Iranian Journal of Numerical Analysis and Optimization
Vol. 13, No. 1, 2023, pp 1–18
https://doi.org/10.22067/ijnao.2022.72900.1063
https://ijnao.um.ac.ir/

How to cite this article
Research Article

Applying the meshless Fragile Points method to solve the two-dimensional linear Schrödinger equation on arbitrary domains

D. Haghighi, S. Abbasbandy*and E. Shivanian

Abstract

The meshless Fragile Points method (FPM) is applied to find the numerical solutions of the Schrödinger equation on arbitrary domains. This method is based on Galerkin's weak-form formulation, and the generalized finite difference method has been used to obtain the test and trial functions. For partitioning the problem domain into subdomains, Voronoi diagram has been applied. These functions are simple, local, and discontinuous polynomials. Because of the discontinuity of test and trial functions, FPM may be inconsistent. To deal with these inconsistencies, we use numerical flux corrections. Finally, numerical results are presented for some examples of domains with different geometric shapes to demonstrate accuracy, reliability, and efficiency.

AMS subject classifications (2020): 35J10, 65M99, 65M20, 65N99.

Keywords: Fragile Points Method; Numerical Fluxes; Schrödinger equation; Voronoi Diagram.

---

*Corresponding author

Received 5 October 2021; revised 20 February 2022; accepted 6 March 2022

Donya Haghighi
Department of Applied Mathematics, Faculty of Science, Imam Khomeini International University, Qazvin, 34149-16818, Iran. e-mail: haghighi.donya@edu.ikiu.ac.ir

Saeid Abbasbandy
Department of Applied Mathematics, Faculty of Science, Imam Khomeini International University, Qazvin, 34149-16818, Iran. e-mail: abbasbandy@sci.ikiu.ac.ir, abbasbandy@yahoo.com.

Elyas Shivanian
Department of Applied Mathematics, Faculty of Science, Imam Khomeini International University, Qazvin, 34149-16818, Iran. e-mail: e_shivanian@yahoo.com

1 Introduction

Numerical methods are mainly used to solve partial differential equations and have been studied, for example, the finite element method [1], finite volume method [4], and boundary element method [13] to discretize the spatial dimension can be mentioned. In these methods, the accuracy of the method may be affected by deforming the elements or meshes. Therefore, meshless methods such as element free Galerkin [3] and meshless local Petrov–Galerkin [2] were considered. In these methods, the trial and test functions must be continuous, and usually, the trial functions in these methods are complicated. Dong et al. [6] introduced a new meshless method in which test and trial functions are considered as simple, local, and discontinuous polynomials. Very recently, this method has been used to solve the two-dimensional hyperbolic telegraph equation [8]. This new method is called the Fragile points method (FPM), which we will study in this article. This method is also used for complex and irregular domains, which are discussed in this study.

Solving the Schrödinger equation is very important in quantum dynamic calculations, and it has received a lot of attention as a model that describes several important chemical and physical phenomena [11]. This equation is derived from the vector wave equation for the electric field, which governs the propagation of electromagnetic waves in an inhomogeneous medium [10]. Schrödinger equations are also applicable in underwater acoustics, optics, and the design of optoelectronic devices [5].

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

$$-i\frac{\partial u}{\partial t}(\mathbf{x},t) = \nabla^2 u(\mathbf{x},t) + w(\mathbf{x})u(\mathbf{x},t), \qquad \mathbf{x}\in\Omega, \tag{1}$$

with initial conditions

$$u(\mathbf{x},0) = g(\mathbf{x}), \tag{2}$$

and the boundary conditions

$$u(\mathbf{x},t) = h_1(\mathbf{x},t), \quad \mathbf{x}\in\Gamma_D, \qquad \nabla u.\mathbf{n}(\mathbf{x},t) = h_2(\mathbf{x},t), \quad \mathbf{x}\in\Gamma_N. \tag{3}$$

In the above equation, $w(\mathbf{x})$ is an arbitrary potential function.

In the rest of this paper, in Section 2, the process of obtaining test and trial functions is described. In Section 3, the implementation of numerical flux corrections is given. Some numerical results and examples are provided in Section 4. Finally, the conclusions reached from using the FPM for the two-dimensional linear Schrödinger equation are expressed in Section 5.

2 Polynomial discontinuous trial and test functions

In this section, we describe the process of obtaining local, simple, discontinuous, polynomial, point-based trial and test functions. In order to divide the domain into subdomains, we consider several scattered points in the domain Ω and its boundary $\partial\Omega$. This subdivision should be such that each subdomain contains only one point. Domain partitioning can be done in several ways, for example, the Voronoi diagram partition, quadrilateral and triangular partition (in 2D), tetrahedron and hexahedron partition (in 3D), and so on. In this study, the Voronoi diagram method has been selected. In the present FPM, only nonuniform or uniform points inside and on the domain boundary are applied, and it is a meshless method.

The trial function u_h in the subdomain E_0 that includes the point P_0, can be written as

$$u_h(\mathbf{x},t) = u_0(\mathbf{x},t) + (\mathbf{x}-\mathbf{x}_0)\nabla u(\mathbf{x},t)|_{P_0}, \qquad \mathbf{x}\in E_0. \tag{4}$$

In the above equation, u_0 is the value of u_h at P_0 and $\mathbf{x}_0$ denotes the coordinate of the point P_0.

