Engineering Analysis with Boundary Elements 128 (2021) 244–256

Contents lists available at ScienceDirect

Engineering Analysis with Boundary Elements

journal homepage: www.elsevier.com/locate/enganabound

Application of Sherman-Morrison formula in adaptive analysis by BEM

Pengfei Chai, Jianming Zhang*, Rui He, Weicheng Lin, Xiaomin Shu

State Key Laboratory of Advanced Design and Manufacturing for Vehicle Body, College of Mechanical and Vehicle Engineering, Hunan University, Changsha 410082, China

ARTICLE INFO

Keywords:
Adaptive analysis
Variable dimension system of linear equations
Sherman-Morrison formula
Mesh refinement

ABSTRACT

Mesh refinement may give rise to variable dimension system of linear equations in adaptive analysis. In each adaptive step, the matrix decomposition methods are adopted to solve linear algebraic equations, which is obviously time-consuming and computationally expensive for large-scale problems. A new algorithm for solving a variable dimension system of linear equations is proposed in this paper, which combines matrix deformation with the Sherman-Morrison formula. This method not only retains the advantages of the Sherman-Morrison formula, but also is particularly suitable for adaptive analysis. Compared with the matrix decomposition method, our method has outstanding performance in promoting computational efficiency. Several numerical examples are used to illustrate the superiority of the proposed method when applying to solve 2D elasticity problems.

1. Introduction

Adaptive analysis is widely adopted by some numerical methods to solve computational engineering problems, such as in finite element method (FEM) [1–4] and boundary element method (BEM) [5–7]. Mesh refinement is an effective measure in adaptive analysis to achieve more accurate solution. Adaptive schemes, such as the h-refinement and the p-refinement, may give rise to variable dimension system of linear equations. In general, conventional methods, such as the Guassian elimination method, matrix decomposition methods and iteration techniques etc., are often employed to solve linear equation in each adaptive step. While, those methods usually suited for solving fixed-order linear equations, which would involve repetitive computation in adaptive analysis. To avoid redundant work, some researchers directly dedicate to solving variable dimension linear equations. Bisschop and Meeraus [8, 9] proposed the augmented matrix method. According to the matrix augmentation form and the Schur complement, Zhou et al. [10] presented a method which fitted for the perturbation occurred on the rim of the matrix. While for arbitrary perturbation in the matrix, this method seems to be non-universal.

Sherman-Morrison and Woodbury formulas (SMW) [11–15] are very efficient in solving matrix inversion with small rank perturbation arisen in anywhere of the original matrix. SMW emanated from studies of partitioned matrices, which provided an expression for the inverse of Schur complement [20]. As the special form, the Capacitance Matrix usually presented to computational engineering, such as in 3D linear elasticity problem [17] and linear elastic contact problems [18]. On account of the cumulative error, algorithms concerning SMW were usually confined to small rank perturbation case. So, Powell [16] proposed a new form of Sherman-Morrison formula through introducing the original matrix and successfully reduced the cumulative error, but the CPU time increased. Despite the existence of the restriction, however, SMW were still significant in practical application during the recent decades. In the context of modern structural analysis, it is possible to show that all kind of reanalysis methods are variations on SMW [19], these methods have been used as efficient tools in exact structural reanalysis relating to low-rank modifications [21–25]. Meanwhile, SMW were developed in the dynamic problems. In this field, Sherman–Morrison and Woodbury formulas were used to calculate the optimal absorber parameter [26] and estimated the corrected frequency response [27]. Additionally, the SMW were also used in the multidomain problems [28] to obtain the inverse of the hierarchical off-diagonal low-rank matrix to accelerate the solution.

However, previous researches about Sherman-Morrison formula are often focused on matrix with fixed-order. In this paper, we propose a variable dimension matrix inversion method which takes full use of the advantages of the Sherman-Morrison formula. This method is performed on the basis of the boundary face method (BFM) developed by Zhang, as reference [29–33]. In adaptive analysis, when mesh refinement is implemented, we only need to take advantage of the coefficient matrix and its inverse of last adaptive step to solve the system of linear algebraic equations in current step through using the Sherman-Morrison formula twice. This method has been implemented successfully combined with LU decomposition method in adaptive analysis (hereinafter our algorithm) and has compared with pure adaptive algorithm with LU decomposition method (hereinafter original algorithm).

* Corresponding author.
E-mail address: zhangjm@hnu.edu.cn (J. Zhang).

https://doi.org/10.1016/j.enganabound.2021.04.009
Received 25 September 2020; Received in revised form 3 April 2021; Accepted 12 April 2021
Available online 22 April 2021

Our paper is organized as follow: The boundary integral equation is briefly described in Section 2. Section 3 gives a brief introduction about the Sherman-Morrison formula. The detail of our algorithm is explained in Section 4, and the validity is illustrated through four numerical examples in Section 5. The paper ends with discussion and conclusions in Section 6.

2. The boundary integral equation

Considering 2D elasticity body in absence of body force, the boundary integral equation can be written in the following form

$$c_{ij}(P)u_j(P) = \int_\Gamma U_{ij}(P,Q)t_j(Q)d\Gamma(Q) - \int_\Gamma T_{ij}(P,Q)u_j(Q)d\Gamma(Q), \quad P,Q \in \Gamma, \tag{1}$$

where u_j and t_j (j = 1, 2) are the displacement and traction components, respectively. $c_{ij}(P)$ is the coefficient matrix of the free term. For 2D plane-strain problems:

$$U_{ij}(P,Q) = \frac{1}{8\pi G(1-\upsilon)}\left[(3-4\upsilon)\delta_{ij}\ln\frac{1}{r} + r_{,i}r_{,j}\right] \tag{2}$$

$$T_{ij}(P,Q) = -\frac{1}{4\pi(1-\upsilon)r}\left\{\frac{\partial r}{\partial n}\left[(1-2\upsilon)\delta_{ij} + 2r_{,i}r_{,j}\right] - (1-2\upsilon)(r_{,i}n_j - r_{,j}n_i)\right\} \tag{3}$$

where r is the distance between the source node P and the field point Q; n is the outward normal at point Q; G and ν are the shear modulus and the Poisson's ratio, respectively.

Through reassembling the linear equations obtained by discretizing the BIE Eq. (1), we get the equation

$$\sum_{j=i}^{N}\mathbf{H}_{ij}\mathbf{u}_j = \sum_{j=i}^{N}\mathbf{G}_{ij}\mathbf{t}_j \tag{4}$$

According to the boundary condition of displacement and surface force of the source point, the known part is moved to right sides, and the unknowns to left, Eq. (4) could be re-written as matrix form

$$\mathbf{Ax} = \mathbf{b} \tag{5}$$

where $\mathbf{A}$ is the system coefficient matrix, $\mathbf{b}$ is the right hand vector.

3. A brief introduction of Sherman–Morrison formula

Sherman-Morrison formula has been revealed in [16,17] to obtain the inverse of a matrix with some entries modified. Considering a row or column perturbation of the coefficient matrix $\mathbf{A}$ in Eq. (5), the modified coefficient matrix is written as the following form$\mathbf{A} + \mathbf{uv}^T$. From the Sherman-Morrison formula we can know

$$(\mathbf{A}+\mathbf{uv}^T)^{-1} = \mathbf{A}^{-1} - \frac{(\mathbf{A}^{-1}\cdot\mathbf{u})\otimes(\mathbf{v}\cdot\mathbf{A}^{-1})}{1+\lambda} \tag{6}$$

$$\lambda = \mathbf{v}\cdot\mathbf{A}^{-1}\cdot\mathbf{u} \tag{7}$$

$$\mathbf{z} = \mathbf{A}^{-1}\cdot\mathbf{u} \tag{8}$$

$$\mathbf{w} = (\mathbf{A}^{-1})^T\cdot\mathbf{v} \tag{9}$$

According to Eq. (8), Eq. (7) can be written as:

$$\lambda = \mathbf{v}\cdot\mathbf{z} \tag{10}$$

From Eqs. (8) to (10), we can write Eq. (6) as:

$$(\mathbf{A}+\mathbf{uv}^T)^{-1} = \mathbf{A}^{-1} - \frac{\mathbf{z}\otimes\mathbf{w}}{1+\lambda} \tag{11}$$

where **u, v, w, z** are vectors,$\lambda$is scalar. When matrix $\mathbf{A}$ has a corrections in ith row, $\mathbf{u}$ is the unite vector from ith column of identity matrix, and $\mathbf{v}$ is the difference between the row corrections and the ith row of matrix $\mathbf{A}$. When matrix $\mathbf{A}$ has a perturbation in jth column, $\mathbf{v}$ is the unite vector from jth row of identity matrix, and $\mathbf{u}$ is the difference between the column corrections and the jth column from matrix $\mathbf{A}$.

