Acta Mechanica Solida Sinica
https://doi.org/10.1007/s10338-019-00151-1

ISSN 1860-2134

PIMOL: A New Semi-analytical Method Based on the Finite Difference Method of Lines and the Precise Integration Method

Yongjun Xu[1*]

([1] *Key Laboratory for Mechanics in Fluid Solid Coupling Systems, Institute of Mechanics, Chinese Academy of Sciences, Beijing 100190, China*)

Received 2 May 2019; revision received 8 October 2019; Accepted 10 October 2019

ABSTRACT The aim of this paper is to introduce a new semi-analytical method, namely PIMOL (precise integration method of lines, the parametric finite difference method of lines based on the precise integration method), which is developed and used to solve the ordinary differential equation (ODEs) systems based on the finite difference method of lines and the precise integration method (PIM). Two examples of Poisson's equation problems are given: a boundary value problem and an ODE eigenvalue problem. The PIMOL can effectively reduce a semi-discrete ODE problem to a linear algebraic matrix equation. Numerical results show that the PIMOL is a powerful method.

KEY WORDS PIMOL, ODE, FDMOL, PIM, Poisson's equation, Semi-analytical, Semi-discrete

1. Introduction

The precise integration method and method of lines (PIMOL) discussed in this paper is a newly developed semi-analytical algorithm scheme for solving the boundary value problems (BVPs) of elliptic type. It is based on the finite difference method of lines (FDMOL) and the precise integration method (PIM). According to the review of MOL-related studies (see, for instance, [1]), the key of the method of lines (MOL) is to semi-discretize a partial differential equation (PDE) into a system of ordinary differential equations (ODEs) defined on discrete lines by means of replacing the derivatives with respect to all but independent variables with the finite differences (FDs). The resulting two-point boundary value ODEs may then be solved by analytical or numerical methods. Due to the requirement of regular domain, inflexibility of meshes and ODE solving, the conventional MOL did not attract much attention, and the related investigations and applications were limited. Some applications of MOL in the Poisson's equation and other BVPs can be found in [2–5] by Meyer and Janac, respectively. The MOL has also been applied to solid mechanics by Irob [6], Gyekenyesi and Mendelson [7], Malik and Fu [8, 9], Mendelson and Alam [10], and Alam and Mendelson [11]. In most of the aforementioned applications, the ODEs were solved by *ad hoc* shooting-like numerical processes. Jones et al. [12], however, studied the convergence of the MOL solution and found that the ODEs resulting from the MOL may be inherently unstable for shooting methods. Xanthis [13, 14] and Yuan [15] solved the system of ODEs by using an ODE solver and further developed a new computational tool in structural

* Corresponding author. E-mail: yjxu@imech.ac.cn

Published online: 30 October 2019

analysis, i.e., the FEMOL based on the finite element discrete ideas and a modern ODE solver [15–17]. The PIM is a powerful method for solving ODEs of both initial problems and boundary value problems [18–20]. In this paper, the FDMOL is equipped with the PIM, and the old method will be gaining new value, power, and efficiency.

2. The Finite Difference Method of Lines

To explore the finite difference method of lines, we consider the following Poisson's equation defined on a rectangular domain as shown in Fig. 1 [17].

$$\nabla^2 u = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = -f \tag{1}$$

which is subject to the Dirichlet boundary condition

$$u = 0, \qquad x = \pm a, \quad y = \pm b \tag{2}$$

For simplicity, we assume that $f(x, y)$ is bi-symmetric. Thus, we can solve the problem on a quarter of the domain, which is semi-discrete by $N + 1$ equally spaced vertical lines with distance $h = a/N$ as shown in Fig. 2.

By defining

$$u_i = u_i(y) = u(x_i, y) \tag{3}$$

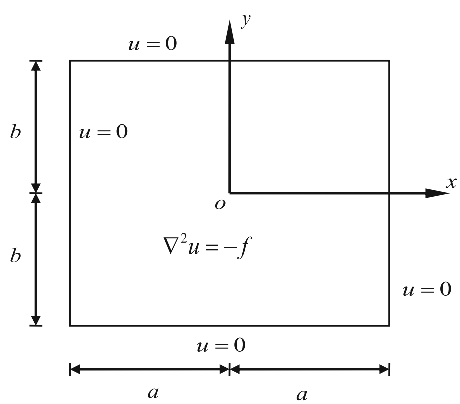

Fig. 1. The Poisson's equation

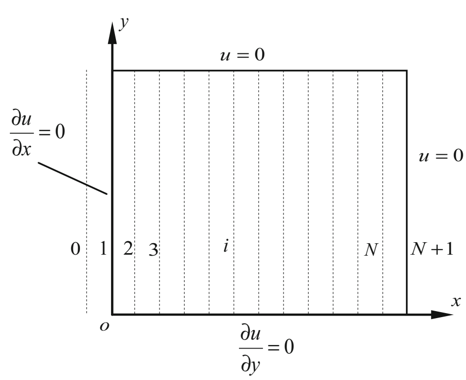

Fig. 2. FDMOL mesh for a quarter domain

and using the three-point central difference of accuracy $O\left(h^2\right)$ to approximate the partial derivatives with respect to the independent variable x at $x = x_i$,