The gradient of ∇u at P_0 is yet unknown. We employ the generalized finite difference method to calculate ∇u at P_0 in terms of the values of u_h at several neighboring points of P_0. We name these neighboring points as $q_1, q_2, \ldots, q_m$. In the following, to calculate the amount of the gradient of ∇u at P_0, we minimize a weighted discrete L^2 norm $\mathbf{J}$ so that

$$\mathbf{J} = \sum_{i=0}^{m} \Big(\nabla u|_{P_0}.(\mathbf{x}_i-\mathbf{x}_0)^T - (u_i-u_0)\Big)^2 w_i, \tag{5}$$

where w_i denotes the value of weight function at q_i, $\mathbf{x}_i$ is the coordinate vector of q_i, and u_i is the value of u_h at q_i $(i=1,2,\ldots,m)$. For convenience, we assume that w is constant. Due to the stationarity of $\mathbf{J}$, we have

$$\nabla u = \left(A^T A\right)^{-1} A^T \left(\mathbf{u}_m - u_0 \mathbf{I}_m\right), \tag{6}$$

where

$$A = \begin{bmatrix} x_1-x_0 & y_1-y_0 \\ x_2-x_0 & y_2-y_0 \\ \vdots & \vdots \\ x_m-x_0 & y_m-y_0 \end{bmatrix}, \qquad \mathbf{u}_m = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_m \end{bmatrix}, \qquad \mathbf{I}_m = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}_{m\times 1}.$$

Also equation (6) can be expressed at point P_0 as follows:

$$\nabla u = \mathbf{B}\mathbf{u}_E, \tag{7}$$

where

$$\mathbf{B} = \left(A^T A\right)^{-1} A^T \begin{bmatrix} -1 & 1 & 0 & \dots & 0 \\ -1 & 0 & 1 & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ -1 & 0 & \cdots & 0 & 1 \end{bmatrix}_{m\times(m+1)}, \qquad \mathbf{u}_E = \begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ u_m \end{bmatrix}.$$

Also by substituting (7) into (4), the relation between u_h and $\mathbf{u}_E$ will be obtained as

$$u_h = \mathbf{N}\mathbf{u}_E, \quad \text{for all } \mathbf{x} \in E_0, \qquad \mathbf{N} = [\mathbf{x} - \mathbf{x}_0]\mathbf{B} + [1, 0, \dots, 0]_{1\times(m+1)}. \tag{8}$$

3 Numerical flux corrections

We can rewrite Schrödinger equation (1)–(3) using mixed form as follows:

$$\begin{cases} \sigma(\mathbf{x}, t) = \nabla u(\mathbf{x}, t), & in\ \ \Omega, \\ -\nabla.\sigma(\mathbf{x}, t) = i\dfrac{\partial u}{\partial t}(\mathbf{x}, t) + w(\mathbf{x})u(\mathbf{x}, t), & in\ \ \Omega, \\ u(\mathbf{x}, t) = h_1(\mathbf{x}, t), & in\ \ \Gamma_D, \\ \sigma.\mathbf{n}(\mathbf{x}, t) = h_2(\mathbf{x}, t), & in\ \ \Gamma_N. \end{cases} \tag{9}$$

By multiplying the first and second equations in (9) by the test functions τ and ν, respectively, and integrating it on the subdomain E,

$$\int_E \sigma_h \cdot \tau d\Omega = \int_E \nabla u_h(\mathbf{x}, t) \cdot \tau d\Omega, \tag{10}$$

$$\int_E -\nabla \cdot \sigma\nu d\Omega = i\int_E \frac{\partial u}{\partial t}(\mathbf{x}, t)\nu d\Omega + \int_E w(\mathbf{x})u(\mathbf{x}, t)\nu d\Omega, \tag{11}$$

using the Green formula and by summing these equations over all subdomains, we have

$$\int_\Omega \sigma_h.\tau d\Omega = -\int_\Omega u_h \nabla.\tau d\Omega + \sum_{E\in\Omega}\int_{\partial E} \hat{u}_h \mathbf{n}.\tau d\Gamma, \tag{12}$$

$$\int_\Omega \sigma_h.\nabla\nu d\Omega = \sum_{E\in\Omega}\int_{\partial E} \hat{\sigma}_h.\mathbf{n}\nu d\Gamma + i\int_\Omega \frac{\partial u}{\partial t}(\mathbf{x}, t)\nu d\Omega + \int_\Omega w(\mathbf{x})u(\mathbf{x}, t)\nu d\Omega. \tag{13}$$

In the above equations, values $\hat{\sigma}_h$ and $\hat{u}_h$ represent approximations σ_h and u_h on ∂E. These values are named numerical fluxes. To simplify (12) and (13), we define the operators *average* and *jump*, where by these operators,

we can manage the numerical fluxes. As regards, $\Gamma = \Gamma_h + \Gamma_D + \Gamma_N$ (Γ_h is the set of all internal boundaries of subdomains), using [6, Table 3.1], and substituting the interior penalty numerical fluxes, we have

$$\sum_{E\in\Omega}\int_E \nabla u_h.\nabla\nu d\Omega - \sum_{e\in\Gamma_h\cup\Gamma_D}\int_e \left(\{\nabla u_h\}[\nu] + \{\nabla\nu\}[u_h]\right) d\Gamma$$
$$+ \sum_{e\in\Gamma_h\cup\Gamma_D}\frac{\eta}{h_e}\int_e [\nu][u_h]\, d\Gamma = \sum_{e\in\Gamma_D}\int_e \left(\frac{\eta}{h_e}\nu - \nabla\nu.\mathbf{n}\right) h_1(\mathbf{x},t)d\Gamma$$
$$+ \int_\Omega w(\mathbf{x})u(\mathbf{x},t)\nu d\Omega + \sum_{e\in\Gamma_N}\int_e \nu h_2(\mathbf{x},t)d\Gamma$$
$$+ i\int_\Omega \frac{\partial u}{\partial t}(\mathbf{x},t)\nu d\Omega.$$

The above equation is the formula of FPM, which is called FPM-primal method.