The above example is for single row or single column perturbation. When a matrix is modified by p rank 1corrections, $\mathbf{U}$ and $\mathbf{V}$ are matrices composed of corresponding vectors, the modified coefficient matrix is written as the following form$\mathbf{A} + \mathbf{UV}^T$. Assume that matrix $\mathbf{A}$ is$2N \times 2N$, $\mathbf{U}$ and $\mathbf{V}$ are$2N \times P, P \ll 2N$,we can recycle Eq. (11) P times. where

$$\mathbf{U} = \begin{bmatrix} \mathbf{u}_1 & \cdots & \mathbf{u}_p \end{bmatrix}, \mathbf{V} = \begin{bmatrix} \mathbf{v}_1 & \cdots & \mathbf{v}_p \end{bmatrix} \tag{12}$$

The modified matrix can be written as the following form

$$\mathbf{A}+\mathbf{UV}^T = \left(\mathbf{A} + \sum_{k=1}^{P}\mathbf{u}_k\otimes\mathbf{v}_k\right) \tag{13}$$

And the inverse of the modified matrix can be written as

$$(\mathbf{A}+\mathbf{UV}^T)^{-1} = \mathbf{A}^{-1} - \sum_{k=1}^{p}\frac{\mathbf{z}_k\otimes\mathbf{w}_k}{1+\lambda_k} \tag{14}$$

4. Implementation of our algorithm

4.1. Process of adaptive analysis

The geometric model of 2D elasticity body with the concentrated load (Fig. 1(a)) is used to perform adaptive analysis. Assuming that there are n elements (constant element) at the corner E will be implemented mesh refinement (Fig. 1(b)) in last adaptive step. In current adaptive step, the new added mesh can be seen in Fig. 1(c). The Mises stress of each element will be computed under the meshes of last adaptive step and current adaptive step, respectively. Where the relative error of one element, between current and last step, out of the limitation will be implemented mesh refinement in current adaptive step. Subsequently, it would use our method to solve linear algebraic equation when the number of new added source points n less than the limitation. On the contrary, it would use original method. The adaptive will be terminated if there are no elements satisfying the mesh refinement condition.

The flow chart of our algorithm is shown in Fig. 2. In the flow chart, "i" represents the index of the element; "j" represents the number of elements needing to implement mesh refinement next adaptive step; "$\bar{\eta}_1$" is the limitation of the element relative error; "$\bar{\eta}_2$" is the limit number of elements to determine whether our method should be chosen or not next adaptive step.

4.2. Variable order matrix inversion method

Our method for solving linear equation can work, when less mesh are implemented mesh refinement. As shown in Fig. 1, the number of source points is N in last adaptive step (see Fig. 1(b)), the number of new added source points is n (where $n << N$) in current adaptive step (see Fig. 1(c)). $2n$ rows and $2n$ columns in matrices $\mathbf{A}$ ($2N \times 2N$) (last adaptive step) will disappear, these can be represented as vectors $\boldsymbol{i}$ + **1**, …, $\boldsymbol{i}$ + **2n** and $\boldsymbol{j}$ + **1**, …, $\boldsymbol{j}$ + **2n** ($j = i$); $4n$ new rows and columns in $\boldsymbol{H}$,$\mathbf{G}$matrices ($2(N+n) \times 2(N+n)$)(current adaptive step) will be added, and represented as $\mathbf{r}_{i+1}$, …, $\mathbf{r}_{i+2n}$, $\mathbf{r}_{i+2n+1}$, …, $\mathbf{r}_{i+4n}$ and $\mathbf{c}_{j+1}$, …, $\mathbf{c}_{j+2n}$,

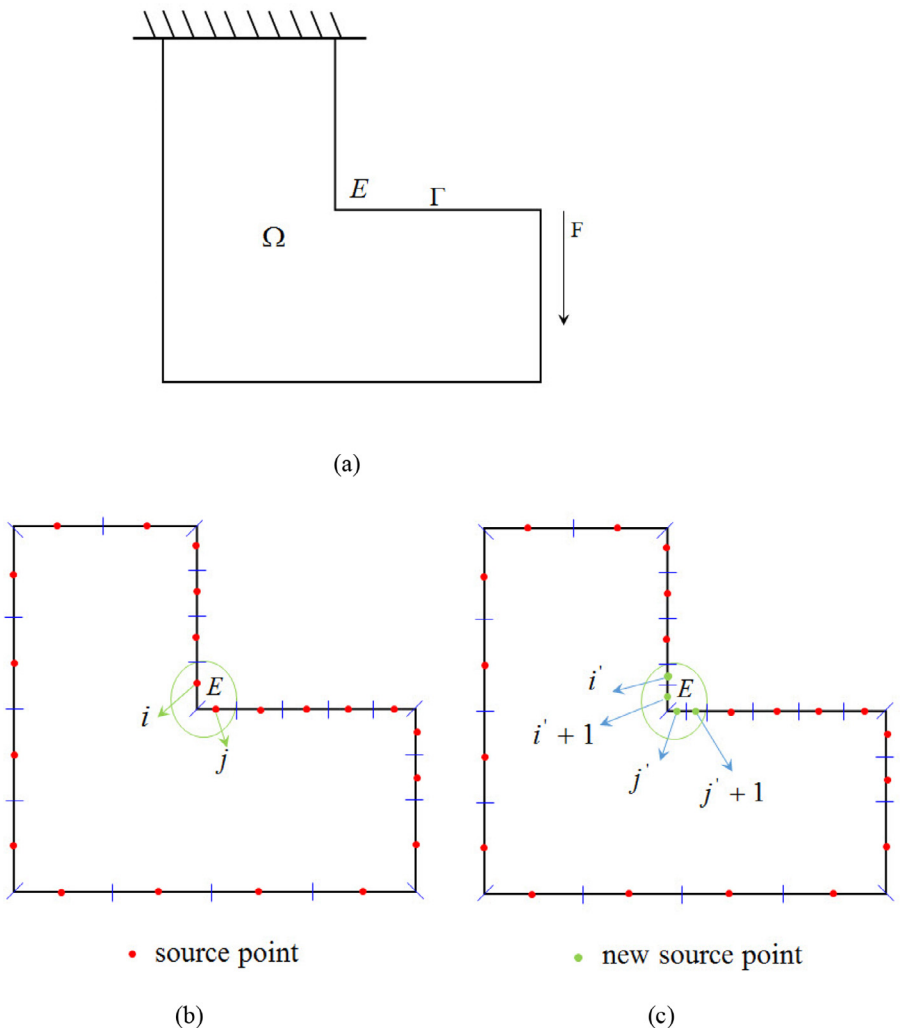

Fig. 1. 2D elasticity problem: (a) geometric model, (b) Last adaptive step, (c) current adaptive step, where *i*, *j*represent the *i* th and *j*th original elements; i',$i'+1$, j', $j'+1$ represent new added elements.

$\mathbf{c}_{j+2n+1}$, …, $\mathbf{c}_{j+4n}$ (Each vector with $2(N+n)$ elements), where

$$
\begin{aligned}
\mathbf{i+1} &= \begin{bmatrix} A_{i+1,1} & \cdots & A_{i+1,2N} \end{bmatrix}^T,\\
\mathbf{i+2n} &= \begin{bmatrix} A_{i+2n,1} & \cdots & A_{i+2n,2N} \end{bmatrix}^T,\\
\mathbf{j+1} &= \begin{bmatrix} A_{1,j+1} & \cdots & A_{2N,j+1} \end{bmatrix}^T,\\
\mathbf{j+2n} &= \begin{bmatrix} A_{1,j+2n} & \cdots & A_{2N,j+2n} \end{bmatrix}^T,\\
\mathbf{r}_{i+1} &= \begin{bmatrix} H_{i+1,1} & \cdots & H_{i+1,i+2n} & \cdots & H_{i+1,i+4n} & \cdots & H_{i+1,2(N+n)} \end{bmatrix}^T,\\
&or\\
\mathbf{r}_{i+1} &= \begin{bmatrix} G_{i+1,1} & \cdots & G_{i+1,i+2n} & \cdots & G_{i+1,i+4n} & \cdots & G_{i+1,2(N+n)} \end{bmatrix}^T,\\
&\vdots\\
\mathbf{r}_{i+4n} &= \begin{bmatrix} H_{i+4n,1} & \cdots & H_{i+4n,i+2n} & \cdots & H_{i+4n,i+4n} & \cdots & H_{i+4n,2(N+n)} \end{bmatrix}^T,\\
&or\\
\mathbf{r}_{i+4n} &= \begin{bmatrix} G_{i+4n,1} & \cdots & G_{i+4n,i+2n} & \cdots & G_{i+4n,i+4n} & \cdots & G_{i+4n,2(N+n)} \end{bmatrix}^T,\\
\mathbf{c}_{j+1} &= \begin{bmatrix} H_{1,j+1} & \cdots & H_{j+2n,j+1} & \cdots & H_{j+4n,j+1} & \cdots & H_{2(N+n),j+1} \end{bmatrix}^T,\\
&or\\
\mathbf{c}_{j+1} &= \begin{bmatrix} G_{1,j+1} & \cdots & G_{j+2n,j+1} & \cdots & G_{j+4n,j+1} & \cdots & G_{2(N+n),j+1} \end{bmatrix}^T,\\
&\vdots\\
\mathbf{c}_{j+4n} &= \begin{bmatrix} H_{1,j+4n} & \cdots & H_{j+2n,j+4n} & \cdots & H_{j+4n,j+4n} & \cdots & H_{2(N+n),j+4n} \end{bmatrix}^T,\\
&or\\
\mathbf{c}_{j+4n} &= \begin{bmatrix} G_{1,j+4n} & \cdots & G_{j+2n,j+4n} & \cdots & G_{j+4n,j+4n} & \cdots & G_{2(N+n),j+4n} \end{bmatrix}^T.
\end{aligned}
\tag{15}
$$

In the following, those vectors above will be used to get the inverse of variable order matrix. The process is divided into three steps as shown in Table 1.