$$\left(\frac{\partial u}{\partial x}\right)_i = \frac{u_{i+1} - u_{i-1}}{2h} + O\left(h^2\right) \; ; \qquad \left(\frac{\partial^2 u}{\partial x^2}\right)_i = \frac{u_{i+1} - 2u_i + u_{i-1}}{h^2} + O\left(h^2\right) \tag{4}$$

the typical FDMOL equation on an interior line i can be written as a second-order ODE of the following

$$u_i'' = -\frac{u_{i+1} - 2u_i + u_{i-1}}{h^2} - f_i, \qquad y \in (0, b)\,, \quad i = 2, 3, \ldots, N-1 \tag{5}$$

where $()' = \partial\,()/\partial y$, $\;f_i = f_i\,(y) = f\,(x_i, y)$. To establish the FDMOL equation on the first line, an auxiliary line $i = 0$ is introduced. Using the Neumann-type boundary condition $\partial u/\partial x = 0$ at the left boundary line yields $u_0 = u_2$, which eliminates the line function u_0 at the auxiliary line. The ODE of the first line can be rewritten as

$$u_1'' = -\frac{2u_2 - 2u_1}{h^2} - f_1, \qquad y \in (0, b) \tag{6}$$

For the right boundary line $i = N + 1$, since $u_{N+1} = 0$, the ODE on line $i = N$ can be rewritten as

$$u_N'' = -\frac{-2u_N + u_{N-1}}{h^2} - f_N, \qquad y \in (0, b) \tag{7}$$

The end-point boundary conditions for each line are

$$u_i'\,(0) = 0, \qquad u_i\,(b) = 0, \qquad i = 1, 2, 3, \ldots, N \tag{8}$$

3. PIMOL: Standard Formulation of FDMOL Based on PIM

3.1. Precise Integration Method [18]

The ordinary differential equations of any order can always be changed into an equivalent system of first-order ODEs. A set of ODEs can be given in the matrix/vector form as

$$\boldsymbol{v}' = \boldsymbol{A v} + \boldsymbol{f} \tag{9}$$

where a prime (') stands for the derivative with respect to ξ, $\boldsymbol{v}\,(\xi)$ is an n-dimensional vector function to be determined, $\boldsymbol{A}$ is a given $N \times N$ constant matrix, and $\boldsymbol{f}\,(\xi)$ is a given n-dimensional external force vector.

For the homogeneous equations,

$$\boldsymbol{v}' = \boldsymbol{A v} \tag{10}$$

Because $\boldsymbol{A}$ is a ξ-invariant matrix, its general solution can be given as

$$\boldsymbol{v} = \exp(\boldsymbol{A}\xi) \cdot \boldsymbol{v}_0 \tag{11}$$

where $\boldsymbol{v}_0 = \boldsymbol{v}\,(\xi_0)$ is assumed to be a known vector boundary condition.

The solution of Eq. (9) can be obtained by using Duhamel integration

$$\boldsymbol{v} = \exp(\boldsymbol{A} \cdot (\xi - \xi_0)) \cdot \boldsymbol{v}_0 + \int_{\xi_0}^{\xi} \exp\left(\boldsymbol{A} \cdot (\xi - \zeta)\right) \boldsymbol{f}\,(\zeta)\, \mathrm{d}\zeta \tag{12}$$

For calculation of $\exp(\boldsymbol{A}t)$, $t = \xi - \xi_0$ for a given ξ and the precise numerical calculation of the second integration part, and the precise numerical integration is also focused on the precise computation of $\exp(\boldsymbol{A}t)$ for a given $t = \xi - \zeta$.

3.2. PIMOL Algorithm

In order to change the governing equations of Eqs. (5–7) into an equivalent system of first-order ODEs, we define a new identity function on each line as

$$v_i = u_i', \qquad y \in (0, b), \qquad i = 1, 2, 3, \ldots, N \tag{13}$$

and then the governing equations of Eqs. (5–7) can be rewritten as the following equivalent system of first-order ODEs

$$\begin{cases} v_1' = -\frac{2u_2 - 2u_1}{h^2} - f_1 \\ v_i' = -\frac{u_{i+1} - 2u_i + u_{i-1}}{h^2} - f_i \qquad y \in (0, b), \quad i = 2, 3, \ldots, N-1 \\ v_N' = -\frac{-2u_N + u_{N-1}}{h^2} - f_N \end{cases} \tag{14}$$

Based on Eqs. (13) and (14), a set of first-order ODEs can be given in the matrix/vector form as

$$\boldsymbol{U'} = \boldsymbol{A}\boldsymbol{U} + \boldsymbol{F}, \qquad y \in (0, b) \tag{15}$$

where $\boldsymbol{A}$ is a $2N \times 2N$ matrix:

$$\boldsymbol{A} = \begin{bmatrix} \mathbf{0} & \boldsymbol{a} \\ \boldsymbol{I} & \mathbf{0} \end{bmatrix}, \quad \boldsymbol{a} = \frac{1}{h^2} \begin{array}{cc} \begin{bmatrix} 2 & -2 & 0 & \ldots & 0 & 0 & 0 & \ldots & 0 & 0 & 0 \\ -1 & 2 & -1 & \ldots & 0 & 0 & 0 & \ldots & 0 & 0 & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & 0 & \ldots & -1 & 2 & -1 & \ldots & 0 & 0 & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & 0 & \ldots & 0 & 0 & 0 & \ldots & -1 & 2 & -1 \\ 0 & 0 & 0 & \ldots & 0 & 0 & 0 & \ldots & 0 & -1 & 2 \end{bmatrix} & \begin{array}{c} \\ \\ \\ i \\ \\ \\ \\ \end{array} \\ \begin{array}{ccccccccccc} \quad & \quad & \quad & i & \quad & \quad & \quad & \quad & \quad & \quad & \quad \end{array} & \end{array}$$