The method (matrix form) can be expressed as follows:

$$(\mathbf{K}-\mathbf{W})\mathbf{u} - i\mathbf{C}\dot{\mathbf{u}} = \mathbf{F}. \tag{14}$$

Using θ-weighted scheme [12], the above equation can be written as follows:

$$(\mathbf{K}-\mathbf{W})(\theta\mathbf{u}^{n+1} + (1-\theta)\mathbf{u}^n) - i\mathbf{C}(\frac{\mathbf{u}^{n+1}-\mathbf{u}^n}{\Delta t}) = \mathbf{F}^n. \tag{15}$$

In (15), $\mathbf{u}^k(\mathbf{x}) = \mathbf{u}(\mathbf{x}, k\Delta t)$, where Δt is the time step and $0 \le \theta \le 1$. By substituting values $\mathbf{B}$ instead of $\nabla\nu$ and ∇u, $\mathbf{N}$ instead of u_h and ν in the formula of FPM-primal, the point stiffness matrices $\mathbf{K}$, $\mathbf{C}$, $\mathbf{W}$ and also the right vector $\mathbf{F}$ will be achieved as follows:

$$\mathbf{W} = \int_E \mathbf{N}^T\mathbf{N}w(\mathbf{x})d\Omega, \quad \mathbf{C} = \int_E \mathbf{N}^T\mathbf{N}d\Omega, \quad \mathbf{K}_E = \int_E \mathbf{B}^T\mathbf{B}\, d\Omega, \qquad E\in\Omega, \tag{16}$$

$$\mathbf{K}_h = \frac{-1}{2}\int_e (\mathbf{B}_1^T\mathbf{n}_1^T\mathbf{N}_1 + \mathbf{N}_1^T\mathbf{n}_1\mathbf{B}_1)d\Gamma + \frac{\eta}{h_e}\int_e \mathbf{N}_1^T\mathbf{N}_1 d\Gamma$$
$$+\frac{-1}{2}\int_e (\mathbf{B}_2^T\mathbf{n}_2^T\mathbf{N}_2 + \mathbf{N}_2^T\mathbf{n}_2\mathbf{B}_2)d\Gamma + \frac{\eta}{h_e}\int_e \mathbf{N}_2^T\mathbf{N}_2 d\Gamma$$
$$+\frac{-1}{2}\int_e (\mathbf{B}_2^T\mathbf{n}_1^T\mathbf{N}_1 + \mathbf{N}_2^T\mathbf{n}_2\mathbf{B}_1)d\Gamma + \frac{\eta}{h_e}\int_e \mathbf{N}_1^T\mathbf{N}_2 d\Gamma$$
$$+\frac{-1}{2}\int_e (\mathbf{B}_1^T\mathbf{n}_2^T\mathbf{N}_2 + \mathbf{N}_1^T\mathbf{n}_1\mathbf{B}_2)d\Gamma + \frac{\eta}{h_e}\int_e \mathbf{N}_2^T\mathbf{N}_1 d\Gamma, \quad e\in\partial E_1\cap\partial E_2,$$

$$\mathbf{K}_D = -\int_e (\mathbf{B}^T\mathbf{n}^T\mathbf{N} + \mathbf{N}^T\mathbf{n}\mathbf{B})d\Gamma + \frac{\eta}{h_e}\int_e \mathbf{N}^T\mathbf{N}d\Gamma, \qquad e \in \Gamma_D, \quad (17)$$

and it can also be written

$$\begin{aligned} \mathbf{F}_N &= \int_e \mathbf{N}^T h_2(\mathbf{x},t)d\Gamma, && e \in \Gamma_N, \\ \mathbf{F}_D &= \int_e (\frac{\eta}{h_e}\mathbf{N}^T - \mathbf{B}^T\mathbf{n})h_1(\mathbf{x},t)d\Gamma, && e \in \Gamma_D. \end{aligned} \quad (18)$$

4 Numerical results

In this section, we will study some numerical examples, and using the results obtained from the application of FPM on these examples, the accuracy and stability of the method are investigated. All examples were done in MATLAB software on a Core i5, 2.67 GHz CPU machine with 4 GB of memory. The relative errors used in this section are defined as follows:

$$r_0 = \frac{\|u_h - u\|_{L^2}}{\|u\|_{L^2}}, \qquad r_1 = \frac{\|\nabla u_h - \nabla u\|_{L^2}}{\|\nabla u\|_{L^2}}.$$

Also we calculate the convergence orders in space and time via

$$C - order(space) = \frac{\log 10\left(\frac{e_1}{e_2}\right)}{\log 10\left(\frac{h_1}{h_2}\right)}, \quad C - order(time) = \frac{\log 10\left(\frac{e_1}{e_2}\right)}{\log 10\left(\frac{\Delta t_1}{\Delta t_2}\right)},$$

such that h_1 and Δt_1 correspond to e_1 and also h_2 and Δt_2 correspond to error e_2. In numerical examples, we consider errors e_1 and e_2 as relative error r_0.

Example 1. (a) We first consider (1) with potential function $w(x,y) = 1 - \frac{2}{x^2} - \frac{2}{y^2}$ and exact solution $u(x,y,t) = e^{it}x^2y^2$ in the region $\Omega = [0,1]\times[0,1]$. Boundary and initial conditions are obtained using the exact solution [7, 11]. In Table 1 relative errors are shown for the number of different points of the domain that are uniformly distributed and all boundary conditions are considered Dirichlet. This table shows the good accuracy and stability of the method, and as the number of points increases, the accuracy of the method improves. In addition, as shown in Figure 1, the relative errors decrease as the number of points increases.

