Zagros J. Appl. Math. & Data. Anal.

Available online at https://zjamda.abru.ac.ir/
ISSN 0000-0000
Vol. 1, No. 1, 2025, 82–100
Research Article

A Partition of Unity Meshless Method for Solving Two-Dimensional Coupled Klein-Gordon-Schrödinger Equations

Ahmad Jafarabadi*
Department of Applied Mathematics, Imam Khomeini International University, Qazvin, 34149-16818, Iran

Abstract

Abstract: In this paper, we introduce a meshless method based on the Partition of Unity (PU) interpolation technique to solve two-dimensional coupled Klein-Gordon-Schrödinger (KGS) equations on scattered data within arbitrary domains. The approach begins with a time-discrete scheme derived from a finite difference approximation of the time derivative, followed by the application of the PU method to discretize the spatial derivatives. The PU technique combines local approximations using radial basis functions with compactly supported weight functions to ensure efficient and accurate interpolation over irregular node distributions. To handle the nonlinearity inherent in the KGS system, a predictor-corrector scheme is employed. Numerical experiments demonstrate the effectiveness of the PU-based method, with results compared against analytical solutions and stability analyses confirming its accuracy and computational efficiency for complex geometries.
Keywords: Partition of Unity (PU) method; Meshless interpolation; Klein-Gordon-Schrödinger (KGS) equations; Scattered data; Arbitrary domains.

1 Introduction

The coupled Klein-Gordon-Schrödinger (KGS) equations are formulated as follows [1]:

$$\begin{cases} \mathbf{i}\dfrac{\partial u(\mathbf{x},t)}{\partial t} + \beta\Delta u(\mathbf{x},t) + u(\mathbf{x},t)v(\mathbf{x},t) + \mathbf{i}\alpha u(\mathbf{x},t) = f(\mathbf{x},t), \\ \varepsilon^2\dfrac{\partial^2 v(\mathbf{x},t)}{\partial t^2} + \gamma\varepsilon\dfrac{\partial v(\mathbf{x},t)}{\partial t} - \Delta v(\mathbf{x},t) + v(\mathbf{x},t) - |u(\mathbf{x},t)|^2 = g(\mathbf{x},t), \\ \mathbf{x}\in\mathbb{R}^d, t\in[0,T], \end{cases} \tag{1}$$

*Corresponding author: jafarabdi.ahmad@yahoo.com

subject to the initial conditions

$$u(\mathbf{x},0)=u_0(\mathbf{x}), \qquad v(\mathbf{x},0)=v_0(\mathbf{x}), \qquad \frac{\partial v(\mathbf{x},t)}{\partial t}|_{t=0}=v_1(\mathbf{x}), \qquad \mathbf{x}\in\Omega, \tag{2}$$

and Dirichlet boundary conditions

$$u(\mathbf{x},t)=h_1(\mathbf{x},t), \qquad v(\mathbf{x},t)=h_2(\mathbf{x},t), \qquad \mathbf{x}\in\partial\Omega, t>0, \tag{3}$$

where $u(\mathbf{x},t)$ denotes a complex scalar nucleon field, $v(\mathbf{x},t)$ a real scalar meson field, and $\mathbf{i}=\sqrt{-1}$. This system models conserved scalar nucleons interacting with neutral scalar mesons via a Yukawa-type coupling [2], with broad applications in various physical contexts [3, 4, 5, 6, 7].

In Eq. (1), the parameter $\varepsilon\geq 0$ is inversely related to the speed of light, while $\gamma\geq 0$ and $\alpha\geq 0$ are nonnegative constants. Here, Δ represents the Laplacian operator. The terms $f(\mathbf{x},t)$ and $g(\mathbf{x},t)$ act as source functions, with f complex-valued and g real-valued.

Special cases include the following: - For $\varepsilon=1$, $\gamma=0$, $\alpha=0$, and vanishing source terms $f(\mathbf{x},t)=g(\mathbf{x},t)=0$, Eq. (1) simplifies to the classical KGS system [3]. When $\alpha>0$, a linear damping contribution is introduced into the nonlinear Schrödinger equation, i.e., the first equation of (1). For $\gamma>0$, a damping mechanism is incorporated into the Klein-Gordon part of the system, i.e., the second equation of (1). In the limiting case $\varepsilon\to 0$, the equations reduce to the Schrödinger-Yukawa (S-Y) model [1].

For the particular scenario with $\beta=1$ and homogeneous forcing terms $f(\mathbf{x},t)=g(\mathbf{x},t)=0$, further details can be found in [1, 8].

The coupled Klein-Gordon-Schrödinger (KGS) equations have been the subject of extensive analytical and numerical investigations due to their significant role in modeling the interaction between conserved scalar nucleons and neutral scalar mesons via Yukawa coupling [3, 4]. While foundational studies have established well-posedness, stability, and asymptotic behavior of solutions [3, 9, 10, 11], the development of efficient and accurate numerical methods remains crucial for simulating their complex physical behavior in realistic scenarios.

Early numerical approaches focused on finite difference and spectral methods. Seminal work by Bao and Yang [1] introduced efficient, unconditionally stable, and accurate schemes that set a benchmark for subsequent research. Conservative finite difference schemes were further analyzed by Zhang [12], ensuring structure-preserving discretizations. The multi-symplectic structure of the KGS equations was first leveraged by Kong et al. [8], who proposed a multi-symplectic mid-point scheme, while Hong et al. [13] developed explicit multi-symplectic integrators by combining symplectic Runge-Kutta and Runge-Kutta-Nyström-type methods. Dehghan and Taleei [14] studied an extension of the pseudospectral method based on the overlapping domain decomposition algorithm for the KGS equations. In [15], three meshless numerical techniques, namely Kansa's approach, the RBF pseudo-spectral method, and the generalized moving least squares method, were applied to solve the coupled Klein–Gordon–Schrödinger equations in one, two, and three dimensions. The authors discretized the time derivative via an implicit Crank–Nicolson scheme and approximated the spatial derivatives using the aforementioned meshless approaches, demonstrating the applicability and effectiveness of these methods through several numerical simulations.

As a new list of works, we mention to the following papers. In [16], a linearized and energy-conserving scheme was developed for the nonlocal Klein–Gordon–Schrödinger system with fractional Laplacian in unbounded domains, combining a Hermite–Galerkin spectral method for space with an

ESAV-based Crank–Nicolson scheme and adaptive time-stepping. The authors of proposed [17] a semi-linearized time-stepping scheme combining finite differences in time with Fourier pseudospectral discretization in space for the Klein–Gordon–Schrödinger equation, and shown to conserve mass and energy while achieving second-order temporal and spectral spatial accuracy. In [18], the blowup behavior of solutions to the Klein–Gordon–Schrödinger system with power nonlinearity in higher dimensions was studied, where perturbed virial identities and Lyapunov functionals were used to establish finite-time blowup results. The authors of [19] established optimal error estimates for two finite difference schemes applied to the multi-dimensional Klein–Gordon–Schrödinger equations, where the methods preserve mass and energy while achieving second-order accuracy in both time and space. In [20], three energy-preserving numerical schemes based on different SAV and Lagrange multiplier approaches were developed for the Klein–Gordon–Schrödinger equations, and their accuracy and efficiency were compared through numerical experiments. In [21], a linearized and unconditionally stable energy-preserving scheme was proposed for the Klein–Gordon–Schrödinger equations by reformulating the system with auxiliary variables and applying the invariant energy quadratization approach.

Our previous work [22] introduced a meshless approach based on radial basis functions (RBFs) for the KGS system. Building upon this foundation, the present study proposes an efficient partition of unity method (PUM) framework. The PUM synergistically combines the strengths of local RBF approximations-employed here with polyharmonic splines (PHS) augmented with polynomials for guaranteed solvability-and compactly supported weight functions. This combination effectively handles highly irregular node distributions and complex geometries while maintaining high-order accuracy. A robust predictor-corrector scheme is integrated to efficiently manage the system's nonlinearities.

