# Solving unsteady Schrödinger equation using the improved element-free Galerkin method*

Rong-Jun Cheng(程荣军)[1] and Yu-Min Cheng(程玉民)[2,†]

[1] *Faculty of Maritime and Transportation, Ningbo University, Ningbo* 315211, *China*
[2] *Shanghai Institute of Applied Mathematics and Mechanics, Shanghai University, Shanghai* 200072, *China*

(Received 9 July 2015; revised manuscript received 1 November 2015; published online 10 January 2016)

By employing the improved moving least-square (IMLS) approximation, the improved element-free Galerkin (IEFG) method is presented for the unsteady Schrödinger equation. In the IEFG method, the two-dimensional (2D) trial function is approximated by the IMLS approximation, the variation method is used to obtain the discrete equations, and the essential boundary conditions are imposed by the penalty method. Because the number of coefficients in the IMLS approximation is less than in the moving least-square (MLS) approximation, fewer nodes are needed in the entire domain when the IMLS approximation is used than when the MLS approximation is adopted. Then the IEFG method has high computational efficiency and accuracy. Several numerical examples are given to verify the accuracy and efficiency of the IEFG method in this paper.

**Keywords:** meshless method, improved moving least-square (IMLS) approximation, improved element-free Galerkin (IEFG) method, Schrödinger equation

**PACS:** 02.60.Lj, 03.65.Ge **DOI:** 10.1088/1674-1056/25/2/020203

## 1. Introduction

As a fundamental model to describe the nonlinear phenomenon of nature, the nonlinear Schrödinger (NLS) equation is widely used in kinds of fields, such as nonlinear optical medium,[1–3] photonics,[4] plasmas,[5] Bose–Einstein condensates,[6,7] quantum mechanics,[8] electromagnetic wave propagation,[9] underwater acoustics,[10–12] etc.

The Schrödinger equation has been widely explored and investigated in the last few years. The methods of numerical modeling for Schrödinger equation have been put forward by finite difference schemes,[13] a Crank–Nicolson implicit scheme,[14] higher-order compact scheme,[15] radial basis functions,[16] a fully implicit formula,[17] meshless local Petrov–Galerkin (MLPG) method,[18] and the dual reciprocity boundary element method (DRBEM),[19] and so on.

As important numerical methods, meshless methods[20,21] have been used for solving many complicated problems in science and engineering. It is important to develop various meshless methods of solving linear and nonlinear problems accurately and efficiently.[22–24] The moving least-square (MLS) approximation was first used for data fitting.[25] The MLS technique has been applied to the analysis of solid mechanics and often used to construct shape function in the element-free Galerkin (EFG) method.[20] The EFG method is one of the most important meshless methods that have attracted much attention of researchers.[20] This method does not require any element connectivity data, and does not suffer much degradation in accuracy when nodal arrangements are very irregular. The method has been successfully applied to a lot of science and engineering problems, such as elasticity, heat conduction, and crack growth. The MLS technique is derived from the traditional least squares method, which means that the least squares method is used at each computing point in the actual process of calculation procedure. Unfortunately, the algebraic equation system in the MLS method is sometimes ill-conditioned. In order to solve this problem, Cheng and Chen presented the improved moving least-square (IMLS) approximation.[26] In the IMLS approximation, the basis function is chosen as an orthogonal function system, and the resulting final algebraic equation system is not ill-conditioned. Based on the IMLS approximation, the boundary element-free method is used for studying the elasticity, fracture, elastodynamics and potential problems.[27–30] The improved element-free Galerkin (IEFG) method is a combination of IMLS approximation and EFG method. In the IEFG method,[31–34] fewer nodes are selected in the entire domain than in the conventional EFG method. Hence, the IEFG method will definitely increase the computational efficiency. Furthermore, the IEFG method has greater computational accuracy with the same nodes than the EFG method. None of the shape functions of the MLS approximation and the IMLS approximation satisfies the property of Kronecker $\delta$ function. In order to overcome the disadvantage, Ren *et al.* presented

*Project supported by the National Natural Science Foundation of China (Grant No. 11171208), the Natural Science Foundation of Zhejiang Province, China (Grant No. LY15A020007), the Natural Science Foundation of Ningbo City (Grant No. 2014A610028), and the K. C. Wong Magna Fund in Ningbo University, China.

†Corresponding author. E-mail: ymcheng@shu.edu.cn

© 2016 Chinese Physical Society and IOP Publishing Ltd 

Lancaster's improved IMLS method.[35] Based on the new shape function of IMLS method, Ren and Cheng presented interpolating element-free Galerkin (IEFG) method and interpolating boundary element-free method for potential and elasticity problems,[35–38] and the corresponding mathematical theory was discussed.[39] Cheng and Liew presented the IMLS Ritz and Kp Ritz method to solve heat conduction problem, wave problem and Sine-Gordon equation.[40–43] Wang *et al.* put forward the interpolating element-free method and the interpolating boundary element-free method with nonsingular weight function.[44–46] Based on the complex variable moving least-square (CVMLS) approximation and the complex variable reproducing kernel particle method (CVRKPM), the meshless method with complex variable has been presented. Chen *et al.* presented the complex variable reproducing kernel particle method (CVRKPM) for two-dimensional (2D) elastoplasticity, elastodynamics problems and the analysis of Kirchhoff plates.[47–49] Cheng *et al.* presented the complex variable element-free Galerkin (CVEFG) method of investigating the elasticity, elastoplasticity, viscoelasticity and large deformation problems.[50–53] Based on an improved CVMLS approximation, an improved complex variable element-free Galerkin method has been developed to solve the 2D elasticity problems and elastoplasticity.[54,55] Deng *et al.* presented an interpolating complex variable element-free Galerkin method of solving the temperature field problems.[56]

The improved element-free Galerkin (IEFG) method for the 2D unsteady Schrödinger equation is developed in this paper. In the IEFG method, the trial functions are approximated by the IMLS technique, the energy functional and the final algebraic equation system that is gained via the Galerkin method, the boundary conditions are imposed by penalty function. Numerical examples show that the present technique leads to good accuracy and efficiency.

## 2. IEFG method for 2D unsteady Schrödinger equation

### 2.1. Weak form of 2D unsteady Schrödinger equation

Consider the following 2D unsteady Schrödinger equation:

$$-\mathrm{i}\frac{\partial u(x,y,t)}{\partial t} = \Delta u(x,y,t) + f(x,y)u(x,y,t), \quad (x,y)\in\Omega,\ 0\le t\le T, \tag{1a}$$

where $u(x,y,t)$ is the wave function and $f(x,y)$ is the potential function. The Schrödinger equation has the following initial and boundary conditions:

$$u(x,y,0) = \varphi_1(x,y),\ (x,y)\in\Omega, \tag{1b}$$

$$u(x,y,t) = \varphi_2(x,y),\ (x,y)\in\Gamma_1, \tag{1c}$$

$$\frac{\partial u(x,y,t)}{\partial n} = 0,\quad (x,y)\in\Gamma_2, \tag{1d}$$

where $\mathrm{i}=\sqrt{-1}$, $T$ is the time, $\varphi_1(x,y)$ and $\varphi_2(x,y)$ are the known functions, $\Omega=\{(x,y)|a\le x\le b, c\le y\le d\}$, $\Gamma=\Gamma_1\cup\Gamma_2$ is the boundary of $\Omega$, and $n$ is the outward normal direction of the boundary $\Gamma_2$.

