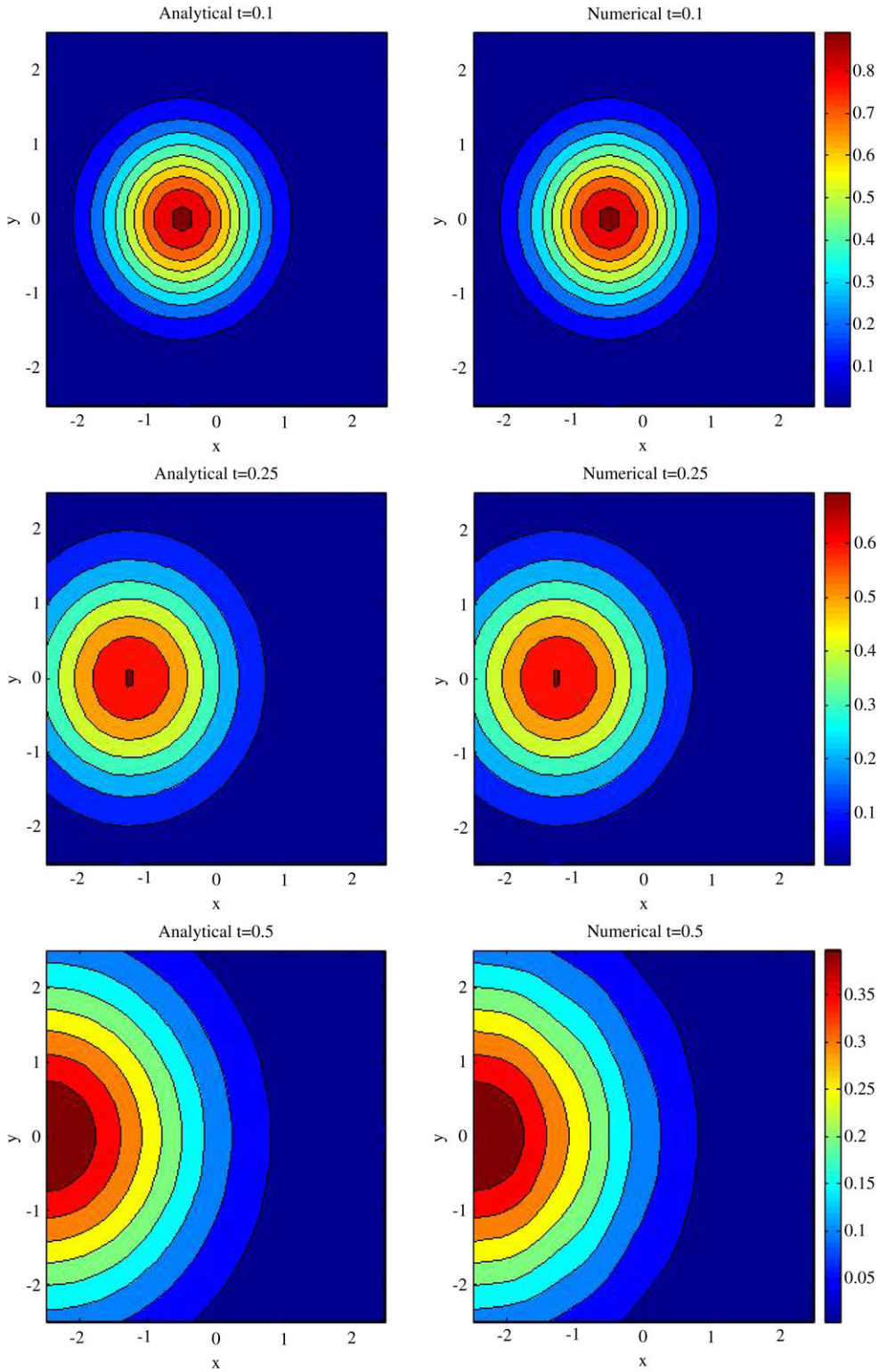

Fig. 3. Modulus of analytical and numerical solutions at three time stations $t = 0.1, 0.25$ and 0.5 with $\mathrm{d}x = \mathrm{d}y = 0.2$, $\mathrm{d}t = 0.001$ and using MQ (with $c = 0.7$) as the radial basis function for example 3.

Table 5
Maximum absolute error and Max error of multiquadrics and thin plate spline based scheme at different times for example 3 with $dx = dy = 0.2$, $dt = 0.001$, and $c = 0.7$ for MQ

t	Maximum absolute error		Max error ϵ	CPU time used
	Real part	Imaginary part		
0.10	9.5813×10^{-5}	1.3705×10^{-4}	1.8167×10^{-4}	16
	2.1128×10^{-3}	1.8044×10^{-3}	2.2974×10^{-3}	19
0.25	3.0058×10^{-3}	2.7889×10^{-3}	4.9110×10^{-3}	33
	9.8778×10^{-3}	7.9629×10^{-3}	1.4186×10^{-2}	35
0.50	3.6903×10^{-3}	3.6072×10^{-3}	8.2866×10^{-3}	61
	7.0321×10^{-3}	6.7913×10^{-3}	1.5981×10^{-2}	64
0.75	3.6905×10^{-3}	4.3421×10^{-3}	1.6078×10^{-2}	90
	6.2051×10^{-3}	9.4070×10^{-3}	3.6423×10^{-2}	92

For every value of t, the first and second rows of data correspond to the use of MQ and TPS as the radial basis function in the scheme, respectively.

contour plot of modulus of analytical and numerical solutions at three time stations $t = 0.1,\ 0.25$ and 0.5 with $dx = dy = 0.2,\ dt = 0.001$ and using MQ (with $c = 0.7$) as the radial basis function are given in Fig. 3.

