 CMES, vol.103, no.1, pp.49-70, 2014

# Solution of Two-dimensional Linear and Nonlinear Unsteady Schrödinger Equation using "Quantum Hydrodynamics" Formulation with a MLPG Collocation Method

**V. C. Loukopoulos**[1] **and G. C. Bourantas**[2]

**Abstract:** A numerical solution of the linear and nonlinear time-dependent Schrödinger equation is obtained, using the strong form MLPG Collocation method. Schrödinger equation is replaced by a system of coupled partial differential equations in terms of particle density and velocity potential, by separating the real and imaginary parts of a general solution, called a quantum hydrodynamic (QHD) equation, which is formally analogous to the equations of irrotational motion in a classical fluid. The approximation of the field variables is obtained with the Moving Least Squares (MLS) approximation and the implicit Crank-Nicolson scheme is used for time discretization. For the two-dimensional nonlinear Schrödinger equation, the lagging of coefficients method has been utilized to eliminate the nonlinearity of the corresponding examined problem. A Type-I nodal distribution is used in order to provide convergence for the discrete Laplacian operator used at the governing equation. Numerical results are validated, comparing them with analytical and numerical solutions.

**Keywords:** MLPG Collocation Method, Moving Least Squares, Schrödinger Equation, Quantum Hydrodynamics.

## 1 Introduction

The meshless (or meshfree) methods are being actively developed as a powerful numerical tool for various engineering and physical applications. The primary reason for the significant interest in meshless computational procedures is that most of the established numerical techniques, such as the Finite Element Method (FEM), the Finite Volume Method (FVM), the Finite Difference Method (FDM) and the

[1] Department of Physics, University of Patras, Patras, 26500, Rion, Greece.
[2] aculty of Science, Technology and Communication, University of Luxembourg, Campus Kirchberg, 6, rue Richard Coudenhove-Kalergi L-1359, Luxembourg.

Spectral Method (SP) require a mesh. The automatic generation of a good quality mesh poses a significant problem in the analysis of practical engineering systems. Moreover, the simulation and the analysis of certain types of problems (like dynamic crack propagation, pulsatile and transient flows) often require an expensive remeshing operation. Meshless techniques overcome these difficulties, associated with the meshing and re-meshing procedures, by eliminating the mesh altogether. Interpolation is performed in terms of nodal points scattered at the spatial domain using functions having compact support. A weighted residual technique is used to generate the discrete set of equations corresponding to the governing partial differential equations [Liu (2003), Liu and Gu (2005)].

Since the meshless methods emerged as a potential alternative for solutions in computational mechanics, a variety of such approaches have appeared. Over the last decades, several meshfree methods have been proposed since the prototype of the meshfree methods, the Smoothed Particle Hydrodynamics (SPH), was born [Gingold and Monaghan (1977)]. These methods include the Diffuse Approximation Method (DAM) [Nayroles, Touzot and Villon (1991)], that is closely related to the Moving Least Squares method; the Diffuse Element Method (DEM) [Nayroles, Touzot and Villon (1992)], developed by the Moving Least Squares approximation, and the Element Free Galerkin method (EFG) [Lu, Belytschko and Gu (1994)]; the Reproducing Kernel Particle Method (RKPM) [Liu, Jun and Zhang (1995), Liu, Jun, Li, Adee and Belytschko (1995)], which is used to improve the SPH approximation; the Partition of Unity Finite Element Method (PUFEM) [Melenk and Babuska (1996)]; the *hp*-Clouds [Duarte and Oden (1996)]; the Moving Least-Square Reproducing Kernel Method (MLSRK) [Liu, Li and Belytschko (1996)]; the meshless Local Boundary Integral Equation Method (LBIE) [Zhu, Zhang and Atluri (1998)]; the Meshless Local Petrov–Galerkin method (MLPG) [Atluri, Kim and Cho (1999), Atluri and Shen (2002)]; the Finite Point method (FPM) [Onate, Idelsohn, Zienkiewicz and Taylor (1995)]; the meshless point collocation methods (MPC) [Aluru (2000)], and more.

The present paper is referred to the numerical computation of the two-dimensional (2D) time-dependent Schrödinger equation. Linear Schrödinger equation is written as

$$-i\frac{\partial \psi}{\partial t} = \frac{\partial^2 \psi}{\partial x^2} + \frac{\partial^2 \psi}{\partial y^2} + V(x,y)\,\psi, \quad (x,y) \in \Omega, \quad 0 \le t \le T \tag{1a}$$

in some continuous domain with suitable initial Dirichlet and Neumann boundary conditions and an arbitrary potential function $V(x,y)$. The corresponding initial condition is given by

$$\psi(x,y,0) = h(x,y,0), \quad (x,y) \in \Omega \tag{1b}$$

and the boundary conditions by

$$\psi(x,y,t) = s(x,y,t), \quad (x,y) \in \partial\Omega^D, \quad 0 \le t \le T \tag{1c}$$

$$\frac{\partial \psi}{\partial \boldsymbol{n}}(x,y,t) = g(x,y,t), \quad (x,y) \in \partial\Omega^N, \quad 0 \le t \le T \tag{1d}$$

where $i = \sqrt{-1}$ is the unit imaginary number, $T$ is the final time, $h$, $s$ and $g$ are known functions, and $\partial\Omega = \partial\Omega^D \cup \partial\Omega^N$, where $\partial\Omega^D$ and $\partial\Omega^N$ are the Dirichlet and the Neumann parts of the boundary $\partial\Omega$ and $\boldsymbol{n}$ is the unit outward vector to $\partial\Omega$.

This type of partial equation models many physical problems and find applications in quantum mechanics and various quantum dynamics calculations [Arnold (1998), Hajj (1985), Ixaru (1997)], in electromagnetic wave propagation and the design of certain optoelectronic devices [Levy, (2000), Huang, Xu, Chu and Chaudhuri (1992)], and finally, in underwater acoustics [Tappert (1977)]. The time-dependent Schrödinger equation can be represented in a hydrodynamical form, called a quantum hydrodynamic (QHD) equation, a formulation which is analogous to the equations of irrotational motion in a classical fluid [Gasser, Lin and Markowich (2000), Kalita, Chhabra and Kumar (2006)]. In this formulation, system (1) is replaced by a system of partial differential equations in terms of particle density and velocity potential, by separating the real and imaginary parts of a general solution

$$\begin{aligned} -\frac{\partial u}{\partial t} &= \nabla^2 \upsilon + V\upsilon, \\ \frac{\partial \upsilon}{\partial t} &= \nabla^2 u + Vu, \end{aligned} \tag{2}$$

obtained by expressing $\psi$ as $\psi = u + i\upsilon$, where $u$ and $\upsilon$ are real-values functions.

There have been numerous attempts to develop numerical schemes for equations (1) or the system (2). In [Simos (2008), Simos (2007)] trigonometrically-fitted methods were utilized for the numerical solution of the Schrödinger equation. The authors of [Kalita, Chhabra and Kumar (2006), Subasi (2002)] studied models similar to the present problem using finite-difference techniques. Finite-difference methods are well-known as the first technique for solving partial differential equations (PDEs). In [Dehghan (2002)] explicit finite difference methods were used for solving the governing equations, while in [Dehghan (1999)] the need of using a large amount of CPU time in implicit finite-difference schemes 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. Thus, alternative computational methods, such as global Radial Basis Functions [Dehghan and Shokri (2007)], were used for the numerical solution of the Eq. (1).