By taking variation and the subsection integration of Eq. (1a), the weak form of the unsteady Schrödinger equation is obtained as

$$\int_\Omega \left[\delta(\nabla u)^{\mathrm{T}}\nabla u - \delta u^{\mathrm{T}} f u - \mathrm{i}\delta u^{\mathrm{T}}\frac{\partial u}{\partial t}\right]\mathrm{d}\Omega - \int_{\Gamma_u}\delta u^{\mathrm{T}}\frac{\partial u}{\partial n}\mathrm{d}\Gamma + \frac{\alpha}{2}\delta\int_{\Gamma_u}(u-\bar{u})^{\mathrm{T}}(u-\bar{u})\mathrm{d}\Gamma = 0, \tag{2}$$

where $\alpha$ is the penalty factor.

### 2.2. IMLS shape function

In the IMLS approximation, we define the following local approximation:

$$u^h(\boldsymbol{x},\bar{\boldsymbol{x}}) = \sum_{j=1}^{m} p_j(\bar{\boldsymbol{x}})a_j(\boldsymbol{x}) \equiv \boldsymbol{p}^{\mathrm{T}}(\bar{\boldsymbol{x}})\boldsymbol{a}(\boldsymbol{x}). \tag{3}$$

Define a functional as

$$J = \sum_{I=1}^{n} w(x-x_I)[u^h(x,x_I)-u(x_I)]^2, \tag{4}$$

where $w(x-x_I)$ are weight functions, $x_I$ are the nodes in influenced domain of $x$. Equation (4) can be rewritten in the vector form as

$$J = (\boldsymbol{p}\boldsymbol{a}-\boldsymbol{u})^{\mathrm{T}}\boldsymbol{W}(\boldsymbol{x})(\boldsymbol{p}\boldsymbol{a}-\boldsymbol{u}), \tag{5}$$

where

$$u^{\mathrm{T}} = (u_1,u_2,\ldots,u_n), \tag{6}$$

$$\boldsymbol{p} = \begin{bmatrix} p_1(\boldsymbol{x}_1) & p_2(\boldsymbol{x}_1) & \cdots & p_m(\boldsymbol{x}_1) \\ p_1(\boldsymbol{x}_2) & p_2(\boldsymbol{x}_2) & \cdots & p_m(\boldsymbol{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ p_1(\boldsymbol{x}_n) & p_2(\boldsymbol{x}_n) & \cdots & p_m(\boldsymbol{x}_n) \end{bmatrix}, \tag{7}$$

and

$$\boldsymbol{W}(\boldsymbol{x}) = \begin{bmatrix} w(\boldsymbol{x}-\boldsymbol{x}_1) & 0 & \cdots & 0 \\ 0 & w(\boldsymbol{x}-\boldsymbol{x}_2) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & w(\boldsymbol{x}-\boldsymbol{x}_n) \end{bmatrix}. \tag{8}$$

By solving the extremum of $J$, we can find the coefficients $\boldsymbol{a}(\boldsymbol{x})$ as follows:

$$\frac{\partial J}{\partial \boldsymbol{a}} = \boldsymbol{A}(\boldsymbol{x})\boldsymbol{a}(\boldsymbol{x}) - \boldsymbol{B}(\boldsymbol{x})\boldsymbol{u} = 0, \tag{9}$$

$$\boldsymbol{A}(\boldsymbol{x})\boldsymbol{a}(\boldsymbol{x}) = \boldsymbol{B}(\boldsymbol{x})\boldsymbol{u}. \tag{10}$$

If $p_1(\boldsymbol{x}), p_2(\boldsymbol{x}), \ldots, p_m(\boldsymbol{x})$ satisfy the following conditions:

$$(p_k, p_j) = \sum_{i=1}^{n} w_i p_k(\boldsymbol{x}_i) p_j(\boldsymbol{x}_i)$$

$$= \begin{cases} 0, & k \neq j, \\ A_k, & k = j, \end{cases} \quad (k, j = 1, 2, \ldots, m), \tag{11}$$

then the weighted functions $p_1(\boldsymbol{x}), p_2(\boldsymbol{x}), \ldots, p_m(\boldsymbol{x})$ are called an orthogonal function set. The orthogonal function set $\boldsymbol{p} = (p_i)$ can be obtained by using Schmidt method as

$$\begin{aligned} p_1 &= 1 \\ p_i &= r^{i-1} - \sum_{k=1}^{i-1} \frac{(r^{i-1}, p_k)}{(p_k, p_k)} p_k, \quad i = 2, 3, \ldots. \end{aligned} \tag{12}$$

Equation (10) can be rewritten as

$$\begin{aligned} &\begin{bmatrix} (p_1, p_1) & 0 & \cdots & 0 \\ 0 & (p_2, p_2) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & (p_m, p_m) \end{bmatrix} \begin{bmatrix} a_1(\boldsymbol{x}) \\ a_2(\boldsymbol{x}) \\ \vdots \\ a_m(\boldsymbol{x}) \end{bmatrix} \\ &= \begin{bmatrix} (p_1, u_I) \\ (p_2, u_I) \\ \vdots \\ (p_m, u_I) \end{bmatrix}. \end{aligned} \tag{13}$$

The coefficients $a_i(\boldsymbol{x})$ can be obtained as

$$a_i(\boldsymbol{x}) = \frac{(p_i, u_I)}{(p_i, p_i)}; \quad (i = 1, 2, \ldots, m), \tag{14}$$

i.e.,

$$\boldsymbol{a}(\boldsymbol{x}) = \bar{\boldsymbol{A}}(\boldsymbol{x}) \boldsymbol{B}(\boldsymbol{x}) \boldsymbol{u}, \tag{15}$$

where

$$\bar{\boldsymbol{A}}(\boldsymbol{x}) = \begin{bmatrix} \dfrac{1}{(p_1, p_1)} & 0 & \cdots & 0 \\ 0 & \dfrac{1}{(p_2, p_2)} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \dfrac{1}{(p_m, p_m)} \end{bmatrix}. \tag{16}$$

From Eq. (15), $u^h(\boldsymbol{x})$ can be written as

$$u^h(\boldsymbol{x}) = \bar{\boldsymbol{\Phi}}(\boldsymbol{x}) \boldsymbol{u} = \sum_{I=1}^{n} \bar{\Phi}_I(\boldsymbol{x}) u_I, \tag{17}$$

where

$$\begin{aligned} \bar{\boldsymbol{\Phi}}(\boldsymbol{x}) &= (\bar{\Phi}_1(\boldsymbol{x}), \bar{\Phi}_2(\boldsymbol{x}), \ldots, \bar{\Phi}_n(\boldsymbol{x})) \\ &= \boldsymbol{p}^{\mathrm{T}}(\boldsymbol{x}) \bar{\boldsymbol{A}}(\boldsymbol{x}) \boldsymbol{B}(\boldsymbol{x}). \end{aligned} \tag{18}$$

By differentiate Eq. (18), we will obtain the derivatives of shape function $\bar{\boldsymbol{\Phi}}(\boldsymbol{x})$ as

$$\bar{\boldsymbol{\Phi}}_{I,i}(\boldsymbol{x}) = \sum_{j=1}^{m} [p_{j,i} (\bar{\boldsymbol{A}} \boldsymbol{B})_{jI} + p_j (\bar{\boldsymbol{A}}_{,i} \boldsymbol{B} + \bar{\boldsymbol{A}} \boldsymbol{B}_{,i})_{jI}]. \tag{19}$$