Step 1. Get the inverse of matrix $\tilde{\mathbf{A}}$

$$
\mathbf{A}\begin{matrix} \mathbf{i+1} \overset{replaced\,by}{\rightarrow} \bar{\mathbf{r}}_{i+1} \\ \vdots \\ \mathbf{i+2n} \overset{replaced\,by}{\rightarrow} \bar{\mathbf{r}}_{i+2n} \\ \mathbf{j+1} \overset{replaced\,by}{\rightarrow} \bar{\mathbf{c}}_{j+1} \\ \vdots \\ \mathbf{j+2n} \overset{replaced\,by}{\rightarrow} \bar{\mathbf{c}}_{j+2n} \end{matrix}\tilde{\mathbf{A}} \tag{16}
$$

According the boundary condition, matrix$\tilde{\mathbf{A}}(2N\times 2N)$ is constructed through replacing the rows $\boldsymbol{i+1}$, …, $\boldsymbol{i+2n}$ and columns $\boldsymbol{j+1}$, …, $\boldsymbol{j+2n}$ in matrix $\mathbf{A}$ by rows $\bar{\mathbf{r}}_{i+1}$, …, $\bar{\mathbf{r}}_{i+2n}$ (modified by $\mathbf{r}_{i+1}$, …, $\mathbf{r}_{i+2n}$) and columns $\bar{\mathbf{c}}_{j+1}$, …, $\bar{\mathbf{c}}_{j+2n}$ (modified by $\mathbf{c}_{j+1}$, …, $\mathbf{c}_{j+2n}$), where

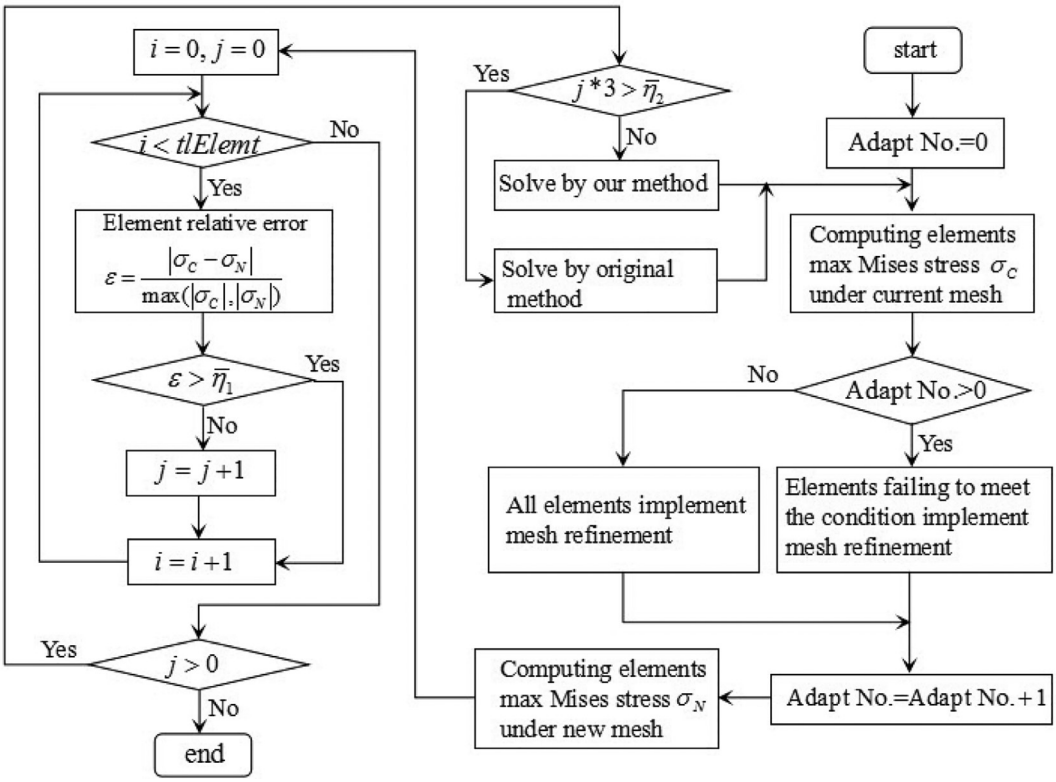

Fig. 2. Flow chart of our algorithm.

Table 1
Process of variable order matrix inversion.

Step 1: get inverse of matrix $\tilde{\mathbf{A}}$using the Sherman-Morrison formula.
Step 2: get inverse of matrix **M**.
Step 3: get the inverse of matrix $\tilde{\mathbf{M}}$using the Sherman-Morrison formula.
where matrix $\tilde{\mathbf{A}}$ is the deformation of matrix **A**. Matrix **M** is block matrix composed of $\tilde{\mathbf{A}}$ and identity matrix **I**. Matrix $\tilde{\mathbf{M}}$ is the deformation of matrix **M**.

$$
\begin{aligned}
&\bar{\mathbf{r}}_{i+1}=\begin{bmatrix} \mathrm{H}_{i+1,1} & \cdots & \mathrm{H}_{i+1,i+2n} & \mathrm{H}_{i+1,i+4n+1} & \cdots & \mathrm{H}_{i+1,2(N+n)} \end{bmatrix}^T,\\
&or\\
&\bar{\mathbf{r}}_{i+1}=\begin{bmatrix} \mathrm{G}_{i+1,1} & \cdots & \mathrm{G}_{i+1,i+2n} & \mathrm{G}_{i+1,i+4n+1} & \cdots & \mathrm{G}_{i+1,2(N+n)} \end{bmatrix}^T,\\
&\vdots\\
&\bar{\mathbf{r}}_{i+2n}=\begin{bmatrix} \mathrm{H}_{i+2n,1} & \cdots & \mathrm{H}_{i+2n,i+2n} & \mathrm{H}_{i+2n,i+4n+1} & \cdots & \mathrm{H}_{i+2n,2(N+n)} \end{bmatrix}^T,\\
&or\\
&\bar{\mathbf{r}}_{i+2n}=\begin{bmatrix} \mathrm{G}_{i+2n,1} & \cdots & \mathrm{G}_{i+2n,i+2n} & \mathrm{G}_{i+2n,i+4n+1} & \cdots & \mathrm{G}_{i+2n,2(N+n)} \end{bmatrix}^T,\\
&\bar{\mathbf{c}}_{j+1}=\begin{bmatrix} \mathrm{H}_{1,j+1} & \cdots & \mathrm{H}_{j-1,j+1} & 0 & \mathrm{H}_{j+1,j+1} & \cdots & \mathrm{H}_{j+2n,j+1} & \mathrm{H}_{j+4n+1,j+1} & \cdots & \mathrm{H}_{2(N+n),j+1} \end{bmatrix}^T,\\
&or\\
&\bar{\mathbf{c}}_{j+1}=\begin{bmatrix} \mathrm{G}_{1,j+1} & \cdots & \mathrm{G}_{j-1,j+1} & 0 & \mathrm{G}_{j+1,j+1} & \cdots & \mathrm{G}_{j+2n,j+1} & \mathrm{G}_{j+4n+1,j+1} & \cdots & \mathrm{G}_{2(N+n),j+1} \end{bmatrix}^T,\\
&\vdots\\
&\bar{\mathbf{c}}_{j+2n}=\begin{bmatrix} \mathrm{H}_{1,j+2n} & \cdots & \mathrm{H}_{j+2n-1,j+2n} & 0 & \mathrm{H}_{j+4n+1,j+2n} & \cdots & \mathrm{H}_{2(N+n),j+2n} \end{bmatrix}^T,\\
&or\\
&\bar{\mathbf{c}}_{j+2n}=\begin{bmatrix} \mathrm{G}_{1,j+2n} & \cdots & \mathrm{G}_{j+2n-1,j+2n} & 0 & \mathrm{G}_{j+4n+1,j+2n} & \cdots & \mathrm{G}_{2(N+n),j+2n} \end{bmatrix}^T.
\end{aligned}
\tag{17}
$$

Each vector has 2 N elements.
Then we can get

$$
\begin{aligned}
\mathbf{U}_1 &= \begin{bmatrix} \mathbf{u}_{1,1} & \cdots & \mathbf{u}_{1,2n} & \mathbf{u}_{1,2n+1} & \cdots & \mathbf{u}_{1,4n} \end{bmatrix}\\
&= \begin{bmatrix} \mathbf{e}_{i+1} & \cdots & \mathbf{e}_{i+2n} & \bar{\mathbf{c}}_{j+1}-(\mathbf{j}+1) & \cdots & \bar{\mathbf{c}}_{j+2n}-(\mathbf{j}+\mathbf{2n}) \end{bmatrix},\\
\mathbf{V}_1 &= \begin{bmatrix} \mathbf{v}_{1,1} & \cdots & \mathbf{v}_{1,2n} & \mathbf{v}_{1,2n+1} & \cdots & \mathbf{v}_{1,4n} \end{bmatrix}\\
&= \begin{bmatrix} \bar{\mathbf{r}}_{i+1}-(\mathbf{i}+1) & \cdots & \bar{\mathbf{r}}_{i+2n}-(\mathbf{i}+\mathbf{2n}) & \mathbf{e}_{j+1} & \cdots & \mathbf{e}_{j+2n} \end{bmatrix}.
\end{aligned}
\tag{18}
$$

$$
\begin{aligned}
&\mathbf{e}_{i+1}=\begin{bmatrix} 0 & \cdots & 1 & \cdots & 0 \end{bmatrix}^{\mathrm{T}}(i+1thelementis1),\\
&\vdots\\
&\mathbf{e}_{i+2n}=\begin{bmatrix} 0 & \cdots & 1 & \cdots & 0 \end{bmatrix}^{\mathrm{T}}(i+2nthelementis1).
\end{aligned}
\tag{19}
$$

$\mathbf{e}_{i+1}$, …, $\mathbf{e}_{i+2n}$, $\mathbf{e}_{j+1}$, …, $\mathbf{e}_{j+2n}$ (where $i=j$, each vector with 2 N elements) are unit vectors from identity matrix, respectively.