$$\begin{aligned} &a_{11} = 2, \quad a_{12} = -2 \\ &a_{i\,i-1} = -1, \quad a_{ii} = 2, \quad a_{i\,i+1} = -1, \quad i = 2, 3, 4, \ldots, N \\ &\text{the others: } a_{ij} = 0 \end{aligned}$$

$$\begin{aligned} &\boldsymbol{U} = \{v_1, v_2, v_3, \ldots v_i, \ldots, v_N, u_1, u_2, u_3, \ldots u_i, \ldots u_N\}^{\mathrm{T}} \\ &\boldsymbol{F} = \{-f_1, -f_2, -f_3, \ldots, -f_i, \ldots, -f_N, 0, 0, 0, \ldots, 0 \ldots, 0\}^{\mathrm{T}} \end{aligned}$$

In addition, the end-point boundary conditions for each line can also be rewritten as

$$v_i(0) = 0, \qquad u_i(b) = 0, \qquad i = 1, 2, 3, \ldots, N \tag{16}$$

3.3. Solution Algorithm

The solutions of Eq. (15) can be expressed in the following form by using Duhamel integration as Eq. (12)

$$\boldsymbol{U}(y) = \exp(\boldsymbol{A}y)\boldsymbol{U}_0 + \int_0^y \exp(\boldsymbol{A} \cdot (y - t))\boldsymbol{F}(t)\,\mathrm{d}t, \qquad y \in (0, b) \tag{17}$$

When $y = b$, we have

$$\boldsymbol{U}_b = \boldsymbol{T}_b \boldsymbol{U}_0 + \hat{\boldsymbol{F}}_b, \quad \boldsymbol{T}_b = \exp(\boldsymbol{A}b), \quad \hat{\boldsymbol{F}}_b = \int_0^b \exp(\boldsymbol{A} \cdot (b - t))\boldsymbol{F}(t)\,\mathrm{d}t \tag{18}$$

where $\boldsymbol{T}_b$ is a $2N \times 2N$ matrix and $\hat{\boldsymbol{F}}_b$ is a $2N$ column vector.

Substituting the end-point boundary conditions Eq. (15) in Eq. (18)

$$\begin{aligned} &\boldsymbol{U}_b = \boldsymbol{T}_b \boldsymbol{U}_0 + \hat{\boldsymbol{F}}_b \\ &\quad \boldsymbol{U}_0 = \{0, 0, 0, \ldots 0, u_1(0), u_2(0), u_3(0), \ldots u_i(0), \ldots u_N(0)\}^{\mathrm{T}} = \{\mathbf{0}\ \bar{\boldsymbol{u}}_0\}^{\mathrm{T}} \\ &\quad \boldsymbol{U}_b = \{v_1(b), v_2(b), v_3(b), \ldots v_i(b), \ldots, v_N(b), 0, 0, 0, \ldots 0\}^{\mathrm{T}} = \{\bar{\boldsymbol{v}}_b\ \mathbf{0}\}^{\mathrm{T}} \\ &\quad \boldsymbol{T}_b = \begin{bmatrix} \bar{\boldsymbol{T}}_{11} & \bar{\boldsymbol{T}}_{12} \\ \bar{\boldsymbol{T}}_{21} & \bar{\boldsymbol{T}}_{22} \end{bmatrix}, \quad \hat{\boldsymbol{F}}_b = \{\bar{\boldsymbol{F}}_1\ \bar{\boldsymbol{F}}_2\}^{\mathrm{T}} \end{aligned} \tag{19}$$

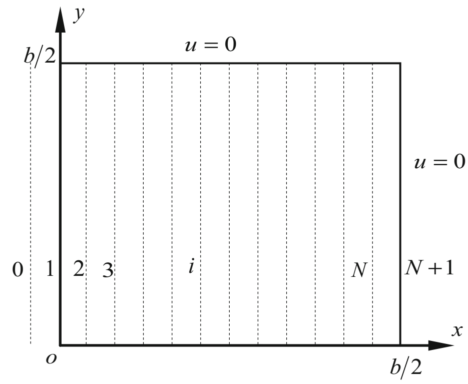

Fig. 3. FDMOL mesh for a square membrane

Notice that a semi-discrete BVP ODE problem is reduced to solving a set of linear algebraic equations with $2N$ unknowns. It is easy to obtain that

$$\begin{cases} \bar{\boldsymbol{u}}_0^{\mathrm{T}} = -\bar{\boldsymbol{T}}_{22}^{-1}\bar{\boldsymbol{F}}_2^{\mathrm{T}} \\ \bar{\boldsymbol{v}}_b^{\mathrm{T}} = \bar{\boldsymbol{T}}_{12}\bar{\boldsymbol{u}}_0^{\mathrm{T}} + \bar{\boldsymbol{F}}_1^{\mathrm{T}} \end{cases} \tag{20}$$

At this point, we can say that the problem is solved. For any point (x, y) in the domain in Fig. 2, the semi-analytical solutions with respect to y on each line can be obtained by using Eq. (17); for any x which falls out of the mesh lines, the interpolating method and many other methods can be used to obtain a relatively accurate solution.