The cubic spline weight function is chosen as

$$w_I = w(\boldsymbol{x} - \boldsymbol{x}_I) \equiv w(r)$$

$$= \begin{cases} \dfrac{2}{3} - 4r^2 + 4r^3, & r \leq \dfrac{1}{2}, \\ \dfrac{4}{3} - 4r + 4r^2 - \dfrac{4}{3} r^3, & \dfrac{1}{2} < r \leq 1, \\ 0, & r > 1, \end{cases} \tag{20}$$

where

$$r = \frac{d_I}{d_{m_I}}, \tag{21}$$

$$d_I = \|x - x_I\|, \tag{22}$$

$$d_{m_I} = d_{\max} c_I, \tag{23}$$

and $d_{\max}$ is a scaling parameter, $c_I$ is the distance and chosen so that matrix $M(\boldsymbol{x})$ is not singular. The first and second order derivatives of the weight functions can be easily obtained by chain rule as

$$\begin{aligned} \frac{\mathrm{d}w_I}{\mathrm{d}x} &= \frac{\mathrm{d}w_I}{\mathrm{d}r} \frac{\mathrm{d}r}{\mathrm{d}x} \\ &= \begin{cases} (-8r + 12r^2) \dfrac{\mathrm{sign}(x - x_I)}{d_{m_I}}, & r \leq \dfrac{1}{2}, \\ (-4 + 8r - 4r^2) \dfrac{\mathrm{sign}(x - x_I)}{d_{m_I}}, & \dfrac{1}{2} < r \leq 1, \\ 0, & r > 1, \end{cases} \end{aligned} \tag{24}$$

$$\begin{aligned} \frac{\mathrm{d}^2 w_I}{\mathrm{d}x^2} &= \frac{\mathrm{d}}{\mathrm{d}x} \left( \frac{\mathrm{d}w_I}{\mathrm{d}x} \right) \\ &= \begin{cases} (-8 + 24r), & r \leq \dfrac{1}{2}, \\ (8 - 8r), & \dfrac{1}{2} < r \leq 1, \\ 0, & r > 1. \end{cases} \end{aligned} \tag{25}$$

### 2.3. Discretization procedure

From Eq. (17), we obtain the approximation of trial function as follows:

$$u^h(\boldsymbol{x}, t) = \sum_{I=1}^{n} \Phi_I(\boldsymbol{x}) T_I(t) = \boldsymbol{\Phi}(\boldsymbol{x}) \boldsymbol{U}, \tag{26}$$

$$\begin{aligned} \frac{\partial u(\boldsymbol{x}, t)}{\partial t} &= \frac{\partial}{\partial t} \sum_{I=1}^{n} \Phi_I(\boldsymbol{x}) T_I(t) = \sum_{I=1}^{n} \Phi_I(\boldsymbol{x}) \frac{\partial T_I(t)}{\partial t} \\ &= \boldsymbol{\Phi}(\boldsymbol{x}) \boldsymbol{U}_t, \end{aligned} \tag{27}$$

where

$$\boldsymbol{\Phi}(\boldsymbol{x}) = (\Phi_1(\boldsymbol{x}), \Phi_2(\boldsymbol{x}), \ldots, \Phi_n(\boldsymbol{x})), \tag{28}$$

$$\boldsymbol{U} = (T_1(t), T_2(t), \ldots, T_n(t))^{\mathrm{T}}, \tag{29}$$

$$\boldsymbol{U}_t = \left( \frac{\partial T_1(t)}{\partial t}, \frac{\partial T_2(t)}{\partial t}, \ldots, \frac{\partial T_n(t)}{\partial t} \right)^{\mathrm{T}}, \tag{30}$$

$$x = (x, y). \tag{31}$$

Substituting Eqs. (26)–(31) into Eq. (2), we obtain the vector form as follows:

$$\mathrm{i} \boldsymbol{C} \boldsymbol{U}_t + \boldsymbol{K} \boldsymbol{U} = \boldsymbol{Q}, \tag{32}$$

where

$$C = \int_{\Omega} \boldsymbol{\Phi}^{\mathrm{T}}(\boldsymbol{x}) \boldsymbol{\Phi}(\boldsymbol{x}) \mathrm{d}\Omega, \tag{33}$$

$$C_{IJ} = \int_{\Omega} \Phi_I^{\mathrm{T}}(\boldsymbol{x})\Phi_J(\boldsymbol{x})\mathrm{d}\Omega, \tag{34}$$

$$\boldsymbol{K} = \int_{\Omega} \boldsymbol{f}(\boldsymbol{x})\boldsymbol{\Phi}^{\mathrm{T}}(\boldsymbol{x})\boldsymbol{\Phi}(\boldsymbol{x})\mathrm{d}\Omega - \int_{\Omega} \nabla\boldsymbol{\Phi}^{\mathrm{T}}(\boldsymbol{x})\nabla\boldsymbol{\Phi}(\boldsymbol{x})\mathrm{d}\Omega + \int_{\Gamma} \boldsymbol{\Phi}^{\mathrm{T}}(\boldsymbol{x})\alpha\boldsymbol{\Phi}(\boldsymbol{x})\mathrm{d}\Gamma, \tag{35}$$

$$K_{IJ} = \int_{\Omega} f(\boldsymbol{x})\Phi_I^{\mathrm{T}}(\boldsymbol{x})\Phi_J(\boldsymbol{x})\mathrm{d}\Omega - \int_{\Omega} \nabla\Phi_I^{\mathrm{T}}(\boldsymbol{x})\nabla\Phi_J(\boldsymbol{x})\mathrm{d}\Omega + \int_{\Gamma} \Phi_I^{\mathrm{T}}(\boldsymbol{x})\alpha\Phi_J(\boldsymbol{x})\mathrm{d}\Gamma, \tag{36}$$

$$Q_I = \int_{\Gamma} \alpha\Phi_I(\boldsymbol{x})\bar{u}\mathrm{d}\Gamma. \tag{37}$$

Since $\boldsymbol{U}$ and $\boldsymbol{Q}$ are complex numbers, we let

$$\boldsymbol{U} = \boldsymbol{V} + \mathrm{i}\boldsymbol{W}, \tag{38}$$

$$\boldsymbol{Q} = \boldsymbol{R} + \mathrm{i}\boldsymbol{S}, \tag{39}$$

where $\boldsymbol{V}$ and $\boldsymbol{W}$ are the real and imaginary part of $\boldsymbol{U}$; $\boldsymbol{R}$ and $\boldsymbol{S}$ are the real and imaginary parts of $\boldsymbol{Q}$.

Substituting Eqs. (38) and (39) into Eq. (32), we obtain

$$\mathrm{i}\boldsymbol{C}(\boldsymbol{V} + \mathrm{i}\boldsymbol{W})_t + \boldsymbol{K}(\boldsymbol{V} + \mathrm{i}\boldsymbol{W}) = \boldsymbol{R} + \mathrm{i}\boldsymbol{S}. \tag{40}$$

Separate the real part from the imaginary part of Eq. (40), we derive

$$\begin{cases} -\boldsymbol{C}\boldsymbol{W}_t + \boldsymbol{K}\boldsymbol{V} = \boldsymbol{R}, \\ \boldsymbol{C}\boldsymbol{V}_t + \boldsymbol{K}\boldsymbol{W} = \boldsymbol{S}. \end{cases} \tag{41}$$