Using Eqs. (14) and (18) we can obtain

$$\tilde{\mathbf{A}}^{-1}=(\mathbf{A}+\mathbf{U}_1\mathbf{V}_1{}^T)^{-1}=\mathbf{A}^{-1}-\sum_{k=0}^{2n}\frac{\mathbf{z}_k\otimes\mathbf{w}_k}{1+\lambda_k}. \tag{20}$$

Step 2. Get the inverse of matrix $\mathbf{M}$

The block matrix $\mathbf{M}$ $(2(N+n)\times 2(N+n)$ is composed of matrix$\tilde{\mathbf{A}}$and identity matrix $\mathbf{I}$ $(2n\times 2n)$. The inverse of the block matrix $\mathbf{M}$ can be directly assembled by the inverse of matrix $\tilde{\mathbf{A}}$(see Eq. (21)).

$$\mathbf{M}=\begin{bmatrix}\tilde{\mathbf{A}} & \mathbf{0}\\ \mathbf{0} & \mathbf{I}\end{bmatrix},\ \mathbf{M}^{-1}=\begin{bmatrix}\tilde{\mathbf{A}}^{-1} & \mathbf{0}\\ \mathbf{0} & \mathbf{I}\end{bmatrix}. \tag{21}$$

Step 3. Get the inverse of matrix $\bar{\mathbf{M}}$

Matrix $\bar{\mathbf{M}}(2(N+n)\times 2(N+n))$ is obtained through fill in the null block and the identity part of $\mathbf{M}$ by rows $\tilde{\mathbf{r}}_{i+2n+1}$, …, $\tilde{\mathbf{r}}_{i+4n}$(modified by $\mathbf{r}_{i+2n+1}$, …, $\mathbf{r}_{4n}$) and columns $\tilde{\mathbf{c}}_{j+2n+1}$, …, $\tilde{\mathbf{c}}_{j+4n}$ (modified by $\mathbf{c}_{j+2n+1}$, …, $\mathbf{c}_{4n}$), respectively, where

$$\begin{aligned}
&\tilde{\mathbf{r}}_{i+2n+1}=\begin{bmatrix}\mathrm{H}_{i+2n+1,1} & \cdots & \mathrm{H}_{i+2n+1,i+2n} & \mathrm{H}_{i+2n+1,i+4n+1} & \cdots & \mathrm{H}_{i+2n+1,2(N+n)} & \mathrm{H}_{i+2n+1,i+2n+1} & \cdots & \mathrm{H}_{i+2n+1,i+4n}\end{bmatrix}^T,\\
&or\\
&\tilde{\mathbf{r}}_{i+2n+1}=\begin{bmatrix}\mathrm{G}_{i+2n+1,1} & \cdots & \mathrm{G}_{i+2n+1,i+2n} & \mathrm{G}_{i+2n+1,i+4n+1} & \cdots & \mathrm{G}_{i+2n+1,2(N+n)} & \mathrm{G}_{i+2n+1,i+2n+1} & \cdots & \mathrm{G}_{i+2n+1,i+4n}\end{bmatrix}^T,\\
&\vdots\\
&\tilde{\mathbf{r}}_{i+4n}=\begin{bmatrix}\mathrm{H}_{i+4n,1} & \cdots & \mathrm{H}_{i+4n,i+2n} & \mathrm{H}_{i+4n,i+4n+1} & \cdots & \mathrm{H}_{i+4n,2(N+n)} & \mathrm{H}_{i+4n,i+2n+1} & \cdots & \mathrm{H}_{i+4n,i+4n}\end{bmatrix}^T,\\
&or\\
&\tilde{\mathbf{r}}_{i+4n}=\begin{bmatrix}\mathrm{G}_{i+4n,1} & \cdots & \mathrm{G}_{i+4n,i+2n} & \mathrm{G}_{i+4n,i+4n+1} & \cdots & \mathrm{G}_{i+4n,2(N+n)} & \mathrm{G}_{i+4n,i+2n+1} & \cdots & \mathrm{G}_{i+4n,i+4n}\end{bmatrix}^T,\\
&\tilde{\mathbf{c}}_{j+2n+1}=\begin{bmatrix}\mathrm{H}_{1,j+2n+1} & \cdots & \mathrm{H}_{j+2n,j+2n+1} & \mathrm{H}_{j+4n+1,j+2n+1} & \cdots & \mathrm{H}_{2(N+n),j+2n+1} & 0 & \cdots & -1 & \cdots & 0\end{bmatrix}^T,\\
&or\\
&\tilde{\mathbf{c}}_{j+2n+1}=\begin{bmatrix}\mathrm{G}_{1,j+2n+1} & \cdots & \mathrm{G}_{j+2n,j+2n+1} & \mathrm{G}_{j+4n+1,j+2n+1} & \cdots & \mathrm{G}_{2(N+n),j+2n+1} & 0 & \cdots & -1 & \cdots & 0\end{bmatrix}^T\\
&(2N+1 the lementis-1),\\
&\vdots\\
&\tilde{\mathbf{c}}_{j+4n}=\begin{bmatrix}\mathrm{H}_{1,j+4n} & \cdots & \mathrm{H}_{j+2n,j+4n} & \mathrm{H}_{j+4n+1,j+4n} & \cdots & \mathrm{H}_{2(N+n),j+4n} & 0 & \cdots & -1\end{bmatrix}^T,\\
&or\\
&\tilde{\mathbf{c}}_{j+4n}=\begin{bmatrix}\mathrm{G}_{1,j+4n} & \cdots & \mathrm{G}_{j+2n,j+4n} & \mathrm{G}_{j+4n+1,j+4n} & \cdots & \mathrm{G}_{2(N+n),j+4n} & 0 & \cdots & -1\end{bmatrix}^T\\
&(2(N+n) the lementis-1).
\end{aligned} \tag{22}$$

Each vector has $2(N+n)$ elements.

Then we can get

$$\begin{aligned}
\mathbf{U}_2&=\begin{bmatrix}\mathbf{u}_{2,1} & \cdots & \mathbf{u}_{2,2n} & \mathbf{u}_{2,2n+1} & \cdots & \mathbf{u}_{2,4n}\end{bmatrix}\\
&=\begin{bmatrix}\mathbf{e}_{2N+1} & \cdots & \mathbf{e}_{2(N+n)} & \tilde{\mathbf{c}}_{j+2n+1} & \cdots & \tilde{\mathbf{c}}_{j+4n}\end{bmatrix},\\
\mathbf{V}_2&=\begin{bmatrix}\mathbf{v}_{2,1} & \cdots & \mathbf{v}_{2,2n} & \mathbf{v}_{2,2n+1} & \cdots & \mathbf{v}_{2,4n}\end{bmatrix}\\
&=\begin{bmatrix}\tilde{\mathbf{r}}_{i+2n+1} & \cdots & \tilde{\mathbf{r}}_{i+4n} & \mathbf{e}_{2N+1} & \cdots & \mathbf{e}_{2(N+n)}\end{bmatrix},
\end{aligned} \tag{23}$$

where

$$\begin{aligned}
&\mathbf{e}_{2N+1}=\begin{bmatrix}0 & \cdots & 1 & \cdots & 0\end{bmatrix}^{\mathrm{T}}(2N+1 the lementis1),\\
&\vdots\\
&\mathbf{e}_{2(N+n)}=\begin{bmatrix}0 & \cdots & 1\end{bmatrix}^{\mathrm{T}}(2(N+n) the lementis1).
\end{aligned} \tag{24}$$

$\mathbf{e}_{2N+1}$, …, $\mathbf{e}_{2(N+n)}$(each vector with $2(N+n)$ elements) represent unit vectors, respectively.

Combining Eqs. (14) and (23) we can obtain

$$\bar{\mathbf{M}}^{-1}=(\mathbf{M}+\mathbf{U}_2\mathbf{V}_2{}^T)^{-1}=\mathbf{M}^{-1}-\sum_{k=0}^{2n}\frac{\mathbf{z}_k\otimes\mathbf{w}_k}{1+\lambda_k}. \tag{25}$$

When mesh refinement are implemented multiple times, we can also get the inverse of the deformed matrix$\bar{\mathbf{M}}$through repeating above steps.

5. Numerical examples

In this part, four numerical examples are presented to highlight the benefits of the proposed method in adaptive analysis. To survey the convergence, in Sections 5.1 and 5.2, the variables are approximated by quadratic and linear elements, respectively. Simultaneously, two real engineering background examples with stress concentration feature demonstrate the general applicability, computational efficiency for solving 2D elasticity problems in Sections 5.3 and 5.4, while the fourth example in Section 5.4 reveals the ability of the proposed method to handle complex geometry.

For the purpose of error estimation and convergence study, the relative error is defined as:

$$error=\frac{1}{|v|_{\max}}\sqrt{\frac{1}{M}\sum_{i=1}^{M}\left[v_i^{(e)}-v_i^{(n)}\right]^2}, \tag{26}$$

where $|v|_{\max}$is the maximum value of Mises stress, displacement u and traction t over M sample points, and the superscripts (e) and (n) refer to the exact and numerical solutions, respectively.

In the following numerical examples, "Our", "Original" and "Exact" in the figures represent our algorithm, original algorithm and analytical solution, respectively. "*NS*" and "*NE*" are the number of the source nodes and elements. "Time" is the CPU time spent on each adaptive step. "*Err_t*$_1$", "*Err_u*$_2$", and "*Err_u*$_r$" denote the relative error of t_1, u_2 and u_r. "*Err_*σ_r" and "*Err_*σ_θ" denote the relative error of $\sigma_r$and σ_θ. "Max_Mises" represents the max Mises stress in each adaptive step.

5.1. Analytical field problem

In the first example, analytical field problem is considered. The geometric model is shown in **Fig. 3**(a), the length of every edge $l=100$ mm. The assignment of initial elements could be seen in **Fig. 3**(b). Plane strain cases with Young's modulus $E=2000.0Mpa$ and Poisson's ratio $\nu=0.25$ are considered in this problem. The analytical solution is given by:

$$\begin{cases}U_{x_1}=-3{x_1}^2x_2+{x_2}^3\\ U_{x_2}=-{x_1}^3+3x_1{x_2}^2\end{cases}, \tag{27}$$

The physical variables are approximated by quadratic elements. The comparison of the computational time and the relative error between our and original algorithm are list in **Tables 2** and **3**. According to these tables, we can see that our method possesses higher computational efficiency than original method under the same adaptive process and the precision can be promoted following the increment of source point. The numerical results of our and original algorithm along edge *AB* are plotted in **Figs. 4** and **5**, which are same as the analytical solution. Through the information above, the reliability of our algorithm can be illustrated.