(b) Next, we consider the boundary conditions as follows:

$$u(0,y,t) = u(x,0,t) = 0, \quad \nabla u.n(1,y,t) = 2e^{it}y^2, \quad u(x,1,t) = e^{it}x^2.$$

Table 1: Relative errors of the method for example 1(a) at $T = 1$ and $\Delta t = 0.05$ with $\theta = 0.6$ and uniform points.

h	Number of point	parameters	Errors	CPU time (s)	C-order
0.25	$N = 25$	$h_e = 0.1$	3.942832×10^{-2}	0.31	-
		$\eta = 2.5$	2.523913×10^{-1}		
0.1	$N = 121$	$h_e = 0.1$	5.73321×10^{-3}	0.63	2.1044
		$\eta = 4$	3.52258×10^{-2}		
0.04	$N = 676$	$h_e = 0.1$	9.91802×10^{-4}	9.34	1.9148
		$\eta = 9$	9.26574×10^{-3}		
0.02	$N = 2601$	$h_e = 0.1$	4.30894×10^{-4}	$163s$	1.2027
		$\eta = 9$	4.27138×10^{-3}		

In Table 2, relative errors have been reported for the number of different points that are uniformly and nonuniformly distributed over the domain. Comparing the results, we find that in this example, how the points are distributed does not have much effect on the accuracy of the method. Also Figure 2 shows error plots for $N = 676$ uniform and nonuniform points. Figures 3 and 4 demonstrate the appropriate accuracy of the FPM for different times for $x = 0.6$, $\theta = 0.51$, $\Delta t = 0.01$, $N = 121$,$h_e = 0.1$, and $\eta = 4$. Therefore, according to these results, the method has appropriate accuracy for different boundary conditions and is also convergent. In addition, as the final time increases, accuracy is maintained and FPM is stable.

Compared to [15] and [16], the proposed method achieves almost the same accuracy in much less time.

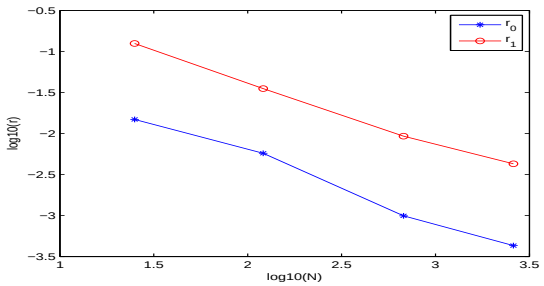

Figure 1: Relative errors for Example 1 at $T = 1$ and $\Delta t = 0.05$.

Table 2: Relative errors of the method for Example 1(b) at $T = 1$, $\theta = 0.51$ and $\Delta t = 0.01$ for points with uniform and nonuniform distribution.

uniform points

h	Number of points	Errors	CPU time (s)	C-order
0.25	$N = 25$	6.476626×10^{-2} 2.636926×10^{-1}	0.40	-
0.1	$N = 121$	5.763461×10^{-3} 3.419285×10^{-2}	1.08	2.1864
0.04	$N = 676$	2.552031×10^{-3} 1.216389×10^{-2}	27.9	0.8891

nonuniform points

Number of points	Errors	CPU time (s)
$N = 25$	4.273177×10^{-2} 1.484748×10^{-1}	0.33
$N = 121$	9.238729×10^{-3} 9.161839×10^{-2}	0.94
$N = 676$	2.423225×10^{-3} 1.216389×10^{-2}	27.7

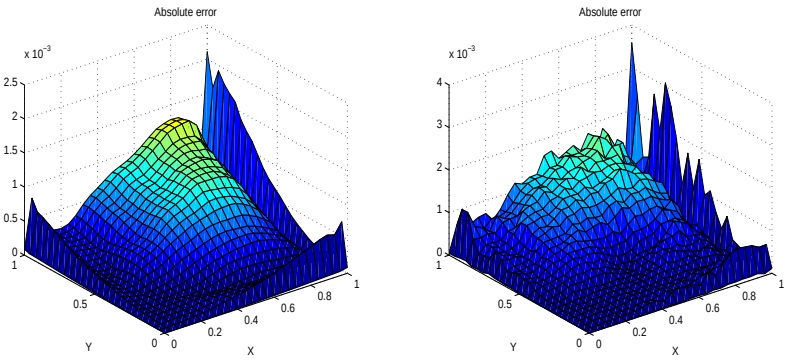

Figure 2: Comparison the absolute errors related to Example 1(b) for $T = 1$, $\Delta t = 0.01$, $\theta = 0.51$, and $N = 676$ for uniform and nonuniform points.

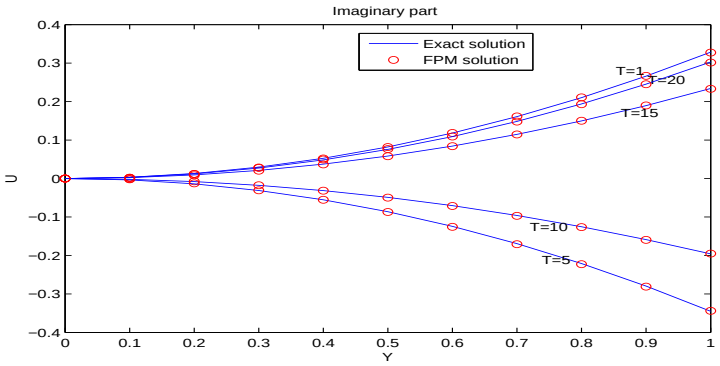

Figure 3: Comparison the imaginary parts of exact and numerical solutions related to Example 1 for $x = 0.6$, $\Delta t = 0.01$, and $N = 121$.