By using the center difference method, we make the discretization of Eq. (41) as

$$\begin{cases} -\boldsymbol{C}\dfrac{\boldsymbol{W}^{n+1} - \boldsymbol{W}^n}{\Delta t} + \boldsymbol{K}\dfrac{\boldsymbol{V}^{n+1} + \boldsymbol{V}^n}{2} = \dfrac{\boldsymbol{R}^{n+1} + \boldsymbol{R}^n}{2}, \\ \boldsymbol{C}\dfrac{\boldsymbol{V}^{n+1} - \boldsymbol{V}^n}{\Delta t} + \boldsymbol{K}\dfrac{\boldsymbol{W}^{n+1} + \boldsymbol{W}^n}{2} = \dfrac{\boldsymbol{S}^{n+1} + \boldsymbol{S}^n}{2}. \end{cases} \tag{42}$$

Equation (42) is equivalent to

$$\begin{bmatrix} \Delta t\boldsymbol{K} & -2\boldsymbol{C} \\ 2\boldsymbol{C} & \Delta t\boldsymbol{K} \end{bmatrix} \begin{bmatrix} \boldsymbol{V}^{n+1} \\ \boldsymbol{W}^{n+1} \end{bmatrix} = \begin{bmatrix} -\Delta t\boldsymbol{K} & -2\boldsymbol{C} \\ 2\boldsymbol{C} & -\Delta t\boldsymbol{K} \end{bmatrix} \begin{bmatrix} \boldsymbol{V}^n \\ \boldsymbol{W}^n \end{bmatrix} + \begin{bmatrix} \dfrac{\boldsymbol{R}^{n+1} + \boldsymbol{R}^n}{2} \\ \dfrac{\boldsymbol{S}^{n+1} + \boldsymbol{S}^n}{2} \end{bmatrix}. \tag{43}$$

The numerical results of 2D Schrödinger equation by the IEFG method can be obtained through solving Eq. (43).

## 3. Numerical examples

In this section, three numerical examples are demonstrated to verify the efficiency and accuracy of the IEFG method. The implementation of the IEFG method is also presented. In all numerical examples, cubic spline function is chosen as a weight function, and the linear basis function is used.

### 3.1. Example 1

Consider Eq. (1) on $\Omega = [0,1] \times [0,1]$ with

$$u(x,y,0) = \varphi_1(x,y) = \frac{\mathrm{i}}{\cosh x \cosh y}, \tag{44}$$

$$u(1,y,t) = \frac{\mathrm{i}\exp(\mathrm{i}t)}{\cosh 1 \cosh y}, \quad x = 1,\ 0 \le y \le 1, \tag{45}$$

$$u(x,1,t) = \frac{\mathrm{i}\exp(\mathrm{i}t)}{\cosh x \cosh 1}, \quad y = 1,\ 0 \le x \le 1, \tag{46}$$

$$\frac{\partial u(x,y,t)}{\partial n} = 0, \quad x = 0,\ 0 \le y \le 1, \tag{47}$$

$$\frac{\partial u(x,y,t)}{\partial n} = 0, \quad y = 0,\ 0 \le x \le 1, \tag{48}$$

and

$$f(x,y) = 3 - 2\tanh^2 x - 2\tanh^2 y. \tag{49}$$

The analytical solution is

$$u(x,y,t) = \frac{\mathrm{i}\exp(\mathrm{i}t)}{\cosh x \cosh y}. \tag{50}$$

The IEFG method is used to solve the above equation with penalty factor $\alpha = 10^5$ and $d_{\max} = 2.3$ in time steps of $\Delta t = 0.01$. Figure 1 shows the numerical and solutions for real part of $u(x,y,t)$ at $t = 0.5$, 1, 2, 3, 4, and 6, respectively. Figure 2 describes the numerical solution and exact solution for imaginary part of $u(x,y,t)$ when $t = 0.5$, 1, 2, 3, 4, and 6, respectively. In Figs. 3, 4, and 5, the surfaces of real part and imaginary part of $u(x,y,t)$ are plotted when $t = 3$, 4, and 6, respectively. Table 1 presents the relative errors for the real part and imaginary part of the solution and the CPU time at $t = 2$. As can seen from Table 1, the smaller time step is chosen, the smaller relative error is obtained. Table 2 shows the relationship between the relative error and number of nodes for the IEFG method at time $t = 1$. Tables 1 and 2 show that the error norm decreases with the increase of the number of nodes and decrease of time step.

**Table 1.** Relative errors and CPU times for the IEFG method at the time $t = 2$.

| $\Delta t$ | Real part | Imaginary part | CPU time |
|---|---|---|---|
| 0.1 | $8.7895 \times 10^{-4}$ | $7.8986 \times 10^{-4}$ | 3.5 |
| 0.01 | $3.1875 \times 10^{-4}$ | $3.2848 \times 10^{-4}$ | 7.6 |
| 0.001 | $2.1079 \times 10^{-5}$ | $1.9846 \times 10^{-5}$ | 11.4 |
| 0.0001 | $1.3195 \times 10^{-5}$ | $8.5743 \times 10^{-6}$ | 26.5 |

**Table 2.** Dependences of relative error on node number for the IEFG method at $t = 1$.

| Number of nodes | 225 | 144 | 100 | 70 |
|---|---|---|---|---|
| relative error norm | $5.7479 \times 10^{-6}$ | $3.2745 \times 10^{-5}$ | $6.3679 \times 10^{-5}$ | $1.8794 \times 10^{-4}$ |