5. Conclusion

In this paper, we proposed a numerical scheme to solve the two-dimensional (2D) time-dependent Schrödinger equation using collocation points and approximating directly the solution using multiquadrics (MQ) and the thin plate splines (TPS) radial basis function. The scheme works in a similar fashion as finite-difference methods [24]. The numerical results given in the previous section demonstrate the good accuracy of this scheme. It has to be emphasized that the shape parameter c in multiquadrics for all the calculations performed in this paper is found experimentally. When using global radial functions, the resulting matrix coefficient is non-sparse and the system may also be ill-conditioned. To avoid this problem, we can use compact support radial basis functions or preconditioning methods.

References

[1] A. Arnold, Numerically absorbing boundary conditions for quantum evolution equations, VLSI Design 6 (1998) 313–319.
[2] M. L'evy, Parabolic Equation Methods for Electromagnetic Wave Propagation, IEE, 2000.
[3] F.D. Tappert, The parabolic approximation method, in: J.B. Keller, J.S. Papadakis (Eds.), Wave Propagation and Underwater Acoustics, in: Lecture Notes in Physics, vol. 70, Springer, Berlin, 1977, pp. 224–287.
[4] W. Huang, C. Xu, S.T. Chu, S.K. Chaudhuri, The finite-difference vector beam propagation method, J. Lightwave Technol. 10 (3) (1992) 295–304.
[5] F.Y. Hajj, Solution of the Schrodinger equation in two and three dimensions, J. Phys. B At. Mol. Phys. 18 (1985) 1–11.
[6] L.Gr. Ixaru, Operations on oscillatory functions, Comput. Phys. Comm. 105 (1997) 1–9.
[7] I. Gasser, C.K. Lin, P.A. Markowich, A review of dispersive limits of (non) linear Schrödinger type equations, Taiwanese J. Math. 4 (4) (2000) 501–529.
[8] J.C. Kalita, P. Chhabra, S. Kumar, A semi-discrete higher order compact scheme for the unsteady two-dimensional Schrödinger equation, J. Comput. Appl. Math. 197 (2006) 141–149.
[9] M. Dehghan, Finite difference procedures for solving a problem arising in modeling and design of certain optoelectronic devices, Math. Comput. Simulation 71 (2006) 16–30.
[10] M. Subaşi, On the finite-difference schemes for the numerical solution of two dimensional Schrödinger equation, Numer. Methods Partial Differential Equations 18 (2002) 752–758.
[11] M. Dehghan, Fully explicit finite-difference methods for two-dimensional diffusion with an integral condition, Nonlinear Anal., TMA 48 (5(A)) (2002) 637–650.
[12] M. Dehghan, Fully implicit finite differences methods for two-dimensional diffusion with a non-local boundary condition, J. Comput. Appl. Math. 106 (2) (1999) 255–269.
[13] X. Antoine, C. Besse, V. Mouysett, Numerical schemes for the simulation of the two-dimensional Schrödinger equation using non-reflecting boundary conditions, Math. Comput. 73 (2004) 1779–1799.
[14] H. Liu, J. Yan, A local discontinuous Galerkin method for the Korteweg de Vries equation with boundary effect, J. Comput. Phys. 215 (1) (2005) 197–218.
[15] E.J. Kansa, Multiquadrics – A scattered data approximation scheme with applications to computational fluid dynamics – I, Comput. Math. Appl. 19 (1990) 127–145.
[16] E.J. Kansa, Multiquadrics – A scattered data approximation scheme with applications to computational fluid dynamics – II, Comput. Math. Appl. 19 (1990) 147–161.

[17] Y.C. Hon, X.Z. Mao, An efficient numerical scheme for Burgers equation, Appl. Math. Comput. 95 (1) (1998) 37–50.
[18] Y.C. Hon, K.F. Cheung, X.Z. Mao, E.J. Kansa, Multiquadric solution for shallow water equations, ASCE J. Hydraulic Eng. 125 (5) (1999) 524–533.
[19] M. Zerroukat, H. Power, C.S. Chen, A numerical method for heat transfer problem using collocation and radial basis functions, Internat. J. Numer. Meth. Engrg. 42 (1992) 1263–1278.
[20] Y.C. Hon, X.Z. Mao, A radial basis function method for solving options pricing model, Financial Engrg. 8 (1) (1999) 31–49.
[21] M. Marcozzi, S. Choi, C.S. Chen, On the use of boundary conditions for variational formulations arising in financial mathematics, Appl. Math. Comput. 124 (2001) 197–214.
[22] R.E. Carlson, T.A. Foley, The parameter R2 in multiquadric interpolation, Comput. Math. Appl. 21 (1991) 29–42.
[23] A.E. Tarwater, A parameter study of Hardys multiquadric method for scattered data interpolation, Technical report UCRL-54670, Lawrence Livermore National Laboratory, 1985.
[24] M. Dehghan, The one-dimensional heat equation subject to a boundary integral specification, Chaos Solitons Fractals 32 (2007) 661–675.