This approach distinguishes itself from existing methods by its ability to operate on truly arbitrary, complex domains with scattered nodes, a feature not always efficiently handled by standard spectral or finite element methods. The numerical experiments presented herein demonstrate that our PUM-based scheme achieves spectral-like accuracy, competes with the efficiency of modern meshless techniques like RBF-FD, and maintains excellent long-term stability.

The remainder of this paper is structured as follows. Section 2 is devoted to the numerical methodology. We begin by introducing the temporal discretization scheme based on finite differences in Section 3. Subsequently, we provide a detailed exposition of the spatial discretization using the partition of unity meshless framework with radial basis functions. This section also includes the description of the iterative predictor-corrector algorithm implemented to handle the nonlinear terms of the KGS system efficiently. The numerical experiments and a comprehensive discussion of the results are presented in Section 4. The proposed method is validated on several test problems encompassing both regular and irregular computational domains. Finally, concluding remarks is summarized in Section 5.

2 A brief review of the proposed method

2.1 Radial Basis Function Collocation Schemes

Let N distinct points $x_1, \ldots, x_N \in \mathbb{R}^d$ be given, together with the corresponding scalar data values $u(x_1), u(x_2), \ldots, u(x_N)$. The objective of the Radial Basis Function (RBF) interpolation problem is to construct an interpolant of the form:

$$s(x) = \sum_{j=1}^{N} a_j \phi(\|x - x_j\|) + \sum_{k=1}^{Q} b_k p_k(x), \tag{4}$$

where ϕ denotes a radial basis function, $\{p_k(\mathbf{x})\}_{k=1}^{Q}$ is a basis for the d-variate polynomial space of total degree up to $m-1$ (where m is the order of conditional positive definiteness of ϕ), $\|\cdot\|$ is the Euclidean norm, $a_j \in \mathbb{R}$ and $b_k \in \mathbb{R}$ are the unknown coefficients. These coefficients are determined by enforcing the interpolation conditions $s(x_i) = u(x_i)$ alongside the additional constraints $\sum_{j=1}^{N} a_j p_k(\mathbf{x}_j) = 0$ for $k = 1, \ldots, Q$, to ensure uniqueness. This requirement yields the block linear system:

$$\begin{bmatrix} A & P \\ P^T & 0 \end{bmatrix} \begin{bmatrix} \underline{a} \\ \underline{b} \end{bmatrix} = \begin{bmatrix} \underline{u} \\ \underline{0} \end{bmatrix}, \tag{5}$$

with entries $A_{ij} = \phi(\|x_i - x_j\|)$ and $P_{ik} = p_k(\mathbf{x}_i)$, while $\underline{u} = [u(x_1), \ldots, u(x_N)]^{\mathrm{T}}$, $\underline{a} = [a_1, \ldots, a_N]^{\mathrm{T}}$, and $\underline{b} = [b_1, \ldots, b_Q]^{\mathrm{T}}$. For conditionally positive definite basis functions ϕ of order m, the system matrix is nonsingular provided that the data points are Π_{m-1}^{d}-unisolvent, ensuring the existence of a unique solution.

Alternatively, the interpolant can be represented by introducing cardinal basis functions $\psi_j(x)$ which incorporate the polynomial augmentation and satisfy the Kronecker delta property:

$$\psi_j(x_i) = \begin{cases} 1, & \text{if } i = j, \\ 0, & \text{if } i \neq j. \end{cases} \tag{6}$$

This leads to an equivalent expression for the interpolant:

$$s(x) = \sum_{j=1}^{N} \psi_j(x) u(x_j). \tag{7}$$

For a linear differential operator $\mathcal{L}$, we have:

$$\mathcal{L}s(x) = \sum_{j=1}^{N} \mathcal{L}\psi_j(x) u(x_j). \tag{8}$$

The differentiation process is then encoded in the matrix $\Psi_{\mathcal{L}}$, allowing us to compute:

$$\underline{s}_{\mathcal{L}} = \Psi_{\mathcal{L}} \underline{u}. \tag{9}$$

When dealing with time-dependent partial differential equations, the solution $u(x,t)$ is approximated in the form:

$$s(x,t) = \sum_{j=1}^{N} \psi_j(x) u_j(t), \tag{10}$$

where $u_j(t) \approx u(x_j, t)$ are the unknown time-dependent coefficients to be determined.

2.2 Radial Basis Function-based PUM

In the RBF-PUM framework, the global approximation $s(x)$ is formulated through a partition of unity representation:

$$s(x) = \sum_{j=1}^{M} w_j(x) s_j(x), \tag{11}$$

where each $s_j(x)$ denotes a local RBF interpolant defined over the subdomain (or patch) Ω_j, and the functions $w_j(x)$ are partition weights constructed using the Shepard approach [23]:

$$w_j(x) = \frac{\varphi_j(x)}{\sum_{k \in I(x)} \varphi_k(x)}. \tag{12}$$

Here, the compactly supported functions $\varphi_j(x)$ are typically chosen as Wendland functions. By construction, the weight functions satisfy the partition of unity condition:

$$\sum_{j \in I(x)} w_j(x) = 1. \tag{13}$$

Therefore, the global interpolant can equivalently be expressed as:

$$s(x) = \sum_{j \in I(x)} w_j(x) s_j(x). \tag{14}$$

For time-dependent problems, the approximate solution $u(x,t)$ is represented by:

$$s(x,t) = \sum_{j \in I(x)} w_j(x) s_j(x,t), \tag{15}$$

where each local term $s_j(x,t)$ is further approximated as:

$$s_j(x,t) = \sum_{k \in J(\Omega_j)} \psi_k(x) u_k(t). \tag{16}$$

Combining these expressions, the global time-dependent approximation becomes:

$$s(x,t) = \sum_{j \in I(x)} \sum_{k \in J(\Omega_j)} w_j(x) \psi_k(x) u_k(t). \tag{17}$$

For spatial derivatives, applying the Leibniz rule yields:

$$\frac{\partial^{|\sigma|}}{\partial x^\sigma} s(x,t) = \sum_{j \in I(x)} \sum_{k \in J(\Omega_j)} \frac{\partial^{|\sigma|}}{\partial x^\sigma} \left(w_j(x) \psi_k(x) \right) u_k(t). \tag{18}$$

The density of nodes and patches is quantified through the local and global fill distances, denoted by h and H, respectively:

$$h = \max\{h_j\}, \quad H = \sup_{x \in \Omega} \min_{1 \le j \le M} \|x - X_j\|.$$

For uniform discretizations, h is proportional to the nodal spacing, while H reflects the characteristic size of the patches.

3 Time discretize scheme

Let us denote

$$t_k = k\delta t, \qquad k = 0, 1, ..., M,$$

where $\delta t = T/M$ represents the time step size. In this part, the temporal variable is discretized using a combination of forward and central finite difference formulas for the first- and second-order time derivatives, respectively, together with an appropriate Crank–Nicolson treatment. Considering Eq. (1) at the grid point $(\mathbf{x}, t_{k+1})$, we obtain

$$\begin{cases} \mathbf{i}\dfrac{u^{k+1}(\mathbf{x}) - u^k(\mathbf{x})}{\delta t} + \dfrac{\beta}{2}\big(\Delta u^{k+1}(\mathbf{x}) + \Delta u^k(\mathbf{x})\big) + u^k(\mathbf{x})v^k(\mathbf{x}) + \dfrac{\mathbf{i}\alpha}{2}\big(u^{k+1}(\mathbf{x}) + u^k(\mathbf{x})\big) \\ = \dfrac{1}{2}\big(f^{k+1}(\mathbf{x}) + f^k(\mathbf{x})\big), \\ \varepsilon^2\dfrac{v^{k+1}(\mathbf{x}) - 2v^k(\mathbf{x}) + v^{k-1}(\mathbf{x})}{\delta t^2} + \gamma\varepsilon\dfrac{v^{k+1}(\mathbf{x}) - v^{k-1}(\mathbf{x})}{2\delta t} - \dfrac{1}{3}\big(\Delta v^{k+1}(\mathbf{x}) + \Delta v^k(\mathbf{x}) + \Delta v^{k-1}(\mathbf{x})\big) \\ +\dfrac{1}{3}\big(v^{k+1}(\mathbf{x}) + v^k(\mathbf{x}) + v^{k-1}(\mathbf{x})\big) - |u^k(\mathbf{x})|^2 = \dfrac{1}{3}\big(g^{k+1}(\mathbf{x}) + g^k(\mathbf{x}) + g^{k-1}(\mathbf{x})\big), \end{cases} \tag{19}$$