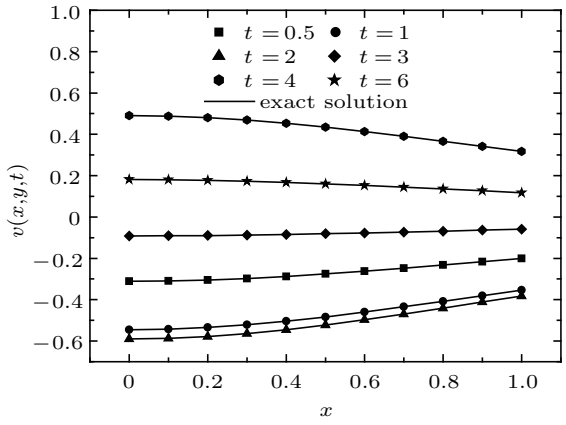



**Fig. 1.** Real parts of exact solution and numerical solution of $u(x,y,t)$.

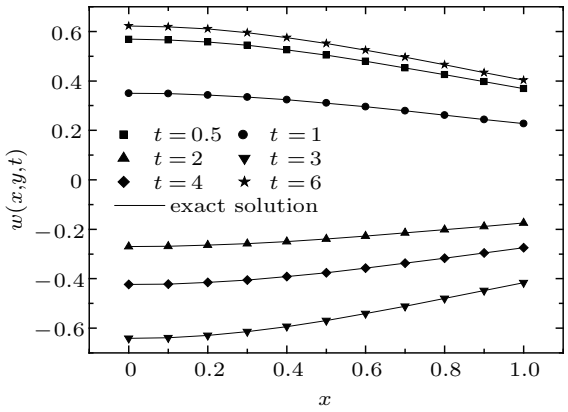



**Fig. 2.** Imaginary parts of exact solution and numerical solution of $u(x,y,t)$.

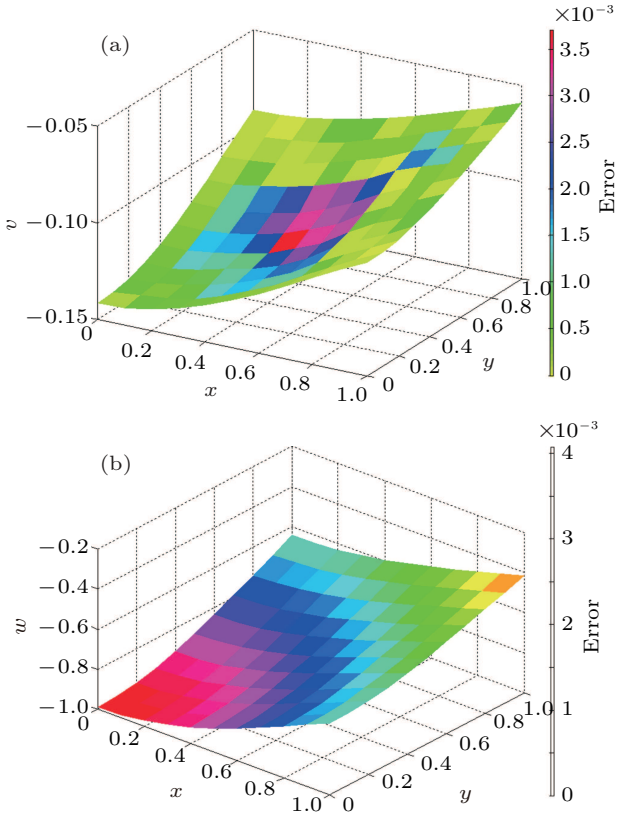



**Fig. 3.** (color online) Surfaces of (a) real part and (b) imaginary part of $u(x,y,t)$ at time $t = 3$.

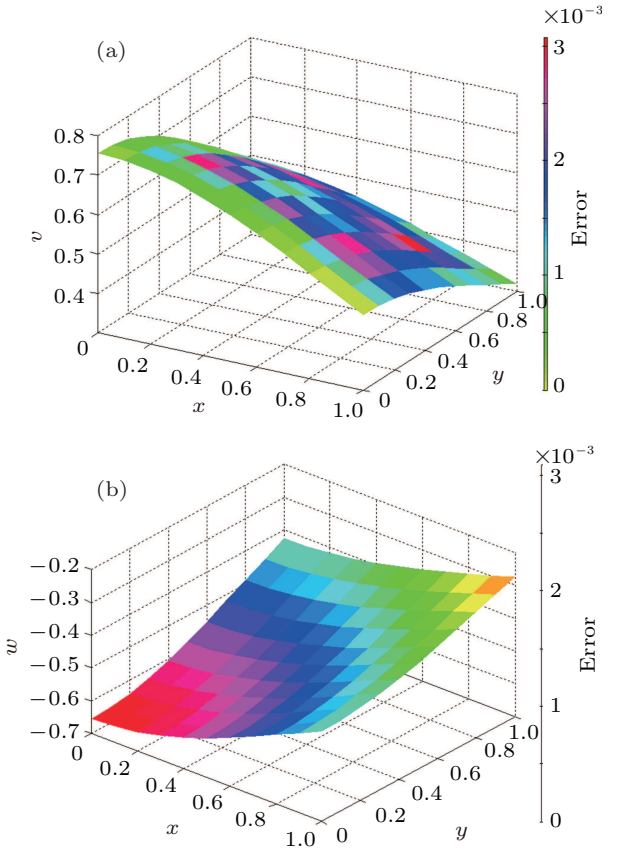



**Fig. 4.** (color online) Surfaces of (a) real part and (b) imaginary part of $u(x,y,t)$ at time $t = 4$.

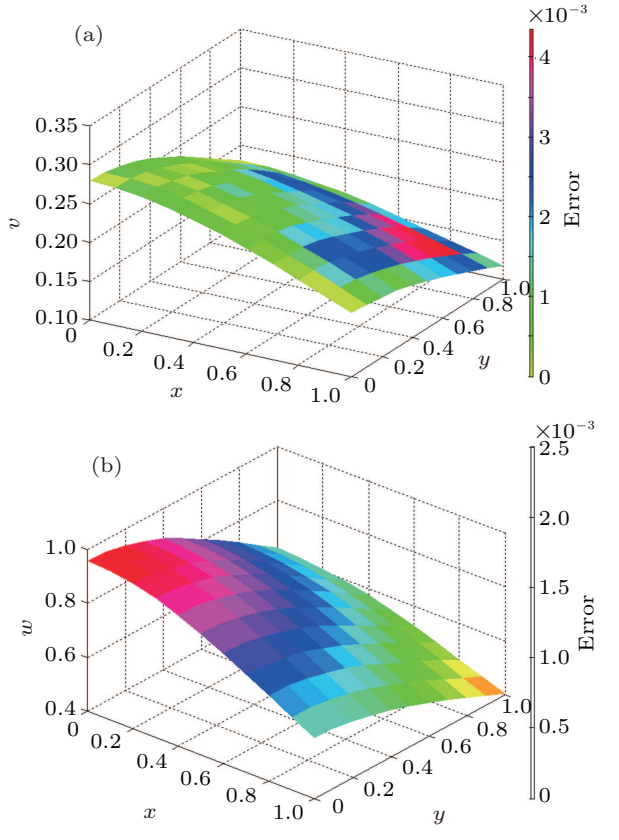



**Fig. 5.** (color online) Surfaces of (a) real part and (b) imaginary part of $u(x,y,t)$ at time $t = 6$.

### 3.2. Example 2

Consider the case $f(x,y)=0$ in Eq. (1) on $\Omega=[0,1]\times[0,1]$ with

$$u(x,y,0)=\varphi_1(x,y)=\exp(\mathrm{i}(x+y)), \tag{51}$$

which derives the analytical solution

$$u(x,y,t)=\exp(\mathrm{i}(x+y-2t)), \tag{52}$$

and the Dirichlet boundary conditions can be obtained from Eq. (52).

The IEFG method is used with penalty factor $\alpha=10^5$ and $d_{\max}=2.3$ in time steps of $\Delta t=0.01$.

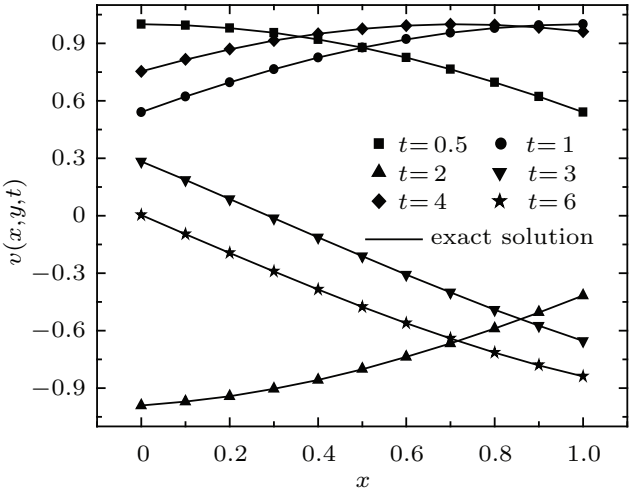



**Fig. 6.** Real parts of analytical solution and numerical solution of $u(x,y,t)$.

Figure 6 shows the numerical solution and analytical solution for real part of $u(x,y,t)$ at time $t=0.5$, 1, 2, 3, 4, and 6, respectively. Figure 7 shows the numerical solution and exact solution for imaginary part of $u(x,y,t)$ at $t=0.5$, 1, 2, 3, 4, and 6, respectively. In Figs. 8–10, the surfaces of real part and imaginary part of $u(x,y,t)$ are plotted at $t=1$, 3, and 6, respectively.