4. ODE Eigenproblem Formulation of PIMOL

The free vibration of a unit square membrane is governed by the following eigenproblem PDE

$$\begin{array}{lll} \nabla^2 u + \lambda u = 0, & & -b/2 < x, y < b/2 \\ BCs: \quad u = 0, & & x = \pm b/2, \quad y = \pm b/2 \end{array} \tag{21}$$

By exploiting the symmetry and antisymmetry of the vibration modes, we can also solve this problem on a quarter of the entire domain with the following four boundary conditions on $x = 0$ and $y = 0$

(i) $x-$symmetric and $y-$symmetric

$$\frac{\partial u(0, y)}{\partial x} = 0, \qquad \frac{\partial u(x, 0)}{\partial y} = 0 \tag{22a}$$

(ii) $x-$symmetric and $y-$antisymmetric

$$\frac{\partial u(0, y)}{\partial x} = 0, \qquad u(x, 0) = 0 \tag{22b}$$

(iii) $x-$antisymmetric and $y-$symmetric

$$u(0, y) = 0, \qquad \frac{\partial u(x, 0)}{\partial y} = 0 \tag{22c}$$

(iv) $x-$antisymmetric and $y-$antisymmetric

$$u(0, y) = 0, \qquad u(x, 0) = 0 \tag{22d}$$

As shown in Fig. 3, by means of the FDMOL, the domain is meshed by $N + 1$ equally spaced vertical lines with distance $h = b/2N$. By defining $u_i = u_i(y) = u(x_i, y)$ and using the three-point central difference of accuracy $O\left(h^2\right)$ to replace the partial derivative with respect to x, the PDE is reduced to a set of BVP ODEs.

$$v_i' = -\frac{u_{i-1} - 2u_i + u_{i+1}}{h^2} - \lambda u_i, \qquad y \in \left(0, \frac{b}{2}\right) \tag{23}$$

with the modification of the last line and the upper end-point boundary condition on each line

$$u_{N+1} = 0, \qquad u_i\left(\frac{b}{2}\right) = 0 \tag{24}$$

The modification of the first line, the lower end-point boundary condition on each line and the range of line number are dependent on the boundary conditions as follows

$$\begin{array}{l}
\text{(i) } x- \text{ symmetric and } y- \text{symmetric} \\
\quad u_0 = u_2, \quad u_i'(0) = 0, \quad 1 \le i \le N \quad \text{(a)} \\
\text{(ii) } x- \text{ symmetric and } y- \text{ antisymmetric} \\
\quad u_0 = u_2, \quad u_i(0) = 0, \quad 1 \le i \le N \quad \text{(b)} \\
\text{(iii) } x- \text{ antisymmetric and } y- \text{ symmetric} \\
\quad u_1 = 0, \qquad u_i'(0) = 0, \quad 2 \le i \le N \quad \text{(c)} \\
\text{(iv) } x- \text{ antisymmetric and } y- \text{ antisymmetric} \\
\quad u_1 = 0, \qquad u_i(0) = 0, \quad 2 \le i \le N \quad \text{(d)}
\end{array} \tag{25}$$

By defining $v_i = u_i'$ on each line, a set of first-order ODEs can be given in matrix/vector form as

$$\boldsymbol{U}' = \boldsymbol{A}\boldsymbol{U} \tag{26}$$

where, for different boundary conditions as Eq. 25a–d, we have
(i) $x-$ symmetric and $y-$ symmetric
where $\boldsymbol{A}$ is a $2N \times 2N$ matrix and $\boldsymbol{U}$ is a $2N$ column vector

$$\boldsymbol{A} = \begin{bmatrix} \mathbf{0} & \boldsymbol{a} \\ \boldsymbol{I} & \mathbf{0} \end{bmatrix}$$

$$\boldsymbol{a} = \frac{1}{h^2}\begin{array}{cc}
\left[\begin{array}{ccccccccccc}
2-\lambda h^2 & -2 & 0 & \ldots & 0 & 0 & 0 & \ldots & 0 & 0 & 0 \\
-1 & 2-\lambda h^2 & -1 & \ldots & 0 & 0 & 0 & \ldots & 0 & 0 & 0 \\
\ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\
0 & 0 & 0 & \ldots & -1 & 2-\lambda h^2 & -1 & \ldots & 0 & 0 & 0 \\
\ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\
0 & 0 & 0 & \ldots & 0 & 0 & 0 & \ldots & -1 & 2-\lambda h^2 & -1 \\
0 & 0 & 0 & \ldots & 0 & 0 & 0 & \ldots & 0 & -1 & 2-\lambda h^2
\end{array}\right] &
\begin{array}{c} \\ \\ \\ i \\ \\ \\ \\ \end{array} \\
\begin{array}{ccccccccccc} & & & & i & & & & & & \end{array} &
\end{array}$$

$$\boldsymbol{U} = \{v_1, v_2, v_3, \ldots v_i, \ldots, v_N, u_1, u_2, u_3, \ldots u_i, \ldots u_N\}^{\mathrm{T}} \tag{27}$$

In addition, the end-point boundary conditions of each line can also be rewritten as

$$v_i(0) = 0, \qquad u_i\left(\frac{b}{2}\right) = 0, \qquad\qquad i = 1, 2, 3, \ldots, N$$