where $\mathbf{i} = \sqrt{-1}$, and $u^{k+1}(\mathbf{x})$, $v^{k+1}(\mathbf{x})$ denote the approximations of the unknown fields at $(\mathbf{x}, t_{k+1})$. Equation (19) can be equivalently rewritten in the form

$$\begin{cases} (\dfrac{2\mathbf{i}}{\delta t} + \mathbf{i}\alpha)u^{k+1}(\mathbf{x}) + \beta\Delta u^{k+1}(\mathbf{x}) = (\dfrac{2\mathbf{i}}{\delta t} - \mathbf{i}\alpha)u^k(\mathbf{x}) - \beta\Delta u^k(\mathbf{x}) - 2u^k(\mathbf{x})v^k(\mathbf{x}) + F(\mathbf{x}; k), \\ \big(\dfrac{3\varepsilon^2}{\delta t^2} + \dfrac{3\gamma\varepsilon}{2\delta t} + 1\big)v^{k+1}(\mathbf{x}) - \Delta v^{k+1}(\mathbf{x}) = \big(\dfrac{6\varepsilon^2}{\delta t^2} - 1\big)v^k(\mathbf{x}) + \Delta v^k(\mathbf{x}) + \big(\dfrac{3\gamma\varepsilon}{2\delta t} - \dfrac{3\varepsilon^2}{\delta t^2} - 1\big)v^{k-1}(\mathbf{x}) \\ +\Delta v^{k-1}(\mathbf{x}) + 3|u^k(\mathbf{x})|^2 + G(\mathbf{x}; k), \end{cases} \tag{20}$$

with $F(\mathbf{x}; k) = f^{k+1}(\mathbf{x}) + f^k(\mathbf{x})$ and $G(\mathbf{x}; k) = g^{k+1}(\mathbf{x}) + g^k(\mathbf{x}) + g^{k-1}(\mathbf{x})$.

Now, let us assume N nodes are arbitrarily distributed across the boundary and inside the computational domain. Provided that the discrete values $\{u(\mathbf{x}_i, k\delta t), v(\mathbf{x}_i, k\delta t)\}_{i=1}^N$ are known, the task is to compute the next time level values $\{u(\mathbf{x}_i, (k+1)\delta t), v(\mathbf{x}_i, (k+1)\delta t)\}_{i=1}^N$. Consequently, there are $2N$ unknowns, and to determine them, one requires $2N$ independent equations. As explained, one equation corresponds to each node. Substituting the interior nodes $\mathbf{x}_i$, $i = 1, 2, ..., N_\Omega$, of Ω into Eq. (20), we focus on the two-dimensional case of the KGS equations.

3.1 Spatial discretization by PUM

Substituting the PUM approximation (17) into Eq. (20), for the two–dimensional case we obtain

$$\begin{cases} \left(\frac{2\mathbf{i}}{\delta t}+\mathbf{i}\alpha\right)\sum_{j=1}^{N}\Phi_j(x_i,y_i)u_j^{k+1}+\beta\left(\sum_{j=1}^{N}\frac{\partial^2\Phi_j(x_i,y_i)}{\partial x^2}u_j^{k+1}+\sum_{j=1}^{N}\frac{\partial^2\Phi_j(x_i,y_i)}{\partial y^2}u_j^{k+1}\right) \\ =\left(\frac{2\mathbf{i}}{\delta t}-\mathbf{i}\alpha\right)\sum_{j=1}^{N}\Phi_j(x_i,y_i)u_j^{k}-\beta\left(\sum_{j=1}^{N}\frac{\partial^2\Phi_j(x_i,y_i)}{\partial x^2}u_j^{k}+\sum_{j=1}^{N}\frac{\partial^2\Phi_j(x_i,y_i)}{\partial y^2}u_j^{k}\right) \\ -2\Psi^k(x_i,y_i)+F(x_i,y_i;k), \\ \left(\frac{3\varepsilon^2}{\delta t^2}+\frac{3\gamma\varepsilon}{2\delta t}+1\right)\sum_{j=1}^{N}\Phi_j(x_i,y_i)v_j^{k+1}-\left(\sum_{j=1}^{N}\frac{\partial^2\Phi_j(x_i,y_i)}{\partial x^2}v_j^{k+1}+\sum_{j=1}^{N}\frac{\partial^2\Phi_j(x_i,y_i)}{\partial y^2}v_j^{k+1}\right) \\ =\left(\frac{6\varepsilon^2}{\delta t^2}-1\right)\sum_{j=1}^{N}\Phi_j(x_i,y_i)v_j^{k}+\sum_{j=1}^{N}\frac{\partial^2\Phi_j(x_i,y_i)}{\partial x^2}v_j^{k}+\sum_{j=1}^{N}\frac{\partial^2\Phi_j(x_i,y_i)}{\partial y^2}v_j^{k} \\ +\left(\frac{3\gamma\varepsilon}{2\delta t}-\frac{3\varepsilon^2}{\delta t^2}-1\right)\sum_{j=1}^{N}\Phi_j(x_i,y_i)v_j^{k-1}+\sum_{j=1}^{N}\frac{\partial^2\Phi_j(x_i,y_i)}{\partial x^2}v_j^{k-1}+\sum_{j=1}^{N}\frac{\partial^2\Phi_j(x_i,y_i)}{\partial y^2}v_j^{k-1} \\ +3\Lambda^k(x_i,y_i)+G(x_i,y_i;k), \end{cases} \tag{21}$$

for $i=1,2,\ldots,N_\Omega$, where $\Psi^k(x_i,y_i)=u^k(x_i,y_i)v^k(x_i,y_i)$ and $\Lambda^k(x_i,y_i)=|u^k(x_i,y_i)|^2$.

Using the notation of local PU basis functions and their derivatives, system (21) can be expressed as

$$\begin{cases} \left(\frac{2\mathbf{i}}{\delta t}+\mathbf{i}\alpha\right)u_i^{k+1}+\beta\sum_{j=1}^{N}\left(D_{x,ij}^{(2)}+D_{y,ij}^{(2)}\right)u_j^{k+1}=\left(\frac{2\mathbf{i}}{\delta t}-\mathbf{i}\alpha\right)u_i^{k}-\beta\sum_{j=1}^{N}\left(D_{x,ij}^{(2)}+D_{y,ij}^{(2)}\right)u_j^{k} \\ -2\Psi^k(x_i,y_i)+F(x_i,y_i;k), \\ \left(\frac{3\varepsilon^2}{\delta t^2}+\frac{3\gamma\varepsilon}{2\delta t}+1\right)v_i^{k+1}-\sum_{j=1}^{N}\left(D_{x,ij}^{(2)}+D_{y,ij}^{(2)}\right)v_j^{k+1}=\left(\frac{6\varepsilon^2}{\delta t^2}-1\right)v_i^{k}+\sum_{j=1}^{N}\left(D_{x,ij}^{(2)}+D_{y,ij}^{(2)}\right)v_j^{k} \\ +\left(\frac{3\gamma\varepsilon}{2\delta t}-\frac{3\varepsilon^2}{\delta t^2}-1\right)v_i^{k-1}+\sum_{j=1}^{N}\left(D_{x,ij}^{(2)}+D_{y,ij}^{(2)}\right)v_j^{k-1}+3\Lambda^k(x_i,y_i)+G(x_i,y_i;k), \end{cases} \tag{22}$$

where $D_{x,ij}^{(2)}$ and $D_{y,ij}^{(2)}$ denote differentiation matrices obtained from PUM basis functions with respect to x and y.