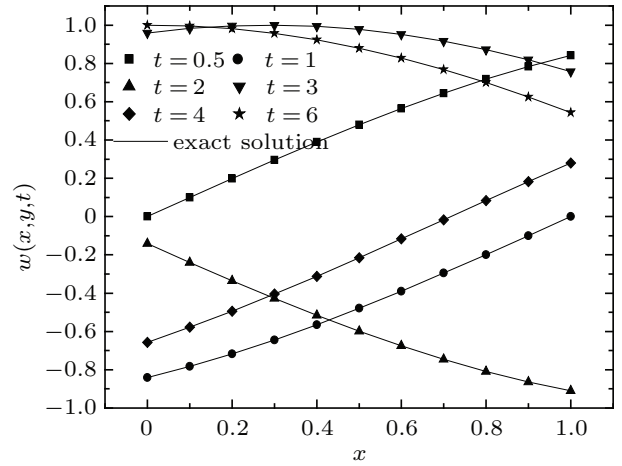



**Fig. 7.** Imiginary parts of analytical solution and numerical solution of $u(x,y,t)$.

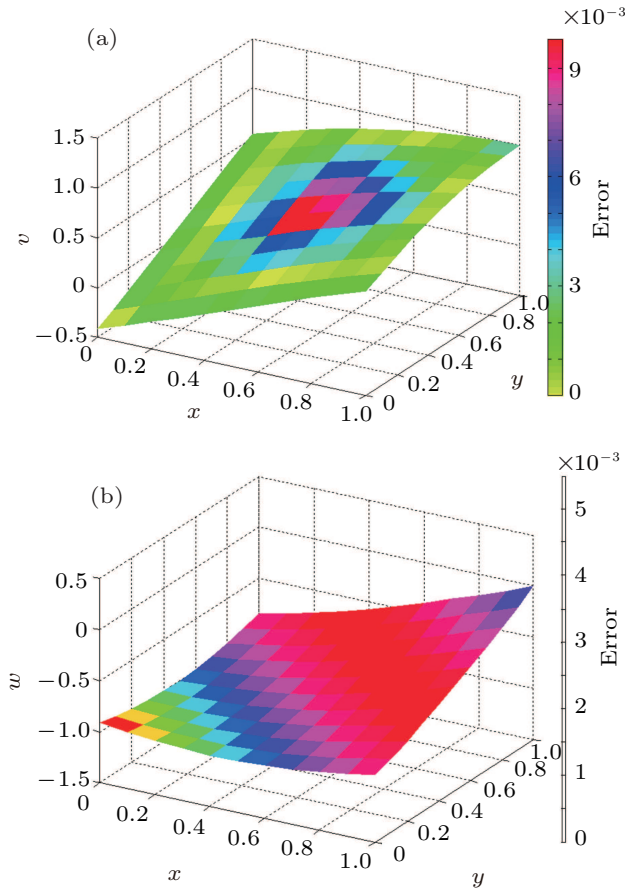



**Fig. 8.** (color online) Surfaces of (a) real part and (b) imaginary part of $u(x,y,t)$ at time $t=1$.

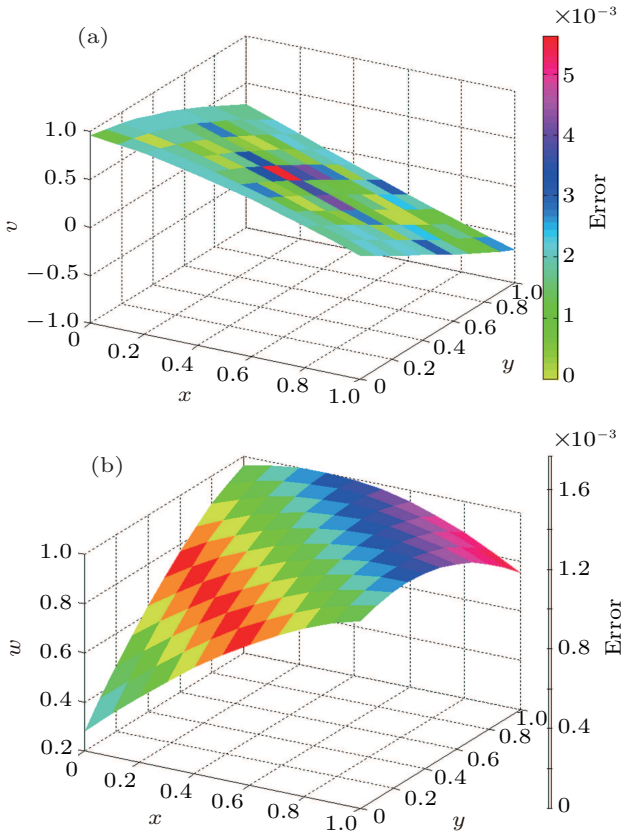



**Fig. 9.** (color online) Surfaces of (a) real part and (b) imaginary part of $u(x,y,t)$ at time $t=3$.

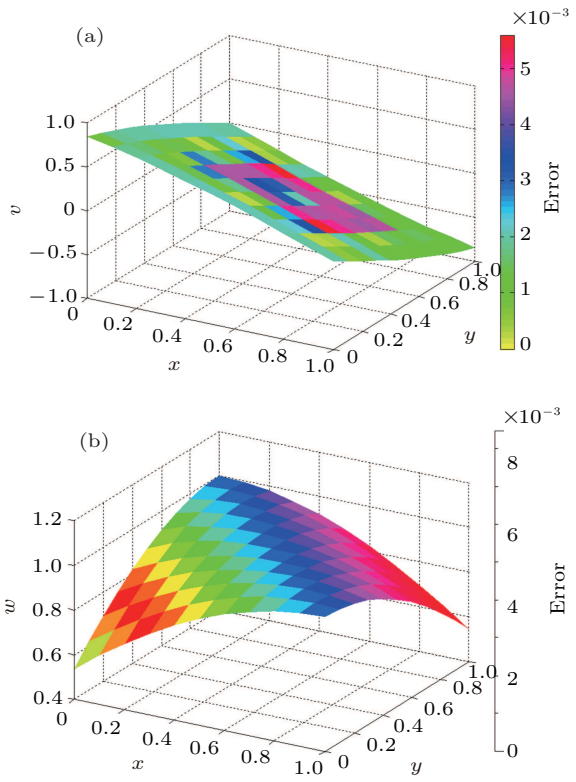



**Fig. 10.** (color online) Surfaces of (a) real part and (b) imaginary part of $u(x,y.t)$ at time $t = 6$.

### 3.3. Example 3

Consider Eq. (1) on $\Omega = [0,1] \times [0,1]$ with

$$f(x,y) = 1 - 2/x^2 - 2/y^2, \tag{53}$$

and the analytical solution is

$$u(x,y,t) = x^2 y^2 \exp(\mathrm{i}t). \tag{54}$$

The initial and boundary conditions are easily derived from Eq. (54) as

$$u(x,y,0) = \varphi_1(x,y) = x^2 y^2, \tag{55}$$

$$u(x,0,t) = u(0,y,t) = 0, \tag{56}$$

$$u(x,1,t) = x^2 \exp(\mathrm{i}t), \tag{57}$$

$$u(1,y,t) = y^2 \exp(\mathrm{i}t). \tag{58}$$

The IEFG method is used with penalty factor $\alpha = 10^5$ and $d_{\max} = 2.3$ in time steps of $\Delta t = 0.01$. Figure 11 shows the numerical solution and analytical solution for real part of $u(x,y,t)$ when $t = 0.5$, 1, 2, 3, 4, and 6, respectively. Figure 12 describes the numerical solution and exact solution for imaginary part of $u(x,y,t)$ when $t = 0.5$, 1, 2, 3, 4, and 6, respectively. The surfaces of real and imaginary part of $u(x,y,t)$ are plotted at $t = 1$, 3, and 6, respectively, in Figs. 13–15.