In the present paper we investigate a different approach to find the solution of linear and nonlinear Schrödinger equation. We present a numerical scheme to solve the two-dimensional (2D) time-dependent Schrödinger equation using the Collocation method, while we approximate the solution directly using Moving Least Squares. Actually, the meshless point collocation (MPC) method is a case of MLPG when the collocation Dirac's Delta function is used as the test function [Atluri and Shen (2002)]. To test the robustness, the accuracy and the efficiency of the proposed scheme, it is applied to four examples having analytical solutions, with our results exhibiting very good agreement with the analytical ones. Additionally, our results are compared with a meshless collocation and radial basis function method using multiquadrics (MQ) and the Thin Plate Splines (TPS). The layout of the paper is as follows. In Section 2 we present the methodology for the implementation of the Moving Least Squares approximation for the solution of QHD equations. In Section 3 we apply this technique on the two-dimensional (2D) time-dependent Schrödinger equation. The results of the numerical experiments are presented in Section 4, while Section 5 is dedicated to a brief conclusion.

## 2 Moving Least Squares Approximation

### 2.1 Methodology

In the moving least-squares technique, the approximation $u^h(\boldsymbol{x})$ is expressed as the inner product of a vector of the polynomial basis, $\boldsymbol{p}(\boldsymbol{x})$ and a vector of the coefficients, $\boldsymbol{a}(\boldsymbol{x})$

$$u^h(\boldsymbol{x}) = \boldsymbol{p}^T(\boldsymbol{x})\,\boldsymbol{a}(\boldsymbol{x}), \tag{3}$$

where $\boldsymbol{p}(\boldsymbol{x}) \in \boldsymbol{R}^m$, $\boldsymbol{a}(\boldsymbol{x}) \in \boldsymbol{R}^m$ and $m$ is the number of monomials in the polynomial basis (in the present study $m$=6). The local character of the moving least-squares (MLS) approximation can be viewed as a generalization of the traditional least-squares approximation, in which the vector $\boldsymbol{a}$ is not a function of $\boldsymbol{x}$.

Equation (3) is commonly referred to as the global least-squares approximation. In addition, there exists a unique local approximation associated with each point in the domain. In order to determine the form of $\boldsymbol{a}(\boldsymbol{x})$, a weighted discrete error norm,

$$J(\boldsymbol{x}) = \sum_{I=1}^{n} w_I(\boldsymbol{x}) \left[ \sum_{j=1}^{m} p_j^T(\boldsymbol{x}_I)\,\boldsymbol{a}(\boldsymbol{x}) - u_i \right]^2 \tag{4}$$

is constructed and sequentially minimized. Here, $w_I(\boldsymbol{x})$ denotes the weight function, $w_I(\boldsymbol{x}) \equiv w(\boldsymbol{x} - \boldsymbol{x}_I)$, associated with node $I$, and the quantity in brackets is the difference between the local approximation at node $I$ and the data at nodes $I$, that

is $u_i$, and $n$ is the number of nodes in the support of $w_I(\boldsymbol{x})$. The minimization of Eq.(4) with respect to $\boldsymbol{a}(\boldsymbol{x})$ determines $\boldsymbol{a}(\boldsymbol{x})$. The local approximation associated with point $\boldsymbol{x}$ is used only in the minimization process and is equivalent to the global approximation at the single point $\boldsymbol{x}$. Compact support of the weight functions gives the moving least-squares method its local character.

### *2.2 Shape functions and their derivatives*

The minimization of Eq. (4),

$$\frac{\partial J(\boldsymbol{x})}{\partial \boldsymbol{a}(\boldsymbol{x})} = 0 \tag{5}$$

results in the linear system

$$\boldsymbol{A}(\boldsymbol{x})\boldsymbol{a}(\boldsymbol{x}) = \boldsymbol{B}(\boldsymbol{x})\boldsymbol{U}_s, \tag{6}$$

where $\boldsymbol{U}_s$ is a vector containing the nodal data, $\boldsymbol{U}_s^T = [u_1, u_2, ..., u_n]$, and

$$\boldsymbol{A}(\boldsymbol{x}) = \sum_{I=1}^{n} w_I(\boldsymbol{x})\boldsymbol{p}(\boldsymbol{x}_I)\boldsymbol{p}^T(\boldsymbol{x}_I), \tag{7}$$

$$\boldsymbol{B}(\boldsymbol{x}) = \left[ \begin{array}{cccc} w_1(\boldsymbol{x})\boldsymbol{p}(\boldsymbol{x}_1) & w_2(\boldsymbol{x})\boldsymbol{p}(\boldsymbol{x}_2) & \ldots & w_n(\boldsymbol{x})\boldsymbol{p}(\boldsymbol{x}_n) \end{array} \right], \tag{8}$$

where $\boldsymbol{A} \in \boldsymbol{R}^{m\times m}$ and $\boldsymbol{B} \in \boldsymbol{R}^{m\times n}$. The matrix $\boldsymbol{A}$ must be inverted at every sampling point. Substitution of the solution of (Eq.(6)) into the global approximation (Eq.(3)), completes the least-squares approximation,

$$u^h(\boldsymbol{x}) = \underbrace{\boldsymbol{p}^T(\boldsymbol{x})\boldsymbol{A}^{-1}(\boldsymbol{x})\boldsymbol{B}(\boldsymbol{x})}_{\boldsymbol{\varphi}(x)}\boldsymbol{U}_s. \tag{9}$$

Here, the spatial dependence has been lumped into one row matrix, $\boldsymbol{\varphi}(\boldsymbol{x})$ and, therefore, the approximation takes the form of a product of a matrix of shape functions with a vector of nodal data. Derivatives of the shape functions may be calculated by applying the product rule to

$$\boldsymbol{\varphi} = \boldsymbol{p}^T\boldsymbol{A}^{-1}\boldsymbol{B}. \tag{10}$$

In order to obtain the spatial derivatives of the approximation function, $u^h(\boldsymbol{x})$, it is necessary to obtain the derivatives of the MLS shape functions, $\varphi_i(\boldsymbol{x})$,

$$\frac{\partial}{\partial x_j}u^h(\boldsymbol{x}) = \frac{\partial}{\partial x_j}\sum_{i=1}^{n}\varphi_i(\boldsymbol{x})u_i = \sum_{i=1}^{n}\left\{\frac{\partial}{\partial x_j}\varphi_i(\boldsymbol{x})\right\}u_i, \quad x_j = x,y,z. \tag{11}$$

The derivative of the shape function is given as

$$\frac{\partial \phi(\boldsymbol{x})}{\partial x_j} = \frac{\partial\left(\boldsymbol{p}^T \boldsymbol{A}^{-1} \boldsymbol{B}_i\right)}{\partial x_j} = \frac{\partial \boldsymbol{p}^T}{\partial x_j} \boldsymbol{A}^{-1} \boldsymbol{B}_i + \boldsymbol{p}^T \frac{\partial\left(\boldsymbol{A}^{-1}\right)}{\partial x_j} \boldsymbol{B}_i + \boldsymbol{p}^T \boldsymbol{A}^{-1} \frac{\partial \boldsymbol{B}_i}{\partial x_j}, \quad x_j = x, y, z \tag{12}$$

where $\frac{\partial\left(\boldsymbol{A}^{-1}\right)}{\partial x_j} = -\boldsymbol{A}^{-1}(\boldsymbol{x}) \boldsymbol{A}_{,j}(\boldsymbol{x}) \boldsymbol{A}^{-1}(\boldsymbol{x})$. Regarding the second order derivative of the unknown function we get