(ii) $x-$ symmetric and $y-$ antisymmetric
Both $\boldsymbol{A}$ and $\boldsymbol{U}$ are the same as in case (i). In addition, the end-point boundary conditions of each line can also be rewritten as

$$u_i(0) = 0, \qquad u_i\left(\frac{b}{2}\right) = 0, \qquad\qquad i = 1, 2, 3, \ldots, N$$

(iii) $x-$ antisymmetric and $y-$ symmetric where $\boldsymbol{A}$ is a $2(N-1)\times 2(N-1)$ matrix and $\boldsymbol{U}$ is a $2(N-1)$ column vector:

$$\boldsymbol{A}=\begin{bmatrix}\boldsymbol{0} & \boldsymbol{a}\\ \boldsymbol{I} & \boldsymbol{0}\end{bmatrix}$$

$$\boldsymbol{a}=\frac{1}{h^2}\begin{array}{l}\left[\begin{array}{lllllllllll}
2-\lambda h^2 & -1 & 0 & \dots & 0 & 0 & 0 & \dots & 0 & 0 & 0\\
-1 & 2-\lambda h^2 & -1 & \dots & 0 & 0 & 0 & \dots & 0 & 0 & 0\\
\dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots\\
0 & 0 & 0 & \dots & -1 & 2-\lambda h^2 & -1 & \dots & 0 & 0 & 0\\
\dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots\\
0 & 0 & 0 & \dots & 0 & 0 & 0 & \dots & -1 & 2-\lambda h^2 & -1\\
0 & 0 & 0 & \dots & 0 & 0 & 0 & \dots & 0 & -1 & 2-\lambda h^2
\end{array}\right]\begin{array}{l}2\\ \\ \\ i\\ \\ \\ N\end{array}\\
\begin{array}{lllllllllll}2 & & i & & & N & & & & & \end{array}\end{array}$$

$$\boldsymbol{U}=\{v_2,v_3,\dots v_i,\dots,v_N,u_2,u_3,\dots u_i,\dots u_N\}^{\mathrm{T}} \tag{28}$$

The end-point boundary conditions of each line can be rewritten as

$$v_i(0)=0,\qquad u_i\left(\frac{b}{2}\right)=0,\qquad\qquad i=2,3,\dots,N$$

(iv) $x-$antisymmetric and $y-$antisymmetric

Both $\boldsymbol{A}$ and $\boldsymbol{U}$ are the same as in case (iii). Furthermore, the end-point boundary conditions of each line can also be rewritten as

$$u_i(0)=0,\qquad u_i\left(\frac{b}{2}\right)=0,\qquad\qquad i=2,3,\dots,N$$

The general solution can be expressed as

$$\boldsymbol{U}(y)=\boldsymbol{T}\boldsymbol{U}_0,\qquad \boldsymbol{T}=\exp(\boldsymbol{A}y),\quad y\in\left(0,\frac{b}{2}\right) \tag{29}$$

In the above example, we consider case (i) and substitute the end-point boundary conditions in Eq. (26)

$$\begin{aligned}&\boldsymbol{U}_{\frac{b}{2}}=\boldsymbol{T}_{\frac{b}{2}}\boldsymbol{U}_0\\
&\quad\boldsymbol{U}_0=\{0,0,0,\dots 0,u_1(0),u_2(0),u_3(0),\dots u_i(0),\dots u_N(0)\}^{\mathrm{T}}=\left\{\boldsymbol{0}\ \bar{\boldsymbol{u}}_0\right\}^{\mathrm{T}}\\
&\quad\boldsymbol{U}_{\frac{b}{2}}=\left\{v_1\left(\tfrac{b}{2}\right),v_2\left(\tfrac{b}{2}\right),v_3\left(\tfrac{b}{2}\right),\dots v_i\left(\tfrac{b}{2}\right),\dots,v_N\left(\tfrac{b}{2}\right),0,0,0,\dots 0\right\}^{\mathrm{T}}=\left\{\bar{\boldsymbol{v}}_{\frac{b}{2}}\ \boldsymbol{0}\right\}^{\mathrm{T}}\\
&\quad\boldsymbol{T}_{\frac{b}{2}}=\begin{bmatrix}\bar{\boldsymbol{T}}_{11} & \bar{\boldsymbol{T}}_{12}\\ \bar{\boldsymbol{T}}_{21} & \bar{\boldsymbol{T}}_{22}\end{bmatrix}\end{aligned} \tag{30}$$

which can be rearranged as a set of linear algebraic equations with $2N$ unknowns:

$$\begin{bmatrix}\bar{\boldsymbol{T}}_{12} & -\boldsymbol{I}\\ \bar{\boldsymbol{T}}_{22} & \boldsymbol{0}\end{bmatrix}\begin{Bmatrix}\bar{\boldsymbol{u}}_0^{\mathrm{T}}\\ \bar{\boldsymbol{v}}_{\frac{b}{2}}^{\mathrm{T}}\end{Bmatrix}=0 \tag{31}$$

The eigenfunction with respect to λ can be obtained as

$$B(\lambda)=\det\begin{vmatrix}\bar{\boldsymbol{T}}_{12} & -\boldsymbol{I}\\ \bar{\boldsymbol{T}}_{22} & \boldsymbol{0}\end{vmatrix}=0\qquad\qquad \text{(i)} \tag{32}$$

and the corresponding eigenfunctions for the other three cases can also be easily obtained.