For boundary nodes $i=N_\Omega+1,\ldots,N$ lying on $\partial\Omega$, the Dirichlet conditions (3) yield

$$u_i^{k+1}=h_1^{k+1}(x_i,y_i),\qquad v_i^{k+1}=h_2^{k+1}(x_i,y_i). \tag{23}$$

Let $N=N_\Omega+N_{\partial\Omega}$ be the total number of nodes. The discrete system can be assembled into the block matrix form

$$\mathbf{A}\mathbf{U}^{k+1}=\mathbf{B}\mathbf{U}^{k}+\mathbf{C}\mathbf{U}^{k-1}+\mathbf{D}+\mathbf{E}, \tag{24}$$

where

$$\mathbf{A}=\begin{pmatrix}[A_1]_{N\times N} & [0]_{N\times N}\\ [0]_{N\times N} & [A_2]_{N\times N}\end{pmatrix}_{2N\times 2N},\qquad \mathbf{B}=\begin{pmatrix}[B_1]_{N\times N} & [0]_{N\times N}\\ [0]_{N\times N} & [B_2]_{N\times N}\end{pmatrix}_{2N\times 2N}, \tag{25}$$

$$\mathbf{C}=\begin{pmatrix}[0]_{N\times N} & [0]_{N\times N}\\ [0]_{N\times N} & [C_2]_{N\times N}\end{pmatrix}_{2N\times 2N},\qquad \mathbf{D}=\begin{pmatrix}[D_1]_{N\times 1}\\ [D_2]_{N\times 1}\end{pmatrix}_{2N\times 1},\qquad \mathbf{E}=\begin{pmatrix}[E_1]_{N\times 1}\\ [E_2]_{N\times 1}\end{pmatrix}_{2N\times 1}. \tag{26}$$

The entries of these block matrices are defined as

$$(A_1)_{ij}=\begin{cases}\left(\dfrac{2\mathbf{i}}{\delta t}+\mathbf{i}\alpha\right)\delta_{ij}+\beta\left(D^{(2)}_{x,ij}+D^{(2)}_{y,ij}\right), & i=1,\dots,N_\Omega,\\ \delta_{ij}, & i=N_\Omega+1,\dots,N,\end{cases} \tag{27}$$

$$(A_2)_{ij}=\begin{cases}\left(\dfrac{3\varepsilon^2}{\delta t^2}+\dfrac{3\gamma\varepsilon}{2\delta t}+1\right)\delta_{ij}-\left(D^{(2)}_{x,ij}+D^{(2)}_{y,ij}\right), & i=1,\dots,N_\Omega,\\ \delta_{ij}, & i=N_\Omega+1,\dots,N,\end{cases} \tag{28}$$

$$(B_1)_{ij}=\begin{cases}\left(\dfrac{2\mathbf{i}}{\delta t}-\mathbf{i}\alpha\right)\delta_{ij}-\beta\left(D^{(2)}_{x,ij}+D^{(2)}_{y,ij}\right), & i=1,\dots,N_\Omega,\\ 0, & i=N_\Omega+1,\dots,N,\end{cases} \tag{29}$$

$$(B_2)_{ij}=\begin{cases}\left(\dfrac{6\varepsilon^2}{\delta t^2}-1\right)\delta_{ij}+\left(D^{(2)}_{x,ij}+D^{(2)}_{y,ij}\right), & i=1,\dots,N_\Omega,\\ 0, & i=N_\Omega+1,\dots,N,\end{cases} \tag{30}$$

$$(C_2)_{ij}=\begin{cases}\left(\dfrac{3\gamma\varepsilon}{2\delta t}-\dfrac{3\varepsilon^2}{\delta t^2}-1\right)\delta_{ij}+\left(D^{(2)}_{x,ij}+D^{(2)}_{y,ij}\right), & i=1,\dots,N_\Omega,\\ 0, & i=N_\Omega+1,\dots,N,\end{cases} \tag{31}$$

$$(D_1)_i=\begin{cases}-2\Psi^k(x_i,y_i), & i=1,\dots,N_\Omega,\\ 0, & i=N_\Omega+1,\dots,N,\end{cases}\qquad (D_2)_i=\begin{cases}3\Lambda^k(x_i,y_i), & i=1,\dots,N_\Omega,\\ 0, & i=N_\Omega+1,\dots,N,\end{cases} \tag{32}$$

$$(E_1)_i=\begin{cases}F(x_i,y_i;k), & i=1,\dots,N_\Omega,\\ h_1(x_i,y_i), & i=N_\Omega+1,\dots,N,\end{cases}\qquad (E_2)_i=\begin{cases}G(x_i,y_i;k), & i=1,\dots,N_\Omega,\\ h_2(x_i,y_i), & i=N_\Omega+1,\dots,N.\end{cases} \tag{33}$$

Finally, the vector of unknowns is given by

$$\mathbf{U}^{k+1}=\left(u_1^{k+1},u_2^{k+1},\dots,u_N^{k+1},v_1^{k+1},v_2^{k+1},\dots,v_N^{k+1}\right)^{tr}. \tag{34}$$

At the initial level $k = 0$, using the prescribed initial conditions (2), we set

$$\mathbf{U}^0 = \begin{pmatrix} [\mathbf{u}_0]_{N\times 1} \\ [\mathbf{v}_0]_{N\times 1} \end{pmatrix}_{2N\times 1}, \qquad \mathbf{U}^{-1} = \mathbf{U}^1 - 2\delta t \begin{pmatrix} [0]_{N\times 1} \\ [\mathbf{v}_1]_{N\times 1} \end{pmatrix}_{2N\times 1}, \tag{35}$$

where

$$\mathbf{u}_0 = (u_0(x_1, y_1), u_0(x_2, y_2), \ldots, u_0(x_N, y_N))^{tr}, \quad \mathbf{v}_0 = (v_0(x_1, y_1), v_0(x_2, y_2), \ldots, v_0(x_N, y_N))^{tr},$$

$$\mathbf{v}_1 = (v_1(x_1, y_1), v_1(x_2, y_2), \ldots, v_1(x_N, y_N))^{tr}.$$

Here δ_{ij} denotes the Kronecker delta

$$\delta_{ij} = \begin{cases} 1, & i = j, \\ 0, & i \neq j. \end{cases} \tag{36}$$

Remark: To handle the nonlinear terms in the system (24), an iterative predictor-corrector scheme is employed [24]. The vector $\mathbf{D}$, which contains the nonlinear terms, is updated iteratively at each time level until convergence is achieved.

Initially, for $k = 0$, the nonlinear vector $\mathbf{D}$ is constructed using the initial conditions:

$$\mathbf{D}^0 = \begin{pmatrix} -2\Psi^0(\mathbf{x}_i) \\ 3\Lambda^0(\mathbf{x}_i) \end{pmatrix}_{2N\times 1}, \quad i = 1, 2, \ldots, N,$$

where $\Psi^0(\mathbf{x}_i) = u^0(\mathbf{x}_i)v^0(\mathbf{x}_i)$ and $\Lambda^0(\mathbf{x}_i) = |u^0(\mathbf{x}_i)|^2$. Equation (24) is then solved as a linear system for the unknown $\mathbf{U}^1$.

For subsequent time levels $k + 1$, the following iterative process is used:

1. Set $l = 1$ and initialize $\mathbf{U}_0^{k+1}$ (e.g., by extrapolation from previous time steps).

2. Compute the nonlinear vector for the corrector step:

$$\mathbf{D}_l = \frac{1}{2}\begin{pmatrix} -2\Psi^k(\mathbf{x}_i) \\ 3\Lambda^k(\mathbf{x}_i) \end{pmatrix}_{2N\times 1} + \frac{1}{2}\begin{pmatrix} -2\Psi_{l-1}^{k+1}(\mathbf{x}_i) \\ 3\Lambda_{l-1}^{k+1}(\mathbf{x}_i) \end{pmatrix}_{2N\times 1},$$

   where $\Psi_{l-1}^{k+1}(\mathbf{x}_i) = u_{l-1}^{k+1}(\mathbf{x}_i)v_{l-1}^{k+1}(\mathbf{x}_i)$ and $\Lambda_{l-1}^{k+1}(\mathbf{x}_i) = |u_{l-1}^{k+1}(\mathbf{x}_i)|^2$.