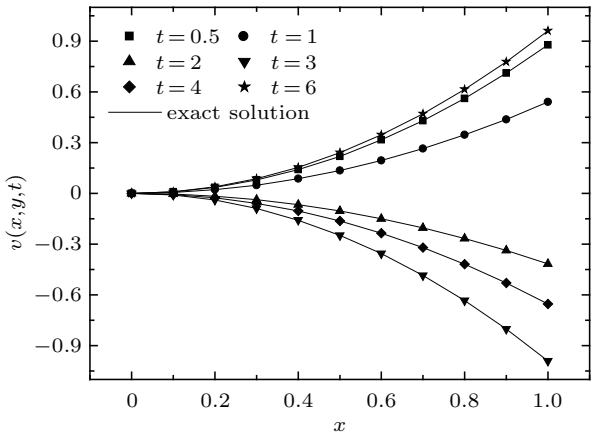



**Fig. 11.** Real parts of analytical solution and numerical solution of $u(x,y,t)$.

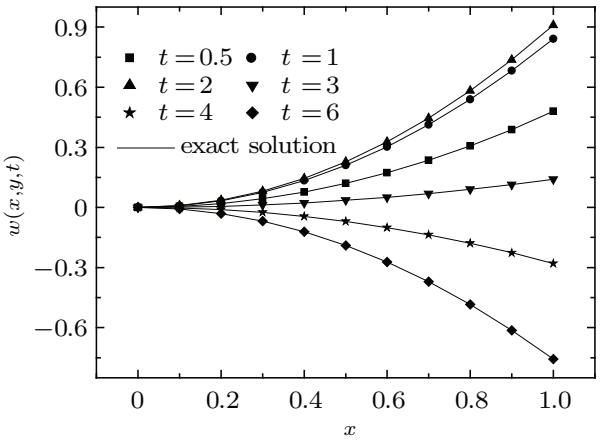



**Fig. 12.** Imaginary parts of analytical solution and numerical solution of $u(x,y,t)$.

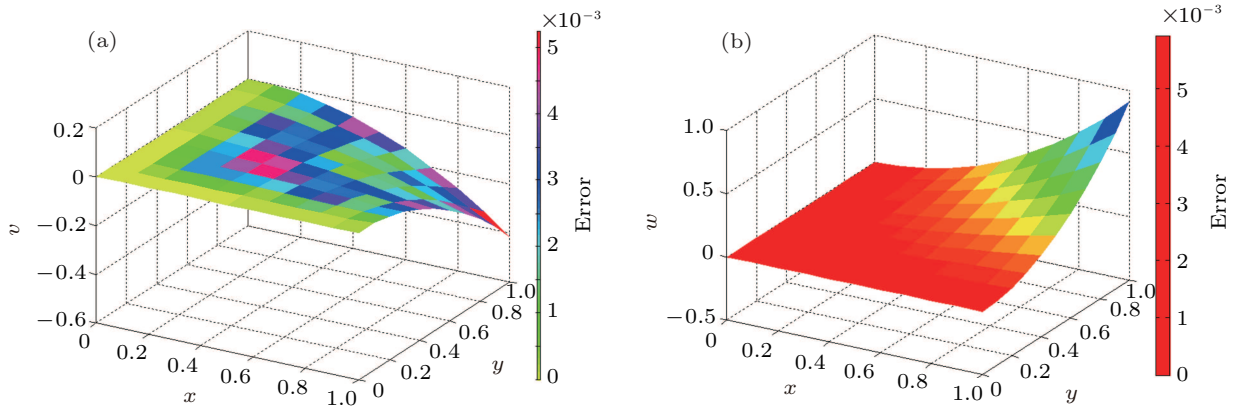



**Fig. 13.** (color online) Surfaces of (a) real part and (b) imaginary part of $u(x,y,t)$ at time $t = 6$.

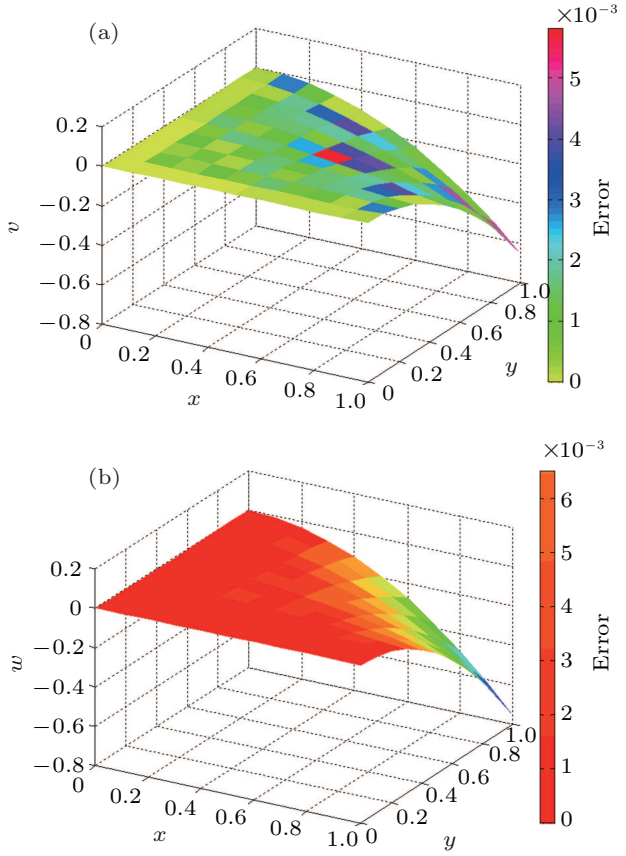



**Fig. 14.** (color online) Surfaces of (a) real part and (b) imaginary part of $u(x,y,t)$ at time $t = 4$.

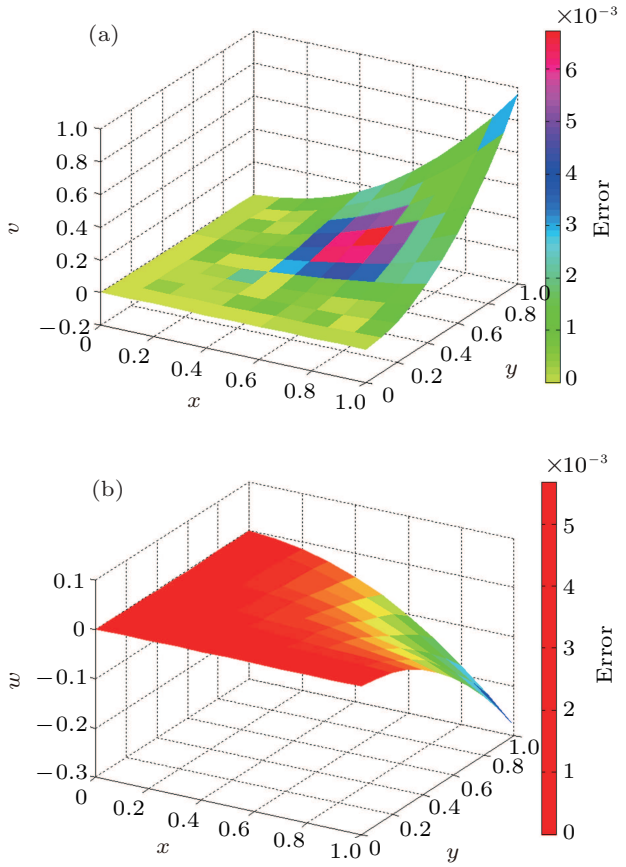



**Fig. 15.** (color online) Surfaces of (a) real part and (b) imaginary part of $u(x,y,t)$ at time $t = 6$.

From these figures, it is evident that the present numerical results obtained by the IEFG method are in excellent agreement with the analytical solutions.