5.2. Thick-wall cylinder under internal pressure

The second example is a thick-walled cylinder (**Fig. 6**(a)) subjected to a uniform pressure $p=1.0$Mpa on the inner surface, the inner and outer radius of the cylinder are $a=100$ mm, $b=200$ mm. Due to the symmetry of the problem, only quarter of the structure is considered

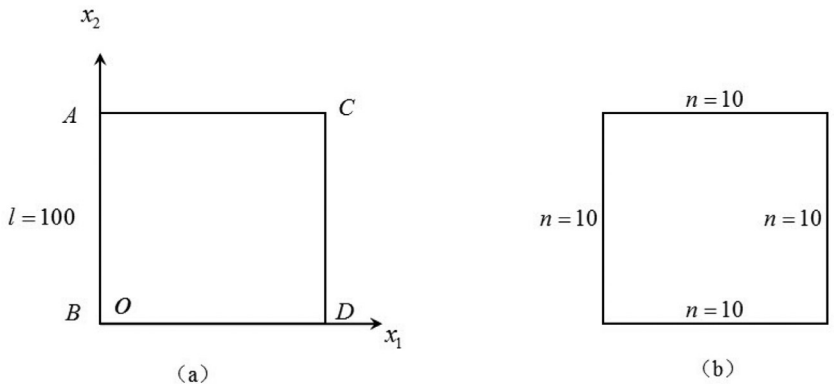

Fig. 3. Analytical field problem: (a) geometric model (b) initial elements.

Table 2
Time required for each adaptive step for analytical field problem.

Our algorithm				Original algorithm			
Adapt No.	NE	NS	Time(*s*)	Adapt No.	NE	NS	Time(*s*)
0	40	120	1.72	0	40	120	1.77
1	80	240	4.24	1	80	240	4.37
2	83	249	0.71	2	83	249	0.99
3	87	261	0.91	3	87	261	1.31
4	91	273	1.35	4	91	273	1.48
5	98	294	1.61	5	98	294	2.14

Table 3
Relative error of each adaptive step for analytical field problem.

Our algorithm	Adapt No.	0	1	2	3	4	5
	NE	40	80	83	87	91	98
	NS	120	240	249	261	273	294
	Err_u_1	1.64e-05	2.23e-06	2.21e-06	2.21e-06	2.20e-06	2.20e-06
	Err_t_2	2.49e-04	6.53e-05	6.73e-05	6.71e-05	6.71e-05	6.71e-05
Original algorithm	Adapt No.	0	1	2	3	4	5
	NE	40	80	83	87	91	98
	NS	120	240	249	261	273	294
	Err_u_1	1.64e-05	2.23e-06	2.21e-06	2.21e-06	2.20e-06	2.20e-06
	Err_t_2	2.49e-04	6.53e-05	6.73e-05	6.71e-05	6.71e-05	6.71e-05

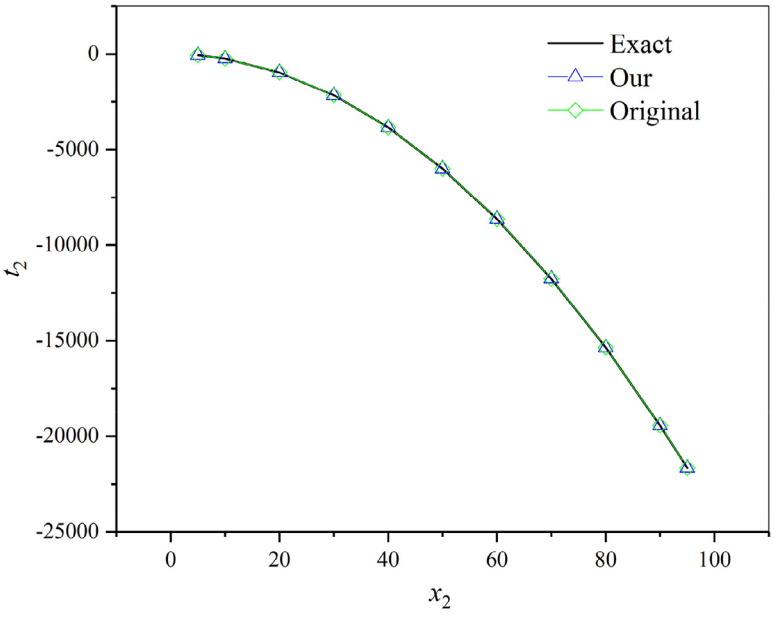

Fig. 4. Surface traction t_2 along the edge AB.

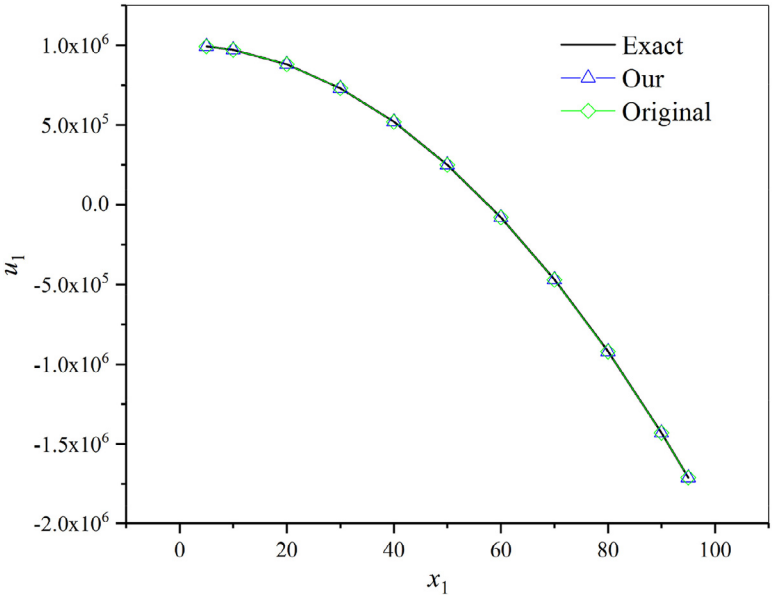

Fig. 5. Displacement u_1 along the edge AC.

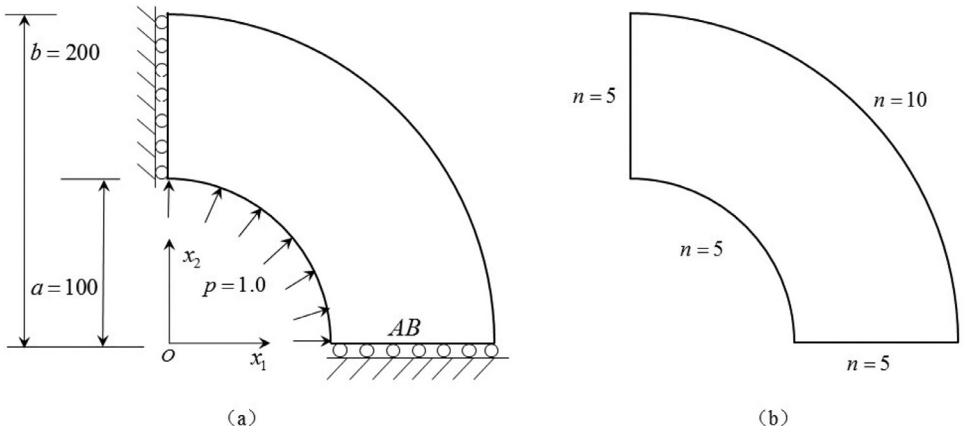

Fig. 6. Thick-wall cylinder under internal pressure: (a) geometric model of Thick-wall cylinder, (b) initial elements.

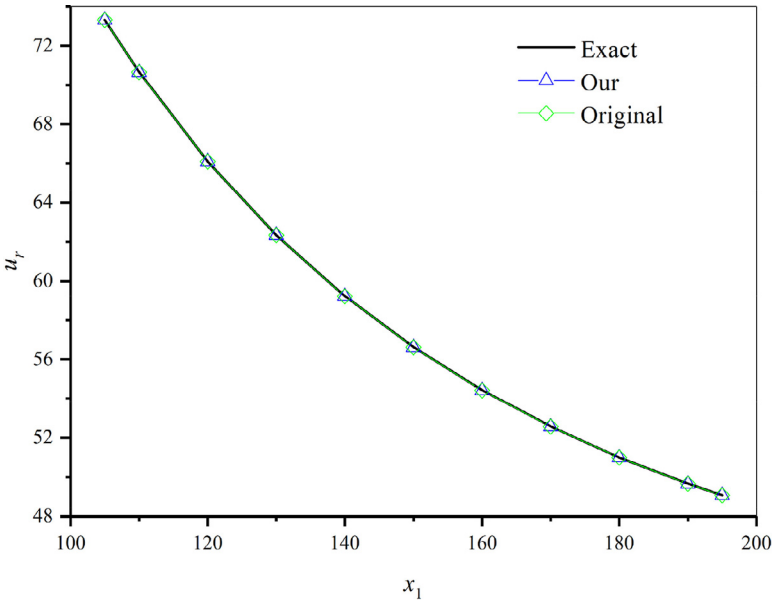

Fig. 7. Displacement$u_r$along the edge AB.