$$\begin{aligned}
&\frac{\partial^2 \varphi(x)}{\partial x_j^2} = \frac{\partial}{\partial x_j}\left(\frac{\partial \varphi(x)}{\partial x_j}\right) = \frac{\partial}{\partial x_j}\left(\frac{\partial \boldsymbol{p}^T}{\partial x_j} \boldsymbol{A}^{-1} \boldsymbol{B}_i + \boldsymbol{p}^T \frac{\partial\left(\boldsymbol{A}^{-1}\right)}{\partial x_j} \boldsymbol{B}_i + \boldsymbol{p}^T \boldsymbol{A}^{-1} \frac{\partial \boldsymbol{B}_i}{\partial x_j}\right) \\
&= \frac{\partial^2 \boldsymbol{p}^T}{\partial x_j^2} \boldsymbol{A}^{-1} \boldsymbol{B}_i + \frac{\partial \boldsymbol{p}^T}{\partial x_j} \frac{\partial\left(\boldsymbol{A}^{-1}\right)}{\partial x_j} \boldsymbol{B}_i + \frac{\partial \boldsymbol{p}^T}{\partial x_j} \boldsymbol{A}^{-1} \frac{\partial \boldsymbol{B}_i}{\partial x_j} + \\
&+ \frac{\partial \boldsymbol{p}^T}{\partial x_j} \frac{\partial\left(\boldsymbol{A}^{-1}\right)}{\partial x_j} \boldsymbol{B}_i + \boldsymbol{p}^T \frac{\partial^2\left(\boldsymbol{A}^{-1}\right)}{\partial x_j^2} \boldsymbol{B}_i + \boldsymbol{p}^T \frac{\partial\left(\boldsymbol{A}^{-1}\right)}{\partial x_j} \frac{\partial \boldsymbol{B}_i}{\partial x_j} + \\
&+ \frac{\partial \boldsymbol{p}^T}{\partial x_j} \boldsymbol{A}^{-1} \frac{\partial \boldsymbol{B}_i}{\partial x_j} + \boldsymbol{p}^T \frac{\partial\left(\boldsymbol{A}^{-1}\right)}{\partial x_j} \frac{\partial \boldsymbol{B}_i}{\partial x_j} + \boldsymbol{p}^T \boldsymbol{A}^{-1} \frac{\partial^2 \boldsymbol{B}_i}{\partial x_j^2},
\end{aligned} \tag{13}$$

where $x_j = x, y, z$ and $\frac{\partial^2\left(\boldsymbol{A}^{-1}\right)}{\partial x_j^2} = -\frac{\partial\left(\boldsymbol{A}^{-1}\right)}{\partial x_j} \boldsymbol{A} \boldsymbol{A}^{-1} - \boldsymbol{A}^{-1} \frac{\partial \boldsymbol{A}}{\partial x_j} \boldsymbol{A}^{-1} - \boldsymbol{A}^{-1} \boldsymbol{A} \frac{\partial\left(\boldsymbol{A}^{-1}\right)}{\partial x_j}$.

### *2.3 Weight Function*

The weight function is non-zero over a small neighborhood of $\boldsymbol{x}_i$, called the support domain of node $i$. The choice of the weight function $W(\boldsymbol{x} - \boldsymbol{x}_i)$ affects the resulting approximation $u^h(\boldsymbol{x}_i)$ inherently. In the present paper a Gaussian weight function is used [Liu (2003), Bourantas, Skouras and Nikiforidis (2009)], yet the support domain does not have a standard point density value. Instead, a constant number of nodes are used for the approximation of the field function.

$$W(\boldsymbol{x} - \boldsymbol{x}_i) \equiv W(d) = \left\{ \begin{array}{c} e^{-\left(\frac{d_I}{a}\right)^2} \\ 0 \end{array} \right\}, \tag{14}$$

where $I = 1, 2, 3, ..., q$ are the nodes that produce the support domain of node $x_i$, and $d = \frac{|\boldsymbol{x} - \boldsymbol{x}_i|}{a_0^2}$ with $a_0$ a prescribed constant (often $a_0 = 0.2$).

## 3 Collocation formulation

### *3.1 General description*

The Meshless Point Collocation method is a MFree "strong-form" description method. The "strong-form" of the governing equations and the boundary conditions are used and discretized by collocation techniques. The aforementioned formulations possess the following attractive advantages. They are truly meshless and the implementing procedure is straightforward, while the algorithms and the implementation can be kept simple, particularly when handling problems with Dirichlet boundary conditions solely. Under these conditions, these methods are highly efficient computationally, even with the application of polynomial approximation functions, and the solution can be systematically obtained with increased accuracy, compared to FEM, FVM, FDM, or other computational methods. In general, MFree strong-form methods may still suffer from some local stability and accuracy issues, depending on the problem [Liu and Gu (2005)]. However, these local restrictions are now systematically avoided with the utilization of specific nodal distributions (Type-I) and proper local point cloud refinement procedures, in accordance with [Bourantas, Skouras and Nikiforidis (2009), Kim and Liu (2006)], even for natural or mixed type boundary conditions. The robustness of these methods has, however, been an issue especially for scattered set of points. The stability and the convergence of the collocation methods are ensured by the resulting linear or linearized algebraic system. If the latter possesses some attractive features then both the stability and the convergence are ensured. In fact, the robustness of the collocation methods can be improved by understanding the possible sources of errors. Specifically, the errors could arise because of the way the meshless approximation functions and their derivatives have been constructed for a scattered set of points or because of the way the discretization of the governing equations has been performed. When the meshless approximation functions and its derivatives do not satisfy certain conditions (referred to as the positivity conditions) for a given point distribution, it is possible to get large numerical errors when using collocation methods. To satisfy the positivity conditions, the weighting function used in the construction of the approximation functions can play an important role. These studies suggest that positivity conditions can be important when using meshless collocation methods. Additionally, the convergence of the discrete Laplacian operator for Dirichlet boundary conditions has been proved when a regular grid (named Type-I) is used. Thus, both the stability and the convergence of the meshless point collocation method, using MLS approximation and regular nodal distribution are ensured.

Collocation method using MLS may be considered as a special case of the "weak–

form" methods [Atluri, (2004)]. Moreover, this collocation method may be considered as a "weak-solution", with a Dirac delta function as the test (weight) function [Atluri, Liu and Han (2006)]. The weighted residual method provides a flexible mathematical framework for the construction of a variety of numerical solution schemes for the differential equations arising in the field of both science and engineering. Its application, in conjunction with the Moving Least Square (MLS) approximation method, yields powerful solution algorithms for the governing equations.

### *3.2 Time-dependent Meshless Point Collocation method*

The collocation scheme using the Moving Least Squares approximation used in the present work and applied for the spatial discretization of the unsteady homogeneous diffusion equation will be discussed next, along with the explicit Euler, $\theta$-weighted time-stepping scheme used for temporal discretization.

Consider the governing equations of the unsteady problem

$$-\frac{\partial u}{\partial t} = \nabla^2 \upsilon + V\upsilon, \tag{15}$$

$$\frac{\partial \upsilon}{\partial t} = \nabla^2 u + Vu, \tag{16}$$

with the aforementioned boundary and initial conditions. By the MLS approximation one gets $u(\boldsymbol{x}) = \sum_{i=1}^{N} \Phi_i(\boldsymbol{x}) u_i \equiv \boldsymbol{\Phi U}_s$ for the unknown function, $u_q(\boldsymbol{x}) = \sum_{i=1}^{N} \frac{\partial \Phi_i(\boldsymbol{x})}{\partial q} u_i \equiv \boldsymbol{\Phi}_q \boldsymbol{U}_s$ for the partial $x, y$ derivative and $u_{qq}(x) = \sum_{i=1}^{N} \frac{\partial^2 \Phi_i(x)}{\partial q^2} u_i \equiv \boldsymbol{\Phi}_{qq} \boldsymbol{U}_s$ the second $x, y$ partial derivative. Additionally, we set $n_d$ as the number of nodes in the interior, $n_b$ as the number of nodes on the boundary, and the final number of nodes as $N\,(N = n_d + n_b)$. The first equation, Eq. (15) can be written as

$$\frac{\partial u}{\partial t} + \left(\nabla^2 \upsilon + V\upsilon\right) = 0. \tag{17}$$