$$\begin{cases}B(\lambda)=\det\begin{vmatrix}\bar{\boldsymbol{T}}_{11} & -\boldsymbol{I}\\ \bar{\boldsymbol{T}}_{21} & \boldsymbol{0}\end{vmatrix}=0 & \qquad\text{(ii)}\\
B(\lambda)=\det\begin{vmatrix}\hat{\boldsymbol{T}}_{12} & -\boldsymbol{I}\\ \hat{\boldsymbol{T}}_{22} & \boldsymbol{0}\end{vmatrix}=0 & \qquad\text{(iii)}\\
B(\lambda)=\det\begin{vmatrix}\hat{\boldsymbol{T}}_{11} & -\boldsymbol{I}\\ \hat{\boldsymbol{T}}_{21} & \boldsymbol{0}\end{vmatrix}=0 & \qquad\text{(iv)}\end{cases} \tag{33}$$

So far, the semi-discrete ODE eigenproblem is thereby reduced to a matrix eigenvalue problem. In this example, the FDMOL based on PIM is used to solve the free vibration of square membranes,

Table 1. **PIMOL solution on a square domain**

NI	h	$u\|_o$	e (%) (Error)	$\frac{e}{100h^2}$	$\left.\frac{\partial u}{\partial y}\right\|_A$	e (%) (Error)	$\frac{e}{100h^2}$
2	0.5	0.57557	2.34674	0.0939	−1.32414	1.98839	0.0795
3	1/3	0.58307	1.07473	0.0967	−1.33856	0.92076	0.0829
4	0.25	0.58579	0.61267	0.0980	−1.34378	0.53444	0.0855
5	0.2	0.58707	0.39569	0.0989	−1.34623	0.35329	0.0883
6	1/6	0.58777	0.27698	0.0997	−1.34757	0.25425	0.0915
7	1/7	0.58819	0.20512	0.1005	−1.34838	0.19432	0.0952
8	0.125	0.58847	0.15835	0.1013	−1.34890	0.15532	0.0994
9	1/9	0.58866	0.12623	0.1022	−1.34926	0.12855	0.1041
10	0.1	0.58879	0.10324	0.1032	−1.34952	0.10938	0.1094
11	1/11	0.58889	0.08621	0.1043	−1.34971	0.09519	0.1152
12	1/12	0.58897	0.07325	0.1055	−1.34986	0.08438	0.1215
13	1/13	0.58903	0.06316	0.1067	−1.34997	0.07597	0.1284
14	1/14	0.58908	0.05515	0.0932	−1.35006	0.06928	0.1171
15	1/15	0.58911	0.04868	0.0823	−1.35011	0.06556	0.1108
16	1/16	0.58914	0.04339	0.0733	−1.35021	0.05844	0.0988
Analytical [21]		0.5894			−1.351		

NL = NI + 1, NI: number of intervals, NL: number of lines

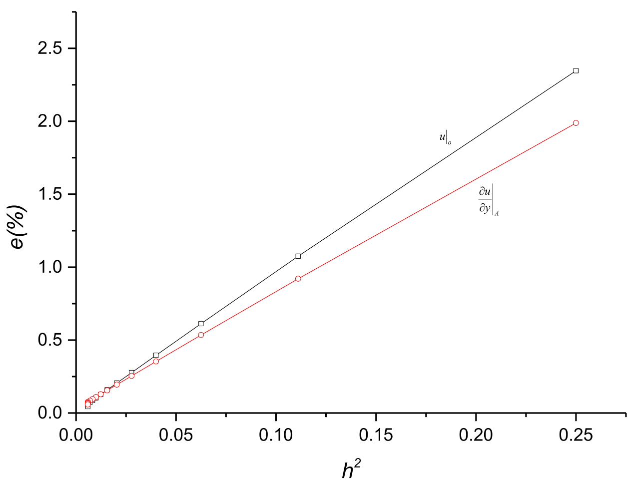

Fig. 4. The error of displacement and its derivative along h^2

an eigenvalue problem. The problem is reduced to an ODE eigenvalue problem by semi-discretization with FDMOL, and then the ODE eigenproblem is reduced to a matrix eigenproblem by PIM, which can be solved by many methods, such as the imbedding method (IBM) [22], the Müller Method [25], the inverse iteration method [23], the super inverse iteration method [24].

5. Numerical Examples

We use the following two examples, calculated by self-programming program with the computer package Maple 17.00 [26], to explore the precision and efficiency of this new semi-analytical algorithm of PIMOL. The focus is on the numerical integration part in Eq. (15) for a given y. In this paper, the Gaussian integral method is adopted to guarantee the convergence of a large-scale matrix $\boldsymbol{A}$.

Before any numerical examples are given, let us remark that, as a semi-discrete method, the discretization errors introduced in the FDMOL formulation are limited to the x direction in terms of h, as long as the associated ODE system can be accurately solved based on the PIM. Analytical solution is obtained along the mesh line by ignoring the error from the numerical integration.