Example 2. In this example, we consider (1) with $N = 676$ uniform points in the domain $\Omega = [0, 1] \times [0, 1]$ such that

$$w(x, y) = -\frac{4x^2 + 4y^2 - 4x - 4y + \beta^2 - 4\beta + 2}{\beta^2},$$

$$u(x, y, t) = \exp(-\frac{(x - 0.5)^2}{\beta} - \frac{(y - 0.5)^2}{\beta} - it).$$

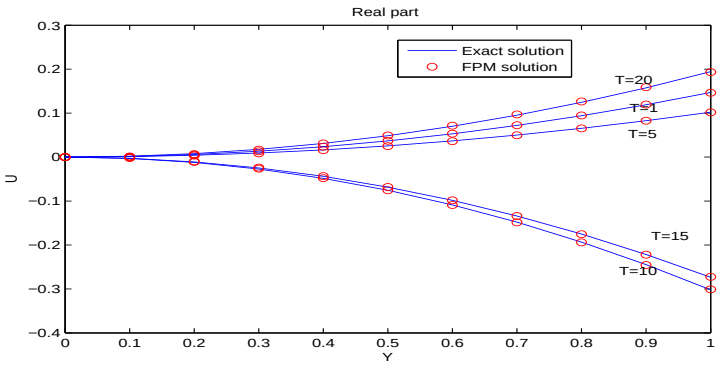

Figure 4: Comparison the real parts of exact and numerical solutions related to Example 1 for $x = 0.6$, $\Delta t = 0.01$ and $N = 121$.

Table 3: Relative errors of the method for Example 2 with $h_e = 0.1$, $\theta = 0.52$, $\eta = 11$, and $\Delta t = 0.01$.

Final time	r_0	r_1	CPU time (s)
$T = 1$	1.85272×10^{-4}	1.18535×10^{-2}	28.3
$T = 5$	3.84086×10^{-4}	2.00094×10^{-2}	99.5
$T = 10$	5.48035×10^{-4}	2.10124×10^{-2}	201.3
$T = 15$	5.00376×10^{-4}	2.07114×10^{-2}	281.5

With Dirichlet boundary conditions for $\Delta t = 0.01$, $\theta = 0.52$, and $\beta = 2$, relative errors related to different final times are shown in Table 3. This table shows the stability of the method over time. In the following, Figures 5 and 6 show the plots of imaginary and real parts of numerical and exact solutions for $N = 2601$ uniform points with $h_e = 0.1$ and $\eta = 11$. These figures indicate that the method is also accurate for a large number of points. Also, the plot of errors for $N = 676$ uniform and nonuniform points is provided in Figure 7. As this figure shows, under similar conditions, the error of the proposed method is less for points with a uniform distribution.

Example 3. a) Now we solve the previous example in an L-shaped domain that has the following boundaries:

$$\Omega_1 = \{0\} \times [0, 1], \quad \Omega_2 = \{0.36\} \times [0.52, 1],$$

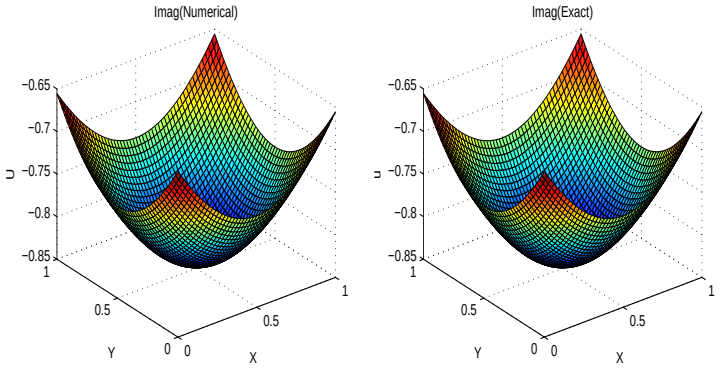

Figure 5: Comparison of imaginary parts of the numerical and exact solutions for Example 2 with $\eta = 11$, $h_e = 0.1$, $\theta = 0.52$, $\Delta t = 0.01$, $N = 2601$, and $T = 1$.

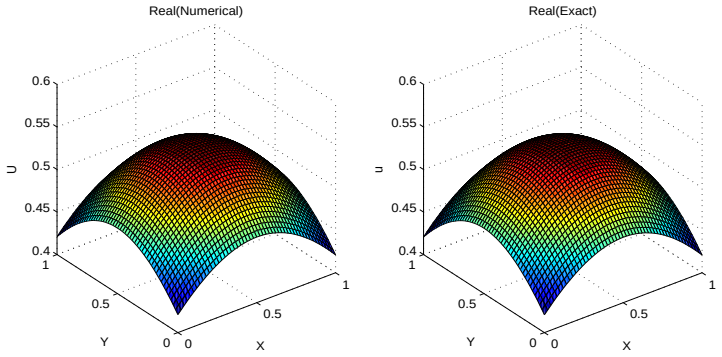

Figure 6: Comparison of real parts of the numerical and exact solutions for Example 2 with $\eta = 11$, $h_e = 0.1$, $\theta = 0.52$, $\Delta t = 0.01$, $N = 2601$, and $T = 1$.

$$\Omega_3 = [0,1] \times \{0\}\,, \quad \Omega_4 = [0.36,1] \times \{0.52\}\,.$$

Figure 8 shows the uniform distribution of $N = 484$ points in this domain. If we consider the boundary conditions completely Dirichlet, then we have Table 4 for the number of different points of the domain that are uniformly selected. As this table shows, according to the number of points, the numerical results have good accuracy that is obtained in a short time. Figure 9 shows the relation between the distance between points with uniform distributions with relative errors.

b) Next, we consider a circular domain as follows:

$$\Omega = \left\{(x,y) \in \mathbb{R}^2 : \sqrt{x^2+y^2} \leq 1\right\},$$

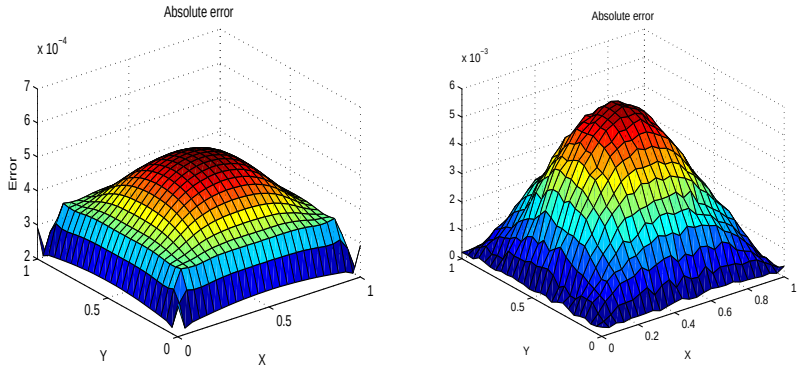

Figure 7: Plot of error for Example 2 based on uniform points for $\theta = 0.52$ $N = 676$, $T = 1$, and $\Delta t = 0.01$.