3. Solve the linear system (24) for $\mathbf{U}_l^{k+1}$ using $\mathbf{D}_l$.

4. Check the convergence criterion:

$$\|\mathbf{U}_l^{k+1} - \mathbf{U}_{l-1}^{k+1}\|_\infty < \epsilon,$$

   where $\epsilon = 10^{-15}$. If converged, set $\mathbf{U}^{k+1} = \mathbf{U}_l^{k+1}$ and proceed to the next time level. Otherwise, set $l = l + 1$ and return to step 2.

This iterative process effectively linearizes the nonlinear system at each time level, ensuring accurate and stable solution advancement.

4 Computational outcomes

This section is devoted to assessing the numerical performance of the proposed methodology through its application to two benchmark problems. The accuracy and stability of the presented scheme are evaluated by executing simulations across a range of values for the final time T, the nodal spacing h (in either the x or y direction), the number of circular patches M, patch fill distance H, patch radius R, and the time step size δt. In the subsequent test cases, we use circular patches to ensure complete coverage of the domain and its boundaries. By adjusting the patch radii, it can be guaranteed that overlapping between patches occurs. The radius is set as $R = \frac{\sqrt{2}}{2} c_R H$, where the coefficient c_R is determined based on the value of h to achieve optimal results.

To quantify the precision and convergence rate of the method, we employ the maximum absolute error, defined as:

$$\|u_{\text{exact}} - u_{\text{approx}}\|_\infty = \max_{1\le i\le N} |u_{\text{exact}}(\mathbf{x}_i, T) - u_{\text{approx}}(\mathbf{x}_i, T)|, \tag{37}$$

where u_{exact} and u_{approx} denote the exact and numerically approximated solutions, respectively.

In our implementation, the polyharmonic spline (PHS) is selected as the radial basis function in Eq. (4). This RBF is mathematically expressed by:

$$\phi(r) = r^{2\lambda} \ln(r), \qquad r = \|\mathbf{x} - \mathbf{x}_i\|, \qquad \lambda \in \mathbb{Z}^+, \tag{38}$$

and is employed within a support domain centered at the point $\mathbf{x}_i$. For solving the second-order PDE system (1), the parameter $\lambda = 2$ is chosen for the polyharmonic splines. A polynomial basis of order 6 is appended to the RBF approximation in Eq. (4), which is given by the set:

$$\{x^{i-j} y^j \mid j = 0, 1, \ldots, i;\ i = 0, 1, \ldots, 6\}. \tag{39}$$

4.1 Example 1

Taking Eqs. (1)-(3) under the parameter set $\varepsilon = 1$, $\alpha = 0$, $\beta = 1$, and $\gamma = 0$, the analytical solutions are provided as follows [15, 22]:

$$\begin{cases} u(x, y, t) = t\,\text{sech}(x + y)\cos(-x - y) + \mathbf{i}t\,\text{sech}(x + y)\sin(-x - y), \\ \\ v(x, y, t) = -t\,\text{sech}(x + y). \end{cases} \tag{40}$$

The functions $f(x, y, t)$ and $g(x, y, t)$, alongside the initial and boundary conditions, can be derived directly from these analytical solutions.

The numerical outcomes for this example, summarized in Table 1 and depicted in Figures 1-4, demonstrate the efficacy of the proposed PU meshless method. The data in Table 1 clearly illustrates the convergence of the scheme; as the spatial fill distance (h) and time step (δt) are refined, the maximum absolute errors for both the real and imaginary parts of u and the field v decrease significantly, reaching the order of 10^{-7} to 10^{-8} for the finest discretization. This confirms the high accuracy of the spatial and temporal discretization techniques employed. Furthermore, the notable improvement in the condition number of the system matrix $\mathbf{A}$ with refinement (from 10^{20} to 10^4) indicates enhanced numerical stability of the algorithm for smaller h and δt. The ability of the method to handle complex geometries is also confirmed by the results on the irregular domain Ω_1

Table 1: Numerical results of maximum absolute error for real and imaginary parts of $u(x, y, T)$ and $v(x, y, T)$ on $[0, 1]^2$ at $T = 1$ with $M = 25$ for Example 1.

	$Re(u)$	$Im(u)$	v	$cond(\mathbf{A})$
$h = 1/10, \delta t = 1/10$	$9.6079e-01$	$5.8828e-01$	$9.9021e-01$	$5.7541e+20$
$h = 1/14, \delta t = 1/30$	$9.7988e-01$	$5.6309e-01$	$1.0351e+00$	$5.5781e+19$
$h = 1/20, \delta t = 1/50$	$2.0316e-05$	$1.9938e-05$	$7.8397e-04$	$4.1657e+04$
$h = 1/26, \delta t = 1/100$	$9.4697e-06$	$9.6601e-06$	$3.9419e-04$	$6.5298e+04$

shown in Figures 1 and 2, where it maintains good accuracy despite the non-rectangular shape. The temporal evolution of the absolute errors, presented in Figures 3 and 4 for $T \in \{1, 2, 3, 4\}$, shows that the error remains bounded and controlled over extended simulation times, further validating the long-term stability of the method.

4.2 Example 2

As the second example, our focus is on the Klein–Gordon–Schrödinger system given by Eqs. (1)-(3), where the parameters are assigned the values $\varepsilon = 1$, $\alpha = 0$, $\beta = 1$, and $\gamma = 0$. The closed-form solutions for this configuration are presented below [15, 22]:

$$\begin{cases} u(x, y, t) = t \sin(x + y) + \mathbf{i}t \cos(x + y), \\ \\ v(x, y, t) = t \exp(-x^2 - y^2). \end{cases} \tag{41}$$

Consequently, the expressions for $f(x, y, t)$ and $g(x, y, t)$, as well as the requisite initial and boundary conditions, can be determined based on the aforementioned exact solutions.

The numerical experiments for this example further substantiate the robustness and versatility of the proposed partition of unity method. The error analysis presented in Table 2 reveals a similar trend of convergence as observed in Example 1. Refining the spatial and temporal discretization leads to a drastic reduction in the maximum absolute errors for all field components, ultimately achieving machine precision levels (on the order of 10^{-8}). This underscores the high accuracy inherent in the method's formulation. The significant decrease in the condition number of the system matrix $\mathbf{A}$ with mesh refinement also confirms the improved numerical stability of the algorithm for finer resolutions. Figure 5 provides a visual comparison between the exact and approximate solutions on the domain $[-1, 1]^2$, demonstrating an excellent visual agreement and a high-fidelity reconstruction of the complex wave fields. The corresponding pointwise absolute errors, depicted in Figure 6, are minimal and uniformly distributed, confirming the consistency of the high accuracy across the entire computational domain. Finally, the results in Figure 7 highlight a key strength of the meshless approach: its ability to maintain accuracy on irregular node distributions, such

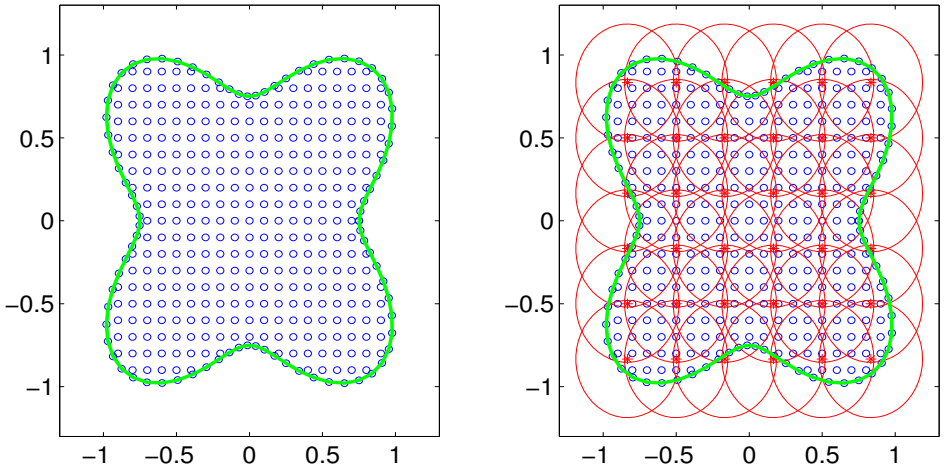

Figure 1: Uniform node distribution for the irregular domain $\Omega_1 = \{(r, \theta) : r = 1 - 1/4 \cos 4\theta,\ 0 \leq \theta \leq 2\pi\}$, with partitioning using $h = 2/20$. Initially, $N = 441$ nodes are selected within the square domain $[-1, 1]^2$, and the desired region Ω_1 is obtained, resulting in $M = 36$ circular patches.

as those generated using Halton sequences. The method shows consistent convergence behavior even when the number of interior nodes is increased within the same patch structure, proving its effectiveness for problems where structured meshing is challenging or impossible.