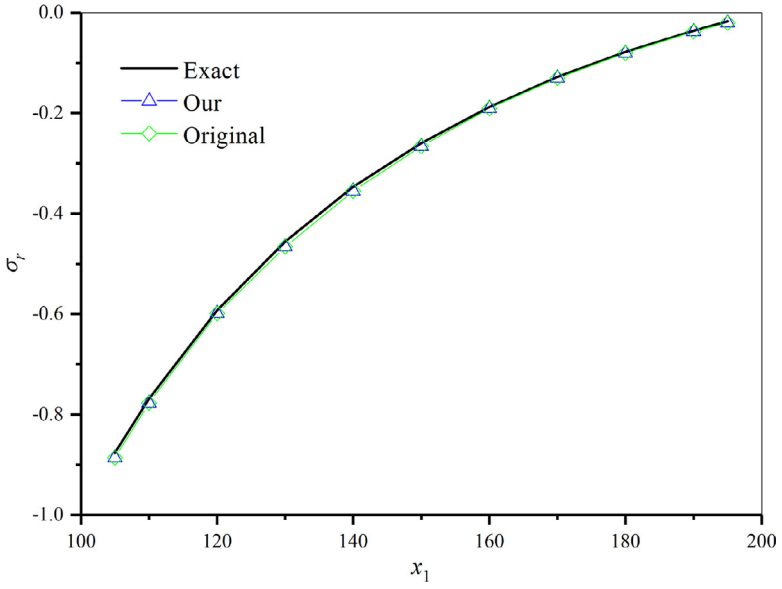

Fig. 8. Stressσ_r along the edge AB.

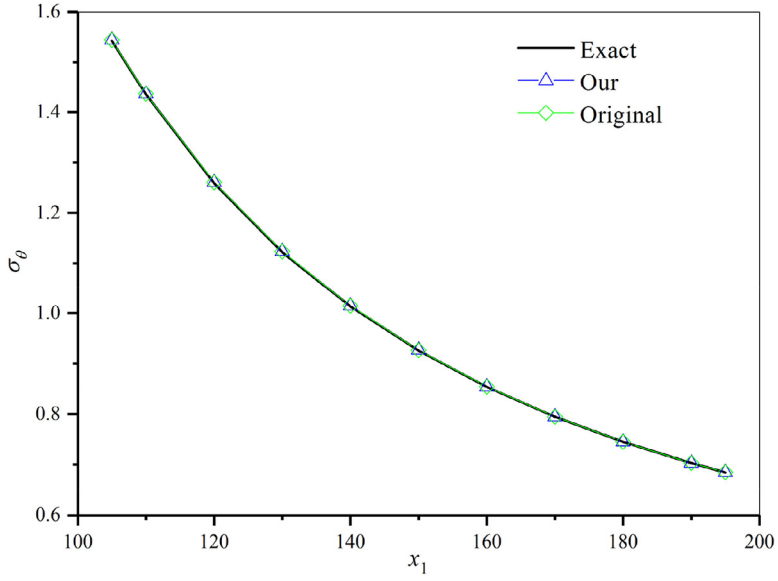

Fig. 9. Stressσ_θ along the edge AB.

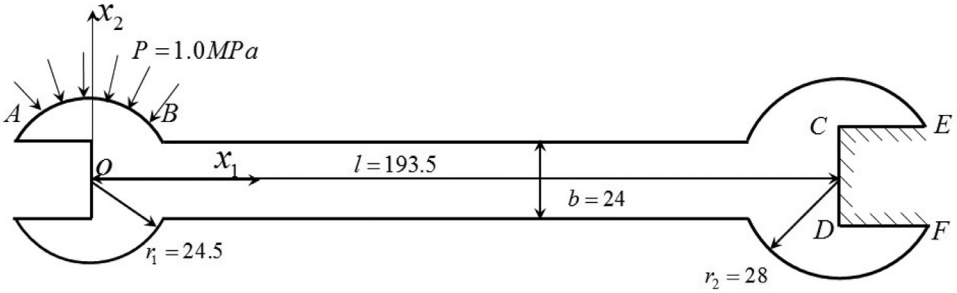

Fig. 10. Double open-end wrench: geometric model.

(Fig. 6(b)). Plane strain cases with Young's modulus $E = 2.5$Mpa and the Poisson's ratio $\nu = 0.25$ are considered in this problem. The analytical solution for this problem is given by:

$$\begin{cases} \sigma_r = \frac{a^2 p}{b^2-a^2}\left(1-\frac{b^2}{r^2}\right) \\ \sigma_\theta = \frac{a^2 p}{b^2-a^2}\left(1+\frac{b^2}{r^2}\right) \\ u_r = \frac{(1+v)}{E}\frac{a^2 p}{b^2-a^2}\left[(1-2v)r+\frac{b^2}{r}\right] \end{cases} \tag{28}$$

The physical variables are approximated by linear elements. Initial elements are shown in Fig. 6(b). Table 4 gives the computational time. The relative error of u_r,σ_r and σ_θ is list in Table 5. Apparently, our method takes less time than original algorithm under the same precision. Numerical results of displacement u_r, stress σ_r and σ_θ for our method and original algorithm along edge AB with 350 elements (700 source nodes) are plotted in Figs. 7–9. The numerical results can fully reveal that our algorithm is convergent with low-order interpolation elements.

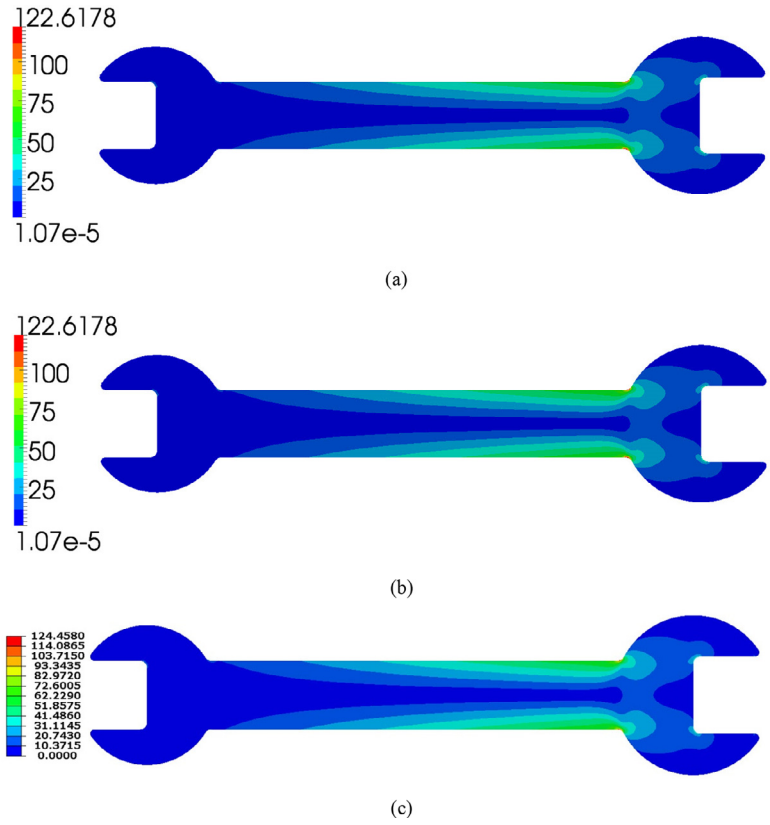

Fig. 11. Von-Mises stress for double open-end wrench: (a) our method: with 382 elements, 1146 source nodes, (b) original method: with 382 elements, 1146 source nodes, (c) reference solution.

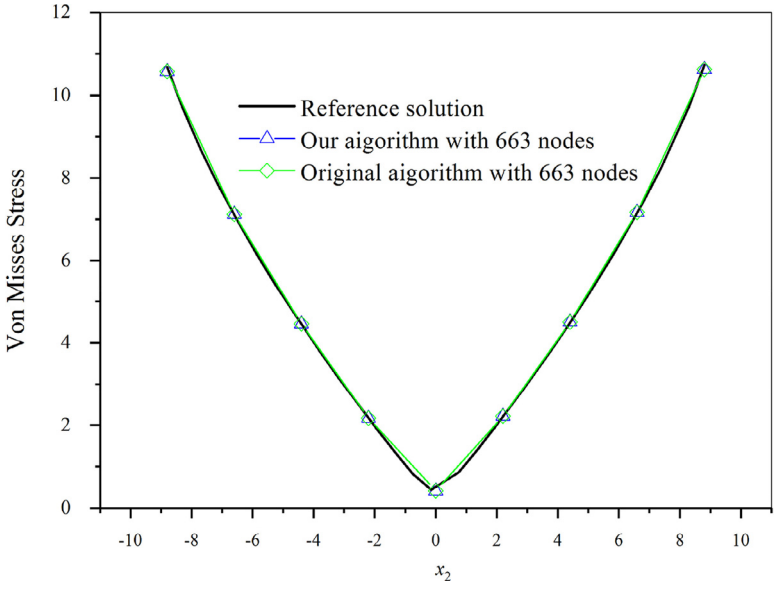

Fig. 12. Von-Mises stress along the edge *CD*.

Table 4
Time required for each adaptive step for Thick-wall cylinder problem.