and the equation is solved by FPM. Table 5 shows the relative error values and convergence orders over time. According to this table, there is no need to reduce the time step too much, because making it smaller does not have much effect on accuracy, and we should improve the accuracy by changing other parameters or the number of points. Also, due to the circular amplitude and the number of points used, the relative errors obtained are acceptable and are obtained in a short time.

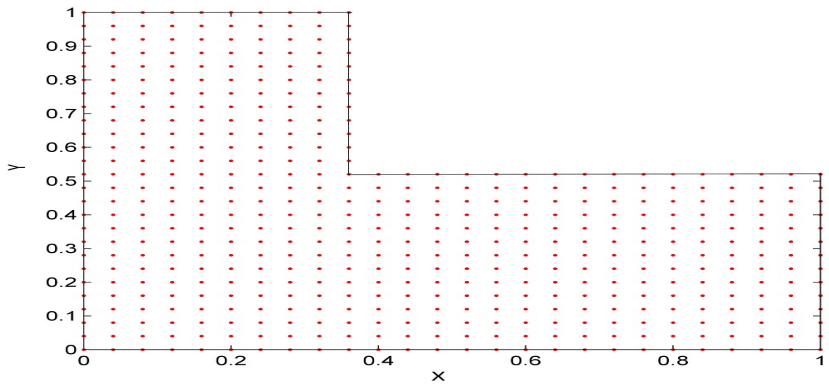

Figure 8: L-shaped domain related to Example 3(a) with $N = 484$ uniform points.

Example 4. In this example, we consider the time-dependent Schrödinger (1)–(3) in $(x, y) \in \Omega = [0, \pi] \times [0, \pi]$ with initial condition $u(x, y, 0) =$

Table 4: Relative errors of the method for Example 3(a) at $T = 1$ and $\Delta t = 0.01$, $\theta = 0.65$.

Number of points	parameters	Errors	CPU time(s)
$N = 49$	$h_e = 1$	$r_0 = 4.796603 \times 10^{-3}$	0.69
	$\eta = 75$	$r_1 = 1.608921 \times 10^{-1}$	
$N = 121$	$h_e = 0.001$	$r_0 = 3.516616 \times 10^{-3}$	1.20
	$\eta = 190$	$r_1 = 1.044712 \times 10^{-1}$	
$N = 484$	$h_e = 0.1$	$r_0 = 3.649402 \times 10^{-4}$	12.60
	$\eta = 75$	$r_1 = 2.631145 \times 10^{-2}$	
$N = 1849$	$h_e = 0.01$	$r_0 = 3.196693 \times 10^{-4}$	280.72
	$\eta = 27$	$r_1 = 1.131135 \times 10^{-2}$	

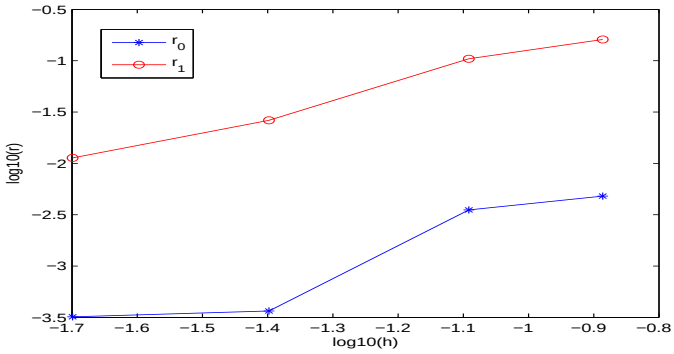

Figure 9: Error curves with respect to the distance of selected points from the domain to each other for example 3(a) with $\Delta t = 0.01$ and $T = 1s$.

Table 5: Relative error values and convergence orders over time for example 3(b) with $N = 529$ uniform points, $T = 1$, $\theta = 0.65$, $h_e = 0.01$, and $\eta = 16$.

Time step	r_0	r_1	CPU time(s)	C-order
$\Delta t = 0.08$	1.261254×10^{-2}	6.019296×10^{-2}	6.88	-
$\Delta t = 0.04$	7.115889×10^{-3}	4.330438×10^{-2}	9.04	0.8257
$\Delta t = 0.02$	3.882812×10^{-3}	3.440278×10^{-2}	13.67	0.8739
$\Delta t = 0.01$	3.042351×10^{-3}	3.105425×10^{-2}	23.26	0.3519

$\sin(x)\sin(y)$ and Dirichlet boundary conditions that are zero on all sides, with the given potential function as $w(x, y) = 3, (x, y) \in \Omega$. The analytical solution is as $u(x, y, t) = e^{it}\sin(x)\sin(y)$.

As you can see in Table 6, for a number of different points, the method has the appropriate accuracy to solve this example. Also, Figures 10 and 11 demonstrate the plots for $h_e = 0, 1$, $\eta = 3.8$, and $N = 676$ nonuniform points for $T = 1s$ and $T = 3s$, respectively. These figures show the accuracy of FPM for the case where the points are considered nonuniform. Compared to [9], the proposed method reports better computational times and accuracy.