5 Conclusion

This paper has presented an implementation of the Partition of Unity meshless method for solving the two-dimensional coupled Klein-Gordon-Schrödinger equations. The numerical scheme combines a finite difference time discretization with a spatial approximation based on radial basis functions within a partition of unity framework. The method demonstrates several advantageous properties, including effective handling of complex geometries and irregular node distributions, high accuracy with errors reaching machine precision for refined discretizations, good numerical stability over extended time simulations, and flexibility in accommodating different domain configurations. Numerical experiments on both regular and irregular domains confirm the reliability and efficiency of the approach. The results indicate that the method provides a viable alternative to mesh-based techniques for solving coupled nonlinear partial differential equations on complex domains.

Acknowledgements

The author thanks the anonymous reviewer for his (her) valuable comments.

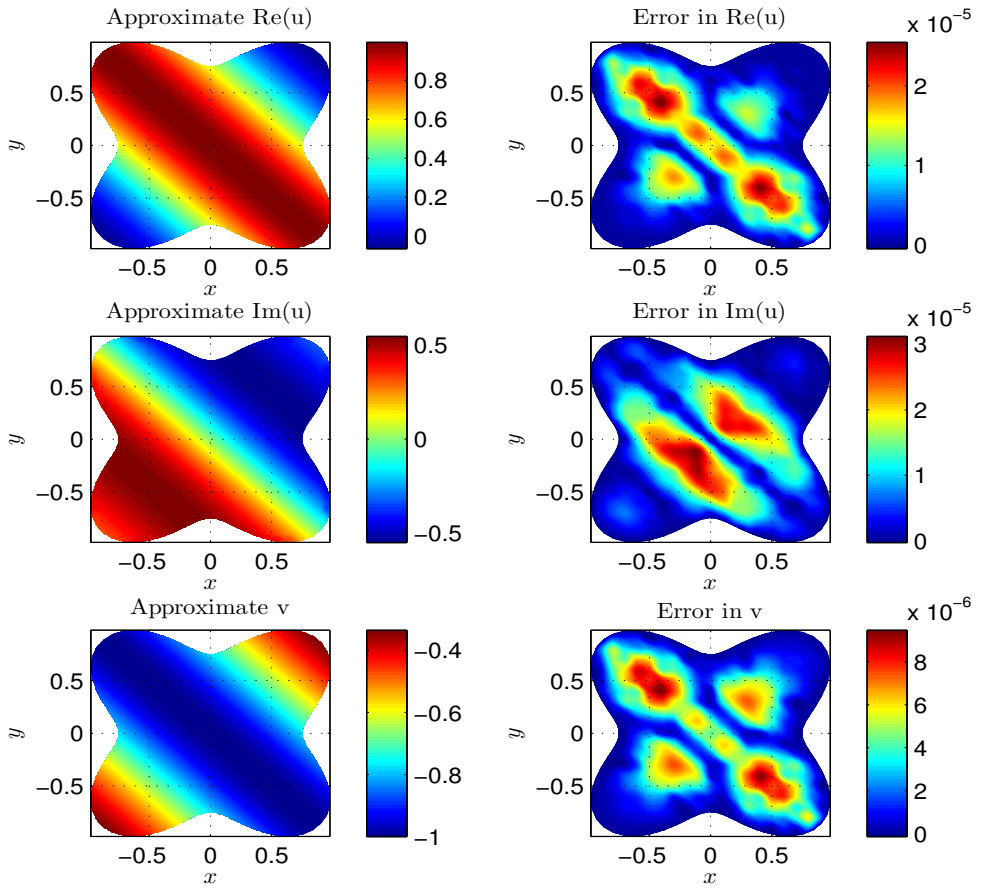

Figure 2: Graphs of approximate solution and absolute error for real and imaginary parts of $u(x, y, T)$, and $v(x, y, T)$ by means of $h = 2/20(N = 441)$, $\delta t = 1/30$ and $T = 1$, on the irregular domain Ω_1 for Example 1.

Table 2: Numerical results of maximum absolute error for real and imaginary parts of $u(x, y, T)$ and $v(x, y, T)$ on $[0, 1]^2$ at $T = 1$ with $M = 25$ for Example 2.

	$Re(u)$	$Im(u)$	v	$cond(\mathbf{A})$
$h = 1/10, \delta t = 1/10$	$1.0634e+00$	$9.8007e-01$	$9.8020e-01$	$5.7541e+20$
$h = 1/14, \delta t = 1/30$	$1.0140e+00$	$9.8990e-01$	$1.0189e+00$	$5.5781e+19$
$h = 1/20, \delta t = 1/50$	$3.7298e-08$	$2.4980e-08$	$6.6085e-07$	$4.1657e+04$
$h = 1/26, \delta t = 1/100$	$2.3350e-08$	$1.6559e-08$	$1.8283e-07$	$6.5298e+04$

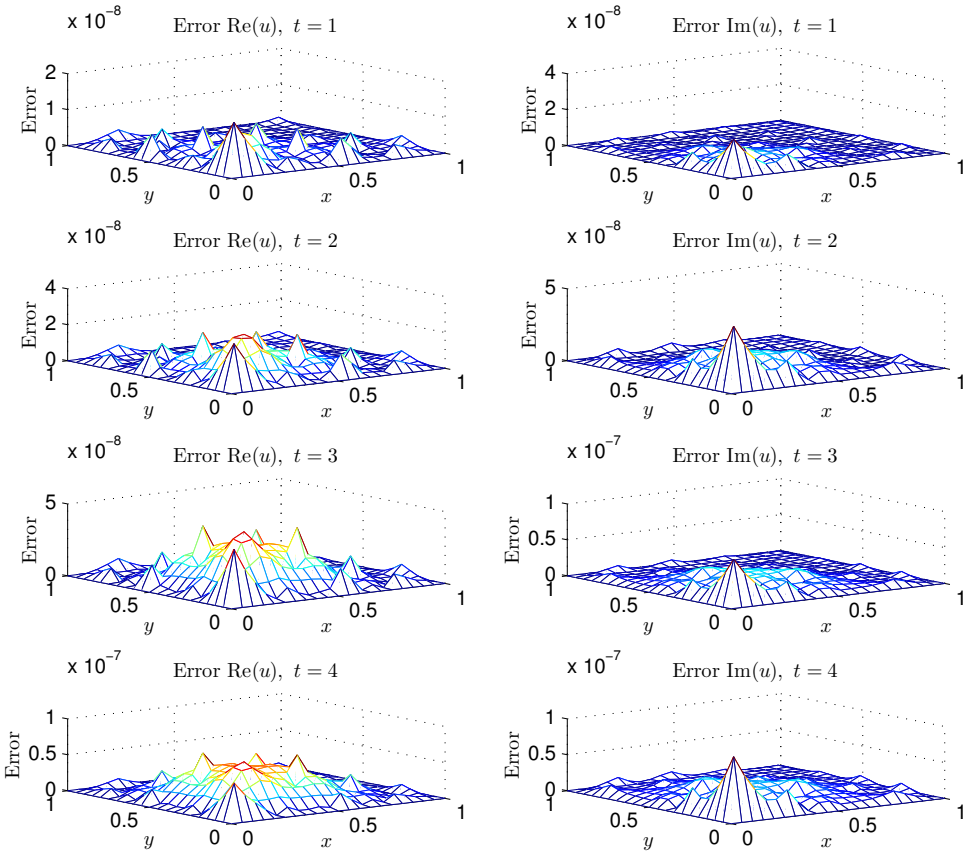

Figure 3: Graphs of absolute error for real and imaginary parts of $u(x, y, T)$ by means of $h = 1/20(N = 441)$, $\delta t = 1/50$ and $T \in \{1, 2, 3, 4\}$, on $[0, 1]^2$ for Example 1.