From the notation described above and using the Euler's $\theta$-weighted time-stepping scheme for temporal discretization, for the interior nodes one gets

$$\begin{aligned} &\Phi_d \tfrac{u^{n+1}-u^n}{\delta t} + \theta\left(\left(\Phi_{d,xx}+\Phi_{d,yy}\right)\upsilon^{n+1} + V\upsilon^{n+1}\right) + \\ &+ (1-\theta)\left(\left(\Phi_{d,xx}+\Phi_{d,yy}\right)\upsilon^{n} + V\upsilon^{n}\right) = 0. \end{aligned} \tag{18}$$

Multiplying both parts by $\delta t$ one gets

$$\begin{aligned} &\Phi_d u^{n+1} - \Phi_d u^n + \theta\,\delta t\left(\left(\Phi_{d,xx}+\Phi_{d,yy}\right)\upsilon^{n+1}\right) + \theta\,\delta t\left(V\upsilon^{n+1}\right) + \\ &+ (1-\theta)\,\delta t\left(\left(\Phi_{d,xx}+\Phi_{d,yy}\right)\upsilon^{n}\right) + (1-\theta)\,\delta t\left(V\upsilon^{n}\right) = 0 \end{aligned} \tag{19}$$

In matrix notation, for all points, incorporating the boundary conditions at $n_b$ boundary nodes one has

$$\begin{bmatrix} \Phi_d & \theta\delta t\left(\Phi_{d,xx}+\Phi_{d,yy}+V\right) \\ G_V\Phi_b & 0 \end{bmatrix}\begin{bmatrix} u^{n+1} \\ v^{n+1} \end{bmatrix}$$

$$=\begin{bmatrix} \Phi_d & -(1-\theta)\,\delta t\left(\Phi_{d,xx}+\Phi_{d,yy}+V\right) \\ 0 & 0 \end{bmatrix}\begin{bmatrix} u^{n} \\ v^{n} \end{bmatrix}+\begin{bmatrix} 0 \\ g_1^{n+1} \end{bmatrix}, \tag{20}$$

where $G_V$ is the operator defining the boundary conditions for velocity (Dirichlet type on $\partial\Omega$).

These equations can be written in a more compact manner by setting

$$\boldsymbol{H}_A^+=\begin{bmatrix} \Phi_d & \theta\delta t\left(\Phi_{d,xx}+\Phi_{d,yy}+V\right) \\ G_V\Phi_b & 0 \end{bmatrix},$$

$$\boldsymbol{H}_A^-=\begin{bmatrix} \Phi_d & -(1-\theta)\,\delta t\left(\Phi_{d,xx}+\Phi_{d,yy}+V\right) \\ 0 & 0 \end{bmatrix}$$

and $\boldsymbol{F}_A=\begin{bmatrix} 0 \\ g_1^{n+1} \end{bmatrix}$,

where $\boldsymbol{H}_A^+\in\boldsymbol{R}^{N\times 2N}$, $\boldsymbol{H}_A^-\in\boldsymbol{R}^{N\times 2N}$, and $\boldsymbol{F}_A\in\boldsymbol{R}^{N\times 1}$.

Regarding the second Eq. (16) and following the same procedure described for Eq. (15) one can derive (in matrix notation)

$$\begin{bmatrix} \theta\delta t\left(-\Phi_{d,xx}-\Phi_{d,yy}-V\right) & \Phi_d \\ 0 & G_B\Phi_b \end{bmatrix}\begin{bmatrix} u^{n+1} \\ v^{n+1} \end{bmatrix}$$

$$=\begin{bmatrix} -(1-\theta)\,\delta t\left(-\Phi_{d,xx}-\Phi_{d,yy}-V\right) & \Phi_d \\ 0 & 0 \end{bmatrix}\begin{bmatrix} u^{n} \\ v^{n} \end{bmatrix}+\begin{bmatrix} 0 \\ g_2^{n+1} \end{bmatrix}, \tag{21}$$

where $G_B$ is the operator defining the boundary conditions for the induced magnetic field on $\partial\Omega$. Once again, the above equations can be written in more compact form by setting

$$\boldsymbol{H}_B^+=\begin{bmatrix} \theta\delta t\left(-\Phi_{d,xx}-\Phi_{d,yy}-V\right) & \Phi_d \\ 0 & G_B\Phi_b \end{bmatrix},$$

$$\boldsymbol{H}_B^-=\begin{bmatrix} -(1-\theta)\,\delta t\left(-\Phi_{d,xx}-\Phi_{d,yy}-V\right) & \Phi_d \\ 0 & 0 \end{bmatrix},$$

and $\boldsymbol{F}_B=\begin{bmatrix} 0 \\ g_2^{n+1} \end{bmatrix}$,

where $\boldsymbol{H}_B^+ \in \boldsymbol{R}^{N\times 2N}$, $\boldsymbol{H}_B^- \in \boldsymbol{R}^{N\times 2N}$ and $\boldsymbol{F}_B \in \boldsymbol{R}^{N\times 1}$.

The final system of the QHD coupled partial differential equations can be written as

$$\begin{bmatrix} \boldsymbol{H}_A^+ \\ \boldsymbol{H}_B^+ \end{bmatrix} \begin{bmatrix} u^{n+1} \\ \upsilon^{n+1} \end{bmatrix} = \begin{bmatrix} \boldsymbol{H}_A^- \\ \boldsymbol{H}_B^- \end{bmatrix} \begin{bmatrix} u^{n} \\ \upsilon^{n} \end{bmatrix} + \begin{bmatrix} \boldsymbol{F}_A \\ \boldsymbol{F}_B \end{bmatrix}. \tag{22}$$

Finally, setting

$$\boldsymbol{u}^n = \begin{bmatrix} u^n \\ \upsilon^n \end{bmatrix}, \quad \boldsymbol{F} = \begin{bmatrix} \boldsymbol{F}_A \\ \boldsymbol{F}_B \end{bmatrix}, \quad \boldsymbol{Q}^+ = \begin{bmatrix} \boldsymbol{H}_A^+ \\ \boldsymbol{H}_B^+ \end{bmatrix}, \quad \boldsymbol{Q}^- = \begin{bmatrix} \boldsymbol{H}_A^- \\ \boldsymbol{H}_B^- \end{bmatrix},$$

the discretized PDEs of QHD flow are summed as

$$\boldsymbol{u}^{n+1} = \left(\boldsymbol{Q}^+\right)^{-1}\left(\boldsymbol{Q}^-\boldsymbol{u}^n + \boldsymbol{F}\right), \tag{23}$$

where $\boldsymbol{Q}^+ \in \boldsymbol{R}^{2N\times 2N}$, $\boldsymbol{Q}^- \in \boldsymbol{R}^{2N\times 2N}$ and $\boldsymbol{F} \in \boldsymbol{R}^{2N\times 1}$.

## 4 Numerical experiments

In order to examine the validity and the effectiveness of the proposed scheme, four representative case studies were examined [Dehghan and Shokri (2007), Dehghan, and Mirzaei (2008), Dehghan, and Mirzaei (2008)]; thee cases for the linear Schrödinger equation with and without the potential function present, and a fourth one for nonlinear Schrödinger equation.