Example 1 A square membrane subject to a uniform transverse load

Table 2. **Eigenvalues for vibration of a square membrane**

Case (i) $x-$sym. $y-$sym.			Case (iv) $x-$antisym. and $y-$antisym.		
PIMOL	Exact [17]	Error (%)	PIMOL	Exact	Error (%)
19.71893	19.739	0.102	78.63320	78.957	0.410
97.06439	98.696	1.653	192.26482	197.39	2.596
98.67577	98.696	0.021	197.06846	197.39	0.163
176.02122	177.65	0.917	310.70007	315.83	1.624
244.18418	256.61	4.842	369.25022	394.78	6.467
Case (ii) $x-$sym. and $y-$antisym.			Case (iii) $x-$antisym. $y-$sym.		
PIMOL	Exact	Error (%)	PIMOL	Exact	Error (%)
49.32775	49.348	0.041	49.02439	49.348	0.656
126.67320	128.3	1.268	127.98123	128.3	0.248
167.76300	167.78	0.010	162.65601	167.78	3.054
245.10845	246.74	0.661	241.61284	246.74	2.078
273.79299	286.22	4.342	285.89490	286.22	0.114

NI = 10

Table 3. **The first eigenvalue with different mesh lines and errors**

NI	PIMOL	e (Error)	h^2	e/h^2
2	19.24219	0.02517	0.06250	0.40271
3	19.51578	0.01131	0.02778	0.40712
4	19.61302	0.00638	0.01563	0.40845
5	19.65830	0.00409	0.01000	0.40883
6	19.68297	0.00284	0.00694	0.40878
7	19.69786	0.00208	0.00510	0.40847
8	19.70754	0.00159	0.00391	0.40800
9	19.71418	0.00126	0.00309	0.40738
10	19.71893	0.00102	0.00250	0.40667
11	19.72245	0.00084	0.00207	0.40584
12	19.72512	0.00070	0.00174	0.40491
13	19.72721	0.00060	0.00148	0.40389
14	19.72886	0.00051	0.00128	0.40277
15	19.73019	0.00045	0.00111	0.40157
16	19.73128	0.00039	0.00098	0.40027
17	19.73219	0.00035	0.00087	0.39889
18	19.73295	0.00031	0.00077	0.39743
19	19.73359	0.00027	0.00069	0.39591
20	19.73440	0.00023	0.00063	0.37264

NL = NI + 1, Exact = 19.739

Let $a = b = 1,\ f(x, y) = 2$. The corresponding physical model is an elastic torsion of a square bar or a small deflection of a square membrane subject to a uniform transverse load. The computed results are given in Table 1; comparing with the analytical solution, it can be seen that the accuracy is satisfactory and the convergence of u is indeed within the order of h^2, as shown in Fig. 4.

Example 2 Free vibration of square membranes

As shown in Fig. 3, with $b = 1$, the corresponding physical model is a free vibration of a square membrane. This example is solved by using an efficient algorithm based on the imbedding method (IBM) [22] and the Müller method [25]. The computed results are given in Tables 2 and 3 (Fig. 5). We can still see that the accuracy of the analytical solution is satisfactory and the convergence of eigenvalue is indeed within the order of h^2, as shown in Fig. 6.

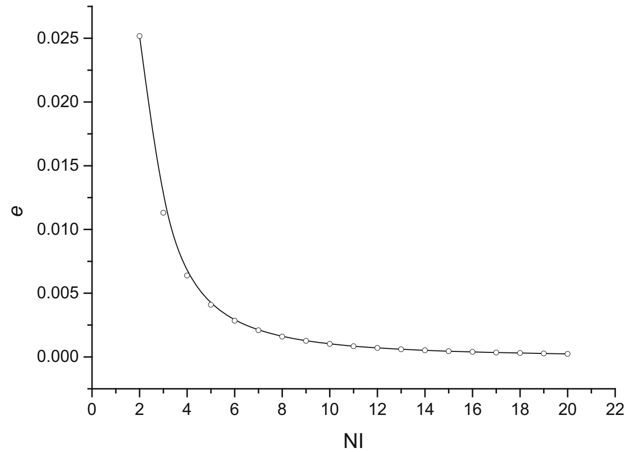

Fig. 5. The error along the interval number

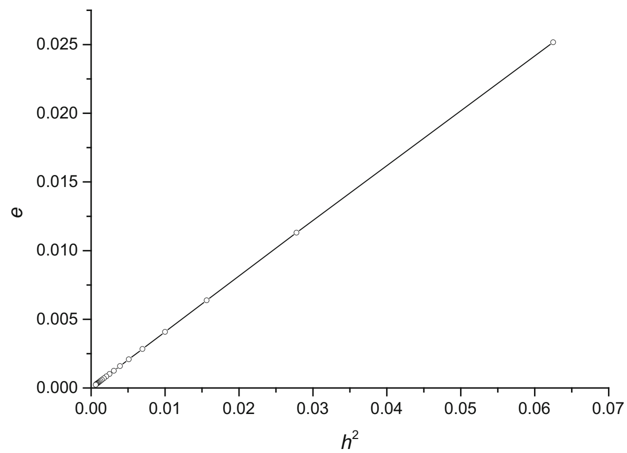

Fig. 6. The error along h^2

6. Conclusions

A new semi-analytical method for solving BVPs of elliptic type is presented, two examples of BVPs of elliptic type and ODE eigenvalue problem are given, and the numerical results show that PIMOL is a powerful method. It is of great value that PIMOL can reduce a semi-discrete BVP problem to a linear algebraic matrix equation problem.

On the basis of numerical experimentation discussed above, the following conclusions can be drawn.