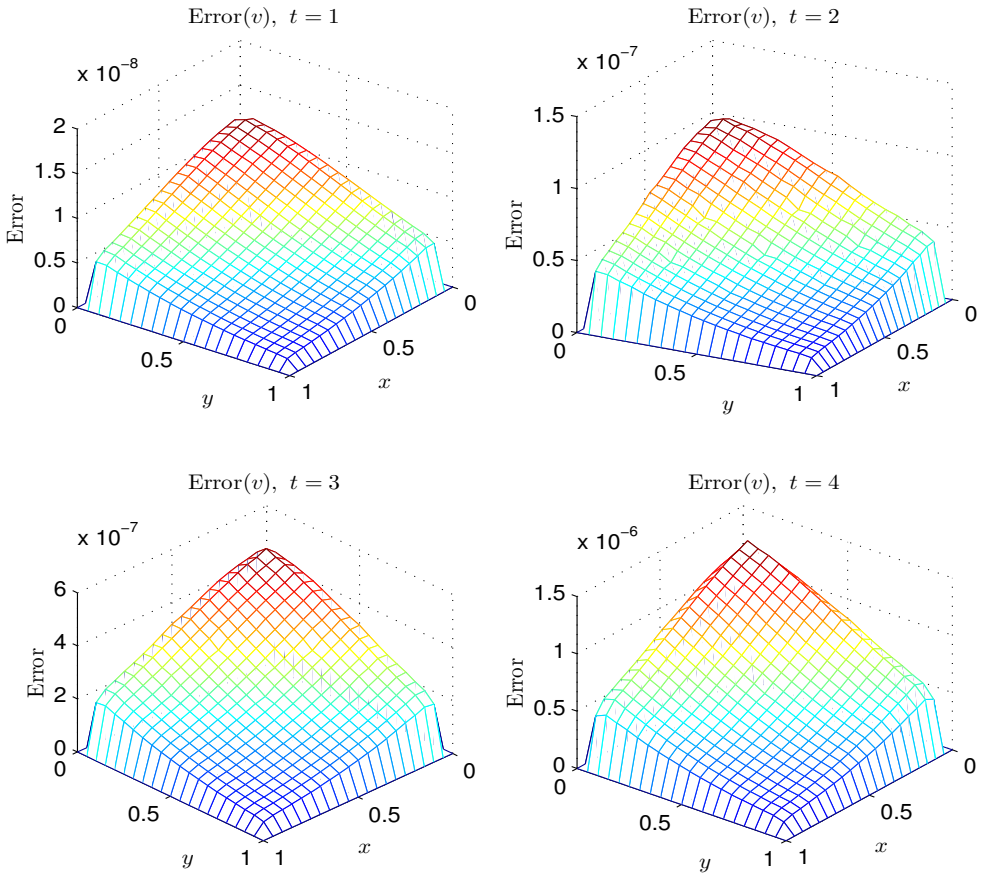

Figure 4: Graphs of absolute error for $v(x, y, T)$ by means of $h = 1/20(N = 441)$, $\delta t = 1/50$ and $T \in \{1, 2, 3, 4\}$, on $[0, 1]^2$ for Example 1.

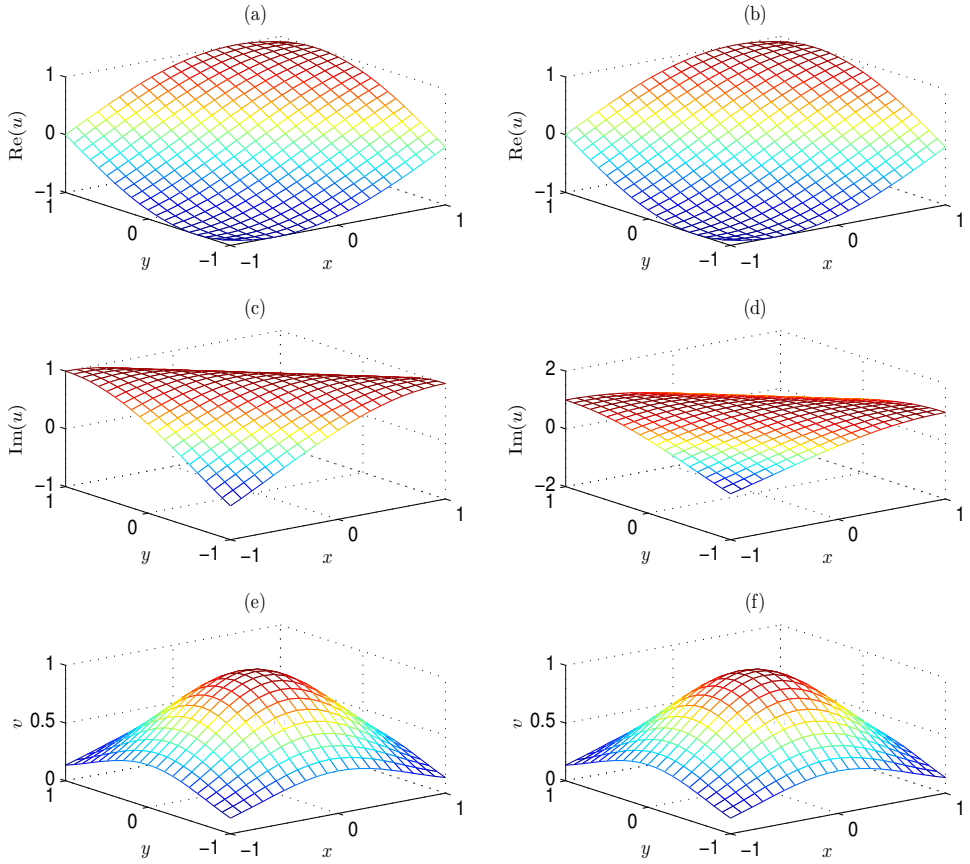

Figure 5: The exact (a, c, e) and approximate (b, d, f) solutions of real and imaginary parts of the nucleon field u and the meson field v with $h = 2/20(N = 441)$, $\delta t = 1/10$, $M = 49$ circular patches, $T = 1$, on the computational domain $[-1, 1]^2$, for Example 2.

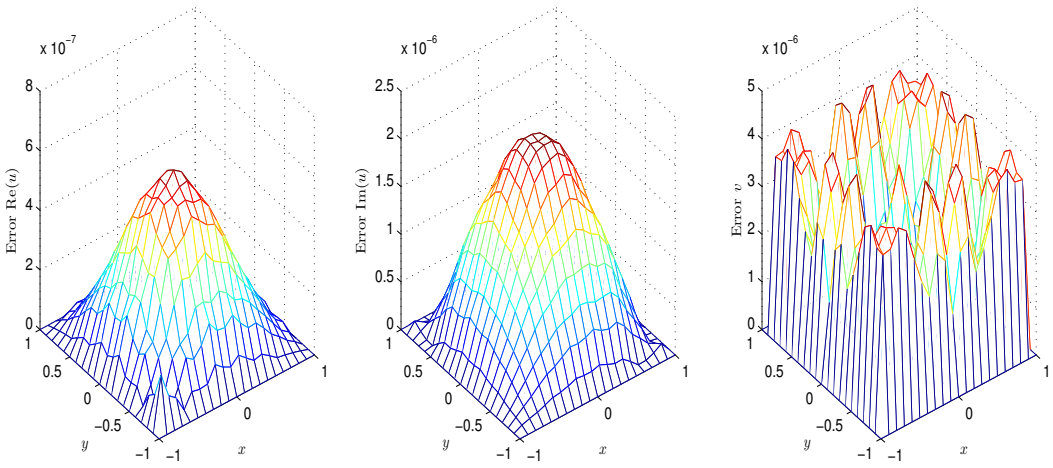

Figure 6: Graphs of the absolute errors corresponding to Fig. 5.