### *Example 1*

Initially, we consider the case with potential $V = 0$ at the Schrödinger equation, in the spatial domain $(0,1)\times(0,1)$ and initial conditions [Dehghan, and Mirzaei (2008)]

$$\psi(x,y,0) = e^{i(x+y)},$$

which generates the exact solution

$$\psi(x,y,t) = e^{i(x+y-2t)}.$$

The Dirichlet boundary conditions were extracted from the analytical solution. Table 1 presents the relative error of both real and imaginary parts, defined as $\varepsilon = \frac{\|u_{num}-u_{exact}\|_2}{\|u_{exact}\|_2}$, for $t = 5$ and $t = 20$ sec. The meshless point method with MLS approximation depends on several parameters that have to be chosen properly in order to achieve convergence and accuracy. These parameters include the proper

nodal distribution, the number of nodes in the support domain, and the user-defined variables used in the weight function. For our investigation purposes we use a regular nodal distribution of Type-I [Kim and Liu (2006)], which ensures the fulfillment of the so-called positivity conditions [Jin, Li and Aluru (2004)]. Additionally, we set the user-defined parameter $\alpha_0$ at the weight function to be $\alpha_0 = 0.2$, the number of nodes in the support domain 10, and time step $dt = 0.05$. As pointed out elsewhere [Bourantas, Skouras and Nikiforidis (2009)], when the number of nodes in the total domain is increased, the accuracy is improved. This also is depicted at the Table 1.

The MLS approximation is obtained by a special least-squares method [Liu and Gu (2005)]. The function obtained by the MLS approximation is a smooth curve (or surface), which does not pass through the nodal values inherently. Therefore, the MLS shape functions do not, in general, satisfy the Kronecker delta condition. Thus, when the nodes in the support domain increase, the Gaussian weight function loses its local character (delta function property), resulting in truncated errors which decrease the accuracy of the numerical results. Thus, in Table 2, we present the dependence of the accuracy from the number of nodes in the support domain. To do that, we used a constant grid of $31 \times 31$ nodes and altered the number of nodes at the support domain. The results obtained show the very good accuracy of the proposed scheme when the number of the nodes in the support domain is kept small. Moreover, in Fig. 1, plots are presented for numerical and exact solutions for the real and imaginary part at $t = 20$, using a $21 \times 21$ regular grid and 10 nodes in the support domain.

Table 1: Relative errors at $t = 5$ and $t = 20$ for different grids, $dt = 0.05$ for support domain 10.

| | $t = 5$ | | $t = 20$ | |
|---|---|---|---|---|
| **Grid** | **Real** | **Imaginary** | **Real** | **Imaginary** |
| 11x11 | 7.6981E-05 | 1.2284E-04 | 1.6446E-05 | 2.9136E-04 |
| 16x16 | 2.2556E-05 | 6.7048E-05 | 1.5732E-04 | 9.3423E-05 |
| 21x21 | 8.7172E-06 | 4.8229E-05 | 1.1227E-04 | 3.6805E-05 |
| 26x26 | 5.8460E-06 | 3.9876E-05 | 7.9893E-05 | 1.7957E-05 |
| 31x31 | 5.4051E-06 | 3.5865E-05 | 5.9303E-05 | 1.4117E-05 |

***Example 2***

As a second example, we consider the Schrödinger equation in the spatial domain $(0,1) \times (0,1)$, with potential function [Dehghan and Shokri (2007), Dehghan, and

Table 2: Relative errors at $t = 5$ and $t = 20$ for different number of nodes in the support domain, $dt = 0.05$.

| | $t = 5$ | | $t = 20$ | |
|---|---|---|---|---|
| **Sup. Domain** | **Real** | **Imaginary** | **Real** | **Imaginary** |
| 10 | 5.4051E-06 | 3.5865E-05 | 5.9303E-05 | 1.4117E-05 |
| 15 | 7.1039E-05 | 7.1072E-05 | 5.9706E-05 | 1.4208E-05 |
| 20 | 6.3606E-05 | 1.8795E-05 | 2.5153E-04 | 7.6710E-05 |
| 25 | 6.2948E-04 | 1.9208E-03 | 2.2634E-03 | 8.1771E-04 |
| 30 | 5.3568E-03 | 7.4155E-03 | 6.9545E-03 | 2.8022E-03 |
| 35 | 1.5133E-02 | 3.0736E-02 | 3.0478E-02 | 1.2842E-02 |

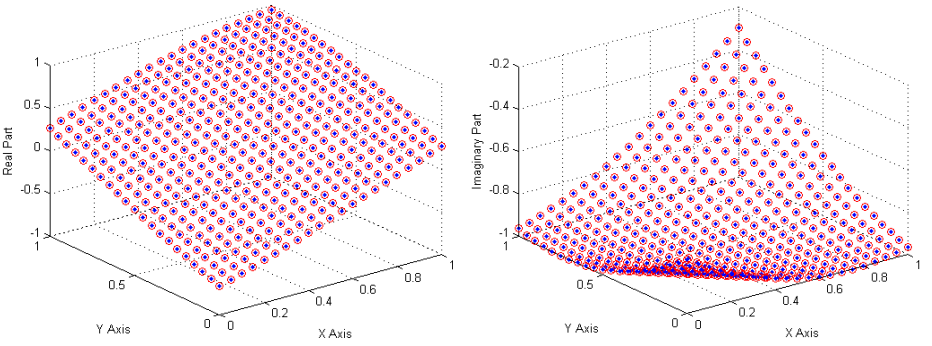


Figure 1: Plots of numerical and exact solutions for the real and imaginary part at $t = 20$, using a $21 \times 21$ regular grid and 10 nodes in the support domain.

Mirzaei (2008)]

$V(x,y) = 3 - 2\tanh^2 x - 2\tanh^2 y.$

Initial and boundary conditions are defined as

$$\psi(x,y,0) = \frac{i}{\cosh(x)\cosh(y)}, \quad 0 \le x,y \le 1$$

and

$$\psi(0,y,t) = \frac{ie^{it}}{\cosh(y)}, \quad \psi(1,y,t) = \frac{ie^{it}}{\cosh(1)\cosh(y)},$$

$$\psi(x,0,t) = \frac{ie^{it}}{\cosh(x)}, \quad \psi(x,1,t) = \frac{ie^{it}}{\cosh(x)\cosh(1)}.$$

The analytical solution is given by

$$\psi(x,y,t) = \frac{ie^{it}}{\cosh x \cosh y}.$$

Table 3 presents the maximum absolute error for the real and the imaginary parts of the solution at different times up to $t = 1$, using meshless point collocation method with MLS approximation. For comparison purposes, numerical results are also presented using meshless collocation method with global Radial Basis Functions approximation using multiquadrics (MQ) and thin plate splines (TPS) respectively [Dehghan and Shokri (2007)]. These results were obtained for $dx = dy = 0.1$, and $dt = 0.001$. The maximum relative error, $\varepsilon$, defined as $\varepsilon = Max_{(x,y)\in\Omega}\left(\frac{|u_{exact}(x,y,t) - u_{approximate}(x,y,t)|}{|u_{exact}(x,y,t)|}\right)$, was also reported. The total number of nodes was 121 $(11 \times 11)$, the number of nodes in the support domain was set to 10, ensuring the inversion of the moment matrix, $\boldsymbol{A}(\boldsymbol{x})$, and the parameter $\alpha_0$ was set to $\alpha_0 = 0.2$ [Liu (2003)].

At Table 4 the CPU time (in seconds) is presented, in order to demonstrate the efficiency of the meshless point collocation method. The shape functions are not pre-defined, and they must be constructed before the numerical solution of the resulting algebraic system. Thus, in our in-house code, the numerical procedure contains two parts; first comes the construction of the shape functions and, then, the solution of the resulting linear system.