## 4. Conclusions

The IEFG approach to solving 2D Schrödinger equation is developed and numerically implemented in the present work. The IMLS approximation is used to approximate the 2D trial function. By using the variation method, the final algebraic equation system is obtained due to energy functional. In the IMLS approximation, the basis function is chosen as an orthogonal function system, and the resulting final equation system is no more ill-conditioned due to involving no inverse matrix. It is demonstrated that the IMLS technique has a higher efficiency and precision than the traditional MLS approximation. Owing to its simple implementation, the IEFG method has the potential to replace the difference method and the finite element method for solving 2D Schrödinger equation and any other nonlinear partial differential equation.

## References

[1] Kivshar Y and Agrawal G P 2003 *Optical Solitons: From Fibers to Photonic Crystals* (New York: Academic Press)
[2] Deng Y X, Tu C H and Lu F Y 2009 *Acta Phys. Sin.* **58** 3173 (in Chinses)
[3] Zhao L, Sui Z, Zhu Q H, Zhang Y and Zuo Y L 2009 *Acta Phys. Sin.* **58** 4731 (in Chinses)
[4] Hasegawa A 1989 *Optical Solitons in Fibers* (Berlin: Springer-Verlag)
[5] Dodd R K, Eilbeck J C, Gibbon J D and Morris H C 1982 *Solitons and Nonlinear Wave Equations* (New York: Academic Press)
[6] Pitaevskii L P and Stringari S 2003 *Bose–Einstein Condensation* (Oxford: Oxford University Press)
[7] Scott A 1999 *Nonlinear Science: Emergence and Dynamics of Coherent Structures* (Vol. 1) (Oxford: Oxford University Press)
[8] Arnold A 1998 *VLSI Design* **6** 313
[9] Levy M 2000 *Institution of Electrical Engineers*, London
[10] Tappert F D 1977 *Lecture Notes in Physics* (Berlin: Springer)
[11] Kopylov Y V, Popov A V and Vinogradov A V 1995 *Opt. Commun.* **118** 619
[12] Huang W, Xu C, Chu S T and Chaudhuri S K 1992 *J. Lightwave Technology* **10** 295
[13] Subasi M 2002 *Numerical Methods for Partial Differential Equations* **18** 752
[14] Antoine X, Besse C and Mouysett V 2004 *Math. Comput.* **73** 1779
[15] Kalita J C, Chhabra P and Kumar S 2006 *J. Comput. Appl. Math.* **197** 141
[16] Dehghan M and Shokri A 2007 *Comput. Math. Appl.* **54** 136
[17] Dehghan M 2006 *Math. Comput. Simul.* **71** 16
[18] Dehghan M and Mirzaei D 2008 *Engineering Analysis with Boundary Elements* **32** 747
[19] Ang W T and Ang K C 2004 *Numerical Methods for Partial Differential Equations* **20** 843
[20] Belytschko T, Krongauz Y, Organ D, Fleming M and Krysl P 1996 *Comput. Method Appl. Mech. Eng.* **139** 3
[21] Cheng Y M and Li J H 2006 *Sci. China-Ser. G Phys. Mech. Astron.* **49** 46
[22] Cheng R J and Cheng Y M 2008 *Appl. Numer. Math.* **58** 884
[23] Peng M J and Cheng Y M 2009 *Engineering Analysis with Boundary Elements* **33** 77
[24] Li D M and Bai F N and Cheng Y M and Liew K M 2012 *Comput. Methods Appl. Mech. Eng.* **233–236** 1
[25] Lancaster P and Salkauskas K 1981 *Math. Comput.* **37** 141
[26] Cheng Y M and Chen M J 2003 *Acta Mech. Sin.* **35** 181

[27] Cheng Y M and Peng M J 2005 *Sci. China-Ser. G Phys. Mech. Astron.* **48** 641
[28] Liew K M, Cheng Y M and Kitipornchai S 2006 *Int. J. Numer. Methods Eng.* **65** 1310
[29] Kitipornchai S, Liew K M and Cheng Y M 2005 *Comput. Mech.* **36** 13
[30] Liew K M, Cheng Y M and Kitipornchai S 2005 *Int. J. Numer. Methods Eng.* **64** 1610
[31] Zhang Z, Liew K M and Cheng Y M 2008 *Engineering Analysis with Boundary Elements* **32** 100
[32] Zhang Z, Liew K M, Cheng Y M and Li Y Y 2008 *Engineering Analysis with Boundary Elements* **32** 241
[33] Zhang Z, Wang J F, Cheng Y M and Liew K M 2013 *Sci. China-Phys. Mech. Astron.* **56** 1568
[34] Zhang Z, Cheng Y M and Liew K M 2013 *Engineering Analysis with-Boundary Elements* **37** 1576
[35] Ren H P, Cheng Y M and Zhang W 2009 *Chin. Phys. B* **18** 4065
[36] Ren H P, Cheng Y M and Zhang W 2010 *Sci. China-Ser. G Phys. Mech. Astron.* **53** 758
[37] Ren H P and Cheng Y M 2011 *Int. J. Appl. Mech.* **3** 735
[38] Ren H P and Cheng Y M 2012 *Engineering Analysis with Boundary Elements* **36** 873
[39] Sun F X, Wang J F, Cheng Y M and Huang A X 2015 *Appl. Numer. Math.* **98** 79
[40] Cheng R J and Liew K M 2012 *Engineering Analysis with Boundary Elements* **36** 203
[41] Cheng R J and Liew K M 2009 *Comput. Mech.* **45** 1
[42] Cheng R J and Liew K M 2012 *Engineering Analysis with Boundary Elements* **36** 1322
[43] Cheng R J and Liew K M 2012 *Comput. Method Appl. Mech. Eng.* **245–246** 132
[44] Wang J F, Sun F X and Cheng Y M 2012 *Chin. Phys. B* **21** 090204
[45] Wang J F, Wang J F, Sun F X and Cheng Y M 2013 *Int. J. Comput. Methods* **10** 1350043
[46] Sun F X, Wang J F, Cheng Y M and Huang A X 2015 *Appl. Numer. Math.* **98** 79
[47] Chen L and Cheng Y M 2010 *Sci. China-Ser. G Phys. Mech. Astron.* **53** 954
[48] Chen L and Cheng Y M 2010 *Chin. Phys. B* **19** 090204
[49] Chen L, Cheng Y M and Ma H P 2015 *Comput. Mech.* **55** 591
[50] Peng M J, Liu P and Cheng Y M 2009 *Int. J. Appl. Mech.* **1** 367
[51] Peng M J, Li D M and Cheng Y M 2011 *Engineering Structure* **33** 127
[52] Li D M, Peng M J and Cheng Y M 2011 *Sci. China-Ser. G Phys. Mech. Astron.* **41** 1003
[53] Cheng Y M, Li R X and Peng M J 2012 *Chin. Phys. B* **21** 090205
[54] Bai F N, Li D M, Wang J F and Cheng Y M 2012 *Chin. Phys. B* **21** 020204
[55] Cheng Y M, Liu C, Bai F N and Peng M J 2015 *Chin. Phys. B* **24** 100202
[56] Deng Y J, Liu C, Peng M J and Cheng Y M 2015 *Int. J. Appl. Mech.* **7** 1550017