(1) *New semi-analytical method* The presented method, PIMOL, is a newly developed semi-analytical method for elliptic BVPs. In this method, the PDEs defined on arbitrary domains (regular domain is discussed in this paper, and arbitrary domains will be given in another paper) are semi-discredited by MOL into a system of ODEs defined on discrete mesh lines, and then the analytical result is expressed in algebraic matrix equations by using the precise integration method. The PIMOL completely changes the PDEs of elliptic type into a linear algebraic matrix equation.

(2) *Generality* PIMOL is not only restricted to only the Poisson's equation problems: It is easy to be extended to plane problems, plate and shell problems, 3D problems, and so on. It can also be extended to other subjects such as the pipes conveying fluid, fluid–structure interaction, and multibody dynamics. And it is also easy to be extended to the parametric finite deference method of lines and the finite element method of lines.

(3) *Accuracy* Theoretically, PIMOL is a semi-analytical method. As such, the highly precise results are guaranteed by semi-discrete approximation of MOL. However, when the exponential matrix is large,

the analytical expression can be computed directly, and the numerical integration by PIM is inevitable. Fortunately, several numerical algorithms guarantee that the solutions have a desirable accuracy.

(4) *Reliability* The results are compared with the exact solutions which show good agreement.

(5) *Efficiency* The present work has demonstrated that PIMOL has high precision and computational efficiency in solving PDEs of elliptic type.

References

[1] Liskovets A. The method of lines (review). English translation appeared in Differential Equation'. 1965;(1):1308–23.
[2] Meyer GH. An application of the method of lines to mutidimensional free boundary problems. J Inst Math Appl. 1977;20(3):317–29.
[3] Meyer GH. The method of line for Poisson's equation with nonlinear or free boundary conditions. Numerische Mathematik. 1978;29:329–44.
[4] Meyer GH. The method of line and invariant imbedding for elliptic and parabolic free boundary problem. SIAM J Numer Anal. 1981;18(1):150–64.
[5] Janac K. Hybrid solution of the Poisson equation. In: Vichnetsky R (ed) Advances in computer methods for partial differential equation;1975. p. 154–8.
[6] Irobe M. Method of numerical analysis for three dimensional elastic problem. In: Proceedings of the 16th Japan National Congress for Applied Mechanics, Central Scientific Publ., 1968:1–7.
[7] Gyekenyesi JP, Mendelson A. Three-dimensional elastic stress and displacement analysis of finite geometry solids containing cracks. Int J Fract. 1975;11(3):409–29.
[8] Malik SN, Fu LS. Elastic-plastic response of an externally cracked cylinder analyzed by the method of lines. Eng Fract Mech. 1979;12(3):377–85.
[9] Fu LS, Malik SN. The method lf lines applied to crack problems including the plasticity effect. Comput Struct. 1979;10(3):447–56.
[10] Mendelson A, Alam J. The use of the method of lines in 3-D fracture mechanics analyses with applications to compact tension specimens. Int J Fract. 1983;22:105–16.
[11] Alam J, Mendelson A. Elastic-plastic stress analysis of CT specimens. Int J Fract. 1986;31:17–28.
[12] Jones DJ, South JC Jr, Klunker EB. On the numerical solution of elliptic partial differential equations by the method of lines. J Comput Phys. 1972;9(3):496–527.
[13] Xanthis LS. The numerical method of lines (MOL) and ODE solvers can provide a new powerful computational fracture mechanics tool. In: International conference on computational mechanics, Tokyo Japan, 1986.
[14] Xanthis LS. A pseudo-ODE modeling trick for the direct method of lines computation of important fracture mechanics parameters. ACM SIGNUM Newsletter, 1986;21(1–2).
[15] Yuan S. The finite element method of lines—theory and applications, science press. New York: Beijing; 1993.
[16] Xanthis LS, Schwab C. The method of arbitrary lines. C R Acad Sci Paris, t. 312, Serie I, 1991:181–187.
[17] Yuan S. The finite element method of lines. J Numer Methods Comput Appl. 1992;13(4):252–60 (in Chinese).
[18] Zhong WX. On precise integration method. J Comput Appl Math. 2004;163:59–78.
[19] Moler CB, Van Loan CF. Nineteen dubious ways to compute exponential of a matrix. SIAM Rev. 1978;20:801–36.
[20] Zhong WX, Zhu JP, Zhong XX. A precise time integration algorithm for nonlinear systems. Proc WCCM-3, 1994;(1):12–17.
[21] Timoshenko SP, Goodier JN. Theory of elasticity. 3rd ed. New York: McGraw-Hill; 1970.
[22] Xu YJ, Yuan S. Complete eigen-solution for plane notches with multi-materials by the imbedding method. Int J Fract. 1996;18(4):373–81.
[23] Yuan S, Zhang YG. Solution of eigenprobolems in ODEs by standard ODE solvers. Earthq Eng Eng Vib. 1993;3(2):94–102 (in Chinese).
[24] Xu YJ, Yuan S. Complete eigensolutions for anti-plane notches with multi-materials by super-inverse iteration. Acta Mehcanica Solida Sinica. 1997;10(2):157–66.
[25] Yuan S, Xu YJ. Complete Eigen-solutions for cracks by combination of imbedding method and Müller method. Chin J Comput Mech (suppl). 1997;14:639–42 (in Chinese).
[26] Maple 17, Maplesoft, a division of Waterloo Maple Inc. 1981–2013.