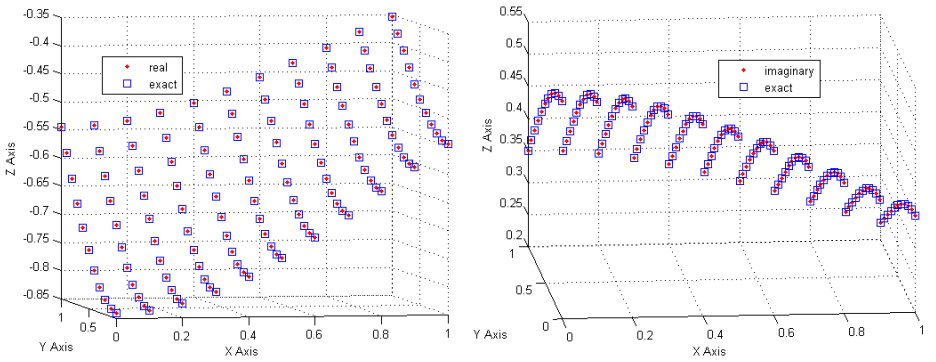


Figure 2: Plots of numerical and exact solutions for the real and imaginary part at $t = 1$, using a $11 \times 11$ grid.

In Fig. 2 the graphs of the real part and the imaginary parts of the numerical and the analytical solutions using MLS are shown at time $t = 1$, with $dx = dy = 0.1$,

Table 3: Maximum absolute error of multiquadrics and thin plate spline based scheme at different times with $dx = dy = 0.1$, $dt = 0.001$ and $c = 0.7$ for MQ. For every value of $t$, the first and second rows of data correspond to the use of MQ, TPS as the radial basis function respectively and the third for the MPC.

| | | **Maximum absolute error** | |
|---|---|---|---|
| **t** | | **Real** | **Imaginary** |
| 0.1 | MQ | 2.4407E-05 | 2.9974E-05 |
| | TPS | 7.8895E-05 | 9.8635E-05 |
| | MLS | 1.4644E-04 | 1.5220E-04 |
| 0.3 | MQ | 2.9466E-05 | 2.3861E-05 |
| | TPS | 1.0368E-05 | 8.6876E-05 |
| | MLS | 1.3317E-04 | 9.8297E-05 |
| 0.5 | MQ | 2.7468E-05 | 3.4044E-05 |
| | TPS | 7.7545E-05 | 9.1676E-05 |
| | MLS | 8.3716E-05 | 1.9683E-04 |
| 0.7 | MQ | 2.5495E-05 | 1.8694E-05 |
| | TPS | 8.9137E-05 | 7.7454E-05 |
| | MLS | 1.5182E-05 | 1.7088E-05 |
| 1.0 | MQ | 2.9444E-05 | 2.4222E-05 |
| | TPS | 1.0626E-04 | 9.3425E-05 |
| | MLS | 1.5138E-04 | 9.5315E-05 |

Table 4: CPU time in seconds for shape construction and solution of the resulting transient, linear system.

| **nodes** | **Shape Functions (sec)** | **Linear system (sec)** |
|---|---|---|
| 121 | 0.53125 | 9.54687 |
| 441 | 1.90625 | 35.98437 |
| 961 | 3.78125 | 108.01562 |
| 1681 | 6.9375 | 282.06250 |

$dt = 0.001$. Note that there is no essential divergence between the exact solution and the numerical solution in Fig. 2, for the given accuracy.

One can notice that, for coarse grids, as in the case of 121 nodes, the numerical results obtained by the meshless point collocation with MLS approximation are less accurate than those obtained by the global multiquadrics Radial Basis Function. Although full-domain RBF methods are highly flexible and can exhibit high-order convergence rates [Madych and Nelson (1990)], in their basic implementation the fully-populated matrix systems produced lead to poor numerical conditioning as the size of the dataset increases. This problem is described by Schaback [Schaback (1993)] as the "uncertainty relation", in which better conditioning is associated with worse accuracy, and worse conditioning is associated with improved accuracy. With increasingly large datasets and increasingly flat basis functions, this problem becomes more pronounced. Thus, global RBF are not appropriate for real world applications, were the number of the degrees of freedom (nodes) are large. On the other hand, MLS approximation, being a localized-type approximation, uses a small number of neighboring nodes for interpolation. This makes the MLS approximation more suitable for many applications arising in science and engineering. Furthermore, the small number of nodes used makes the method computationally time and memory saving. This is evident at Table 5 where doubling the nodal distribution density increases the accuracy of the numerical solution by an order of magnitude, while the computational efficiency of the scheme is retained.

Table 5: Absolute and relative errors at different times for $dx = dy = 0.05$ and 0.025, and $dt = 0.001$.

| | | **Maximum absolute error** | | **Maximum relative error** | |
|---|---|---|---|---|---|
| **t** | | **Real** | **Imaginary** | **Real** | **Imaginary** |
| 0.1 | $dx = 0.05$ | 3.6969E-05 | 3.6482E-05 | 1.6920E-04 | 1.9557E-05 |
| | $dx = 0.025$ | 8.6811E-06 | 9.0573E-06 | 4.3579E-05 | 4.9694E-06 |
| 0.3 | $dx = 0.05$ | 3.9436E-05 | 2.9388E-05 | 6.2200E-05 | 0.4472E-05 |
| | $dx = 0.025$ | 9.4746E-06 | 7.6511E-06 | 1.6000E-05 | 3.8843E-06 |
| 0.5 | $dx = 0.05$ | 2.3815E-05 | 4.1528E-05 | 2.8327E-05 | 2.6131E-05 |
| | $dx = 0.025$ | 7.6615E-06 | 1.0489E-05 | 7.3170E-06 | 6.4270E-06 |
| 0.7 | $dx = 0.05$ | 4.2027E-05 | 1.8415E-05 | 3.4707E-05 | 1.2463E-05 |
| | $dx = 0.025$ | 9.9946E-06 | 4.7105E-06 | 9.0405E-06 | 3.8419E-06 |
| 1.0 | $dx = 0.05$ | 2.5305E-05 | 3.4800E-05 | 1.7559E-05 | 3.8282E-05 |
| | $dx = 0.025$ | 6.1169E-06 | 9.7405E-06 | 3.6556E-06 | 9.9219E-06 |

***Example 3***

Following, we consider the Schrödinger equation in $(0,1)\times(0,1)$ spatial domain and with potential function [Dehghan and Shokri (2007), Dehghan, and Mirzaei (2008)]

$$V(x,y) = 1 - \frac{2}{x^2} - \frac{2}{y^2}$$

and initial and boundary conditions

$$\psi(x,y,0) = x^2 y^2$$

and

$$\psi(0,y,t) = 0, \quad \psi(1,y,t) = y^2 e^{it}, \quad \psi(x,0,t) = 0, \quad \psi(x,1,t) = x^2 e^{it},$$

The analytical solution is given as

$$\psi(x,y,t) = x^2 y^2 e^{it}.$$

Table 6 presents the maximum absolute error for the real part and imaginary part at different times up to $t = 1$, using MLS approximation and time step $dt = 0.05$. The results obtained were compared with those obtained using the multiquadrics and the thin plate spline RBF with the same nodal distribution and time step, $dt = 0.0005$ [Dehghan and Shokri (2007)]. One can observe that, for MPC with MLS approximation of localized type, using a time step two orders lower than the time step used in global RBF, the absolute errors present two orders higher accuracy.

Finally, in Fig. 3, the graphs of the real part and the imaginary parts of the numerical and the analytical solutions using MLS are shown at time $t = 1$, with $dx = dy = 0.1$, $dt = 0.05$. Note that there is no essential divergence between the exact solution and the numerical solution in Fig. 2, for the given accuracy.