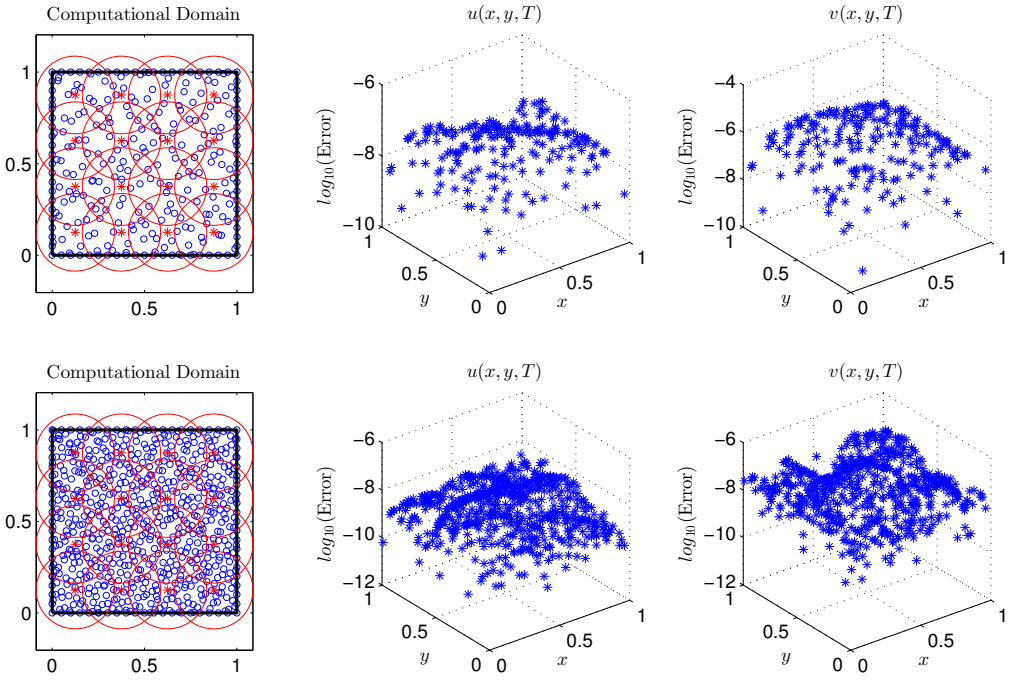

Figure 7: Partitioning of the square domain using $M = 16$ circular patches and the corresponding absolute errors of $u(x, y, T)$ and $v(x, y, T)$ for Example 2. Above subfigures use $N = 280$ nodes (80 uniform boundary and 200 Halton interior), and below subfigures use $N = 680$ nodes (80 uniform boundary and 600 Halton interior), with parameters $\delta t = 1/30$ and $T = 1$.

References

[1] Weizhu Bao and Li Yang. Efficient and accurate numerical methods for the Klein–Gordon–Schrödinger equations. *Journal of Computational Physics*, 225(2):1863–1893, 2007.

[2] Hua Liang. Linearly implicit conservative schemes for long-term numerical simulation of Klein–Gordon–Schrödinger equations. *Applied Mathematics and Computation*, 238:475–484, 2014.

[3] Isamu Fukuda and Masayoshi TSUTSUMT. On the yukawa-coupled Klein-Gordon-Schrödinger equations in three space dimensions. *Proceedings of the Japan Academy*, 51(6):402–405, 1975.

[4] AW Schreiber and R Rosenfelder. First-order variational calculation of form factor in a scalar nucleon-meson theory. *Nuclear Physics A*, 601(3):397–424, 1996.

[5] Christopher J Bardeen, Vladislav V Yakovlev, Kent R Wilson, Scott D Carpenter, Peter M Weber, and Warren S Warren. Feedback quantum control of molecular electronic population transfer. *Chemical Physics Letters*, 280(1):151–158, 1997.

[6] Mingliang Wang and Yubin Zhou. The periodic wave solutions for the Klein–Gordon–Schrödinger equations. *Physics Letters A*, 318(1):84–92, 2003.

[7] Fábio Natali and Ademir Pastor. Stability properties of periodic standing waves for the Klein-Gordon-Schrödinger system. *arXiv preprint arXiv:0907.2142*, 2009.

[8] Linghua Kong, Jingjing Zhang, Ying Cao, Yali Duan, and Hong Huang. Semi-explicit symplectic partitioned Runge–Kutta Fourier pseudo-spectral scheme for Klein–Gordon–Schrödinger equations. *Computer Physics Communications*, 181(8):1369–1377, 2010.

[9] Tohru Ozawa and Yoshio Tsutsumi. *Asymptotic behavior of solutions for the coupled Klein-Gordon-Schrödinger equations*. Kyoto University, Research Institute for Mathematical Sciences, 1991.

[10] Masahito Ohta. Stability of stationary states for the coupled Klein—Gordon—Schrödinger equations. *Nonlinear Analysis: Theory, Methods & Applications*, 27(4):455–461, 1996.

[11] Kening Lu and Bixiang Wang. Global attractors for the Klein–Gordon–Schrödinger equation in unbounded domains. *Journal of Differential Equations*, 170(2):281–316, 2001.

[12] Luming Zhang. Convergence of a conservative difference scheme for a class of Klein–Gordon–Schrödinger equations in one space dimension. *Applied Mathematics and Computation*, 163(1):343–355, 2005.

[13] Jialin Hong, Shanshan Jiang, and Chun Li. Explicit multi-symplectic methods for Klein–Gordon–Schrödinger equations. *Journal of Computational Physics*, 228(9):3517–3532, 2009.

[14] Mehdi Dehghan and Ameneh Taleei. Numerical solution of the Yukawa-coupled Klein–Gordon–Schrödinger equations via a Chebyshev pseudospectral multidomain method. *Applied Mathematical Modelling*, 36(6):2340–2349, 2012.

[15] Mehdi Dehghan and Vahid Mohammadi. Two numerical meshless techniques based on radial basis functions (RBFs) and the method of generalized moving least squares (GMLS) for simulation of coupled Klein–Gordon–Schrödinger (KGS) equations. *Computers & Mathematics with Applications*, 71(4):892–921, 2016.

[16] Shimin Guo, Can Li, Xiaoli Li, and Liquan Mei. Energy-conserving and time-stepping-varying ESAV-Hermite–Galerkin spectral scheme for nonlocal Klein–Gordon–Schrödinger system with fractional laplacian in unbounded domains. *Journal of Computational Physics*, 458:111096, 2022.

[17] Bingquan Ji and Luming Zhang. Error estimates of a conservative finite difference fourier pseudospectral method for the Klein–Gordon–Schrödinger equation. *Computers & Mathematics with Applications*, 79(7):1956–1971, 2020.

[18] Qihong Shi, Xiao-Bing Zhang, Changyou Wang, and Shu Wang. Finite time blowup for Klein–Gordon–Schrödinger system. *Mathematical Methods in the Applied Sciences*, 42(11):3929–3941, 2019.

[19] Tingchun Wang, Xiaofei Zhao, and Jiaping Jiang. Unconditional and optimal H 2-error estimates of two linear and conservative finite difference schemes for the Klein–Gordon–Schrödinger equation in high dimensions. *Advances in Computational Mathematics*, 44(2):477–503, 2018.

[20] Yanrong Zhang and Jie Shen. Efficient structure preserving schemes for the Klein–Gordon–Schrödinger equations. *Journal of Scientific Computing*, 89(2):47, 2021.

[21] Hongwei Li, Yuna Yang, and Xiangkun Li. An efficient linearly implicit and energy-conservative scheme for two dimensional Klein–Gordon–Schrödinger equations. *Numerical Methods for Partial Differential Equations*, 40(1):e23064, 2024.

[22] Elyas Shivanian and Ahmad Jafarabadi. An improved meshless method for solving two-and three-dimensional coupled Klein–Gordon–Schrödinger equations on scattered data of general-shaped domains. *Engineering with Computers*, 34(4):757–774, 2018.

[23] Donald Shepard. A two-dimensional interpolation function for irregularly-spaced data. In *Proceedings of the 1968 23rd ACM national conference*, pages 517–524, 1968.

[24] Mehdi Dehghan and Arezou Ghesmati. Numerical simulation of two-dimensional sine-Gordon solitons via a local weak meshless technique based on the radial point interpolation method (RPIM). *Computer Physics Communications*, 181(4):772–786, 2010.