Our algorithm				Original algorithm			
Adapt No.	NE	NS	Time(s)	Adapt No.	NE	NS	Time(s)
0	65	130	1.84	0	65	130	1.95
1	130	260	4.67	1	130	260	4.66
2	170	340	4.84	2	170	340	4.71
3	250	500	11.69	3	250	500	11.63
4	346	692	22.31	4	346	692	21.67
5	350	700	2.92	5	350	700	11.09

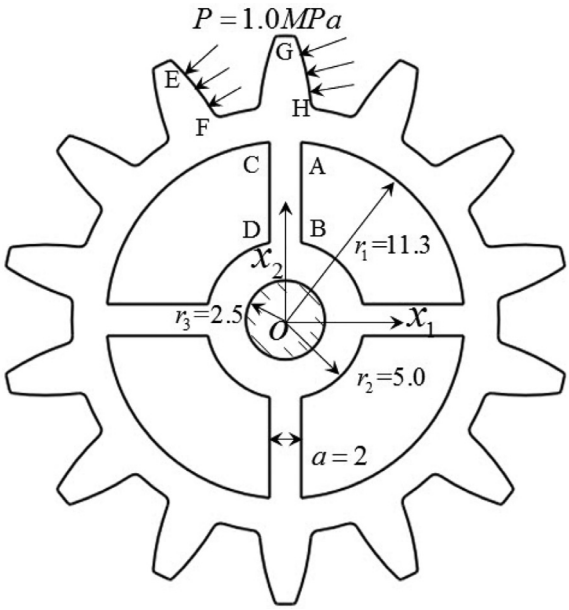

Fig. 13. Gear problems: geometric model.

5.3. Double open-end wrench

The third geometric model is a double open-end wrench (Fig. 10). The radius of two ports are r_1= 24.5 mm, r_2= 28.0 mm, and the distance between two ports is L = 193.5 mm, respectively. The width of the wrench is b = 24.0 mm. A radial pressure P = 1.0 *Mpa* is prescribed along the arc *AB*. The edges *CD, CE* and *DF* are fixed as shown in Fig. 10. Plane strain cases with Young's modulus E = 200.0Mpa and Poisson's ratio ν = 0.25 are considered in this problem.

In this example, the number of initial elements is n = 62, the physical variables are approximated by quadratic elements and the numerical result obtained by FEM with 2929,327 nodes is regarded as the reference solution. The comparison of the computation time between our method and original algorithm is list in Table 6. As reveal in this table, the benefit is considerable when our method is selected in the adaptive analysis. The values of the maximum Von-Mises stress listed in Table 7 show us that our algorithm converges gradually with the increment of the source point number. The comparison of the ultimate results of the Von-Mises stress is plotted in Fig. 11(a)–(c), and the values of Von-Mises stress along the edge *CD* in different methods are shown in Fig. 12. The obtained results can demonstrate that the values of Von-Mises stress calculated by our method and original algorithm are indistinguishable with the reference solution. Thus, our algorithm is suitable for stress concentration problem, and accurate results can also be obtained.

5.4. Gear problems

The fourth geometric model is a gear shown in Fig. 13. Pressures P = 1.0 Mpa are prescribed along the arc *EF* and arc *GH*. The inner hole is fixed as shown in Fig. 13. Plane strain cases with Young's modulus E = 200.0Mpa, Poisson's ratio ν = 0.25 are considered in this problem.

In this example, the initial elements number is n = 150, the physical variables are approximated by quadratic elements and the numerical result obtained by FEM with 1593,333 nodes is regarded as the reference solution. The computational time listed in Table 8 makes our algorithm competitive in adaptive analysis. The values of maximum Von-Mises stress are list in Table 9, from which we can know that the convergence of our algorithm still could be guaranteed. Fig. 14(a)–(c) show the ultimate Von-Mises stress, and Fig. 15 gives the values of Von-Mises stress along the edge *AB* with different methods. In summary, the nu-

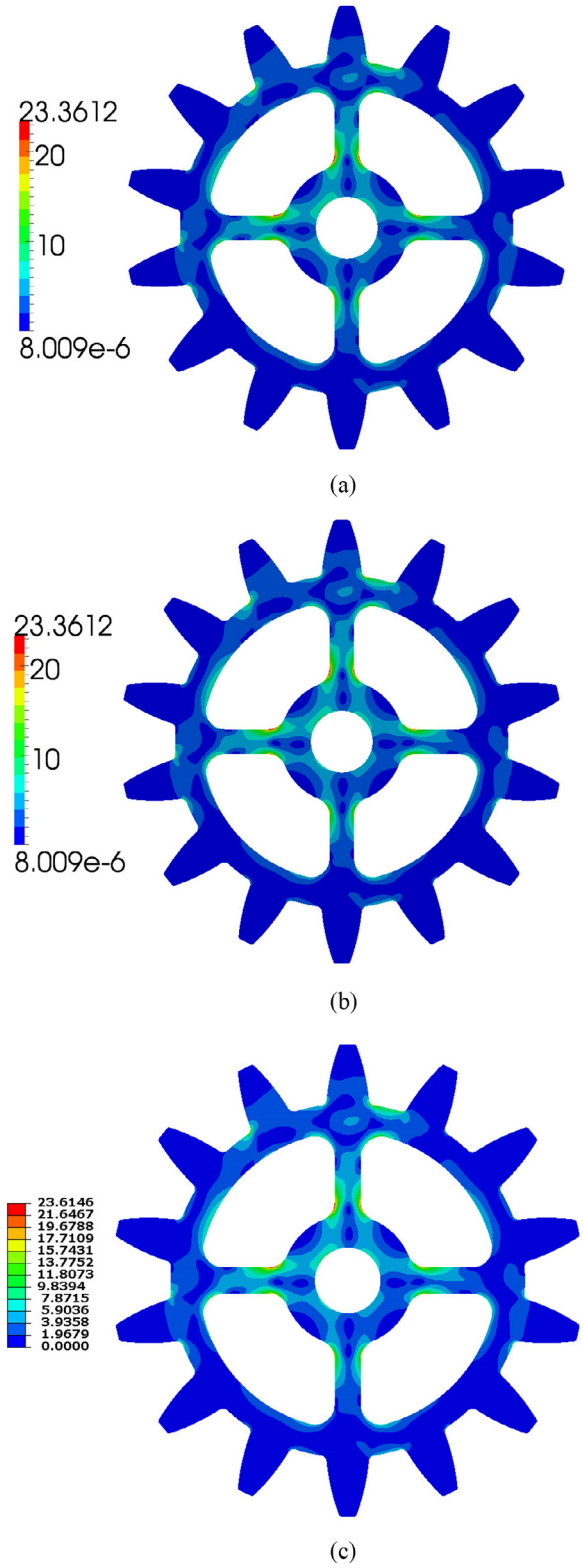

Fig. 14. Von-Mises stress for gear: (a) our method: with 351 elements, 1053 source nodes. (b) original method: with 351elements, 1,053source nodes (c) reference solution.

Table 5
Relative error of each adaptive step for Thick-wall cylinder problem (liner element).

Our	Adapt No.	0	1	2	3	4	5
algorithm	NE	65	130	170	250	346	350
	NS	130	260	340	500	692	700
	Err_u_r	4.92e-04	1.18e-04	8.47e-05	7.93e-05	7.85e-05	7.85e-05
	Err_σ_r	2.22e-02	1.31e-02	6.37e-03	3.20e-03	2.08e-03	2.08e-03
	Err_σ_θ	1.85e-02	1.18e-02	6.09e-03	3.15e-03	2.31e-03	2.31e-03
Original	Adapt No.	0	1	2	3	4	5
algorithm	NE	65	130	170	250	346	350
	NS	130	260	340	500	692	700
	Err_u_r	4.92e-04	1.18e-04	8.47e-05	7.93e-05	7.85e-05	7.85e-05
	Err_σ_r	2.22e-02	1.31e-02	6.37e-03	3.20e-03	2.08e-03	2.08e-03
	Err_σ_θ	1.85e-02	1.18e-02	6.09e-03	3.15e-03	2.31e-03	2.31e-03

Table 6
Time required for each adaptive step for double open-end wrench.

Our algorithm				Original algorithm			
Adapt No.	NE	NS	Time(*s*)	Adapt No.	NE	NS	Time(*s*)
0	62	186	3.7	0	62	186	3.7
1	124	372	9.9	1	124	372	9.9
2	155	465	9.1	2	155	465	9.1
3	178	534	9.9	3	178	534	9.7
4	201	603	12.4	4	201	603	12.1
5	219	657	11.2	5	219	657	13.7
6	221	663	7.8	6	221	663	9.9

Table 7
Numerical results of our algorithm, original algorithm and FEM for double open-end wrench.

	NE	155	178	201	219	221
Our	NS	465	534	603	657	663
algorithm	Max_Mises(Mpa)	122.54	122.63	122.62	122.62	122.62
Original	NE	155	178	201	219	221
algorithm	NS	465	534	603	657	663
	Max_Mises(Mpa)	122.54	122.63	122.62	122.62	122.62
FEM	NE	53,406	84,133	148,198	342,625	1461,420
	NS	108,111	169,888	298,559	688,498	2929,327
	Max_Mises(Mpa)	124.23	124.61	124.78	124.45	124.46

Table 8
Time required for each adaptive step for gear.

Our algorithm				Original algorithm			
Adapt No.	NE	NS	Time(*s*)	Adapt No.	NE	NS	Time(*s*)
0	150	450	17.2	0	150	450	17.1
1	300	900	59.4	1	300	900	58.3
2	350	1050	57.9	2	350	1050	54.7
3	351	1053	4.6	3	351	1053	36.8

Table 9
Numerical results of our algorithm, original algorithm and FEM for gear.

Our	NE	150	300	350	351
algorithm	NS	450	900	1050	1053
	Max_Mises(MPa)	22.24	23.35	23.36	23.36
Original	NE	150	300	350	351
algorithm	NS	450	900	1050	1053
	Max_Mises(MPa)	22.24	23.35	23.36	23.36
	NE	49,088	190,943	368,347	1593,333
FEM	NS	23,718	93,835	743,233	3199,769
	Max_Mises(MPa)	23.73	23.73	23.69	23.62

merical results reflect that the convergence and higher computational efficiency of our algorithm are available even when applied to complex geometries.