***Example 4***

Finally, we consider the generalized nonlinear two-dimensional Schrödinger equation written as [Dehghan, and Mirzaei (2008)]:

$$-i\frac{\partial \psi}{\partial t} + \frac{\partial^2 \psi}{\partial x^2} + \frac{\partial^2 \psi}{\partial y^2} = B(x,y,t)\,\psi + C(x,y,t)\,|\psi|^p\,\psi,$$

with the initial and boundary conditions

$$\psi(x,y,0) = \cos(x)\cos(y), \quad (x,y) \in \Omega$$

Table 6: Maximum absolute error of multiquadrics and thin plate spline-based scheme at different times with $dx = dy = 0.1$, $dt = 0.0005$ and $c = 0.45$ for MQ. 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, respectively and the third for the MPC when $dx = dy = 0.1$, $dt = 0.05$.

| | | **Maximum absolute error** | |
|---|---|---|---|
| **t** | | **Real** | **Imaginary** |
| 0.1 | MQ | 4.0410E-04 | 3.5722E-04 |
| | TPS | 8.6297E-04 | 8.3522E-04 |
| | MLS | 2.7156E-06 | 1.1912E-06 |
| 0.3 | MQ | 5.1291E-04 | 3.0509E-04 |
| | TPS | 8.0754E-04 | 7.1756e-04 |
| | MLS | 3.1253E-06 | 1.5355E-06 |
| 0.5 | MQ | 4.6396E-04 | 3.9520E-04 |
| | TPS | 5.0822E-04 | 7.7982E-04 |
| | MLS | 1.7575E-06 | 2.2252E-06 |
| 0.7 | MQ | 3.8999E-04 | 4.1646E-04 |
| | TPS | 7.5356E-04 | 9.2228E-04 |
| | MLS | 2.2781E-06 | 3.7907E-06 |
| 1.0 | MQ | 3.7209E-04 | 4.1267E-04 |
| | TPS | 6.5917E-04 | 8.9195E-04 |
| | MLS | 1.4423E-06 | 1.0944E-06 |

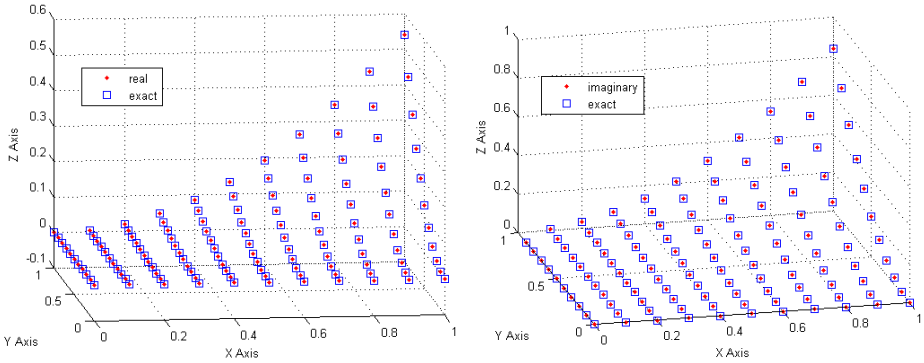


Figure 3: Plots of the exact and the numerical solution at $t = 1.0$.

and Neumann boundary conditions on all sides of the spatial domain

$$\frac{\partial \psi}{\partial \boldsymbol{n}} = 0.$$

The analytical solution is given as

$$\psi(x,y,t) = e^{-it}\cos(\pi x)\cos(\pi y).$$

The lagging of coefficients method has been utilized to eliminate the non-linearity of the examined problem. The spatial domain of the problem is defined as 0≤x,y≤1. The function used in the present problem are defined as $C(x,y) = 1-2\pi^2$, $B(x,y) = \left(1-2\pi^2\right)\left(1-\cos^2(\pi x)\cos^2(\pi y)\right)$ and $p=2$. We have to notice that the accuracy of the case under consideration agrees with the exact solution at about two significant digits and, as the time increases it becomes worse. This is due to the imposition of the Neumann boundary conditions. When using Dirichlet boundary conditions the accuracy of the numerical results increases. Following the aforementioned procedure the final linearized system in matrix notation can be written as

$$\boldsymbol{H}_A^+ = \begin{bmatrix} \Phi_d & \theta\delta t\left(\Phi_{d,xx}+\Phi_{d,yy}-B\Phi_d-C\left(|\Psi|^p\right)^n\Phi_d\right) \\ G_V\Phi_b & 0 \end{bmatrix},$$

$$\boldsymbol{H}_A^- = \begin{bmatrix} \Phi_d & -(1-\theta)\delta t\left(\Phi_{d,xx}+\Phi_{d,yy}-B\Phi_d-C\left(|\Psi|^p\right)^n\Phi_d\right) \\ 0 & 0 \end{bmatrix}$$

and $\boldsymbol{F}_A = \begin{bmatrix} 0 \\ g_1^{n+1} \end{bmatrix}$,

$$\boldsymbol{H}_B^+ = \begin{bmatrix} \theta\delta t\left(\Phi_{d,xx}+\Phi_{d,yy}-B\Phi_d-C\left(|\Psi|^p\right)^n\Phi_d\right) & \Phi_d \\ 0 & G_B\Phi_b \end{bmatrix},$$

$$\boldsymbol{H}_B^- = \begin{bmatrix} -(1-\theta)\delta t\left(\Phi_{d,xx}+\Phi_{d,yy}-B\Phi_d-C\left(|\Psi|^p\right)^n\Phi_d\right) & \Phi_d \\ 0 & 0 \end{bmatrix},$$

and $\boldsymbol{F}_B = \begin{bmatrix} 0 \\ g_2^{n+1} \end{bmatrix}$.

## 5 Conclusions

In the present work we used the meshless numerical scheme to solve the two-dimensional time-dependent linear and nonlinear Schrödinger equation using the point collocation method with MLS approximation. For the Schrödinger equation we developed a fully coupled, transient, and strong-form solver for the real and

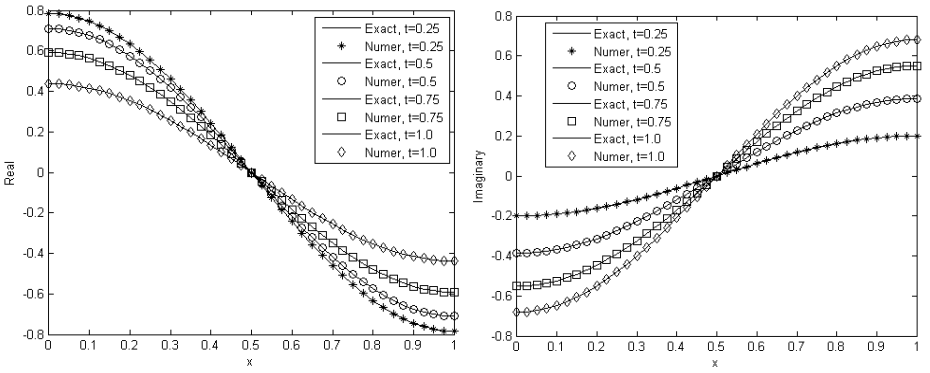


Figure 4: Analytical and numerical solutions at various time.

the imaginary parts of the general solution of the so-called quantum hydrodynamic (QHD) equation. The proposed scheme is applied to four benchmark cases having analytical solutions, with our results exhibiting excellent agreement with all the analytical ones. The numerical results were also compared with those provided by another collocation method, that is, the global Radial Basis Function method. The numerical results provided by the proposed scheme are highly accurate, compared with the ones provided by the multiquadrics and the thin plates splines RBF. Furthermore, in some cases they are also less CPU time and memory consuming. This makes the application of the MLS approximation very attractive for the numerical solution of this kind of physical problems.

## References

**Aluru, N. R.** (2000): A point collocation method based on reproducing kernel approximations. *International Journal for Numerical Methods in Engineering*, vol. 47, pp. 1083-1121.