Table 6: Relative errors of the method for Example 4 at $T = 1$, $\theta = 0.52$, and $\Delta t = 0.01$.

h	Number of points	parameters	Errors	CPU time (s)	C-order
0.25	$N = 25$	$h_e = 0.1$	1.797325×10^{-2}	0.41	-
		$\eta = 1$	1.185351×10^{-1}		
0.1	$N = 121$	$h_e = 0.1$	1.89251×10^{-3}	0.97	2.4566
		$\eta = 2.7$	1.92942×10^{-2}		
0.04	$N = 676$	$h_e = 0.1$	6.002710×10^{-4}	9.47	1.2532
		$\eta = 3.8$	4.150509×10^{-3}		
0.02	$N = 2601$	$h_e = 0.1$	4.561287×10^{-4}	107	0.3962
		$\eta = 2$	6.729947×10^{-3}		

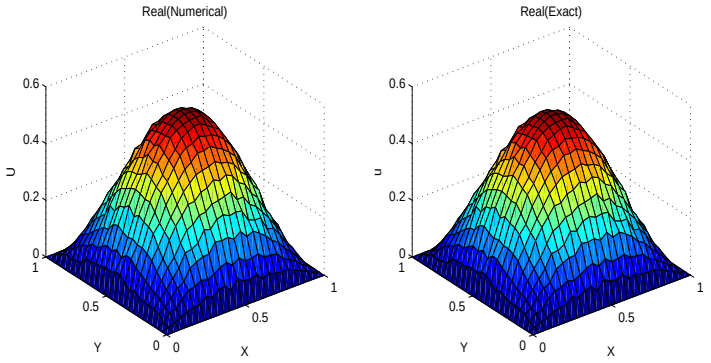

Figure 10: Plots for real parts of the numerical and exact solutions related to Example 4 for $dt = 0.01$, $N = 676$, $T = 1$, $\theta = 0.54$, $\eta = 38$, and $h_e = 1$

Example 5. For the last example, we consider Schrödinger equation with the following exact solution and initial conditions:

$$u(x, y, t) = e^{(-it)} \left(\sin(x) + \cos(y)\right), \quad u(x, y, 0) = \left(\sin(x) + \cos(y)\right).$$

This equation is solved using the proposed method on the connected amoeba-like domain according to Figure 12 with $N = 100$ points that are nonuniformly distributed in the domain. Table 7 shows relative errors and CPU

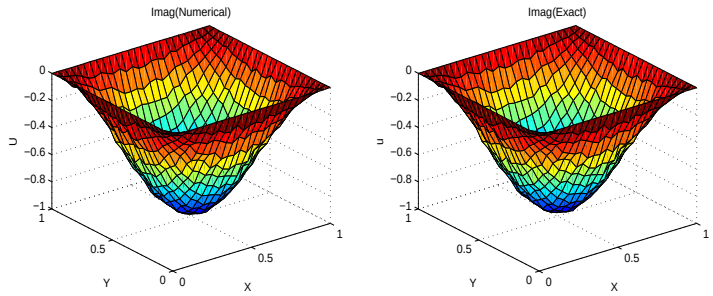

Figure 11: Plots for imaginary parts of the numerical and exact solutions related to Example 4 for $\Delta t = 0.01$, $N = 676$, $T = 5s$, $\theta = 0.54$, $\eta = 38$, and $h_e = 1$.

time related to different final times. Due to the nonuniform points and the domain of the problem in this example, the accuracy of the method is acceptable. Also, plots related to numerical and exact solutions for $T = 2s$ are presented in Figure 13, which confirms the suitability of the method for irregular domains.

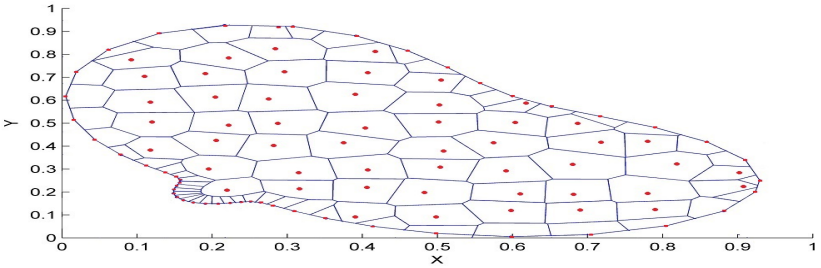

Figure 12: Domain of the problem in Example 5 with $N = 100$ selected points that are nonuniformly distributed.

Table 7: Relative errors of the method for Example 5 for $T = 1, 5, 10, 15, 20$, $\Delta t = 0.0097$, and $\theta = 0.5$.

T	Parameters	r_0	r_1	CPU time (s)
1	$h_e = 0.001$, $\eta = 215$	5.697623×10^{-3}	9.610123×10^{-2}	1.06
5	$h_e = 0.001$, $\eta = 600$	6.276758×10^{-3}	9.613064×10^{-2}	3.22
10	$h_e = 0.001$, $\eta = 500$	6.286391×10^{-3}	9.620433×10^{-2}	6.05
15	$h_e = 0.001$, $\eta = 550$	6.282664×10^{-3}	9.618269×10^{-2}	8.69
20	$h_e = 0.001$, $\eta = 600$	6.279557×10^{-3}	9.616465×10^{-2}	11.33