6. Discussion and conclusions

A new method suitable for solving variable dimension system of linear equations has been carried out successfully in 2D elasticity adaptive scheme with h-refinement. In this method, only the data of the coefficient matrix and its inverse in original mesh need to be reserved, and the Sherman-Morrison formula is employed two times in this process. Due to the introduction of the Sherman-Morrison formula in variable dimension system of linear equations, our method successfully avoids the trouble of repetitive computation, and has an outstanding performance in computational efficiency. Additionally, the accuracy of stress, displacement and traction inside the domain and on the boundary could

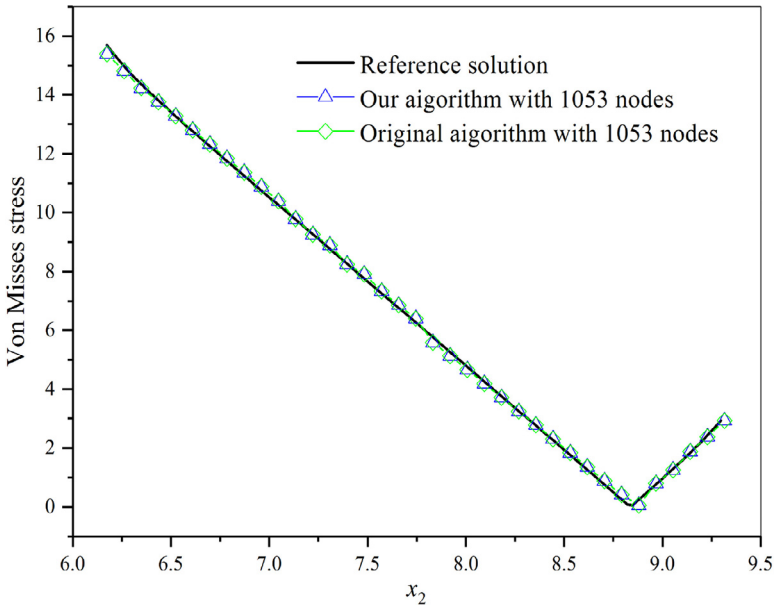

Fig. 15. Von-Mises stress along the edge AB.

also be guaranteed under the circumstance where a few grids are implemented mesh refinement. The validity and reliability of our method has been smoothly testified by different kinds of numerical examples with real engineering background in adaptive analysis.

In this study, our algorithm is implemented in pure h-version adaptive analysis. In the following work, we plan to apply our method to p-version or hp-version adaptive analysis. Extension of our algorithm to solving 3D elasticity problems and large-scale problems is already in progress.

Declaration of Competing Interest

None.

Acknowledgment

This work was supported by National Natural Science Foundation of China under grant numbers 11772125 and 11972010.

In the end of this paper, we want to give our respect to Mr. Rizzo for his contribution in BEM. Before his research, academic studies about solid mechanics often focused on indirect methods for BIE. In those methods, the unknown variables are not the relating boundary value of the field problems. Until the publication of the paper of Rizzo [34], the direct method arisen in the territory of solid mechanics. In this method, the boundary value is regarded as the unknown variable directly. Thereafter, the direct method was used in transient elastodynamic problem [35] and inclusion problem [36]. His works about the BIE discretization [37–39] promote the application of the BEM in computer technology. In [40], Rizzo presented an advanced boundary integral equation method, which can transform the volume integral to the boundary integral, therefore, avoiding the volume mesh in essence. All of the works, he did, promote the development of BEM and provide convenience and reference for the application and future studies.

References

[1] Alshoaibi AM, Hadi M, Ariffin MSA. An adaptive finite element procedure for crack propagation analysis. J Zhejiang Univ 2007;8(2):228–36.
[2] Wriggers P, Scherf O, Carstensen C. Adaptive techniques for the contact of elastic bodies. Recent Dev Finite Elem Anal 1994:78–86.
[3] Kulak RF. Adaptive contact elements for three-dimensional explicit transient analysis. Comput Methods Appl Mech Eng 1989;72(2):125–51.
[4] Carstensen C, Scherf O, Wriggers P. Adaptive finite elements for elastic bodies in contact. Soc Ind Appl Math 1999;20(5):1605–26.
[5] Zhao Z. A simple error indicator for adaptive boundary element method. Comput Struct 1998;68(5):433–43.
[6] Yuuki R, Cao GQ, Tamaki M. Efficient error estimation and adaptive meshing method for boundary element analysis. Adv Eng Softw 1992;15(3–4):279–87.
[7] Chen JT, Chen KH, Chen CT. Adaptive boundary element method of time-harmonic exterior acoustics in two dimensions. Comput Methods Appl Mech Eng 2002;191(31):3331–45.
[8] Bisschop J, Meeraus A. Matrix augmentation and partitioning in the updating of the basis inverse. Math Program 1977;13:241–54.
[9] Yeung YH, Pothen A, Halappanavar M, Huang Z. An augmented matrix formulation for principal submatrix updates with application to power grids. J Sci Comput 2017;39:809–27.
[10] Zhou AF, Hui KC, Tang YM, Wang CCL. An accelerated BEM approach for the simulation of deformable objects. Comput Aided Des Appl 2006;3(6):761–9.
[11] Sherman J, Morrison WJ. Adjustment of an inverse matrix corresponding to changes in the elements of a given column or a given row of the original matrix. Ann Math Stat 1949;20(4):621–2.
[12] Woodbury M. Inverting modified matrices. Memorandum Report 42 Statistical Research Group. (1950).
[13] Householder AS. A survey of some closed methods for inverting matrices. J Soc Ind Appl Math 1957;5(3):155–69.
[14] Press WH, Flannery BP, Teukolsky SA, Vetterling WT. Numerical recipes in C. Cambridge: Cambridge University Press; 1988.
[15] Hager WW. Updating the inverse of a matrix. SIAM Rev 1989;31(2):221–39.
[16] Powell MJD. A theorem on rank one modifications to a matrix and its inverse. Comput J 1969;12(3):288–90.
[17] Ghosh N, Mukherjee S. A new boundary element method for three dimensional problems in linear elasticity. Acta Mech 1987;67:107–19.
[18] Ezawa Y, Okamoto N. High speed boundary element contact stress analysis using a super computer. In: Proceedings of the 4th international conference on boundary element technology. Computational Mechanics Publications; 1989. p. 405–16.
[19] Akgun M A, Garcelon J H, Haftka R T. Fast exact linear and nonlinear structural reanalysis and the Sherman-Morrison-Woodbury formulas. Int J Numer Methods Eng 2001;50(7):1587–606.
[20] The Schur complement and its applications. Springer Science & Business Media; 2006.
[21] Kirsch U. Reanalysis of structures. Netherlands: Springer; 2008.
[22] Argyris JH, Broenlund OE, Roy JR, Scharpf DW. A direct modification procedure for the displacement method. AIAA J 2015;9(9):1861–4.
[23] Huang C, Verchery G. An exact structural static reanalysis method. Commun Numer Methods Eng 2010;13(2):103–12.
[24] Chen M. On the solution of circulant linear systems. Soc Ind Appl Math 1987.
[25] Pais MJ, Yeralan SN, Davis TA, Kim NH. An exact reanalysis algorithm using incremental Cholesky factorization and its application to crack growth modeling. Int J Numer Methods Eng 2012;91(12):1358–64.
[26] Ozer MB, Royston TJ. Application of Sherman–Morrison matrix inversion formula to damped vibration absorbers attached to multi-degree of freedom systems. J Sound Vib 2005;283(3–5):1235–49.
[27] Song Q, Liu Z, Wan Y, et al. Application of Sherman–Morrison–Woodbury formulas in instantaneous dynamic of peripheral milling for thin-walled component. Int J Mech Sci 2015;96:79–90.
[28] Huang S, Liu YJ. A new multidomain approach and fast direct solver for the boundary element method. Comput Mech 2017;60(3):379–92 No..
[29] Zhang JM, Qin XY, Han X, Li GY. A boundary face method for potential problems in three dimensions. Int J Numer Methods Eng 2009;80:320–37.

[30] Qin XY, Zhang JM, Li GY, Sheng XM, Song Q, Mu DH. An element implementation of the boundary face method for 3D potential problems. Eng Anal Bound Elem 2010;34:934–43.
[31] Wang XH, Zhang JM, Zhou FL. An adaptive fast multipole boundary face method with higher order elements for acoustic problems in three-dimension. Eng Anal Bound Elem 2013;37:114–52.
[32] Xie GZ, Zhang JM, Cheng H. A direct traction boundary integral equation method for three-dimension crack problems in infinite and finite domains. Comput Mech 2014;53(4):575–86.
[33] Zhang JM, Lin WC, Dong YQ. A double-layer interpolation method for implementation of BEM analysis of problems in potential theory. Appl Math Model 2017;51:250–69.
[34] Rizzo FJ. An integral equation approach to boundary value problems of classical elastostatics. Q Appl Math 1967;25(1):83.
[35] Cruse TA, Rizzo FJ. A direct formulation and numerical solution of the general transient elastodynamic problem - I. J Math Anal Appl 1968;22:244–59.
[36] Rizzo FJ, Shippy DJ. A formulation and solution procedure for the general non-homogeneous elastic inclusion problem. Int J Solids Struct 1968;4(12):1161–79.
[37] Rizzo FJ, Shippy DJ. A method for stress determination in plane anisotropic elastic bodies. J Compos Mater 1970;4(1):36–61.
[38] Rasmussen B, R, et al. A method of solution for certain problems of transient heat conduction. AIAA J 1970.
[39] Rizzo FJ, Shippy DJ. An application of the correspondence principle of linear viscoelasticity theory. SIAM J Appl Math 1971;21(2):321–30.
[40] Rizzo FJ, Shippy DJ. An advanced boundary integral equation method for three dimensional thermoelasticity. Int J Numer Methods Eng 2010;11(11):1753–68.