**Arnold, A.** (1998): Numerically absorbing boundary conditions for quantum evolution equations. *VLSI Design*, vol. 6, pp. 313-319.

**Atluri, S. N.; Kim, H. G.; Cho, J. Y.** (1999): A critical assessment of the truly Meshless local Petrov–Galerkin (MLPG) methods. *Computational Mechanics*, vol. 24, pp. 348-372.

**Atluri, S. N.; Liu, H. T.; Han, Z. D.** (2006): Meshless Local Petrov-Galerking (MLPG) Mixed Finite Differences Method for Solid Mechanics. *CMES: Computer Modeling in Engineering and Sciences*, vol. 15, pp. 1-16.

**Atluri, S. N.; Shen, S. P.** (2002): *The Meshless Local Petrov–Galerkin (MLPG)*

*Method*, Tech Science Press, Encino USA.

**Atluri, S. N.** (2004): *The Mesheless Method (MLPG) for Domain & BIE Discretizations*, Tech Science Press.

**Bourantas, G. C.; Skouras, E. D.; Nikiforidis, G. C.** (2009): Adaptive support domain implementation on the Moving Least Squares approximation for Mfree methods applied on elliptic and parabolic PDE problems using strong-form description. *CMES: Computer Modeling in Engineering and Sciences,* vol. 43, pp. 1-25.

**Dehghan, M.** (1999): Fully implicit finite differences methods for two-dimensional diffusion with a non-local boundary condition. *Journal of Computational and Applied Mathematics*, vol. 106, pp. 255-269.

**Dehghan, M.** (2002): Fully explicit finite-difference methods for two-dimensional diffusion with an integral condition. *Nonlinear Analysis: Theory, Methods & Applications*, vol. 48, pp. 637-650.

**Dehghan, M.; Mirzaei, D.** (2008): Numerical solution to the unsteady two-dimensional Schrödinger equation using meshless local boundary integral equation method. *International Journal for Numerical Methods in Engineering*, vol. 76, pp. 501-520.

**Dehghan, M.; Mirzaei, D.** (2008): The meshless local Petrov–Galerkin (MLPG) method for the generalized two-dimensional non-linear Schrodinger equation. *Engineering Analysis with Boundary Elements*, vol. 32, pp. 747–756.

**Dehghan, M.; Shokri, A.** (2007): A numerical method for two-dimensional Schrödinger equation using collocation and radial basis functions. *Computers & Mathematics with Applications*, vol. 54, pp. 136-146.

**Duarte, C. A.; Oden, J. T.** (1996): An h-p adaptive method using clouds. *Computer Methods in Applied Mechanics and Engineering*, vol. 139, pp. 237-262.

**Gasser, I.; Lin, C. K.; Markowich, P. A.** (2000): A review of dispersive limits of (non) linear Schrödinger type equations. *Taiwanese Journal of Mathematics*, vol. 4, pp. 501-529.

**Gingold, R. A.; Monaghan, J. J.** (1977): Smoothed particle hydrodynamics: theory and application to non-spherical stars. *Monthly Notices of the Royal Astronomical Society*, vol. 181, pp. 375-389.

**Hajj, F. Y.** (1985): Solution of the Schrodinger equation in two and three dimensions. *Journal of Physics B: Atomic, Molecular and Optical Physics*, vol. 18, pp. 1-11.

**Huang, W.; Xu, C.; Chu, S. T.; Chaudhuri, S. K.** (1992): The finite-difference vector beam propagation method. *Journal of Lightwave Technology*, vol. 10, pp. 295-304.

**Ixaru, L. Gr.** (1997): Operations on oscillatory functions. *Computer Physics Communications*, vol. 105, pp. 1-9.

**Jin, X.; Li, G.; Aluru, N. R.** (2004): Positivity conditions in meshless collocation methods. *Computer Methods in Applied Mechanics and Engineering*, vol. 193, pp. 1171-1202.

**Kalita, J. C.; Chhabra, P.; Kumar, S.** (2006): A semi-discrete higher order compact scheme for the unsteady two dimensional Schrödinger equation. *Journal of Computational and Applied Mathematic,* vol. 197, pp. 141-149.

**Kim, D. W.; Liu, W.K.** (2006): Maximum principle and convergence analysis for the meshfree point collocation method. *SIAM Journal on Numerical Analysis*, vol. 44, pp. 515-539.

**Levy, M.** (2000): *Parabolic Equation Methods for Electromagnetic Wave Propagation*, IEE.

**Liu, G. R.** (2003): *Meshfree method: Moving beyond the Finite Element Method*. CRC Press.

**Liu, G. R.; Gu, Y. T.** (2005): *An Introduction to Meshfree Methods and Their Programming*. Springer: Dordrecht.

**Liu, W. K.; Jun, S.; Zhang, Y. F.** (1995): Reproducing kernel particle methods. *International Journal for Numerical Methods in Fluids*, vol. 20, pp. 1081-1106.

**Liu, W. K.; Li, S.; Belytschko, T.** (1996): Moving least square reproducing kernel methods (I) methodology and convergence. *Computer Methods in Applied Mechanics and Engineering*, vol. 143, pp. 422-433.

**Liu, W.K.; Jun, S.; Li, S.; Adee, J.; Belytschko, T.** (1995): Reproducing kernel particle methods for structural dynamics. *International Journal for Numerical Methods in Engineering*, vol. 38, pp. 1655-1679.

**Lu, Y. Y.; Belytschko, T.; Gu, L.** (1994): A new implementation of the element free Galerkin method. *Computer Methods in Applied Mechanics and Engineering*, vol. 113, pp. 397-414.

**Madych, W.; Nelson, S.** (1990): Multivariate interpolation and conditionally positive definite functions II. *Mathematics of Computation*, vol. 44, pp. 211-230.

**Melenk, J. M.; Babuska, I.** (1996): The partition of unity finite element method: basic theory and applications. *Computer Methods in Applied Mechanics and Engineering*, vol. 139, pp. 289-314.

**Nayroles, B.; Touzot, G.; Villon, P.** (1991): The diffuse approximations. *C.R. Acad. Sci. Paris Ser. II*, vol. 313, pp. 133-138.

**Nayroles, B.; Touzot, G.; Villon, P.** (1992): Generalizing the finite element method:

diffuse approximation and diffuse elements. *Computational Mechanics*, vol. 10, pp. 307-318.

**Onate, E.; Idelsohn, S.; Zienkiewicz, O.; Taylor, R.L.** (1995): A finite point method in computational mechanics application to convective transport and fluid flow. *International Journal for Numerical Methods in Engineering*, vol. 39, pp. 3839-3866.

**Schaback, R.** (1993): *Comparisons of radial basis function interpolants, in: Multivariate Approximation: From CAGD to Wavelets*, World Scientific, Singapore.

**Simos, T. E.** (2007): Stabilization of a four-step exponentially-fitted method and its application to the Schrodinger equation. *International Journal of Modern Physics*, vol. 18, pp. 315-328.

**Simos, T. E.** (2008): A family of four-step trigonometrically-fitted methods and its application to the Schrödinger equation. *Journal of Mathematical Chemistry*, vol. 44, pp. 447-466.

**Subasi, M.** (2002): On the finite-difference schemes for the numerical solution of two dimensional Schrödinger equation. *Numerical Methods for Partial Differential Equations*, vol. 18, pp. 752-758.

**Tappert, F. D.** (1977): 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, pp. 224-287.

**Zhu, T.; Zhang, J.; Atluri, S. N.** (1998): A meshless local boundary integral equation (LBIE) method for solving nonlinear problems. *Computational Mechanics*, vol. 22, pp. 174-186.