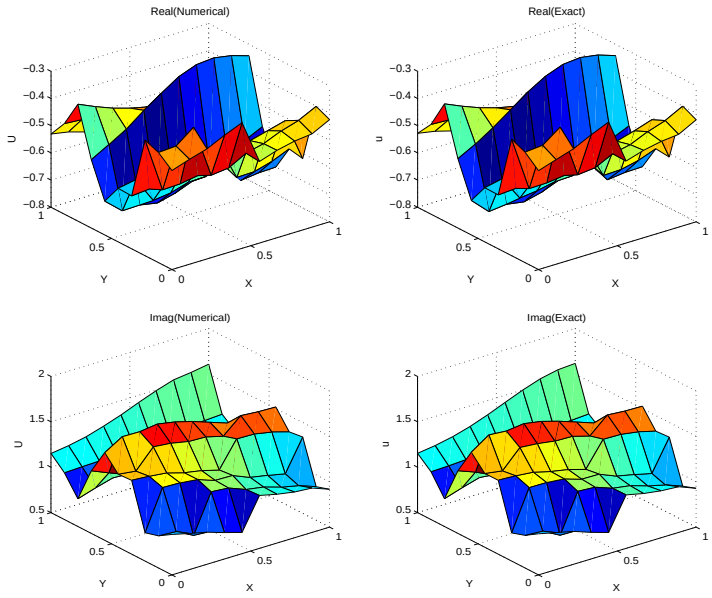

Figure 13: Plots related to numerical and exact solutions for Example 5 with $\theta = 0.5$, $h_e = 0.001$, $\eta = 50$,$T = 2s$ and $\Delta t = 0.0097$.

5 Conclusion

In this paper, the meshless Fragile Points Method (FPM) is used to obtain numerical solutions to the two-dimensional linear Schrödinger equation. This method is based on Galerkin's weak form, and the test and trial functions are considered simple, local, and discontinuous polynomials. Numerical flux corrections have been used to deal with inconsistencies due to the discontinuity of trial functions. Finally, the efficiency, stability, and accuracy of the method were evaluated with several numerical examples. In these numerical examples, the accuracy and stability of the method were evaluated both for a large number of points and for the case where the points were selected nonuniformly. We also got good solutions for larger final times and the problems with irregular domains.

According to the results of the tables and comparison of the curves obtained by FPM with the exact curves, it can be seen that the method is stable and has good accuracy. Also, the method does not have much computational cost and depending on the number of points used, it will achieve numerical solutions with good accuracy in a short time, which is an advantage over the finite element. Other advantages of this method over other numerical methods are described in detail in [14, Table 1].

Acknowledgements

Authors are grateful to there anonymous referees and editor for their constructive comments.

References

[1] Asadzadeh, M. *An introduction to the finite element method (FEM) for differential equations*, Chalmers: Lecture notes. 2010.

[2] Atluri, S.N. and Zhu, T. *A new meshless local Petrov-Galerkin (MLPG) approach in computational mechanics*, Comput. Mech. 22(2) (1998), 117–127.

[3] Belytschko, T., Lu, Y.Y. and Gu, L. *Element free Galerkin methods,* Int. J. Numer. Methods. Eng. 37(2) (1994), 229–256.

[4] Chai, J.C., Lee, H.S. and Patankar, S.V. *Finite volume method for radiation heat transfer,* J. Thermophys. Heat. Trans. 8(3) (1994), 419–425.

[5] Dehghan, M. and Emami-Naeini, F., *The Sinc-collocation and Sinc–Galerkin methods for solving the two-dimensional Schrödinger equation with nonhomogeneous boundary conditions,* Appl. Math. Mode. 37(22) 2013, 9379–9397.

[6] Dong, L., Yang, T., Wang, K. and Atluri, S.N. *A new fragile points method (FPM) in computational mechanics, based on the concepts of Point Stiffnesses and Numerical Flux Corrections,* Eng. Anal. Bound. Elem. 107 (2019), 124–133.

[7] Gao, Z., Xie, S. *Fourth-order alternating direction implicit compact finite difference schemes for two-dimensional Schrödinger equations,* Appl. Numer. Math. 61(4) (2011), 593–614.

[8] Haghighi, D., Abbasbandy, S., Shivanian, E., Dong, L. and Atluri, S.N. *The fragile points method (FPM) to solve two-dimensional hyperbolic telegraph equation using point stiffness matrices,* Eng. Anal. Bound. Elem., 134 (2022), 11–21.

[9] Karabaş, N. İ., Korkut, S. Ö., Tanoğlu, G. and Aziz, I. *An efficient approach for solving nonlinear multidimensional Schrödinger equations,* Eng. Anal. Bound. Elem, 132 (2021), 263–270.

[10] Levy, M. *Parabolic equation methods for electromagnetic wave propagation,* IEE Electromagnetic Waves Series, 45. Institution of Electrical Engineers (IEE), London, 2000.

[11] Subaşi, M. *On the finite differences schemes for the numerical solution of two dimensional Schrödinger equation,* Numer. Methods Partial Differential Equations 18(6) (2002), 752–758.

[12] Tian, Z.F. and Yu, P.X. *High-order compact ADI (HOC-ADI) method for solving unsteady 2D Schrödinger equation,* Comput. Phys. Commun. 181(5) (2010), 861–868.

[13] Wrobel, L.C. *The Boundary Element Method, Volume 1: Applications in Thermo-Fluids and Acoustics,* Vol. 1. John Wiley & Sons. 2002.

[14] Yang, T., Dong, L. and Atluri, S.N. *A simple Galerkin meshless method, the fragile points method using point stiffness matrices, for 2D linear elastic problems in complex domains with crack and rupture propagation,* Int. J. Numer. Meth. Eng. 122(2) (2021), 348–385.

[15] Zhang, L.W., Deng, Y.J., Liew, K.M. and Cheng, Y.M. *The improved complex variable element-free Galerkin method for two-dimensional Schrödinger equation,* Comput. Math. with Appl. 68(10) (2014), 1093–1106.

[16] Zhang, S. and Chen, S. *A meshless symplectic method for two-dimensional Schrödinger equation with radial basis functions,* Comput. Math. with Appl. 72(9) (2016), 2143–2150.

How to cite this article
Haghighi, D., Abbasbandy, S. and Shivanian, E., Applying the meshless Fragile Points method to solve the two-dimensional linear Schrödinger equation on arbitrary domains. *Iran. j. numer. anal. optim.*, 2023; 13(1): 1-18. https://doi.org/10.22067/ijnao.2022.72900.1063.